Application of boundary integral equation method to two-dimensional frictional contact problems
by Bhushan Wasudeo Dandekar

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering
Montana State University

Abstract:
In this thesis, the boundary integral equation technique is used to analyze the two-dimensional elastic contact problems with Coulomb friction. In this method, since only the boundary needs to be modeled, which contains the unknown contact zone, the data preparation time is considerably reduced. Also, since both tractions and displacements are retained as unknowns, the contact stresses are calculated directly.

Navier's equations for the bodies in contact are transformed into simultaneous linear equations using the standard BIE technique. Each body is divided into a potential contact zone and a non-contact zone. The equations obtained from the contact conditions are written explicitly such that a blocked coefficient matrix is obtained. An incremental-iterative technique is applied to the potential contact zone equations to find the correct contact area and the associated surface tractions.

Results for several frictionless and frictional problems are presented. In the case of frictionless problems the results are found to agree well with the analytical results. The frictional problems include an unloading problem and an advancing contact problem, for which no analytical solutions exist.

In most cases, equal size linear elements are found to give best results. The arrangement of stick-slip zones is affected by the node locations and large number of elements may be required to produce satisfactory results. It is shown that in the presence of friction, the problem needs to be solved by applying the load in a large number of load increments or by computing the load increment sizes such that each node pair just barely comes in contact and then dividing these load increments into smaller ones.

APPLICATION OF BOUNDARY INTEGRAL EQUATION METHOD TO TWO-DIMENSIONAL FRICTIONAL CONTACT PROBLEMS

by

Bhushan Wasudeo Dandekar

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

of

Doctor of Philosophy

in

Mechanical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

October 1987

APPROVAL

of a thesis submitted by

Bhushan Wasudeo Dandekar

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citation, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

10/30/87	R. Jerry Conant
Date	Chairperson, Graduate Committee

Approved for the Major Department

10-30-87	
Date	Head, Major Department

Approved for the College of Graduate Studies

11-17-87	Michael P. Malone
Date	Graduate Dean

STATEMENT OF PERMISSION TO USE

ACKNOWLEDGEMENTS

I would like to thank Dr. Jay Conant for his guidance and the many useful discussions concerning the project.

I would also like to thank Dr. Robert Brown and Dr. Bruce Suprenant for their guidance and encouragement during the course of my studies at Montana State University.

Finally, I would like to express a special thank you to my parents for their continuous support and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS-Continued

TABLE OF CONTENTS-Continued

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this thesis, the boundary integral equation technique is used to analyze the two-dimensional elastic contact problems with Coulomb friction. In this method, since only the boundary needs to be modeled, which contains the unknown contact zone, the data preparation time is considerably reduced. Also, since both tractions and displacements are retained as unknowns, the contact stresses are calculated directly.

Navier's equations for the bodies in contact are transformed into simultaneous linear equations using the standard BIE technique. Each body is divided into a potential contact zone and a non-contact zone. The equations obtained from the contact conditions are written explicitly such that a blocked coefficient matrix is obtained. An incremental-iterative technique is applied to the potential contact zone equations to find the correct contact area and the associated surface tractions.

Results for several frictionless and frictional problems are presented. In the case of frictionless problems the results are found to agree well with the analytical results. The frictional problems include an unloading problem and an advancing contact problem, for which no analytical solutions exist.

In most cases, equal size linear elements are found to give best results. The arrangement of stick-slip zones is affected by the node locations and large number of elements may be required to produce satisfactory results. It is shown that in the presence of friction, the problem needs to be solved by applying the load in a large number of load increments or by computing the load increment sizes such that each node pair just barely comes in contact and then dividing these load increments into smaller ones.

CHAPTER 1

INTRODUCTION

Many problems in mechanical system design involve parts that come in contact with each other. Bolted joints, bearings, gears and rings of a chain are some examples where the external forces are transmitted through the contact of two or more deformable bodies. When these parts come in contact under the action of external forces, a region of contact is formed and a pressure and shear stress distribution develops in that region. Normally, the pressure distribution that exists in the contact region is neglected and the transmission of forces is simplified into resultant forces which are important in the analysis of the structure. This simplification is justified by SAINT VENANT'S principle which roughly states that the pressure distribution in the contact region has a negligible effect at points far from the contact region.

In many cases, the load carrying capacity of the structure largely depends on the stress distribution in the contact area through which load transfer takes place. If the contact area is small, local regions of high stress may be formed, increasing the risk of crack initiation and propagation.

Wear of the mating parts along the contact surface is also an important factor in the study of contact mechanics. Due to surface roughness of the solids in contact, initial contact occurs only at the asperities. The load is therefore supported by the asperities which, even under small loads, deform plastically. These then break under an imposed tangential traction, resulting in material loss (Sarkar, 1976). A potential wear situation also exists whenever there is relative motion between two solids in contact. Sliding motion can cause adhesive and abrasive wear, while cyclic loading can cause fretting and fatigue wear thereby increasing the risk of fatigue failure. These friction and wear processes in materials depend on the nature of the true contact area (Sarkar,1976).

Development of different techniques to analyze contact problems with friction is thus important in order to increase understanding of the influence of different parameters that affect the nature of the contact area, and to enable the study of associated deformation and stress patterns.

The general problem of elastic bodies in contact remains one of the most difficult problems in solid mechanics. As the two deformable bodies come in contact under the action of external loads, the contact area changes progressively. In most cases the extent of contact is a priori unknown. This presents a geometric non-linearity even if the material is assumed to be linearly elastic in behavior.

When frictional forces are present between two contacting surfaces, the problem becomes even more difficult. In the presence of frictional forces the contact area may exhibit a region of adhesion and a region of slip which are unknown and often of a more complicated form than the contact area itself (Turner,1979). Frictional forces are inherently non-conservative in nature and dissipate energy over any load cycle applied to a system. Thus, when friction is considered, the stress distribution in the two deformable bodies in contact may depend on the entire history of loading.

The presence of frictional forces and geometric non-linearity has limited the analytical study of contact problems. Analytical solutions to frictionless problems are restricted to simple geometries while solutions to contact problems with friction are very rare. In fact, the existence of solutions to problems in which Coulomb's law of friction holds still remains to be proven (Campos et.al., 1982).

It is apparent that the only way of overcoming the above mentioned difficulties encountered in the study of contact problems is through the use of numerical techniques, and much of the current research in contact mechanics is directed towards developing such techniques (Torstenfelt, 1983, Cheng and Kikuchi, 1985, Bathe and Chaudhary, 1985).

In developing these numerical techniques the following factors are usually taken into account:

1. Contact problems may involve complex geometries and different friction laws.
2. Primary interest lies in obtaining the unknown contact area and the contact stresses.
3. Incremental-iterative solution techniques are essential to find the contact area and the contact stresses in the presence of frictional forces.

The finite element method has been used with considerable success to solve contact problems. The major drawback of the finite element method is its large data preparation time. Since both contacting bodies must be modeled entirely, the data preparation time and the execution time can be high; this is especially true when only the contact area and the contact stresses are of importance. In many cases the contact region is unknown and the model may have to be modified several times before reliable results are obtained. This means that substantial modeling experience is necessary to produce acceptable results.

Another numerical technique which has been developed to solve contact problems is the integral equation method. This method imposes certain restrictions on the shape of the body and is not as general as the finite element method. These two methods are discussed in detail in the following chapter.

In recent years, the boundary element method has emerged as a strong alternative to the finite element method. This method is well suited for the analysis of contact problems, and offers several advantages over current numerical methods. Since the governing differential equations are transformed into integral equations defined on the boundary, only the boundary need be modeled. By employing the boundary element method, the unknowns in the potential contact zone can be obtained. Once the equations are solved on the boundary, stresses and displacements inside the domain can be obtained by using the computed boundary tractions and displacements. Thus, the unnecessary computation time required in solving the equations over the entire domain is saved. Furthermore, reduction in the dimensionality of the problem makes the modeling simple and reduces data preparation time. This offers a significant advantage over the finite element method when solutions to contact problems are sought.

Approach

In this dissertation the boundary element analysis of two-dimensional contact problems is considered. The problems involve the contact of homogeneous isotropic linearly elastic bodies for which Coulomb's law of friction holds. In contact problems the solution of Navier's equations coupled with contact conditions is required. Navier's equations are transformed into integral equations defined on the boundary and then the standard collocation method is employed to obtain a system of simultaneous linear equations. These equations, along with the contact conditions provide sufficient equations to solve for unknown displacements and tractions in the contact zone. Each body is divided into a potential contact zone and a non-contact zone. For the potential contact zone, the equations obtained from the contact conditions are written explicitly so that the equations associated with the potential contact zone can be separated from those outside of the contact zone. After separating these equations, an incremental-iterative technique is applied to the potential contact zone equations to find the correct contact area and the associated surface tractions and displacements. The boundary unknowns outside of the contact zone can then be found by using the computed surface tractions and displacements. A Fortran computer code is developed to obtain numerical solutions for two-dimensional contact problems.

Solutions to several problems with and without friction are presented. For frictionless problems the results are compared with analytical solutions and in the case of frictional problems the results are compared with analytical solutions and other numerical methods whenever possible.

CHAPTER 2

LITERATURE REVIEW

Analytical Treatment of Contact Problems

The classical problem of frictionless contact between two elastic spheres was solved by Hertz in the 1880's. Since then problems concerning elastic bodies in contact have received considerable attention. Excellent accounts of the analytical techniques used in the treatment of contact problems, as well as the solutions to many interesting problems can be found in the works of Gladwell (1980) and de Pater and Kalker (1975).

The complexity of contact problems with friction has restricted most of the analytical treatment to frictionless problems. The work done by Spence (1973,1975,1986) accounts for most of the analytical solutions to frictional contact problems and provides valuable information on the role of friction in elastic contact problems. Spence (1973) shows that when a monotonically increasing load is applied to a rigid plane indentor in contact with an elastic half space, the region of contact divides into an inner adhesive region and an outer annulus of inward slip. The radius of adhesion is a function of material constants and the coefficient of friction, and is evaluated as an eigenvalue of a Fredholm integral equation. In a recent paper Spence (1986) considers the problem of a rigid plane indentor on an elastic half space subjected to a monotonically increasing normal force P, and a shear force Q. The ratio Q/mP, where m is the friction coefficient, is constant and is less than 1 during deformation. A solution exhibiting an unsymmetrical region of adhesion surrounded by regions of slip in opposite directions is obtained. The results include the effect of the shear force Q on the extent of the adhesion region.

Motivated by Spence's earlier work, Prasad and Dasgupta (1975), and Chiu and Prasad (1976) study the plane strain compression of an elastic rectangle by rigid rough planes and a pair of rigid rough punches, respectively. Numerical results consist of shearing traction, normal pressure and the zone of adhesion for various values of Poisson's ratio and the friction coefficient. The zone of adhesion in both problems is found to be independent of the magnitude of loading, but depends on the friction coefficient, Poisson's ratio and the geometrical parameters.

The problems discussed above have a stationary contact zone, i.e. the contact surface is known a priori. Frictional contact problems with an unknown contact zone have not yet been solved. In fact, as noted in Chapter 1, the existence of solutions to such problems where Coulomb's friction law is assumed to hold remains to be proven.

Numerical Treatment of Contact Problems

The vast majority of numerical solutions to contact problems have been obtained using finite element methods. Numerous articles have been written on the use of these methods to solve contact problems. Some of these articles, along with integral equation techniques, are reviewed in this section.

Chan and Tuba (1971) write the finite element equations for each body, separating the force vector into externally applied loads and loads in the possible contact zone. The external loads are then applied incrementally and at each increment an iterative scheme is used to determine the contact zone. Plane stress and plane strain problems using Coulomb friction are solved. A similar approach is used by Ohte (1973), and Tsuta and Yamaji (1973).

Francavilla and Zienkiewicz (1975) write the equations for displacements in the possible contact zone in terms of the contact pressure and the flexibility coefficients. Contact is brought about by assuming a portion of one body to be fixed, while a portion of the other body is given rigid movement. An expression is developed relating the pressure in the contact zone to the initial clearance and the rigid displacements. Once a suitable pressure distribution is found by an iterative scheme, the displacements in the assumed contact zone can be determined. A major advantage of this method is that only a relatively small system of equations is used during the iteration process. Their work is extended by Sachdeva, et.al. (1981) to include force boundary conditions and by Sachdeva and Ramakrishna (1981) to include Coulomb friction. The latter paper presents several examples of frictional contact for problems with proportional loading.

Schafer (1975) introduces special bond elements in the contact zone. With the assumption that a relation between the shear stress and the frictional deformation is known, the stiffness matrix for a bond element is developed by employing the principle of virtual work. The bond element stiffness matrix, along with the usual element stiffness matrix are then used to develop a finite element program. Various bond elements which can be used in construction technology are discussed. Stadter and Weiss (1979) construct a special gap element which replaces the open space between the two bodies, in the potential contact zone. The elastic modulus of a gap element is adjusted according to the normal strain in that element. Conditions for separation and contact are developed using normal strain. An iterative technique based on a stress invariance principle is developed to solve frictionless contact problems.

Gaertner (1977) addresses the plane frictional contact problem by utilizing triangular elements in which the six nodal unknowns are the normal and tangential components of displacement, rotation, tangential strain, and the normal and the tangential stress components. An incremental iterative procedure is used to solve the problem. Results for the frictionless case include Hertz's problem and contact between a connecting rod and a shaft.

Tseng and Olson (1981) use the mixed finite element method, in which both nodal displacements and stresses are retained as unknowns. Thus, contact conditions can be incorporated directly. An incremental loading procedure is used. The numerical results show rather poor agreement with the theoretical results.

Okamoto and Nakazawa (1979) construct contact elements for three dimensional frictional contact problems in which mixed contact conditions are considered. The use of contact elements allows the equilibrium equations and the geometrical boundary conditions to be treated as additional equations independent of the stiffness equations. As a result, only a part of the system of equations related to the contact zone is required to be solved at each load increment. The size of the load increments is calculated such that the contact status of only one node pair changes in each increment. Torstenfelt (1983, 1984) uses a similar approach for automatic load incrementation. His work concentrates on developing a method that can be implemented in any general purpose finite element computer program and does not use any special elements in the contact zone. Numerical results consist of the problem of an indentation of an elastic half space by a rigid circular punch. Both loading and unloading phases are considered. Results are in good agreement with the closed form solution for the loading phase and with the numerical results obtained by Turner (1979) for the unloading phase.

Campos, et. al. (1982) develop variational principles for the analysis of contact problems with Coulomb friction, in which the normal stress distribution in the contact zone is prescribed. First, a finite element approximation of the problem without friction is obtained. The normal contact pressure determined from this solution is then used as a first approximation for the corresponding problem with friction but with prescribed normal pressure. Pires and Oden (1983) extend this work to develop variational principles for incremental analysis of quasi-static contact problems with friction. Non-classical friction laws developed by Oden and Pires (1983) are used. Non-proportional oscillating loads are considered in analyzing the indentation of an elastic body by a rigid punch. Using variational principles, Turner (1979) has solved the problem of contact between a rigid circular cylinder and an isotropic homogeneous linearly elastic half space for the cases of frictionless contact, adhesive contact, and frictional contact for which analytical solutions are available. Turner also considers a problem of frictional unloading for which no analytical solutions are available.

Another numerical technique used to solve contact problems is the integral equation approach in which the normal component of the surface displacement of each body is written as the product of the normal pressure and an influence function, integrated over the entire contact zone. These integrals are then used in a kinematic constraint equation, resulting in a singular Fredholm integral equation of the first kind. Discretization of the integral equation is accomplished by dividing the contact region into a number of rectangular cells over which contact pressure is assumed to be uniform. The resulting system of linear algebraic equations is then solved numerically. For details of this technique the reader is referred to the work of Singh and Paul (1974), Paul and Hashemi (1980, 1981).

The main advantages of the integral equation approach lies in the simplicity of the resulting equations, and the relatively small amount of computational effort required to obtain a numerical solution. Only the potential contact zone is modeled so that the data preparation time is minimal. However, the influence functions are generally based on the Boussinesq function for the half space, so only the problems where geometries are reasonably approximated by a half space can be solved. In conformal contact the Boussinesq function cannot be assumed to be valid. Some alternate means of developing the influence functions, such as the empirical approach used by Paul and Hashemi (1980), must be found to solve the vast majority of problems. In these methods the shape of the bodies away from the contact region is disregarded, restricting their use in the analysis and design of machine elements where deformation and stress patterns of the entire body are often necessary. The integral equation approach is thus a special technique used for solving contact problems and does not have the generality offered by finite element and boundary element methods.

To date, the finite element method remains the most widely employed means of solving contact problems. Complex geometries can be handled routinely, and determination of the complete stress distribution in the body can be readily accomplished once the displacements in the contact zone are known. In recent years general purpose, efficient algorithms for stress analysis using finite element methods have been developed which can be extended to analyze contact problems with relative ease. However, since elements have to be defined over the entire body, modeling complex geometries can result in significant data preparation time. High execution time is inevitable since it is necessary to construct the complete stiffness matrix for both bodies. In the stiffness formulation of the finite element method the displacements are the primary unknowns. In the analysis of contact problems the contact forces are of primary interest, especially when friction is present. Except in the mixed formulation of Tseng and Olson, equilibrium conditions in the contact zone cannot be used directly in the formulation. If the flexibility formulation is used, then a matrix inversion step is required for each body. Introduction of special gap or contact elements makes it possible to retain both displacements and contact forces as primary unknowns but the total number of unknowns which necessarily have to be involved in the iterative process increases. In some of these formulations the system matrix is not symmetric (Okamoto and Nakazawa 1979), which nullifies one of the benefits of using finite element method.

In lieu of the above discussion the boundary integral equation method seems a natural choice for solving elastic contact problems. It retains the generality of the finite element method and offers the simplicity of the integral equation approach in modeling contact problems.

Boundary Integral Equation Method

To solve boundary value problems in classical elastostatics, the boundary integral equation (BIE) method uses the singular solution of the Navier equations. This solution along with the reciprocal work theorem yield an integral equation which provides a relation between boundary displacements and corresponding boundary tractions. The equation can be used to generate a set of simultaneous integral equations from suitable boundary data. In all but the simplest of problems, the integral equation cannot be solved analytically. However, suitable numerical techniques have been developed for solving these equations.

The original development of the BIE method is due to Jawson and Ponter (1963), who apply the method to elastic torsion problems. Their work is extended by Jawson (1963) and Symm (1963) to the solution of problems in potential theory.

Rizzo (1967) develops the BIE formulation for two dimensional problems in linear elasticity. Cruse (1969) extends it to three dimensional problems and also presents a numerical solution technique analogous to the finite element method. Computation of internal displacements and stresses is also included.

Lachat (1975) and Lachat and Watson (1975, 1976) develop a general numerical technique for two and three dimensional elastostatics problems. They introduce several improvements that provide a structure to the BIE method as a numerical technique, and aid in improving the accuracy and efficiency of the BIE method. They represent both geometry and boundary data parametrically in which the variation of these quantities is expressed in terms of the nodal values and shape functions of intrinsic coordinates. This representation allows use of Gaussian quadrature for efficient numerical integration. The order of the quadrature formula to be used is determined for each segment based on calculated error bounds for the formulae and the rapidity of the variation of the integrand within the segment. Examples are presented using linear, quadratic and cubic variation for the boundary data.

The boundary integral equation method has also been extended to solve inelastic problems. Solutions techniques for applying the BIE method to problems with constitutive equations representing various material responses are now being developed. For current developments and research trends in the BIE method the reader is referred to Brebbia et. al. (1984).

The BIE method offers several important advantages over the finite element method and the integral equation method for the numerical solution of contact problems. As mentioned previously, the data preparation time is reduced considerably relative to the finite element methods, since only the boundary needs to be modeled. Input data can be easily changed and less experience is required by the analyst to produce acceptable results.

Since both tractions and displacements are retained as unknowns, the contact equations can be written explicitly and hence, different friction laws can be incorporated easily. Tractions and displacements are computed with the same accuracy and surface tractions in the contact zone, often the most important part of the solutions, are calculated directly.

The BIE method does not impose any restrictions on the shape of the boundary, and therefore complex geometries can be handled routinely. The half-space restrictions on the shape of the body imposed by the integral equation method do not apply.

CHAPTER 3

FORMULATION OF BOUNDARY INTEGRAL EQUATION

In this chapter the basic equations of boundary integral equations (BIE) are developed within the context of the linear theory of elasticity. The development of BIE is now well documented; for details the reader is referred to Brebbia et.al. (1984).

In what follows, all work is referred to a Cartesian coordinate system. The notation used is the usual Cartesian tensor notation, with implied summation on repeated indices, and partial differentiation denoted by comma-index.

The analysis is restricted to the classical elastostatics problems, which is based on the following assumptions:

1. Linear stress-strain relations.
2. Small deformation theory holds, i.e. equilibrium equations can be referred to the undeformed configuration.

In small deformation theory the strain tensor components ε_{IJ} are expressed in terms of the displacement vector components u_I as

$$\epsilon_{IJ} = 1/2(\partial u_I/\partial X_J + \partial u_J/\partial X_I) \tag{3.1}$$

Consider an elastic body occupying a domain Ω bounded by a surface Γ. If σ_{IJ} are the stress tensor components and B_I

are the body force vector components then the equilibrium equations are

$$\partial\sigma_{IJ}/\partial X_J + B_I = 0 \qquad \text{in } \Omega. \tag{3.2}$$

Moment equilibrium can be used to show that the stress tensor is symmetric, i.e.,

$$\sigma_{IJ} = \sigma_{JI}. \tag{3.3}$$

For a homogeneous isotropic elastic medium the stress-strain relations are

$$\sigma_{IJ} = 2G\epsilon_{IJ} + 2G\upsilon/(1-2\upsilon)\epsilon_{KK}\delta_{IJ}. \tag{3.4}$$

G and ν designate the shear modulus and the Poisson's ratio, respectively. δ_{IJ} is the Kronecker delta whose properties are

$$\begin{aligned} \delta_{IJ} &= 0 \qquad I \neq J \\ &= 1 \qquad I = J. \end{aligned} \tag{3.5}$$

Using Eq. (3.4) in Eq. (3.2), and using Eq. (3.1) in the result, the usual Navier's equations are obtained

$$Gu_{I,JJ} + G/(1-2\upsilon)u_{J,JI} + B_I = 0 \qquad \text{in } \Omega. \tag{3.6}$$

A particular solution to these partial differential equations must satisfy, in addition, the boundary conditions for displacements on the surface Γ_u and tractions on Γ_t, respectively given as

$$u_I(x) = q_I(x) \qquad x \in \Gamma_u,$$

and

$$t_I(x) = \sigma_{IJ}n_J = p_I(x) \qquad x \in \Gamma_t. \tag{3.7}$$

where $\underline{n}$ is the unit outward normal to the surface Γ and $\underline{t}$ is the traction vector at a point $x \in \Gamma$. $q_I(x)$ are the prescribed displacements on the surface Γ_u, and $p_I(x)$ are the prescribed tractions on Γ_t. In a well posed problem at any given point on the boundary either the displacements $u_I(x)$ or the tractions $t_I(x)$ are known. The problem is to find a solution to Eq. (3.6) subject to the boundary conditions (3.7).

Plane State of Strain or Stress

In solving two-dimensional problems in elasticity, if the X_3 component of displacement vector $\underline{u}$ is constant and if the displacements u_1, u_2 are functions of X_1 and X_2 only, the body is said to be in plane strain state parallel to the X_1, X_2-plane. In plane strain the following conditions hold:

$$\partial u_1/\partial X_3 = \partial u_2/\partial X_3 = 0 \; ; \; u_3 = \text{constant}. \tag{3.8}$$

The Navier's equations for plane strain (I, J =1,2) are given by Eq. (3.6).

If the stress components in the X_3 direction vanish everywhere,

$$\sigma_{31} = \sigma_{32} = \sigma_{33} = 0, \tag{3.9}$$

the state of stress is said to be plane stress parallel to the X_1, X_2-plane. Substituting Eq. (3.4) and (3.9) in Eq. (3.2) the Navier's equations for plane stress are obtained (Fung, 1965),

$$G u_{I,JJ} + G(1+\upsilon)/(1-\upsilon)\, u_{J,JI} + B_I = 0 \quad \text{in } \Omega. \tag{3.10}$$

Eq. (3.6) is identical to Eq. (3.10), if ν in Eq. (3.6) is replaced by $\nu/(1+\nu)$. Hence, a plane strain formulation may be used to solve plane stress problems if ν is replaced by $\nu/(1+\nu)$.

Somigliana's Identity

Boundary integral equations for elasticity can be developed using the method of weighted residuals (Brebbia, et.al., 1984). These methods usually bear no physical relation to the problem, therefore the equations presented here are derived via Somigliana's identity, which can be deduced from Betti's reciprocal work theorem. This theorem states (Sokolnikoff, 1956):

If an elastic body is subjected to two systems of body and surface forces, then the work that would be done by the first system t_I, B_I in acting through the displacements u'_I due to the second system of forces is equal to the work that would be done by the second system t'_I, B'_I in acting through the displacements u_I due to the first system of forces.

The theorem can be expressed mathematically as

$$\int_\Omega B_I u'_I \, d\Omega + \int_\Gamma t_I u'_I \, d\Gamma = \int_\Omega B'_I u_I \, d\Omega + \int_\Gamma t'_I u_I \, d\Gamma \, . \tag{3.11}$$

where Ω is the region bounded by Γ.

Let the body force components B'_I be the unit loads applied at a source point $\underline{\xi} \in \Omega$ in each of the coordinate directions given by unit vectors e_I. Then the body force distribution can be expressed as

$$B'_I = \Delta(\underline{\xi} - \underline{x})e_I \, . \tag{3.12}$$

where $\Delta(\underline{\xi} - \underline{x})$ is the Dirac delta function, $\underline{x} \in \Omega$ is an observation point, and the underscore indicates a vector. The Dirac delta function has the following properties:

$$\Delta(\underline{\xi} - \underline{x}) = 0 \qquad \underline{x} \neq \underline{\xi} \, ,$$

$$\begin{aligned} \int_\Omega g(\underline{x})\, \Delta(\underline{\xi} - \underline{x}) &= g(\underline{\xi}) & \underline{\xi} \in \Omega \\ &= 0 & \underline{\xi} \notin \Omega \, . \end{aligned} \tag{3.13}$$

Therefore the first integral on the right hand side in Eq. (3.11) then becomes

$$\int_\Omega B'_I u_I \, d\Omega = \int_\Omega e_I \, u_I(\underline{x})\Delta(\underline{\xi} - \underline{x}) \, d\Omega = u_I(\underline{\xi})e_I \, . \tag{3.14}$$

If a unit load is applied successively in each of the coordinate directions, then the displacements u'_J and the tractions t'_J can be represented in the following form:

$$\begin{aligned} u'_J &= U_{IJ}(\underline{\xi}, \underline{x})\, e_I \, , \\ t'_J &= T_{IJ}(\underline{\xi}, \underline{x})\, e_I \, . \end{aligned} \tag{3.15}$$

where $U_{IJ}(\underline{\xi},\underline{x})$ and $T_{IJ}(\underline{\xi},\underline{x})$ represent the displacements and tractions, respectively in the J direction at $\underline{x}$ corresponding to an unit point force acting in the I direction at $\underline{\xi}$. Substituting Eq. (3.14) and (3.15) into Eq. (3.11), the displacement components u_I at $\underline{\xi}$ can be written in the following form

$$\begin{aligned} u_I(\underline{\xi}) = \int_\Gamma U_{IJ}(\underline{\xi}, \underline{x})\, t_J(\underline{x})\, d\Gamma(\underline{x}) &- \int_\Gamma T_{IJ}(\underline{\xi}, \underline{x})\, u_J(\underline{x})\, d\Gamma(\underline{x}) \\ &+ \int_\Omega U_{IJ}(\underline{\xi}, \underline{x})\, B_J(\underline{x})\, d\Omega(\underline{x}) \, . \end{aligned} \tag{3.16}$$

The above equation is Somigliana's identity for displacements. This identity is the starting point for the BIE method. The terms $U_{IJ}(\underline{\xi},\underline{x})$ are the fundamental solutions of Navier equations, i.e., they satisfy

$$Gu_{I,JJ} + G/(1-2\upsilon)u_{J,JI} + \Delta(\underline{\xi} - \underline{x})\, e_I = 0 . \tag{3.17}$$

<u>Fundamental Solutions</u>

These fundamental solutions, or free-space Green's functions, are necessary for the boundary integral equation formulation. They are defined over the entire region Ω, except at the source point $\underline{\xi} \in \Omega$, but are not required to satisfy boundary conditions on Γ. If the fundamental solution is forced to satisfy the boundary conditions then the Green's function for the specific problem is obtained.

The solution of Eq. (3.17) for three-dimensional problems was first obtained by Kelvin (Love, 1929) as

$$U_{IJ}(\underline{\xi}, \underline{x}) = (1+\upsilon)/8\pi E(1-\upsilon)r\ [(3-4\upsilon)\delta_{IJ} + r_{,I}r_{,J}] \tag{3.18}$$

where E is the Young's modulus and its relationship between G and ν is

$$E = 2G(1+\upsilon) . \tag{3.19}$$

$r = r(\underline{\xi}, \underline{x})$ is the distance between source point and observation point:

$$r_I = X_I(\underline{x}) - X_I(\underline{\xi}) ,$$
$$r = (r_I r_I)^{1/2} . \tag{3.20}$$

The notation $r,_I$ indicates differentiation with respect to X_I

$$r,_I = \partial r_I/\partial X_I = r_I/r \,. \tag{3.21}$$

Eq. (3.18) is the solution for the displacements at any point $\underline{x}$ due to a unit point load at a point ξ in an infinite elastic medium; note that the solution is singular at the source point, $\underline{x} = \underline{\xi}$. Using the stress-strain relation Eq. (3.4) and Eq. (3.18) a similar expression for tractions can be obtained, that is:

$$\begin{aligned} T_{IJ}(\underline{\xi}, \underline{x}) \; = \; & \{1/8\pi(1-\upsilon)r^2\} \, [(1-2\upsilon)(r,_I n_J - r,_J n_I) \\ & - n_m r,_m\{(1-2\upsilon)\delta_{IJ} + 3r,_I r,_J\}] \,. \end{aligned} \tag{3.22}$$

Note that $U_{IJ}(\underline{\xi},\underline{x})$ and $T_{IJ}(\underline{\xi},\underline{x})$ are symmetrical in $\underline{\xi}$ and $\underline{x}$.

By integrating this solution along an infinite line, the solution for a line load of unit intensity in an infinite elastic medium is obtained (Danson, 1983) and this modified solution can be used for plane strain problems. In integrating Eq. (3.18) the displacements are found to be infinite since the line load extends to infinity. Even though the displacements are infinite the strains will be finite and by finding the strain field, a fundamental solution for the plane strain problem is then obtained. These expressions are

$$U_{IJ}(\underline{\xi},\underline{x}) = (1+\upsilon)/4\pi E(1-\upsilon)r \;\; [(3-4\upsilon)\ln(1/r)\delta_{IJ} + r,_I r,_J], \tag{3.23}$$

and

$$T_{IJ}(\xi, x) = 1/4\pi(1-\upsilon)r\,[(1-2\upsilon)(r_{,I}n_J - r_{,J}n_I) - n_m r_{,m}\{(1-2\upsilon)\delta_{IJ} + 2r_{,I}r_{,J}\}]\,. \qquad (3.24)$$

The plane strain equations are valid for plane stress problems if ν and E are replaced by

$$\upsilon' = \upsilon/(1+\upsilon)\,,$$
$$E' = E(1+2\upsilon)/(1+\upsilon^2)\,. \qquad (3.25)$$

Boundary Integral Equation

Eq. (3.16) gives displacements at a point $\underline{\xi}$ in Ω, in terms of surface integrals of boundary displacements and tractions, and a volume integral of the body force distribution. It provides a means of determining displacements on the interior once the boundary tractions and displacements are known. If the source point $\underline{\xi} \in \Omega$ is moved to the boundary, an expression containing only boundary data will result, which can be used to obtain the boundary unknowns.

Care must be taken in evaluating Eq. (3.16) as $\underline{\xi}$ is moved to the boundary since Gauss's divergence theorem, which is used in deriving Betti's theorem (Eq. 3.11), requires all functions to be continuous. Since the kernels $T_{IJ}(\underline{\xi},\underline{x})$ and $U_{IJ}(\underline{\xi},\underline{x})$ are singular at the source point, some material must be added where $\underline{\xi} \in \Omega$ intersects the boundary so that the source point can be enclosed in portion of a small sphere of radius δ (Fig. 1). Eq. (3.16) is evaluated over the boundary $\Gamma-\Gamma^*+\Gamma_\delta$ and the domain $\Omega' = \Omega + \Omega_\delta$, as $\delta \rightarrow 0$:

$$\delta_{IJ} u_J(\underline{\xi}) = \mathrm{Lim}\ \delta \rightarrow 0\ \{\int_{\Gamma-\Gamma^*+\Gamma\delta} U_{IJ}(\underline{\xi},\underline{x}) t_J d\Gamma(\underline{x})$$

$$-\int_{\Gamma-\Gamma^*+\Gamma\delta} T_{IJ}(\underline{\xi},\underline{x}) u_J(\underline{x}) d\Gamma(\underline{x}) + \int_{\Omega'} U_{IJ}(\underline{\xi},\underline{x}) B_J(\underline{x}) d\Omega(\underline{x})\}. \qquad (3.26)$$

As $\delta \rightarrow 0$, $\Omega' \rightarrow \Omega$; therefore the integral with body force term B_J does not change. Consider the second integral in the above equation :

$$\mathrm{Lim}\delta\rightarrow 0 \int_{\Gamma-\Gamma^*+\Gamma\delta} T_{IJ}(\underline{\xi},\underline{x})\, u_J(\underline{x})\, d\Gamma(\underline{x}) =$$

$$\mathrm{Lim}\delta\rightarrow 0\ [\int_{\Gamma-\Gamma^*} T_{IJ}(\underline{\xi},\underline{x}) u_J(\underline{x}) d\Gamma(\underline{x}) + \int_{\Gamma\delta} T_{IJ}(\underline{\xi},\underline{x}) u_J(\underline{x}) d\Gamma(\underline{x})]. \qquad (3.27)$$

As $\delta \rightarrow 0, \Gamma - \Gamma^* \rightarrow \Gamma$; therefore only the second integral in Eq. (3.27) needs to be evaluated. The term $u_J(\underline{x})$ can be taken outside the integral since $u_J(\underline{x}) \rightarrow u_J(\underline{\xi})$ as $\delta \rightarrow 0$. Thus the integral to be evaluated is

$$I_{IJ} = \mathrm{Lim}\ \delta \rightarrow 0 \int_{\Gamma\delta} T_{IJ}(\underline{\xi},\underline{x}) d\Gamma(\underline{x})\,. \qquad (3.28)$$

where the kernel $T_{IJ}(\underline{\xi},\underline{x})$ is given by the Eq. (3.22). With the use of polar coordinates, this integral can be evaluated in closed form for the two dimensional case (Danson, 1983). Its value in general depends on the shape of the boundary at the point $\underline{\xi}$ (Fig. 1); for smooth boundary $I_{IJ} = -1/2\, \delta_{IJ}$. For three-dimensional case the form of I_{IJ} is more complex, and is taken in the Cauchy principle value sense. This does not pose any problems, as later it will be shown that I_{IJ} need not be computed explicitly (see Appendix A).

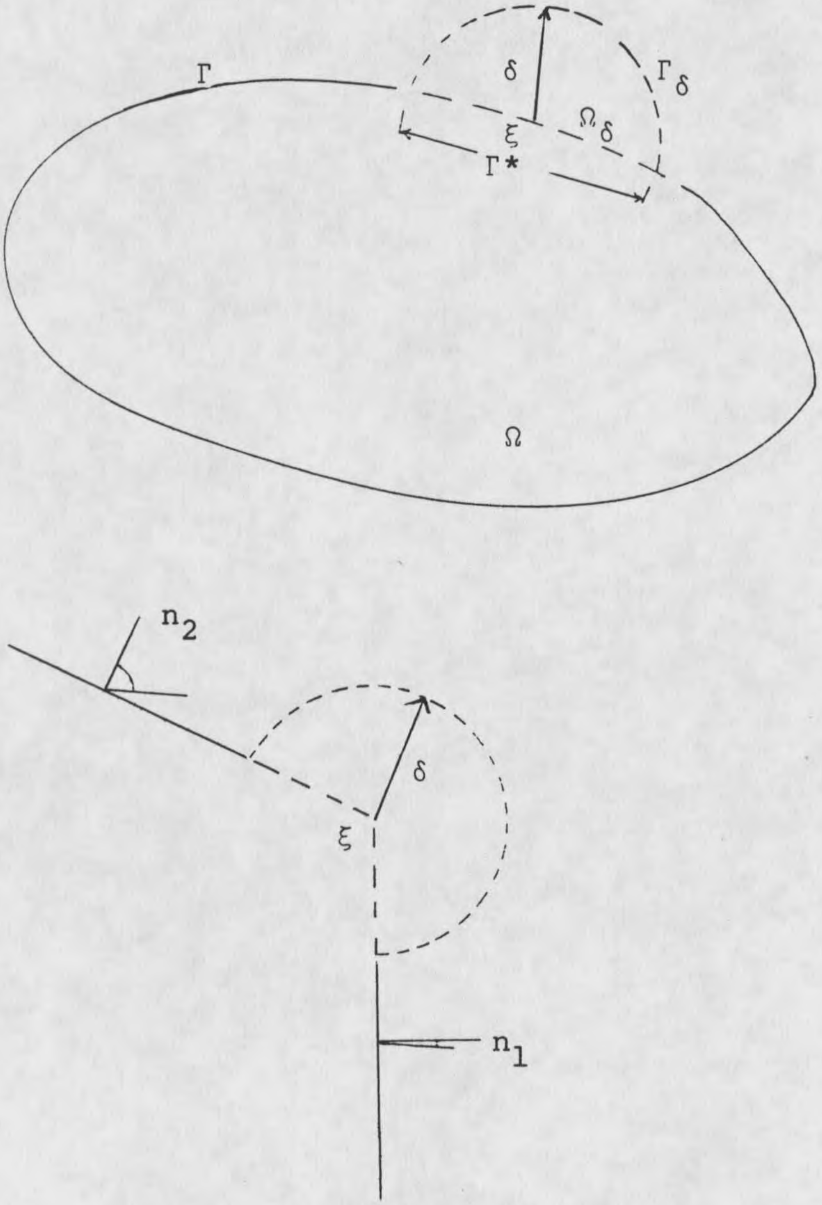


Figure 1. Source point ξ on the boundary, surrounded by a portion of a sphere of radius δ.

Now, consider the first integral in Eq. (3.26). With a similar argument as $\delta \rightarrow 0$, $\Gamma - \Gamma^* \rightarrow \Gamma$, hence the integral to be evaluated is

$$J_{IJ} = \lim_{\delta \rightarrow 0} \int_{\Gamma\delta} U_{IJ}(\underline{\xi},\underline{x})d\Gamma(\underline{x}). \tag{3.29}$$

With the use of polar coordinates for the two-dimensional case and spherical coordinates for the three-dimensional case, J_{IJ} can be shown to be equal to zero.

Thus, when the source point $\underline{\xi}$ is on the boundary Eq. (3.26) becomes

$$C_{IJ}u_J(\underline{\xi}) + \int_{\Gamma} T_{IJ}(\underline{\xi},\underline{x})u_J(\underline{x})d\Gamma(\underline{x}) = \int_{\Gamma} U_{IJ}(\underline{\xi},\underline{x})t_J d\Gamma(\underline{x})$$

$$+ \int_{\Omega} U_{IJ}(\underline{\xi},\underline{x})B_J(\underline{x})d\Omega(\underline{x}). \tag{3.30}$$

where C_{IJ} is given by the equation

$$C_{IJ} = \delta_{IJ} + I_{IJ} = \delta_{IJ} + \lim_{\delta \rightarrow 0} \int_{\Gamma\delta} T_{IJ}(\underline{\xi},\underline{x})d\Gamma(\underline{x}). \tag{3.31}$$

Eq. (3.31) provides a constraint between boundary displacements, boundary tractions, and body forces. It can be solved numerically to obtain the boundary unknowns. The numerical solution technique is presented in Chapter 4.

<u>Internal Points</u>

Once the values of traction and displacement over the entire boundary are known, the internal displacements can be computed using Somigliana's identity, Eq. (3.16). Internal stresses can be computed by differentiating Eq. (3.16) and then using strain-displacement relation (3.1) along with the stress-strain relation (3.4). Thus,

$$\sigma_{ij}(\underline{\xi}) = \int_{\Gamma} D_{ijk}(\underline{\xi},\underline{x})t_k(\underline{x})\, d\Gamma(\underline{x}) - \int_{\Gamma} S_{ijk}(\underline{\xi},\underline{x})u_k(\underline{x})d\Gamma(\underline{x})$$

$$+ \int_{\Omega} S_{ijk}(\underline{\xi},\underline{x})B_k(\underline{x})d\Gamma(\underline{x})\,. \qquad (3.32)$$

where

$$S_{ijk} = 1/4\alpha\pi(1-\upsilon)r^{\alpha}\{(1-2\upsilon)(r_{,i}\delta_{jk} + r_{,k}\delta_{ij}) + \beta r_{,i}r_{,j}r_{,k}\}. \qquad (3.33)$$

and

$$D_{ijk} = E/4\alpha\pi(1-\upsilon)^2 r^{\beta}\{\beta\partial r/\partial n[(1-2\upsilon)\delta_{ij}r_{,k} + \upsilon(\delta_{ik}r_{,J} + \delta_{jk}r_{,I} - \gamma r_{,i}r_{,j}r_{,K}] + \beta\upsilon(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k}) + (1-2\upsilon)(\beta n_k r_{,i}r_{,j} + n_j\delta_{ik} + n_i\delta_{jk}) - (1-4\upsilon)n_k\delta_{ij}\}. \qquad (3.34)$$

where $\alpha = 2, 1$; $\beta = 3, 2$; $\gamma = 5,4$ for three-dimensional and two-dimensional problems, respectively. Direct computation of D_{IJK} and S_{IJK} is not difficult. Danson (1983) has shown the necessary mathematical steps.

Eq. (3.30) is valid for two or three-dimensional problems and, since the body force distribution is always known, provides a boundary integral equation from which the unknown boundary quantities can be obtained once the boundary conditions are specified.

CHAPTER 4

NUMERICAL SOLUTION OF BOUNDARY INTEGRAL EQUATION

A general computational procedure for the solution of two-dimensional problems using the BIE method is presented. For these problems the collocation method is widely used (Lachat, 1975; Jawson and Symm, 1977). The collocation method can be viewed as a weighted residual method with the Dirac delta function as the weighting function. Thus the integral equations are satisfied at the collocation points. One can also apply the least squares method to solve these equations, but the conditioning of the integral equation, in general, becomes worse than the original equation (Baker, 1977; Delves and Mohomed, 1985), and is therefore usually not recommended.

The process proceeds as follows:

1. The boundary Γ is divided into a series of elements over which displacements and tractions are approximated using piecewise polynomials. This allows numerical evaluation of the integrals.

2. Satisfaction of the integral equation in its discretised form is then required at certain nodal points in each element. The integrals are computed numerically, resulting in a system of algebraic equations involving nodal tractions and/or displacements, some of which are known from the boundary conditions.

3. Boundary conditions are imposed and the resulting system of equations is then solved to obtain the unknown boundary tractions and displacements.

Before giving the details of the numerical procedure, body forces need to be mentioned. Since the term involving body forces is a volume integral, the domain Ω has to be divided into a number of elements for numerical integration of this term. This increases the amount of data preparation and partly nullifies the advantages offered by the BIE method. However in many practical problems the body forces are not significant and can be omitted from the analysis. The numerical formulation described in this chapter does not include the body force term.

If the source point with coordinates ξ_i is designated by P, and the observation point with coordinates x_i is designated by Q, then for two-dimensional problems, in the absence of body forces, Eqs. (3.30), (3.23) and (3.24) become:

$$C_{IJ}u_J(P) + \int_\Gamma T_{IJ}(P,Q)u_J(Q)d\Gamma(Q) = \int_\Gamma U_{IJ}(P,Q)t_J(Q)d\Gamma(Q), \quad (4.1)$$

$$U_{IJ}(P, Q) = \{(1+\upsilon)/4\pi E(1-\upsilon)r\}[(3-4\upsilon)\ln(1/r)\delta_{IJ} + r_{,I}r_{,J}], \quad (4.2)$$

$$T_{IJ}(P, Q) = \{1/4\pi(1-\upsilon)r\}\ [(1-2\upsilon)(r_{,I}n_J - r_{,J}n_I)$$
$$- n_m r_{,m}\{(1-2\upsilon)\delta_{IJ} + 2r_{,I}r_{,J}\}]. \quad (4.3)$$

where $P = P(\underline{\xi})$ and $Q = Q(\underline{x})$.

In Eq. (4.1) the integrals to be evaluated have the following form

$$\Psi_I(P) = \int_\Gamma G_{IJ}(P,Q)\, f_J(Q)\, d\Gamma(Q). \quad (4.4)$$

where the function $G_{IJ}(P,Q)$ is a known kernel and the function $f_J(Q)$, which may be unknown, represents either boundary tractions or displacements. For the purpose of numerical evaluation of the integrals, the boundary is divided into K smooth intervals or "boundary elements" so that $\Gamma = \Gamma_1+\Gamma_2+..+\Gamma_K$, and

$$\Psi_I(P) = \sum_{i=1}^{K} \int_{\Gamma i} G_{IJ}(P, Q)\, f_J(Q)\, d\Gamma(Q), \tag{4.5}$$

where i denotes the i^{th} element of Γ. If over each interval the function $f_J(Q)$ is approximated by piecewise polynomials then the integral (4.4) can be evaluated either analytically or numerically. This provides a simple procedure for the evaluation of an integral of the form (4.4). Adopting this procedure and dividing the boundary into K intervals Eq. (4.1) is written in the form:

$$C_{IJ} u_J(P) + \sum_{i=1}^{K} \int_{\Gamma i} T_{IJ}(P, Q)\, u_J(Q) d\Gamma(Q) = \sum_{i=1}^{K} \int_{\Gamma i} U_{IJ}(P, Q)\, t_J(Q) d\Gamma(Q). \tag{4.6}$$

The Parametric Representation of Geometry

Next, each of the K segments is represented in terms of its natural coordinates (Fig 2). The global coordinates x_i, of an arbitrary point Q in the element are determined in terms of the shape functions $N^{\beta}(\eta)$, and the global coordinates of the boundary nodes,

$$X_I(\eta) = \sum_{\beta=1}^{L} N^{\beta}(\eta)\, X^{\beta}_I , \qquad -1 \le \eta \le 1 . \tag{4.7}$$

Therefore, Eq. (4.6) takes the following form:

$$C_{IJ}u_J(P) + \sum_{k=1}^{K} \int_{\Gamma k} T_{IJ}(P,Q(\eta))\, u_J(Q(\eta))\, J(\eta)\, d\eta \approx \sum_{k=1}^{K} \int_{\Gamma k} U_{IJ}(P,Q(\eta))\, t_J(Q(\eta))\, J(\eta)\, d\eta . \tag{4.8}$$

where k is the element number and the Jacobian J is

$$J(\eta) = [(dX_1/d\eta)^2 + (dX_2/d\eta)^2]^{1/2} \tag{4.9}$$

where

$$dX_I/d\eta = \sum_{\beta=1}^{L} (dN^{\beta}(\eta)/d\eta) X^{\beta}_I . \qquad -1 \le \eta \le 1 . \tag{4.10}$$

The geometry can be mapped by linear, quadratic, or cubic shape functions which are defined as follows:

Constant: $N^1(\eta) = 1$

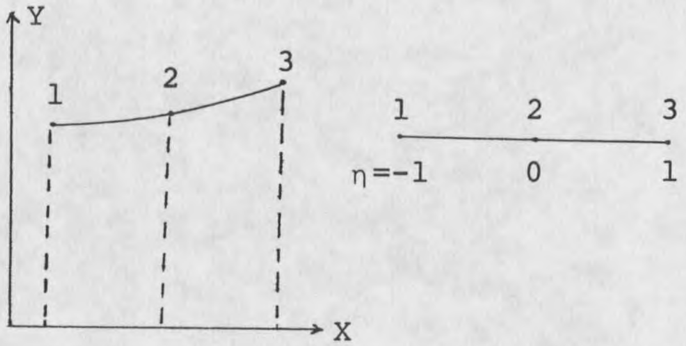
Linear: $N^1(\eta) = -1/2(\eta - 1)$

$N^2(\eta) = 1/2(\eta + 1)$

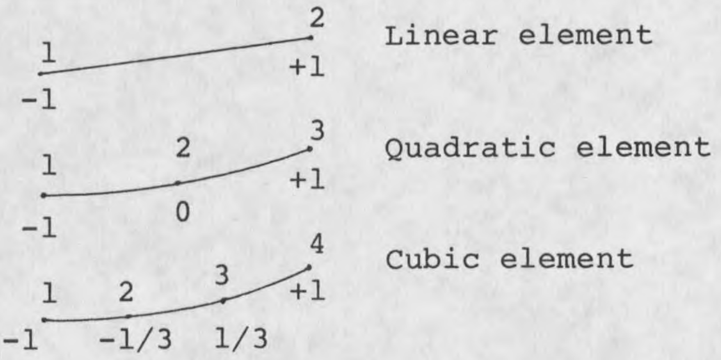
Quadratic: $N^1(\eta) = -1/2(1 - \eta)\, \eta$

$N^2(\eta) = (1 + \eta)(1 - \eta)$

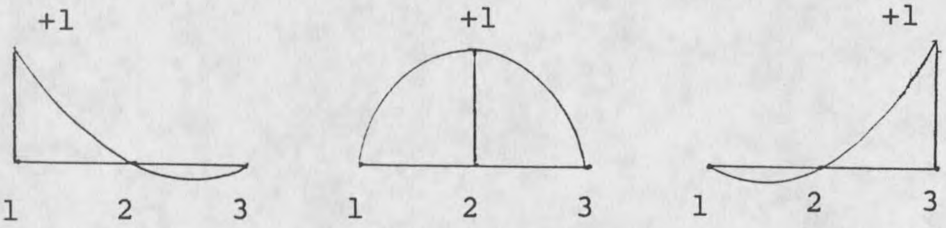
$N^3(\eta) = 1/2(1 + \eta)\, \eta$



a. A line element.



b. Location of nodes for the representation of function.



c. Quadratic shape functions.

Figure 2. Element representation.

$$\begin{aligned}
\text{Cubic:}\quad N^1(\eta) &= -\,9/16(\eta - 1)(\eta + 1/3)(\eta - 1/3) \\
N^2(\eta) &= 27/16(\eta - 1)(\eta + 1)(\eta - 1/3) \\
N^3(\eta) &= -27/16(\eta - 1)(\eta + 1)(\eta + 1/3) \\
N^4(\eta) &= 9/16(\eta + 1)(\eta + 1/3)(\eta - 1/3)\,. \qquad (4.11)
\end{aligned}$$

The constant shape function can not be used for boundary approximation and is only used to approximate tractions and displacements.

The Parametric Representation of Functions

Over each boundary element, tractions and displacements are approximated by constant, linear, quadratic or cubic variation with respect to the natural coordinates. The value of a function at an arbitrary point η of an element is expressed in terms of the functional values at the nodes and the shape functions. Thus,

$$\Phi_l(\eta) = \sum_{l=1}^{L} M^l(\eta)\, \Phi^l_l\,, \qquad -1 \leq \eta \leq 1\,; \qquad (4.12)$$

where the shape functions $M^l(\eta)$ are given by the Eq. (4.11). It is not necessary to have the same order of approximate representation for displacements and tractions, and it may be consistent to approximate displacements with a higher order polynomial than tractions (Brebbia, et.al., 1984). However, in this dissertation, both tractions and displacements are represented by polynomials of the same order.

Let there be K elements, each with L functional nodes. Each node can be given a global number $n(k,\ell)$, $k \varepsilon [1,--,K]$, $\ell \varepsilon [1,---,L]$; where k is the element number and ℓ is the local node number in the element k. n varies between 1 and N, where N is the number of distinct nodes. The discretised equation is

$$C_{IJ}u_J(P) + \sum_{k=1}^{K} \sum_{l=1}^{L} u_J(Q^{n_{kl}})[\int_{\Gamma k} T_{IJ}(P,Q^{n_{kl}}(\eta))\, M^l(\eta)\, J(\eta)d\eta]$$

$$\approx \sum_{k=1}^{K} \sum_{l=1}^{L} t_J(Q^{n_{kl}})[\int_{\Gamma k} U_{IJ}(P,Q^{n_{kl}}(\eta))M^l(\eta)J(\eta)\, d\eta]\,. \qquad (4.13)$$

The method of collocation requires that Eq. (4.13) be satisfied exactly at a set of distinct points. By choosing these points coincident with the nodes, the following set of equations is obtained

$$C_{IJ}u_J(P^m) + \sum_{k=1}^{K} \sum_{l=1}^{L} u_J(Q^{n_{kl}})[\int_{\Gamma k} T_{IJ}(P^m,Q^{n_{kl}}(\eta))\, M^l(\eta)J(\eta)d\eta\ ]$$

$$= \sum_{k=1}^{K} \sum_{l=1}^{L} t_J(Q^{n_{kl}})[\int_{\Gamma k} U_{IJ}(P^m,Q^{n_{kl}}(\eta))\, M^l(\eta)J(\eta)d\eta]. \qquad (4.14)$$

At each node the above equation is evaluated, forming a system of 2N (I = 1,2; m = 1,--,N) linear algebraic equations. The equation coefficients are the integrals of kernel and shape functions products. The kernel is, in general, a function of the distance between the source point and the element (Fig. 3). These integrals are numerically evaluated using Gaussian quadrature. The details of the numerical integration are presented in the appendix A.

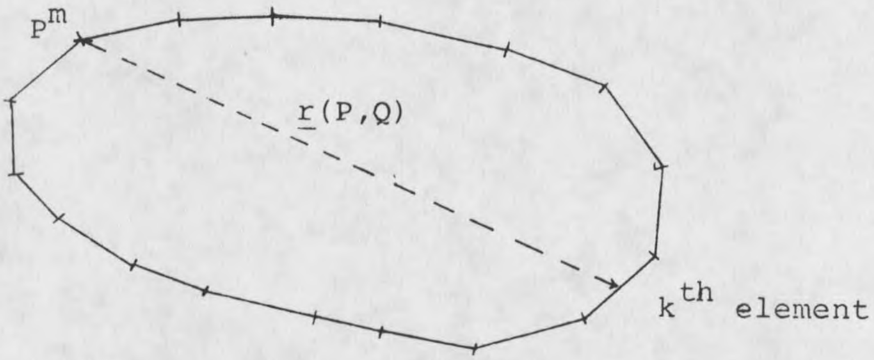


Figure 3. Boundary element discretisation for Eq. (4.14)

System of Equations

Eq. (4.14) can be written as

$$\sum_{n=1}^{N} [C_{IJ}\delta_{mn} + \Delta T_{IJ}(m,n)]\, u_j(n) = \sum_{n=1}^{N} [\Delta U_{IJ}(m,n)]\, t_J(n)\,, \qquad m = 1,2,--,N \tag{4.15}$$

where

$$\Delta T_{IJ}(m,n) = \int_{\Gamma k} T_{IJ}(P^m, Q^n{}_{kl}(\eta))\, M^l(\eta)\, J(\eta)\, d\eta \tag{4.16}$$

$$\Delta U_{IJ}(m,n) = \int_{\Gamma k} U_{IJ}(P^m, Q^n{}_{kl}(\eta))\, M^l(\eta)\, J(\eta)\, d\eta \tag{4.17}$$

which has the matrix representation

$$[C\,[I] + \Delta T]\,\{u\} = [\Delta U]\,\{t\} \tag{4.18}$$

When displacements are prescribed over the entire boundary, Eq. (4.18) can be solved directly for the unknown tractions.

The solution to the mixed boundary-value problem is obtained by rearranging Eq. (4.18) to give

$$[A]\,\{X\} = \{Y\} \tag{4.19}$$

where {X} contains unknown tractions and displacements, while {Y} contains known tractions and displacements multiplied by their corresponding coefficients ΔU_{IJ} and ΔT_{IJ}, respectively. During the construction of each pair of equations, the integrals, after evaluation, are placed directly in the coefficient matrix [A]. If $u_I(Q^n)$ is unknown, then ΔT_{IJ} is placed in the matrix [A], and ΔU_{IJ} is multiplied by the known traction $t_I(Q^n)$ and transferred to the right hand side vector {Y}. The procedure is reversed if $t_I(Q^n)$ is unknown.

Since the system of equations retains both traction and displacements as unknowns, the diagonal terms in [A] are not of the same order and the resulting system is illconditioned (Cruse, 1969). To obtain a numerically stable system of equations, the coefficients are scaled by multiplying ΔU_{IJ} by the modulus of elasticity and dividing known values of tractions by the modulus of elasticity. After the solution is obtained it is necessary to multiply the calculated traction by the modulus of elasticity.

Computation of the integrals is time consuming. However, some time can be saved by assembling the coefficient matrix column wise. This way, integration points for a particular element are computed only once. In most problems, known tractions or displacements are zero over most of the boundary. If an element has zero traction then ΔU_{IJ} need not be evaluated for that element.

The resulting coefficient matrix is fully populated and non-symmetric. However, the scaled system of equation is stable and can be solved by Gaussian elimination.

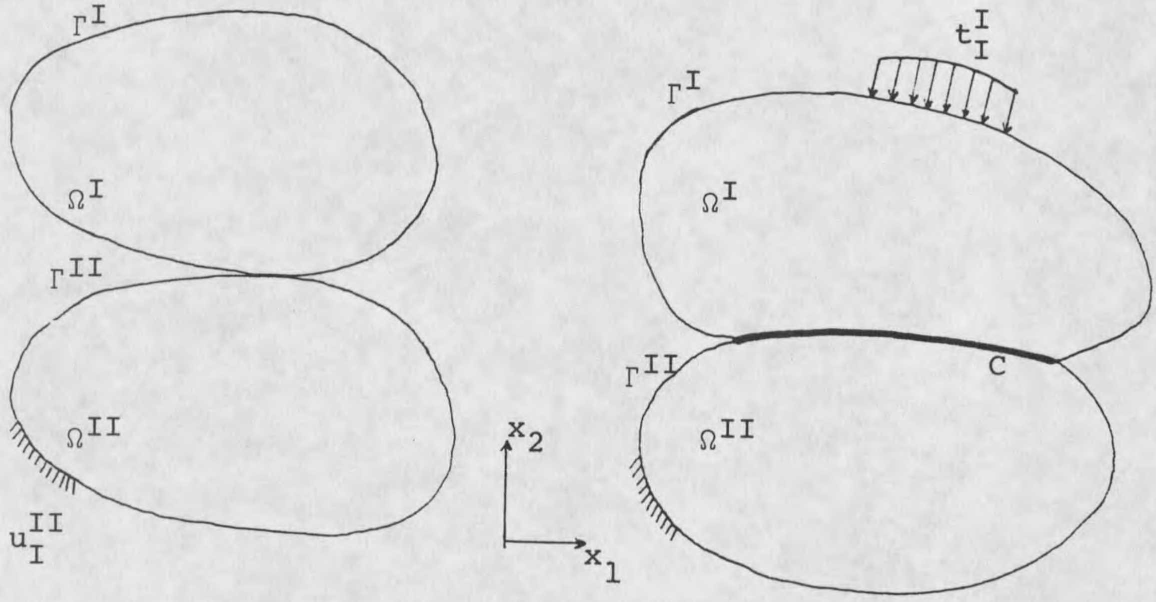
CHAPTER 5

FORMULATION OF FRICTIONLESS CONTACT PROBLEMS

For the purpose of this study it is assumed that contact occurs between two isotropic, homogeneous elastic bodies that, prior to loading, touch at least at one point (Fig. 4). Contact is brought about by prescribed external forces. It is further assumed that the displacements are sufficiently small that the linear theory of elasticity applies.

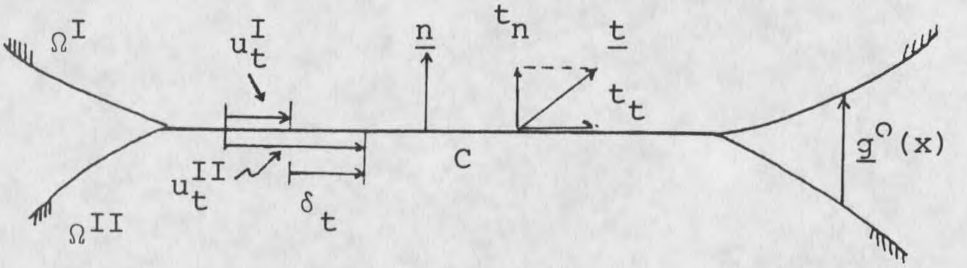
The boundary can be viewed as consisting of points inside the contact zone and points outside the contact zone. Outside the contact zone either displacements or tractions (but not both) are known at every boundary point. Inside the contact zone, neither displacements nor tractions are known. However, kinematic and equilibrium conditions provide a sufficient number of additional equations so that all unknown boundary quantities can be determined.

Consider two bodies I and II bounded by Γ^{I} and Γ^{II}, respectively in the x_1 -x_2 plane (Fig. 4a). The tractions on the boundaries are denoted by $\underline{t}$ and the displacements by $\underline{u}$. A contact region is formed between these bodies due to the application of external forces as shown in Fig. 4b. If $g^o(x_1)$ is the gap between two points which are assumed to come in contact, then the basic conditions in the region of



a. Prior to contact.

b. Deformed configuration.



c. The region of contact, Γ_c.

Figure 4. Two bodies in contact.

contact are:

1. No material interpenetration can occur, therefore,

   $\delta(x_1) = 0$ inside the region of contact,
   $\delta(x_1) > 0$ outside the region of contact, (5.1)

   where $\delta(x_1)$ is the current gap.

2. The normal component of traction in the contact region is compressive.

3. In the absence of friction the tangential component of traction inside the contact zone is zero:

   $$t_t^{I} = t_t^{II} = 0, \tag{5.2}$$

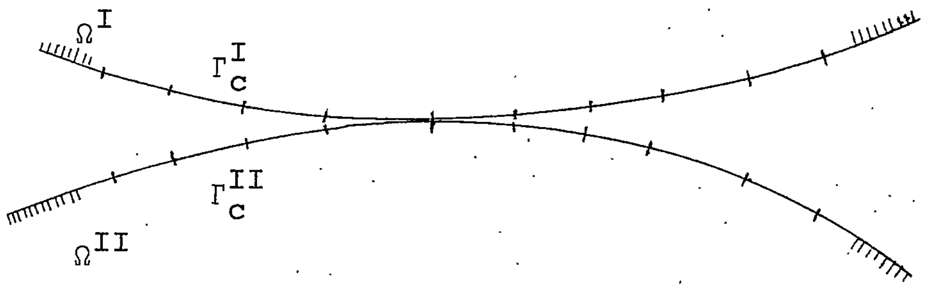
   where t_t^{I} and t_t^{II} are the tangential component of traction for body I and body II, respectively.

In addition to the observed contact conditions, equilibrium across the contact zone requires the normal contact forces to be equal and opposite.

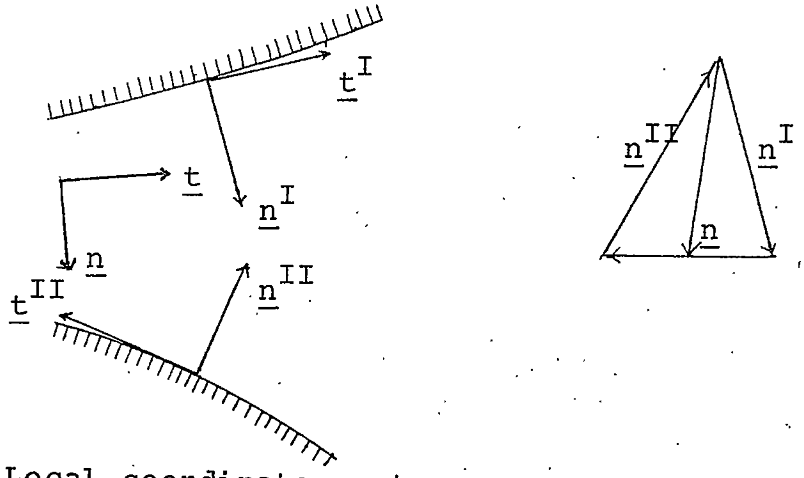
Local Coordinate System

To solve contact problems using the BIE method, the portion of the boundary expected to come in contact, Γ_c, is divided into boundary elements for both bodies I and II, such that nodes on opposite sides of the contact surface form a pair of contact nodes. To write the kinematic and equilibrium equations, for each of the pair of contact nodes a local coordinate system is defined (Fig. 5). If $\underline{n}^{I}$ and $\underline{n}^{II}$ define the unit outer normal vectors for the contact nodes under consideration, then an average normal vector $\underline{n}$ is defined as,

$$\underline{n} = (\underline{n}^{I} - \underline{n}^{II})/|\underline{n}^{I} - \underline{n}^{II}| \tag{5.3}$$



a. The region of contact, Γ_c.



b. Local coordinate system.

Figure 5. Local coordinate system for the contact zone.

Similarly an average tangential vector $\underline{t}$ is defined using $\underline{t}^{I}$ and $\underline{t}^{II}$. The contact conditions are written in the local coordinate system defined by $\underline{n}$ and $\underline{t}$.

The Frictionless Contact Problem

In the case of frictionless problems tangential tractions inside the contact zone vanish and in general the extent of contact zone C, the normal pressure distribution, and the displacements inside C remain as unknowns. The problem is load-path independent. In solving such problems numerically, a possible contact zone is estimated conservatively such that the actual contact zone is contained within it. The actual contact zone is then found through an iterative procedure.

Displacement Compatibility at The Contact Surface

If $\underline{u}^{I}$ and $\underline{u}^{II}$ are the displacement vectors of a pair of contact nodes, and if $\underline{g}^{\circ}$ is the initial gap vector between these nodes, then the normal component of the current gap can be defined as (Fig. 6),

$$\delta_n = \underline{\delta} \cdot \underline{n} = (\underline{u}^{II} - \underline{u}^{I} + \underline{g}^{0}) \cdot \underline{n} \,, \tag{5.4}$$

δ_n should be greater than or equal to zero because material interpenetration is not allowed. Hence, for contact to occur the condition is,

$$\delta_n = 0 \,. \tag{5.5}$$

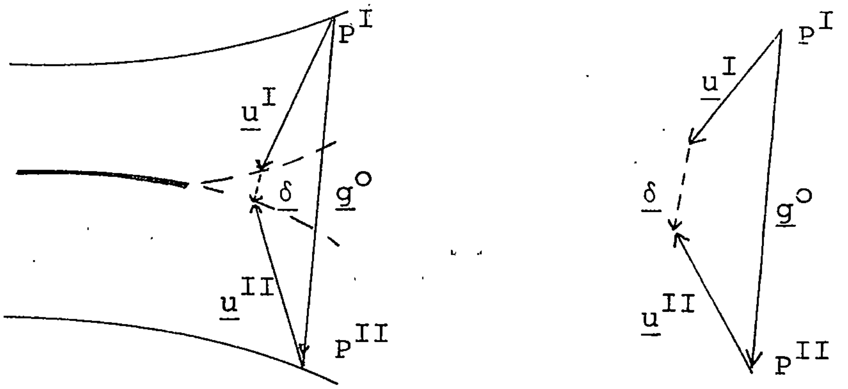


Figure 6. The gap vector $\underset{\sim}{g}$ between a potential contact node pair.

Contact Conditions

The normal component of traction must be non-tensile, i.e.,

$$t^I{}_n \leq 0 \; ; \; t^{II}{}_n \geq 0 \tag{5.6}$$

The sign difference arises because the equations are written in the local coordinate system defined previously, with the average normal vector pointing away from body I.

In the absence of friction the tangential tractions in the contact zone are zero:

$$t^I{}_t = 0 \, , \text{ or } t^{II}{}_t = 0 \, . \tag{5.7}$$

In this case relative sliding can take place and no restriction is placed on relative tangential displacement.

Equilibrium Equations at the Contact Surface

The normal and tangential components of traction across the contact surface must be in equilibrium:

$$(E_1/E_2)\, t^{I}{}_{t} + t^{II}{}_{t} = 0\,, \tag{5.8}$$

$$(E_1/E_2)\, t^{I}{}_{n} + t^{II}{}_{n} = 0\,. \tag{5.9}$$

These equations are multiplied by the ratio of Young's modulus of bodies I and II since, as described in Chapter 4, to obtain a numerically stable system of equations, the tractions of each body are divided by its Young's modulus.

Numerical Solution

The problem is now solvable since for every boundary point there is one equation for every unknown quantity. Outside the contact zone, at every boundary point either tractions or displacements are unknowns for which two boundary integral equations can be written. Inside the contact zone at each boundary point both normal and tangential traction components along with the two displacement components remain unknowns. Hence, for a pair of contact nodes there are eight unknowns. For these eight unknowns, four boundary integral equations (two for each body) along with, Eq. (5.5), (5.7), (5.8), and (5.9) provide eight equations. The unknowns and the corresponding equations are identified in Table 1. For example, if a node pair is not in contact (open), the normal tractions are known to be zero and the normal displacements are found by using the boundary integral equations. If a node pair is in contact, the normal components of traction are defined by a boundary integral equation and Eq. (5.8). Similarly, the normal components of displacement are defined by a boundary integral equation and Eq. (5.5). In frictionless contact problems the tangential traction is zero; and since no restriction is placed on tangential displacements inside the potential contact zone, these are computed by using the boundary integral equations.

Outside the potential contact zone boundary elements are defined in the usual manner. Within the potential contact zone, elements for both bodies are defined in pairs, keeping the length of the elements in each pair approximately the same (Fig. 5 a).

Initially, assumptions are made concerning the contact status of each node pair in the potential contact zone and contact equations are written accordingly. These, along with the discretised boundary integral equations, form a system of linear equations. At each step of the iteration process the contact status of each node pair is checked and updated (see Table 2).

For node pairs that were assumed to be in contact, if the normal traction is tensile, the node pair is released by setting the normal tractions to zero for the next iteration.

For node pairs that assumed not to be in contact, if the resultant gap δ_n in Eq. 5.2, is negative the node pair is brought into contact by setting δ_n to be zero for the next iteration.

As the contact status is checked and updated for each iteration, the contact equations are written according to the updated conditions, thus forming a new system of equations. The system of equations is solved by Gauss elimination. The operation of the equation solver is discussed in detail in Chapter 7.

Table 1. Unknowns and corresponding equations at a potential contact node.

Unknowns at a potential contact node	Equations: Body I — All Cases	Equations: Body II — Open	Equations: Body II — In Contact
t_n	Equilibrium equation (5.8)	$t_n = 0$	Integral equation
t_t	Equilibrium equation (5.9)	$t_t = 0$	$t_t = 0$
u_n	Integral equation	Integral equation	Compatibility equation (5.5)
u_t	Integral equation	Integral equation	Integral equation

Table 2. Criteria for determining contact conditions during the iterative process.

Assumed contact status	Value of δ_n, t_n^{II} after the iteration	Contact status for next iteration
Open	$\delta_n > 0$	Open
	$\delta_n \leqslant 0$	Contact
Contact	$t_n^{II} > 0$	Contact
	$t_n^{II} \leqslant 0$	Open

CHAPTER 6

RESULTS FOR FRICTIONLESS CONTACT PROBLEMS

A FORTRAN computer program called CONTACT is developed for two dimensional elastic contact problems with and without friction. Various contact problems, for which analytical solutions are available, are solved using the program. In the numerical analysis of contact problems, frictionless problems are a subclass of frictional contact problems. Nevertheless, since analytical solutions for several frictionless problems have appeared in the literature, these problems provide a broad set of test cases for evaluating several aspects of the computer program.

Hertz's Problem

An elastic cylinder lying on a flat surface subjected to a concentrated load is analyzed for the case of plane strain. The boundary geometry is approximated using quadratic and linear variation for the cylinder and foundation, respectively. The boundary element models of the two bodies are shown in Fig. 7. The potential contact zone contains 9 elements of equal length; the region outside of the potential contact zone contains 26 and 36 elements for the cylinder and the foundation, respectively.

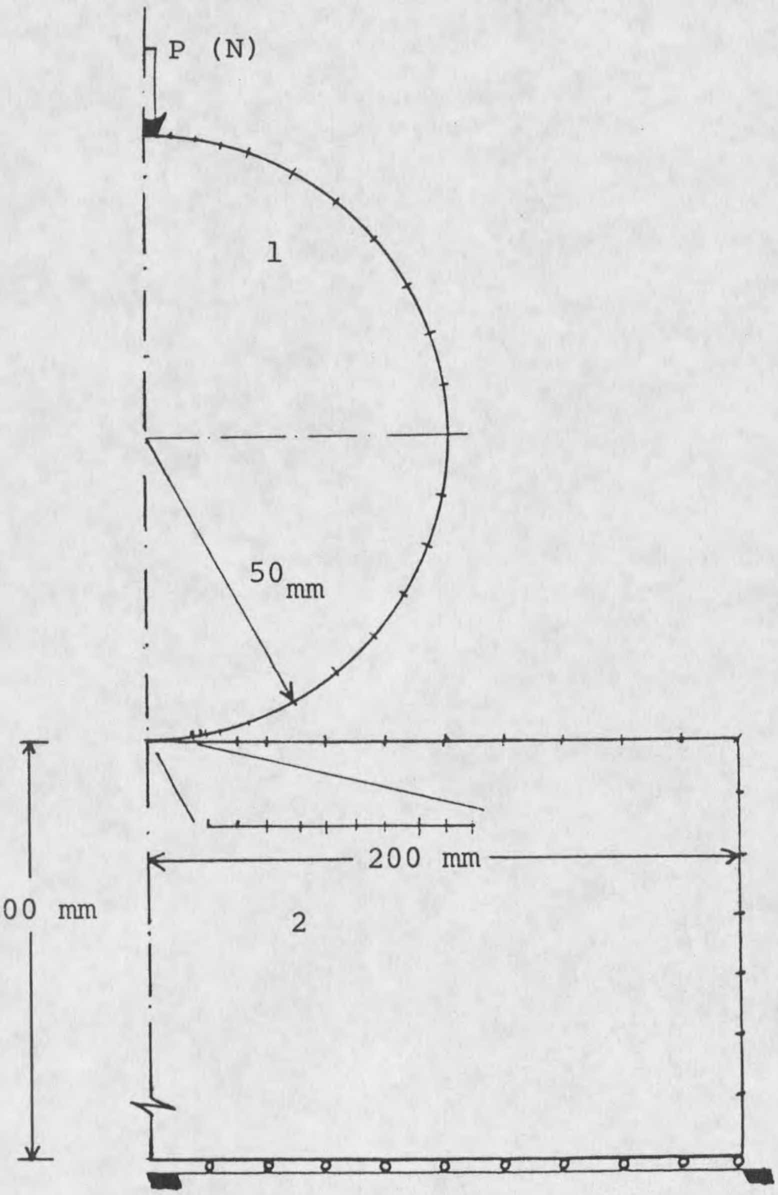


Figure 7. Boundary element model of the Hertz's problem.

The problem is solved using constant and linear functional representations.

Contact pressures for four different cases are shown in Fig. 8. A rigid cylinder is modeled using a high Young's modulus. The results are in very good agreement with theoretical results (Timoshenko and Goodier, 1970). Constant and linear elements give identical results, therefore higher order functional approximations were not used. The maximum error in the normal pressure for the cases considered is found to be 2.5 percent. This error is for the case in which the cylinder is rigid and the foundation is taken to be incompressible, and it can be attributed to the large difference in the Young's moduluii. In all cases the extent of contact is predicted accurately without having to use a fine mesh where the contact ceases.

Sachedeva, et. al. (1981) solve a similar problem using the finite element method. Only part of the cylinder is modeled to keep the size of the problem small. They report that the problem was solved with 326 degrees of freedom, of which 84 are in the potential contact zone. Here the problem is solved with 192 degrees of freedom with all the unknowns in the potential contact zone retained (four per contact node), amounting to 80 degrees of freedom. The excess degrees of freedom in using the FEM are due to the fact that the interior of both bodies must be modeled rather than just their boundaries.

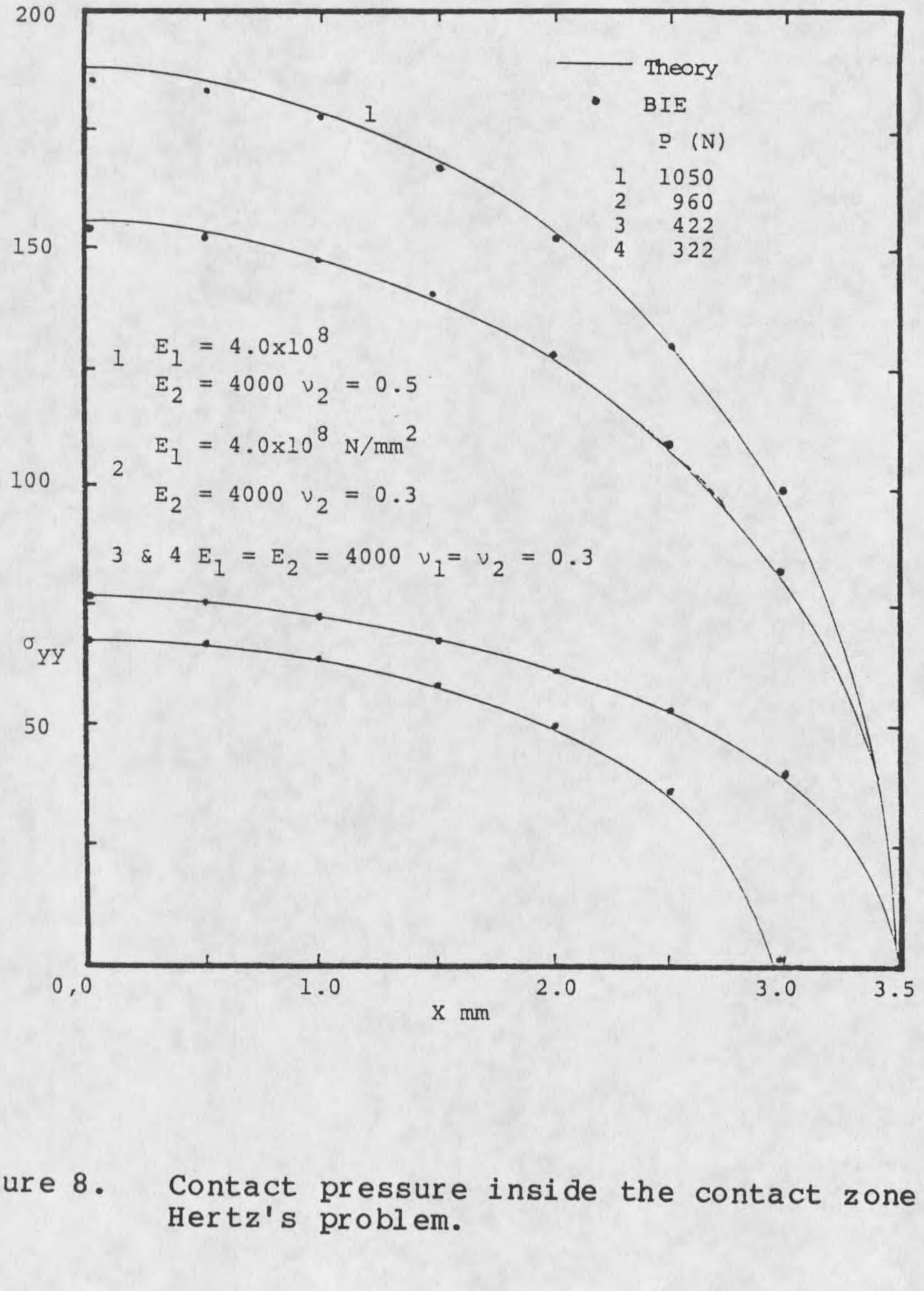


Figure 8. Contact pressure inside the contact zone for the Hertz's problem.

Contact Problem for a Frictionless Elastic Layer

A plane strain contact problem for a frictionless elastic layer pressed against an elastic foundation is considered. Various loading cases give rise to some interesting contact conditions. The problem is solved for three different loading cases which are depicted along with the results for each case. In all cases the initial contact extends over the entire length of the layer. Fig. 9 shows a typical boundary element model for these problems in which symmetry is taken into account to model only half of the layer and the foundation. Either linear or quadratic elements are used and the mesh sizes are changed according to the loading conditions. The mesh patterns inside the potential contact zone for the different load cases are shown in Fig. 9.

Load Case 1

In the first case, a compressive force at $x = 0$ is applied through a weightless layer (Fig. 10). As the point load is applied the layer separates from the foundation. Dundurs (1975) classifies this as a receding contact problem, and shows that the extent of contact for the frictionless case is independent of the level of applied loading. Keer, et.al. (1972) and, Ratwani and Erdogan (1973) have obtained an analytical solution to this problem. Here, the analytical solution presented by Keer, et. al. (1972) is

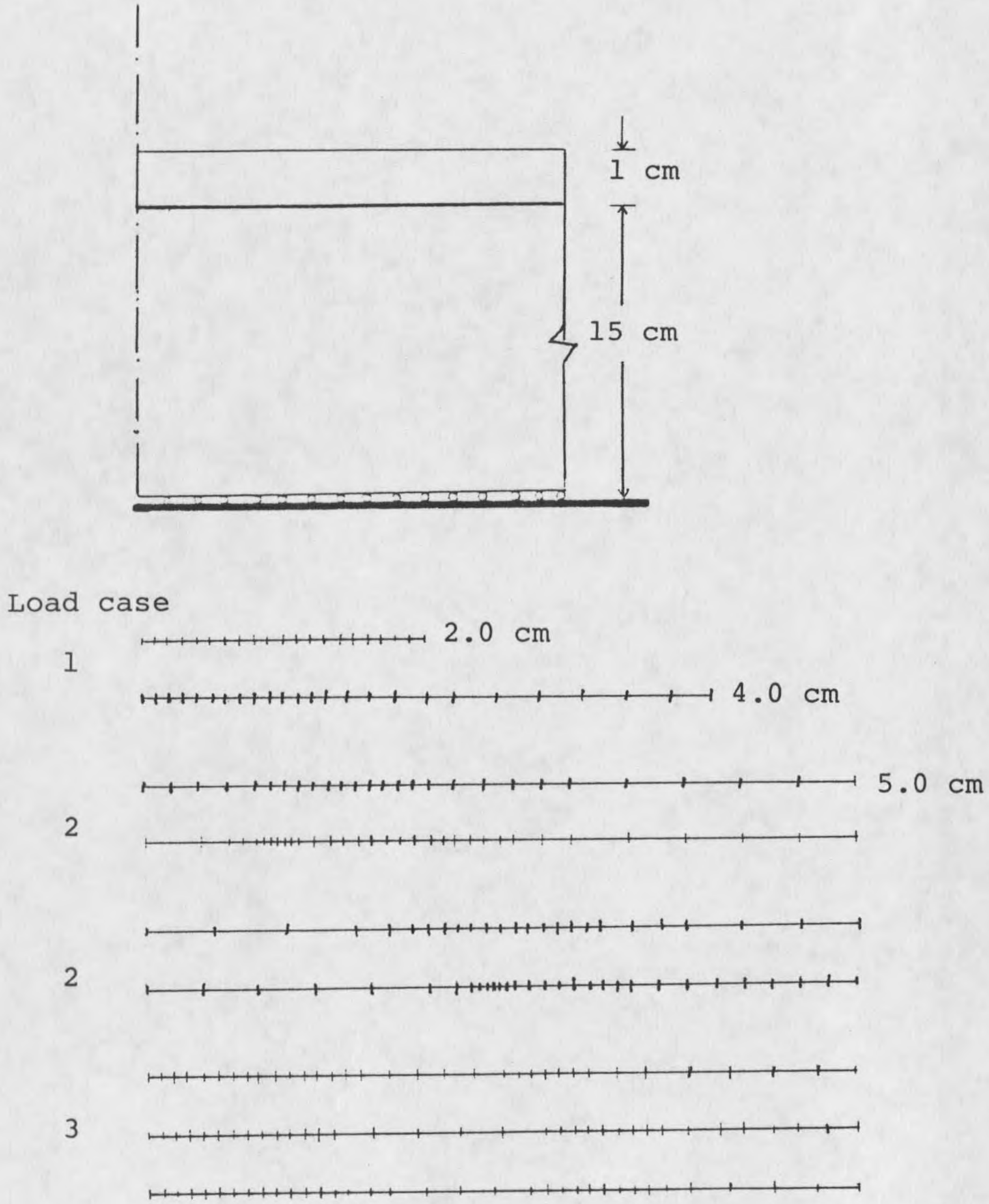


Figure 9. Boundary element model of an elastic layer on an elastic foundation, with different mesh patterns along the contact zone.

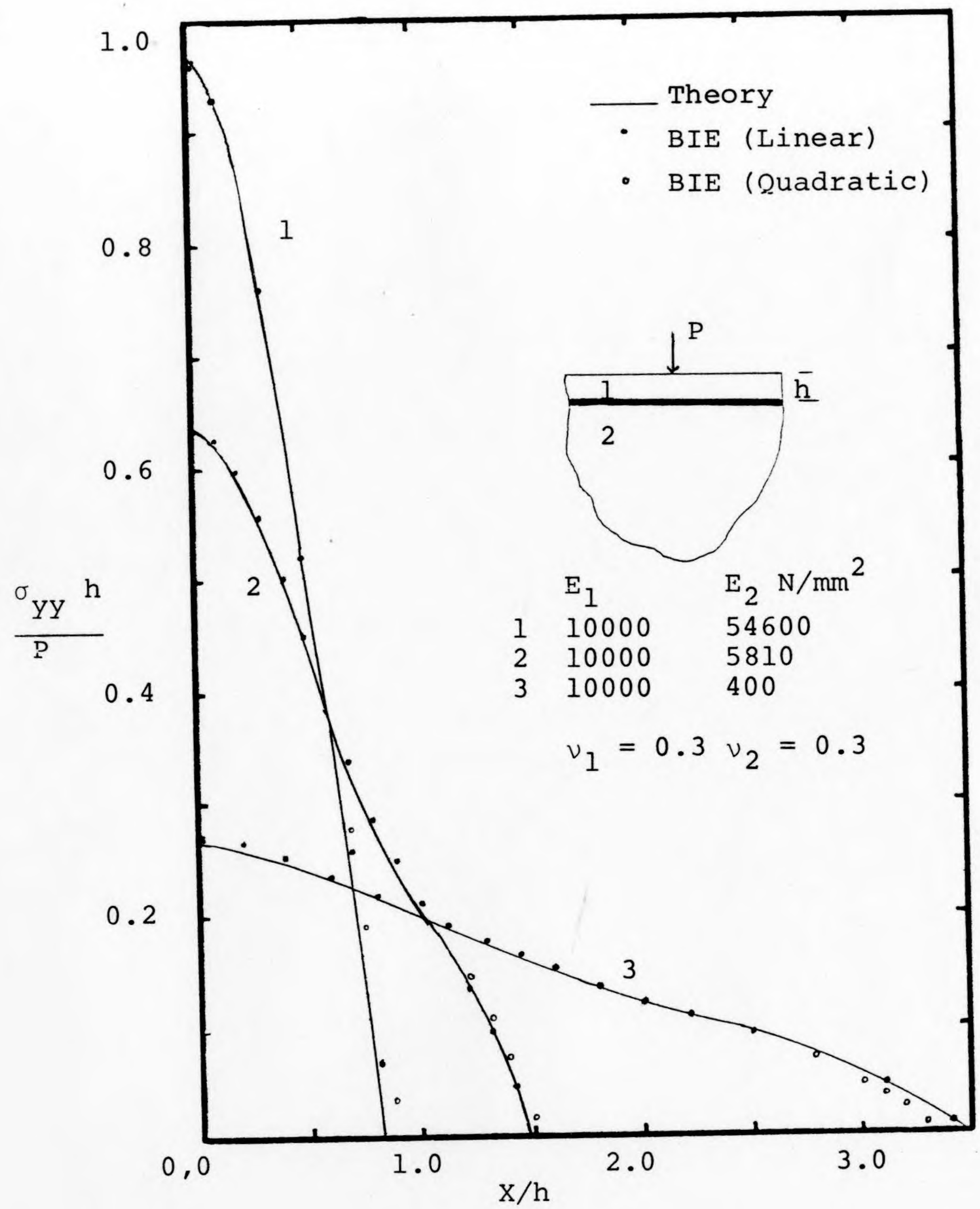


Figure 10. Contact pressure inside the contact zone for a compressive force P.

used for comparison. Three different choices of material parameters are used to study the behavior. The length of the layer for the first two choices is 2 cm., and is modeled with equal size elements of length 0.1 cm.. The length of the layer for the third choice is 4 cm. and a different mesh is used (Fig. 9).

Fig. 10 shows the normal contact pressure for the three different choices of material parameters. As the layer stiffness increases relative to the foundation, less bending occurs in the layer and the extent of contact is increased. In all the three cases the extent of contact and the traction distribution are accurately predicted. Linear and quadratic element give virtually the same results except where the contact ceases. This is to be expected, since for quadratic elements the normal pressure distribution can change depending on whether contact ceases at a corner node or at an interior node. If the contact ceases at an interior node of an element, local tractions in the element are approximated as shown in Fig. 11 which may produce erroneous results. Similar observation are made by Andersson (1982) using the BIE method and, by Torstenfelt (1983) using the FEM.

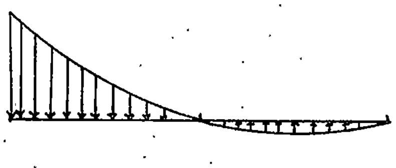


Figure 11. Traction distribution in a quadratic element when the contact ceases at an interior node.

Load Case 2

The second load case consists of an elastic layer of thickness h under the action of a uniform clamping pressure p_e on its top surface, and a tensile force P applied at $x = 0$ (Fig. 12). The foundation is considered to be rigid. Civelek and Erdogan (1975) have solved this problem analytically. If the concentrated load is sufficiently small then the contact along the interface is continuous. Above a critical load partial separation occurs near $x = 0$, and gets larger as the load is increased. The contact zone consists of 26 and 24 elements for $\lambda = P/hp_e$ equal to 2 and 4, respectively. The third and fifth mesh pattern depicted in Fig. 9 relate to these two cases.

Fig. 12 shows the normalized contact pressure distribution for two different values of the tensile force P. The pressure peak near $x = +a$ tends to unity as x/h increases. For $\lambda = 2$ the resultant force due to the difference between the upward and the downward forces acting on the separation zone is upward (tensile), and thus the pressure peak deviates from $x = a$ to keep the layer in equilibrium. When $\lambda = 4$ the resultant force is compressive, and the pressure peak helps support the lifted portion of the layer. For $\lambda = 4$, linear elements show a 4 percent error in the maximum contact pressure, while quadratic elements give less than 2 percent error. Both the quadratic and linear elements have same node locations.

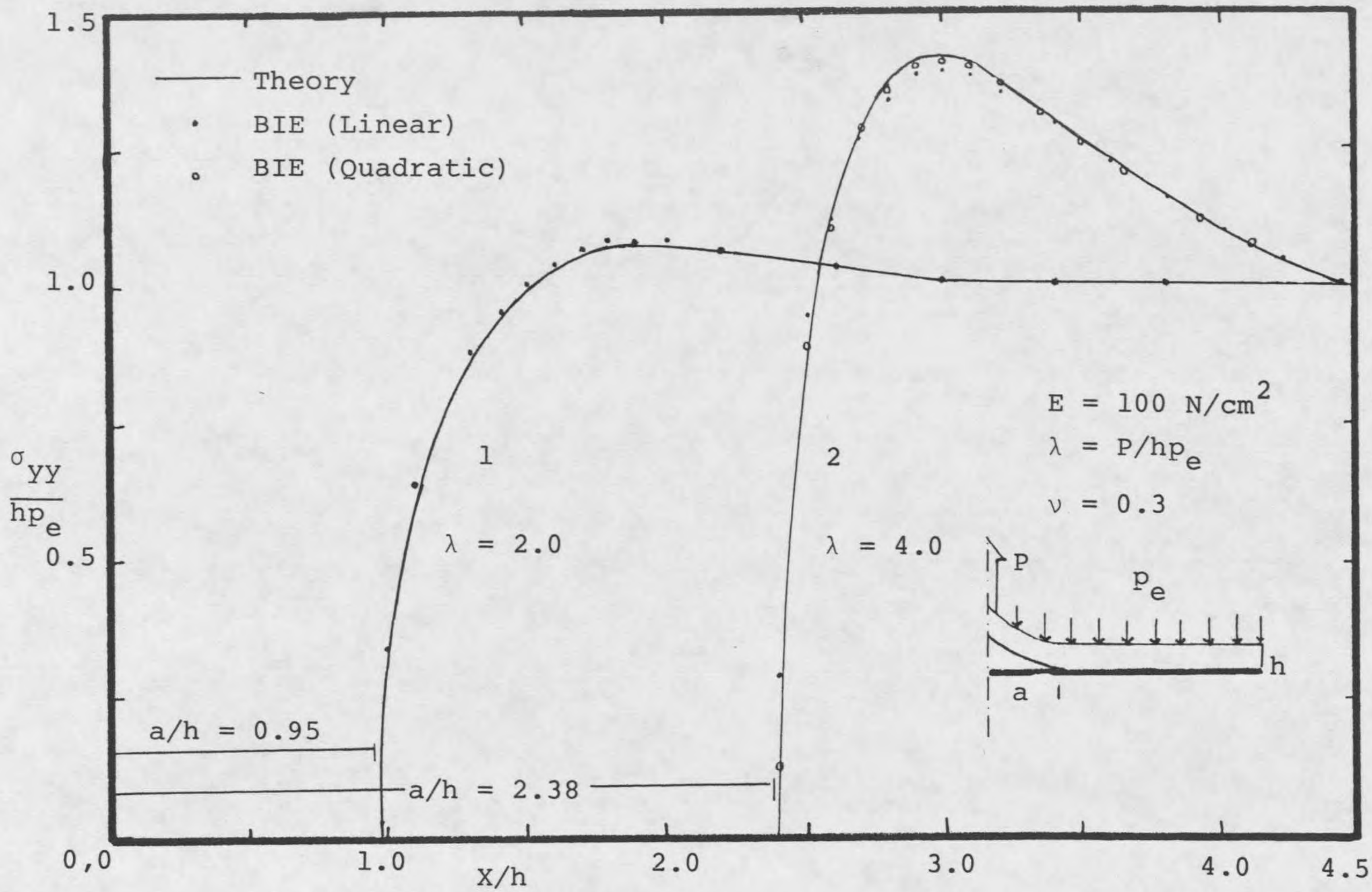


Figure 12. Contact pressure inside the contact zone for a uniform clamping pressure P_e and a tensile force P.

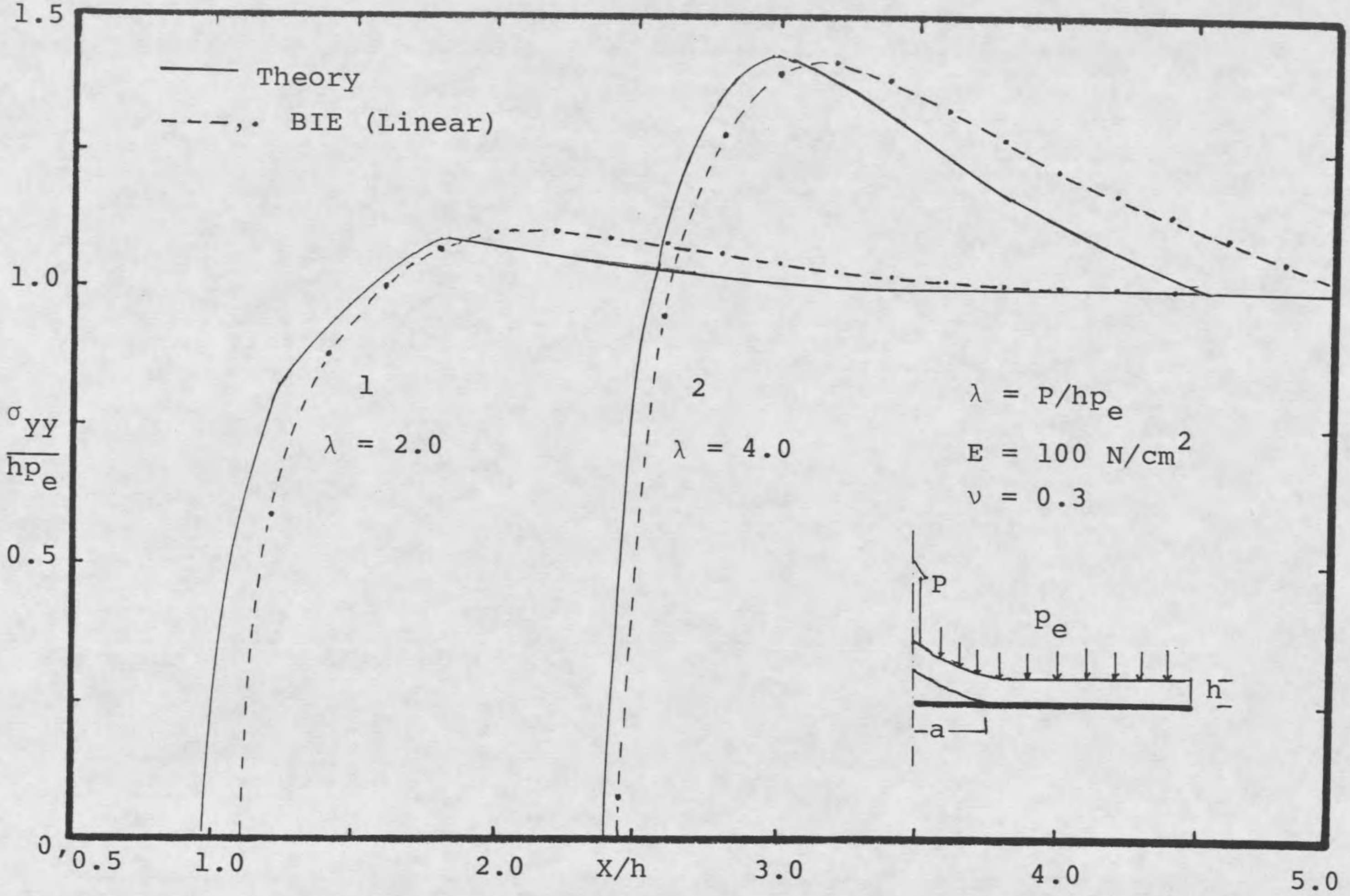


Figure 13. Effect of change in element size on the contact pressure for load case 2.

It is found that this problem is sensitive to mesh size and location of the contact nodes. The pressure gradient where the layer comes in contact is very high and a slight change in location of the contact nodes alters the extent of separation. Each problem is solved with two or three different mesh patterns to check the consistency in the results obtained. The solution is also found to be sensitive to changes in element size. Rapid changes in element size give poor results, and elements of uniform size are needed near the pressure peaks. Fig. 13 shows this shift in the extent of contact zone when a fine mesh pattern is provided where the layer comes in contact. In this case 31 and 29 elements are used in the contact zone for λ equal to 2 and 4, which relate to the fourth and sixth mesh pattern in Fig. 9, respectively.

Load Case 3

The third load case differs from the second load case only in that P is compressive. Again, for sufficiently small values of P continuous contact between the layer and the foundation is maintained. Initiation of separation occurs at a distance b from the axis of symmetry which is practically independent of the level of applied load (Civelek and Erdogan, 1976). Fig. 14 shows the pressure distribution for two different values of P. In order to include the entire pressure distribution and to give sufficient details, different scales are used for $\sigma_{yy}/hp_e > 5$ and for

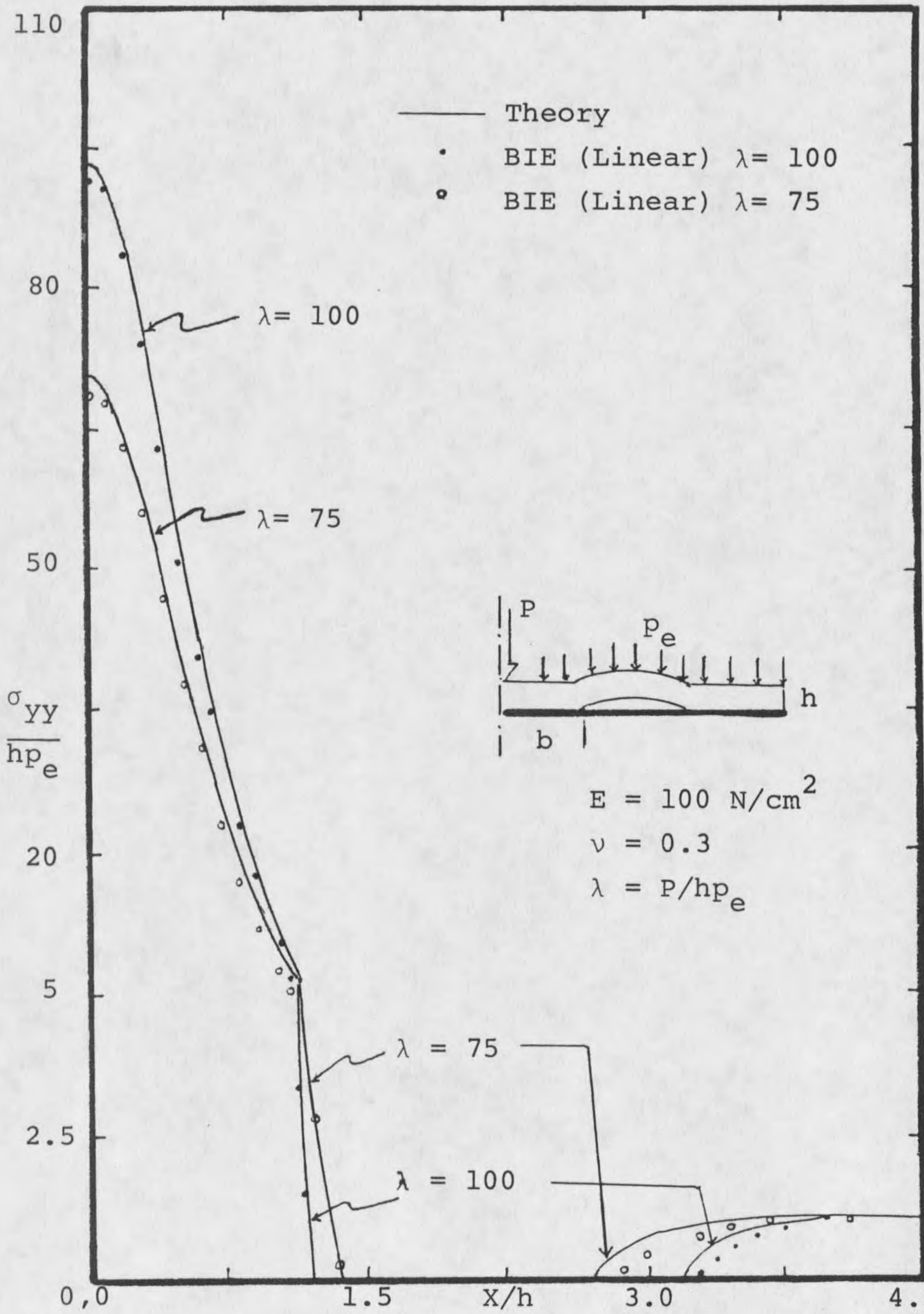


Figure 14. Contact pressure inside the contact zone for a uniform clamping pressure P_e and a compressive force P.

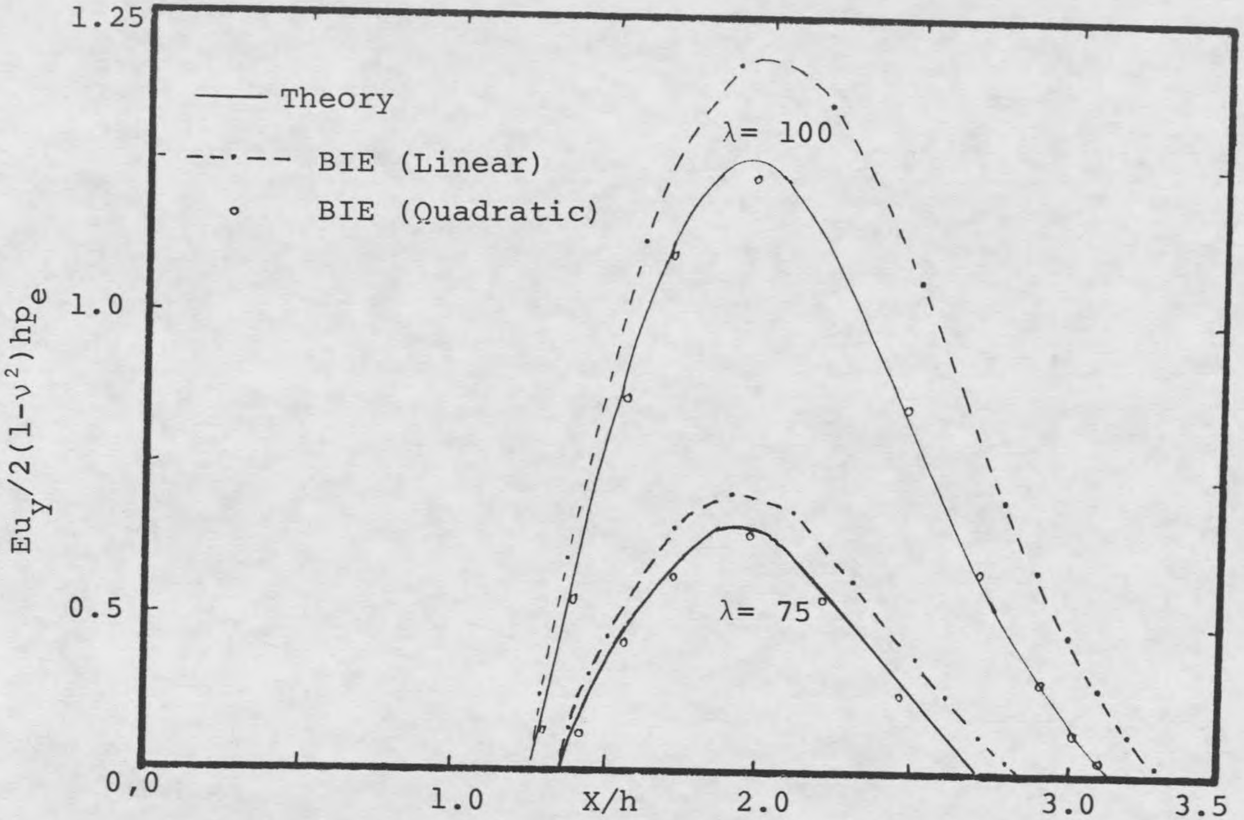


Figure 15. Vertical displacement in the separation zone for load case 3, using linear and quadratic elements.

$\sigma_{yy}/hp_e < 5$. The BIE results are in good agreement with the theory except at the outer edge of the separation zone, where linear elements predict about 6 percent larger extent of separation.

Fig. 15 shows the vertical displacement in the separation zone. Although the linear elements give satisfactory pressure distribution, the displacements are in error by more than 15 percent. Quadratic elements with the same number of contact nodes (34) show a marked improvement in displacement calculations. The error is less than 4 percent with quadratic elements, which shows the importance of the higher order elements. This problem is also found to be sensitive to the mesh patterns.

The author is not aware of any other analytical solutions to contact problems which explicitly give the displacements. This has limited further comparative study of the elements when contact problems are considered. Gecit (1980) solves the problem of load cases 2 and 3 analytically, with the foundation considered to be elastic. Unfortunately, his results are questionable. One way to check the results is to consider equilibrium of forces for each body: the area under the contact pressure distribution curve can be measured approximately and should agree with the externally applied load. Some of his results for the load case 2 do not satisfy this criterion.

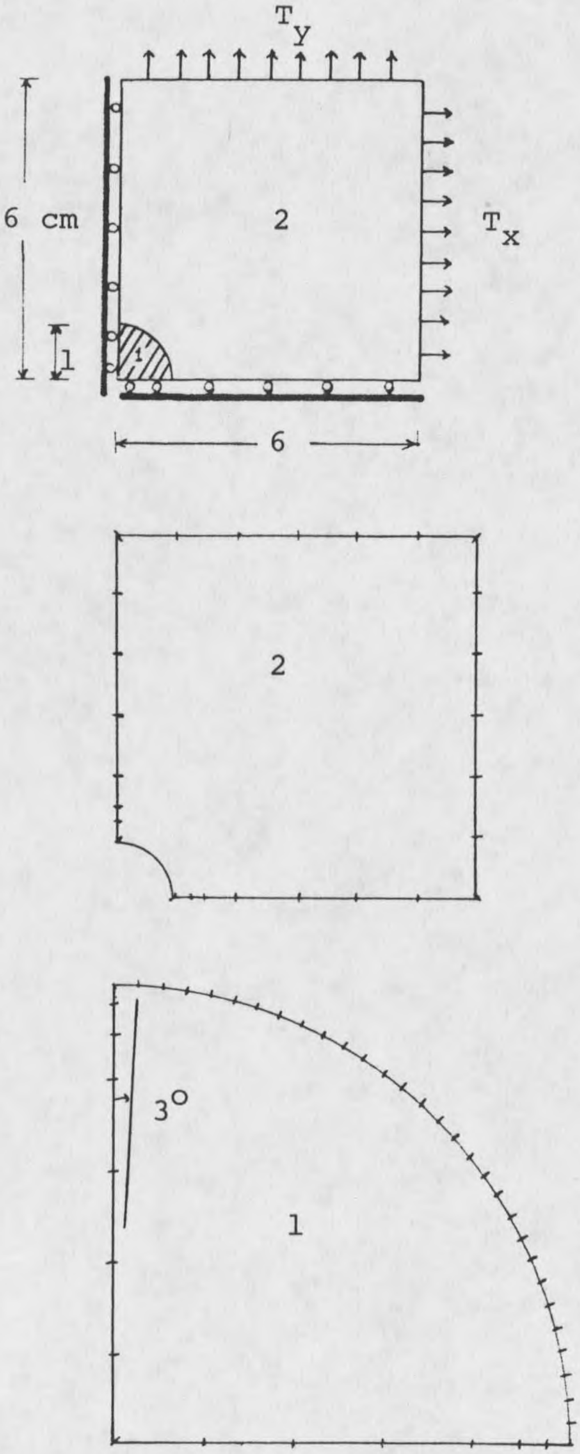


Figure 16. Boundary element model of an elastic plate with a circular inclusion.

Inclusion of a Smooth Circular Disc in a Plate

As an application of conformal contact, consider the following inclusion problem: a circular elastic disc is inserted in a hole in an elastic plate. The disc is smooth and fits exactly into the hole in the unstressed state. The plate is then subjected to stresses so that the disc separates from the plate along part of the boundary. For a frictionless problem, the extent of contact is independent of level of loading (Keer, et. al. 1973).

Fig. 16 shows the boundary element model. In the contact zone the linear elements are of equal length with nodes 3 degrees apart. Fig. 17 shows the normal tractions in the contact zone for identical materials under three different loading conditions. The normal tractions near the crown deviate from the analytical result by about 6 percent. Contact elements with nodes spaced 2, 3, 5 and 10 degrees apart, with linear and quadratic functional approximations, give similar results. A fine mesh near the crown and a coarse mesh away from the crown does not alter the results, and again the problem is found to be sensitive to changes in element size.

To further investigate the problem, different material parameters were used under the same loading conditions. Fig. 18 shows the normal tractions for different material parameters with same extent of contact. The numerical results consistently show higher values of normal traction

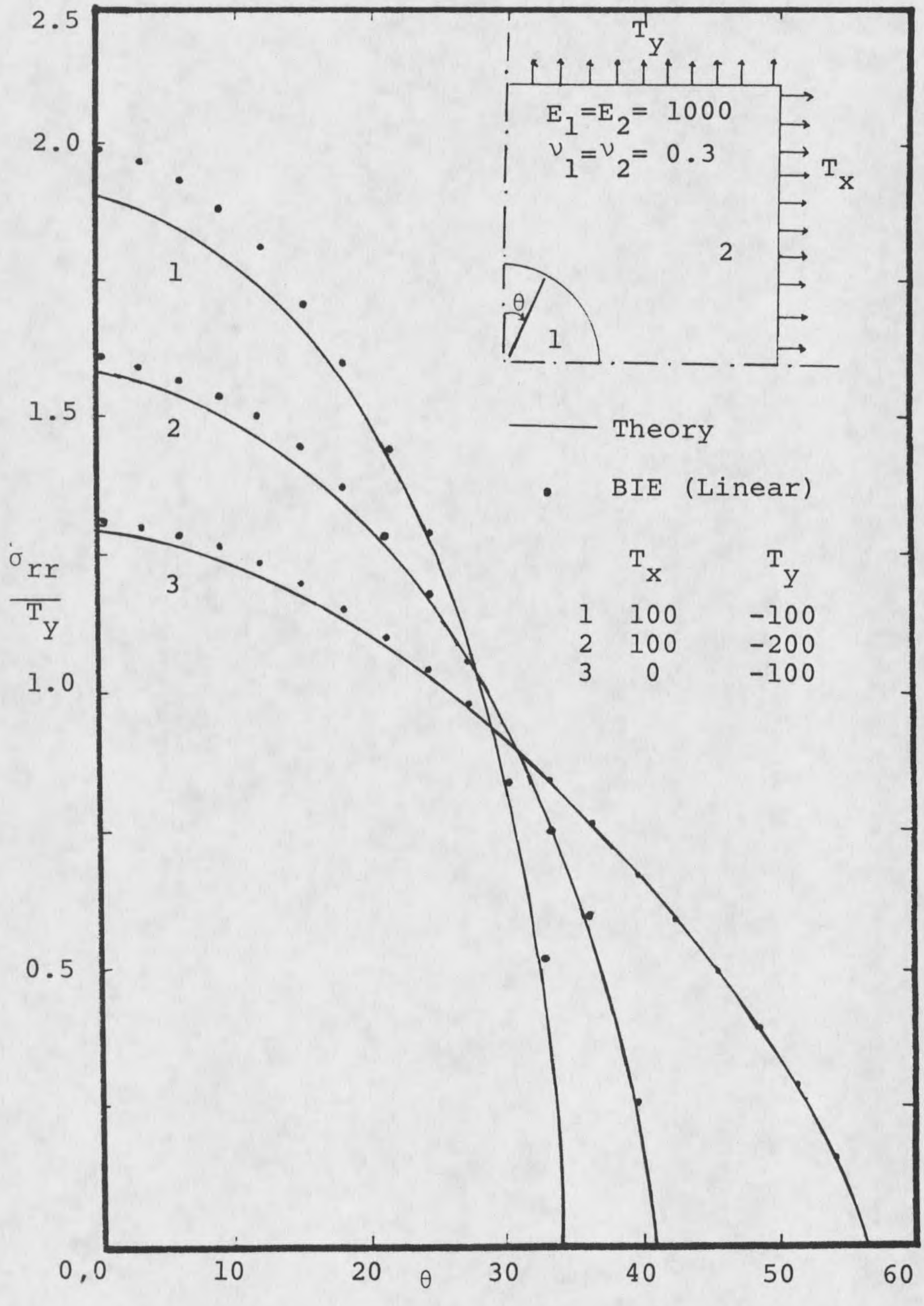


Figure 17. Contact pressure for the inclusion problem for different loading conditions.

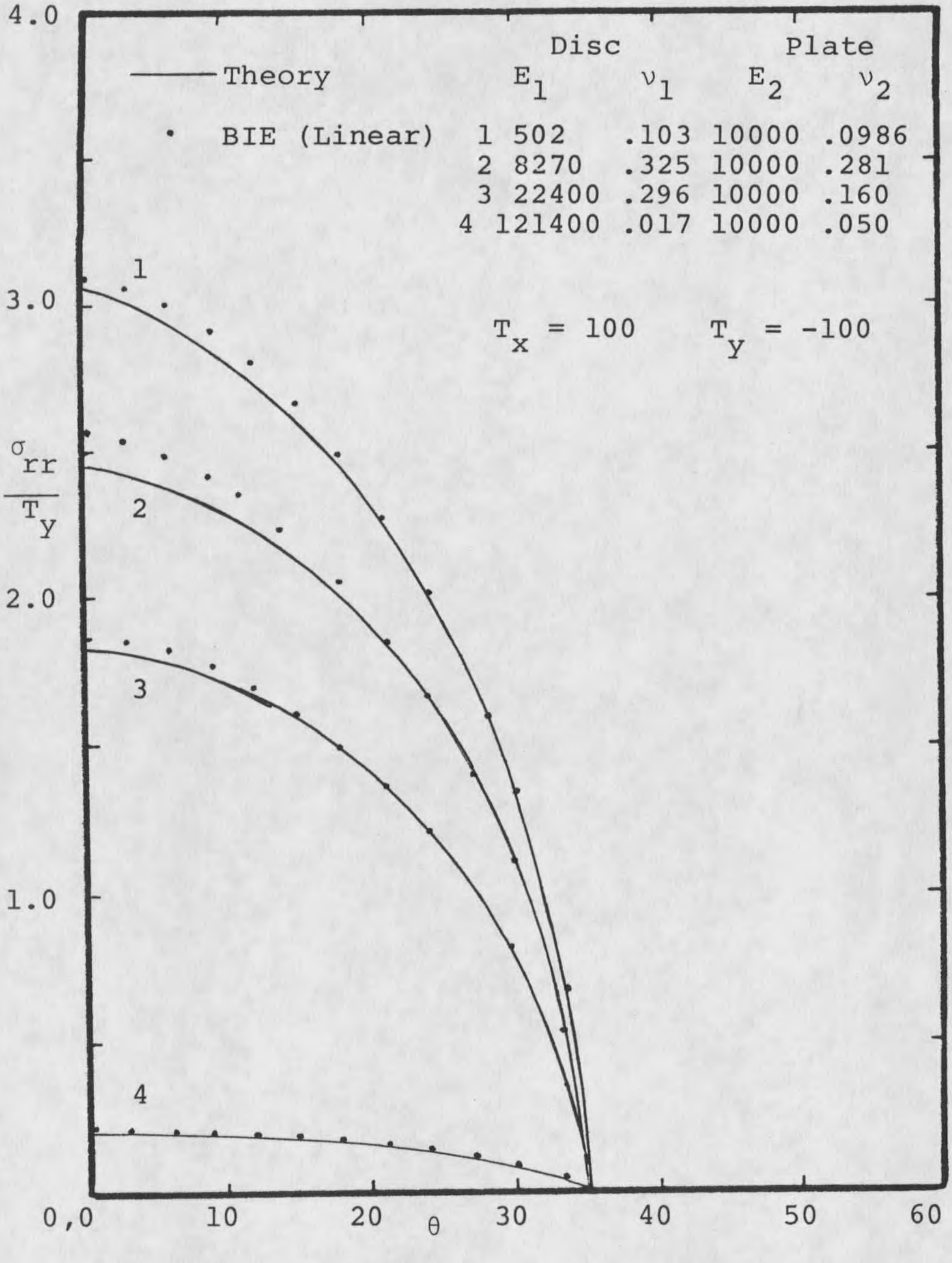


Figure 18. Contact pressure for the inclusion problem with different materials.

near the crown. The analytical results shown here are obtained by Keer, et. al. (1973). Their formulation leads to a Fredholm integral equation which is then numerically solved. They discuss the difficulty encountered in the numerical evaluation due to divergence of an infinite integral in the kernel of the integral equation. The small discrepancy observed here in the calculation of normal tractions may very well be due to the approximations involved in numerical evaluation of the Fredholm integral equation.

CHAPTER 7

FORMULATION OF FRICTIONAL CONTACT PROBLEMS

Frictional forces are associated with energy dissipation in the contact zone. Consequently the final state of the displacements and tractions depends on the loading history. The problem thus has to be solved by applying the load in increments, with the assumption that the behavior is linear during each load increment. Since the non-linearities arise only in the boundary conditions inside the contact zone, the boundary integral equations can be written in incremental form as:

$$C_{IJ}\Delta u_J(P) + \int_\Gamma T_{IJ}(P,Q)\, \Delta u_J(Q)\, d\Gamma(Q)$$

$$= \int_\Gamma U_{IJ}(P,Q)\, \Delta t_J(Q)\, d\Gamma(Q)\,. \tag{7.1}$$

The current tractions and displacements are then calculated as the sum of increments,

$$t^k_I = \Delta t^k_I + t_I^{k-1} = \sum_{k=1}^{K} \Delta t^k_I \tag{7.2a}$$

$$u^k_I = \Delta u^k_I + u_I^{k-1} = \sum_{k=1}^{K} \Delta u^k_I \tag{7.2b}$$

The Frictional Contact Problem

The basic concepts in the numerical solution of frictional problems are same as in frictionless problems and are not repeated here. In addition to the conditions of no material interpenetration and compressive normal traction, in the presence of friction the tangential component of traction must satisfy Coulomb's law of friction:

$$|t_t(x)| \leq \mu |t_n(x)| \ , \ \delta_t(x) = 0 \tag{7.3}$$

$$t_t(x) = - \mu |t_n(x)| \ , \ \delta_t(x) \neq 0 \tag{7.4a}$$

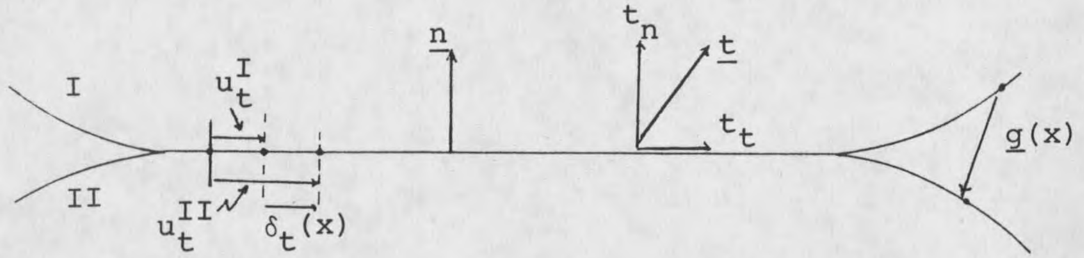
such that

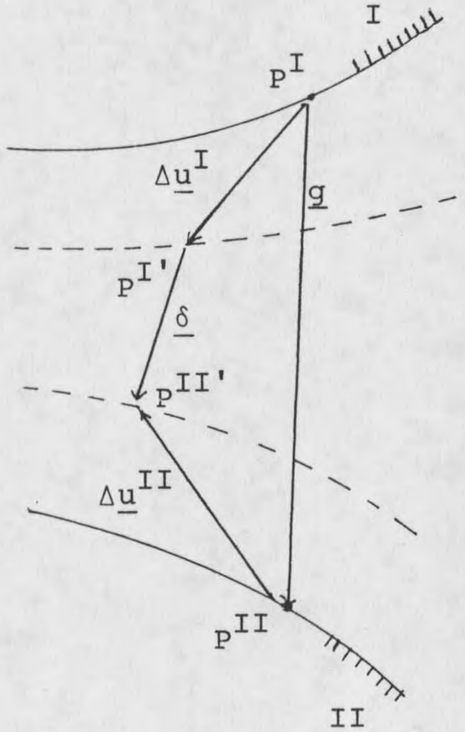
$$t_t(x) \cdot \delta_t(x) < 0 \tag{7.4b}$$

where $t_t(x)$ and $t_n(x)$ are the tangential and normal component of $t(x)$, respectively, μ is the coefficient of friction, and $\delta_t(x)$ denotes the current relative tangential displacement (slip) between bodies I and II (see Fig. 19). In Eq. (7.3), the equality for $\delta_t(x)$ provides a condition for displacements and the inequality defines the region of adhesion. Eq. (7.4a) determines the magnitude of the frictional forces in the region of slip. Eq. (7.4b) simply states that the frictional forces oppose the relative displacement.

Displacement Compatibility at the Contact Surface

If $\underline{u}^{I}$ and $\underline{u}^{II}$ are the incremental displacement vectors of a pair of potential contact nodes for the k^{th} load increment, and if $\underline{g}$ is the resultant gap vector between





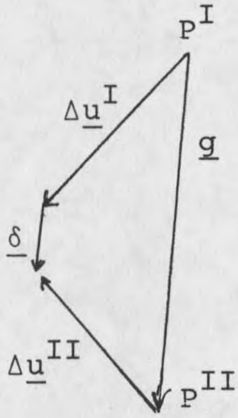


Figure 19. The contact zone c, and the gap vector $\underline{g}$, for frictional problems.

these nodes at the end of k-1 load increments, the normal component of the resultant gap δ_n can be defined as (Fig. 19),

$$\delta_n = \underline{\delta} \cdot \underline{n} = (\Delta \underline{u}^{II} - \Delta \underline{u}^{I} + \underline{g}) \cdot \underline{n}, \tag{7.5}$$

where

$$\underline{g} = \sum_{k=1}^{K} \Delta \underline{u}^{II\,k} - \sum_{k=1}^{K} \Delta \underline{u}^{I\,k} + \underline{g}^{0} \tag{7.6}$$

and $\underline{g}^{o}$ is the initial gap vector. δ_n should be non-negative because material interpenetration is not allowed. Hence, for contact to occur the condition is,

$$\delta_n = 0\,. \tag{7.7}$$

As the loading progresses the gap between the contact nodes is updated according to Eq. (7.6).

Equilibrium Equations at the Contact Surface

The normal and tangential components of tractions on the two bodies in the contact zone must be in equilibrium at each step of the loading process:

$$(E_1/E_2)\, \Delta t_t^{\,I\,k} + \Delta t_t^{\,II\,k} = 0\,, \tag{7.8}$$

$$(E_1/E_2)\, \Delta t_n^{\,I\,k} + \Delta t_n^{\,II\,k} = 0\,. \tag{7.9}$$

These equations are multiplied by the ratio Young's modulus of bodies I and II since, as described in Chapter 4, to obtain a numerically stable system of equations, the tractions of each body are divided by its Young's modulus.

Contact Conditions

The normal component of the current contact forces must be compressive:

$$t_n^{I k-1} + \Delta t_n^{I k} \leq 0 \text{ and } t_n^{II k-1} + \Delta t_n^{II k} \geq 0 . \qquad (7.10)$$

These equations are written in the local coordinate system defined in Chapter 5.

Frictional forces for either body I or II are obtained by substituting Eq. 7.2a in Eq. 7.3:

$$|t_t^{k-1} + \Delta t^k{}_t| \leq \mu |t_n^{k-1} + \Delta t^k{}_n| \qquad (7.11)$$

where $|t_t{}^{k-1} + \Delta t_t{}^k|$ and $|t_n{}^{k-1} + \Delta t_n{}^k|$ are the absolute values of the tangential and normal components of the current traction, respectively.

Stick and slip zones

The contact region Γ_c can be divided into two parts, the stick zone Γ_{ca} where no relative slip occurs and the slip zone Γ_{cs} where relative sliding occurs. In the stick zone Eq. 7.3 requires that

$$\delta_t = \Delta u^I{}_t - \Delta u^{II}{}_t - g_t = 0 \qquad \text{on } \Gamma_{ca} . \qquad (7.12)$$

In the slip zone the tangential traction for either body I or II is given by Eq. 7.4 which, with Eq. 7.2a reads as:

$$|t_t^{k-1} + \Delta t^k{}_t| = \pm\mu |t_n^{k-1} + \Delta t^k{}_n| \qquad \text{on } \Gamma_{cs} . \qquad (7.13)$$

The sign depends on the relative tangential displacement such that the frictional forces oppose the relative tangential displacement. Thus for body II,

$$(t_t^{k-1} + \Delta t^k{}_t)^{II} \cdot (\Delta u^I{}_t - \Delta u^{II}{}_t - g_t) < 0 \quad \text{on } \Gamma_{cs} . \qquad (7.14)$$

The relative displacement with respect to body I has opposite sign of the relative displacement with respect to that of body II.

Iterative Procedure and Convergence Criteria

Inside the potential contact zone both tractions and displacements are unknown, resulting in 8 unknowns per node pair. The potential contact zone may consist of open ($\delta_n > 0$), stick, and slip zones. Each zone has a different set of contact conditions, as described above, which are associated with different unknowns in each zone. The equations for the contact nodes of body I consist of two equilibrium equations for the tractions and the boundary integral equations for the displacement components. The equations for the contact nodes of body II change according to the status of the node. Table 3 shows the unknowns and the associated equations for each possible contact condition.

As described in Chapter 5, an initial assumption is made regarding the contact status of each node pair in the potential contact zone for the first load increment. The actual contact status of node pairs inside the potential contact zone for the first load increment is obtained through an iterative process in which the contact status of each node pair is checked and updated until Eqs. 7.7, 7.10 and 7.11 are satisfied. This then becomes the assumed contact status for the next load increment. For each load increment, the iterative procedure is repeated. The convergence criteria used during each iterative step are discussed below (see Table 4). All of the criteria are identified using tractions and displacements of body II.

At the node pairs that were assumed not to be in contact, if the resultant gap δ_n in Eq. (7.5) is a negative quantity, then the node pair is brought into contact for the next iteration. As the node pair is brought into contact, its status regarding sticking or sliding contact cannot, in general, be decided. Therefore, sticking contact is always assumed. After completion of the next iteration this assumption is verified before changing the contact status of any other node pair.

At the node pairs that were assumed to be in contact, if the normal traction is non-compressive then the node pair is released by setting the normal and tangential tractions to zero for the next iterations. As a node pair is released during the iterative process its incremental normal and tangential traction components are set to zero but not the total traction components, since the contact status of a node pair is finalized only after successful completion of the iterative process. The total traction components of the released nodes are then set to zero.

For node pairs that were assumed to be sticking, if Eq. (7.11) is satisfied, then the node pair continues to stick;

Table 3. Unknowns and corresponding equations at a potential contact node for frictional problems.

Unknowns at a potential contact node	Equations: Body I	Equations: Body II		
	All cases	Stick	Slip	Open
t_n	Equilibrium equn. (7.10)	Integral equation	Integral equation	$t_n = 0$
t_t	Equilibrium equn. (7.9)	Integral equation	Coulomb's law (7.13)	$t_t = 0$
u_n	Integral equation	Equation (7.5)	Equation (7.5)	Integral equation
u_t	Integral equation	Equation (7.12)	Integral equation	Integral equation

Table 4. Criteria for determining the frictional contact conditions during the iterative process.

Assumed contact status of a node pair	Current status of displacements and tractions of body II	Assumed contact status for next iteration
Open	$\delta_n > 0$	Open
	$\delta_n \leq 0$	Stick
Stick	$(t_n^{k-1} + \Delta t^k_n)^{II} \geq 0 : \lvert t_t^{k-1} + \Delta t^k_t \rvert < \mu \lvert t_n^{k-1} + \Delta t^k_n \rvert$	Stick
	$t_n^{k-1} + \Delta t^k_n)^{II} \geq 0 : \lvert t_t^{k-1} + \Delta t^k_t \rvert \geq \mu \lvert t_n^{k-1} + \Delta t^k_n \rvert$	Slip
	$(t_n^{k-1} + \Delta t^k_n)^{II} < 0$	Open
Slip	$(t_n^{k-1} + \Delta t^k_n)^{II} \geq 0 ; \delta^{II}_t (t_t^{k-1} + \Delta t^k_t)^{II} \leq 0$	Slip
	$(t_n^{k-1} + \Delta t^k_n)^{II} \geq 0 ; \delta^{II}_t (t_t^{k-1} + \Delta t^k_t)^{II} > 0$	Stick
	$(t_n^{k-1} + \Delta t^k_n)^{II} < 0$	Open

otherwise the status of the node pair is updated to sliding and Eq. (7.13) is used to evaluate the tangential traction. Note that the sign of tangential traction cannot, in general, be decided, and also the normal traction appears inside an absolute value sign. The sign of $\Delta t_t{}^k$ could be assumed to be the same as that of the total tangential traction when the stick condition is first violated and the known normal traction from the previous iteration is used in Eq. 7.13, giving:

$$\Delta t_t{}^{II}{}_k = (t_t{}^{k-1} + \Delta t_t{}^{0(k)})^{II} / |(t_t{}^{k-1} + \Delta t_t{}^{0(k)})|^{II} \times$$
$$[\mu |t_n{}^{k-1} + \Delta t_n{}^{0(k)}|]^{II} - t_t{}^{II\,k-1} \qquad (7.15)$$

where the superscript o denotes value from the previous iteration. This obviously requires an additional iteration to correctly determine the tangential traction. Alternatively, since the sign of total normal traction is known to be non-negative by Eq. 7.10, the absolute value sign on this quantity can be dropped and Eq. (7.13) can be modified by treating both incremental normal and tangential tractions as unknown quantities. The modified equation is

$$\Delta t_t{}^{II}{}_k + (t_t{}^{k-1} + \Delta t_t{}^{0(k)})^{II} / |(t_t{}^{k-1} + \Delta t_t{}^{0(k)})|^{II} \, \mu \Delta t_n{}^{II}{}_k =$$
$$-(t_t{}^{k-1} + \Delta t_t{}^{0(k)})^{II} / |(t_t{}^{k-1} + \Delta t_t{}^{0(k)})|^{II} (\mu t_n{}^{II\,k-1}) - t_t{}^{II\,k-1} \qquad (7.16)$$

In the case when this assumption fails, an extra iteration is needed to correct the sign of the tangential traction.

For node pairs that were assumed to be in sliding contact, the frictional force opposes the relative sliding between the contacting nodes. Thus, by Eq. 7.14 two conditions can exist:

$\delta^{II}{}_{t}\cdot(t_t{}^{k-1} + \Delta t^k{}_t)^{II} \leq 0$ in which case the node pair continues to slide, or,

$\delta^{II}{}_{t}\cdot(t_t{}^{k-1} + \Delta t^k{}_t)^{II} > 0$ in which case the status of the node pair is updated to sticking contact.

Since the contact status of a node pair is decided by the Eq 7.5 and the inequalities 7.10 and 7.14, the iterative process may fail due to round off and truncation errors. If for a given load increment the iterative process fails to converge then the solution is discarded and the load increment is changed by a predefined amount. The final tractions and displacements are found from Eq. 7.2.

Operation of Equation Solver

In order to develop an efficient computer program, clearly, the scheme used to solve the system of equations must be efficient. This is especially true in contact problems where the system of equations may have to be solved forty to fifty times before the final solution is achieved. As in any numerical solution technique, core storage requirements are also of major concern.

To address these concerns, equation assembly is done such that a blocked coefficient matrix results. In the BIE method, both tractions and displacements are retained as unknowns hence, the contact equations can be written explicitly. In contact problems since two regions occur naturally, equations for the potential contact zone and those outside the contact zone can be assembled separately, resulting in a blocked coefficient matrix.

In contact problems, nodes in the potential contact zone Γ_C require modification of boundary conditions during the iterative process. However, nodes outside the potential contact zone are not affected by the incremental-iterative solution process. If Gaussian elimination is used, after partial forward elimination, the equations pertaining to the nodes outside of the potential contact zone can be transferred to a disc and, once the actual contact zone is established, can be retrieved for back substitution to find the boundary unknowns outside of the potential contact zone, if desired (see Fig. 20).

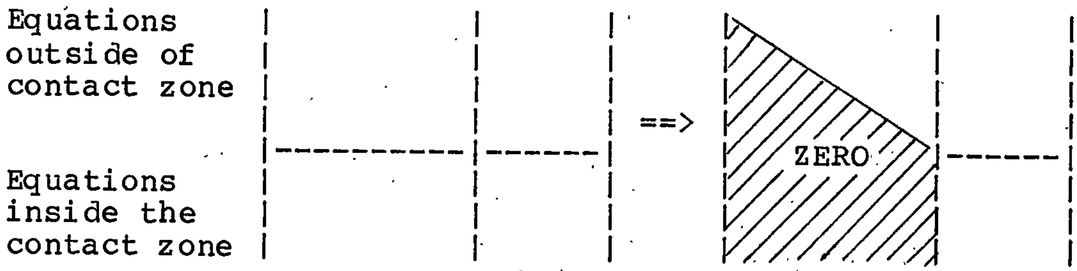


Figure 20. Partial forward elimination of the system of equations.

At the start of the loading process, the contact status of node pairs inside the potential contact zone must be assumed. For simplicity, each node pair in the potential contact zone is assumed to be in sticking contact.

System of Equations

Let M_I be the number of nodes outside of the potential contact zone for body I. Let N be the number of nodes in the potential contact zone for each body. The total number of unknowns outside of the potential contact zone is $2M_I + 2M_{II}$, and the total number of unknowns in the potential contact zone is 8N (four unknowns for each node). The resulting system of equations is

$$[K]\{X\} = \{R\}, \qquad (7.17)$$

where [K] is the coefficient matrix, {X} is the unknown solution vector, and {R} is the known right hand side. The general form of Eq. (7.17) for a two body contact problem is illustrated in Figure 21a, where K_{IJ} is a sub-matrix (block). The blocks X_1 and X_2 contain unknowns outside of the potential contact zone for body I and body II, respectively, where as X_3 and X_4 contain unknowns in the potential contact zone for bodies I and II, respectively. Thus, the first block row of [K] contains the influence coefficients resulting from the integral equations outside of the potential contact zone of body I which, therefore, has non-zero entries only in K_{11} and K_{13}. Similarly, the second block row has non-zero entries only in K_{22} and K_{24}. The third block row consists of the integral equations followed by the contact equations corresponding to the potential contact nodes of body I, in which block K_{34} provides the coupling between bodies I and II. Note that the integral equations have non-zero entries in only blocks K_{31} and K_{33}, while the contact equations have non-zero entries in only blocks K_{33} and K34. A similar description holds for fourth block row.

Gauss Elimination

Eq. (7.17) is reduced by standard Gaussian elimination algorithm in which K and R are modified as follows:

$$k'_{jl} = k_{jl} - (k_{ji}/k_{ii})k_{il} \; ; \qquad i = 1, 2, \cdots, n \qquad (7.18)$$
$$j = i+1, 2, \cdots, n$$
$$l = i+1, 2, \cdots, n .$$

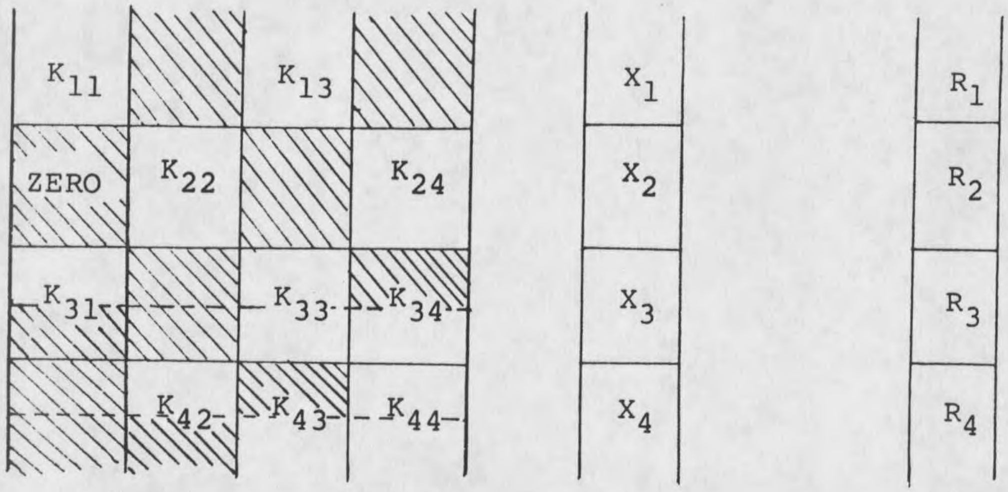
$$r'_j = r_j - (k_{ji}/k_{ii})r_i \; ; \qquad (7.19)$$

where n is the number of equations, k_{ij} is the element in the i^{th} row and j^{th} column of [K], and r_i is the i^{th} element of {R}. {X} is obtained by back substitution in [K]' as follows:

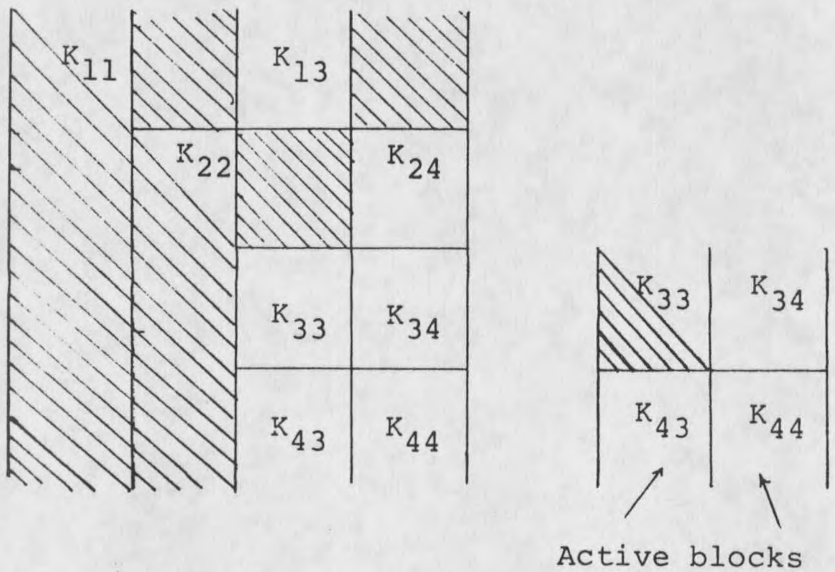
$$x_i = \left(\sum_{j=i+1}^{n} k'_{ij}x_j\right) / k'_{ii} \; ; \qquad i = n-1, n-2, \cdots, 1 . \qquad (7.20)$$

Elimination of Blocks

Each block row is assembled in parts and is stored in a workspace array; zero blocks are not stored. Sparse matrix principles are used in performing forward elimination. In reducing successive diagonal blocks (called pivot blocks) operations on block rows with a zero block in the pivot column are not performed. For example, in performing forward



a. The blocked coefficient matrix.



b. The coefficient matrix after partial forward elimination.

Figure 21. System of equations for two body contact problems.

elimination on the first pivot column, block rows 2 and 4 remain unaltered. Similarly, zero blocks in the pivot row will not alter the blocks below them. Therefore, during elimination of the first pivot column, the only active blocks are K_{11}, K_{13}, K_{31}, and K_{33}.

Forward elimination on each block row is performed successively and is transferred to a sequential file disc. The first block row is then retrieved to perform forward elimination of the third block row, while the second block row is retrieved for the fourth block row. A partially reduced system of equations containing the modified blocks K_{33}, K_{34}, K_{43}, and, K_{44} is obtained (see Fig. 21b).

Reduced System of Equations

During the incremental-iterative process, only the modified blocks K_{33}, K_{34}, K_{43} and, K_{44} are needed. The blocks K_{33} and K_{34} contain coefficients corresponding to the contact zone of body I, which before partial elimination consist of the boundary integral equation and the equilibrium equation coefficients (see Table 3). Since the equilibrium equations (7.8) and (7.9) have no non-zero entries in block K_{31}, these equations remain unaltered during partial forward elimination. While after partial forward elimination, the integral equations are left with modified coefficients in block K_{33}. Furthermore, since the integral equations (7.1) and the equilibrium equations (7.9 and 7.10) are written in the incremental form, the modified blocks K_{33} and K_{34} do not change during the entire incremental-iterative process. The block R_3 contains coefficients resulting from the known incremental tractions and incremental displacements and thus depends on the load increments. In Eq. (7.19), the elements r_i appear only in linear combinations. Since only proportional loading is considered, during each load increment the elements of $R1$, R_2 and R_3 all change in the same proportion. Therefore forward elimination on R_1, R_2 and R_3 is performed only once during the first load increment. For each successive load increment the block R_3 is multiplied by the ratio of the current load increment to the previous load increment, and the iterative process is resumed. Thus, further partial reduction of the system of equations is achieved during the first iteration of the first load increment (see Fig. 21b).

Active Blocks

The blocks K_{43} and K_{44}, contain integral and contact equations for body II. The contact equations (7.5, 7.12 and 7.15) contain current gaps and current tractions, which are history dependent. Hence, these blocks remain active during the incremental-iterative process. At the start of the problem these blocks contain coefficients for sticking contact. In each iteration the coefficients in these two blocks change according to the current contact status of each node (see Tables 3 and 4). To accomplish this, the original coefficients (for sticking contact) are retrieved and changed in each iteration. In doing so, further saving in computation time is achieved by the fact that in forward elimination only the equations below the updated equations are affected. Thus, after the decision regarding contact status of each node is made, only the equations for the nodes whose contact status changes, and all the equations below them, are retrieved and updated. The remainder of the equations which have been reduced in the previous iteration are kept unchanged.

For a new load increment, all the equations in blocks K_{43} and K_{44} are changed according to the contact status at the end of previous load increment. Since the elements of block R_4 relating to the boundary integral equations are not history dependent and since proportional loading is assumed, these elements are multiplied by the ratio of the current load increment to the previous load increment as explained for block R_3 and the iterative process is resumed. Since contact status is based on current tractions and current displacements (not their increments), for the first iteration of the current load increment, the right hand side of the contact equations (block R_4) is updated by adding the incremental displacements and tractions found in the previous load increments (see Eqs. 7.6 and 7.13).

Back Substitution

Once the iterative-incremental process is complete, the first two row blocks may be retrieved to find the boundary unknowns outside of the potential contact zone. The blocks R_1 and R_2 are multiplied by the ratio of the final load to the first load increment and the boundary unknowns are found by back substitution.

The blocks pertaining to contact zone can be handled more efficiently. Half of the equations in blocks K_{33} and K_{34} consist of contact conditions which contain very few non-zero entries. Rearrangement of these equations can lead to further saving of the storage space. A proper pivotal choice is also important. It can restrict the non-zero entry growth, thus increasing the potential for computer time savings. Numerical stability also depends on choice of the pivot. Rounding errors can occur if matrix entries of large magnitude are created by using a small pivot. Algorithms for the solutions of large unsymmetric equation systems are now available (Hood, 1976, Stabrowski, 1981). To further increase the efficiency of the solver, techniques used in these algorithms should be incorporated.

CHAPTER 8

RESULTS FOR FRICTIONAL CONTACT PROBLEMS

Solutions to three different types of frictional contact problems which cover the essential features of frictional contact are presented in this chapter. The BIE solutions are checked with available analytical or numerical solutions. As mentioned in Chapter 2, the only available analytical solutions for frictional problems are for stationary contact. There do not exist any analytical solutions for checking numerical solutions of non-stationary frictional contact problems.

An Elastic Rectangle Compressed by Rigid Planes

The plane strain compression of an elastic rectangle by rigid, rough planes with finite coefficient of friction between the surfaces is considered (Fig. 22). A rectangular domain of height 2a and width 2h is compressed by applying a load of intensity P through rigid punches whose width exceeds the width of the elastic rectangle. Fig. 22 shows the boundary element model in which symmetry is taken in account. The contact zone extends over the width of the rectangle and it is modeled by using 24 linear elements. It is further assumed that the load is applied monotonically so that the direction of slip does not change as the load

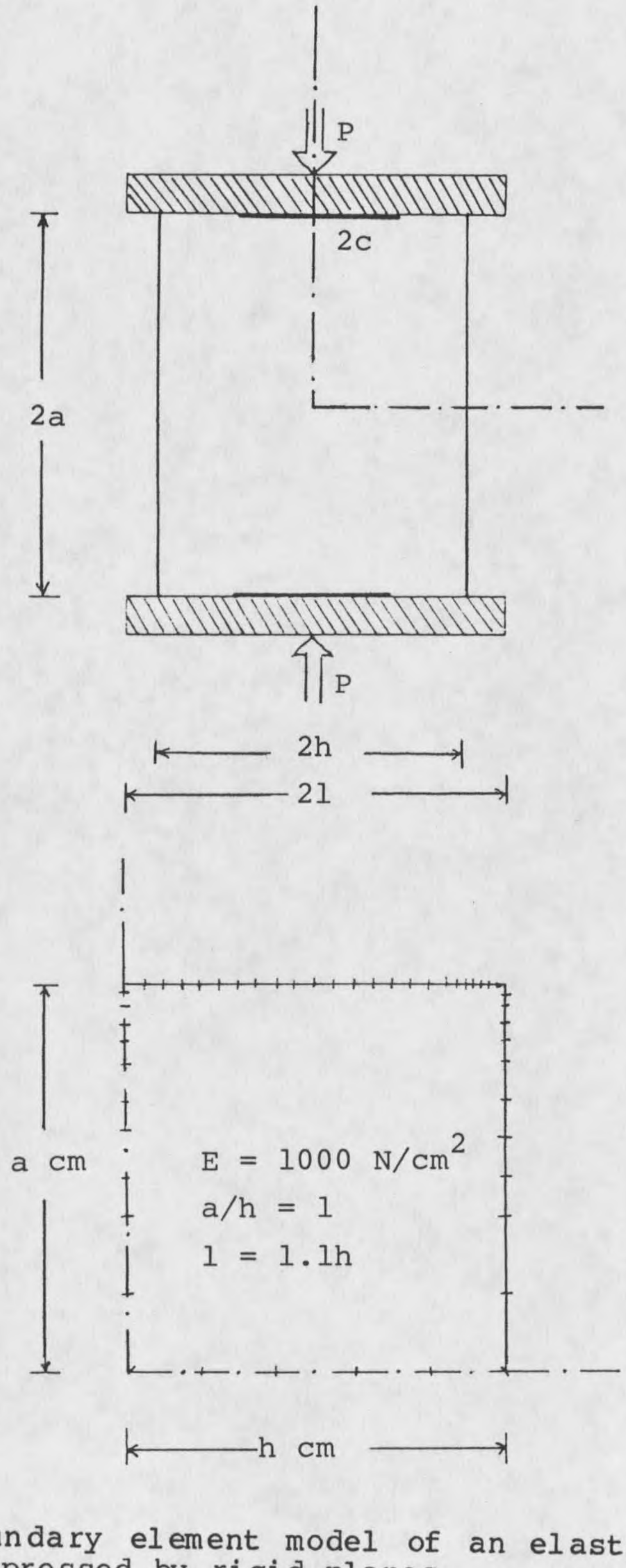


Figure 22. Boundary element model of an elastic rectangle compressed by rigid planes.

changes. The problem is thus load path independent (Prasad and Dasgupta, 1975). The extent of stick zone, c, is a function of the coefficient of friction, μ, the Poisson's ratio, ν, and the depth of the rectangle, a. c does not depend on the magnitude of the applied loading P.

Fig. 23 shows the variation of normal tractions inside the contact zone for various values of μ and ν. The strength of stress singularity increases with increase in μ, but the effect of Poisson's ratio, ν, on the strength of stress singularity is found to be negligible (Prasad and Dasgupta, 1975). The BIE results are in very good agreement with the analytical solution. Fig. 24 shows the variation of frictional forces for different values of μ and ν. The lateral displacement is a function of ν, and for a frictionless case the lateral displacement increases as ν is increased. Thus, the frictional forces, which oppose the lateral displacement, increase with increase in ν. This can be observed from 3rd and 4th graphs of Fig. 24, the frictional forces, which are independent of μ in the stick zone, increase as ν increases. The change in the slope of frictional forces shows the transition from stick to slip zone. The difference in analytical and numerical results for frictional forces is due to the fact the extent of stick zone depends on the element size as explained in the following paragraph.

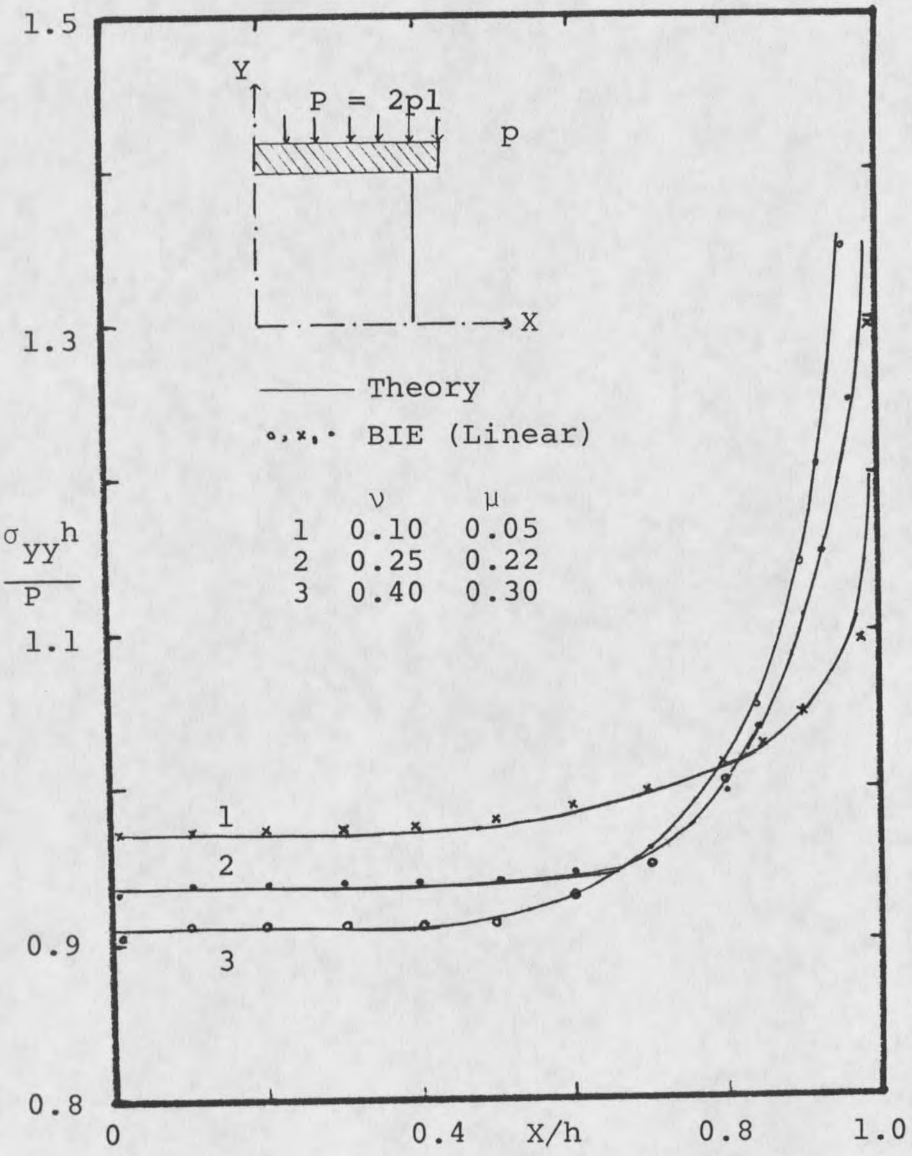


Figure 23. Variation of contact pressure for various values of μ and ν.

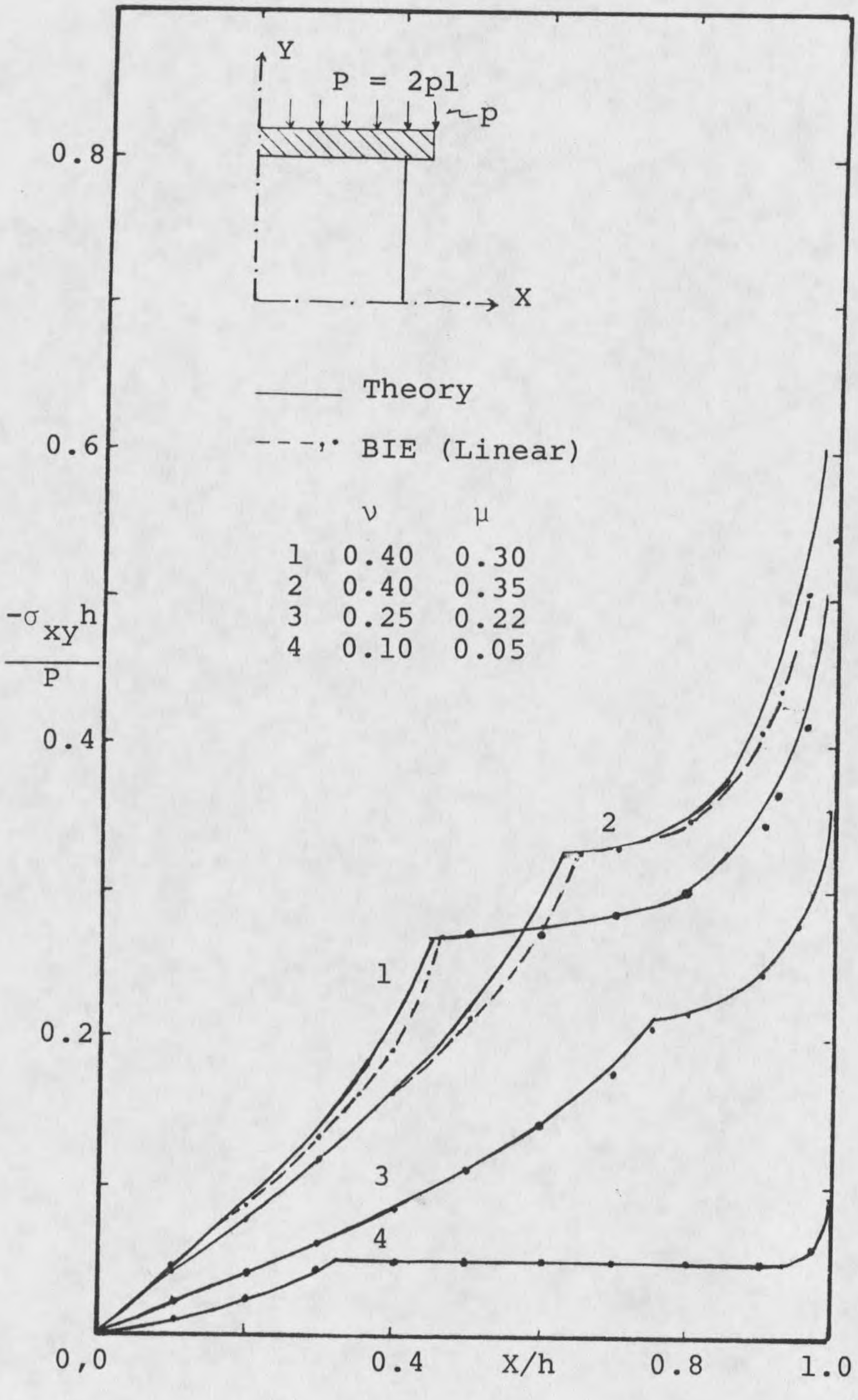


Figure 24. Variation of frictional forces for various values of μ and ν.

Fig. 25 shows the extent of the stick zone as a function of the friction coefficient, μ, for various values of the Poisson's ratio ν. As ν increases, frictional forces increase, resulting in a reduced stick zone for a given μ. The extent of stick zone cannot be determined precisely through numerical computation: a contact node either sticks or slips, but the actual transition from stick to slip may take place between two nodes. Thus, the accuracy of the numerical results depends on the element size. Fig. 25 shows the actual size and location of the element in which the transition from stick to slip occurs. As can be seen, the BIE solution correctly predicts the element in which the stick-slip transition occurs in all but two cases. The present BIE technique cannot model the corner stress singularity, nevertheless, the BIE solutions still correctly predict the extent of stick zone near the stress singularity ($x/h = 1$).

The dotted line shown in Fig. 25 is for a plane stress solution, with $\nu = 0.4$. Since the displacements for plane strain and plane stress problems differ, the frictional forces also differ in these two cases. As noted in Chapter 3, a plane stress solution can be obtained from a plane strain formulation by replacing the value of ν by $\nu/(1+\nu)$. This results in an effective Poisson's ratio that is less than the corresponding Poisson's ratio for plane strain, and hence smaller frictional forces.

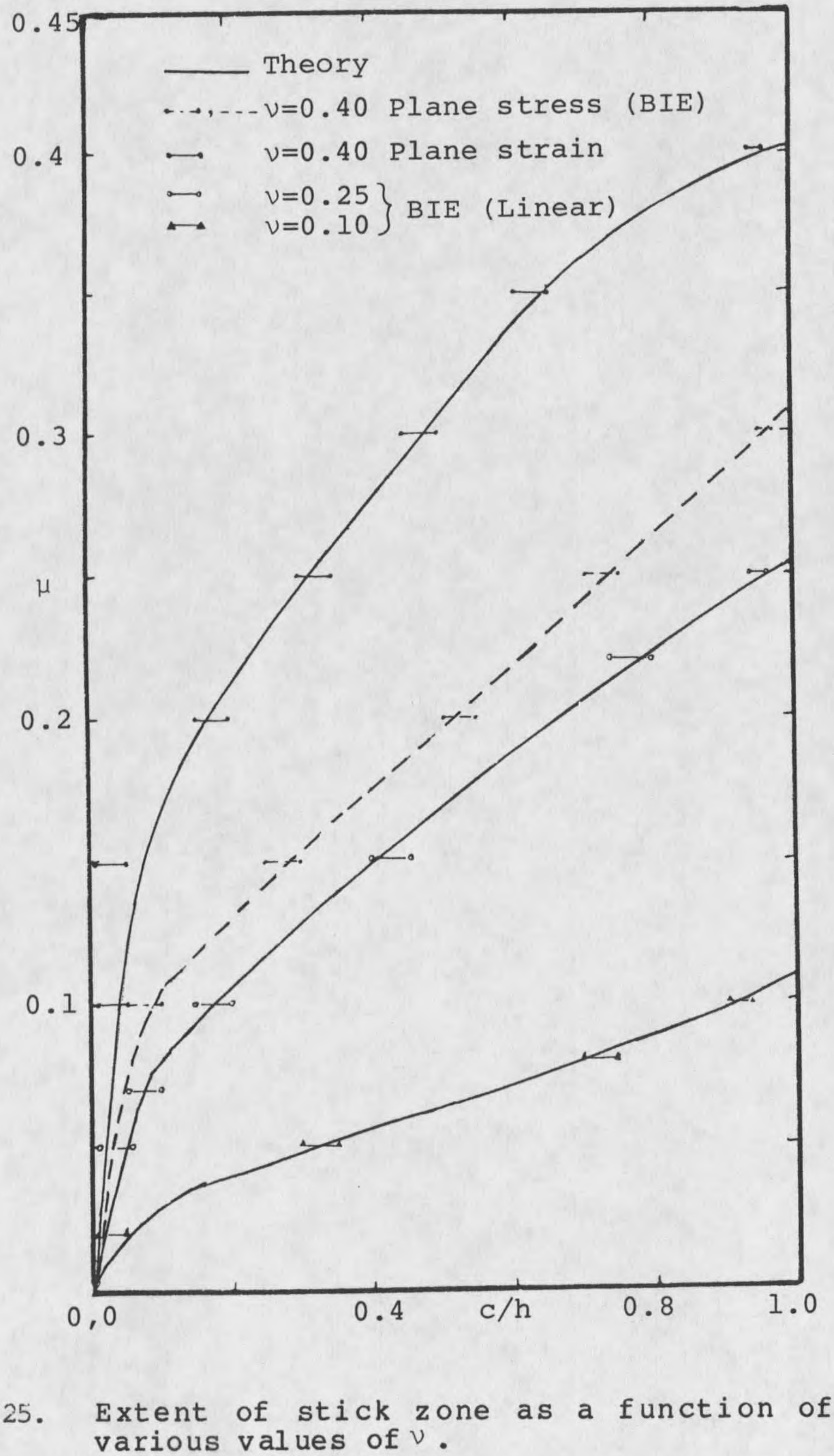


Figure 25. Extent of stick zone as a function of μ for various values of ν.

The Frictional Unloading Problem

The plane strain indentation of an elastic half space by a rigid punch, with a finite coefficient of friction, μ, is considered. The monotonically applied loading problem is solved analytically by Spence (1973). The results for the loading case are similar to the previous problem and are not presented here. In this problem, the extent of stick zone c is also determined in terms of μ and the Poisson's ratio ν, and is independent of the magnitude of the applied loading.

The results shown here are for the frictional unloading problem. A numerical solution obtained by Turner (1979) using a variational formulation provides the basis for comparison. Fig. 26 shows the boundary element model for this problem, for which the contact zone is modeled with 30 linear elements.

At the end of the loading phase, when the current load, W, equals P, the extent of the stick zone c is 0.7, for μ equal to 0.265. Fig. 27 shows the frictional forces for the half space. In the slip zone, the half space displaces laterally inward (inward slip), so the frictional forces are directed outward. It is of interest to note that in the previous problem due to the presence of a free edge which is perpendicular to the contact zone, the elastic rectangle displaces laterally outward. In this case the material is displaced inward due to the continuity of the surface, a part of which forms the contact zone.

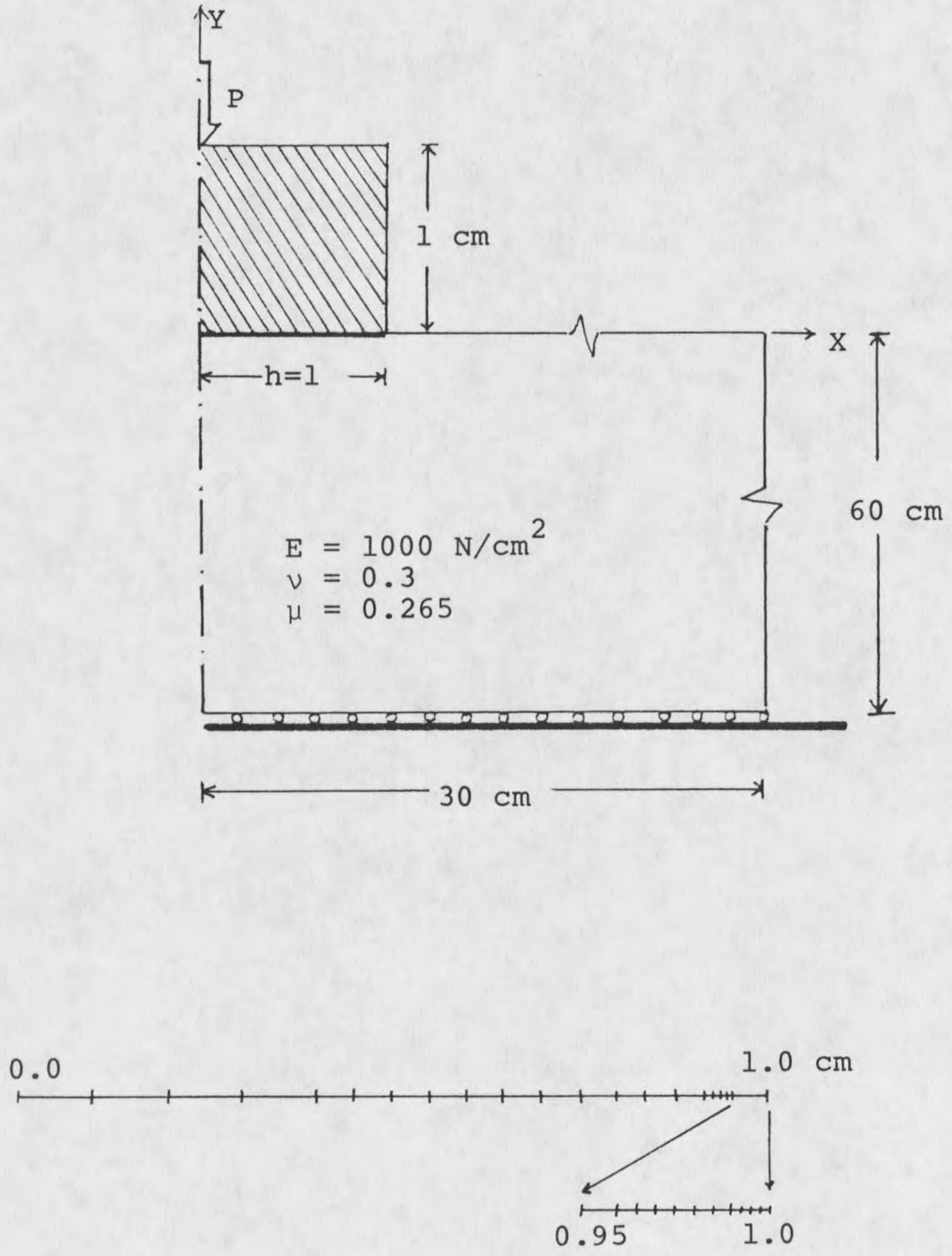


Figure 26. Boundary element model of a rigid punch on an elastic half space.

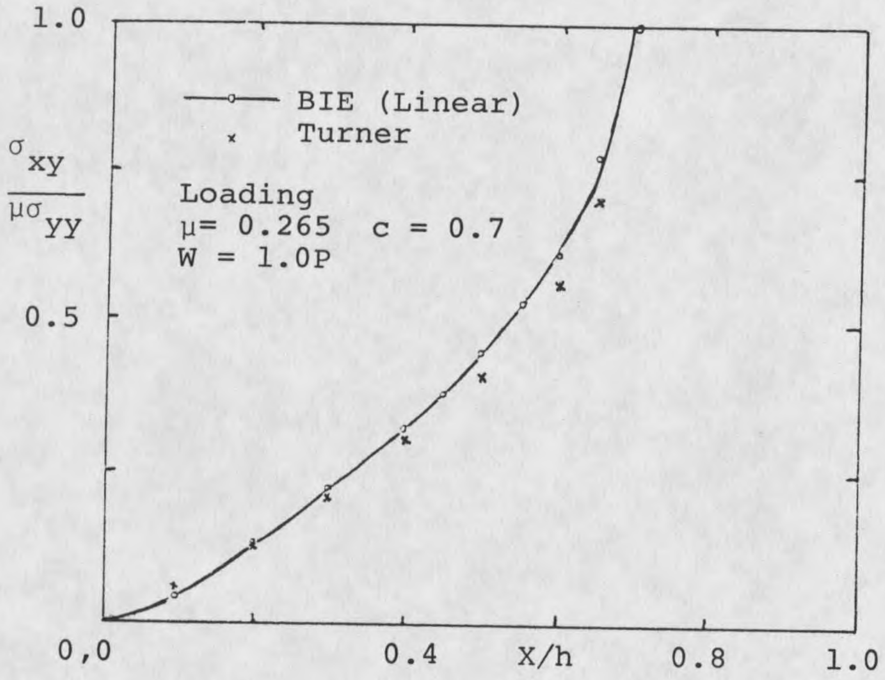


Figure 27. Frictional forces for the loading phase, W = P.

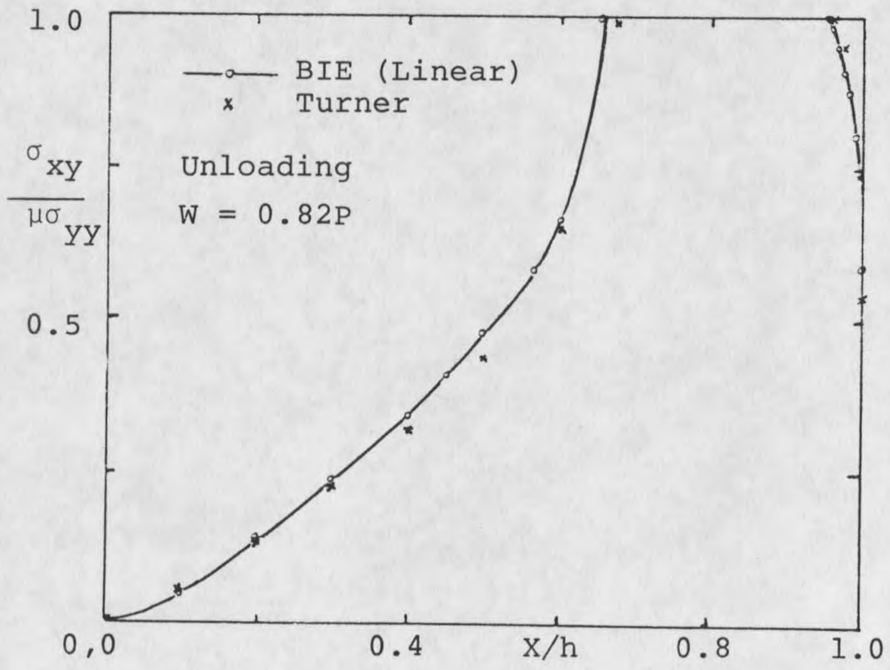


Figure 28. Unloading phase at a load W = 0.82P.

The unloading problem is load path dependent. The problem is solved by decreasing the load in equal step sizes of 0.05P. Fig. 28, 29 and 30 compare the frictional forces during the unloading phase with the results obtained by Turner (1979). The BIE solution is found to be within 5 percent of the solution obtained by Turner (1979).

The complete unloading phase is compared in Fig. 31. The different stick zones, inward and outward slip zones are plotted against a load factor W/P. As W decreases from 1.0P, initially, a stick zone is formed near the edge of the rigid punch. This stick zone remains of constant width till the load is reduced to W = 0.52P. Further reduction in load moves this stick zone inward and a region of outward slip is formed near the edge. Once the region of outward slip has formed the outer stick zone expands into the zone of inward slip.

Since the BIE results depend on the location of nodes, the BIE solution predicts a larger outer stick zone. The difference is about 3 percent. As explained earlier, the transition from one zone to another zone may take place between two nodes altering the results slightly. The unloading phase is continued till W = 0.1P, the dashed line in Fig. 31 is the extrapolated solution. The element size near the center of the rigid punch is too large to accurately predict different regions of stick and slip (Fig.

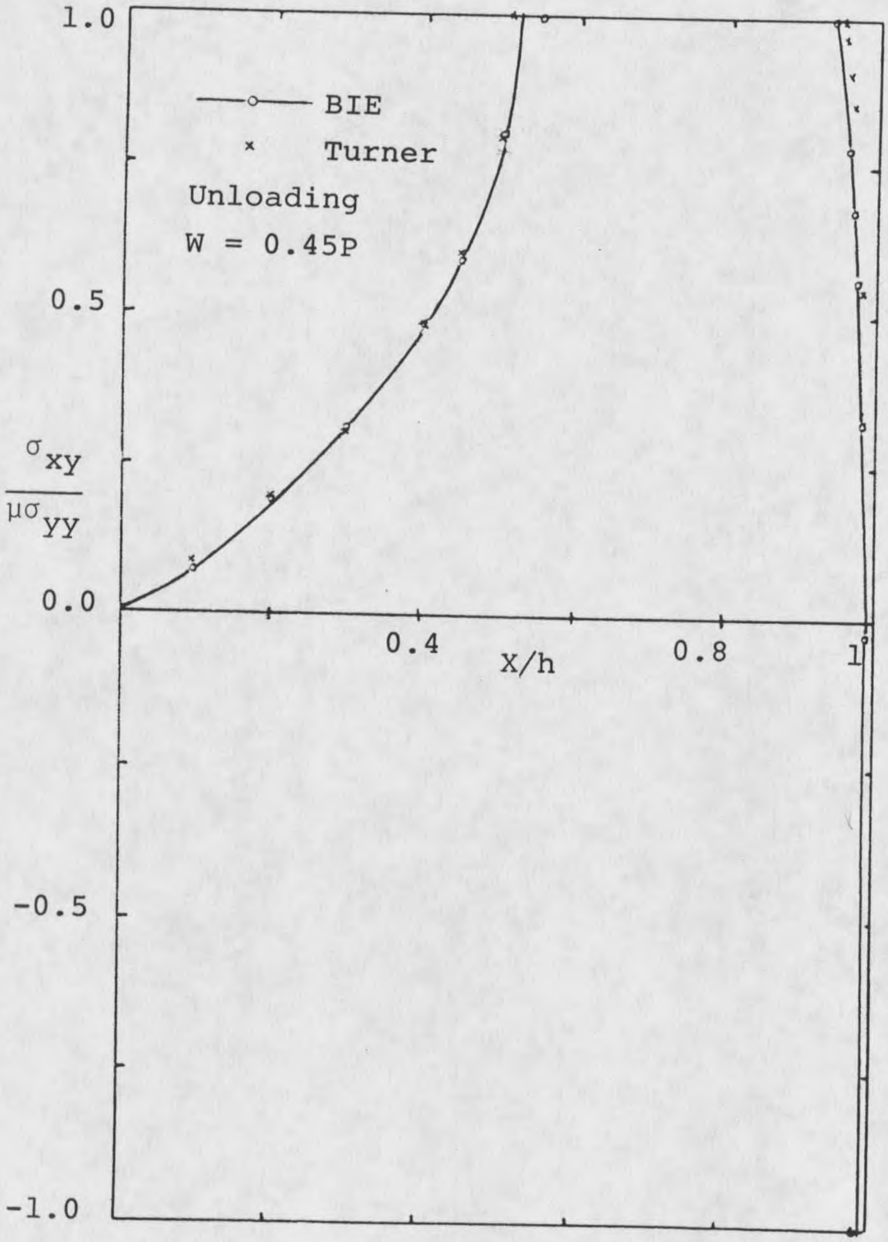


Figure 29. Unloading phase at a load W = 0.45P.

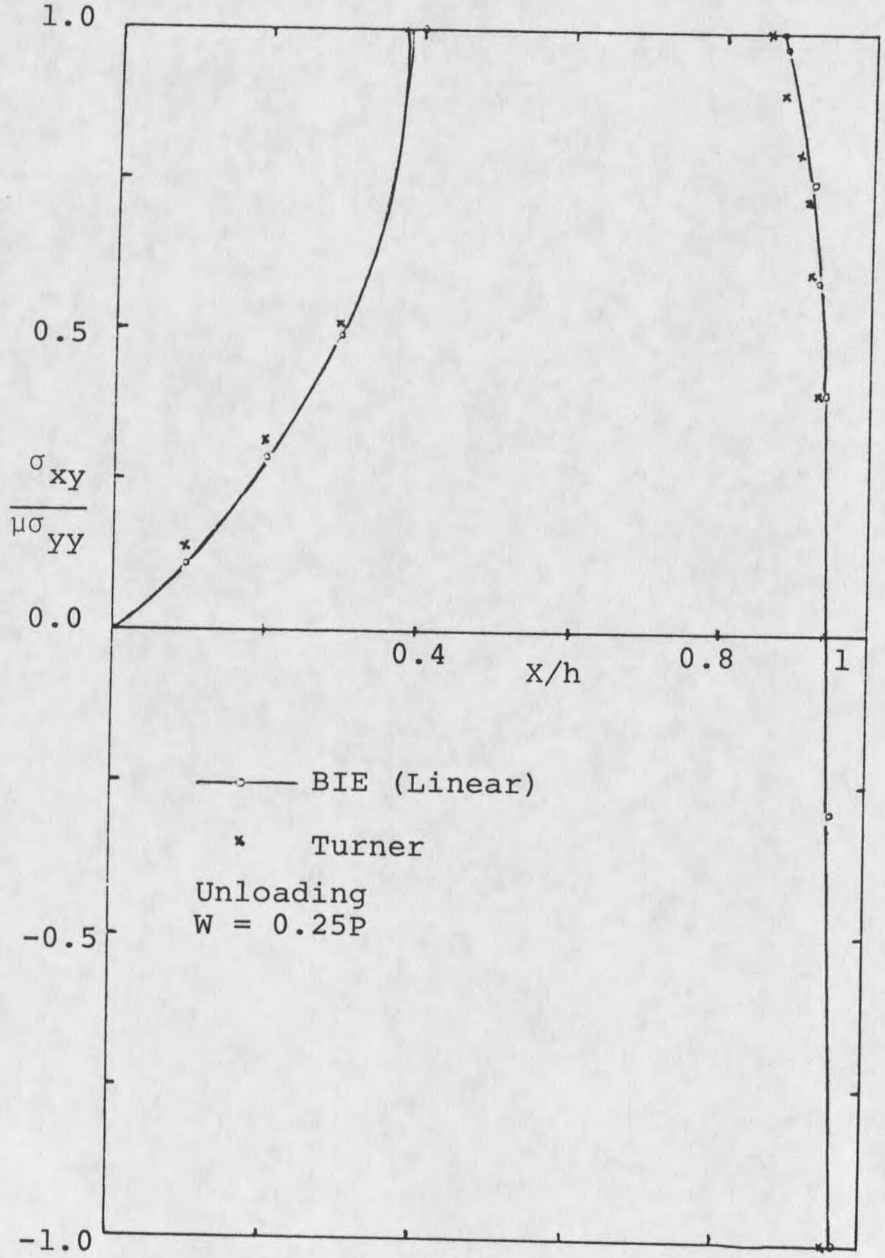


Figure 30. Unloading phase at a load W = 0.25P.

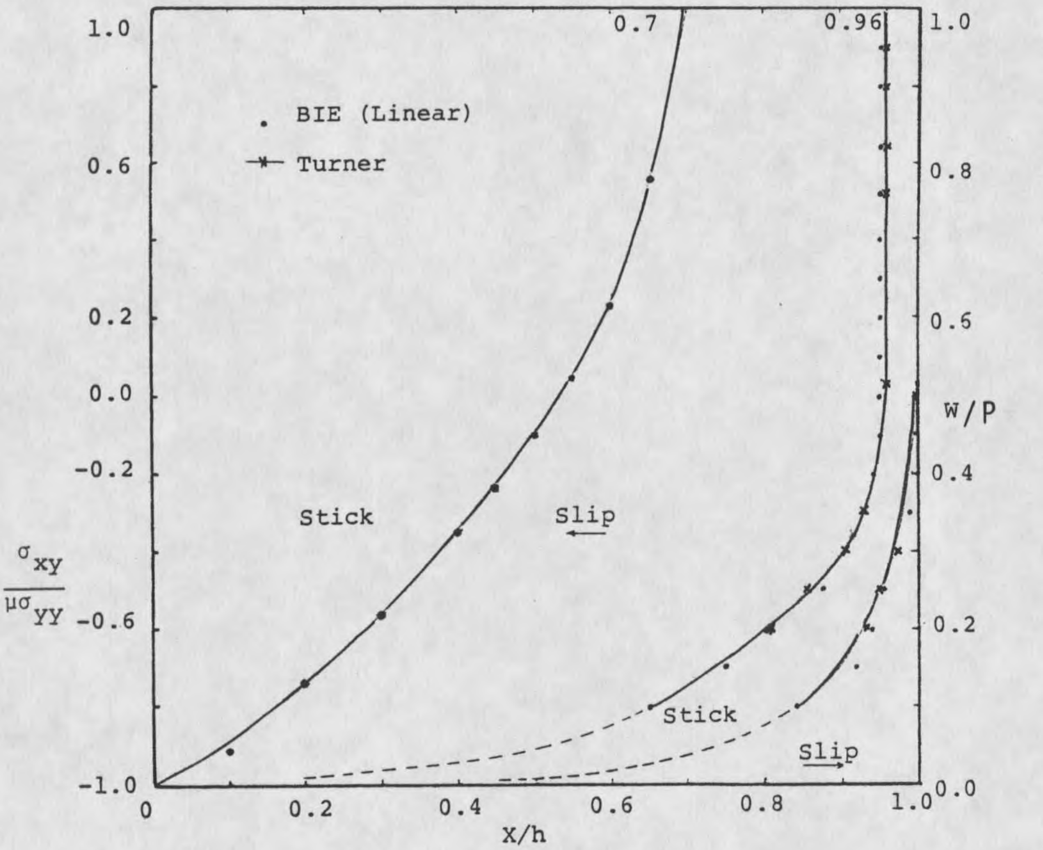


Figure 31. The complete unloading phase compared with Turner (1979).

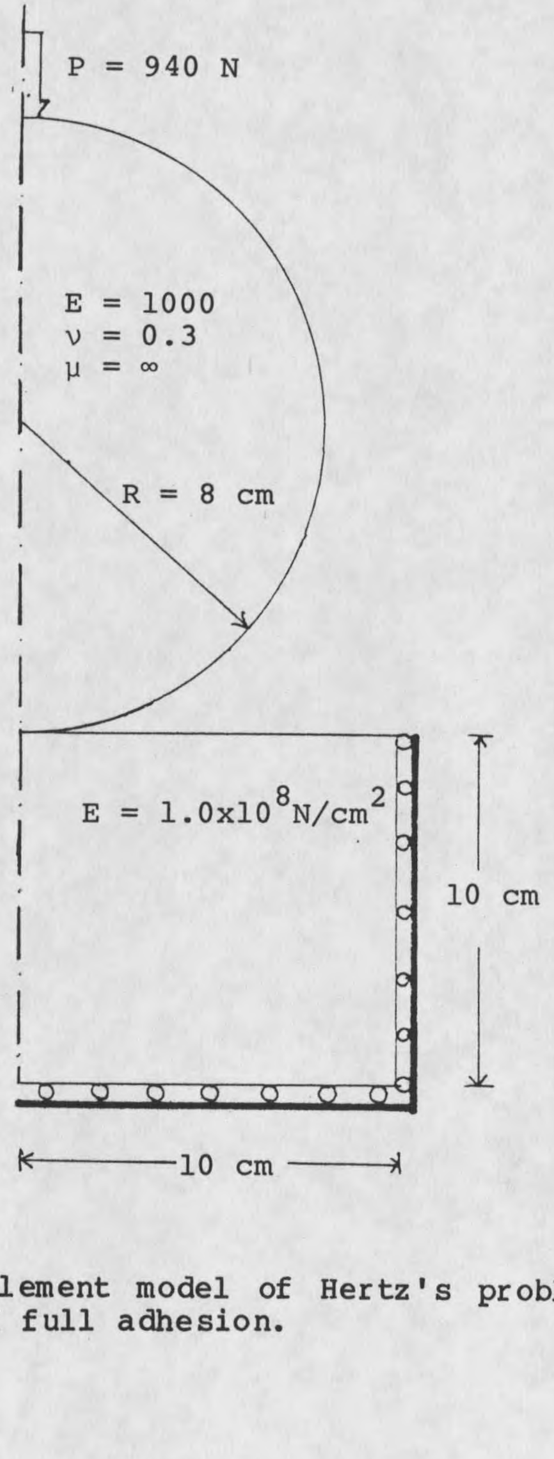


Figure 32. Boundary element model of Hertz's problem for the case of full adhesion.

26). Hence, further unloading will not yield a good numerical solution.

Neither the BIE solution nor Turner's solution explicitly model the corner stress singularity. The numerical results obtained give finite values for the tangential and normal stresses at the edges and, as a result, a stick zone is formed near the edges at the onset of the unloading phase. According to Spence (1973), since both the tangential and the normal stresses near the edge under stick condition are unbounded, some slip must occur near the edges of the punch at the slightest trace of the applied pressure. Development of a corner singularity element, as has been done by Sweldlow and Solecki (1979) in the finite element analysis, should correct this problem.

Hertz's Problem with Friction

Consider the plane strain problem of a cylinder resting on a rigid foundation (Fig. 32). In Chapter 5, a similar problem without friction has been solved. Here, it is assumed that the coefficient of friction is sufficiently high to prevent slip. This problem is load path dependent. To see why this is so, consider two points on each surface, with a normal gap g between them (Fig. 33). Initially they are not in contact. As the loading progresses the cylinder is deformed, displacing the point P both normally and laterally. At some load, the normal displacement is sufficient to close the gap g, bringing the point in contact at P'. Subsequent increase in the load produces no lateral displacement. The numerical solution, however, must reflect the entire lateral displacement history of the points under consideration before they come in contact. This requires the load to be applied in increments so that the extent of contact increases gradually. The process remains essentially same even if the coefficient of friction is small enough to form a region of slip.

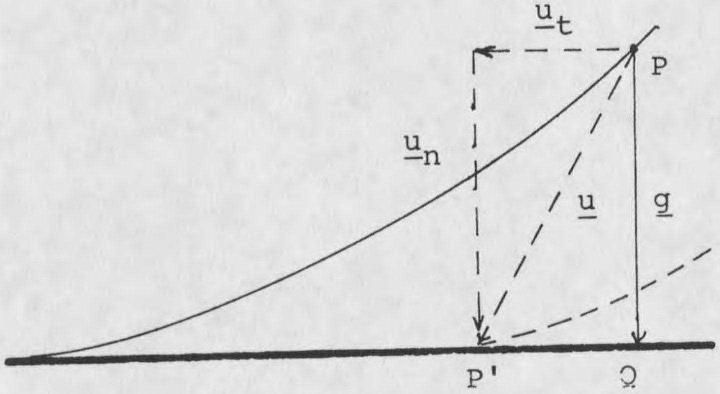


Figure 33. Lateral and normal displacement of a point P on a cylinder (schematic).

The boundary element model is shown in (Fig. 32). The potential contact zone is modeled with 15 equal size linear elements of 0.2 cm length. The problem is solved with 200 equal size load increments. Fig. 34 shows the normal and the tangential stresses inside the contact zone. There is no analytical or numerical solution available for this problem. Goodman (1962) has solved the axisymmetric problem of two normally loaded rough spheres. His results are shown in Fig.

35 and the BIE results for the plane strain case agree qualitatively. The maximum normal stress for the frictionless case is about 5 percent less than the adhesive problem considered. The increase in normal stresses is accompanied by a decrease in contact area so that the total area under the normal stress curve is the same as for the frictionless problem. This resultant force must equal the total applied force which is the same for both the cases. For the frictionless case the cylinder has inward lateral displacements which, in the case of full adhesion, are opposed by the outward frictional forces. These forces reduce the contact area. The overall effect of friction on normal pressure and the contact area is negligible. Campos et. al. (1982) numerically solve the axisymmetric contact problem of a sphere resting on a foundation without considering the load path dependence. They also report an increase in contact pressure of around 6 percent due to the presence of friction.

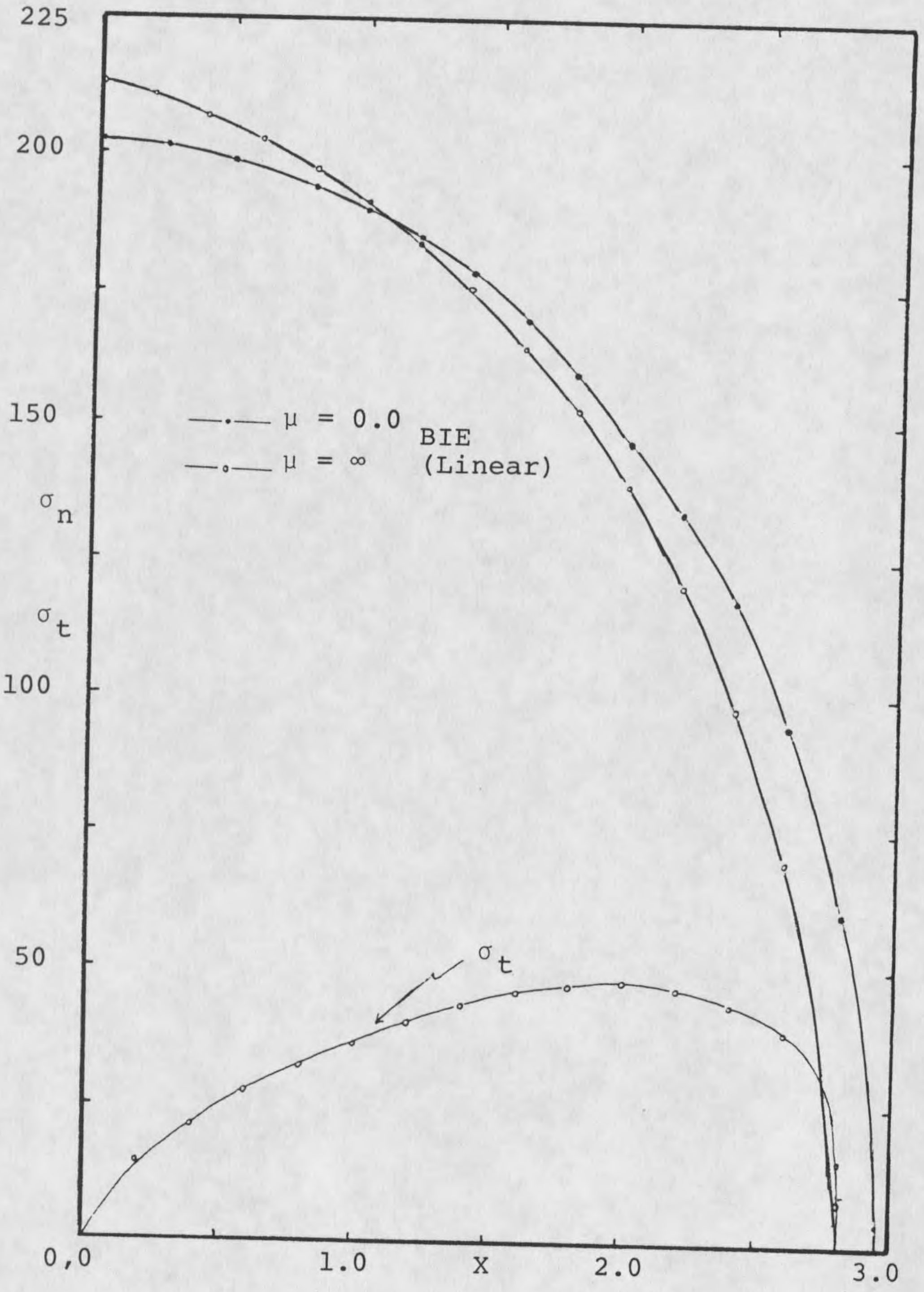
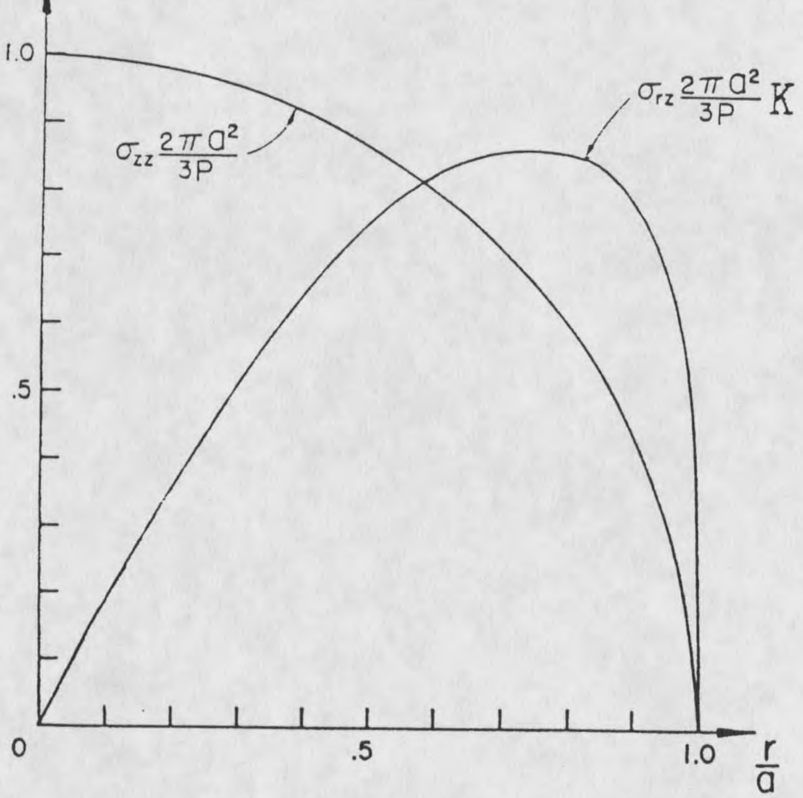


Figure 34. Contact pressure for the frictionless and the case of full adhesion and, frictional forces for the case of full adhesion.



$$K = \pi\left[\frac{1-\nu_1}{\mu_1} + \frac{1-\nu_2}{\mu_2}\right] \div \left[\frac{1-2\nu_1}{\mu_1} - \frac{1-2\nu_2}{\mu_2}\right]$$

Figure 35. Shear and normal stress components inside the contact zone for two rough spheres in contact, where P is the applied load (Goodman, 1962).

CHAPTER 9

NUMERICAL MODELING OF CONTACT PROBLEMS

The validity of numerical analysis largely depends upon its consistency in producing satisfactory results for a variety of problems. It is important that the computer program is tested thoroughly with a wide range of problems involving different boundary conditions, domains and various loading conditions. These results are generally checked with analytical solutions, other numerical techniques, or with experimental observations. Unfortunately, when the numerical analysis of contact problems is considered the analyst is often confronted with a lack of test problems. Several investigators have used finite element methods to solve contact problems with considerable success; these are discussed in detail in Chapter 2. However, no systematic study is available which takes into consideration the various aspects of contact problems. The problems presented are either isolated or insufficiently explored. For example, several investigators present the Hertz's problem with finite friction, but only discuss the effect of friction on normal pressure distribution inside the contact zone and neglect tangential tractions (Okamato and Nakazawa, 1979, Sachdeva, et. el., 1981, and Chaudhary and Bathe, 1985). In this chapter certain aspects of contact problems which are important in constructing the numerical model are discussed.

Contact problems are highly non-linear and some physical insight is necessary to obtain numerical solutions to these problems. The results of a particular problem are usually affected by following factors:

1. Geometry of the problem.
2. Boundary conditions.
3. Material parameters. This aspect is important since there are two or more bodies involved (de Pater and Kalker, 1975).
4. Coefficient of friction μ.
5. History of the applied load.

In addition, when the problems are solved numerically with the FEM or the BIE method the following factors also become important:

6. Functional approximation (type of the element used).
7. Element size and location of the nodes.

In Chapter 5, it was shown that high pressure gradient in the contact zone where separation occurs can be difficult to model. Slight changes in element size inside the contact zone alter the results. In most cases, equal size linear elements are found to give the best results. Although higher order elements are better suited for computation of displacements than the linear elements, higher order elements may not be a proper choice for approximating tractions inside the contact zone. As noted in Chapter 6, the way tractions are approximated within a higher order element may be of concern when friction is present. In the case of frictional problems the arrangement of stick-slip zones is affected by the node locations and a relatively large number of elements may be required to produce satisfactory results.

The most important, but often neglected, aspect of contact problem is the load-path dependence. Okamoto and Nakazawa (1979) and Torstenfelt (1984) present an automatic load incrementation technique which computes the load increment size such that contact status of only one node pair is changed in each load increment. These techniques are incorporated in the algorithm to ensure the stability of the numerical procedure. As discussed in Chapter 7, the contact status of each node has to be checked for three different possibilities. If the contact status of only one node pair is changed the reliability of the algorithm can be guaranteed with regard to the uniqueness of the contact status of the node pair under consideration. However, these load increments may not be small enough to guarantee the convergence of solutions. A converged solution in this context is taken as a solution obtained after a certain number of load increments which does not change with further increase in the number of load increments.

Consider the Hertz's problem for the case of full adhesion. The results of this problem are presented in Chapter 7, in which the load is applied in 200 equal size load increments. The case of full adhesion is considered in order to eliminate an additional unknown factor, namely the formation of slip zone. Fig. 36 shows the frictional forces when the load is applied in 25, 50, and 200 equal size load increments. Clearly, the results for 25 and 50 increments are unacceptable. The frictional forces obtained by these load increments oscillate about the curve obtained by 200 increments and the convergence is found to be slow. Fig. 37 shows the results for 100 equal size load increments. One hundred fifty equal size load increments (not shown here) give results which are within 1 percent of the results obtained by 200 increments. Hence, the results obtained by 200 increments are considered as an accurate solution (a converged solution) within the limitations of numerical analysis. A point to be noted is that in all these cases the normal tractions are not affected by the different load increment sizes.

The oscillations produced by large load increment are found to have only a local effect. Fig. 38 shows the frictional forces in which the first 16 percent of the load is applied in 4 equal load increments, and then the remaining 84 percent of the load is applied in 168 equal load increments. The oscillations produced due to large load increment do not pollute the rest of the solution, in which small size load increment is used. This is because in the case of an adhesive contact, the contact zone acts as a fixed boundary. As the successive node pairs come into

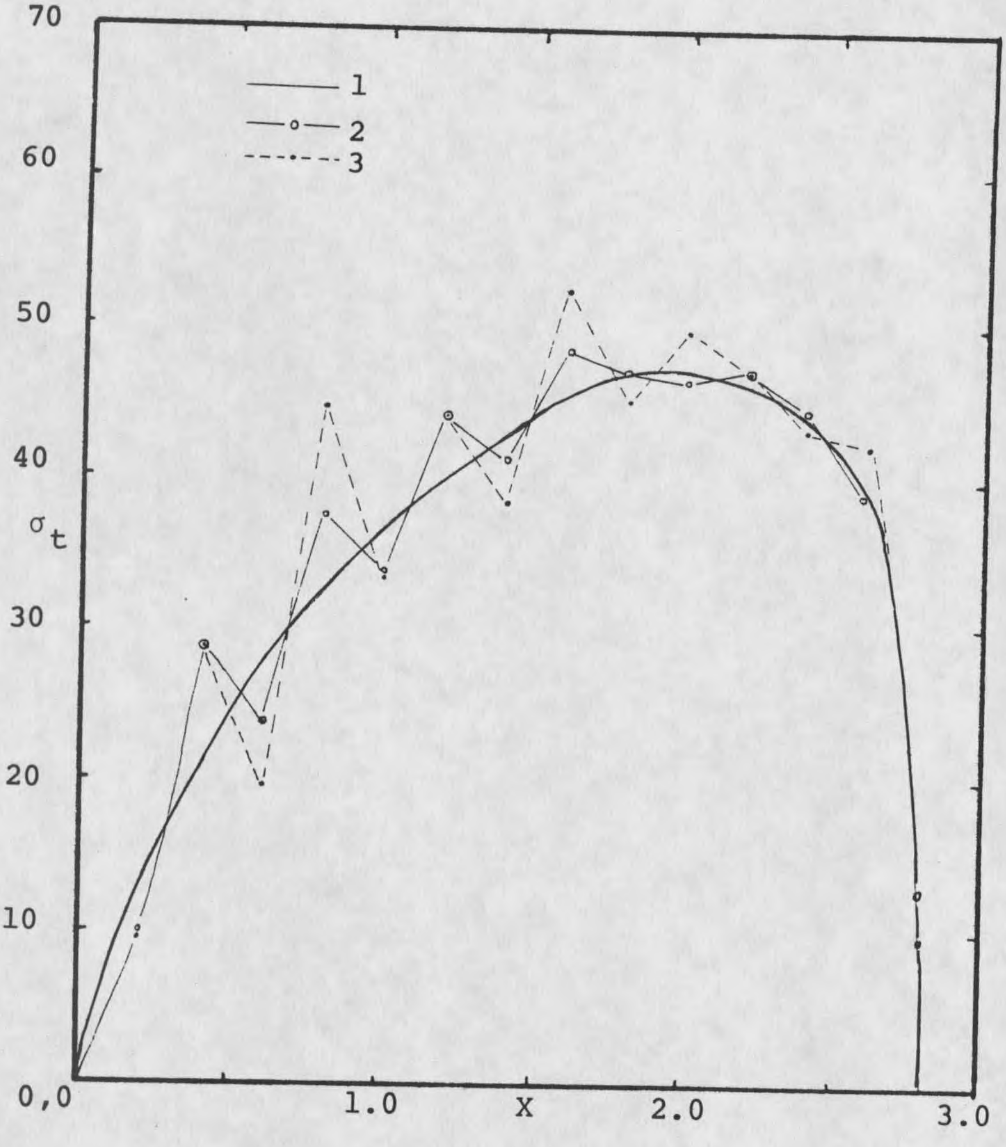


Figure 36. Frictional forces inside the contact zone for the case of full adhesion.
1. 25 equal size load increments.
2. 50 equal size load increments.
3. 200 equal size load increments.

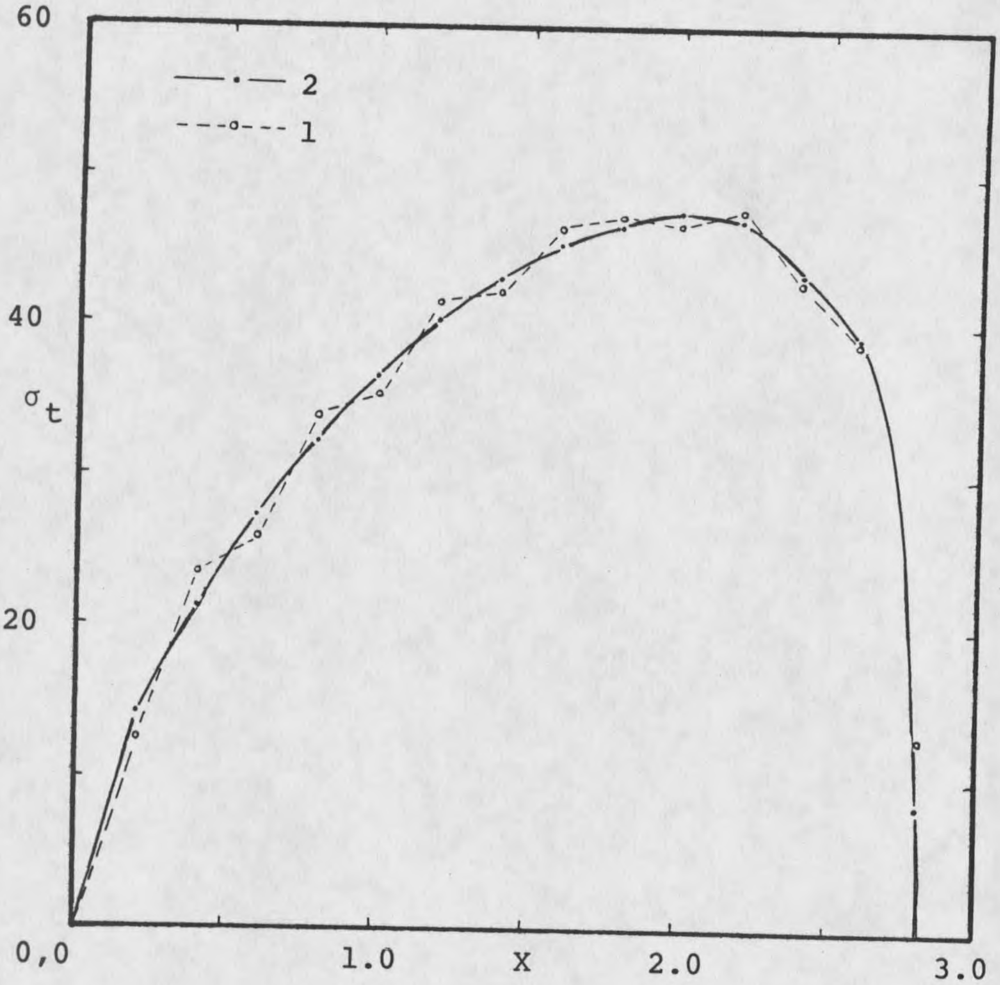


Figure 37. Frictional forces inside the contact zone for, (1) 100 and (2) 200 equal size load increments.

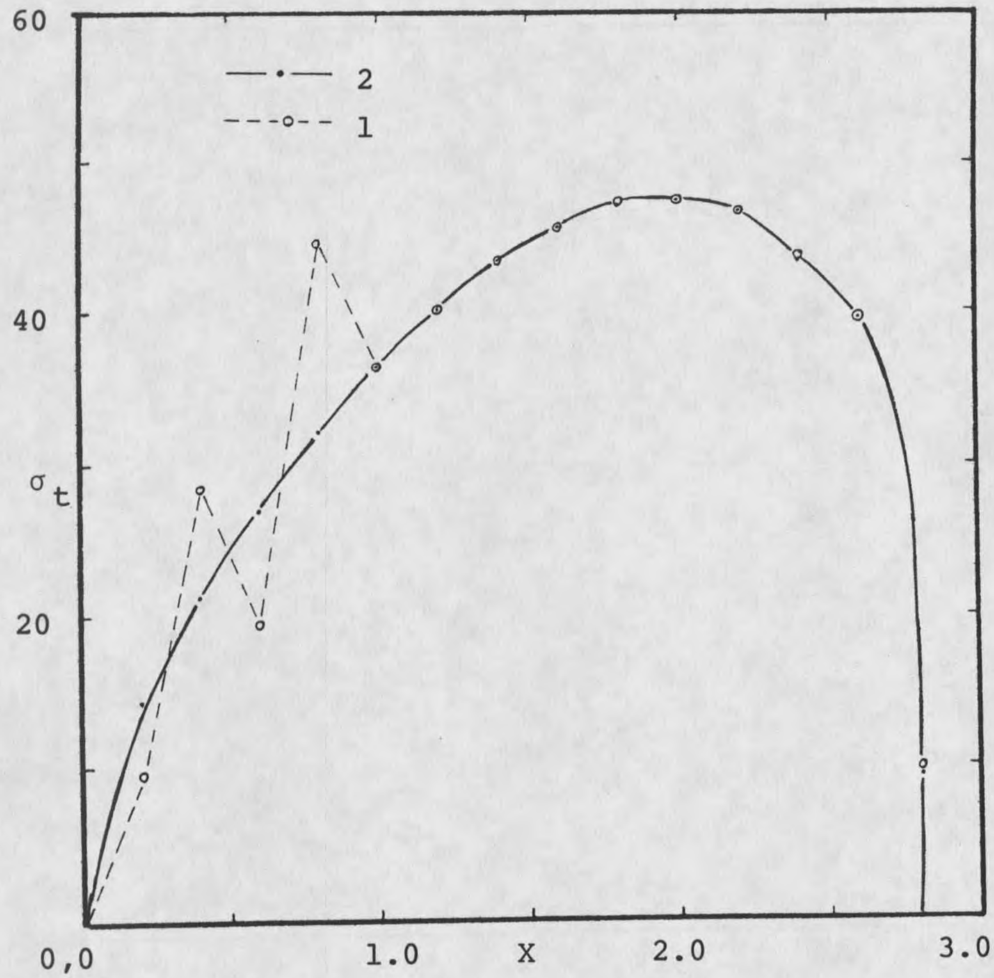


Figure 38. Effect of load increments on the frictional forces.
1. 16 percent of the load is applied in 4 equal increments, and then 84 percent of the load is applied in 168 equal increments.
2. 200 equal size load increments.

contact only the frictional forces in the neighborhood of these node pairs are affected and not all the nodes in the contact zone. If a node just barely comes in contact then the normal and tangential tractions at that node are nearly zero. Goodman (1962) shows that at the point where the contact ceases, the ratio of tangential to normal stress increases without bound. Thus, at a node which just barely comes in contact, the tangential tractions in its vicinity have a very high gradient. In numerical analysis, the nodes are separated by a certain distance and contact can occur only at these nodes. Thus, contact progresses stepwise, not continuously, as the load is increased. With sufficiently small load increments one can insure that when a node pair comes in contact it does so just barely, thereby approximating continuous progression. If a load increment is such that the actual contact boundary falls between two nodes, the load which is in excess to the amount of load necessary to bring a node pair just barely in contact has to be distributed in the neighborhood of this node pair. Hence, the distribution of the traction can be erroneous and as observed in Fig. 36, the results can be unacceptable.

Thus, in order to accurately model the continuous progression of the contact zone, it is necessary to have the load increment sizes such that successive node pairs just barely come in contact. Fig. 39 shows the frictional forces for load increments of unequal size, but chosen such that

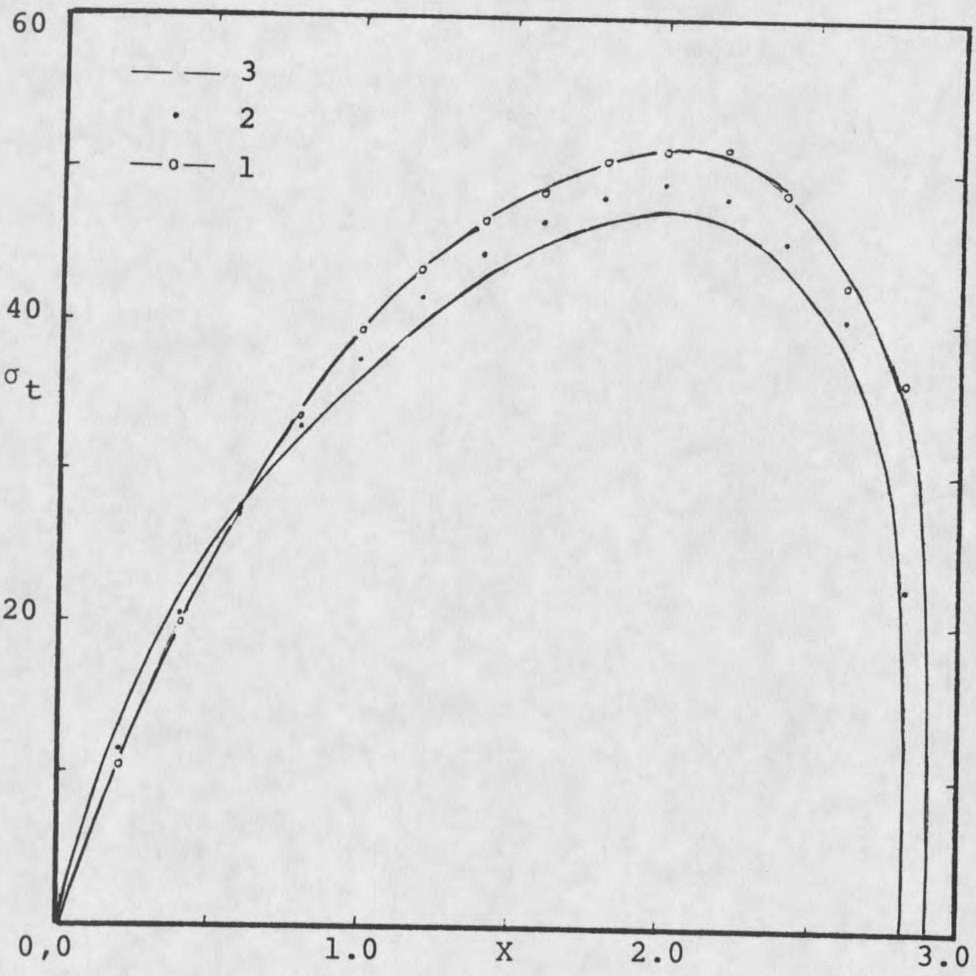


Figure 39. Effect of load increments, which bring only one node in contact, on the frictional forces.

1. Size of load increments is such that only one node just barely comes in contact in each load increment.
2. Size of load increments is such that only one node barely comes in contact every two load increments.
3. 200 equal size load increments.

each load increment brings only one node pair into contact, furthermore, the node pair just barely comes in contact. The results show a sharp improvement when compared to the results obtained by large equal size load increments. Although the curve is smooth, these results still show about 10 percent higher traction values compared to the results obtained by 200 load increments which are considered to be acceptable results. When each of the unequal load increments is further reduced by half the results show about 5 percent higher values. The solution converges to the solution obtained by 200 load increments when each of the unequal load increments is divided into four equal size load increments.

Thus, the problem needs to be solved by applying the load in a large number of small load increments or by computing the load increment sizes such that each node pair just barely comes in contact and then dividing these load increments into smaller ones.

Fig. 40 shows the effect of change in element size on the frictional forces. In all the three cases the load is applied in 200 equal size load increments. For curve 1, a large jump in element size (2.5:1) occurs at $x = 2.0$, which causes oscillations in the vicinity of $x \doteq 2.0$. The results are about 6 percent in error, even when the load increment size is considered to be small (200 increments). A gradual change in element size (0.25-0.20-0.15-0.10) gives results

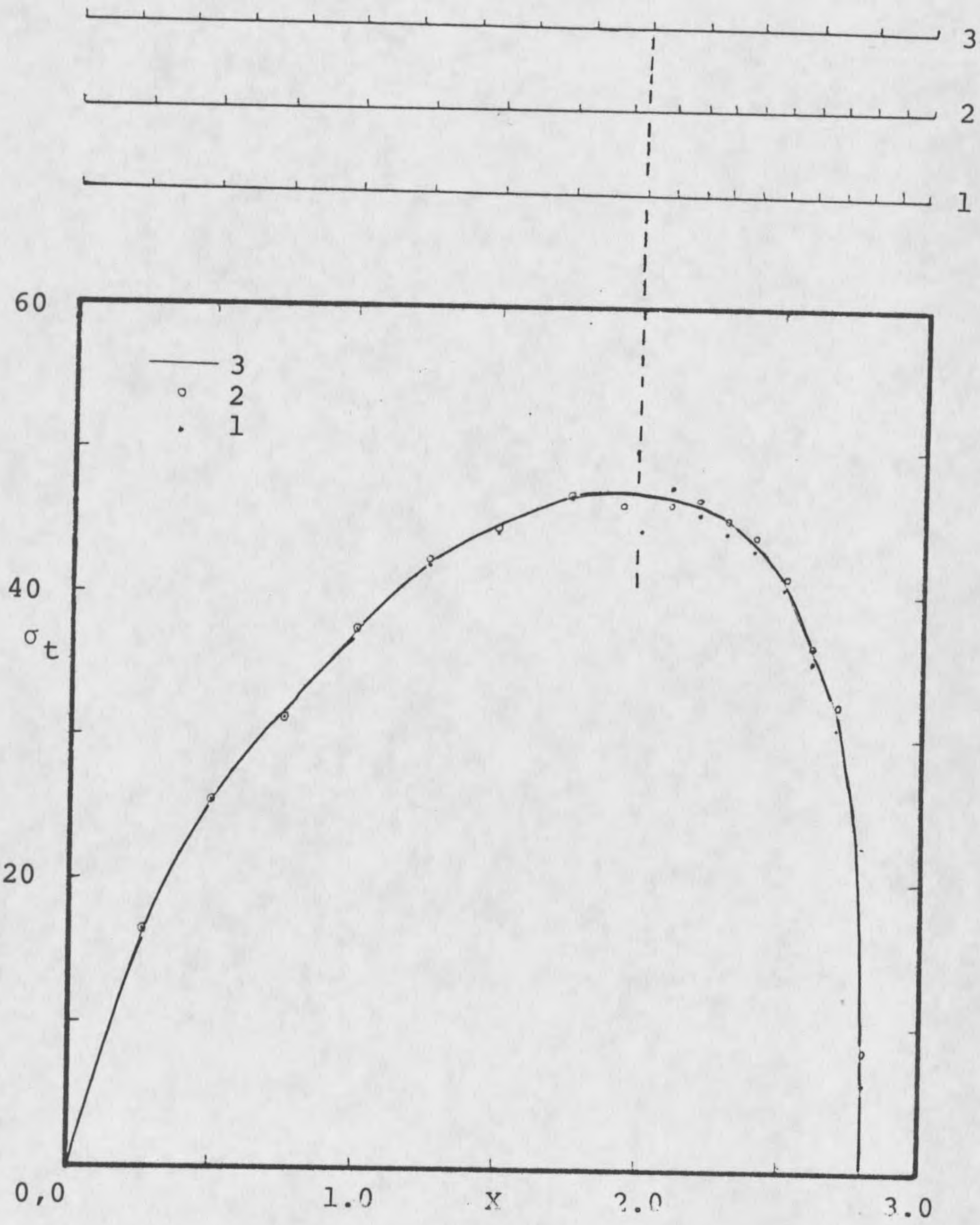


Figure 40. Effect of mesh pattern on the frictional forces. All 3 solutions are obtained by applying the load in 200 equal size load increments.

within 2 percent as shown by curve 2. Equal size elements do not produce any oscillations as can be seen from curve 3.

To this point, Hertz's problem for the two limiting cases, frictionless and full adhesion, has been discussed. In the case of full adhesion, since the ratio of tangential to normal traction increases without bound, for small values of some slip at the edge of contact zone occurs. The extent of the slip zone is usually a strong function of material constants and μ. Spence (1973) shows the effect of material constants and μ on the extent of stick and slip zones for the case of a flat rigid indentor. This is also demonstrated in Chapter 8 (Fig. 25) in connection with the compression of an elastic rectangle. In general, the load-path dependence of the solution will be a function of material constants and μ. If partial slip at the edge of contact zone takes place, the frictional forces are relaxed at the edge as each successive node pair comes in contact, and number of load steps required to converge to a solution is greatly reduced. Fig. 41 shows the frictional forces for μ equal to 0.4 and ∞. While for $\mu = \infty$, 200 load increments are needed for the solution to converge, for $\mu = 0.4$, a converged solution is obtained with 20 equal size load increments. Results for load increment sizes that bring only one node pair into contact at a time are similar to the results obtained by applying the load in 20 load increments.

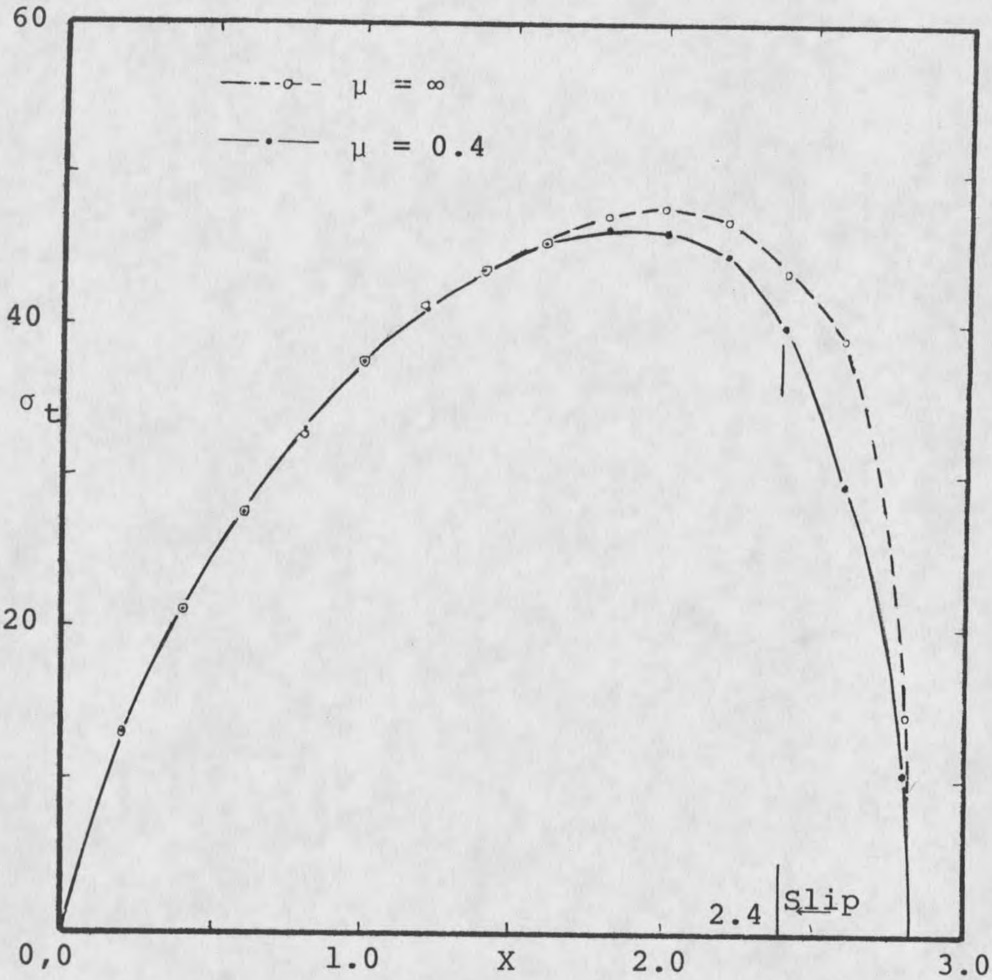


Figure 41. Variation of frictional forces for $\mu = 0.4$ and ∞.
1. For $\mu = 0.4$, the load is applied in 20 equal size load increments.
2. For $\mu = \infty$, the load is applied in 200 equal size load increments.

Since the load-path dependence of the solution is a function of material constants and μ, it may be necessary and also economical to develop an algorithm for computing the load increments so as to just barely change the contact status of only one node per load increment with an option to further divide each load step into number of smaller load steps.

CHAPTER 10

SUMMARY AND DISCUSSION

Numerical analysis of nonlinear problems has some inherent difficulties. Analysis of large problems is expensive and time consuming, so very few numerical experiments can be conducted. These generally consist of convergence tests based on solutions to a given problem using different mesh patterns; sensitivity of solutions to changes in material constants and boundary conditions; effect of changes in geometry; and comparison of selected problems with known solutions and other numerical techniques, when available. Due to complexity of contact problems these experiments are time consuming and can be rarely conducted effectively.

Nevertheless, several contact problems have been solved and numerical experiments have been conducted in as much detail as possible. To the author's best knowledge, these types of numerical experiments for contact problems have not been previously reported in the literature.

Since very few analytical solutions to contact problems with friction are available, various contact problems should be solved using both the FEM and the BIE method. These can provide a basis for comparison among various solution methods, and may suggest certain guidelines for future development of efficient numerical techniques. Problems with non-proportional and cyclic loading have received very little attention in the literature because validity of the solutions obtained by a numerical technique cannot be judged. Certain guidelines can be prepared for these types of problems if few of these problems are solved by the FEM and the BIE method.

Some suggestions can be made in using the BIE method to solve contact problems. Since in the BIE method both tractions and displacements are retained as unknowns, the equilibrium equations and the contact equations are used explicitly. While the contact equations are history dependent, the equilibrium equations (7.8 and 7.9) are not. These equations are therefore never needed in the incremental formulation, and thus can be incorporated into the system of equations implicitly rather than explicitly, as is done in this dissertation. If the equilibrium equations are written implicitly, additional savings in computational time can be made. Furthermore, the contact equations, which are written for each node pair, have only four non-zero entries (two components of tractions or displacements per node). These equations can be written such that they appear in a separate group within the fourth block row (Fig. 20), and can then be stored more efficiently.

The equation solver described here takes advantage of most of the unique features of the contact problems. The most important aspect of the equation solver is that the history dependent equations are separated from the equations which are not. Thus, incremental-iterative technique is then applied on only part of the system of equations after partial forward elimination. This feature can be incorporated in any of the equation solvers that are used to solve history dependent problems.

Since the number of degrees of freedom for constant and linear elements is same for a given mesh pattern, constant elements do not offer any advantages. Furthermore, linear elements are found to be reliable and best suitable for contact problems. As noted in Chapter 6, higher order elements present a problem when the contact ceases at an interior node. This can be of concern when friction is considered. Application of the load in increments which bring only one node in contact will not be suitable for higher order elements (see Fig. 11). The Hertz's problem discussed in Chapter 9 should be solved using higher order elements to check their effect on load steps and consequently the frictional forces.

The conspicuous features in the numerical analysis of contact problems are the load path dependence of contact problems; the development of a decision making algorithm which guarantees the uniqueness of the contact status of a node pair; and since the boundary conditions inside the contact zone change, sensitivity of solutions to location of nodes. These can be of concern in the development of numerical techniques to solve three-dimensional contact problems. Since the contact zone in this case is a surface, the boundaries of stick, slip and contact zones will strongly depend on the type of elements (triangular or quadrilateral), and even with linear elements the results may show inconsistency. Along the contact surface, slip can occur in one direction and not in the other direction thus, further complicating the algorithm. The computation of load increments which just barely change the contact status of only one node pair will be complicated. For example, consider a quadrilateral element of which only one node is in contact, the distribution of normal and tangential stresses over that element depends on the distance by which the nodes are separated from the node in contact and the type functional of approximation used. Thus, in the development of numerical techniques for three-dimensional contact problems, the investigator should be aware of different types of elements and the various possibilities that may arise along the contact surface using these elements.

REFERENCES CITED

REFERENCES CITED

Andersson, T., (1981), "The Boundary Element Method Applied to Two-Dimensional Contact Problems", Proceedings of the Third International Seminar on Recent Advances in Boundary Element Methods, C. A. Brebbia (ed.), Springer Verlag, Berlin.

Andersson, T.,(1982), "The Second Generation Boundary Element Contact Program", Boundary Element Methods in Engineering, Proceedings of the Fourth International Seminar, C. A. Brebbia (ed.), Springer Verlag, New York.

Andersson, T., Persson, B. G. A.,(1983),"The Boundary Element Applied to Two Dimensional Contact Problems", Chapter 5 in Progress in Boundary Element Methods, C.A. Brebbia (ed.), Pentech Press, London.

Baker, C.T.H., (1977), " The Numerical Treatment of Integral Equations", Oxford University Press, Oxford, Great Britan.

Bathe, K. J., Chaudhary, A., (1985), "A Solution Method For Planar And Axisymmetric Contact Problems", International Journal for Numerical Methods in Engineering, Vol. 21, 65-88.

Bettess, J. A., (1983), "Economical Solution Technique for Boundary Integral Matrices", International Journal for Numerical Methods in Engineering, Vol. 19, 1073-1077.

Bolteus, L., Tullberg, O., (1982), "BEMSTAT - A New Type of Boundary Element Program for Two-Dimensional Elasticity Problems", Boundary Element Methods in Engineering, Proceedings of the Fourth International Seminar, C.A. Brebbia (ed.), Springer Verlag, New York.

Brebbia, C. A., (1978), "The Boundary Element Method for Engineers", Halstead Press, New York.

Brebbia, C. A. (ed.), (1981), "Progress in Boundary Element Methods", Vol. I, Halstead Press, New York.

Brebbia, C. A., Telles, J. C. F., Wrobel, L. C., (1984), "Boundary Element Techniques", Springer Verlag, Berlin.

Campos, L. T., Oden, J. T., Kikuchi, N., (1982), " A Numerical Analysis of a Class of Contact Problems With Friction In Elastostatics", Computer Methods in App. Mech. and Engg. , Vol. 34, 821-845.

Chan, S. K., Tuba, I.S., (1971), "A Finite Element Method for Contact Problems of Solid Bodies - Part I. Theory and Validation", Int. J. Mech. Sci., Vol. 13, 615-625.

Cheng, J.H., Kikuchi, N., (1985), " An Incremental Constitutive Relation of Unilateral Contact Friction for Large Deformation Analysis", Journal of Applied Mechanics, Vol. 52, 639-648.

Chiu, P. K., Prasad, S. N., (1976), " Indentation of an Elastic Rectangle by a Pair of Rough Rigid Punches with Finite Friction", Int. J. Solids Structures, Vol. 12, 619-634.

Civelek, M. B., Erdogan, F., (1975), " The Frictionless Contact Problem for an Elastic Layer Under Gravity", J. App. Mech., Trans. ASME, Vol. 97, 136-140.

Civelek, M. B., Erdogan, F., (1976), " Interface Separation in a Frictionless Contact Problem for an Elastic Layer", J. App. Mech., Trans. ASME (Brief Notes), Vol. 98, 175-177.

Cruse, T. A., (1969), " Numerical Solutions in Three Dimensional Elastostatics", Int. J. Solids Structures, Vol. 5, 1259-1274.

Danson, J. A., (1981), "A Boundary Element Formulation of Problems in Linear Isotropic Elasticity with Body Forces" Proceedings of the Third International Seminar on Recent Advances in Boundary Element Methods, C.A. Brebbia (ed.), Irvine, California.

Danson, J. A., (1983), "Linear Isotropic Elasticity with Body Forces" in Progress in Boundary Element Method, Vol. II, C.A. Brebbia (ed.), Pentech Press, London.

Dasgupta, S., Prasad, S. N., (1976), " On The Phenomenon Of Separation During Compression And Frictional Sliding Of An Elastic Rectangle", J. App. Mech., Vol. 43, 268-274.

Delves, L.M., Mohomed, J.L., (1985), " Computational Methods for Integral Equations", Cambridge University Press, Cambridge.

Delves, L.M., Walsh J., (1974), " Numerical Solutions of Integral Equations", Clarendon Press, Oxford, Great Britan.

De Pater, A.D., Kalker, J.J.,(eds.), (1975)," The Mechanics of the Contact Between Deformable Bodies", Proceedings of the Symposium of The International Union of Theoretical and Applied Mechanics, Deft University Press, Enchede, Netherlands.

Dundurs, J. (1975), " Properties of Elastic Bodies in Contact", in The Mechanics of the Contact Between Deformable Bodies, De Pater A.D. and Kalker J.J. (eds.), Deft University Press, Netherlands.

Dundurs, J., Stippes, M., (1970), " Role of Elastic Constants in Certain Contact Problems", J. App. Mech., Trans. ASME, Vol. 37, 965-970.

Francavilla, A., Zieniewicz, O. C., (1975), " A Note on Numerical Computation of Elastic Contact Problems", Int. J. Num. Meth. Engg., Vol. 19, 913-924.

Fung, Y.C., (1965), " Foundations of Solid Mechanics", Prentice Hall Inc., Englewood Cliffs, New Jersey.

Gaertner, R., (1977), " Investigation of Plane Elastic Contact Allowing for Friction", Computers and Structures, Vol. 7, 59-63.

Gecit, M.R., (1980), " A Tensionless Contact Without Friction Between Elastic Layer and an Elastic Foundation", Int. J. Solids and Structures, Vol. 16, 387-396.

Gladwell, G.M.L., (1980), " Contact Problems in the Classical Theory of Elasticity", Sijthoff and Noordoff, Netherlands.

Goodman, L.E., (1962), "Contact Stress Analysis of Normally Loaded Rough Spheres", Journal of App. Mech., Vol. 29, 515-522.

Hertz, H., see A.E.H. Love's book, pp. 193-208

Hood, P., (1976), " Frontal Solution Program for Unsymmetric Matrices", Int. J. Num. Meth. Engg., Vol. 10, 379-399.

Jawson, M.A., Ponter, A.R., (1963), " An Integral Equation Solution of the Torsion Problem", Proc. Roy. Soc. (london), Ser. A, Vol. 273, 237-246.

Jawson, M.A., (1963), " Integral Equation Methods in Potential Theory", Proc. Roy. Soc. (london), Ser. A, Vol. 275, 23-32.

Jawson, M.A., Symm, G.T., (1977), " Integral Equation Methods in Potential Theory and Elastostatics", Academic Press Inc., London.

Katona, M.G., (1983), " A Simple Contact-Friction Interface Element with Applications to Buried Culverts", Int. J. Num. Meth. Engg., Vol. 7, 371-384.

Keer, L. M., Dundurs, J., Kiattikomol, K., (1973), " Separation of a Smooth Circular Inclusion from a Matrix", Int. J. Engng. Sci., Vol. 11, 1221-1233.

Keer, L. M., Dundurs, J., Tsai, K. C., (1972), " Problems Involving a Receding Contact Between a Layer and a Half Space", J. App. Mech., Vol. 39, 1115-1120.

Lachat, J. C., (1975), " A Further Development of The Boundary Integral Technique for Elastostatics", Ph.D. Thesis, University of Southhampton, England.

Lachat, J.C., Watson, J.O., (1975), " A Second Generation Boundary Integral Equation Program for Three Dimensional Elastic Analysis", in Boundary Integral Equation Method: Computational Applications in Applied MEchanics, T.A. Cruse and F.J. Rizzo (eds.), ASME, New York.

Lachat, J.C., Watson, J.O., (1976), " Effective Numerical Treatment of Boundary Integral Equations: A Formulation for Three-Dimensional Elastostatics", Int. J. Num. Meth. Engng, Vol. 10, 991-1005.

Love, A.E.H., (1929)," A Treatise on Mathematical Theory of Elasticity", (4th edition), Dover Publications, New York.

Meyer, C., (1973), " Solution of Linear Equations - State of the Art", Proceedings of the American Society of Civil Engineers, Journal of the Structural Division, Vol. 99, No. ST7, 1507-1527.

Oden, J. T., Pires, E. B., (1983), " Nonlocal and Nonlinear Friction Laws and Variational Principles for Contact Problems in Elasticity", J. App. Mech., Vol. 50, 67-76.

Ohte, S., (1973), " Finite Element Analysis of Elastic Contact Problems", Bull. of JSME, Vol. 16, No. 95, 797-804.

Okamoto, N., Nakazawa, M., (1979), " Finite Element Incremental Contact Analysis with Various Frictional Conditions", Int. J. Num. Meth., Vol. 14, 337-357.

Paul, B., Hashemi, J., (1980), " Contact Pressures on Closely Conforming Elastic Bodies", Solid Contact and Lubrication, H. S. Cheng and L. M. Keer (eds.), ASME, New York.

Paul, B., Hashemi, J., (1981), " Contact Pressures on Closely Conforming Elastic Bodies", J. App. Mech., Vol. 48, 543-548.

Pires, E. B., Oden, J. T., (1983), " Analysis of Contact Problems with Friction Under Oscillating Loads", Computer Meth. App. Mech. Engng., Vol. 39, 337-362.

Prasad, S. N., Chiu, P. K., Dasgupta, S., (1976), " Compression and Sliding of an Elastic Rectangle Fixed Rigidly at the Base", Int. J. Engng. Sci., Vol. 14, 617-629.

Prasad, S. N., Dasgupta S., (1975), " Effect of Sliding Friction on Contact Stresses in a Rectangle Compressed by Rigid Planes", J. App. Mech., Vol. 42, 656-662.

Ratwani, M., Erdogan, F., (1973), " On the Plane Contact Problem for a Frictionless Elastic Layer", Int. J. Solids Structures, Vol. 9, 921-936.

Rizzo, F. J., (1967), " An Integral Equation Approach to Boundary Value Problems of Classical Elastostatics", Q. Appl. Math., Vol. 25, 83-95.

Sachdeva, T. D., Ramakrishnan, C. V., Natarajan, R., (1981), " A Finite Element Method for the Elastic Contact Problems", J. App. Mech., Trans. ASME, Vol. 103, 456-461.

Sachdeva, T. D., Ramakrishnan, C. V., (1981), "A Finite Element Solution for the Two-Dimensional Elastic Contact Problems with Friction", Int. J. Num. Meth. Engng., Vol. 17, 1257-1271.

Saeedvafa, M., Dundurs, J. (1987), " An Example for Load-Path Dependence in Elasticity with Friction" To Be Published.

Sarkar, A.D., (1976), " Wear of Metals", Pergamon Press, Great Britan.

Schafer, H., (1975), " A Contribution to the Solution of Contact Problems with the Aid of Bond Elements", Computer Meth. App. Mech. Engng., Vol. 6, 335-354.

Singh, K. P., Paul, B., (1974), " Numerical Solution of Non-Hertzian Elastic Contact Problems", J. App. Mech., Vol. 41, 484-490.

Sokolnikoff, I.S.,(1956), " Mathematical Theory of Elasticity",(2nd edition), Mcgraw Hill Book Company, New York.

Solecki, J.S., Swedlow, J.L., (1980), " Elastic Stress Analysis of Constrained Cylinders by a Special Finite Element Method", Int. J. Solids Struct., Vol. 16, 959-968.

Spence, D. A., (1968), " Self Similar Solutions to Adhesive Contact Problems with Incremental Loading", Proc. Roy. Soc. Ser. A., Vol. 305, 55-80.

Spence, D. A., (1973), " An Eigenvalue Problem for Elastic Contact with Finite Friction", Proc. Camb. Phil. Soc., Vol. 73, 249-268.

Spence, D. A., (1975), " The Hertz Contact Problem With Finite Friction", J. Elasticity, Vol. 5, Nos. 3-4, 297-319.

Spence, D. A., (1986), " Frictional Contact with Transverse Shear", Q. J. Mech. Appl. Math., Vol. 2, Pt. 2, 233-253.

Stąbrowski, M. M., (1981), " An Algorithm for the Solution of Very Large Banded Unsymmetric Linear Equation Systems", Int. J. Num. Meth. Engng., Vol. 17, 1103-1117.

Stadter, J. T., Weiss, R. O., (1979), " Analysis of Contact Through Finite Element Gaps", Computers and Structures, Vol. 10, 867-873.

Symm, G.T., (1963), " Integral Equation Methods in Potential Theory. II", Proc. Roy. Soc. (London), Vol. 275, 33-46.

Timoshenko, S.P., Goodier, J.N., (1970), " Theory of Elasticity" (3rd edition), Mcgraw Hill Book Company, New York.

Torstenfelt, B. R., (1983), " Contact Problems with Friction in General Purpose Finite Element Computer Programs", Computers and Structures, Vol. 16, NO. 1-4, 487-493 .

Torstenfelt, B. R., (1984), " An Automatic Incrementation Technique for Contact Problems with Friction", Computers and Structures, Vol. 19, No. 3, 393-400.

Tsai, K. C., Dundurs, J., Keer, L. M., (1974), " Elastic Layer Pressed Against a Half Space", J. Appl. Mech., Vol. 41, 703-707.

Tseng, J., Olson, M. D., (1981), " The Mixed Finite Element Method Applied to Two-Dimensional Elastic Contact Problems", Int. J. Num. Meth. Engng., Vol. 17, 991-1014.

Tsuta, T., Yamaji, S., (1973), " Finite Element Analysis of Contact Problems", Theory and Practice in Finite Element Structural Analysis, University of Tokyo Press.

Turner, J. R., (1979), " The Frictional Unloading Problem on a Linear Elastic Half Space", J. Inst. Maths., Applics., Vol. 24, 439-469.

Valliappan, S., Lee, I. K., Boonluolohr, P., (1984), " Non-Linear Analysis of Contact Problems", Numerical Methods in Coupled Systems, R. W. Lewis, P. Bettess, E. Hinton (eds.), John Wiley and Sons Ltd., New York.

Wilson, E. A., Parsons, B., (1970), " Finite Element Analysis of Elastic Contact Problems Using Differential Displacements", Int. J. Num. Meth. Engng., Vol. 2, 387-395.

APPENDICES

APPENDIX A

NUMERICAL INTEGRATION

The terms ΔT_{IJ} and ΔU_{IJ} in Eq. (4.15) are numerically integrated. These terms are

$$\Delta T_{IJ}(m,n) = \int_{\Gamma k} T_{IJ}(P^m, Q^n{}_{kl}(\eta))\, M^l(\eta)\, J(\eta)\, d\eta \,, \qquad (A.1)$$

$$\Delta U_{IJ}(m,n) = \int_{\Gamma k} U_{IJ}(P^m, Q^n{}_{kl}(\eta))\, M^l(\eta)\, J(\eta)\, d\eta \,. \qquad (A.2)$$

For a source point P on the boundary Γ, these terms are evaluated numerically over each boundary element Γ_k. In performing the integration, it is necessary to consider separately the case in which the source point P is located in the element over which integration is being performed and that in which it is not. When the source point is located outside the element to be integrated, the integrand varies smoothly and the Gaussian quadrature formula with weight function equal to 1.0 may be used (Stroud and Secrest, 1966).

$$\int_{-1}^{1} f(\xi)d\xi \approx \sum_{n=1}^{N} f(\xi_n)\, w_n \,. \qquad (A.3)$$

where ξ_n is the coordinate of the n^{th} integration point, w_n is the associated weighting factor, and N is the total number of integration points. This formula integrates exactly any polynomial of degree $\leqslant 2N-1$. However the integrands consist not only of polynomials but also functions like $1/r$ and $\ln(r)$. These functions vary rapidly as the source point gets close to the element and then the Gaussian quadrature is no longer exact. Based on empirical study, Bolteus, et. al.(1982) investigate the optimal number of integration points necessary. They use Gaussian quadrature with 2,4,6,and 10 points to study the accuracy of integration as the distance between source point and element decreases. Their guidelines regarding the order of integration are based on the ratio r/L, where r is the distance between source point and element, and L the element length. These guidelines should prove useful in improving the performance of the numerical integration. Similar guidelines are adapted in this dissertation, they are

1. Use 2 g.p. for $r/L > 5$.
2. Use 4 g.p. for $1.5 < r/L \leqslant 5$.
3. Use 10 g.p. for $r/L \leqslant 1.5$.

When the source point is within the element, the integration has to be computed using special integration technique since the integrand contains $1/r$ and $\ln(r)$ (Eq. 4.2 and 4.3). To integrate ΔU_{IJ} the Gaussian quadrature with weight function $\ln(1/\xi)$ is used.

$$\int_0^1 f(\xi)\ \ln(1/\xi)\ d\xi \approx \sum_{n=1}^{N} f(\xi_n)\ w_n . \tag{A.4}$$

In applying the formula, it is necessary to carry out a linear transformation of the natural coordinates to obtain the interval (0,1). When the source point P is inside the element, the integration has to be done over two different intervals.

The term ΔT_{IJ} contains 1/r term, for which there exists no suitable integration scheme. The singular integral and the free term C_{IJ} (Eq. 4.15) need not be evaluated explicitly. These coefficients can be computed using the fact that stress field corresponding to a rigid body translation is zero. For this case the equation is

$$C_{IJ}(P^m) + \sum_{k=1}^{K} \sum_{l=1}^{L} \delta_{mn} \Delta T_{IJ}(P^m, Q^n_{kl})$$

$$= -\sum_{k=1}^{K} \sum_{l=1}^{L} (1-\delta_{mn}) \Delta T_{IJ}(P^m, Q^n_{kl}) . \qquad \text{(A.5)}$$

where

$$\delta_{mn} = 1 \qquad m = n$$
$$\quad = 0 \qquad m \neq n$$

Thus, the left hand side of the Eq. (A.5) is computed as the sum of all the coefficients ΔT_{IJ} for which $m \neq n$.

APPENDIX B

PROGRAM DOCUMENTATION INCLUDING THE SAMPLE DATA FILE

Documentation for Program Contact

A. HEADING CARDS:

NUMBER OF HEADING CARDS (I5)

columns	variable	entry
1 - 5	J	Total number of information cards to be followed.

HEADING CARD (18A4)

columns	variable	entry
1 - 80	TITLE	Enter the information to be printed out with the results. The last card is retained as a title card.

B. CONTROL CARDS:

PROBLEM TYPE CARD (2I5)

columns	variable	entry	
1 - 5	IPTYPE	EQ.0	Plane strain
		EQ.1	Plane stress
6 - 10	IOUT	LT.1	Data check only
		EQ.0	Two-dimensional stress analysis using BIE.
		EQ.1	Contact problem without back-substitution.
		EQ.2	Contact problem with back-substitution.
		GE.3	Output contains answers after each iteration.

NUMBER OF BODIES AND CONTACT PAIRS (3I5)

columns	variable	entry
1 - 5	NB	Total number of bodies (not more than two).

6 - 10	MCP	Total number of contact zones. Maximum 3 disconnected contact zones are allowed.
11 - 15	NFNODE	Number of functional nodes per element. EQ.1 Constant element. EQ.2 Linear element. EQ.3 Quadratic element. EQ.4 Cubic element.

INFORMATION FOR EACH BODY (4I5)

One card per body in increasing numerical order.

columns	variable	entry
1 - 5	NREN(IB)	Total number of regions for body IB.
6 - 10	NBN(IB)	Number of boundary nodes per element. These are used for geometrical representation. Each body may have separate order of approximation. EQ.2 Linear. EQ.3 Quadratic. EQ.4 Cubic.
11 - 15	NEL(IB)	Total number of elements.
16 - 20	NBP(IB)	Total number of boundary points. NBP(IB) = NEL(IB)*(NBN(IB) -1).

Note: If contact is brought about by specifying only tractions on one of the bodies, that body should be numbered 2.

ELEMENTS PER REGION (2I5)

One card per region of each body.

columns	variable	entry
1 - 5	REGN(IB,IR,1)	First element corresponding to the region in consideration.
6 - 10	REGN(IB,IR,2)	Last element corresponding to the region in consideration.

C. MATERIAL PROPERTIES:

ELASTIC CONSTANTS (2F10.0)

One card per body.

columns	variable	entry
1 - 10	YM(IB)	Modulus of elasticity.
11 - 20	RNU(IB)	Poisson's ratio.

COEFFICIENT OF FRICTION (I5,1X,7F10.0)

columns	variable	entry
1 - 5	NMU	Number of different coefficients of friction. Maximum number 7.
7 - 76	FMU(IMU)	Coefficients of friction. For frictionless problem input 0.

D. NODAL COORDINATES:

For each body, first input the global coordinates of the origin. Local coordinates of nodes are read in ascending order. Both cylindrical and cartesian coordinates can be used. Cylindrical coordinates can be used with origin as the center. Coordinates of missing node numbers are generated as equally spaced nodes along a straight line or an arc. Node numbers for each body can be given separately. These node numbers should then start from 1 and must be numbered in sequence. Maximum number of cards = NBN(IB), IB = 1, NB

GLOBAL COORDINATES OF THE ORIGIN (2F10.0)

columns	variable	entry
1 - 10	ORIGIN(IB,1)	X - coordinate of the origin.
11 - 20	ORIGIN(IB,2)	Y - coordinate of the origin.

NODAL COORDINATES (A1,I4,1X,2F10.0)

columns	variable	entry
1	CT	Symbol describing coordinate system for this node. EQ. ; (blank) cartesian (X,Y) EQ.P; cylindrical (R,)
2 - 5	N	Node number
7 - 16	X	X - coordinate or radius R.
17 - 26	Y	Y - coordinate or angle in degrees.

D. BOUNDARY CONDITIONS:

Boundary conditions and point load information for each body are read separately in a group. For each functional node two displacements and two tractions, all in the global coordinates, are the quantities under consideration. Boundary conditions are read for each element separately. Boundary conditions for contact element need not be read, also by default the boundary conditions for elements outside the contact zone are assumed to be zero traction and unknown displacements. Traction discontinuity can easily be handled as boundary conditions for each element are read separately. Boundary conditions for elements in sequence can also be generated.

BOUNDARY CONDITION CARDS (I5)

columns	variable	entry
1 - 5	NBC(IB)	Total number of boundary condition cards.

BOUNDARY CONDITION (I1,I4,I5,1X,4I1,1X,4F10.0)

columns	variable	entry
1	I1	EQ.0 or blank; BC's are not to be generated. EQ.1; BC's are to be generated (see: note).
2 - 5	ICODE(IE,1)	Element number.
6 - 10	ICODE(IE,2)	Local functional node number.
11	ICODE(IE,3)	BC code for X - traction.
12	ICODE(IE,4)	BC code for Y - traction.
13	ICODE(IE,5)	BC code for X - displacement.
14	ICODE(IE,6)	BC code for Y - displacement. EQ.0 known traction or displacement. EQ.1 unknown traction or displacement (the default is 0011).
16 - 25	BCOND(IE,1)	Value of the X - traction.
26 - 35	BCOND(IE,2)	Value of the Y - traction.
36 - 45	BCOND(IE,3)	Value of the X - displacement.
46 - 55	BCOND(IE,4)	Value of the Y - displacement.

Note: To generate boundary conditions following procedure is adapted. Boundary conditions for the last functional node of the starting element must be read. Next card must contain the last element in sequence and the boundary conditions for the local node number which has the same boundary conditions.

E. POINT LOAD CARDS:

NUMBER OF POINT LOADS (I5)

columns	variable	entry
1 - 5	NPLOAD(IB)	Total number of point loads. Maximum of 15 point load cards.

POINT LOAD INFORMATION (I5,1X,4F10.0)

columns	variable	entry
1 - 5	K	Point load number.
7 - 16	PTLOAD(I,1)	X - coordinate of the load point.
17 - 26	PTLOAD(I,2)	Y - coordinate of the load point. The load point must not coincide a functional node.
27 - 36	PTLOAD(I,3)	Value of X - component.
37 - 46	PTLOAD(I,4)	Value of Y - component.

Note: The boundary conditions and point load information for the second body is read after the point load cards of the first body.

F. CONTACT INFORMATION:

NUMBER OF CONTACT ELEMENTS (4I5)

columns	variable	entry
1 - 5	IB1	Body number in ascending order.
6 - 10	IB2	Corresponding body in contact.
11 - 15	MCF(IP,3)	Number of functional nodes in contact for the pair under consideration.
16 - 20	MCF(IP,4)	Number of elements in contact for the pair under consideration.

ELEMENT NUMBERS (4I5)

columns	variable	entry
1 - 5	MCEL(IP,IB1,1)	First element in contact for the body with lower number.
6 - 10	MCEL(IP,IB1,2)	Last element in contact for the body with lower number.
11 - 15	MCEL(IP,IB2,1)	Element in contact with the first element of the first body.
16 - 20	MCEL(IP,IB2,2)	Element in contact with the last element of the first body.

NODE NUMBERS (4I5)

Same procedure is repeated for global functional node numbers. Each body may have separate global node numbers.

Note: These three sets are repeated for each contact pair.

NUMBER OF NODES AND ELEMENTS (2I5)

One card per body.

columns	variable	entry
1 - 5	MCPB(IB,1)	Total number of nodes in contact.
6 - 10	MCPB(IB,2)	Total number of elements in contact.

G. <u>LOAD PATH INFORMATION:</u>

NUMBER OF LOAD PATHS (2I5)

columns	variable	entry
1 - 5	NPATH	Number of load paths. Each load path can be divided into equal number of load increments.
6 - 10	ITMAX	Maximum number of iterations for each load increment.

LOADING INFORMATION (I5,1X,F10.0)

columns	variable	entry
1 - 5	PATH(I,1)	Number of equal size load increments to be given between the previous load and the current load point. Starting load is assumed to be zero.
7 - 16	PATH(I,2)	Final load at the end of the load path. If all the load is applied at once, the load is assumed to be unit load.

Note: Maximum 30 proportional load paths are allowed. Cyclic loading is possible.

SAMPLE DATA FILE

```
00014
This is a sample data file. In between comment statements are provided.
The first card decides no. of cards provided for the problem statement.
In this case we have 14 cards.
------------------------------------------------------------------
The sample problem is Hertz's problem.
Refer to chapter 9 for the results. Coeff. of friction = 0.4 & 0.3
Body 1 is the foundation. Body 2 is the cylinder with radius 8.
Linear geometry for both bodies. Linear elements (functional approx.)
Contact zone contains 14 equal size elements of .2 cm length.
The load step sizes: 1 node comes in contact per 2 load increments.
Therefore we have 14 load steps. which are divided into two each.
P = 965. CONTACT LENGTH = 2.9 FOR FRICTIONLESS PROBLEM
Next card is used as title card for the output.
HERTZ'S PROBLEM. F= 0.4 & 0.3, EQ. SIZE ELEM = .2  LOAD STEPS = 14*2
C - CONTROL INFORMATION
    0   02
    2    1    2
    1    2   46   46
    1    2   58   58
    1   46
    1   58
C - MATERIAL PARAMETERS AND FRICTION COEFF.
10000000.0  000.3000
1000.00000  000.3000
00002 0000.4000 0000.3000 0000.0000
C - NODAL COORDINATES FOR THE FOUNDATION. WITH 0,0 AS ORIGIN. 46 NODES
C - NODAL NOS. ARE GIVEN SEPARATELY FOR EACH BODY (NOT NECESSARY).
00000.000 00000.0000
    1   00.00    -04.00
```

```
     9  08.00    -04.00
    13  08.00    000.00
    16  05.00    000.00
    18   4.00    000.00
    23   3.00    000.00
    28   2.00    000.00
    33   1.00    000.00
    38   0.00    000.00
    40   0.00   -000.50
    41  00.00   -001.00
    46  00.00   -003.50
C - NODAL COORDINATES FOR THE CYLINDER. WITH 0,8 AS ORIGIN.
C - P FOR POLAR COORDINATES. 58 NODES
   0.0      +08.0000
P    1 0008.0000 0000.0000
P    8   08.0      70.0
P   10   08.00     80.000
P   11 0008.0000 82.8192442
P   12 0008.0000 84.6206210
P   13 0008.0000 86.4166783
P   14 0008.0000 88.2092153
    15   0.00      08.00
    17   0.00      07.50
    18   0.0       07.00
    25   0.0       00.00
    32   0.0      -07.00
    33   0.0      -07.50
    35   0.0      -08.00
P   36 0008.0000 271.432544
P   37 0008.0000 272.865984
P   38 0008.0000 274.301222
P   39 0008.0000 275.739171
P   40 0008.0000 277.180756
P   41 0008.0000 278.626927
P   42 0008.0000 280.078658
P   43 0008.0000 281.536959
P   44 0008.0000 283.002878
P   45 0008.0000 284.477512
P   46 0008.0000 285.962014
P   47 0008.0000 287.457603
P   48 0008.0000 288.965575
P   49 0008.0000 290.487315
P   50 0008.0000 292.024313
P   51 0008.0000 293.578179
P   52 0008.000  295.000000
P   53 0008.000  300.000000
P   58 0008.000  350.000000
C - 13 BOUNDARY CONDITION CARDS FOR THE FOUNDN. 0: KNOWN 1: UNKNOWN
    13
    01      1 0100
    01      2 0110
1   08      1 0110
```

```
   08     2 0100
   09     1 1000
   09     2 1001
1  12     2 1001
   13     1 0001
   37     2 0101
   38     1 1001
   38     2 1001
1  46     1 1001
   46     2 1000
C - 0 CARDS FOR POINT LOADS FOR THE FOUNDATION.
   00
C - 8 B.C. CARDS FOR CYLINDER. 0011 : KNOWN TRACTION & UNKNOWN DISP
C - 1820 COMPRESSIVE LOAD ON ELEM 14 IN THE Y DIRN.
00008
00013     1 0011 00000.000
   13     2 0011
   14     1 0011           -1820.0
   14     2 0001           -1820.0
   15     1 1001
   15     2 1001
1  34     2 1001
   35     1 0101
    0
C - CONTACT INFORMATION. 14 ELEMENTS, 15 NODES. FOLLOWED BY GLOBAL NO.
    1     2    15    14
   24    37    48    35
   24    38    49    35
   15    14
   15    14
C - 14 LOAD STEPS 10 ITERATION. EACH LOAD STEP IS FURTHER DIVIDED
C - IN 2 LOAD STEPS. AT THE END OF FIRST 2 LOAD STEP .05P IS THE LOAD
   14    10
  002 00.00500
   02 00.0200
   02 00.045
   02 00.0800
   02 00.127
   02 00.185
   02 00.255
   02 00.330
   02 00.420
   02 00.515
   02 00.635
   02 00.755
   02 00.890
   02 01.03
```

APPENDIX C

COMPUTER PROGRAM

Figure 42. Listing of the program contact.

```
      COMMON/ALL/ BCOND(180,4), BNODE(220,2), DELTA(2,2),
     1            DJACB(10), EDISP(4,2,2), EDSING(2,2), ELBCOD(4,2),
     2            ELRHS(2), ESTRS(4,2,2), GPCOD(10,2), GPSHAP(10,4)
      COMMON/ALL1/ORIGIN(3,2),POISON(3),POLE(2),POLE2(2),POLEBC(4),
     4            POLE2BC(4), POSGP(16,2), SICOD(4), TITLE(18),
     5            UNORM(10,2), UNT1(2,2), UNT2(2,2), UNTAVG(2,2),
     6            WEIGP(16,2), YM(3), FMU(5)
      INTEGER     CSTD(4,5), REGN(3,3,3,3)
      COMMON/INT/ CSTD, ICODE(180,6), IPCODE(4),REGN,MCEL(3,3,3,3),
     1            MCF(3,4), MCPB(3,2),NBC(3),NBN(3),NBP(3),NEL(3),
     2            NFP(3),NODEF(4), NREN(3)
      COMMON/REF/ ISLIP(100), ITENSE(100), MEPAIR(2,100), MPAIR(2,100)
      COMMON/REAL/ PI, RAD, RMAX, RNU, SMU, YMOD
      COMMON/INTEGER/ IOUT, IPTYPE, ISING, ISOLN, ITMAX, MCP, MCPOIN,
     1                MEQUN, MSIZE, MTAPE1, MTAPE2, MTCEL, NB, NBNODE,
     2              NBPOIN,NDIME, NELEM, NEQUN, NFNODE, NFPOIN, NGAUS,
     3                NNSING, NONCP, NPOLE, NREGN, NSIZE, NTAPE, NTBC,
     4              NTEQUN, NTSIZE, MXSIZE, NSTD, NSLIP, NTENSE, LSTEP
     5               ,MTCP, NMU
      COMMON/EVOLV/SHAPE(4),DERIV(4)
      COMMON/POINT/ NPLOAD(2), PTLOAD(15,4)
      COMMON/REF1/ NPATH, PATH(0:30,2), TLOAD
C-----------------------------------------------------------------------
C                    ****   PROGRAM CONTACT    ****
C                           ---------------
C
C      PROGRAM FOR TWO DIMENSIONAL FRICTIONAL CONTACT PROBLEMS
C                 USING BOUNDARY INTEGRAL EQUATION.
C                    COLLOCATION METHOD IS USED.
C
C     THE PROGRAM IS CAPABLE OF HANDLING TWO BODIES IN CONTACT
C                 WITH MAXIMUM THREE SEPARATE CONTACT ZONES.
C
C         BOUNDARY APRROXIMATION : LINEAR, QUADRATIC, CUBIC.
C     FUNCTIONAL APPROXIMATION : CONSTANT, LINEAR,QUADRATIC,CUBIC.
C         ORDER OF GAUSSIAN QUADRATURE : INTERNALLY DECIDED.
C
C             THREE TAPES ARE USED MTAPE1 MTAPE2 AND NTAPE.
C         MOST OF THE VARIABLES STARTING WITH N ARE FOR NONCONTACT
C                     OR ARE GENERAL VARIABLES.
C         VARIABLES STARTING WITH M ARE FOR CONTACT INFORMATION.
C
C     ONLY MONOTONICALLY INCREASING PROPORTIONAL LOADING AND UNLOADING
C               IS POSSIBLE.   NO UNPROPORTIANAL LOADING.
C
C     THE PROGRAM DOES NOT COMPUTE THE LOAD INCREMENTS.
C    SMALL STEP SIZE SHOULD BE GIVEN TO ENSURE UNIQUE RESULTS.
C    OR COMPUTE LOAD STEPS SUCH THAT ONLY ONE NODE CHANGES ITS STATUS.
C
C     DO NOT CHANGE THE ELEMENT SIZE IN THE CONTACT ZONE DRASTICALLY.
```

```
C
C     PROGRAM DEVELOPED BY BHUSHAN W. DANDEKAR
C
C
C     JANUARY 26, 1987
C-----------------------------------------------------------------------
C
       INCLUDE COMMON, LIST
C-----------------------------------------------------------------------
C     FOLLOWING STATEMENTS ARE KEPT IN A FILE CALLED "COMMON"
C     THESE ARE INCLUDED IN MANY SUBROUTINES USING "INCLUDE" STATEMENT
C
C-----
C     COMMON/ALL/ BCOND(180,4), BNODE(220,2), DELTA(2,2),
C    1            DJACB(10), EDISP(4,2,2), EDSING(2,2), ELBCOD(4,2),
C    2            ELRHS(2), ESTRS(4,2,2), GPCOD(10,2), GPSHAP(10,4)
C     COMMON/ALL1/ORIGIN(3,2),POISON(3),POLE(2),POLE2(2),POLEBC(4),
C    4            POLE2BC(4), POSGP(16,2), SICOD(4), TITLE(18),
C    5            UNORM(10,2), UNT1(2,2), UNT2(2,2), UNTAVG(2,2),
C    6            WEIGP(16,2), YM(3), FMU(5)
C     INTEGER     CSTD(4,5), REGN(3,3,3,3)
C     COMMON/INT/ CSTD, ICODE(180,6), IPCODE(4),REGN,MCEL(3,3,3,3),
C    1            MCF(3,4), MCPB(3,2),NBC(3),NBN(3),NBP(3),NEL(3),
C    2            NFP(3),NODEF(4), NREN(3)
C     COMMON/REF/ ISLIP(100), ITENSE(100), MEPAIR(2,100), MPAIR(2,100)
C     COMMON/REAL/ PI, RAD, RMAX, RNU, SMU, YMOD
C     COMMON/INTEGER/ IOUT, IPTYPE, ISING, ISOLN, ITMAX, MCP, MCPOIN,
C    1                MEQUN, MSIZE, MTAPE1, MTAPE2, MTCEL, NB, NBNODE,
C    2              NBPOIN,NDIME, NELEM, NEQUN, NFNODE, NFPOIN, NGAUS,
C    3                NNSING, NONCP, NPOLE, NREGN, NSIZE, NTAPE, NTBC,
C    4              NTEQUN, NTSIZE, MXSIZE, NSTD, NSLIP, NTENSE, LSTEP
C    5               ,MTCP, NMU
C     COMMON/EVOLV/SHAPE(4),DERIV(4)
C     COMMON/POINT/ NPLOAD(2), PTLOAD(15,4)
C     COMMON/REF1/ NPATH, PATH(0:30,2), TLOAD
C
C-----------------------------------------------DECLARE CONSTANTS
      MTAPE1 = 1 ; MTAPE2 = 2
      NTAPE =  3
      PI    = 2.*ASIN(1.0)
      NDIME = 2
      REWIND MTAPE1
      REWIND MTAPE2
      REWIND NTAPE
C-----------------------------------------------SET THE DIMENSIONS
      DIMENSION CRHS(320), CX(320), RHS(200), UNION(2,520)
      DIMENSION  A(102400)
      MXSIZE = 102400
      MSIZE =  320
      NSIZE = 200
      NTSIZE = MSIZE + NSIZE
C-----------------------------------------------------READ THE DATA
```

```
      CALL   BIGBANG
      IF(IOUT.LT.0) STOP
C-----------------------------------------------------INITIALIZATION
      DELTA(1,1) = DELTA(2,2) = 1.0
      DELTA(1,2) = DELTA(2,1) = 0.0
      RMAX = SMU = 0.0
      NSTD = 0
      NSLIP = NTENSE = 0
C
      DO 100 I = 1, NTEQUN
      UNION(1,I) = UNION(2,I) = 0.0
      IF(I.LE.NEQUN) RHS(I) = 0.0
      IF(I.GT.MEQUN) GO TO 100
      CRHS(I) = 0.0
      CX(I) = 0.0
 100   CONTINUE
C-----------------------------------GAUSS QUADRATURE CONSTANTS AND
C--------------------------------LOCAL COORDINATES FOR FUNCTION NODES
      CALL   GAUSSQ (NFNODE, POSGP, WEIGP, SICOD)
C
C--------------------------------------EQUN ASSEMBLY, DECOMPOSITION
C--------------------------------------DISK STORAGE
C--------------------------------------FOR EACH BODY AND EACH ZONE
      DO  120  IB  = 1, NB
      ICFLAG = 0
      MCPOIN = MCPB(IB,1)
      MCOL  = (NFP(IB) -MCPOIN)*2 + 4*MCPOIN
      MROW  = (NFP(IB) -MCPOIN)*2
      CALL ASSEMBLE (A, CRHS,  RHS, UNION, MROW,MCOL, ICFLAG,IB)
 120  CONTINUE
      IF(IOUT.EQ.0) GO TO 160
C-------------------------------------------------------CONTACT ZONE
C     DO  140  IP  = 1, MCP
      DO  140   J  = 1, 2
      ICFLAG = 1 ; IP = 1
      IB = MCF(IP,J)
      MCPOIN = MCPB(IB,1)
      MROW  = 4*MCPOIN
      MCOL  = 8*MCPOIN + (NFP(IB) -MCPOIN)*2
      CALL ASSEMBLE (A, CRHS,  RHS, UNION, MROW,MCOL, ICFLAG,IB)
 140  CONTINUE
C-----------------------------------IOUT.EQ.0 => 2-D STRESS ANALYS
 160  CONTINUE
      IF(IOUT.EQ.0)THEN
      CALL EXTORT(A, UNION, RHS, NFP(1)*2, NFP(1)*2 , 2 , 2)
      CALL NEWS (CX, RHS)
      GO TO  300
      END IF
C--------------------------------FIND THE CONTACT ZONE FOR EACH SMU
      DO  200  IMU = 1, NMU
      SMU = FMU(IMU)
      MROW = MCOL = 8*MCPOIN
```

```
      CALL REFINE (A, CRHS, CX, UNION,MROW, MCOL)
C
      IF(IOUT.GE.2.AND.IMU.EQ.NMU) THEN
C---------------------------------------------------PRINT THE RESULTS
      M1 = (NFP(1) -MCPOIN)*2
      M2 = (NFP(2) -MCPOIN)*2
      MC = 4*MCPOIN
      DO 180 I = 1, M1+M2
      RHS(I) = RHS(I)*TLOAD
 180  CONTINUE
      CALL    BACKSUB (A,CX,RHS,M1,M2,MC,NTAPE)
      END IF
C
      CALL   NEWS (CX, RHS)
 200  CONTINUE
C
 300  CONTINUE
      STOP
      END

      SUBROUTINE BIGBANG
C     ---------- -------
C
C------ THIS SUBROUTINE READS THE DATA--------------------------------
C
      INCLUDE COMMON, LIST
      DATA PLR/1HP/
      DOUBLE PRECISION DUM, RD
      DO 90 I = 1,3
      NFP(I) = NEL(I) = NBP(I) = NREN(I) = 0
      NBC(I) = NBN(I) = 0
 90   CONTINUE
C----- READ THE TITLE AND BASIC PARAMETERS. J IS NO. OF TITLE CARDS.
C                                    LAST CARD IS SAVED AS TITLE CARD.
      READ (105, 1020) J
      DO  95  I = 1, J
      READ (105, 1000) TITLE
      WRITE (108, 2010) TITLE
 95   CONTINUE
C-------------------------------------------------------------------
C 1>  IOUT.LT.0 : DATA CHECK. EQ.0 : 2-D STRESS ANALYSIS
C     IOUT.EQ.1 : CONTACT PROB. GE.2 : BACKSUBSTITUTION
C     IOUT.GE.3 : INTERMEDIATE OUTPUT (CHECK SUBROUTINE REFINE)
C 2>  IPTYPE.EQ.0 : PLANE STRAIN. EQ.1 : PLANE STRESS
C 3> GEOMETRY MODEL CAN BE SEPARATE FOR EACH BODY. NOT THE FUNCTIONS.
C-------------------------------------------------------------------
      READ (105,1020) IPTYPE, IOUT
      READ (105,1020) NB, MCP, NFNODE
       DO  100  IB = 1, NB
      READ (105,1020) NREN(IB), NBN(IB), NEL(IB), NBP(IB)
      NFP(IB) = NEL(IB) *(NFNODE-1)
      IF(NFNODE.EQ.1) NFP(IB) = NEL(IB)
```

```
 100     CONTINUE
      WRITE (108,2020) IOUT,IPTYPE
      WRITE (108,2030) NB,MCP,NFNODE
C
       DO  120  I = 1, NB
       DO  120  J = 1, 3
       DO  120  K = 1, 3
       DO  120  L = 1, 3
       MCEL (I,L,J,K) = 0
       REGN (I,L,J,K) = 0
 120     CONTINUE
C
C----- READ THE FIRST AND LAST ELEMENT OF EACH REGION ALSO MAT PROP.--
C
      NOLD = 0
      N1 = NE1 = 0
       DO  140  IB = 1, NB
      IF(IB.GT.1) THEN
      N1 = N1 + NBP(IB-1)
      NE1 = NE1 + NEL(IB-1)
      END IF
       DO  140  IREN = 1, NREN(IB)
       READ (105, 1020) REGN(IB,IREN,1,1), REGN(IB,IREN,1,2)
      IF(NOLD.GT.REGN(IB,IREN,1,1)) THEN
      REGN(IB,IREN,1,1) = REGN(IB,IREN,1,1) + NOLD
      REGN(IB,IREN,1,2) = REGN(IB,IREN,1,2) + NOLD
      END IF
      NOLD = REGN(IB,IREN,1,2)
       IF(NFNODE.EQ.1) THEN
       ITEMP  =  NFNODE
      ELSE
      ITEMP = NFNODE - 1
      END IF
      REGN(IB,IREN,2,1) = (REGN(IB,IREN,1,1) -1)*ITEMP + 1
      REGN(IB,IREN,2,2) = (REGN(IB,IREN,1,2) )*ITEMP
      REGN(IB,IREN,3,1)=(REGN(IB,IREN,1,1)-NE1-1)*(NBN(IB)-1)+1+N1
      REGN(IB,IREN,3,2) = (REGN(IB,IREN,1,2)-NE1)*(NBN(IB)-1) + N1
 140     CONTINUE
C
C----------------------------------------- READ : MATERIAL PROPERTIES
C
       DO  150  IB = 1, NB
      READ (105,1040) YM(IB), POISON(IB)
      WRITE (108,2040) IB, NEL(IB),NBN(IB),NBP(IB), NFP(IB),NREN(IB),
     1 YM(IB), POISON(IB)
 150   CONTINUE
      READ (105, 1090) NMU, (FMU(I), I= 1, NMU)
      WRITE (108, 2047) NMU, (FMU(I), I = 1, NMU)
C
C-------------------------------- READ OR GENERATE THE NODAL COORDINATES
C  1> FOR EACH BODY, FIRST READ THE GLOBAL COODINATES OF THE ORIGIN.
C  2> COORDINATES CAN BE INPUT AS LOCAL COORDINATES.
```

```
C  3> NODES MUST BE NUMBERED IN SEQUNCE ALONG THE BOUNDARY.
C  4> NODES ARE READ IN ASCENDING ORDER. MISSING NODES : GENERATED
C  5> EQUALLY SPACED NODES ALONG A ST. LINE OR AN ARC ARE GENARATED
C  6> FOR POLAR COORDINATES INPUT RADIUS AND ANGLE IN DEGREES.
C     FIRST COLUMN SHOULD HAVE " P ".
C  7> THE FIRST AND THE LAST NODE SHOULD NOT BE A CONTACT NODE.
C     PROGRAM MAY HANDLE IT. BUT I AM NOT SURE.
C----------------------------------------------------------------------
      WRITE(108,2000) TITLE
      RD = DATAN(1.0D0)/45.0D0
      NOLD = NUMNP = 0
      YOLD = 0.0
      DO  190  IB  = 1, NB
      NUMNP = NBP(IB) + NUMNP
      NOLD  = NUMNP - NBP(IB)
      NST   = NOLD
      READ (105, 1040) ORIGIN(IB,1), ORIGIN(IB,2)
C
 160  CONTINUE
      READ (105, 1060) CT, N, X, Y
      IF(N.GE.NOLD) THEN
      NODE = N
      ELSE
      NODE = N + NST
      END IF
      NUM = NODE - NOLD
      NUMN = NUM - 1
      XNUM = NUM
      IF(CT.EQ.PLR) GO TO 170
C
      BNODE (NODE, 1) = X
      BNODE (NODE, 2) = Y
      IF(NUMN.LT.1) GO TO 180
C
      DX = (X - BNODE(NOLD, 1))/XNUM
      DY = (Y - BNODE(NOLD, 2))/XNUM
      DO  165  J = 1, NUMN
      K = NOLD + J
      BNODE (K, 1) = BNODE (K-1, 1) + DX
      BNODE (K, 2) = BNODE (K-1, 2) + DY
 165  CONTINUE
      GO TO 180
C
 170  CONTINUE
      DUM = Y*RD
      BNODE (NODE, 1) = X* DCOS(DUM)
      BNODE (NODE, 2) = X* DSIN(DUM)
      IF(NUMN.LT.1) GO TO 180
      DY = (Y - YOLD)/XNUM
      DO  175  J = 1, NUMN
      DUM = J
      DUM = (YOLD + DUM*DY)*RD
```

```
      BNODE (NOLD+J, 1) = X* DCOS(DUM)
      BNODE (NOLD+J, 2) = X* DSIN(DUM)
 175  CONTINUE
C
 180  CONTINUE
      YOLD = Y
      NOLD = NODE
      IF(NODE.LT.NUMNP)GO TO 160
 190  CONTINUE
C
      NST = 0
      DO  200  IB = 1, NB
      DO  195  I  = 1, NBP(IB)
      N = I + NST
      BNODE (N, 1) = BNODE (N, 1) + ORIGIN(IB,1)
      BNODE (N, 2) = BNODE (N, 2) + ORIGIN(IB,2)
 195  CONTINUE
      NST = NST + NBP(IB)
 200  CONTINUE
C
      WRITE(108, 3000)
      N = 0
 205  N = N + 1
      IF(N.GT.NBP(1)) THEN
      X1 = Y1 = 0.0
      ELSE
      X1 = BNODE(N, 1)
      Y1 = BNODE(N, 2)
      END IF
      IF(N.GT.NBP(2)) THEN
      X2 = Y2 = 0.0
      ELSE
      X2 = BNODE(N+NBP(1), 1)
      Y2 = BNODE(N+NBP(1), 2)
      END IF
      IF(N.LE.NBP(1).AND.N.LE.NBP(2)) THEN
      WRITE (108, 3020) N,X1,Y1, N, X2, Y2
      ELSE
      IF(N.LE.NBP(1))THEN
      WRITE (108,3020) N,X1,Y1
      ELSE
      WRITE (108,3025) N,X2,Y2
      END IF
      END IF
      IF(N.LT.NBP(1).OR.N.LT.NBP(2)) GO TO 205
C
C---------------------------------------------- READ : BOUNDARY CONDITIONS
C  1> FIRST READ THE NUMBER OF BOUNDARY CONDITIONS PER BODY.
C  2> THE DEFAULT FOR BC IS NO TRACTION OR POTENTIAL CONTACT NODE.
C  3> ELEMENT NUMBER AND THE LOCAL FUNCTIONAL NODE NO. ARE NEEDED.
C  4> THE BC IS READ FOR EACH ELEMENT. NOTE THE TRACTION DISCONTINUITY
C  5> BC FOR ELEMENTS IN SEQUNCE CAN BE GENERATED.
```

```
C  6> 0 = KNOWN ; 1 = UNKNOWN. EXAMPLE:0011 TRACTIONS KNOWN. DISP NOT
C-----------------------------------------------------------------------
      WRITE (108, 2000) TITLE
      WRITE (108, 2045)
      NOLD = NOLD1 = 0
       DO  235  IB = 1, NB
       NOLD = NOLD + NEL(IB)
      READ (105,1020) NBC(IB)
      IF(IB.EQ.1) IE = 0
       DO  230  I  = 1, NBC(IB)
      IE = IE + 1
      READ (105,1080) I1, ICODE(IE,1), ICODE(IE,2), ICODE(IE,3),
     1                ICODE(IE,4), ICODE(IE,5), ICODE(IE,6),
     2                BCOND(IE,1), BCOND(IE,2),
     3                BCOND(IE,3), BCOND(IE,4)
       IF((ICODE(IE,1)+NOLD1).GT.NOLD) GO TO 210
       ICODE(IE,1) = ICODE(IE,1) + NOLD1
 210   IF(I1.EQ.0) GO TO 230
       I2 = ICODE(IE,1) - ICODE(IE-1,1)
       K1 = ICODE(IE,2)
       IE = IE-1
       DO  225  J = 1, I2
       J1 = ICODE(IE,1) + 1
       DO  225  K = 1, NFNODE
       IF(J.EQ.I2.AND.K.GT.K1) GO TO 225
       IE = IE +1
       IF(J.EQ.1.AND.K.EQ.1) GO TO 220
       DO  215  L = 1, 4
       ICODE(IE,L+2) = ICODE(IE-1,L+2)
       BCOND(IE,L) = BCOND(IE-1,L)
 215   CONTINUE
 220   ICODE(IE,1) = J1
       ICODE(IE,2) = K
 225   CONTINUE
 230   CONTINUE
C---------------- READ POINT LOADS (NO. AND X AND Y COORD, TX AND TY)
C      POINT LOAD CANNOT BE APPLIED AT A NODE. 15 PT LOADS PER BODY.
       N = 0
       READ (105, 1020) NPLOAD(IB)
       IF(NPLOAD(IB).EQ.0)  GO TO 232
       IF(IB.EQ.2) N = NPLOAD(1)
       DO  231  I = 1, NPLOAD(IB)
       READ (105, 1090) K, (PTLOAD(I+N, J) , J = 1, 4)
       PTLOAD(I+N,1) = PTLOAD(I+N,1) +ORIGIN(IB,1)
       PTLOAD(I+N,2) = PTLOAD(I+N,2) +ORIGIN(IB,2)
 231   CONTINUE
 232   CONTINUE
C-----------------------------------------------------------------
       NOLD1 = NOLD1 + NEL(IB)
       NBC(IB) = IE
       IF(IB.EQ.2) THEN
       NBC(IB) = NBC(IB) - NBC(1)
```

```
      I2 = NBC(1)
      ELSE
      I2 = 1
      END IF
      DO  233  I = I2, IE
      WRITE (108, 2050)((ICODE(I,J),J=1,6),(BCOND(I,K),K=1,4))
 233  CONTINUE
      DO  234  I = 1, NPLOAD(IB)
      WRITE (108, 2055)(PTLOAD(I+N,J),J=1,4)
 234  CONTINUE
 235  CONTINUE
C------------------------- READ : CONTACT INFORMATION FOR EACH PAIR
C     TWO BODY PROBLEM. THREE INDEPENDNT CONTACT PAIRS.
C  1> FOR A PAIR : BODIES IN CONTACT, NO. OF NODES, NO. OF ELEMENTS.
C  2> FIRST AND LAST ELEMENT FOR BODY WITH LOWER NUMBER AND ELEMENTS
C     IN CONTACT.
C  3> DO THE SAME FOR FUNCTIONAL NODES. (FIRST AND LAST NODE.)
C  4> READ : NO. OF CONTACT NODES AND ELEMENTS FOR EACH BODY.
C  5> NO. OF LOAD INCREMENTS AND MAXM NO. OF ITERATIONS ALLOWED.
C-------------------------------------------------------------------
       DO  248  IP = 1, MCP
       N1 = N2 = 0
       NE1 = NE2 =0
C----------------------------------------1> BODIES IN CONTACT & ---
      READ (105,1020) MCF(IP,1), MCF(IP,2), MCF(IP,3), MCF(IP,4)
      IB1 = MCF(IP,1)
      IB2 = MCF(IP,2)
      DO  240  I = 1, IB1-1
      N1 = NFP(I) + N1
      NE1 = NEL(I) + NE1
 240  CONTINUE
      DO  245  I = 1, IB2-1
      N2 = NFP(I) + N2
 245  NE2 = NEL(I) + NE2
      NOLD = NEL(IB1) + NEL(IB2)
C----------------------------------------2> FIRST AND LAST ELEMENT
      READ (105,1020) MCEL(IP,IB1,1,1), MCEL(IP,IB1,1,2),
     1MCEL(IP,IB2,1,1), MCEL(IP,IB2,1,2)
C----------------------------------------3> FIRST AND LAST NODE
      READ (105,1020) MCEL(IP,IB1,2,1), MCEL(IP,IB1,2,2),
     1MCEL(IP,IB2,2,1), MCEL(IP,IB2,2,2)
C
      MCEL(IP,IB1,1,1) = MCEL(IP,IB1,1,1) + NE1
      MCEL(IP,IB1,1,2) = MCEL(IP,IB1,1,2) + NE1
      MCEL(IP,IB1,2,1) = MCEL(IP,IB1,2,1) + N1
      MCEL(IP,IB1,2,2) = MCEL(IP,IB1,2,2) + N1
C     IF((MCEL(IP,IB2,1,1)+NEL(IB2)).GT.NOLD) GO TO 248
      MCEL(IP,IB2,1,1) = MCEL(IP,IB2,1,1) + NE2
      MCEL(IP,IB2,1,2) = MCEL(IP,IB2,1,2) + NE2
      MCEL(IP,IB2,2,1) = MCEL(IP,IB2,2,1) + N2
      MCEL(IP,IB2,2,2) = MCEL(IP,IB2,2,2) + N2
 248  CONTINUE
```

```
C-----------------------------------------4> NO. NODES AND ELEMENTS
      DO  250  IB = 1, NB
      READ (105,1020) MCPB(IB,1), MCPB(IB,2)
 250  CONTINUE
C---------------------------------5> LOAD PATH, ITERATION, LOAD INC.
C  1> NO. OF LOAD PATHS AND MAXM ITERATIONS ALLOWED. 30 LOAD STEPS.
C  2> NO. OF EQUAL SIZE LOAD STEPS AND THE TOTAL LOAD AT THE END OF
C     THE LOAD STEPS. CYCLIC LOADING ALLOWED BUT NOT UNPROPORTIONAL.
      READ (105, 1020) NPATH, ITMAX
      WRITE (108, 3040) ITMAX, NPATH
      DO  260  I = 1, NPATH
      READ (105, 1090)  PATH(I,2), PATH(I,1)
      WRITE (108, 3060) I, PATH(I,2), PATH(I,1)
 260  CONTINUE
      PATH(0,1) = PATH(0,2) = 0
C
C----- COMPUTE SOME OF THE VARIABLES.------------------------------
C
      IF(NB.EQ.1) NPLOAD(2)=0
      MEQUN = NEQUN = NTEQUN = NTBC =  NTBP = 0
       DO  280  IB = 1, NB
      NTBC  = NTBC   + NBC(IB)
      NTBP = NTBP + NBP(IB)
      NEQUN = NEQUN  + (NFP(IB) -MCPB(IB,1))*2
      MEQUN = MEQUN  + MCPB(IB,1)*4
      IF(IPTYPE.EQ.1) THEN
      YM(IB) = YM(IB)*(1. +2.*POISON(IB))/((1.+POISON(IB))**2)
      POISON(IB) = POISON(IB)/(1. + POISON(IB))
      END IF
 280   CONTINUE
      NTEQUN = MEQUN + NEQUN
      MTCP    = MTCEL   = 0
      MCPOIN = MCPB(1,1)
       DO  300  I = 1,MCP
      MTCP = MTCP + MCF(I,3)*2
      MTCEL = MTCEL + MCF(I,4)*2
 300   CONTINUE
       NOLD = I1 = 0
       DO  310  IB = 1, NB
       YMOD = YM(IB)
       DO  315  I = 1, NBC(IB)
       I1 = I1 + 1
       BCOND(I1,1) = BCOND(I1,1)/YMOD
       BCOND(I1,2) = BCOND(I1,2)/YMOD
 315   CONTINUE
       DO  317  I = 1, NPLOAD(IB)
       PTLOAD(I,3) = PTLOAD(I,3)/YMOD
       PTLOAD(I,4) = PTLOAD(I,4)/YMOD
 317   CONTINUE
 310   CONTINUE
      IF(NTBP.GT.200) WRITE(108,4000) NTBP ; STOP
      IF(NTBC.GT.160) WRITE(108,4000) NTBC ; STOP
```

```
      IF((MEQUN**2).GT.MXSIZE) WRITE(108,4000) MXSIZE ; STOP
      IF(NMU.LE.0.OR.NMU.GT.7)WRITE(108,4000) NMU ; STOP
C
C------ STORE CORRESPONDING ELEM AND FUNCTIONAL NODES IN CONTACT ZONE
C
        DO  340   I = 1, NB
       DO  320  J  = 1, MTCP
      MPAIR(I,J) = 0
 320   CONTINUE
        DO  340  J = 1, MTCEL
      MEPAIR(I,J) = 0
 340    CONTINUE
C
        DO 420  IP = 1,MCP
      IB1 = MCF(IP,1) ; IB2 = MCF(IP,2)
      NF1 = MCEL(IP,IB1,2,1)
      NF2 = MCEL(IP,IB2,2,1)
      IF(NF1.LT.MCEL(IP,IB1,2,2))THEN
      INC1 = 1
      ELSE
      INC1 = -1
       END IF
       IF(NF2.LT.MCEL(IP,IB2,2,2))THEN
      INC2 = 1
       ELSE
      INC2 = -1
       END IF
C
      NFE1 = MCEL(IP,IB1,1,1)
      NFE2 = MCEL(IP,IB2,1,1)
       IF(NFE1.LT.MCEL(IP,IB1,1,2))THEN
      IEC1 = 1
       ELSE
      IEC1 = -1
       END IF
       IF(NFE2.LT.MCEL(IP,IB2,1,2))THEN
      IEC2 = 1
       ELSE
      IEC2 = -1
       END IF
      IF(IP.EQ.1) INC = IEC = 0
       DO  380  I = 1, MCF(IP,3)
      INC = INC +1
       IF(I.EQ.1) GO TO 360
      NF1 = NF1 + INC1
      NF2 = NF2 + INC2
 360   MPAIR(IB1,INC) = NF1
      MPAIR(IB2,INC) = NF2
 380   CONTINUE
       DO  420  I = 1, MCF(IP,4)
      IEC = IEC + 1
       IF(I.EQ.1) GO TO 400
```

```
      NFE1 = NFE1 + IEC1
      NFE2 = NFE2 + IEC2
 400  MEPAIR(IB1,IEC) = NFE1
      MEPAIR(IB2,IEC) = NFE2
 420  CONTINUE
C
      WRITE (108,2060) MCPOIN
      DO  440 I = 1, NB
      WRITE (108, 2070) I,(MPAIR(I,J), J = 1, MTCP/2)
 440  CONTINUE
      WRITE (108, 2080) MTCEL/2
      DO  450 I = 1, NB
      WRITE (108, 2070) I,(MEPAIR(I,J), J = 1, MTCEL/2)
 450  CONTINUE
C
      RETURN
C
C----------------- FORMATS ----------------------------------------
C
 1000 FORMAT(18A4)
 1020 FORMAT (8I5)
 1040 FORMAT (4F10.0)
 1060 FORMAT (A1,I4,1X,2F10.0)
 1080 FORMAT (I1,I4,I5,1X, 4I1, 1X, 4F10.0)
 1090 FORMAT(I5,1X,7F10.0)
C
 2000 FORMAT('1',10X,18A4,/10X,72('-')/)
 2010 FORMAT(10X,18A4,/)
 2020 FORMAT(/5X,'CONTROL INFORMATION :',//5X,'TYPE OF PROBLEM(IOUT)'
     1,6X,'='I3,/5X,'LT.0 : DATA CHECK'/5X,'EQ.1 : CONTACT ZONE'/5X,
     2'EQ.2 : BACK SUBSTIUTE'/5X,'EQ.0 : 2-D STRESS ANALYSIS'//5X,
     3'TYPE OF ANALYSIS (IPTYPE)  ='I3/5X,'EQ.0 : PLANE STRAIN'/5X,
     4'EQ.1 : PLANE STRESS')
 2030 FORMAT(//5X,'NUMBER OF BODIES   ='I3/5X,'NO. OF CONTACT PAIRS ='
     1,I3,/5X,'NO. OF FUNCTIONAL NODES PER ELEM ='I3,/5X,
     2'ORDER OF INTEGRATION IS VARIABLE : 2,4 OR 10')
 2040 FORMAT(//5X,'INFORMATION FOR BODY 'I2,/5X,'NO. OF ELEMENTS  =',
     1I3/5X,'NO. OF BOUNDARY NODES PER ELEM ='I3/5X,'NO. OF BOUNDARY',
     2' NODES ='I3/5X,'NO. OF FUNCTIONAL NODES ='I3/5X,'NO OF REGIONS'
     3,'='I3/5X,'YOUNGS MODULUS =',E10.3,/5X,'POISSONS RATIO =',E10.3)
 2045 FORMAT(/5X,'BOUNDARY CONDITIONS AND CONTACT INFORMATION :'/)
 2047 FORMAT(/5X,'NO. OF COEFF OF FRICTION ='I3,//5X,
     1'COEFF OF FRICTION = ',7(3X,F10.4)/)
 2050 FORMAT(5X,I3,5X,I3,5X,4I1,4(3X,E10.3))
 2055 FORMAT(25X,4(3X,E10.3))
 2060 FORMAT(/5X,'NODES IN CONTACT :',I3/)
 2070 FORMAT(3X,'BODY 'I1,2X, 25(1X,I3)/)
 2080 FORMAT(/5X,'ELEMENTS IN CONTACT :',I3/)
 3000 FORMAT(/5X,'NODAL COORDINATES FOR BODY 1',20X,
     1'NODAL COORDINATES FOR BODY 2',//5X,'NODE',8X,'X',16X,'Y',20X,
     2'NODE',8X,'X',16X,'Y'//)
 3020 FORMAT(5X,I3,2(5X,E12.5),13X,I3,2(5X,E12.5))
```

```
 3025 FORMAT(55X,I3,2(5X,E12.5))
 3040 FORMAT(///5X,'NUMBER OF ITERATIONS (ITMAX) =' I3, /5X,
     1'NUMBER OF LOAD PATHS (NPATH) ='I3,///5X,'PATH NO.', 3X,
     2'NO. OF STEPS',3X,'TOTAL LOAD'/)
 3060 FORMAT(/7X,I3,7X,F4.1,9X,F10.4,/)
 4000 FORMAT(/5X,'STORAGE LIMIT EXCEEDED CHECK BIGBANG',I6)
C
      END


      SUBROUTINE ASSEMBLE (Z,CRHS,RHS,UNION,MROW,MCOL,ICFLAG ,IB)
C     ---------- -------- ----------------------------------------
C-----------------------------------------------------------------------
C     THIS SUBROUTINE PICKS UP A POLE FOR INTEGRATION ACCORDING
C     TO A BODY AND ITS REGIONS. THEN IT CALLS EQUN OR CEQUN
C     AND ASSEMBLES THE EQUATIONS IN PROPER MANNER. DECOMPOSITOION
C     OF THE EQUATIONS IS DONE AFTER THE ASSEMBLY. THE EQUATIONS ARE
C     WRITTEN IN PROPER FILES.
C-----------------------------------------------------------------------
C
      INCLUDE COMMON, LIST
      DIMENSION CRHS(MSIZE), RHS(NSIZE), UNION(2,NTSIZE)
      DIMENSION Z(MROW,MCOL)
      DIMENSION IC2(4)
C
      ND = 0
      IF(IB.GT.1) THEN
      DO  50  I = IB, NB
      ND = ND + NFP(I-1)
   50 CONTINUE
      END IF
      NBNODE  =  NBN(IB)
      NBPOIN  =  NBP(IB)
      NFPOIN  =  NFP(IB)
      MCPOIN  =  MCPB(IB,1)
      NONCP   =  NFPOIN - MCPOIN
      NREGN   =  NREN(IB)
      NELEM   =  NEL(IB)
      YMOD    =  YM(IB)
      RNU     =  POISON(IB)
      LASTP   =  NPOLE   =  IDIFF   =  0
      IB1 = IB
C-----------------------------------------------LOOP OVER EACH REGION
          DO  800  IREGN  =  1,  NREGN
C---------------------------------------------LOOP OVER EACH ELEMENT
C
        DO  700  IELEM  =  REGN(IB,IREGN,1,1),  REGN(IB,IREGN,1,2)
C
C------------------------------------------------------ESTABLISH NPOLE
        DO  600  INODE  =  1,  NFNODE
      NPOLE  = (IELEM -1)*(NFNODE -1) + INODE
      IF(NFNODE.EQ.1) NPOLE = IELEM
```

```
      IF(NPOLE.GT.REGN(IB,IREGN,2,2))  GO  TO  600
C
C-----FIND IDIFF  (HOW MANY CONTACT POINTS ARE BEFORE THIS NODE ?)
C----------CHECK FOR CONTACT NODE AND FIND CORRESPONDING CONT NODE
C
      IDIFF = IAFLAG = 0
        IF(IOUT.EQ.0)    GO   TO   130
       DO  120  I = 1, MTCP
      IF(MPAIR(IB,I).EQ.0) GO TO 120
      IF(NPOLE.GE.MPAIR(IB,I)) IDIFF = IDIFF + 1
      IF(NPOLE.EQ.MPAIR(IB,I)) THEN
C***   CHECK IF IELEM IS IN MEPAIR
       DO  70 J = 1, MTCEL
       IF(IELEM.EQ.MEPAIR(IB,J))GO TO 80
 70     CONTINUE
       GO  TO 600
 80    IAFLAG = 1
        DO  100  KB = 1, NB
       IF(MPAIR(KB,I).EQ.NPOLE.OR.MPAIR(KB,I).EQ..0) GO TO 100
       NPOLE2 = MPAIR(KB,I)
       IB2 = KB
 100    CONTINUE
      END IF
 120   CONTINUE
C
 130  CONTINUE
      IF(NPOLE.EQ.LASTP)  GO  TO 600
      LASTP = NPOLE
      IF(ICFLAG.EQ.0.AND.IAFLAG.EQ.1) GO TO 600
      IF(ICFLAG.EQ.1.AND.IAFLAG.EQ.0) GO TO 600
C--------------------------------BOUNDARY CONDITIONS AND COORDINATES
       CALL  BCONDN(IELEM, INODE, POLEBC, IPCODE)
      IF(ICFLAG.EQ.0) GO TO 170
C------------------------------------------------STRESS DISCONTINUITY
       ISTD = 0
      DO  140  I  = 1, 4
      ISTD = ISTD + IPCODE(I)
 140  CONTINUE
      IF(ISTD.EQ.4)  GO TO 170
C
      NSTD = NSTD + 1
      IF(INODE.EQ.1)  THEN
      IE2 = IELEM -1
      IN2 = NFNODE
      ELSE IF(INODE.EQ.NFNODE) THEN
      IE2 = IELEM + 1
      IN2 = 1
      ELSE
      IE2 = IELEM
      IN2 = INODE
      END IF
C
```

```
      IF(NFNODE.EQ.1) THEN
      IE2 = IE1
      IN2 = INODE
      END IF
C
      CALL BCONDN (IE2, IN2, POLE2BC, IC2)
      DO  160  I = 1, 2
      IF(IPCODE(I+2).EQ.0) THEN
       CSTD(NSTD,I+3) = 3
      ELSE
       CSTD(NSTD,I+3) = 1
      END IF
      I1 = IPCODE(I) + IC2(I)
      IF(I1.EQ.0) THEN
       CSTD(NSTD,I+1) = 3
       GO TO 160
      ELSE IF(I1.EQ.2) THEN
       CSTD(NSTD,I+1) = 1
       GO TO 160
      END IF
C
      IF(IPCODE(I).EQ.0) THEN
       CSTD(NSTD,I+1) = 2
      ELSE
       CSTD(NSTD,I+1) = 1
      END IF
 160  CONTINUE
      CSTD(NSTD, 1) = NPOLE
C
 170  CONTINUE
      EXISP = SICOD(INODE)
       CALL  COORD(IB, IELEM, EXISP, POLE)
C------------------------------------------INTEGRAL EQUATIONS
      CALL  EQUN (UNION,IB1,IDIFF,ICFLAG,MCOL)
C
      IF(NPLOAD(IB).NE.0) THEN
      CALL DIRAC (IB)
      END IF
C
      DO  200 I = 1, 2
      IF(ICFLAG.EQ.0) THEN
      IROW = (NPOLE -ND -IDIFF -1)*2 + I
       IROW1 = (NPOLE -IDIFF -1)*2 + I
      IF(IB.EQ.2) IROW1 = IROW1 - MCPOIN*2
      ELSE
       IF(IB.EQ.1) THEN
       IROW = (IDIFF -1)*4 + I + 2
       IROW1 = IROW
       ELSE
       IROW = (IDIFF -1)*4 + I
       IROW1 = IROW + 4*MCPOIN
       END IF
```

```
      END IF
      DO  180 J = 1, MCOL
      Z(IROW, J) = UNION(I,J)
 180  CONTINUE
      IF(ICFLAG.EQ.0) THEN
      RHS(IROW1) = ELRHS(I)
      ELSE
      CRHS(IROW1) = ELRHS(I)
      END IF
 200  CONTINUE
               IF(ICFLAG.EQ.0)  GO TO 600
C-----------------------------------------COMPATIBILITY EQUATIONS
C     ASSUMPTION:ALL THE NODES ARE STICKING.
C   1>IB = 1 ==> TX AND TY.(NEVER CHANGING EQUN)
C   2>IB=2:(GT.IB2) ==> UX AND UY EQUNS.
C----------------------------------------------------------------
      CALL CEQUN(UNION,IB1,IB2,IELEM,NPOLE,NPOLE2,MCOL)
C
      DO  240  I  = 1, 2
      IF(IB.LT.IB2) THEN
      IROW = (IDIFF -1)*4 + I
      IROW1 = IROW
      ELSE
      IROW = (IDIFF -1)*4 + I + 2
      IROW1 = IROW + 4*MCPOIN
      END IF
      DO  220  J  = 1, MCOL
      Z(IROW, J) = UNION(I,J)
 220  CONTINUE
      CRHS(IROW1) = ELRHS(I)
 240  CONTINUE
      IF(ISTD.EQ. 4) GO TO 320
C-------------------------------------------DISCONTINIOUS STRESS
C     1 => DONT CHANGE THE EQUATION
C     2 => I.E. IS NEEDED FOR THE STRESSES. (CASE OF DISCONTINUITY)
C     3 => KNOWN TO BE ZERO. PUT 1 ON THE DIAGONAL.
C     4 => I.E. IS NEEDED FOR STRESSES. BUT NO DISCONTINUITY?? BOGUS??
C---------------------------------------------------------------
      DO  300  I = 1, 4
C     ICOL2 = (ID2 -1)*4 + I + (NFP(IB) -MCPOIN)*2
      IROW = (IDIFF -1)*4 + I
      ICOL = (IDIFF -1)*4 + I + (NFP(IB) - MCPOIN)*2
      IROW1 = IROW
C
      IF(IB.GT.IB2) THEN
      IROW1 = IROW + 4*MCPOIN
      ICOL = ICOL + 4*MCPOIN
      END IF
C
      GO TO (300, 260, 280, 260) CSTD(NSTD, I+1)
C
 260  IF(IB.EQ.1) THEN
```

```
      DO  270  J = 1, MCOL
      Z(IROW, J) = Z(IROW +2, J)
 270  CONTINUE
      CRHS(IROW1) = CRHS(IROW1+2)
      END IF
      GO TO 300
C
 280  DO  290  J = 1, MCOL
      Z(IROW, J) = 0.0
 290  CONTINUE
      Z(IROW, ICOL) = 1.0
      CRHS(IROW1) = 0.0
 300  CONTINUE
C---------------------------CHANGE THE EQUN TO AVOID ZERO ON DIAGONAL
 320  CONTINUE
      IF(IB.EQ.1.AND.NPOLE2.EQ.-1) THEN
      DO  340  I = 1, MCOL
      DUMMY = Z(IROW+1, I)
      Z(IROW +1, I) = Z(IROW +2, I)
      Z(IROW +2, I) = DUMMY
 340  CONTINUE
      DUMMY = CRHS(IROW1 +1)
      CRHS(IROW1 +1) = CRHS(IROW1 +2)
      CRHS(IROW1 +2) = DUMMY
      END IF
C
 600     CONTINUE
 700      CONTINUE
 800       CONTINUE
                     IF(IOUT.EQ.0) RETURN
C-------------------------------------------------------DECOMPOSITON
C     THE STATEMENTS WITH X IN THE FIRST COLUMN ARE FOR INTERMEDIATE
C     OUTPUT. COMPILE THE PROGRAM WITH PROPER OPTION SO THAT THESE
C     STATEMENTS ARE COMPILED. ALSO REFER TO IOUT.GE.3 FOR THIS
C
X     WRITE(108,1020) IB,ICFLAG,MROW,MCOL
X     IF(ICFLAG.EQ.1.AND.IB.EQ.2) THEN
X     WRITE (108,1000) RHS(I),I = 1, 2*(NFP(1)+NFP(2)-2*MCPOIN)
X     WRITE (108,1000) CRHS(I),I= 1, 8*MCPOIN
X     END IF
C--------------
      ND1 = (ND -MCPOIN)*2 + 1
      IF(IB.EQ.1) ND1 = 1
      MROW1 = (NFP(IB) -MCPOIN)*2
      MCOL2 = MROW1 + (IB-1)*4*MCPOIN
      ND2   = (IB-1)*4*MCPOIN + 1
C
      IF(ICFLAG.EQ.0) THEN
      CALL  DECOMP( Z, RHS(ND1), MROW, MCOL)
      ELSE
      IF(IB.EQ.1) REWIND NTAPE
      CALL DECOMP1(RHS(ND1),CRHS(ND2),UNION,Z,MROW,MCOL,MROW1,
```

```
     1              MCOL2,MCPOIN,NTAPE)
      END IF
C-----------------------------------------------WRITE THE EQUATIONS
      IF(ICFLAG.EQ.0) THEN
      ITAPE = NTAPE
      IDUMMY = 1
      ELSE
      ITAPE = MTAPE1
      IF(IB.EQ.2) ITAPE = MTAPE2
      IDUMMY = (NFP(IB) -MCPOIN)*2 + 1
      END IF
C
      DO  360 I = 1, MROW
      WRITE(ITAPE) Z(I,J), J = IDUMMY, MCOL
 360  CONTINUE
C
      IF(ICFLAG.EQ.1) THEN
      WRITE(ITAPE) CRHS(I) , I= ND2, IB*4*MCPOIN
      END IF
C-----------------------------------------------------------------
C
X     WRITE(108,1020) IB,ICFLAG,MROW,MCOL
X     OUTPUT(108) 'FOLLOWING COMES OUT OF DECOMP'
X     IF(ICFLAG.EQ.1.AND.IB.EQ.2) THEN
X     WRITE (108,1000) RHS(I),I = 1, 2*(NFP(1)+NFP(2)-2*MCPOIN)
X     WRITE (108,1000) CRHS(I),I= 1, 8*MCPOIN
X     ELSE
X     RETURN
X     END IF
C----------------------
X1000 FORMAT (/8(2X,E10.3))
X1020 FORMAT('1',/5X'IB ='I5,5X'ICFLAG ='I5,5X'MROW ='I5,5X'MCOL='I5/)
C
      RETURN
      END

      SUBROUTINE REFINE (Z,CRHS,CX,UNION,MROW,MCOL)
C     ---------- ------ -------------------------
C-----------------------------------------------------------------
C     THIS SUBROUTINE FINDS THE CORRECT CONTACT AREA FOR A GIVEN
C     LOADING . THE ASSUMPTIONS ARE :
C   1> POTENTIAL CONTACT AREA IS BIGGER THAN THE ACTUAL AREA.
C   2> AT THE START OF THE ITERATIVE SCHEME, ALL NODES ARE STICKING.
C   3> BODY 1 EQUATION CONSISTS OF EQUILIBRIUM EQUNS. NEVER CHANGE.
C   4> NORMAL AND TANGENTIAL STRSS VECTORS ARE EQUAL AND OPPOSITE.
C   5> DISPLACEMENT COMPONENTS ALONG THE AVERAGE NORMAL ARE CHECKED.
C   6> CONTACT MAY CONFORMAL, NONCONFORMAL OR ANY OTHER TYPE(?)
C   7> ONLY BODY2 EQUATIONS ARE READ FROM THE DISK AND CHANGED.
C   8> FORWARD ELIMINATION ON BODY1 EQUATIONS IS PERFORMED ONLY ONCE.
C   9> COULOMB'S LAW OF FRICTION HOLDS.
C  10> USE UNLOADING CAREFULLY!! SMALL LOAD STEPS MAY BE NECESSARY.
C-----------------------------------------------------------------
```

```
C
      INCLUDE COMMON, LIST
      DIMENSION Z(MROW,MCOL)
      DIMENSION CRHS(MSIZE), CX(MSIZE), UNION(2*NTSIZE)
      DIMENSION ICONDN(50,3), KCONDN(3)
      ISOLN = NTENSE = NSLIP = 0
      ALPHA = 1.0
      ALPHA0 = 1.0
      IB1 = 1 ; IB2 = 2
      TLOAD = 0.0
C------------------------------- READ CONTACT EQUNS FOR BODY 1
C                                 THESE ARE READ ONLY ONCE.
      REWIND  MTAPE1
      DO  100  I = 1, 4*MCPOIN
      READ (MTAPE1) Z(I,J),J=1,MCOL
 100  CONTINUE
      READ(MTAPE1) CRHS(I), I = 1, 4*MCPOIN
C
      SGN = 1.0
      DO  120  I = 1, MCPOIN
      ICONDN(I,1) = 0
      DO  120  J = 2, 3
      ICONDN(I,J) = 1
 120  CONTINUE
      DO  140  I = 1, 8*MCPOIN
      UNION(I) = 0.0
 140  CONTINUE
      NSTIK = MCPOIN
      NSLIP = NOPEN = 0
C----------------------------- FIND THE LOAD PATH & LOAD INC.: ALPHA
      DO  901  IPATH = 1, NPATH
      ALPHA = ( PATH(IPATH,1) - PATH(IPATH-1,1))/PATH(IPATH,2)
      ALPHA1 = ALPHA
      ALPHA2 = ALPHA
      LSTEP = PATH(IPATH,2)
C-------------------------- LOAD STEPS FOR A LOAD PATH: PATH(IPATH,2)
      DO  900  ISTEP = 1, LSTEP
C
C------------ IF ITERATIVE SOLUTION HAS FAILED THEN INCREASE ALPHA
C
 143  IF(ISOLN.LT.0) THEN
      ALPHA = 1.05*ALPHA
      IF((TLOAD+ALPHA).LE.0) ALPHA = 0.05 - TLOAD
C
        IF(ISTEP.EQ.LSTEP) THEN
        PATH(IPATH,1) = (ALPHA - ALPHA1) + PATH(IPATH,1)
        ELSE
        ALPHA2 = ALPHA
        END IF
C
      DO  145 I = 1, 4*MCPOIN
      CRHS(I) = CRHS(I)*ALPHA/ALPHA0
```

```
 145   CONTINUE
       DO  147 I = 1, MCOL
       CX(I) = 0.0
 147   CONTINUE
       DO  148  I = 1, MCPOIN
       ICONDN(I,3) = ICONDN(I,2)
 148   CONTINUE
       GO TO 155
C
       ELSE
         IF(ISTEP.EQ.1 .OR. ALPHA2.NE.ALPHA1) THEN
         ALPHA = ALPHA1 + (ALPHA1 - ALPHA2)
         ALPHA2 = ALPHA1
         ELSE
         IF(ALPHA2.NE.ALPHA) THEN
         ALPHA = ALPHA2
          END IF
          END IF
           IF(ALPHA.NE.ALPHA0) THEN
           DO  150  I = 1, 4*MCPOIN
           CRHS(I) = CRHS(I)*ALPHA/ALPHA0
 150       CONTINUE
           END IF
       END IF
 155   CONTINUE
        ALPHA0 = ALPHA
C-----------------------------------------------------------START ITERATIONS
       DO  800  ITER = 1, ITMAX
C
       ICHK = ISOLN = 0
       IFOPEN = IFCONT = IFSTIK = IFSLIP = 0
       ICOPEN = ICSTIK = ICSLIP = 0
       IF(ITER.EQ.1) THEN
         DO  160  I = 1, MCPOIN
         ICONDN(I,1) = ICONDN(I,3)
         ICONDN(I,2) = ICONDN(I,3)
         IF(ISTEP.EQ.1 .AND. IPATH.EQ.1) ICONDN(I,1) = 0
 160     CONTINUE
         GO TO  440
       END IF
C------------------------------------------------------ CHECKING PROCEDURE
       DO  300  INODE = 1, MCPOIN
       IF(MPAIR(IB2,INODE).EQ.0)   GO  TO  300
       NPOLE1 = MPAIR(IB1, INODE)
       NPOLE2 = MPAIR(IB2, INODE)
C
       CALL NFORCE (UNION,CX, NPOLE1,NPOLE2, TX,TY, GAPN,GAPT,G0,RDISP)
C-------------------------------------------------------------------------
       IF(ICHK.GE.-1) THEN
         ICHK = 0
         IF(ICONDN(INODE,2).EQ.0. AND '.ICONDN(INODE,3).EQ.1) ICHK= -1
       END IF
```

```
      ICONDN(INODE, 2) = ICONDN(INODE, 3)
      KCONDN(1) = ICONDN(INODE, 1)
      KCONDN(2) = ICONDN(INODE, 2)
      KCONDN(3) =  100
C---------------------------------------------------------- WRONG GUESS??
C                                           FOR STICK TO SLIP ONLY.
      IF(ICHK.LE.-1 .AND. KCONDN(2).EQ.1) THEN
      IF(ABS(POLE2BC(1)).GT.ABS(SMU*POLE2BC(2))) THEN
        ICHK = ICHK -1
        KCONDN(3) = 2
        IFSLIP = IFSLIP + 1
        ICSLIP = ICSLIP + 1
        GO TO 200
      END IF
      END IF
C---------------------------------------------------------- OPEN => CONTACT ?
C     GAPN = - (POLE2BC(4) - POLEBC(4) - OGAPN)
       IF(KCONDN(2).EQ.0) THEN
      IF(GAPN.GT.0.0) THEN
        KCONDN(3) = 0
        ICOPEN = ICOPEN +1
        GO TO 200
      ELSE
        IFCONT = IFCONT + 1
        KCONDN(3) = 1
        IF(SMU.EQ.0) THEN
          ICSLIP = ICSLIP + 1
          KCONDN(3) = 2
        ELSE
          ICSTIK = ICSTIK + 1
        END IF
        GO TO 200
      END IF
       END IF
C------------------------------------------------------- CONTACT => OPEN ?
      IF(POLE2BC(2).GT.0) THEN
        KCONDN(3) = 0
        IFOPEN = IFOPEN + 1
        ICOPEN = ICOPEN + 1
        GO TO 200
      END IF
C
      IF(SMU.EQ.0) KCONDN(3) = 2 ; GO TO 200
C---------------------------------------------------------- SLIP => STICK ?
       IF(KCONDN(2).EQ.2) THEN
      IF(RDISP*POLE2BC(1)  .LE. 0.0) THEN
        KCONDN(3) = 2
        ICSLIP = ICSLIP + 1
        GO TO 200
      ELSE
        KCONDN(3) = 1
        IFSTIK = IFSTIK + 1
```

```
          ICSTIK = ICSTIK + 1
          GO TO 200
        END IF
         END IF
C----------------------------------------- STICK => SLIP ?
        IF(ABS(POLE2BC(1)). LT .ABS(SMU*POLE2BC(2))) THEN
          KCONDN(3) = 1
          ICSTIK = ICSTIK + 1
          GO TO 200
        ELSE
          IF(ICHK.LE.-1) ICHK = ICHK -1
          KCONDN(3) = 2
          IFSLIP = IFSLIP + 1
          ICSLIP = ICSLIP + 1
        END IF
C
 200    CONTINUE
C-----------------------------------------------------------------------
        ICONDN(INODE, 3) = KCONDN(3)
        IF(KCONDN(3).EQ.100) WRITE(108,2000) NPOLE2,KCONDN(I),I=1,3
 300    CONTINUE
C--------------------------------------------------- FINAL DECISION
C       THIS PROCEDURE NEEDS FURTHER MODIFICATION. AT THIS POINT
C       ONLY ONE "TYPE" OF CHANGE IS ALLOWED IN ALL THE CONTACT NODES.
C       WHAT WE NEED IS ONLY ONE NODE SHOULD CHANGE ITS STATUS.
C       THE DECISION REGARDING WHICH "TYPE" (STICK,SLIP OR OPEN)
C       TO BE CHOSEN FOR CHANGE IS BASED ON THE PROBLEMS I SOLVED!!!
C       IT MAY NEED SOME CHANGES IF I HAVENOT FORSEEN THE TROUBLE.
C-----------------------------------------------------------------------
        IF(SMU.EQ.0) GO TO 400
C-------------------------------------- STICK=>SLIP WRONG GUESS?
        IF(ICHK.LT.-1) THEN
          DO  360  I = 1, MCPOIN
          IF(ICONDN(I,2).EQ.1 .AND. ICONDN(I,3).EQ.2) GO TO 360
          ICONDN(I,3) = ICONDN(I,2)
 360      CONTINUE
          GO TO 400
        END IF
C--------------------------------------- CONTACT=>OPEN?
        IF(IFOPEN.GT.0) THEN
          DO  320  I = 1, MCPOIN
          IF(ICONDN(I,2).NE.0 .AND. ICONDN(I,3).EQ.0) GO TO 320
          ICONDN(I,3) = ICONDN(I,2)
 320      CONTINUE
          GO TO 400
        END IF
C----------------------------------------- OPEN=>CONTACT?
        IF(IFCONT.GT.0) THEN
          DO  340  I = 1, MCPOIN
          IF(ICONDN(I,2).EQ.0 .AND. ICONDN(I,3).EQ.1) GO TO 340
          ICONDN(I,3) = ICONDN(I,2)
 340      CONTINUE
```

```
        GO TO 400
      END IF
C------------------------------------------------ STICK=>SLIP?
      IF(IFSLIP.GT.0) THEN
        DO 380  I = 1, MCPOIN
        IF(ICONDN(I,2).EQ.1 .AND. ICONDN(I,3).EQ.2) GO TO 380
        ICONDN(I,3) = ICONDN(I,2)
  380   CONTINUE
        GO TO 400
      END IF
C-----------------------------------------------------------------
C     NOW WHAT WE HAVE REMAINING IS EITHER SLIP TO STICK CHANGE
C     OR NO CHANGE AT ALL. THIS IS AUTOMATICALLY CHECKED AS WE
C     DECIDE WHETHER SOLUTION IS ACHIEVED.
C------------------------------------------------ ONE MORE TIME ?
  400 CONTINUE
      ISOLN = 0
      DO  420 I = 1, MCPOIN
      IF(ICONDN(I,2) .NE. ICONDN(I,3)) ISOLN = ISOLN + 1
  420 CONTINUE
      IF(ITER.GT.1.AND.ISOLN.EQ.0) GO TO 820
C------------------------------------------------ READ THE EQUATIONS
C                             AND MULTIPLY BY ALPHA THE LOAD INC.
  440 CONTINUE
      REWIND MTAPE2
      DO  460 I = 4*MCPOIN +1, MROW
      READ(MTAPE2) Z(I,J), J = 1, MCOL
  460 CONTINUE
      READ(MTAPE2) CRHS(I), I = 4*MCPOIN +1, MROW
      DO  480  I = 1, MCPOIN
      DO  480  J = 1, 2
      K = 4*MCPOIN + (I-1)*4 + J
      CRHS(K) = CRHS(K)*ALPHA
  480 CONTINUE
      IF(ITER.EQ.1 .AND. IPATH.EQ.1 .AND. ISTEP.EQ.1) GO TO 720
C------------------------------------------------ CHANGE THE EQUATIONS
      DO  700  INODE = 1, MCPOIN
      IF(MPAIR(IB2,INODE).EQ.0) GO TO 700
      NPOLE1 = MPAIR(IB1,INODE)
      NPOLE2 = MPAIR(IB2,INODE)
C
      ISTD = 0
      DO  500  I = 1, NSTD
      IF(CSTD(I,1).EQ.NPOLE2) ISTD = I
  500 CONTINUE
C------------------------------------------------ FIND THE ROW NUMBER
      ID1 = ID2 = 0
      DO  520  I = 1, NB
      DO  520  J = 1, MCPOIN
      IF(MPAIR(I,J).EQ.0) GO TO 520
      IF(NPOLE1 .GE. MPAIR(I,J)) ID1 = ID1 + 1
      IF(NPOLE2 .GE. MPAIR(I,J)) ID2 = ID2 + 1
```

```
 520  CONTINUE
      IROW1 = (ID1 -1)*4
      IROW2 = (ID2 -1)*4
C
      IF(ITER.EQ.1) THEN
       IF(ICONDN(INODE,1).EQ.0) THEN
       UNION(IROW1+1) = 0.0
       UNION(IROW1+2) = 0.0
       UNION(IROW2+1) = 0.0
       UNION(IROW2+2) = 0.0
       END IF
C     IF(INODE.EQ.1) WRITE(108,2005) TLOAD
C     WRITE(108,2020) NPOLE1, UNION(IROW1+1)*YM(1), UNION(IROW1+2)*
C    1 YM(1), UNION(IROW1+3),UNION(IROW1 +4),NPOLE2,UNION(IROW2+3),
C    2 UNION(IROW2+4)
      END IF
C-----------------------------------------------------------------
      CALL NFORCE (UNION,CX, NPOLE1,NPOLE2, TX,TY, GAPN,GAPT,GO,RDISP)
C
      IF(ITER.EQ.1) THEN
      IF(INODE.EQ.1) WRITE(108,2010) TLOAD, TLOADO
      WRITE(108,2020) NPOLE2, TX*YM(2), TY*YM(2),POLE2BC(3),
     1 POLE2BC(4), NPOLE1, POLEBC(3),POLEBC(4), ICONDN(INODE,3)
      END IF
C
      IF(IOUT.GE.3 .AND. ITER.GE.2) THEN
      IF(INODE.EQ.1) WRITE(108,2015) ITER -1
      WRITE(108,1010) NPOLE2, (POLE2BC(1) -TX),(POLE2BC(2) -TY),
     1(POLE2BC(3) -POLE2(1)), (POLE2BC(4) -POLE2(2)), (POLEBC(3) -
     2 POLE(1)), (POLEBC(4) -POLE(2)), RDISP, GAPT, GAPN, NPOLE1,
     3 ICONDN(INODE,2), ICONDN(INODE,3)
      END IF
C
      IF(ICONDN(INODE,3) .EQ. 2) GO TO 620
      IF(ICONDN(INODE,3) .EQ. 1) GO TO 600
C-------------------------------------------------OPEN EQUATIONS
      IF(ISTD.NE.0) THEN
        IF(CSTD(ISTD,4).EQ.1) THEN
          DO 540 I = 1, MCOL
          Z(IROW2+3, I) = Z(IROW2+1, I)
          Z(IROW2+1, I) = 0.0
 540      CONTINUE
          Z(IROW2+1, IROW2+1) = 1.0
          CRHS(IROW2+3) = CRHS(IROW2+1)
          CRHS(IROW2+1) = 0.0
        END IF
C
        IF(CSTD(ISTD,5).EQ.1) THEN
          DO 560 I = 1, MCOL
          Z(IROW2 + 4, I) = Z(IROW2 + 2, I)
          Z(IROW2 + 2, I) = 0.0
 560      CONTINUE
```

```
          CRHS(IROW2 + 4) = CRHS(IROW2 +2)
          CRHS(IROW2 + 2) = 0.0
          Z(IROW2 +2, IROW2 +2) = 1.0
        END IF
        GO TO 700
      END IF
C
      DO  580  I = 1, MCOL
      Z(IROW2 + 3, I) = Z(IROW2 + 1, I)
      Z(IROW2 + 4, I) = Z(IROW2 + 2, I)
      Z(IROW2 + 1, I) = 0.0
      Z(IROW2 + 2, I) = 0.0
 580  CONTINUE
      Z(IROW2 + 1, IROW2 + 1) = UNT2(2,1)
      Z(IROW2 + 1, IROW2 + 2) = UNT2(2,2)
      Z(IROW2 + 2, IROW2 + 1) = UNT2(1,1)
      Z(IROW2 + 2, IROW2 + 2) = UNT2(1,2)
      CRHS(IROW2 + 3) = CRHS(IROW2 + 1)
      CRHS(IROW2 + 4) = CRHS(IROW2 + 2)
      CRHS(IROW2 + 1) = 0.0
      CRHS(IROW2 + 2) = 0.0
      GO TO 700
C------------------------------------------------------- STICK EQUATIONS
 600  CONTINUE
      CRHS(IROW2 + 4) =  -(POLE2(2) -POLE(2) -G0)
C---------------------------------------------------- RELATIVE DELTA UX
      CRHS(IROW2 + 3) = 0.0
C------------
      GO TO 700
C-----------------------------------------------------------SLIP EQUATIONS
 620  CONTINUE
      IF(ISTD.NE.0) GO TO 700
C--------------------------------------------------- DECIDE THE SIGN
      SGN = POLE2BC(1) /ABS(POLE2BC(1))
C     IF(RDISP.NE.0)THEN
C       IF((POLE2BC(1)*RDISP) .GT. 0.0) THEN
C         SGN = -1.* SGN
C         WRITE(108, 3000) RDISP,POLE2BC(1),NPOLE1,NPOLE2,
C    1                        (ICONDN(INODE,J), J=1,3)
C3000 FORMAT(/,5X,'????  ERROR IN SGN DETECTED. VALUES OF RDISP,'
C    1'POLE2BC(1), NPOLE1,NPOLE2,ICONDN(1,2,3) FOLLOW'/5X,2(5X,E14.7),
C    2/5X,2(2X,I5),5X,3(2X,I2))
C       END IF
C     END IF
C-----------------------------------------------------------------------
      DO  640 I = 1, MCOL
      Z(IROW2 + 3, I) = Z(IROW2 + 1, I)
      Z(IROW2 + 1, I) = 0.0
 640  CONTINUE
      CRHS(IROW2 + 3) = CRHS(IROW2 + 1)
      CRHS(IROW2+4) =   -(POLE2(2) -POLE(2) -G0)
C-----------------------------------------------------------------------
```

```
      IF(ITER.LE.ITMAX) THEN
      Z(IROW2 + 1, IROW2 + 1) = UNT2(2,1) + SGN*SMU*UNT2(1,1)
      Z(IROW2 + 1, IROW2 + 2) = UNT2(2,2) + SGN*SMU*UNT2(1,2)
      CRHS(IROW2 + 1) = -1.*(SMU*SGN*TY +TX)
X     CRHS(IROW2+1)= CRHS(IROW2+1) +SGN*SMU*ABS(POLE2BC(2))*ALPHA
      ELSE
C-------------------TX IS BASED ON PREVIOUS NORMAL TRACTION
      Z(IROW2 + 1, IROW2 + 1) = UNT2(2,1)
      Z(IROW2 + 1, IROW2 + 2) = UNT2(2,2)
      CRHS(IROW2 + 1) = SGN*SMU*ABS(POLE2BC(2)) - TX
      END IF
 700  CONTINUE
C------------------------------------------ SOLVE THE EQUATIONS
C     IF FIRST LOAD STEP AND FIRST ITERATION THEN ELIMINATION IS
C     PERFORMED ON ALL THE EQUATIONS. AND THE R.H.S. IS NOT SAVED.
C     ALL OTHER SUBSEQUENT ELIMINATIONS ARE PERFORMED ON THE FOURTH
C     BLOCK ROW AND R.H.S. IS SAVED. (NOT DECOMPOSED.)
C--------------------------------------------------------------------
 720  CONTINUE
      IF(ITER.EQ.1 .AND. IPATH.EQ.1 .AND. ISTEP.EQ.1) THEN
        MR2 = 2
        MR1 = 0
      ELSE
        MR2 = 4*MCPOIN + 1
        MR1 = 1
      END IF
      CALL  EXTORT (Z, CX, CRHS, MROW, MCOL, MR2, MR1)
C-----------------------------------------------------OUTPUT SOLUTION
      IF(ITER.GT.1 .AND. ISOLN.EQ.0) GO TO 820
      IF(ITER.EQ.1) THEN
      WRITE(108, 1020) ISTEP, ALPHA, TLOAD,SMU, MCPOIN
C     WRITE(108, 1040) IFOPEN, IFCONT, IFSTIK, IFSLIP, ICHK
C     WRITE(108, 1060) ICOPEN, ICSTIK, ICSLIP, NOPEN, NSTIK, NSLIP
      WRITE(108, 1070)
      WRITE(108, 1080) MPAIR(IB1,I), I = 1, MCPOIN
      WRITE(108, 1080) MPAIR(IB2,I)- NFP(IB1), I = 1, MCPOIN
      END IF
C     WRITE(108, 1080) ICONDN(I,1), I = 1, MCPOIN
C     WRITE(108, 1080) ICONDN(I,2), I = 1, MCPOIN
      WRITE(108, 1080) ICONDN(I,3), I = 1, MCPOIN
C         IF(IOUT.GE.3 .AND. ITER.EQ.1) THEN
C         WRITE(108, 2005) ITER
C         WRITE(108, 1000) CX(I), I = 1, MCOL
C         END IF
C-----------------------------------------------------------------
C     IF(ITER.GT.1.AND.ISOLN.EQ.0) GO TO 820
 800  CONTINUE
 820  CONTINUE
      WRITE(108,2030) NOPEN, NSTIK, NSLIP
      IF(ITER.LE.ITMAX .AND. ISOLN.EQ.0) THEN
      WRITE(108, 2040)
      ELSE
```

```
C-------------------- ITERATIVE SOLN FAILED. GO BACK ; UPDATE ALPHA.
      WRITE(108,2060)
      ISOLN = -10
      GO TO 143
      END IF
C----------------------------------------------------------- UPDATE----
C                                 ANSWERS ARE ADDED AND STORED IN UNION
      DO  840  I = 1, MCOL
      UNION(I) = UNION(I) + CX(I)
      CX(I) = 0.0
 840  CONTINUE
      NOPEN = ICOPEN
      NSTIK = ICSTIK
      NSLIP = ICSLIP
      TLOAD0 = TLOAD
      TLOAD = TLOAD + ALPHA
      IF(TLOAD.LE.0.0) GO TO 910
C----------------------------------------------------------------------
 900  CONTINUE
 901  CONTINUE
C----------------------------------------------------------------------
 910  CONTINUE
      WRITE(108, 2010) TLOAD, TLOAD0
      NTENSE = NSLIP = 0
      DO 920 I = 1, MCPOIN
      IF(ICONDN(I,3).EQ.0) THEN
      NTENSE = NTENSE +1
      ITENSE(NTENSE) = MPAIR(2,I)
      ELSE IF(ICONDN(I,3).EQ.2) THEN
      NSLIP = NSLIP +1
      ISLIP(NSLIP) = MPAIR(2,I)
      END IF
 920  CONTINUE
      DO  940  I = 1, MCOL
      CX(I) = UNION(I)
 940  CONTINUE
C------------------------------------------------------- FORMATS
C1000 FORMAT(8(2X,1PE11.4))
 1010 FORMAT(1X,I3,9(2X,E9.3),1X,I3,2(1X,I1))
 1020 FORMAT(/15X'LOAD STEP ='I3,//5X,'LOAD FACTOR (ALPHA)  =',F10.4,
     1       3X,'TOTAL LOAD (TLOAD) =',F10.4,/5X,
     2       'COEFF. OF FRICTION (SMU) ='F10.4,3X,
     3       'NO. OF CONTACT POINTS (MCPOIN) ='I3,//)
C1040 FORMAT(2X'IFOPEN ='I3,3X'IFCONT ='I3,3X'IFSTIK ='I3,3X'IFSLIP ='
C    1       I3,3X'ICHK ='I3,/)
C1060 FORMAT(2X'ICOPEN ='I3,3X'ICSTIK ='I3,3X'ICSLIP ='I3,3X'NOPEN ='
C    1       I3,3X'NSTICK ='I3,3X'NSLIP ='I3,/)
 1070 FORMAT(/10X'STATUS OF EACH NODE IN EACH ITERATION: '//)

 1080 FORMAT(/,40(1X,I2),/)
 2000 FORMAT(//12X,'SOMETHING WRONG IN CHECKING',4(3X,I3)//)
C2005 FORMAT(//15X,'ANSWERS FOR THE ITER 'I3,' IN GLOBAL COORD',/)
```

```
 2010 FORMAT(//15X,'ANSWERS FOR THE LOAD 'F10.4,' IN LOCAL COORD',
     1        /15X,'PREVIOUS TOTAL LOAD  'F10.4,/)
 2015 FORMAT(//15X,'ANSWERS FOR THE ITER. 'I3,'  IN LOCAL COORD.'/)
 2020 FORMAT(/3X,I3,3X,4(E11.4,2X),3X,I3,3X,2(E11.4,2X),2X,I1)
 2030 FORMAT(/5X,'NOPEN ='I3,3X,'NSTICK =',I3,3X'NSLIP ='I3,/)
 2040 FORMAT(/5X,'ITERATIVE SOLUTION IS SUCCESSFULL'//)
 2060 FORMAT(/15X,'>>>>>> NO UNIQUE SOLUTION WAS ACHIEVED <<<<<<'//)
C
      RETURN
      END
      SUBROUTINE  NFORCE (UNION,CX, NPOLE1,NPOLE2, TX,TY,
     1                    GAPN,GAPT,G, RDISP)
C
C    ----------- ------- -------------------------------
C    -------------------         COMPUTE THE FORCES IN LOCAL COORDINATES.
C
      INCLUDE COMMON
      DIMENSION CX(MSIZE), UNION(2*NTSIZE)
      DIMENSION OGAP(2)
      IB1 = 1
      IB2 = 2
C
      ISTD = 0
      DO  100  I = 1, NSTD
      IF(CSTD(I,1).EQ.NPOLE2) ISTD = I
 100  CONTINUE
C-------------------------------------ELEM NO. AND LOCAL NODE NUMBER.
      IE1 = IE2 = 0
      DO  120  I = 1, MTCEL/2
      DO  120  J = 1, NFNODE
      IF(MEPAIR(IB1,I).EQ.0)      GO  TO  120
      IN1 = (MEPAIR(IB1,I) -1)*(NFNODE -1) + J
      IN2 = (MEPAIR(IB2,I) -1)*(NFNODE -1) + J
            IF(NFNODE.EQ.1) THEN
            IN1 = MEPAIR(IB1,I)
            IN2 = MEPAIR(IB2,I)
            END IF
      IF(IN1.EQ.NPOLE1) THEN
      ID1 = J
      EXISP1 = SICOD(J)
      IE1 = MEPAIR(IB1,I)
      END IF
            IF(IN2.EQ.NPOLE2) THEN
            ID2 = J
            EXISP2 = SICOD(J)
            IE2 = MEPAIR(IB2,I)
            END IF
      IF(IE1.NE.0.AND.IE2.NE.0)    GO  TO  140
 120  CONTINUE
 140  CONTINUE
C------------------------------------------------NORMALS AND TANGENTS
      CALL NORMAL(IB1, IE1, ID1, UNT1)
```

```
      CALL NORMAL(IB2, IE2, ID2, UNT2)
      CALL COORD(IB1, IE1, EXISP1, POLE)
      CALL COORD(IB2, IE2, EXISP2, POLE2)
      GAPN = GAPT = 0.
C-------------------------------------------AVERAGE NORMAL AND GAP
C       OGAPN = ORIGINAL NORMAL GAP
C       OGAPT = ORIGINAL TANGENTIAL GAP
      DO  180  I = 1, 2
      DUM = SQRT((UNT2(I,1) -UNT1(I,1))**2 +(UNT2(I,2)-UNT1(I,2))**2)
      DO  160  J = 1, 2
      UNTAVG(I,J) = (UNT2(I,J) - UNT1(I,J))/DUM
 160  CONTINUE
      OGAP(I) = POLE(I) - POLE2(I)
 180  CONTINUE
      G = SQRT(OGAP(1)**2 + OGAP(2)**2)
      IF(G.GE. 0.00001) THEN
      UNTAVG(1,1) =  OGAP(1)/G
      UNTAVG(1,2) =  OGAP(2)/G
      UNTAVG(2,1) = -OGAP(2)/G
      UNTAVG(2,2) =  OGAP(1)/G
      END IF
C-------------------------------------------------------THE ROW NUMBER
      ID1 = ID2 = 0
      DO  200  I = 1, NB
      DO  200  J = 1, MTCP
      IF(MPAIR(I,J).EQ.0)       GO  TO  200
      IF(NPOLE1.GE.MPAIR(I,J)) ID1 = ID1 +1
      IF(NPOLE2.GE.MPAIR(I,J)) ID2 = ID2 +1
 200  CONTINUE
      IROW1 = (ID1 -1)*4
      IROW2 = (ID2 -1)*4
C----------------------------------------STRESSES AND DISPLACEMENTS
C      POLE =  TOTAL DISP. FOR BODY1
C      POLE2 = TOTAL DISP. FOR BODY2 UPTO PREVIOUS INC.
C      TX, TY => TRACTIONS UPTO PREVIOUS INCREMENT
C-------------------------------------------------------------------
      TX = TY = 0.0
      POLE(1) = POLE(2) = 0.0
      POLE2(1) = POLE2(2) = 0.0
      DO  220  I = 1, 4
      POLEBC(I) = 0.0
      POLE2BC(I) = 0.0
      DO  220  J = 1, 2
C
      DUM = 1.0
      IF(ISTD.NE.0) THEN
      IF(CSTD(ISTD,J+1).NE.1) DUM =0.0
      END IF
C
      IF(I.EQ.1) THEN
      TX = TX + UNION(IROW2 +J) * UNT2(3-I, J) * DUM
      ELSE IF(I.EQ.2) THEN
```

```
      TY = TY + UNION(IROW2 +J) * UNT2(3-I, J) * DUM
      END IF
                       IF(I.LE.2) THEN
      POLEBC(I) = POLEBC(I) +(UNION(IROW1 + J) +CX(IROW1 + J))
     1          * UNT1(3-I, J) * DUM
      POLE2BC(I) = POLE2BC(I) +(UNION(IROW2 + J) +CX(IROW2 + J))
     1          *  UNT2(3-I, J) * DUM
                       ELSE
      POLEBC(I) = POLEBC(I) +(UNION(IROW1+2+ J) +CX(IROW1+2 +J))
     1          * UNTAVG(5-I,J)
      POLE2BC(I) = POLE2BC(I) +(UNION(IROW2+2+J) +CX(IROW2+2 +J))
     1          * UNTAVG(5-I,J)
      POLE(I-2) = POLE(I-2) + UNION(IROW1 + 2 + J) * UNTAVG(5-I, J)
      POLE2(I-2) = POLE2(I-2) + UNION(IROW2 + 2 + J) * UNTAVG(5-I, J)
                       END IF
  220 CONTINUE
C-----------------------------------------------------------------
C     POLE2(1) =  TANGENTIAL GAP UPTO PREVIOUS INCREMENT
C     POLE2(2) =  NORMAL GAP UPTO PREVIOUS INCREMENT
C     RDISP =  RELATIVE TANGENTIAL DISPLACEMENT FOR THIS INCREMENT
C-----------------------------------------------------------------
      RDISP = (POLE2BC(3) -POLE2(1)) -(POLEBC(3) -POLE(1))
C     POLE2(1) = - (POLE2(1) - POLE(1))
C     POLE2(2) = - (POLE2(2) - POLE(2) - G)
      GAPN = -(POLE2BC(4) - POLEBC(4) -G)
      GAPT = -(POLE2BC(3) - POLEBC(3))
C
      RETURN
      END
      SUBROUTINE CEQUN (UNION,IB1,IB2,IELEM1,NPOLE1,NPOLE2,MCOL)
C     ---------- ----- ---------------------------------------
C--------------------------------------------------------------------
C     CONSTRUCTS THE COMPATIBILITY AND EQUBRM EQUNS FOR A GIVEN NODE.
C     ALL THE NODES ARE ASSUMED TO BE STICKING.
C--------------------------------------------------------------------
      INCLUDE COMMON, LIST
      DIMENSION UNION(2, NTSIZE)
C-------------------------------------------------ELEMENT IN CONTACT
      DO  100  I = 1, MTCEL
      IF(IELEM1.EQ.MEPAIR(IB1,I)) THEN
      IELEM2 = MEPAIR(IB2,I)
      GO TO 120
      END IF
  100 CONTINUE
  120 CONTINUE
C-----------------------------------LOCAL COORDINATE & NODE NUMBER.
      DO  140  I = 1, NFNODE
      IN1 = (IELEM1-1)*(NFNODE -1) + I
      IN2 = (IELEM2-1)*(NFNODE -1) + I
      IF(NFNODE.EQ.1) THEN
      IN1 = IELEM1
      IN2 = IELEM2
```

```
      END IF
      IF(IN1.EQ.NPOLE1) THEN
       ICOL1 = I
CCC    EXISP1 = SICOD(I)
      END IF
      IF(IN2.EQ.NPOLE2) THEN
       ICOL2 = I
       EXISP2 = SICOD(I)
      END IF
 140  CONTINUE
C-----------------------------------------------NORMAL AND TANGENT
      CALL BCONDN (IELEM1, ICOL1, POLEBC, IPCODE)
      CALL  NORMAL (IB1, IELEM1, ICOL1, UNT1)
      CALL  NORMAL (IB2, IELEM2, ICOL2, UNT2)
      DO  150  I = 1, 2
      DUM = SQRT((UNT1(I,1) -UNT2(I,1))**2 +
     1           (UNT1(I,2) -UNT2(I,2))**2)
      DO  150  J = 1, 2
      UNTAVG(I,J) = (UNT1(I,J) -UNT2(I,J))/DUM
      IF(IB1.LT.IB2) UNTAVG(I,J) = -UNTAVG(I,J)
 150  CONTINUE
C----------------------------------------------------COLUMN NUMBERS
      DO  160  I = 1, 2
      DO  160  J = 1, MCOL
      UNION(I,J) = 0.0
 160  CONTINUE
      ELRHS(1) = 0.0
      ELRHS(2) = 0.0
C
      ID1 = ID2 = 0
      DO  200  I = 1, NB
      DO  180  J = 1, MTCP
      IF(MPAIR(I,J).EQ.0) GO  TO 180
      IF(NPOLE1.GE.MPAIR(I,J)) ID1 = ID1 + 1
      IF(NPOLE2.GE.MPAIR(I,J)) ID2 = ID2 + 1
 180  CONTINUE
 200  CONTINUE
      ICOL1 = (NFP(IB1) -MCPOIN)*2 + (ID1 -1)*4
      ICOL2 = (NFP(IB1) -MCPOIN)*2 + (ID2 -1)*4
C
      IF(IB1.GT.1)  GO  TO 240
C------------------------------------------EQUATIONS FOR STRESSES
      DO  220 I = 1, 2
      J = 3 - I
      DO  220  K = 1, 2
C     IF(YM(IB2).LE.YM(IB1)) THEN
      UNION(J, ICOL2 +K) = (YM(IB2)/YM(IB1))*UNT2(I, K)*IPCODE(K)
      UNION(J, ICOL1 +K) = - UNT1(I,K)*IPCODE(K)
C     ELSE
C     UNION(J, ICOL1 +K) = -(YM(IB1)/YM(IB2))*UNT1(I, K)*IPCODE(K)
C     UNION(J, ICOL2 +K) =  UNT2(I,K)*IPCODE(K)
C     END IF
```

```
 220  CONTINUE
      ELRHS(1) = 0.0
      ELRHS(2) = 0.0
      IF(ABS(UNT1(1,1)).GE.0.999) NPOLE2 = -1
                        GO  TO 400
C-----------------------------------------------------EQUNS FOR DISPLACEMENTS
C-----GAP IS STORED IN POLE2
C----- +VE -> BODY2 ->BODY1 ( IB1 -> IB2 !!)
C--------------------------------------------------------------------------
 240  CONTINUE
      CALL COORD (IB2, IELEM2, EXISP2, POLE2)
      DO  260  I = 1, 2
      POLE2(I) = (POLE2(I) -POLE(I))
 260  CONTINUE
      G = SQRT(POLE2(1)**2 + POLE(2)**2)
      IF(G.GE. 0.00001) THEN
      UNTAVG(1,1) = POLE2(1)/G
      UNTAVG(1,2) = POLE2(2)/G
      UNTAVG(2,1) = -POLE2(2)/G
      UNTAVG(2,2) = POLE2(1)/G
      END IF
C-----------------------------------------------------U(IB1) - U(IB2) = GAP
      DO  300  I = 1, 2
      J = 3 -I
      DO  280  K = 1, 2
      UNION(J, ICOL1 + K+2) = UNTAVG(I,K)*IPCODE(K+2)
      UNION(J, ICOL2 + K+2) = -UNTAVG(I,K)*IPCODE(K+2)
      ELRHS(J) =  ELRHS(J) + POLE2(K)*UNTAVG(I,K)
 280  CONTINUE
 300  CONTINUE
C
 400  CONTINUE
      RETURN
      END
      SUBROUTINE NEWS (CX, RHS)
C     ---------- ---- ---------
      INCLUDE COMMON
      DIMENSION CX(MSIZE), RHS(NSIZE)
C
C------ PRINT THE RESULTS---------------------------------------------
C
C     FIRST PRINT RESULTS PERTAINING TO CONTACT ZONE FOR EACH BODY.
C     THEN IF IOUT.GE.2 THEN EVERYTHING IN GLOBAL COORDINSTES.
C     THESE ARE THE FINAL RESULTS. INTERMEDIATE RESULTS SEE REFINE.
      DO  300  IJ = 1, 2
      IFC = 1
      IF(IJ.EQ.2) IFC = 0
      IF(IJ.EQ.2.AND.IOUT.EQ.1) GO TO 300
      IF(IOUT.NE.0) THEN
      IF(IJ.EQ.2.AND.SMU.NE.FMU(NMU))  GO TO 300
      END IF
      DO  280  IB = 1, NB
```

```
      IB2 = 2
      IF(IB.EQ.2) IB2 = 1
      YMOD = YM(IB)
      NREGN = NREN(IB)
      MCPOIN = MCPB(IB,1)
      NBNODE = NBN(IB)
      WRITE (108, 2000) TITLE
      IF(IFC.EQ.1) THEN
      WRITE (108, 2020) IB, SMU
      ELSE
      WRITE (108, 2040) IB
      END IF
      WRITE (108, 2060)
      DO  280  IR = 1, NREGN
      DO  280  IE = REGN(IB,IR,1,1), REGN(IB,IR,1,2)
      DO  280  IN = 1, NFNODE
C
      N = (IE -1)*(NFNODE -1) + IN
      IF(NFNODE.EQ.1) N = IE
      IF(N.GT.REGN(IB,IR,2,2)) N =REGN(IB,IR,2,1)
      ID = IF = INF = 0
      DO  140 I = 1, MTCP
      IF(MPAIR(IB,I).EQ.0) GO TO 140
      IF(N.GE.MPAIR(IB,I)) ID = ID + 1
      IF(N.EQ.MPAIR(IB,I)) THEN
      INF = 1
      DO  100  J = 1, MTCEL
      IF(IE.EQ.MEPAIR(IB,J)) GO TO 120
 100  CONTINUE
      IF = 0
      GO TO 150
 120  IF = 1
      END IF
 140  CONTINUE
C
 150  IF(IFC.EQ.1.AND.IF.EQ.0) GO TO 280
      CALL BCONDN(IE, IN, POLEBC, IPCODE)
      CALL  COORD (IB, IE, SICOD(IN), POLE)
      IF(IFC.EQ.0.AND.IF.EQ.0) GO TO 250
C
      ID = (ID -1)*4
      IF(IB.EQ.2) ID = ID + 4*MCPOIN
      DO 160 I = 1, MCPOIN
      IF(N.EQ.MPAIR(IB,I)) N2 = MPAIR(IB2,I)
 160  CONTINUE
      IF(IFC.EQ.0.AND.IF.EQ.1) GO TO 195
C----------------------------------------------------- UNIT NORMAL
      IE2 = IN2 = 0
      DO 170 I = 1, MTCEL
      DO 170 J = 1, NFNODE
      IF(MEPAIR(IB2,I).EQ.0) GO TO 170
      IN2 = (MEPAIR(IB2,I) -1)*(NFNODE -1) + J
```

```
      IF(NFNODE.EQ.1) IN2 = MEPAIR(IB2,I)
      IF(IN2.EQ.N2) THEN
      IN2 = J
      IE2 = MEPAIR(IB2,I)
      GO TO 180
      END IF
 170  CONTINUE
 180  CONTINUE
      CALL NORMAL (IB, IE, IN, UNT1)
      CALL NORMAL (IB2, IE2, IN2, UNT2)
      DO  190 I = 1, 2
      DUM = SQRT((UNT2(I,1) -UNT1(I,1))**2 +(UNT2(I,2)-UNT1(I,2))**2)
      DO 190  J = 1, 2
      UNTAVG(I,J) = (UNT2(I,J) -UNT1(I,J))/DUM
      IF(IB.GT.IB2) UNTAVG(I,J) = -UNTAVG(I,J)
 190  CONTINUE
C-------------------------------------STRESSES AND DISPLACEMENTS.
 195  CONTINUE
      DO  200 I = 1, 4
      POLEBC(I) = 0.0
      IF(IFC.EQ.1) THEN
C----------------------------------- ANSWERS IN LOCAL COORDINATES.
      DO  197 J = 1, 2
      IF(I.LE.2) THEN
      POLEBC(I) = POLEBC(I) +CX(ID +J)*UNT1(3 -I, J)*YMOD*IPCODE(J)
      ELSE
      POLEBC(I) = POLEBC(I) + CX(ID + 2+ J)*UNTAVG(5-I,J)*IPCODE(J+2)
      END IF
 197  CONTINUE
      ELSE
C----------------------------------- ANSWERS IN GLOBAL COORDINATES.
      POLEBC(I) = CX(ID + I)
      IF(I.LE.2) POLEBC(I) = POLEBC(I)*YMOD
      END IF
 200  CONTINUE
C---------------------------- DECIDE ON CONDITION OF THE CONTACT NODE.
      IDUMMY = 0
      N1 = N
      IF(IB2.GT.IB) N1 = N2
      DO  210  I = 1, NTENSE
      IF(N1.EQ.ITENSE(I)) THEN
      IDUMMY = 1
      GO TO 230
      END IF
 210  CONTINUE
      DO  220  I = 1, NSLIP
      IF(N1.EQ.ISLIP(I)) THEN
      IDUMMY = 2
      GO TO 230
      END IF
 220  CONTINUE
C---------------------------------- PRINT ACCORDING TO THE CONDITION.
```

```
 230  CONTINUE
      IF(IN.EQ.1) WRITE(108,3040)
      IF(IDUMMY.EQ.0) THEN
      WRITE (108, 2080) IE, POLE(1),POLE(2),POLEBC(3),POLEBC(4),
     1                  POLEBC(1),POLEBC(2)
      ELSE IF(IDUMMY.EQ.1) THEN
      WRITE (108, 3000) IE, POLE(1),POLE(2),POLEBC(3),POLEBC(4),
     1                  POLEBC(1),POLEBC(2)
      ELSE
      WRITE (108, 3020) IE, POLE(1),POLE(2),POLEBC(3),POLEBC(4),
     1                  POLEBC(1),POLEBC(2)
      END IF
      GO TO 280
C
C------ PRINT RESULTS FOR NON CONTACT ZONE -----------------------
C
 250  CONTINUE
      IF(INF.EQ.1) THEN
      ID = (ID-1)*4
      IF(IB.EQ.2) ID = ID +4*MCPOIN
      ELSE
      ID = (N -ID -1)*2
      IF(IB.EQ.2) ID = ID - 2*MCPOIN
      END IF
C
          IF(INF.EQ.1) THEN
      DO 260 I = 1, 2
      POLEBC(I) = 0.0
      POLEBC(I+2) = CX(ID+I+2)
 260  CONTINUE
          ELSE
      DO 270  I = 1, 4
      I1 = 1
      IF(IPCODE(I).EQ.1) THEN
      IF(I.EQ.2.OR.I.EQ.4) I1 = 2
      POLEBC(I) = RHS(ID +I1)
      END IF
 270  CONTINUE
          END IF
C
      IF(IN.EQ.1) WRITE(108,3040)
      WRITE (108, 2080) IE, POLE(1),POLE(2), POLEBC(3),POLEBC(4),
     1                  POLEBC(1)*YMOD, POLEBC(2)*YMOD
 280  CONTINUE
 300  CONTINUE
C
C----------------------FORMATS----------------------------------
C
 2000  FORMAT('1',10X,18A4,/,10X,72('-')/)
 2020 FORMAT(//5X,'RESULTS FOR THE CONTACT ZONE OF BODY 'I1,//5X,
     1'STRESSES ARE IN THE LOCAL COORDINATES '//5X,
     2'DISPLACEMENTS ARE IN AVG. LOCAL COORDINATES'/5X,
```

```
     3'AVG. LOCAL COORD. IS +VE : B2 ---> B1'//5X,
     4'COEFF. OF FRICTION :::::::'2X,F10.5,' ' :::::::'/)
 2040 FORMAT(//5X,'ANSWERS IN GLOBAL COORDINATES FOR BODY 'I1,/)
 2060 FORMAT(//4X,'ELEM',9X,'X',15X,'Y',13X,'X-DISP',10X,'Y-DISP',
     1 9X,'TRACTION X',6X,'TRACTION Y'//)
 2080 FORMAT(5X,I3,6(4X,E12.5))
 3000 FORMAT(5X,I3,6(4X,E12.5),1X,'OPEN')
 3020 FORMAT(5X,I3,6(4X,E12.5),1X,'SLIP')
 3040 FORMAT(/)
C
      RETURN
      END
      SUBROUTINE NORMAL (IB, IE1, IN1, UNT)
C     ------------------------------------
C     THIS SUBROUTINE COMPUTES THE UNIT NORMAL IN THE CONTACT ZONE.
C     AT THE CORNER NODE THERE CAN BE DISCONTINUITY.
C
      INCLUDE COMMON
      DIMENSION UNT(2,2), U1(2,2), U2(2,2)
C----------------------------- FIND IF GIVEN NODE IS A CORNER NODE.
      IF(IN1.EQ.1) THEN
      IE2 = IE1 - 1
      IN2 = NFNODE
      ELSE IF(IN1.EQ.NFNODE) THEN
      IE2 = IE1 + 1
      IN2 = 1
      ELSE
      IE2 = IE1
      IN2 = IN1
      END IF
C
      IF(NFNODE.EQ.1) THEN
      IE2 = IE1
      IN2 = IE1
      END IF
C---------------------------------GET THE NORMAL FOR GIVEN ELEMENT.
      DO 100 I = 1, NFNODE
      IF(IN1.EQ.I) EXISP1 = SICOD(I)
      IF(IN2.EQ.I) EXISP2 = SICOD(I)
  100 CONTINUE
      CALL LOCAL (IB, IE1, EXISP1, U1)
      IF(IE1.EQ.IE2) GO TO 140
C-----------------------------GET THE NORMAL FOR CONNECTING ELEMENT.
      DO 120 I = 1, MTCEL
      IF(IE2.EQ.MEPAIR(IB,I)) THEN
      CALL LOCAL (IB, IE2, EXISP2, U2)
      GO TO 180
      END IF
  120 CONTINUE
  140 DO 160 I = 1, 2
      DO 160 J = 1, 2
      U2(I,J) = U1(I,J)
```

```
 160  CONTINUE
C---------------------------------------------- GET THE AVERAGE NORMAL.
 180  CONTINUE
      DO  200  I = 1, 2
      EXISP1 = SQRT((U1(I,1) +U2(I,1))**2 +(U1(I,2) +U2(I,2))**2)
      DO  200  J = 1, 2
      UNT(I,J) = (U1(I,J) + U2(I,J))/EXISP1
 200  CONTINUE
      RETURN
      END
      SUBROUTINE BACKSUB(A,CX,RHS,M1,M2,MC,NTAPE)
C     ---------- ---------------------------------
C     AFTER CORRECT CONTACT ZONE IS FOUND, BACKSUBSTITUE TO FIND
C     DISP AND STRESSES IN NON CONTACT ZONE. ANSWERS ARE STORED IN RHS
C     THIS ROUTINE WORKS ONLY FOR TWO BODY CONTACT.
C     M1 = NO. OF UNKNOWNS IN NON CONTACT ZONE FOR B1
C     M2 = NO. OF UNKNOWNS IN NON CONTACT ZONE FOR B2
C     MC = NO. OF UNKNOWNS IN CONTACT ZONE FOR B1 OR B2(4*MCPOIN)
C
      DIMENSION A(1)
      DIMENSION  CX(M1+M2), RHS(2*MC)
C
      DO  140  IB = 2, 1, -1
      IF(IB.EQ.1) THEN
       MCX = 0
       MR = 0
       N   = M1
      ELSE
       MCX = MC
       MR  = M1
       N   = M2
      END IF
      DO  140  IN = N, 1, -1
      TEMP = RHS(IN +MR)
C---------------------------------------- READ NON-CONTACT ZONE EQUNS
      BACKSPACE NTAPE
      READ (NTAPE) A(I), I = 1, N+MC
C------------------------------------------ BACKSUBSTITUE.
      DO  100  I = 1, MC
      TEMP = TEMP - A(I +N)* CX(I +MCX)
 100  CONTINUE
C
      DO  120  I = IN+1, N
      TEMP = TEMP - A(I)* RHS(I +MR)
 120  CONTINUE
      RHS(IN +MR) = TEMP/A(IN)
      BACKSPACE NTAPE
 140  CONTINUE
C
      RETURN
      END
      SUBROUTINE LOCAL (IBD, IEL, EXISP, ROT)
```

```
C     ---------- ----- --------------------
C-----------------------------------------------------------------------
C     FINDS THE UNIT NORMAL AND TANGENT FOR A GIVEN POIN IN AN ELEM.
C-----------------------------------------------------------------------
      INCLUDE COMMON, LIST
      DIMENSION ROT(2,2)
      ROT(1,1) = ROT(1,2) = 0.0
      ROT(2,1) = ROT(2,2) = 0.0
C------------------------------------------------------GLOBAL BOUNDARY NODES
      IN1 = IE1 = 0
      DO  100  I = 1, IBD -1
      IN1 = IN1 + NBP(I)
      IE1 = IE1 + NEL(I)
 100  CONTINUE
C----------------------------------------------DERIVATIVE & BOUNDARY NODE
      CALL EVOLVE( EXISP, NBN(IBD), 1)
      DO  180  I = 1, NBN(IBD)
      IN = (IEL - IE1 -1)*(NBN(IBD) -1) + I + IN1
      DO  120  J = 1, NB
      DO  120  K = 1, NREN(J)
      IF(IEL.NE.REGN(J,K,1,2)) GO TO 120
      IF(I.NE.NBN(IBD)) GO TO 140
      IN = REGN(J,K,3,1)
      GO TO 140
 120  CONTINUE
 140  CONTINUE
C-----------------------------------------------------NORM OF THE JACOBIAN
      DO  160  IDIME = 1, 2
      ROT(1,IDIME) = ROT(1,IDIME) + DERIV(I)*BNODE(IN,IDIME)
 160  CONTINUE
 180  CONTINUE
C
      ROT(2,2) = SQRT(ROT(1,1)**2 + ROT(1,2)**2)
      ROT(2,1) = ROT(1,1)
C-------------------------------------------------------FINAL EXPRESSION
C     ROT(1,J) -> UNIT NORMAL
C     ROT(2,J) -> UNIT TANGENT
C-----------------------------------------------------------------------
      ROT(1,1)  =  ROT(1,2) / ROT(2,2)
      ROT(1,2)  = -ROT(2,1) / ROT(2,2)
      ROT(2,1)  = -ROT(1,2)
      ROT(2,2)  =  ROT(1,1)
C
      RETURN
      END
      SUBROUTINE COORD (IBD, IEL, EXISP, C)
C     ---------- ----- -------------------
C-----------------------------------------------------------------------
C       GIVEN THE LOCAL COORDINATE,THE ELEMENT NO. AND THE BODY NO.
C       CALCULATE THE GLOBAL COORDINATES.
C-----------------------------------------------------------------------
C
```

```
      DIMENSION C(2)
      INCLUDE COMMON, LIST
C
      NODES = NBN(IBD)
      CALL  EVOLVE ( EXISP, NODES, 0)
C
      C(1) = C(2) = 0.0
      IN1 = 0
      IE1 = 0
      DO  90  I = 1, IBD-1
      IN1 = IN1 + NBP(I)
 90   IE1 = IE1 + NEL(I)
         DO  160  I = 1, NBN(IBD)
      IN = (IEL-IE1-1) * (NBN(IBD)-1)+I + IN1
C----- CHECK FOR THE LAST ELEMENT IN THE REGION
       DO  100  J = 1, NB
       DO  100  K = 1, NREN(J)
      IF(IEL.NE.REGN(J,K,1,2)) GO TO 100
      IF(I.NE.NODES) GO TO 120
      IN = REGN(J,K,3,1)
      GO TO 120
 100   CONTINUE
 120    CONTINUE
        DO  140  IDIME = 1, 2
      C(IDIME) = C(IDIME) + SHAPE(I)*BNODE(IN,IDIME)
 140    CONTINUE
 160     CONTINUE
      RETURN
      END
      SUBROUTINE BCONDN ( IEL, NODE, BC, IBC)
C     ---------- ------ --------------------
C---------------------------------------------------------------------
C      FOR A GIVEN ELEMENT AND LOCAL NODE , THIS SUBROUTINE
C      FINDS THE BOUNDARY CONDITION. THREE POSSIBILITIES EXIST.
C      1> BY DEFAULT   2> CONTACT NODE  3> BC SPECIFIED
C---------------------------------------------------------------------
C
      INCLUDE COMMON, LIST
      DIMENSION BC(4), IBC(4)
C
C----- CHECK FOR CONTACT ELEM-------------------
      II = 0
        DO  120  I = 1, NB
        DO  120  J = 1, MTCEL
      IF(IEL.EQ.MEPAIR(I,J)) THEN
       DO  100  K = 1, 4
       BC(K) = 0.
       IBC(K) = 1
      II = 1
 100   CONTINUE
      END IF
 120    CONTINUE
```

```
C
C----- CHECK FOR SPECIFIED BC (INCLUDES CONTACT ZONE) ------------
        DO  160  I = 1, NTBC
      IF(IEL.EQ.ICODE(I,1).AND.NODE.EQ.ICODE(I,2)) THEN
       DO  140  J = 1,4
      IBC(J) = ICODE(I,J+2)
       BC(J) = BCOND(I,J)
 140   CONTINUE
               GO  TO  180
      END IF
 160    CONTINUE
      IF(II.EQ.1) GO TO 180
C
C----- BC BY DEFAULT -------------------------
      BC(1) = BC(2) = 0.0
      BC(3) = BC(4) = 0.0
      IBC(1)= IBC(2)= 0.
      IBC(3)= IBC(4)= 1
C
 180    RETURN
      END
      SUBROUTINE DECOMP1 (V1,V2,B,Z,MROW,MCOL,MROW1,MCOL2,
     1                       MPOIN,NTAPE)
C     ---------- ------- ------------------------------
C-----------------------------------------------------------------
C     THIS SUBROUTINE DECOMPOSES THE CONTACT EQUATIONS.
C     ONLY THE BLOCKS OF CORRESPONDING BODY ARE DECOMPOSED.
      DIMENSION V1(1), V2(1), B(1)
      DIMENSION Z(MROW, MCOL)
      MCOL1 = MROW1 + 4*MPOIN
      MROW2 = 4*MPOIN
C
C     MROW AND MCOL ARE DIMENSIONS USED FOR Z
C     MCOL1 = NO. OF COLUMNS TOBE READ
C     MCOL2 = STARTING COLUMN NO. TOBE DECOMPOSED
C     MROW1 = NO. OF ROWS TOBE READ = (NFP(IB) -MPOIN)*2
C     MROW2 = NO. OF ROWS DECOMPOSED
C
      IF(MCOL2.EQ.MROW1) THEN
      L3 = 0
      ELSE
      L3 = 4*MPOIN
      END IF
C
      DO  160  I = 1, MROW1
      READ (NTAPE) B(J), J = 1, MCOL1
      DO  140  J = 1, MROW2
      IF(Z(J,I).EQ.0.0) GO TO 140
      DUMMY = - Z(J,I)/B(I)
      DO  120  K = I+1, 4*MPOIN + MROW1
        L1 = K
      IF(B(L1).EQ.0.0) GO TO 120
```

```
         L2 = K
      IF(K.GT.MROW1) L2 = L3 + K
      Z(J,L2) = Z(J,L2) + DUMMY *B(L1)
  120 CONTINUE
C
      V2(J) = V2(J) + V1(I) *DUMMY
  140 CONTINUE
  160 CONTINUE
      RETURN
      END
      SUBROUTINE DECOMP (Z, B, MROW, MCOL)
C     ---------- ------ -----------------
C     THIS ONE IS USED FOR DECOMPOSING NON CONTACT ZONE EQUNS.
C
      DIMENSION Z(MROW,MCOL)
      DIMENSION B(1)
C
      DO  140  I = 1, MROW-1
      DO  140  J = I+1, MROW
      IF(Z(I,I).EQ.0.) WRITE(108,1000) I ; STOP
      IF(Z(J,I).EQ.0.0) GO TO 140
      DUMMY = - Z(J,I)/Z(I,I)
      DO  120  K = I+1, MCOL
C
       Z(J,K)  =  Z(J,K) + DUMMY * Z(I,K)
  120 CONTINUE
C
      B(J)  =  B(J) + DUMMY * B(I)
  140 CONTINUE
      RETURN
 1000  FORMAT(//,10X,'.ZERO ON DIAGONAL IN DECOMP *** I =',I5)
      END
      SUBROUTINE GAUSSQ (NFNODE, POSGP, WEIGP, SICOD)
C     ---------- ------ -------------------------
C     GAUSS POINTS AND WEIGHTING FACTORS : 2, 4 10 PT FORMULAE.
C
      DIMENSION POSGP(16,2), WEIGP(16,2), SICOD(4)
C--------------------------------------------- TWO POINT
      POSGP(1,1) = -0.577350269189626
      WEIGP(1,1) =  1.0
      POSGP(2,1) = -POSGP(1,1)
      WEIGP(2,1) =  WEIGP(1,1)
C--------------------------------- FOUR POINT
      POSGP(3,1) = -0.861136311594053
      POSGP(4,1) = -0.339981043584856
      WEIGP(3,1) =  0.347854845137454
      WEIGP(4,1) =  0.652145154862546
      POSGP(6,1) = -POSGP(3,1)
      POSGP(5,1) = -POSGP(4,1)
      WEIGP(6,1) =  WEIGP(3,1)
      WEIGP(5,1) =  WEIGP(4,1)
C--------------------------------- TEN POINT
```

```
      POSGP(7,1) = -0.973906528517172
      POSGP(8,1) = -0.865063366688985
      POSGP(9,1) = -0.679409568299024
      POSGP(10,1)= -0.433395394129247
      POSGP(11,1)= -0.148874338981631
      WEIGP(7,1) =  0.066671344308688
      WEIGP(8,1) =  0.149451394150581
      WEIGP(9,1) =  0.219086362515982
      WEIGP(10,1)=  0.269266719309996
      WEIGP(11,1)=  0.295524224714753
      DO 100 I = 1, 5
      POSGP(17-I,1)= -POSGP(I+6,1)
      WEIGP(17-I,1)=  WEIGP(I+6,1)
  100 CONTINUE
C------------------------------------------- TWO POINT : SINGULAR
       POSGP (1,2) = 0.11200820
       POSGP(2,2) =  0.60227690
       WEIGP (1,2) = 0.71853931
       WEIGP (2,2) = 0.28146068
C------------------------------------------- FOUR POINT
       POSGP (3,2) = 0.04133848
       POSGP (4,2) = 0.24527491
       POSGP (5,2) =  0.55616545
       POSGP (6,2) =  0.84898239
       WEIGP (3,2) = 0.38346406
       WEIGP (4,2) = 0.38687531
       WEIGP (5,2) = 0.19043512
       WEIGP (6,2) = 0.03922548
C------------------------------------------- TEN POINT
       POSGP (7,2) = 0.0090425944
       POSGP (8,2) = 0.053971054
       POSGP (9,2) = 0.13531134
       POSGP(10,2) = 0.24705169
       POSGP(11,2) = 0.38021171
       POSGP(12,2) = 0.52379159
       POSGP(13,2) = 0.66577472
       POSGP(14,2) = 0.79419019
       POSGP(15,2) = 0.89816102
       POSGP(16,2) = 0.96884798
       WEIGP (7,2) = 0.12095474
       WEIGP (8,2) = 0.18636310
       WEIGP (9,2) = 0.19566066
       WEIGP(10,2) = 0.17357723
       WEIGP(11,2) = 0.13569597
       WEIGP(12,2) = 0.093647084
       WEIGP(13,2) = 0.055787938
       WEIGP(14,2) = 0.027159893
       WEIGP(15,2) = 0.0095151992
       WEIGP(16,2) = 0.0016381586
C
C------------------------------------------- LOCAL COORD.: FUNCTIONAL NODES
       GO TO (200, 220, 240, 260) NFNODE
```

```
C
 200    SICOD(1) =0.
      GO TO 300
C
 220    SICOD(1) = -1.
      SICOD(2)   =  1.
      GO TO 300
C
 240    SICOD(1) = -1.
      SICOD(2)   =  0.
      SICOD(3)   =  1.
      GO TO 300
 260    SICOD(1) = -1.
      SICOD(2)   = -1./3.
      SICOD(3)   =  1./3.
      SICOD(4)   =  1.
C
 300   CONTINUE
      RETURN
      END
      SUBROUTINE   EVOLVE (S, NODES, IDERIV)
C
C***    THIS SUBROUTINE CALCULATES THE SHAPE FUNCTIONS AND THEIR DERIV
C***    DERIVATIVES FOR 1-DIMENSIONAL ELEMENT
C
      COMMON/EVOLV/ SHAPE(4),DERIV(4)
C
      GO TO (100, 120, 140, 160 ) NODES
C
C***    CONSTANT ELEMENT
C
 100    SHAPE(1) = 1.
      GO TO 180
C
C***    LINEAR ELEMENT
C
 120    SHAPE(1) = (1. -S)/2.
      SHAPE(2)   = (S +1.)/2.
      GO TO 180
C
C***    QUADRATIC ELEMENT
C
 140    SHAPE(1) = -S*(1. -S)/2.
      SHAPE(2)   =  (S+1)*(1-S)
      SHAPE(3)   =  S*(1. +S)/2.
      GO TO 180
C
C***    CUBIC ELEMENT
C
 160    SHAPE(1) =  -9.*(S -1.)*(S +1./3.)*(S -1./3.) /16.0
      SHAPE(2)   =  27.*(S +1.)*(S -1.)*(S -1./3.) /16.0
      SHAPE(3)   = -27.*(S +1.)*(S -1.)*(S +1./3.) /16.0
```

```
      SHAPE(4)    =   9.*(S +1./3.)*(S -1./3.)*(S +1.) /16.0
C
C***    CHECK IF DERIVATIVES ARE NEEDED
C
 180    IF(IDERIV.NE.1) GO TO  280
C
      GO TO (200, 220, 240, 260) NODES
C
 200    DERIV(1) =  0.
      GO TO  280
C
 220    DERIV(1) = -0.5
      DERIV(2)   = +0.5
      GO TO  280
C
 240    DERIV(1) = -0.5 +S
      DERIV(2)   = -2.0 *S
      DERIV(3)   = +0.5 +S
      GO TO  280
C
 260    DERIV(1) = - 9.*(3. *S**2 -2. *S -1./9.) /16.0
      DERIV(2)   =  27.*(3. *S**2 -2. *S/3. -1.) /16.0
      DERIV(3)   = -27.*(3. *S**2 +2. *S/3. -1.) /16.0
      DERIV(4)   =   9.*(3. *S**2 +2. *S -1./9.) /16.0
C
 280    CONTINUE
C
      RETURN
      END
      SUBROUTINE   JACOB1
C     ----------   ------
C***    THIS SUBROUTINE COMPUTES THE INTEGRATION DATA FOR THE ELEMENT
C***    UNDER CONSIDERATION. DATA IS GENERATED FOR ALL GAUSS POINTS.
C
      INCLUDE   COMMON, LIST
C------------------------------------------------------------ INITIALISE
      NG1 = 0
      IF(NGAUS.EQ.4) THEN
      NG1 = 2
      ELSE IF(NGAUS.EQ.10) THEN
      NG1 = 6
      END IF
      DO   100 IGAUS = 1, NGAUS
      DJACB(IGAUS) = 0.
      UNORM(IGAUS,1) = UNORM(IGAUS,2) = 0.
      GPCOD(IGAUS,1) = GPCOD(IGAUS,2) = 0.
      DO   100   INODE = 1, NFNODE
      GPSHAP(IGAUS,INODE) = 0.
 100       CONTINUE
C------------------------------------------------- LOOP OVER EACH GAUSS POINT
      DO   200   IGAUS = 1, NGAUS
C
```

```
C------------------------- GLOBAL COORDINATES OF THE GAUSS POINTS
      EXISP = POSGP(IGAUS+NG1,1)
C
      CALL   EVOLVE (EXISP, NBNODE, 1)
C
      DO   120   IDIME = 1, NDIME
      DO   120   INODE = 1, NBNODE
      GPCOD(IGAUS,IDIME) = GPCOD(IGAUS,IDIME) + SHAPE(INODE)*
     1                     ELBCOD(INODE,IDIME)
 120       CONTINUE
C
C------------------------------ COMPUTE THE NORM OF THE JACOBIAN
      DO   160   IDIME = 1,NDIME
      DO   140   INODE = 1, NBNODE
      UNORM(IGAUS,IDIME) = UNORM(IGAUS,IDIME) +DERIV(INODE)*
     1                     ELBCOD(INODE,IDIME)
 140       CONTINUE
      DJACB(IGAUS) = DJACB(IGAUS) + UNORM(IGAUS,IDIME)**2
 160       CONTINUE
      DJACB(IGAUS) = SQRT(DJACB(IGAUS))
C
C---------------------------- COMPUTE UNIT NORMAL AT THE GAUSS POINT
C
      DUMMY = UNORM(IGAUS,1)
      UNORM(IGAUS,1) = UNORM(IGAUS,2)/DJACB(IGAUS)
      UNORM(IGAUS,2) = -DUMMY/DJACB(IGAUS)
C
C--------------------- GET THE SHAPE FUNCTIONS AT THE GAUSS POINTS
C
      CALL   EVOLVE (EXISP, NFNODE, 0)
C
      DO   180   INODE = 1,NFNODE
      GPSHAP(IGAUS,INODE) = SHAPE(INODE)
 180       CONTINUE
C
 200        CONTINUE
C
       RETURN
      END
      SUBROUTINE   RING
C     ----------   ----
C------------------ THIS SUBROUTINE PERFORMS THR REGULAR INTEGRATION
C
      INCLUDE   COMMON, LIST
      DIMENSION  RDER(2), YBAR(4)
C------------------------------------------------------- INITIALISE
      NG1 = 0
      IF(NGAUS.EQ.4) THEN
      NG1 = 2
      ELSE IF(NGAUS.EQ.10) THEN
      NG1 = 6
      END IF
```

```
      DO    80    I = 1,2
      RDER(I) = 0.0
      DO    80    J = 1,2
      DO    80    K = 1, NFNODE
      EDISP(K,I,J) = 0.
      ESTRS(K,I,J) = 0.
      YBAR(K) = 0.
 80         CONTINUE
C
C------------------------- COMPUTE THE PROJECTION OF THE ELEM ON XBAR
      RLOCAL = 0.
      DO 300 I = 1, NDIME
      RLOCAL = RLOCAL + (ELBCOD(1,I) - ELBCOD(NBNODE,I))**2
 300  CONTINUE
C----------------------------------------- IF LINEAR ELEM THEN SKIP
      IF(ELBCOD(1,1).EQ.ELBCOD(NBNODE,1)) THETA = PI/2. ; GOTO 320
      ARG1  = ELBCOD(NBNODE,2)- ELBCOD(1,2)
      ARG2  = ELBCOD(NBNODE,1) - ELBCOD(1,1)
      THETA = ATAN2(ARG1,ARG2)
      YBAR(1) = 0.; YBAR(NBNODE) = 0.
 320  DO 340 I = 2, NBNODE-1
      YBAR(I) = (ELBCOD(I,1) - ELBCOD(1,1))*SIN(THETA)
     1        - (ELBCOD(I,2) - ELBCOD(1,2))*COS(THETA)
 340  CONTINUE
C
C---------------------------------------- LOOP OVER EACH GAUSS POINT
      DO    220    IGAUS = 1, NGAUS
      RDER(1) = RDER(2) = 0.
      RAD = 0.
      DRDN = 0.
      EXIWP = WEIGP(IGAUS+NG1,1)

C
C------------------------------- CALCULATE RADIUS AT EACH GAUSS POINT
      DO    100    IDIME = 1, NDIME
      RAD = RAD + (POLE(IDIME) - GPCOD(IGAUS,IDIME))**2
 100        CONTINUE
      RAD = SQRT(RAD)
      IF(RAD.GE.RMAX) RMAX = RAD
C----------------- CALCULATE THE DERIVATIVE AND THE NORMAL DERIVATIVE
C
      DO    120 IDIME = 1, NDIME
      IF(RAD.EQ.0.0) RDER(1)=COS(THETA); RDER(2)=SIN(THETA);GOTO 500
      RDER(IDIME) = (GPCOD(IGAUS,IDIME)- POLE(IDIME))/RAD
      DRDN = DRDN + UNORM(IGAUS,IDIME)*RDER(IDIME)
 120        CONTINUE
C
C--------------------------------- ELEMENT STRESSES AND DISPLACEMENTS
C
 500  DO    200    I = 1, 2
      DO    180    J = 1, 2
C
      IF(RAD.EQ.0.0) TIJ=0.0; UIJ=0.0 ; GO TO 520
```

```
      TIJ = -(DRDN*(2.*RDER(I)*RDER(J) + (1. -2.*RNU)*DELTA(I,J))
     1       -(1. -2.*RNU)*(RDER(I)*UNORM(IGAUS,J) - RDER(J)*
     2       UNORM(IGAUS,I)))/(4.*PI*RAD*(1. -RNU))
C
      UIJ =  (1. +RNU)*((3. -4.*RNU)*(ALOG(1/RAD))*DELTA(I,J) +
     1        RDER(I)*RDER(J))/(4.*PI*(1. -RNU))
C
C------------------- COMPUTE THE INTEGRATION FOR ALL FUNCTIONAL NODES
C
 520  DO   160   INODE = 1, NFNODE
C------------------------------------------------- CHECK FOR SINGULARITY
C
      IF(ISING.EQ.2.AND.NNSING.EQ.INODE)  GO TO 140
C
C----------------------------------------- PERFORM THE REGULAR INTGRN
C
      ESTRS(INODE,I,J) = ESTRS(INODE,I,J) +TIJ*GPSHAP(IGAUS,INODE)*
     1                   EXIWP*DJACB(IGAUS)
C
      EDISP(INODE,I,J) = EDISP(INODE,I,J) + UIJ*GPSHAP(IGAUS,INODE)*
     2                   EXIWP*DJACB(IGAUS)
       GO TO  160
C
C------------------------------------------------- SINGULARITY CALCULATION
C
 140    ESTRS(INODE,I,J) = 0.
C
C------------------------------------  CALCULATE YBAR AT THE GAUSS POINT
C
       CONST = 0.
      IF(NBNODE.EQ.2) CONST = 0. ; GO TO 420
      EXISP = POSGP(IGAUS+NG1,1)
      SIP = SICOD(NNSING)
      IF(EXISP.EQ.SIP) CONST= 0.0 ; GO TO 420
C
      CALL   EVOLVE (EXISP, NBNODE, 0)
C
      YGP = 0.
      DO  400 IN = 2, NBNODE-1
      YGP = YGP + SHAPE(IN)*YBAR(IN)
 400   CONTINUE
      CONST = ((YGP - YBAR(NNSING))/(EXISP- SIP))**2
 420   CONTINUE
C
C------------------------------------------------------ CALCULATE EDISP
C
      EDISP(INODE,I,J) = EDISP(INODE,I,J) - (1. +RNU)*0.5*
     1             (3.-4.*RNU)*(ALOG(RLOCAL/4. + CONST))*DELTA(I,J)
     2               *GPSHAP(IGAUS,INODE)* EXIWP *DJACB(IGAUS)
     3               /(4.*PI*(1. -RNU))
      EDISP(INODE,I,J) = EDISP(INODE,I,J) + (1. +RNU)*RDER(I)*RDER(J)
     2               *GPSHAP(IGAUS,INODE)* EXIWP *DJACB(IGAUS)
```

```
     3                /(4.*PI*(1. -RNU))
C
 160      CONTINUE
 180       CONTINUE
 200        CONTINUE
 220         CONTINUE
C
          RETURN
        END
        SUBROUTINE    ETERNAL
C       ----------    -------
C--------------------- THIS SUBROUTINE COMPUTES THE SINGULAR INTEGRAL
C
        INCLUDE   COMMON, LIST
        DIMENSION GCOD(2)
C------------------------------------------------------------ INITIALISE
        EDSING(1,1) = EDSING(2,2) = 0.
        EDSING(1,2) = EDSING(2,1) = 0.
        GCOD(1) = GCOD(2) = 0.
        NG1 = 0
        IF(NGAUS.EQ.4) THEN
        NG1 = 2
        ELSE IF(NGAUS.EQ.10) THEN
        NG1 = 6
        END IF
C------------------------- DETERMINE THE SUBDIVISIONS OF THE INTEGRAL
C       SINCE THE INTEGRAL IS PERFORMED OVER THE INTERVAL 0:1,
C       THE INTEGRATION IS PERFORMED IN PARTS FOR QUAD AND CUBIC ELEMS.
C
        ICOUNT = 2
        IF(NNSING.EQ.1.AND.NFNODE.NE.1) ICOUNT =1 ; GO TO  100
        IF(NNSING.EQ.NFNODE.AND.NFNODE.NE.1) ICOUNT = 1
C
 100      DO   260   IC = 1, ICOUNT
        IF(NFNODE.NE.1) GO TO 120
        SI1 = 0.
        SI2 = 1.
        IF(IC.EQ.2) SI2 = -1.
        GO TO   140
 120      SI1 = SICOD(NNSING)
        SI2   = +1.
        IF(IC.EQ.2) SI2 = -1.
        IF(NNSING.EQ.NFNODE) SI2 = -1.
C
C----------------------- COMPUTE THE COEFF'S FOR LINEAR TRANSFORMATION
C                       THE EQUATION IS            ETA = A*SI + B
C
 140      ETA1 = 0.
        ETA2 = 1.
        A = (ETA1 - ETA2)/(SI1 - SI2)
        A1 = -2./(SI1 - SI2)
        B1 = (SI1 + SI2)/(SI1 - SI2)
```

```
      IF(SI2.EQ.-1.) A1 = -A1 ; B1 = -B1
      JCOUNT = 2
      SING = SING1 = 0.
C     IF(NFNODE.EQ.1) JCOUNT = 1
      B = (SI1*ETA2 - SI2*ETA1)/(SI1 - SI2)
C
C------------------------------------------ LOOP OVER EACH GAUSS POINT
      DO 250  JC = 1, JCOUNT
      DO   240   IGAUS = 1, NGAUS
      IF(JC.EQ.2) GO TO 400
      EXISP = (POSGP(IGAUS+NG1,2) - B)/A
      EXIWP = WEIGP(IGAUS+NG1,2)
      GO TO 420
C
C------------------------------------------  CHANGE THE EXISP AND WEIGP
C
 400   EXISP = (POSGP(IGAUS+NG1,1) - B1)/A1
       EXIWP = WEIGP(IGAUS+NG1,1)
 420    CONTINUE
C----------------------- GET THE GLOBAL COORDINATE OF THE GAUSS POINT
C
      CALL   EVOLVE (EXISP, NBNODE, 1)
C
      DO     160   IDIME = 1, NDIME
      DO     160   INODE = 1, NBNODE
      GCOD(IDIME) = GCOD(IDIME) + SHAPE(INODE)*ELBCOD(INODE,IDIME)
 160        CONTINUE
C------------------------------------- COMPUTE THE NORM OF THE JACOBIAN
C
      SJNORM = 0.
      DO   220   IDIME = 1, NDIME
      DUMMY = 0.
      DO   200   INODE = 1, NBNODE
      DUMMY = DUMMY + DERIV(INODE)*ELBCOD(INODE,IDIME)
 200       CONTINUE
C------------------------------------- DIVIDE BY A, (A = D(ETA)/D(SI))
C
      SJNORM = SJNORM +  (DUMMY)**2
 220       CONTINUE
       DETA = ABS(1./A)
       IF(JC.EQ.2) DETA = 1./A1
      SJNORM = SQRT(SJNORM)*DETA
C
C------------------------------------------   COMPUTE THE SHAPE FUNCTION
      CALL   EVOLVE (EXISP, NFNODE, 0)
      SHP = SHAPE(NNSING)
C
C-------------------------------------------------   COMPUTE THE INTEGRAL
C
      IF(JC.EQ.1) GO TO 460
      SING1 = SING1 + ALOG(ABS(A1/2.))*SJNORM*SHP*EXIWP
      GO TO  240
```

```
 460   SING = SING + SJNORM*EXIWP*SHP
 240        CONTINUE
 250     CONTINUE
C
         SING = (SING + SING1)*(1. +RNU)*(3. -4.*RNU)/(4.*PI*(1.-RNU))
       EDSING (1,1) = EDSING(1,1) + SING
       EDSING(2,2)  = EDSING(2,2) + SING
 260        CONTINUE
C
         RETURN
       END
         SUBROUTINE EQUN ( UNION, IBP, IDIFF, ICFLAG, MCOL)
C        ---------- ----   -----  ---  -----  ------  ----
C***       FOR A GIVEN POLE THIS  SUBROUTINE FORMS THE TWO INTEGRAL EQUNS.
C
         INCLUDE COMMON, LIST
C
       DIMENSION  UNION(2, NTSIZE)
       DIMENSION BC(4), ELEMBC(4,4), IBC(4), IECODE(4,4)
C
        DO  300  I = 1,2
        DO  300  J = 1, MCOL
       UNION(I,J)  = 0.0
 300    CONTINUE
        DO  320  I = 1, 2
       ELRHS(I) = 0.
 320    CONTINUE
       INEQUN  =  (NFP(IBP) - MCPOIN)*2
       ND  =  0
       KN1 = KE1 =0
       DO  110  I  = 1, IBP-1
       ND = ND + NFP(I)
       KN1 = KN1 + NBP(I)
       KE1 = KE1 + NEL(I)
 110   CONTINUE
       IF(IBP.GT.1.AND.ICFLAG.EQ.1) INEQUN = INEQUN + 4*MCPOIN
C
C---------------  LOOP OVER EACH REGION OF THE BODY UNDER INTEGRATION
C
        DO  900  KREGN  = 1,  NREGN
C
C-------------------------- LOOP OVER ALL THE ELEMENTS OF THE REGION
C
        DO  800  KELEM  = REGN(IBP, KREGN, 1,1) ,REGN(IBP,KREGN, 1,2)
C
C----------------------------     ESTABLISH THE GEOMETRY OF THE ELEMENT
C
        DO  140  KNODE  = 1, NBNODE
       KN  = (KELEM -KE1-1)*(NBNODE -1) + KNODE +KN1
       IF(KELEM.NE.REGN(IBP,KREGN,1,2))  GO  TO  120
       IF(KNODE.NE.NBNODE)  GO  TO  120
       KN  = REGN(IBP, KREGN, 3,1)
```

```
 120   ELBCOD(KNODE ,1) = BNODE(KN, 1)
      ELBCOD(KNODE, 2)  = BNODE(KN, 2)
 140   CONTINUE
C----------------------------------- SKIP THE ELEMENT WITH ZERO LENGTH
C----------------------------------- DOUBLE NODE : ONLY LINEAR ELEMENT
      EL = 0.0
      DO 150 I = 1, 2
      EL = EL + (ELBCOD(1,I) - ELBCOD(NBNODE,I))**2
 150  CONTINUE
      IF(EL.EQ.0.0) GO TO 800
C------------------------------------- STORE THE BOUNDARY CONDITIONS
C
       DO  160  KNODE  = 1, NFNODE
      CALL  BCONDN (KELEM, KNODE, BC, IBC)
       DO  160  K  = 1, 4
      IECODE(KNODE, K) = IBC(K)
      ELEMBC(KNODE, K) =  BC(K)
 160   CONTINUE
C-------------------------- STORE THE GLOBAL FUNCTIONAL NODE NUMBERS
C
       DO  200  KNODE  = 1, NFNODE
      KN  = (KELEM -1)*(NFNODE -1) + KNODE
      IF(NFNODE.EQ.1) KN = KELEM ;  GO  TO  180
      IF(KELEM.NE.REGN(IBP, KREGN, 1,2))  GO TO 180
      IF(KNODE.NE.NFNODE) GO TO  180
      KN  = REGN(IBP, KREGN, 2,1)
 180   NODEF(KNODE)  = KN
 200   CONTINUE
C------------------------------------------- ORDER OF INTEGRATION
      RAD = EL =0.0
      DO  210  I = 1, NBNODE
      R1 = EL = 0.0
      DO  205  J = 1, NDIME
      EL = EL +(ELBCOD(1,J) -ELBCOD(NBNODE,J))**2
      R1 = R1 + (ELBCOD(I,J) - POLE(J))**2
 205  CONTINUE
      IF(I.EQ.1) RAD = R1
      IF(R1.LE.RAD) RAD = R1
 210  CONTINUE
      R1 = SQRT(RAD/EL)
      IF(R1.LE.1.50) THEN
      NGAUS = 10
      ELSE IF(R1.LE.5.0) THEN
      NGAUS = 4
      ELSE
      NGAUS = 2
      END IF
C------------------------------------------ COMPUTE THE INTEGRATION DATA
       CALL JACOB1
C------------------------------------------- CHECK FOR SINGULARITY
C-------------------------- IF SINGULAR : ISING = 2 ; NNSING = NODE
       ISING = 1
```

```
      DO  220  KNODE  = 1, NFNODE
      IF(NPOLE.NE.NODEF(KNODE))  GO  TO  220
      NNSING = KNODE
C-----NESING = KELEM
      ISING  = 2
      GO  TO  240
 220  CONTINUE
 240   CONTINUE
C
      CALL   RING
C
      IF (ISING.EQ.1)  GO TO  280
C
      CALL   ETERNAL
C
      DO  260  I = 1, 2
      DO  260  J = 1, 2
      EDISP(NNSING, I,J) = EDISP(NNSING, I,J) + EDSING(I,J)
 260  CONTINUE
 280  CONTINUE
C
C--------------     EQUATION ASSEMBLY    -------------------------------
C
C     FOR EACH NODE FIND KDIFF. CHECK IF IT IS A CONTACT NODE.
C     DIAGONAL ELEMENT ADD ALL STRESS COEFFICIENTS. RIGID BODY.....
C     IF CONTACT NODE ALL FOUR UNKNOWNS. STRESS DISCONTINUITY
C     AT THE END CONTACT NODE. HOW TO HANDLE THAT WITH RELEASED NODE
C     REMAINS A QUESTION.
C-----------------------------------------------------------------------
C
      DO  600  KNODE  = 1, NFNODE
      NODE = NODEF(KNODE)
C
      KCFLAG = 0 ; KDIFF =0
      DO  340  K = 1, MTCP
      IF(MPAIR(IBP,K).EQ.0) GO TO 340
      IF(NODE.GE.MPAIR(IBP,K)) KDIFF = KDIFF + 1
      IF(NODE.EQ.MPAIR(IBP,K)) KCFLAG = 1
 340  CONTINUE
C
       DO  500  IR  =  1,  2
       DO  500  JC  =  1,  2
      IF(KCFLAG.EQ.1) THEN
       JCOL = INEQUN + (KDIFF -1)*4 + JC
        IF(IECODE(KNODE, JC).EQ.1) THEN
        UNION(IR, JCOL) = UNION(IR, JCOL) - EDISP(KNODE, IR, JC)
        END IF
          IF(IECODE(KNODE, JC +2).EQ.1) THEN
          UNION(IR, JCOL +2) = UNION(IR, JCOL+2) +ESTRS(KNODE,IR,JC)
          END IF
      ELSE
C
```

```
C--------------  NON CONTACT NODES  -------------------------
C
      JCOL = (NODE -ND -KDIFF -1)*2 + JC
        IF(IECODE(KNODE, JC).EQ.1) THEN
      UNION(IR, JCOL) = UNION(IR, JCOL) - EDISP(KNODE, IR, JC)
        ELSE
      ELRHS(IR) = ELRHS(IR) + ELEMBC(KNODE, JC)*EDISP(KNODE, IR, JC)
        END IF
C----------------------------  STORE THE COEFF CORRESPONDING TO DISP.
C
        IF(IECODE(KNODE, JC + 2).EQ.1) THEN
      UNION(IR, JCOL) = UNION(IR,JCOL) + ESTRS(KNODE, IR, JC)
        ELSE
     ELRHS(IR) = ELRHS(IR) -ELEMBC(KNODE,JC +2)*ESTRS(KNODE,IR, JC)
       END IF
      END IF
C------------------  DIAGOANAL ELEMENT. USING RBM TO THE SINGULAR TERM
C
     KDIAG = NPOLE
     IF(ICFLAG.EQ.1) THEN
      JCROW = INEQUN + (IDIFF -1)*4 + (JC +2)
     ELSE
      JCROW = (KDIAG - IDIFF -ND - 1)*2 + JC
     END IF
C
C---------------------  CHANGE THE SIGN OF ESTRS (SUBTRACT FROM UNION!)
C
     IF(IPCODE(JC +2).EQ.1) THEN
      UNION(IR, JCROW) = UNION(IR, JCROW) - ESTRS(KNODE, IR, JC)
     ELSE
      ELRHS(IR) = ELRHS(IR) + POLEBC(JC +2) *ESTRS(KNODE, IR, JC)
     END IF
C
C------------------------------  COMPLETE THE LOOPS FOR IR, JC, KNODE
C
 400    CONTINUE
 500     CONTINUE
 600      CONTINUE
C
C--------------------------------  COMPLETE THE LOOPS FOR KREGN, KELEM
C
 800       CONTINUE
 900        CONTINUE
C
                  RETURN
                  END
      SUBROUTINE DIRAC (IBP)
C     ---------- ----- -----
C---------------------------  COMPUTES THE POINT LOAD INFLUENCE.
      INCLUDE COMMON
      DIMENSION RDER(2)
C
```

```
      N= 0
      IF(IBP.EQ.2) N = NPLOAD(1)
      DO  160  IL = 1, NPLOAD(IBP)
      RAD = 0.0
      RDER(1) = RDER(2) = 0.0
      DO  100  IDIME = 1, NDIME
      RAD = RAD + (POLE(IDIME) - PTLOAD(IL+N, IDIME))**2
 100  CONTINUE
      RAD = SQRT(RAD)
      IF(RAD.EQ.0.0) THEN
      WRITE (108, 1000) POLE(1), POLE(2), (PTLOAD(IL+N,J), J=1,4)
      GO TO 160
      END IF
      DO  120  IDIME = 1, NDIME
      RDER(IDIME) = (PTLOAD(IL+N, IDIME) - POLE(IDIME))/RAD
 120  CONTINUE
C
      DO  140  I = 1, 2
      DO  140  J = 1, 2
      UIJ = (1. +RNU)*((3. -4.*RNU)*(ALOG(1./RAD))*DELTA(I,J) +
     1      RDER(I)*RDER(J))/(4.*PI*(1. -RNU))
      ELRHS(I) = ELRHS(I) + UIJ* PTLOAD(IL+N, J+2)
 140  CONTINUE
 160  CONTINUE
C
 1000 FORMAT(/5X,'FROM SUBROUTINE DIRAC: RADIUS IS ZERO FOR THESE',
     1' VALUES OF POLE AND PTLOAD',/5X,2(2X,E12.5),//5X,
     2 4(2X,E12.5)//)
C
      RETURN
      END
      SUBROUTINE   EXTORT (B,P,Q,N,NDIM,NSTART, NSAVE)
C     ----------   ------ ----------------------------
C---------------------------------------------------------------------
C     GAUSS ELIMINATION. THE EQUATION IS [B] (P) = (Q)
C     ANSWERS ARE STORED IN (P). AND (Q) IS NOT SAVED.
C     N = NO. OF UNKNOWNS ; NDIM = ACTUAL DIMENSION OF B AND Q
C     NSTART = FORWARD ELIMINATE IS DONE FROM THIS EQUN. NORMALLY NS=2
C     NSAVE = 0 => Q IS NOT SAVED. NSAVE = 1 => SAVE Q.
C     NSAVE > 1  ==> ANSWERS ARE STORED IN Q
C     THE THREE DIFFERNT OPTIONS FOR NSAVE ARE USED AT DIFFERENT
C     LEVELS. REFER TO SUBROUTINE REFINE. NSAVE > 1 IS USED FOR
C     2-D STRESS ANALYSIS.
C
C     NSTART IS ALSO USED AT DEIFFERNT LEVELS. SEE: REFINE
C---------------------------------------------------------------------
      DIMENSION B(NDIM,NDIM),  Q(NDIM), P(NDIM)
C
C     ENORM = 0.0
C     DO 30 I = 1, N
C     DO 30 J = 1, N
C30   ENORM = ENORM + B(I,J)**2
```

```
C
      IF(NSAVE.GT.0) THEN
      DO  40  I = 1, N
 40   P(I) = Q(I)
      END IF
C
C***  REDUCE B TO UPPER TRAINGULAR
C
      DO 140 I = 1, N-1
      N1 = NSTART
      IF(I.GE.NSTART) N1 = I+1
      DO 140 J = N1 , N
      IF(B(I,I).EQ.0) WRITE(108,1000) I ; STOP
      IF(B(J,I).EQ.0) GO TO 140
      BDUMMY = -B(J,I)/B(I,I)
      DO 120 K = I+1, N
      B(J,K) = B(J,K) + BDUMMY*B(I,K)
  120 CONTINUE
C
      IF(NSAVE.EQ.0) THEN
      Q(J) = Q(J) + BDUMMY*Q(I)
      ELSE
      P(J) =  P(J) + BDUMMY*P(I)
      END IF
  140 CONTINUE
C
C***  CALCULATE THE DETERMINANT AND THE CONDITION NUMBER
C     THE DETERRMINANT CAN BE CALCULATED FROM DIAGONALS AFTER
C     FORWARD REDUCTION.
C
C     DET = 1. ;  CONDN = 0.0
C     DO 500 I = 1, N
C     DET = DET*B(I,I)
C500  CONTINUE
C     CONDN = DET/ SQRT(ENORM)
C     WRITE(108,1020)DET,SQRT(ENORM),CONDN
C
C***  BACK SUBSTITUTION
C
      IF(NSAVE.EQ.0) THEN
      P(N) = Q(N)/B(N,N)
      ELSE
      P(N) = P(N)/B(N,N)
      END IF
C
      DO 180 I = N-1,1,-1
      IF(NSAVE.EQ.0) THEN
      P(I) = Q(I)
      END IF
      DO 160 J = I+1, N
      P(I)= P(I) - B(I,J)*P(J)
  160 CONTINUE
```

```
      P(I) = P(I)/B(I,I)
  180 CONTINUE
C
      IF(NSAVE.GT.1) THEN
      DO 200 I = 1, N
 200  Q(I) = P(I)
      END IF
C
 1000 FORMAT(//10X,'ZERO ON THE DIAGONAL. PROGRAM HALTED IN GEL'/10X,
     1 'ROW NO = 'I5)
C1020 FORMAT(/5X,'NORM FROM GEL',/5X,'DETERMINANT ='E10.3,3X,
C    1'EUCLIDIAN NORM ='E10.3,3X,'CONDITION NO. ='E10.3,//)
      RETURN
      END
      SUBROUTINE NORM (A, X, B, N)
C     ---------- ---- ------------
C     CALCULATE THE EUCLEADIAN AND MAXM NORMS
C
      DIMENSION  A(N,N), X(N), B(N)
C
      BIGA = ANORM = 0.0
      BIGB = BNORM = 0.0
      DO 150 I = 1, N
      X(I) = 0.0
      DO 110 J = 1, N
      IF(ABS(A(I,J)).GT.BIGA) THEN
      BIGA = ABS(A(I,J))
      IA = I
      JA = J
      END IF
      IF(I.EQ.J) GO TO 100
      X(I) = X(I) + ABS(A(I,J)/A(I,I))
 100  ANORM = ANORM + A(I,J)**2
 110  CONTINUE
      IF(ABS(B(I)).GT.BIGB) THEN
      BIGB = ABS(B(I))
      IB = I
      END IF
      BNORM = B(I)**2 + BNORM
 150  CONTINUE
C
      WRITE(108, 1000)
      WRITE(108,1020) A(I,I), I= 1, N
      WRITE(108,1040)
      WRITE(108,1020) X(I), I= 1, N
      WRITE(108,1060) SQRT(ANORM), SQRT(BNORM)
      WRITE(108,1080) BIGA,IA,JA, BIGB,IB
1000  FORMAT('1',//5X,'THE DIAGONAL TERMS OF THE COEFF. MATRIX ARE'//)
1020  FORMAT(8(2X,E10.3)/)
1040  FORMAT(//,5X,'ROWWISE ABSOLUTE SUM OF THE NORMALISED OFF',
     1' DIAGONAL TERMS'//)
1060  FORMAT(/5X,'EUCLIDEAN NORM OF THE COEFF. MATRIX ='E10.3,/5X,
```

```
     1'EUCLIDIAN NORM OF THE RHS ='E10.3,/)
1080  FORMAT(/5X,'MAXM. NORM OF COEFF.MATRIX ='E10.3,2X,'ROW NO.'I3,
     12X,'COL NO.'I3,/5X,'MAXM. NORM OF RHS ='E10.3,2X,'COL NO.'I3)
      RETURN
      END
```