ELASTOSTATIC INTERACTION OF MULTIPLE ARBITRARILY SHAPED CRACKS IN PLANE INHOMOGENEOUS REGIONS

by

VASANTHA MOHAN NARENDRAN

S.B., Massachusetts Institute of Technology
(1981)

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE IN
MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1982

Signature of Author _____
Department of Mechanical Engineering
May 14, 1982

Certified by _____
Michael P. Cleary
Thesis Supervisor

Accepted by _____
Warren M. Rohsenow
Archives Graduate Registration Officer

ELASTOSTATIC INTERACTION OF MULTIPLE ARBITRARILY SHAPED CRACKS IN PLANE INHOMOGENEOUS REGIONS

by

VASANTHA MOHAN NARENDRAN

Submitted to the Department of Mechanical Engineering
on May 14, 1982 in partial fulfillment of the
requirements for the Degree of Master of Science in
Mechanical Engineering

ABSTRACT

A numerical technique has been developed for the determination of stress fields associated with multiple arbitrarily shaped cracks in plane inhomogeneous regions. The procedure allows the elastostatic analysis of cracks interacting with one or more straight bimaterial interfaces; of cracks located near, or emanating from, circular inclusions; and of cracks that emanate from single or multiple origins. The cracks may be branched or blunted, and may be subjected to arbitrarily applied stresses. The technique employs an efficient surface integral method, using distributions of edge dislocations to represent the cracks. The resulting singular integral equations are solved using a Gauss-Chebyshev integration formula; appropriate conditions are developed for closing the set of equations governing cracks intersecting inhomogeneity boundaries, based on a consideration of the stresses and displacements at the points of intersections. Crack-tip stress intensity factor results are presented for several crack configurations. The overall scheme provides a more general, direct, and convenient approach than other available schemes. A computer program has been developed to implement the various formulations in a single framework.

Thesis Supervisor: Michael P. Cleary

Title: Associate Professor of
Mechanical Engineering

ACKNOWLEDGEMENTS

The successful completion of this thesis has required more than a personal effort, and I am thankful to many individuals for their contributions. The continual support and supervision provided by Professor Michael P. Cleary has contributed in no small way towards the smooth progress of the research, and I have also gained from his efforts to share his ideas and interests in the area of resource extraction. I am grateful.

Special thanks are also due to Dave Barr, Stephen Wong, Don Petersen, Khin Yong Lam, Jim Papadopoulos and Dr. Zekai Celep for the many useful suggestions and discussions. Financial support for the project, which came from the MIT UFRAC program supported by several production and service companies in the energy industry, is gratefully acknowledged.

A final note of thanks goes to my parents and sisters. I deeply appreciate their moral support during these past two years of graduate school, and want to thank my parents for their prayers, wisdom and understanding.

TABLE OF CONTENTS

INTRODUCTION

Although plane elastostatic problems have been the subject of considerable research in fracture mechanics, they have been primarily limited to either rectilinear cracks in homogeneous and inhomogeneous media (for example, Barr and Cleary [1], Erdogan and coworkers [2-6], Segall and Pollard [7]), or to only slightly curved or kinked single cracks in the infinite plane (Banichuk [8], Goldstein and Salganik [9,10], Cotterell and Rice [11]). These solutions are clearly inadequate for the more general study of the initiation and propagation of curvilinear cracks, inasmuch as even initially rectilinear cracks will grow on arbitrarily shaped trajectories, dictated by non-uniform stress fields; these stresses may be due to asymmetric applied loading, or may arise from the interaction of multiple cracks with one another and with phase boundaries, material interfaces, etc.

This study describes a general-purpose numerical scheme for analyzing the elastostatic crack-induced stress fields for arbitrarily shaped multiple cracks in plane inhomogeneous regions. The model employs a surface integral method, based on the highly efficient and physically realistic insertion of mathematically equivalent dislocation distributions to represent crack openings. Besides computational efficiency (e.g., as compared to finite element

methods), the approach in developing the formulation presented here is more direct than that of previous researchers: it affords remarkable simplicity in the determination of the stress intensity factors and crack opening displacements. The general methodology is outlined in the first chapter; it is then adapted in Chapter 2 to get more specific formulations for each of the crack configurations considered as sample applications. Some interesting results are also presented in the second chapter; these are compared with existing results, for the special cases where those are available in the literature. The much more general uses of the formulation and computer programs are then described, especially as they have been implemented by us in the context of hydraulic fracture simulation.

CHAPTER 1

FORMULATION AND NUMERICAL SCHEMES FOR PLANE ELASTOSTATIC ANALYSIS

This section describes the formulation and numerical schemes used to solve the displacement discontinuity problems associated with quasi-static curvilinear cracks in an elastic solid. The formulation allows the determination of the stress fields and crack shapes of arbitrarily curved single or multiple cracks. The cracks may be in an infinite medium, or may interact with either one or two material interfaces. As special limiting cases, the cracks may be located near a free boundary or in an infinite strip. The cracks may also be located near a circular inclusion or emanate from the inclusion. Finally, they may emanate from single or multiple origins; the configuration may also include branch cracks at the tips of the main cracks. The crack surfaces may be subjected to arbitrarily applied pre-existing stresses and internal pressures.

General Formulation

Consider an elastic solid containing several curvilinear cracks subject to an arbitrary equilibrium state of stress as shown in Fig. 1.1a. Since this elastic problem is linear, the solution of the problems of traction-free cracks under the given set of external loads can be expressed as

the sum of two solutions: the first obtained for the given external loads and the given medium without the cracks (Fig. 1.1b), and the second obtained for the cracked medium where the only external loads are the change in tractions on the crack surfaces (Fig. 1.1c). These tractions are equal and opposite to the stresses found in the first solution on the presumed location of the cracks. Since the first problem preserves the continuity of displacement at all points, it is clear that only the second solution, which gives the perturbed stress fields, will have singularities. Thus, the net stresses acting to cause the crack displacement discontinuity are those shown in Fig. 1.1c, namely $p\underset{\sim}{n}-\underset{\sim}{t}^0(\underset{\sim}{x})$ applied directly to the crack faces, where $p(\underset{\sim}{x})$ is the fluid pressure induced by the fluid being pumped into the opening fracture. The problem is thereby reduced to a study of the crack locus (approximately embedded in the surrounding inhomogeneous medium), provided we can estimate $\underset{\sim}{t}^0(\underset{\sim}{x})$ from a standard continuum analysis.

The cracks are represented by continuous distributions of edge dislocation singularities, which have previously been used to analyze rectilinear crack configurations [1]. The dislocations are distributed in such a fashion so as to satisfy the traction conditions on the crack faces. If $\delta'_{\beta n}(\underset{\sim}{x})$ represents the β-th component of dislocation density at the point $\underset{\sim}{x}$ on the n-th surface S_n, the strength of the dislocation over an infinitesimal segment dS_n of the n-th surface is $db_{\beta n} = \delta'_{\beta n}(\underset{\sim}{x})\cdot dS_n$. The α-th traction

the sum of two solutions: the first obtained for the given external loads and the given medium without the cracks (Fig. 1.1b), and the second obtained for the cracked medium where the only external loads are the change in tractions on the crack surfaces (Fig. 1.1c). These tractions are equal and opposite to the stresses found in the first solution on the presumed location of the cracks. Since the first problem preserves the continuity of displacement at all points, it is clear that only the second solution, which gives the perturbed stress fields, will have singularities. Thus, the net stresses acting to cause the crack displacement discontinuity are those shown in Fig. 1.1c, namely $p\underset{\sim}{n}-\underset{\sim}{t}^0(\underset{\sim}{x})$ applied directly to the crack faces, where $p(\underset{\sim}{x})$ is the fluid pressure induced by the fluid being pumped into the opening fracture. The problem is thereby reduced to a study of the crack locus (approximately embedded in the surrounding inhomogeneous medium), provided we can estimate $\underset{\sim}{t}^0(\underset{\sim}{x})$ from a standard continuum analysis.

The cracks are represented by continuous distributions of edge dislocation singularities, which have previously been used to analyze rectilinear crack configurations [1]. The dislocations are distributed in such a fashion so as to satisfy the traction conditions on the crack faces. If $\delta'_{\beta n}(\underset{\sim}{x})$ represents the β-th component of dislocation density at the point $\underset{\sim}{x}$ on the n-th surface S_n, the strength of the dislocation over an infinitesimal segment dS_n of the n-th surface is $db_{\beta n} = \delta'_{\beta n}(\underset{\sim}{x})\cdot dS_n$. The α-th traction

component at point $\underset{\sim}{x}_o$ on the m-th surface is obtained by integrating over all the crack surfaces in the configuration,

$$\sigma_{\alpha m}(\underset{\sim}{x}_o) = \sum_{\beta,n} \int_{S_n} \Gamma^{\beta n}_{\alpha m}(\underset{\sim}{x}_o, \underset{\sim}{x}) \delta'_{\beta n}(\underset{\sim}{x}) dS_n \qquad (1.1)$$

where the influence functions $\Gamma^{\beta n}_{\alpha m}$ constitute the traction component at $\underset{\sim}{x}_o$ due to a concentrated density $\delta'_{\beta n}$ at $\underset{\sim}{x}$. These fundamental solutions are derived for a geometry and linear material response as close as possible to the crack configuration being studied. A comprehensive listing of available fundamental solutions has been provided by Cleary (see Hanson et al. [12]) for both plane strain and fully three-dimensional crack simulation; further approximate influence functions for an edge dislocation in an isotropic layer of finite width bonded between two isotropic half-planes have also been developed (see Wong [13]).

For a system of N cracks in the plane (Fig. 1.2), the hitherto general 3-D equation (1.1) can be specialized to the form

$$\sigma_\alpha(\xi_m) = \sum_{n=1}^{N} \sum_{\beta=1}^{2} \int_{-1}^{1} ds_n \, \delta'_\beta(s_n) \Gamma^\beta_\alpha(\xi_m, s_n) \qquad (1.2)$$

which gives the stresses σ_α at point ξ_m (with associated unit normal to surface S_m) due to the normalized unit

dislocation ($b_\beta/L_n=1$) in the β-direction located at point s_n. The dimensionless curvilinear coordinates s_n and ξ_m are defined on the interval (-1,+1), and are obtained from a transformation of the dimensional Cartesian coordinates as detailed in Appendix A. The curvilinear length of the n-th crack is L_n, while β takes on values 1,2 to signify Burgers vectors b_β in the x- and y-directions, respectively. In general, the tractions on the crack surfaces are due to the stresses induced by the boundary conditions and by any fluid pressure in the crack. If these tractions are specified, the dislocation densities can be determined by requiring that

$$\begin{Bmatrix} \sigma_\nu(\xi_m) \\ \\ \sigma_\tau(\xi_m) \end{Bmatrix} = \sum_{n=1}^{N} \sum_{\beta=1}^{2} \int_{-1}^{1} ds_n \delta'_\beta(s_n) \begin{Bmatrix} \Gamma_\nu^\beta(\xi_m, s_n) \\ \\ \Gamma_\tau^\beta(\xi_m, s_n) \end{Bmatrix} \quad (1.3)$$

where $\sigma_\nu(\xi_m)$ and $\sigma_\tau(\xi_m)$ are the specified normal and shear tractions, respectively. Since these tractions act on the plane of the crack surface at ξ_m, the normal and shear influence functions for a global coordinate system must be transformed to the local coordinates (using $i=\sqrt{-1}$) by

$$\begin{Bmatrix} \Gamma_\nu^\beta(\xi_m, s_n) \\ \\ \Gamma_\tau^\beta(\xi_m, s_n) \end{Bmatrix} = \begin{Bmatrix} \mathrm{Re} \\ \mathrm{Im} \end{Bmatrix} [\Gamma_H^\beta(\xi_m, s_n) + e^{-2i\psi_m} \Gamma_D^\beta(\xi_m, s_n)] \quad (1.4a)$$

where

$$\Gamma_H^\beta(\xi,s) = 1/2[\Gamma_{xx}^\beta(\xi,s) + \Gamma_{yy}^\beta(\xi,s)] \tag{1.4b}$$

$$\Gamma_D^\beta(\xi,s) = 1/2[\Gamma_{xx}^\beta(\xi,s) - \Gamma_{yy}^\beta(\xi,s)] + i\Gamma_{xy}^\beta(\xi,s) \tag{1.4c}$$

When the relevant influence functions, given in Appendix B, are written in terms of ξ_m and S_n, they are all found to be of the form

$$\pi\Gamma_\alpha^\beta(\xi_m,s_n) = \bar{E}g_{\alpha m}^{\beta n}/(\xi_m-s_n) + \bar{E}h_\alpha^\beta(\xi_m,s_n) \tag{1.5}$$

where $\bar{E}$ ($\equiv E/4(1-\nu^2)$) will be recognized as the "crack-opening modulus"; the latter is derived from Young's modulus E and Poisson's ratio ν for an isotropic medium but corresponding forms apply also for an anisotropic medium (see references by Cleary [14]). Here $g_{\alpha m}^{\beta n}$ is zero for $m\neq n$ and is nonsingular even when m=n; $h_\alpha^\beta(\xi_m,s_n)$ is nonsingular unless m=n and ξ,s both lie at a free surface. However, the overall influence functions are always singular when $\xi_m=s_n$, and the resulting singular integral equations (1.3) must usually be solved numerically except for some special cases.

Numerical Implementation

The various numerical schemes that can be employed to solve these equations have already been described elsewhere

(e.g., Wong [13], Erdogan and Gupta [15]). The numerical scheme adopted in this study is based on the global interpolation method [13,15], whereby the unknown dislocation density function is discretized by choosing the interpolation functions

$$\delta'_{\beta}(s) = f_{\beta}(s)/(1-s)^{a}(1+s)^{b} \tag{1.6}$$

where $f_{\beta}(s)$ is nonsingular and the powers a and b reflect the strength of the singularity at the crack tip. For a crack tip surrounded by a homogeneous medium, a=b=0.5, but the character of the singularity changes when the tip reaches a bimaterial interface; such "non-square-root" singularities formally require more complex numerical methods (based on Gauss-Jacobi polynomials) than those used below. However, a simple and apparently adequate approximation (suggested by Cleary [14]) would be to still employ a≈b≈0.5 for crack tips near inhomogeneity boundaries and allow the amplitude $f_{\beta}(\pm 1)$ to be unbounded. Hence, we adopt as unknown the amplitude f_{β} defined by

$$\delta'_{\beta}(s) = f_{\beta}(s)/\sqrt{(1-s^2)} \tag{1.7}$$

Combining eqns. (1.3), (1.5) and (1.7) gives, for m=1,2,...,N,

$$\pi \begin{Bmatrix} \sigma_\nu(\xi_m) \\ \sigma_\nu(\xi_m) \end{Bmatrix} = \bar{E} \sum_{n=1}^{N} \sum_{\beta=1}^{2} \int_{-1}^{1} \frac{ds_n f_\beta(s_n)}{\sqrt{(1-s_n^2)}} \left[\frac{1}{(\xi_m - s_n)} \begin{Bmatrix} g_{\nu m}^{\beta n} \\ g_{\tau m}^{\beta n} \end{Bmatrix} + \begin{Bmatrix} h_\nu^\beta(\xi_m, s_n) \\ h_\tau^\beta(\xi_m, s_n) \end{Bmatrix} \right] \tag{1.8}$$

where the notation of eqn. (1.4) has been applied to g and h.

In accordance with the Gauss-Chebyshev integration formula, the interpolation functions are chosen to be the Chebyshev polynomials; this scheme both exploits the square-root singularity on the density and is extremely useful in determining the strength of that singularity. Following a procedure such as that outlined in Appendix 2 of [16] (presented here in Appendix C) eqn. (1.8) can be reduced to (arbitrarily precise) approximation in series form:

$$\begin{Bmatrix} \sigma_\nu(\xi_{mr}) \\ \sigma_\tau(\xi_{mr}) \end{Bmatrix} \simeq \sum_{n=1}^{N} \frac{\pi}{M_n} \sum_{\beta=1}^{2} \sum_{k=1}^{M_n} f_\beta(s_{nk}) \begin{Bmatrix} \Gamma_\nu^\beta(\xi_{mr}, s_{nk}) \\ \Gamma_\tau^\beta(\xi_{mr}, s_{nk}) \end{Bmatrix} \tag{1.9}$$

for $m=1,2,\ldots,N$ and $r=1,2,\ldots,M_m-1$.

Eqn. (1.9) is a standard matrix equation in which the dislocation density strengths $f_\beta(s_n)$ are evaluated at the M_n zeros of T_{M_n}, the Chebyshev polynomials of the first kind, and the crack tractions are specified at the M_n-1 zeros of $U_{M_n}-1$, the Chebyshev polynomials of the second kind. Since eqn. (1.9) gives $2\sum_{m=1}^{N}(M_m-1)$ equations for

$2\sum_{n=1}^{N} M_n$ unknowns, 2N more equations are needed to complete the system of equations: two more for each crack. The approximate equations to be used depend on the specific crack configuration being modelled.

For a completely embedded crack, the two extra equations arise from the physical situation that the crack must have a determinate amplitude of entrapped dislocations,

$$\delta_{\beta E} = \int_{-1}^{1} \delta'_{\beta}(s_n)ds_n = \int_{-1}^{1} \frac{f_{\beta}(s_n)ds_n}{\sqrt{(1-s_n^2)}} \simeq \frac{\pi}{M_n} \sum_{k=1}^{M_n} f_{\beta}(s_k), \quad \beta = 1,2 \qquad (1.10)$$

In the case of no net entrapped dislocation ($\delta_{\beta E}=0$), the "closure condition," eqn. (1.10), satisfies the single-valuedness condition of the displacement discontinuity.

When the configuration includes only N nonintersecting cracks, the 2N extra conditions are provided by the N sets of closure conditions. In modelling different crack configurations however, i.e., multiple intersecting cracks or cracks intersecting an interface, the individual closure conditions imposed over each crack surface do not provide a physically reasonable account of the dislocation density distribution. A modified closure condition must be provided, together with other conditions necessary to ensure that the stress state at the point of intersection is bounded. These "matching conditions" are extensions of those suggested in [1] for surface and branch cracks and must be separately

discussed, as in the next chapter for the different configurations considered.

Eqn. (1.9) along with the appropriate closure and matching conditions forms a complete set of linear equations which can be solved for the $f_\beta(s_{nk})$. The stress $\sigma_{\alpha\gamma}$ at any point (x,y) in the body can then be computed as

$$\sigma_{\alpha\gamma}(x,y) \simeq \sum_{n=1}^{N} \frac{\pi}{M_n} \sum_{\beta=1}^{2} \sum_{k=1}^{M_n} f_\beta(s_{nk}) \Gamma^{\beta}_{\alpha\gamma}([x,y],s_{nk}) \qquad (1.11)$$

The crack-tip stress intensity factors and crack opening displacements can then be calculated from the density distribution, once determined.

Stress Intensity Factors

For crack tips embedded in a homogeneous medium, the stress intensity factors in the plane of the tips are defined in terms of the normal and shear stresses

$$K_\alpha(\underset{\sim}{x}^a_m) = \lim_{\underset{\sim}{x}\to\underset{\sim}{x}^a_m} [2|\underset{\sim}{x}^a_m - \underset{\sim}{x}|]^{1/2} \sigma_\alpha(\underset{\sim}{x}) \qquad (\alpha=1,2) \qquad (1.12)$$

and are related to the (bounded) amplitude functions f_1 and f_2 as follows:

$$\begin{Bmatrix} K_1(\underset{\sim}{x}^a_m) \\ K_2(\underset{\sim}{x}^a_m) \end{Bmatrix} \simeq \bar{E}\left(\frac{\pi L_m}{2}\right)^{1/2} \begin{bmatrix} -\cos\psi_{m1} & -\sin\psi_{m1} \\ \sin\psi_{m1} & -\cos\psi_{m1} \end{bmatrix} \begin{Bmatrix} f_1(s_{m1}) \\ f_2(s_{m1}) \end{Bmatrix} \qquad (1.13)$$

A positive mode II factor, K_2, is associated with a path

that diverges to the right side of the crack plane for an observer looking outwards at the crack tip. A negative K_2 is associated with a path that diverges to the left.

Crack Opening Displacements

The crack opening displacements at any point along a curvilinear surface can be computed by integrating the dislocation densities δ'_β over that surface,

$$\delta_\beta(s_n) = \delta_\beta(+1) - \left(\frac{L_n}{2}\right)\int_1^{s_n} \delta'_\beta(s)ds$$

$$= \delta_\beta(+1) - \left(\frac{L_n}{2}\right)\int_1^{s_n} \frac{f_\beta(s)}{\sqrt{(1-s^2)}}\,ds \qquad (1.14)$$

where β takes on values 1,2 to signify normal and shear components, respectively.

The integral in eqn. (1.13) is evaluated by choosing an appropriate interpolation function for the density $f_\beta(s)$. As described in Appendix B of [13] (reproduced here in Appendix C), $f_\beta(s)$ is first approximated by Chebyshev polynomials of the first kind; then the term-by-term integration leads to

$$\delta_\beta(s_n) = \delta_\beta(+1) + \frac{2}{M_n}\left(\frac{L_n}{2}\right)\sum_{k=1}^{M_n} \{1/2\,\cos^{-1}s_n$$

$$+ \sum_{m=1}^{M_n-1} \frac{1}{m}\cos\frac{m(2M_n-2k+1)\pi}{2M_n}\sin(m\cos^{-1}s_n)\}f(s_{nk}) \qquad (1.15)$$

The opening and shear displacements δ_1, δ_2 on the plane of the crack surface at s_n are obtained by transforming the global displacements δ_β as follows:

$$\begin{Bmatrix} \delta_1(s_n) \\ \delta_2(s_n) \end{Bmatrix} = \begin{bmatrix} -\cos\psi_n & -\sin\psi_n \\ -\sin\psi_n & \cos\psi_n \end{bmatrix} \begin{Bmatrix} \delta_x(s_n) \\ \delta_y(s_n) \end{Bmatrix} \qquad (1.16)$$

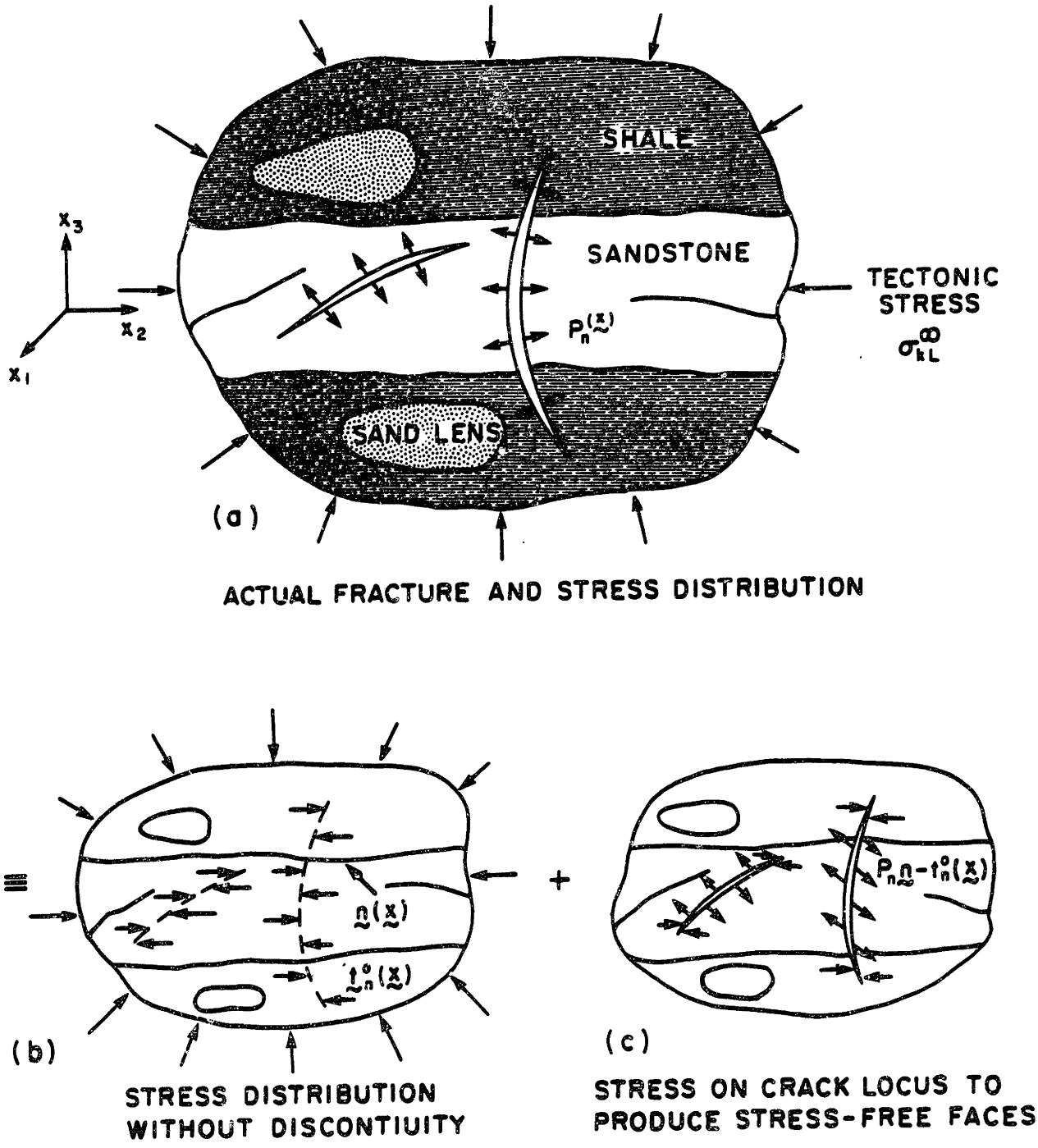


Figure 1.1 Schematic of the superposition process for simulating hydraulic fracture evolution, showing driving stresses around crack tips.

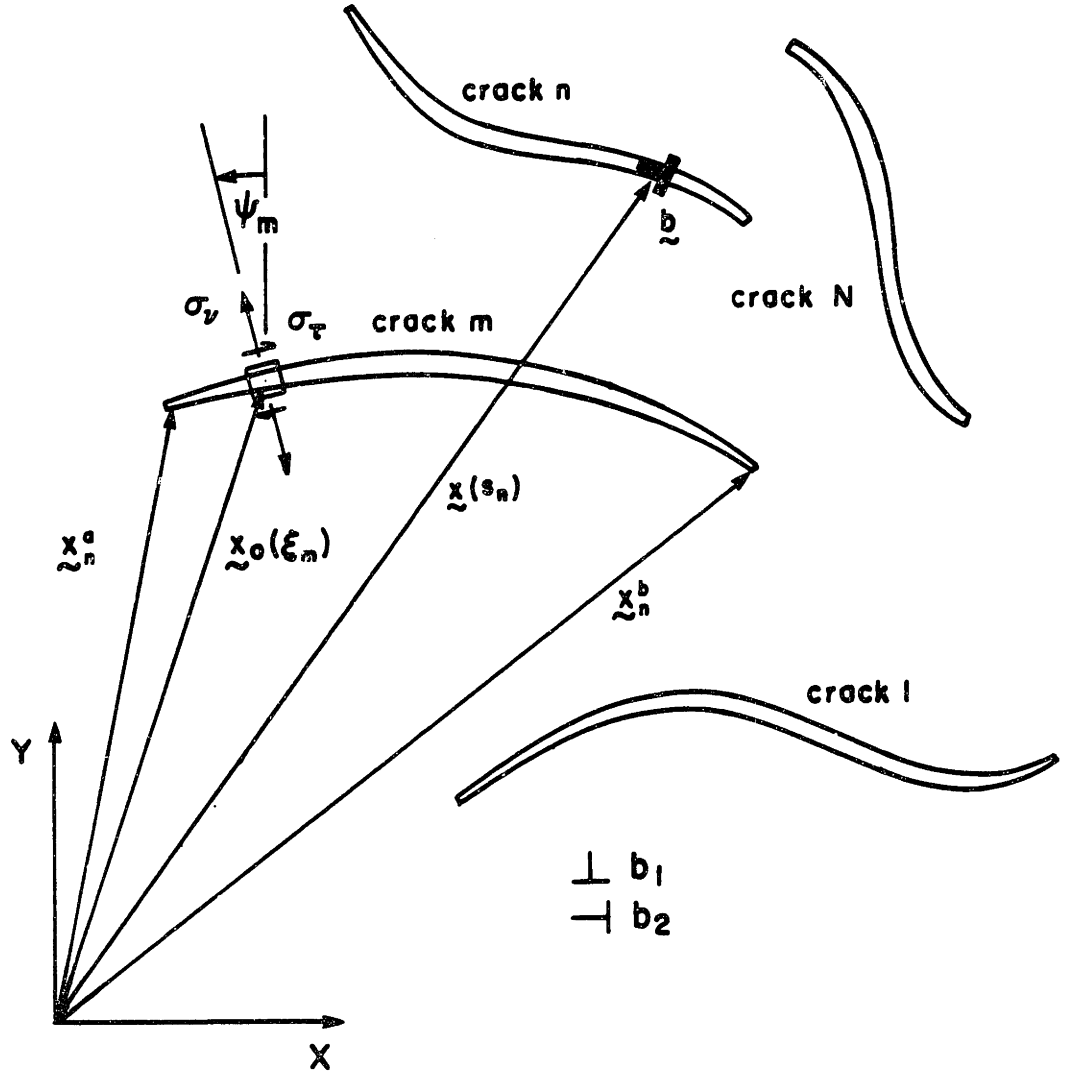


Figure 1.2 System of N curvilinear cracks in the plane

CHAPTER 2

ANALYSIS OF VARIOUS CRACK CONFIGURATIONS

As described in the previous chapter, the closure and matching conditions required to close our system of equations depend on the specific crack configuration being analyzed. These conditions are chosen so as to provide a physically realistic account of the dislocation density distribution over all the crack surfaces, as well as to ensure boundedness of stress states at points of crack intersection. The following sections describe the detailed modelling for several configurations of interest, shown schematically in Fig. 2.1; particularly, results are presented for the crack-tip stress intensity factors, of most interest among the data generated by the computer program developed to implement the various formulations in a single framework.

2.1 Multiple Nonintersecting Rectilinear Cracks

This configuration may include any number of rectilinear cracks in the plane, as shown in Fig. 2.1a. The crack surfaces may be distinguished as consisting of a "main" crack and any number of "branch" cracks. The branch cracks may fork from a single origin, and/or form a piecewise linear segment extending from the tip of the main crack. Each main and branch crack is transformed (by the scheme described in Appendix A) onto the interval (-1,+1), with a distinct number of nodal points for each crack. If a surface is discretized into N branch crack segments, a total of (2N+2) additional equations is required to complete the system of linear equations for that surface.

The closure condition enforces that the main and branch cracks must together have a known (usually zero) amount of entrapped dislocations, δ_E, whereby additional integrals are added to eqn. (1.10) to account for the dislocations of the branch crack(s),

$$\sum_{m=1}^{N+1} \left(\frac{L_m}{2}\right) \int_{-1}^{1} \frac{f_{\beta m}(s_n)\,ds_n}{\sqrt{(1-s_n)^2}} \simeq \sum_{m=1}^{N+1} \frac{\pi}{M_{nm}} \frac{L_m}{2} \Sigma\, f_{\beta m}(s_n) = \delta_{\beta E} \qquad (\beta = 1,2) \qquad (2.1)$$

The additional 2N equations to be used for the branch cracks are chosen to satisfy the physical requirement that the stress singularity at the point of crack intersection is less than that at the crack tip. This could be done by

enforcing a continuous distribution of the dislocation density across the point of intersection, but it has been found that somewhat more accurate results are obtained by setting the amplitude of the singular dislocation density of the branch crack equal to zero at the point where it intersects the main crack or preceding branch crack [1]. These matching conditions can be expressed as

$$f_{\beta m}(s_n=-1) \simeq f_{\beta m}(s_{n1}) = 0 \qquad (\beta=1,2;m=1,\ldots,N) \qquad (2.2)$$

The stress intensity factors for 2-crack and 3-crack configurations is presented in Figs. 2.2-2.4, where 21 nodal points were used on each crack. Comparison of these results with those in the literature shows excellent agreement. Problems involving general branching at crack tips (as schematized in Fig. 2.1a) have also been studied extensively (see Lam [19]) using a similar formulation; the results are again in very good agreement with those in the literature, obtained by a variety of different (usually more complex) techniques.

2.2 Curved Cracks

The analysis of curvilinear cracks has thus far been limited to parabolic arc cracks [20] and slightly curved or kinked single cracks in the infinite plane [8-11]. The present scheme, however, allows for crack surfaces of any arbitrary shape (Fig. 2.1b). The surfaces are described by a number of 'specification points' (see Appendix A) and, as for rectilinear cracks, may consist of main and branch cracks. The closure and matching conditions are also entirely analogous to those for the rectilinear cracks.

The stress intensity factors for circular, parabolic, and elliptical arc cracks are shown in Figs. 2.5-2.13. The results were obtained by discretizing the curved surfaces into piecewise linear segments which each subtend 1° at the center or focus of the arc, and with 29 nodal points on each crack. Three load configurations are considered for each geometry, viz. horizontal and vertical tension loadings, and uniform internal pressure in the crack.

Note that the stress intensity factors for any combination of these loads may be obtained from the results presented here by a simple superposition of the different loads acting individually. The excellent agreement of the results, with the various special calculations already available, testifies to the accuracy of the coordinate transformation and surface discretization schemes, which may now be used to analyze crack geometries of any shape.

2.3 Radial Cracks

The so-called 'star-crack' configuration considered in this section is shown schematically in Fig. 2.1c; here the arms (or wings) of the crack may have different lengths. Two types of expansion loads are considered: uniform all-round tension and uniform internal pressure.

The modelling of the radial crack system differs from that of the other configurations in that it was found more accurate to transform the main crack onto the integration interval (0,+1) instead of (-1,+1). Any branch crack is still considered on (-1,+1). For a system of M main cracks and B branch cracks, a total of 2(M+B) extra equations is required. The two closure conditions enforce a determinate amplitude (usually zero) of entrapped dislocations in the entire system,

$$\sum_{m=1}^{M}\left\{L_m\int_0^{+1}\frac{f_{\beta m}(s_n)\,ds_n}{\sqrt{(1-s_n^2)}}+\sum_{b=1}^{B_m}\left(\frac{L_b}{2}\right)\int_{-1}^{+1}\frac{f_{\beta b}(s_n)\,ds_n}{\sqrt{(1-s_n^2)}}\right\}=\delta_{\beta E}$$

$$(\beta=1,2) \qquad (2.3)$$

where L_m, L_b are the lengths of the main and branch cracks respectively. Following the integration formula developed in Appendix C, these conditions may be approximated by

$$\sum_{m=1}^{M}\left\{\frac{2}{M_{nm}}L_m\sum_{k=1}^{M_{nm}}\left\{\frac{\pi}{4}+\sum_{\ell=1}^{M_{nm}-1}\frac{1}{\ell}\cos\frac{\ell(2M_{nm}-2k+1)}{2M_{nm}}\sin\left(\frac{\pi}{2}\ell\right)\right\}f_{\beta m}(s_k)\right.$$

$$\left.+\sum_{b=1}^{B_m}\left\{\frac{\pi}{M_{nb}}L_b\sum_{k=1}^{M_{nb}}f_{\beta b}(s_k)\right\}\right\}=\delta_{\beta E} \qquad (\beta=1,2) \qquad (2.4)$$

The remaining conditions have been sought to ensure bounded stresses at points of crack intersection; in addition to the 2B matching conditions provided by the branch cracks, (2M-2) conditions need to be imposed at the point of intersection of the main cracks, or at the origin of the radial crack system. One possible approach would be to set the amplitude of the singular dislocation densities of any (M-1) of the main cracks to zero at the point of intersection. It seems physically intuitive however to visualize the radial system as cracks intersecting a free surface, as schematically shown in Fig. 2.14. The free surface is formed by the intersection of two cracks (labelled A,B in Fig. 2.14), and testing indicates that these two cracks should be chosen such that the joined surface is as straight as possible. A free surface that included a sharp turn in its geometry (for example, that formed by cracks A and C) is unsuitable because of the discontinuous distribution of dislocation density at the kink. A careful selection of the free surface thus makes it reasonable to enforce continuity of the dislocation density,

$$f_{\beta A}(s_n=0 \simeq s_{(M_{nA}+1)/2}) = f_{\beta B}(s_n=0 \simeq s_{(M_{nB}+1)/2}) \quad (\beta=1,2) \tag{2.5}$$

For a crack intersecting a free surface, it has been found that the dislocation at the point of intersection has

an opening component and no sliding component [1]. Setting both of these components to zero at the surface was shown to give values of stress intensity factors that are virtually identical to those obtained through other methods (e.g., precise conformal mapping). Consequently, it seems plausible to impose similar conditions on all the cracks that intersect the free surface formed by the other two cracks. These matching conditions are schematically illustrated in Fig. 2.14.

Figs. 2.15-2.17 compare values for the stress intensity factors for an x-formed crack and a star-crack obtained using 15 nodal points on each wing with those available in the literature. The results of Aksogan [21] were obtained through the joint use of the Mellin transform and the Green's function technique, while Ouchterlony [22] used conformal mapping to implement the method of Muskhelishvili. The agreement with their results is very good, and gives confidence to the choice of the (0,+1) interval and the associated closure/matching conditions employed.

2.4 Cracks Near a Circular Inclusion

While results are available in the literature for one or two cracks [2,23] and even for branched cracks [24] in the vicinity of a circular inclusion, the analyses are restricted in the number, shape, and location of the cracks. The formulations presented in Chapter 1 and Section 2.1 are directly applicable to the study of an arbitrary crack configuration containing a circular inclusion, depicted in Fig. 2.1d; the only changes required are in the influence functions employed. The appropriate dislocation solutions are those given by Dundurs and Mura [25], and Dundurs and Sendeckyj [26], as rationalized in Appendix B. For applied stresses at infinity or for internal pressures in the inclusion, the stresses in the matrix may be found, e.g., using the Muskhelishvili method [27].

The results for a crack near a hole, obtained with 13 nodal points, are compared in Fig. 2.18 with those from Tada et al. [17]. They are presented solely for the purpose of verifying the basic methodology, which may now be used to study multiple (nonintersecting or intersecting) arbitrarily shaped crack systems near an inclusion.

2.5 Cracks Emanating from a Circular Inclusion

The problem of crack growth from, or crack termination at, circular inclusions (especially holes) has been the subject of considerable research [3,22,28-29], given the broad range of applications, from prediction of the severity of flaws emanating from fastener holes to preliminary analysis of fracture processes in rock blasting. The analysis to date, besides focusing on rather limited shape and load configurations, have also involved (perhaps unwarranted) computational efforts which made analysis of realistic problems prohibitive. Referring to the inclusion crack geometry shown in Fig. 2.1e, each of the M main cracks may have any number B_m of branch cracks extending from their tips, and the matching conditions for the branch cracks are as before. Since the inclusion is perfectly bonded to the matrix, there is no slip at the interface when the inclusion has a nonzero shear modulus (Fig. 2.19a). Consequently, for $G_2>0$, any crack tip terminating at the bimaterial interface will be closed and the closure condition may be imposed on each crack surface,

$$\frac{L_m}{2}\int_{-1}^{+1}\frac{f_{\beta n}(s_n)ds_n}{\sqrt{(1-s_n^2)}}+\sum_{b=1}^{B_m}\frac{L_b}{2}\int_{-1}^{+1}\frac{f_{\beta b}(s_n)ds_n}{\sqrt{(1-s_n^2)}}=0$$

$$(\beta=1,2;\ m=1,\ldots,M) \qquad (2.6)$$

Table 2.1 compares the values of the stress intensity factors at the embedded tip with those computed by Erdogan

and Gupta [3]. The results in [3] were obtained by using a Gauss-Jacobi integration formula to solve a system of singular integral equations with generalized Cauchy kernels; their solution for the dislocation density function takes into account that the strength of the singularity at the crack tip at the interface is no longer square-root as per eqn. (1.6), but depends on the elastic properties of the inclusion and the matrix. While such an approach is formally necessary to precisely determine the stress intensity factors at tip B, the Gauss-Chebyshev scheme is clearly adequate to obtain the results for tip A (which still has a square-root stress singularity); the computational effort is also considerably reduced.

When the inclusion is a hole (G_2=0), there are no shear stresses at the interface and the crack tip B slips, resulting in an opening displacement δ_I (see Fig. 2.19b). If this displacement is known a priori, the dislocation density distribution must satisfy an equilibrium or compatibility condition of the form of eqn. (2.6), but with sum of integrals being equal to δ_I. Since δ_I depends on the matrix properties and the crack geometry, it is generally not a known constant and hence such a condition cannot provide the additional equations required. These equations can be extracted however by considering the limiting cases of a crack which is short and long by comparison with the hole radius.

For a short enough crack, the surface of the hole appears as a straight free surface (Fig. 2.20a), in which case it seems reasonable to impose the matching conditions suggested in [1] for a surface crack, i.e.,

$$f_\beta(-1) \simeq f_\beta(s_{n1}) = 0 \qquad (\beta=1,2) \qquad (2.7)$$

On the other hand, if the ratio of the crack length to the hole radius is large, the curvature of the hole can no longer be neglected and eqn. (2.7) is no longer applicable. As the crack length increases, so does the maximum crack opening; if the displacement δ_I is small relative to the maximum displacement (Fig. 2.20b), we may neglect δ_I and use eqn. (2.6) as an approximate closure condition.

The results obtained using these approximations (with 49 nodal points) is shown in Fig. 2.21, where they are compared with those in Tada et al. [17]. For crack lengths of up to 10% of the hole radius, the free surface matching condition, eqn. (2.7), is more appropriate, while the closure condition, eqn. (2.6), gives more accurate results for longer lengths.

For the case of multiple cracks emanating from a hole, the free surface condition can be imposed on each tip terminating at the hole. It is more physically rigorous, however, to impose a combination of the closure and matching conditions; even while each tip at the hole may have a finite opening, there is a specified (usually zero) net

entrapped dislocation in the entire configuration. This requirement is achieved by imposing that

$$\sum_{m=1}^{M}\left\{\frac{L_m}{2}\int_{-1}^{+1}\frac{f_{\beta m}(s_n)\,ds_n}{\sqrt{(1-s_n^2)}}+\sum_{b=1}^{B_m}\frac{L_b}{2}\int_{-1}^{+1}\frac{f_{\beta b}(s_n)\,ds_n}{\sqrt{(1-s_n^2)}}\right\}=\delta_{\beta E} \qquad (\beta=1,2) \quad (2.8)$$

which provides the two closure conditions. The additional equations are provided by setting the amplitude of the dislocation densities of any (M-1) cracks to zero at the points where they intersect the hole,

$$f_{\beta m}(s_n=-1)\simeq f_{\beta m}(s_{n1})\simeq 0 \qquad (\beta=1,2;\ m=1,\ldots,M-1) \quad (2.9)$$

The results for radial cracks from a hole are presented in Figs. 2.22-2.23 for two load configurations. A total of 49 nodal points was used for each crack. Previous results ([17,22]) were obtained by a mapping function method. The figures again evidence our belief that the Gauss-Chebyshev scheme, together with the appropriate closure/matching conditions, provides an accurate but more general, direct, and convenient approach to the analysis of inclusion/crack configurations than other available schemes, e.g., Gauss-Jacobi integration, mapping function methods, and finite element analysis.

2.6 Cracks Near One or More Straight Bimaterial Interfaces

The influence functions for an edge dislocation near a straight material interface is also available (e.g., [25]) and may be used with the scheme outlined in Chapter 1 to analyze crack configurations near an interface (Fig. 2.1f). The extra conditions to be used are exactly the same as for the nonintersecting and intersecting cracks in the infinite medium (see sections 2.1-2.3). It is to be noted that the analyses available in the literature (e.g., [5,30]) all again involve more mathematical formalism and less generality than the present scheme. For purposes of comparison, the stress intensity factors for an arbitrarily oriented crack near a free surface and in a bonded half plane are shown in Figs. 2.24-2.25, where the results were obtained using only 13 nodal points on the crack.

The study of cracks in an elastic layer bonded to two half planes (Fig. 2.1g) is motivated by a broad range of problems, from predicting the severity of flaws in composite materials to predicting the growth of natural faults and hydraulic fractures in stratified rock. Though the exact influence functions for an edge dislocation interacting with two material interfaces is as yet unavailable, approximate solution methods were developed in [13]. As detailed in Appendix B, the approximation consists of modifying the solution for a dislocation near a single interface to account for the second interface. The overall influence

function has the desired symmetry of allowing the interfaces to be interchanged, and can also be specialized to yield the solution for a dislocation in a homogeneous medium or near a single interface.

Using this approximation and 25 nodal points on the crack, the stress intensity factors for three different double interface configurations are shown in Figs. 2.26-2.28. It can be seen that the approximate influence function leads to reasonably accurate results, which improve as the crack dimension becomes smaller compared to the width of the elastic layer. The specialized numerical formulations presented by Erdogan and Gupta [5], and Erdogan and Arin [6], are evidently accurate for any crack dimension but pending the development of the exact influence function for an edge dislocation interaction with two material interfaces, the approximate solution seems adequate and more convenient for the analysis of arbitrary crack configurations in an elastic layer.

Table 2.1 The effect of modulus ratio on the stress intensity factors at the embedded tip A of a crack terminating at a circular inclusion (see Fig. 2.19(a), with a=R, $\nu_1=\nu_2=0.3$). Comparison with results from Erdogan and Gupta [3].

	$K_1(A)/\sigma\sqrt{(\pi a/2)}$	
$m = G_2/G_1$	Own	Erdogan and Gupta
0.05	1.5922	1.615
0.33	1.2280	1.229
1.0	0.9995	1.000
3.0	0.8603	0.8610
10.0	0.7960	0.7969
23.0	0.7787	0.7796
100	0.7681	0.7691
300	0.7660	0.7667

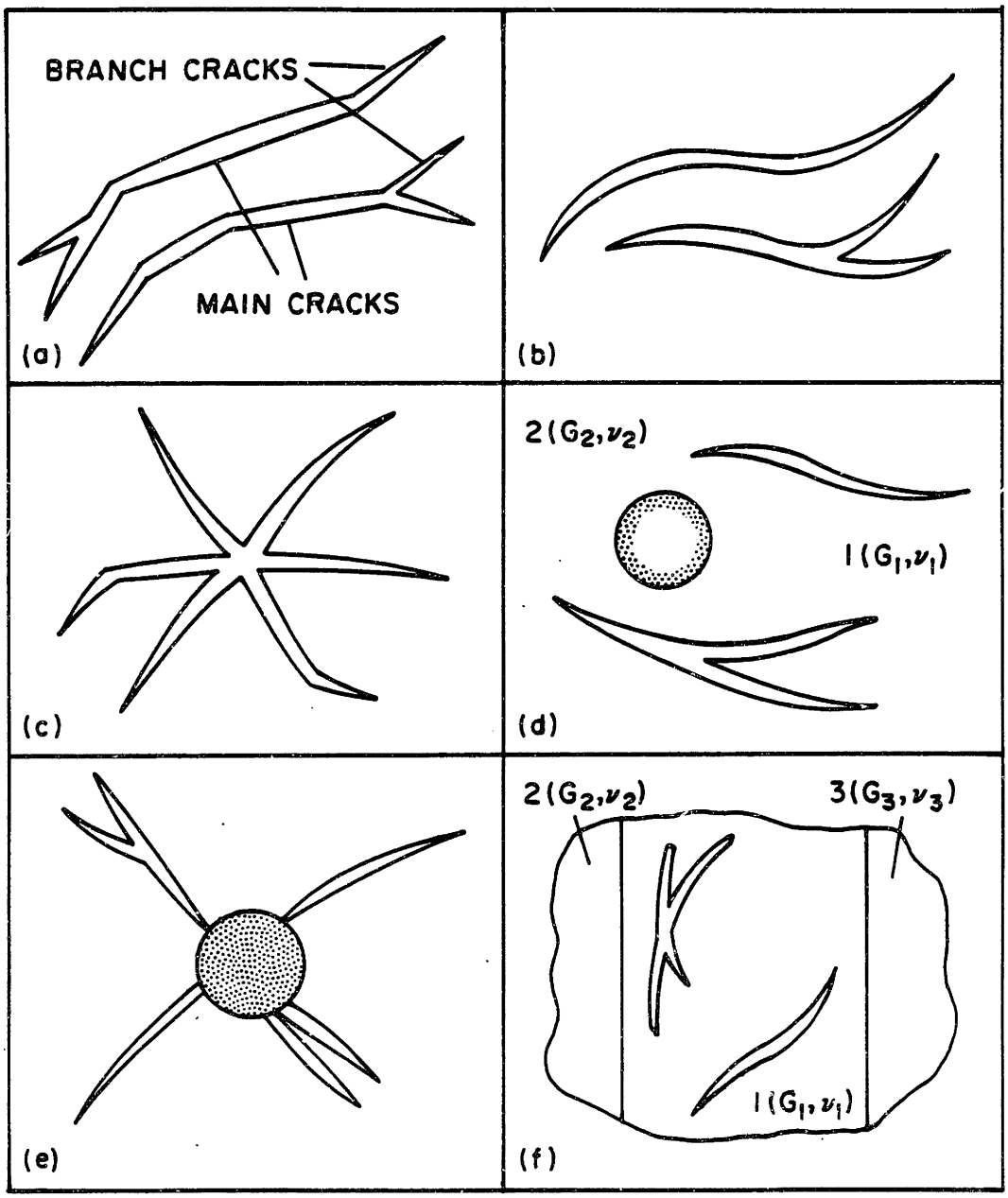


Figure 2.1 Schematic of crack configurations considered in this study. (a) Multiple nonintersecting rectilinear cracks; (b) Curved cracks; (c) Radial cracks; (d); Cracks near a circular inclusion; (e) Cracks emanating from a circular inclusion; and (f) Cracks near bimaterial interfaces.

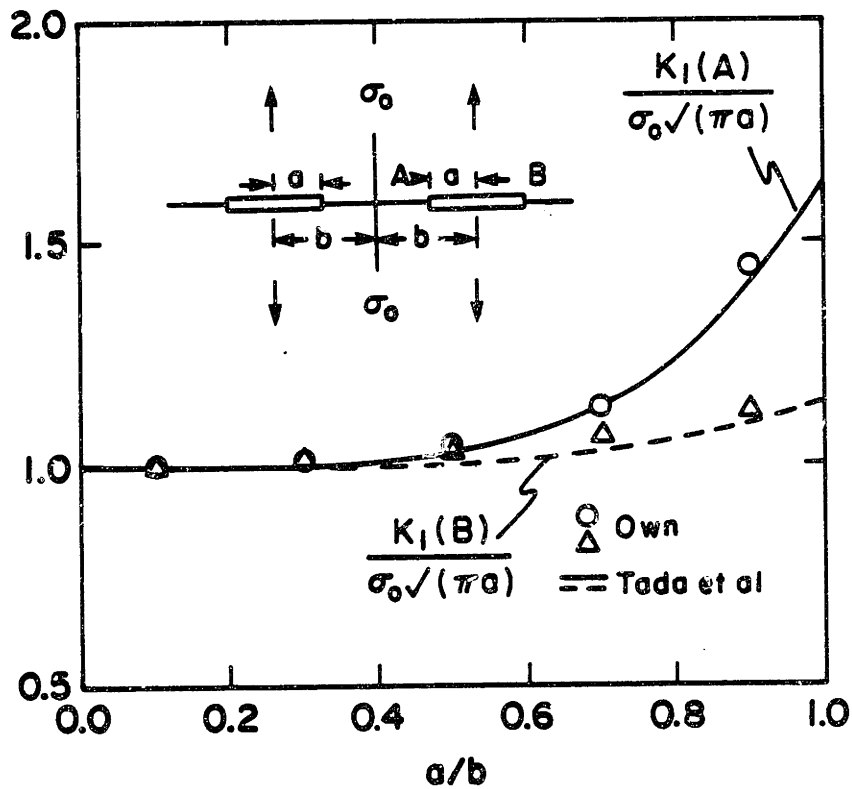


Figure 2.2 Comparison of stress intensity factors for two collinear cracks in infinite medium with those from Tada et al. [17].

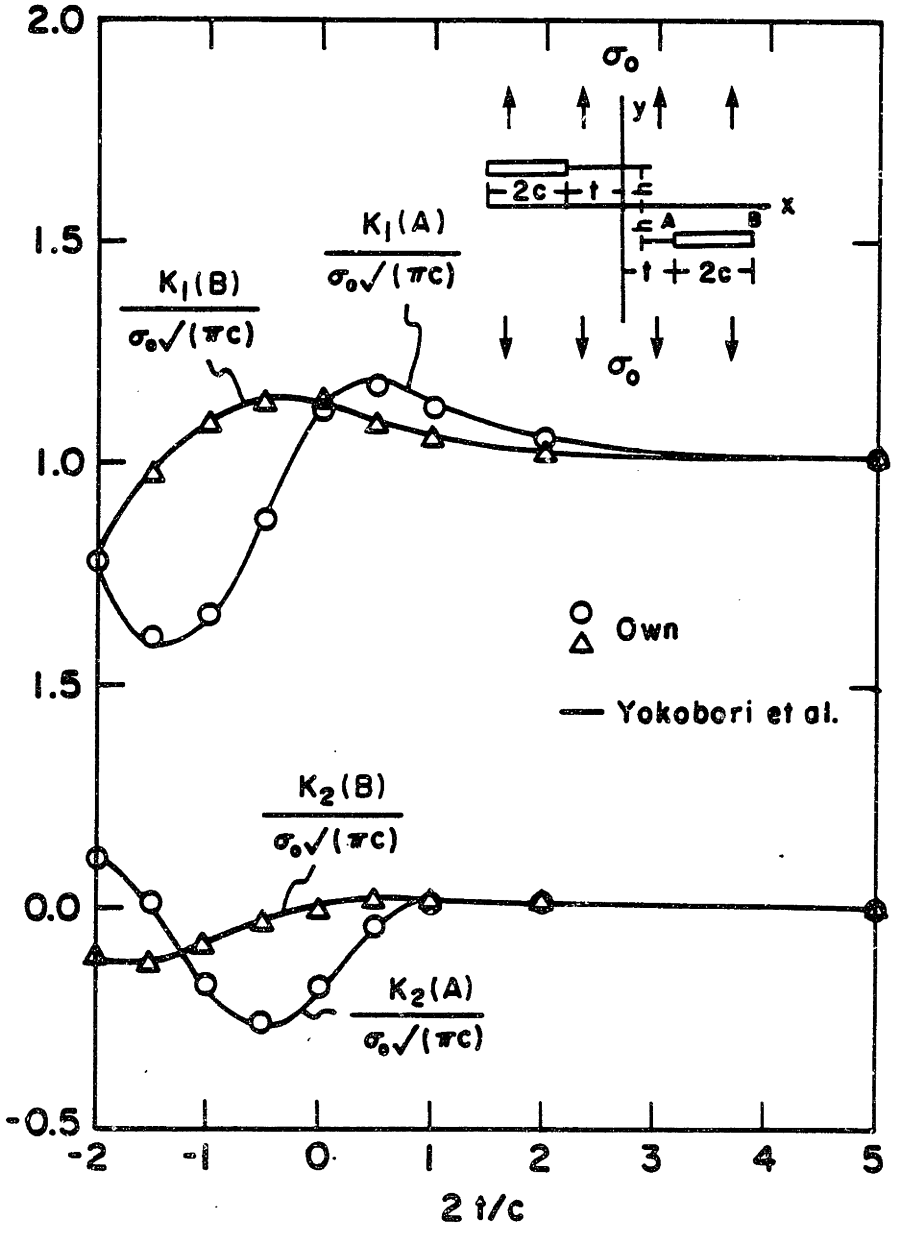


Figure 2.3 Comparison of stress intensity factors for two non-coplanar parallel staggered cracks in infinite medium with those from Yokobori et al. [18].

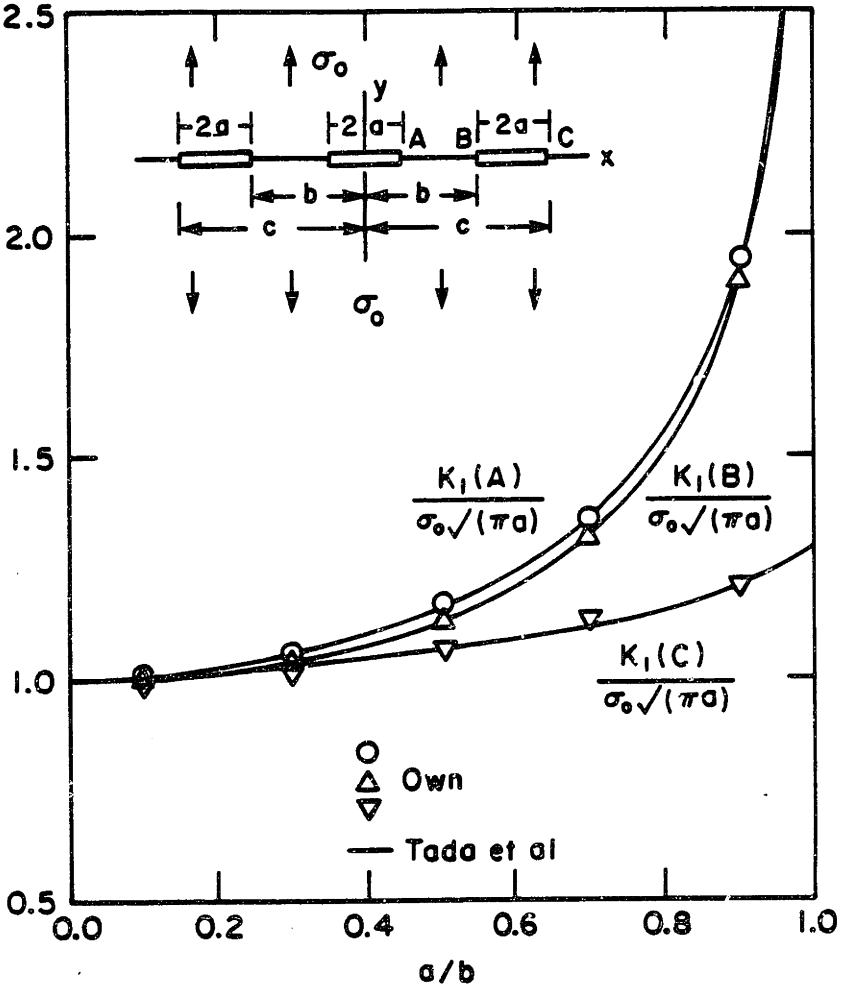


Figure 2.4 Comparison of stress intensity factors for three collinear cracks in infinite medium with those from Tada et al. [17].

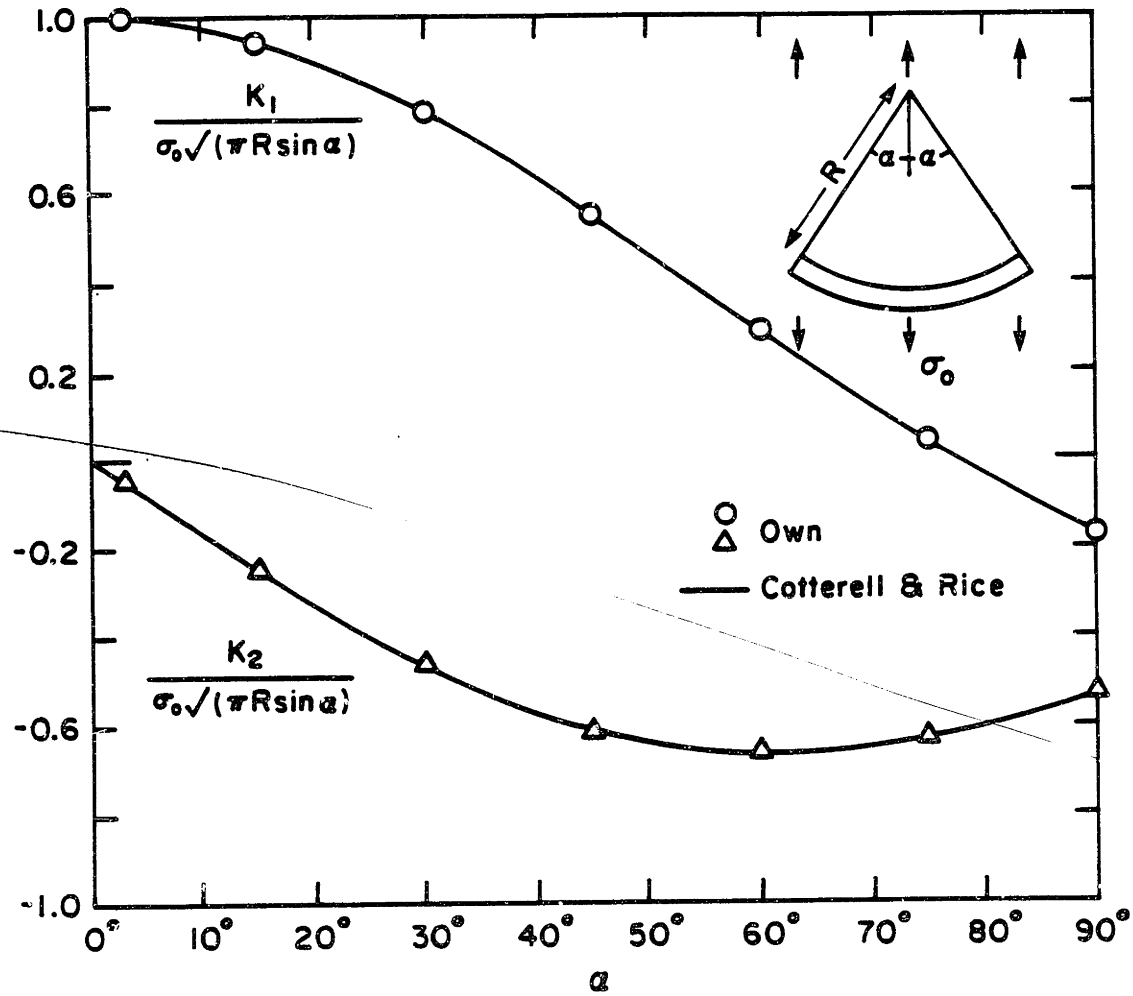


Figure 2.5 Comparison of stress intensity factor for circular arc crack under vertical stress with those from Cotterell and Rice [11].

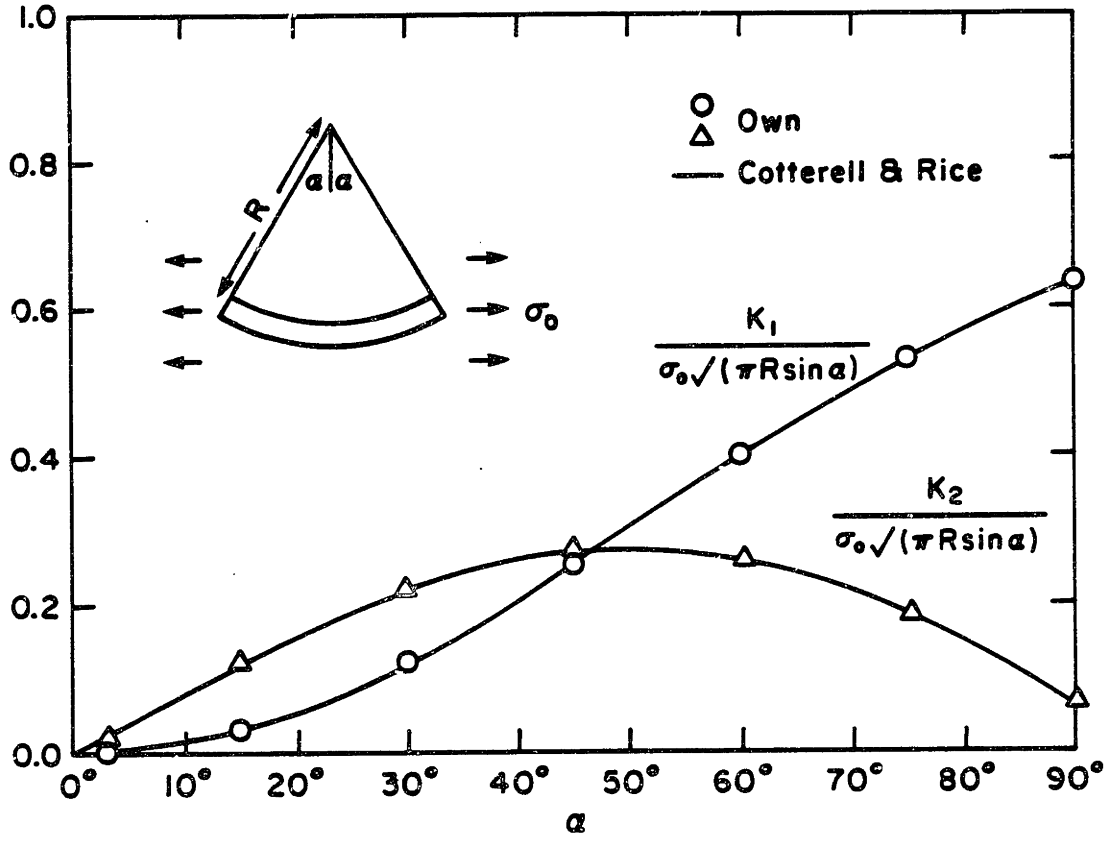


Figure 2.6 Comparison of stress intensity factors for circular arc crack under horizontal stress with those from Cotterell and Rice [11].

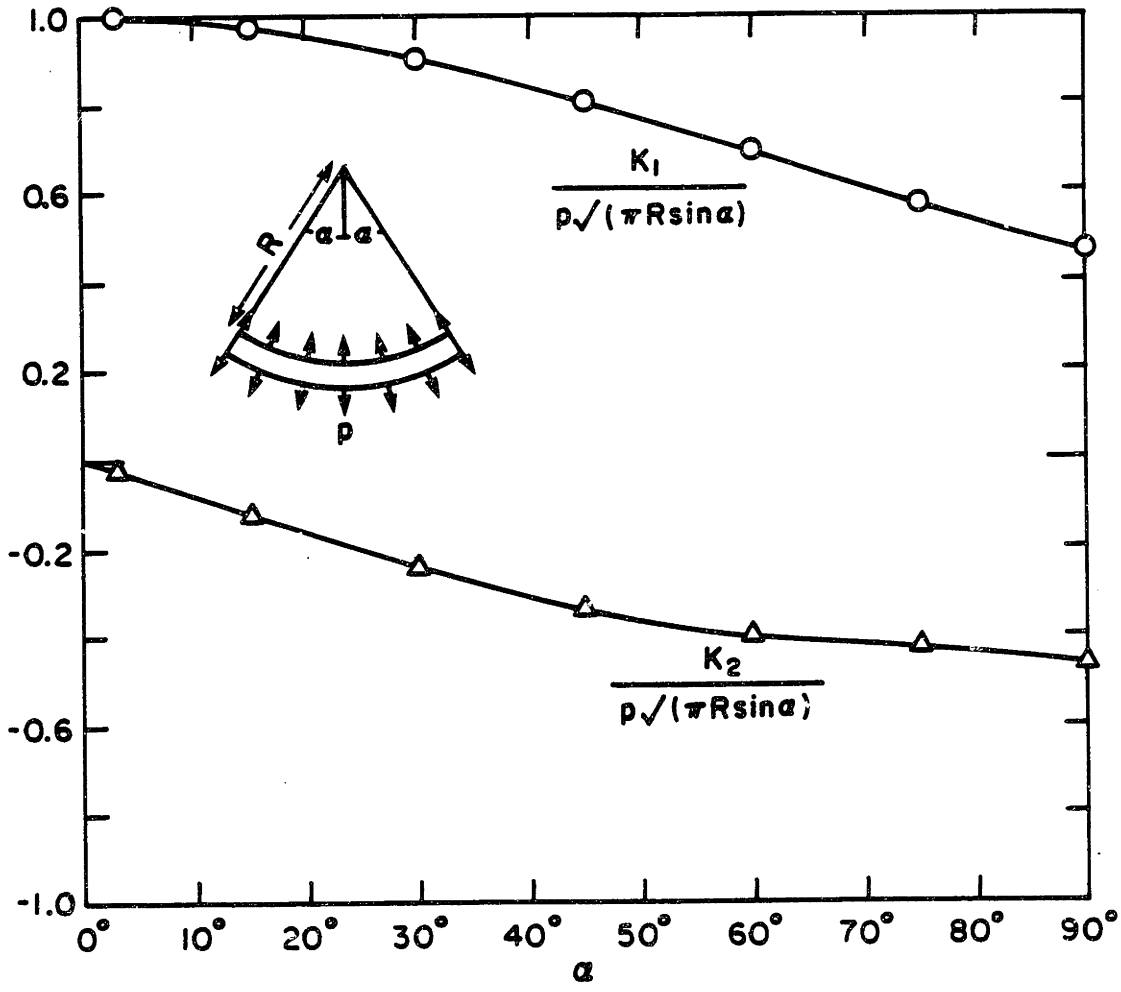


Figure 2.7 Stress intensity factors for circular arc crack under uniform internal pressure.

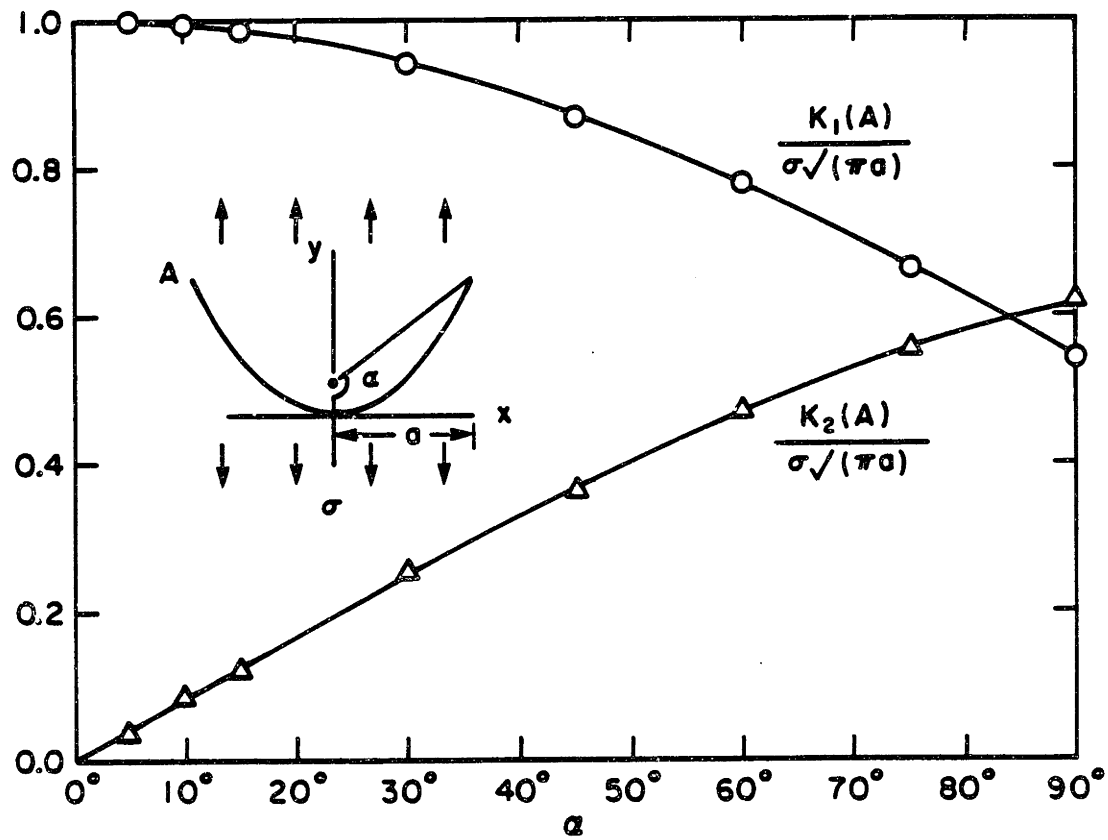


Figure 2.8 Stress intensity factors for parabolic arc crack under vertical stress.

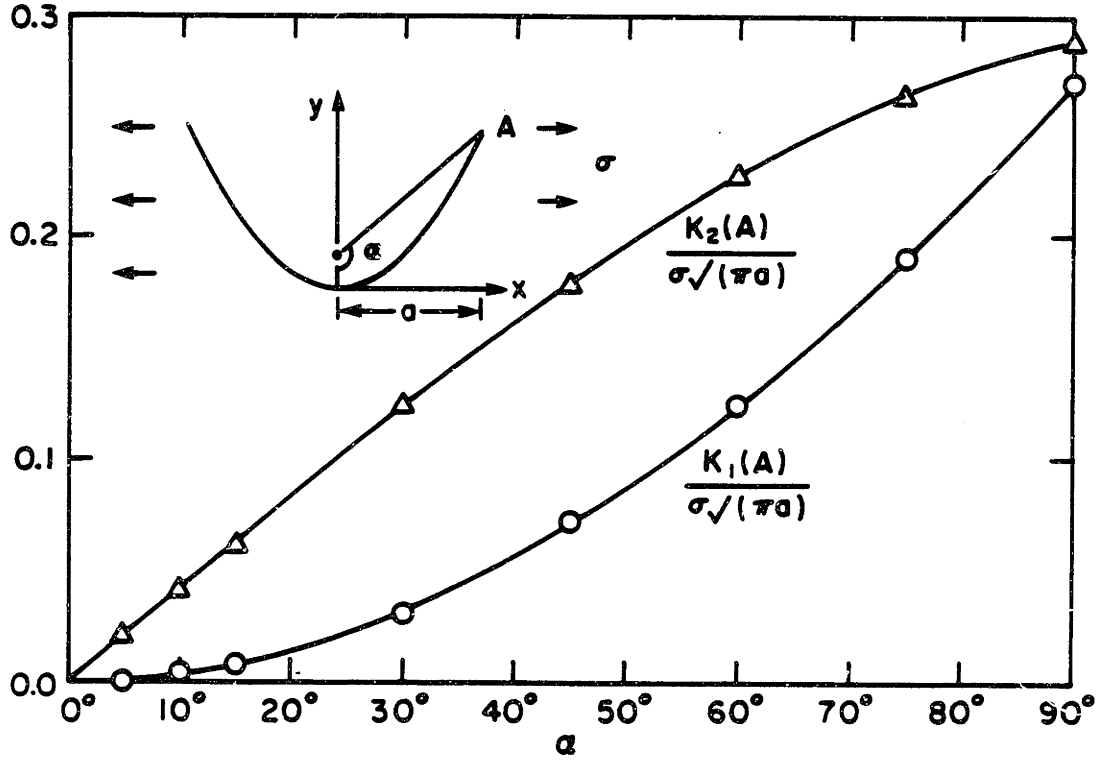


Figure 2.9 Stress intensity factors for parabolic arc crack under horizontal stress.

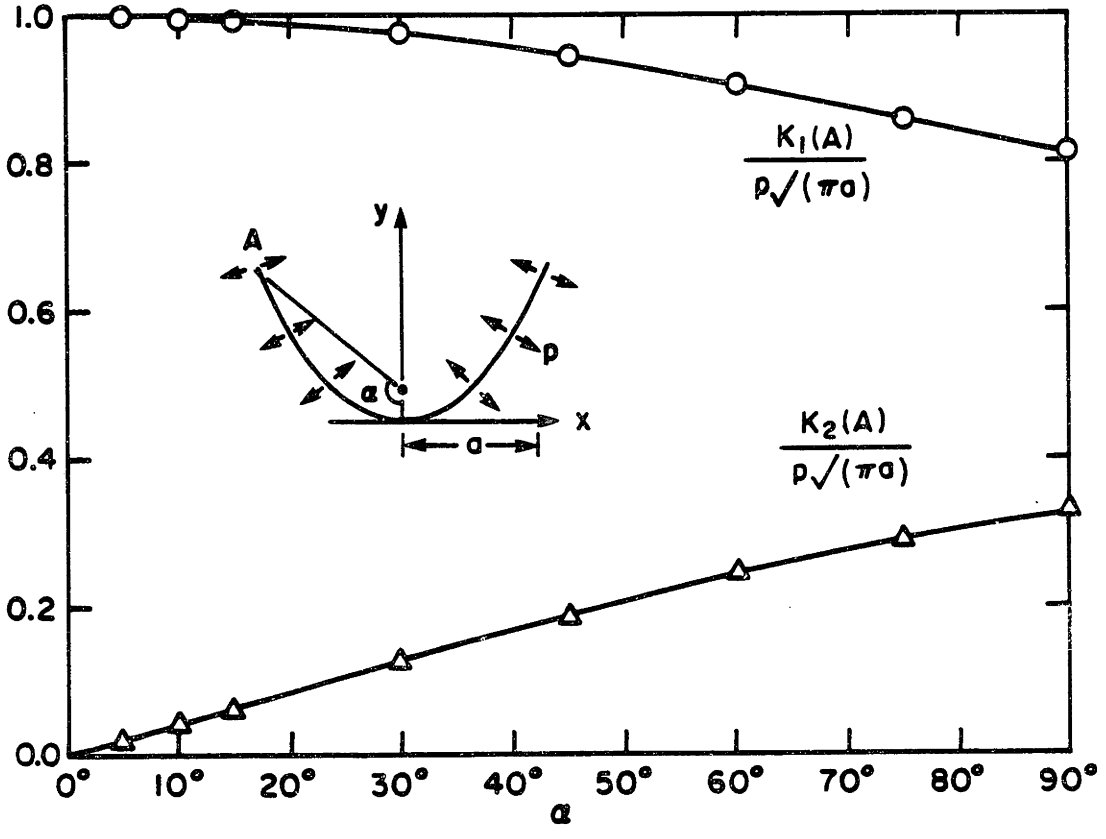


Figure 2.10 Stress intensity factors for parabolic arc crack under uniform internal pressure.

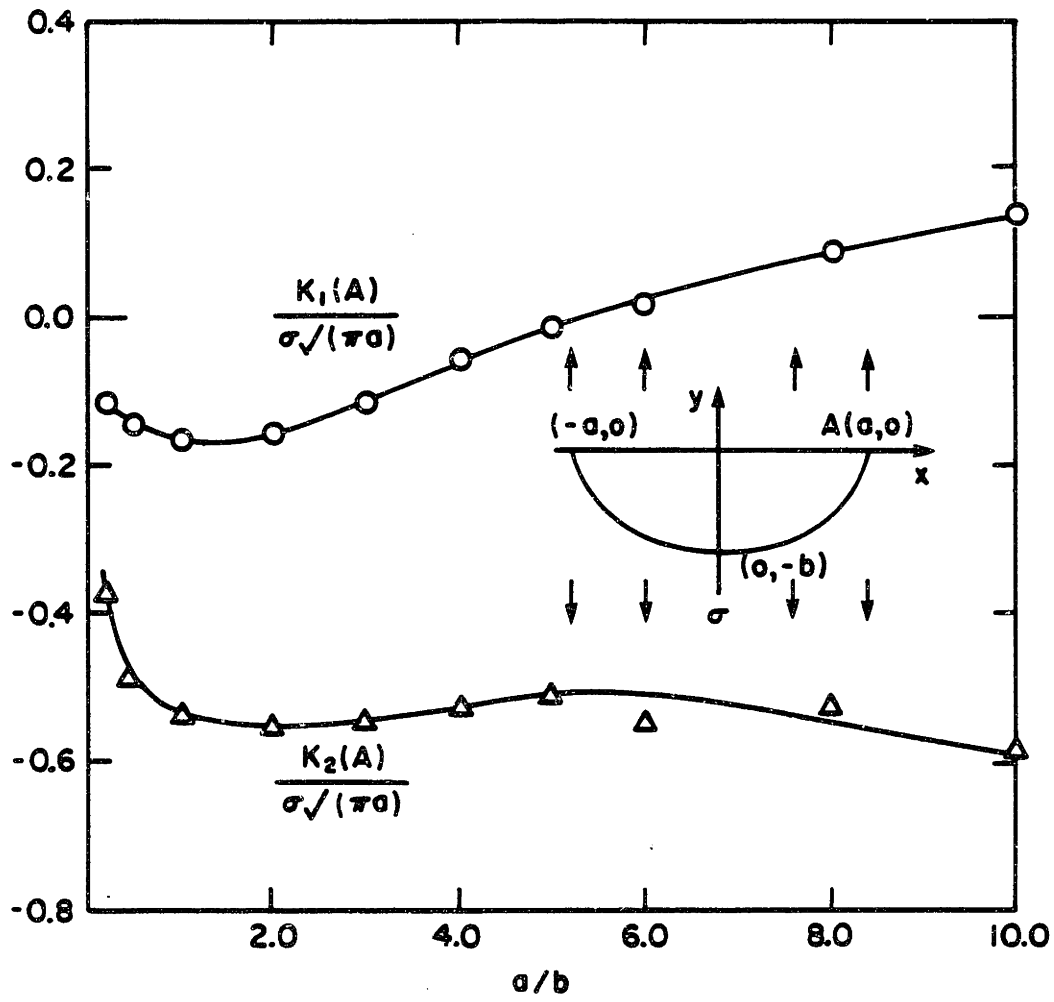


Figure 2.11 Stress intensity factors for elliptical arc crack under vertical stress.

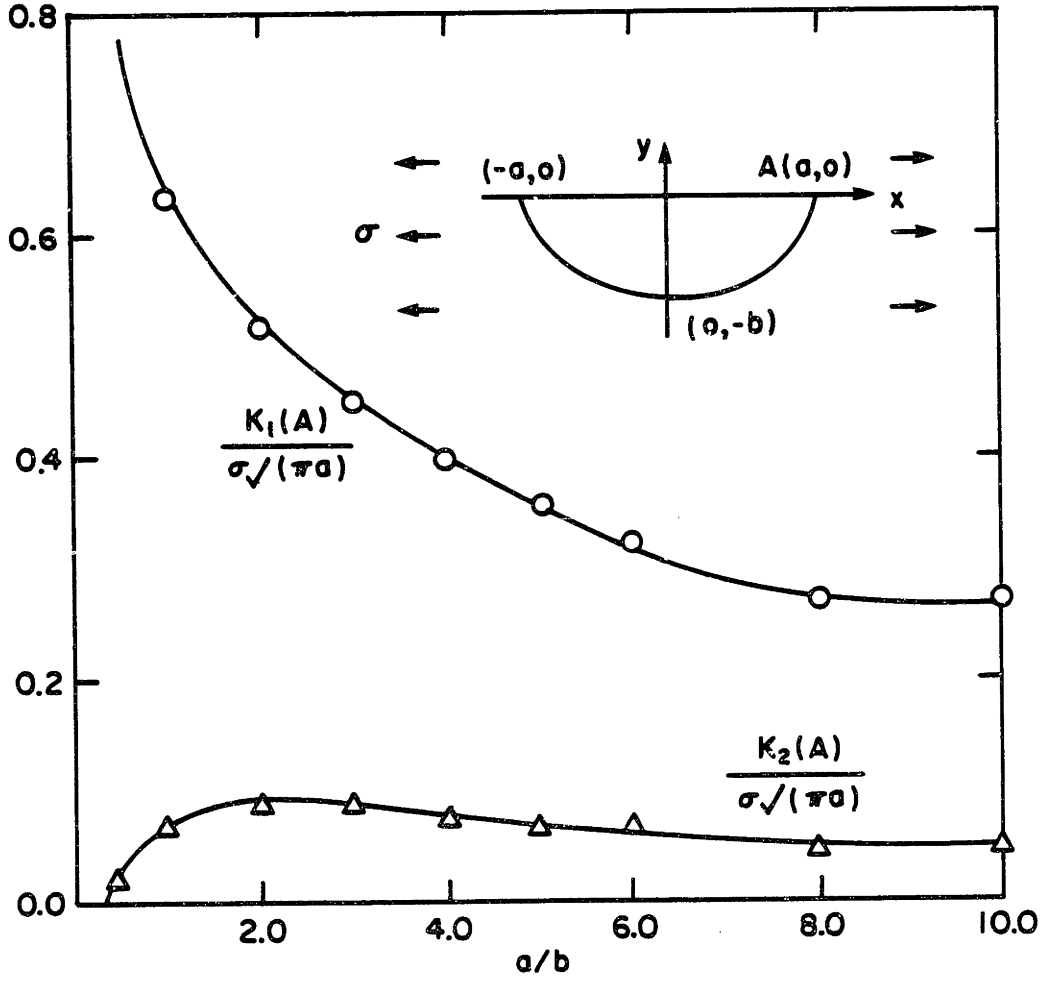


Figure 2.12 Stress intensity factors for elliptical arc crack under horizontal stress.

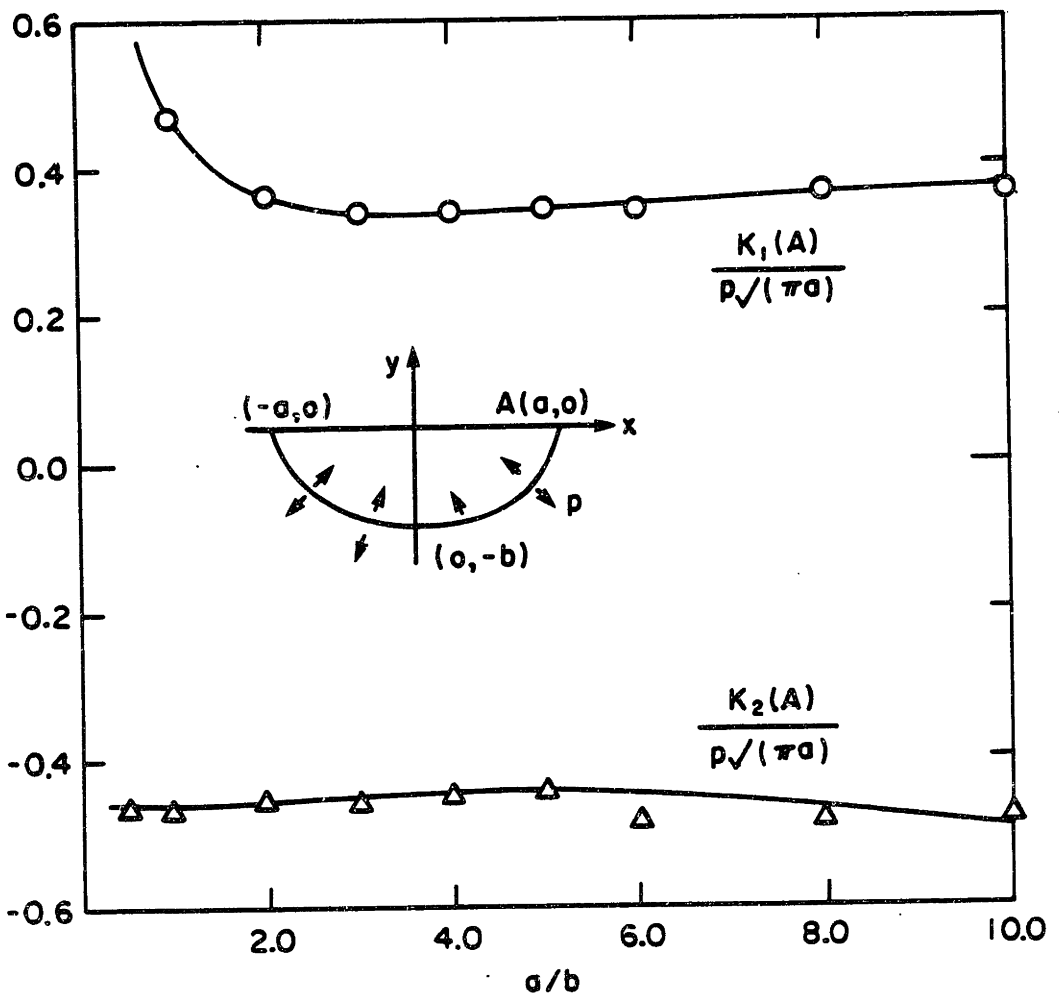


Figure 2.13 Stress intensity factors for elliptical arc crack under uniform internal pressure.

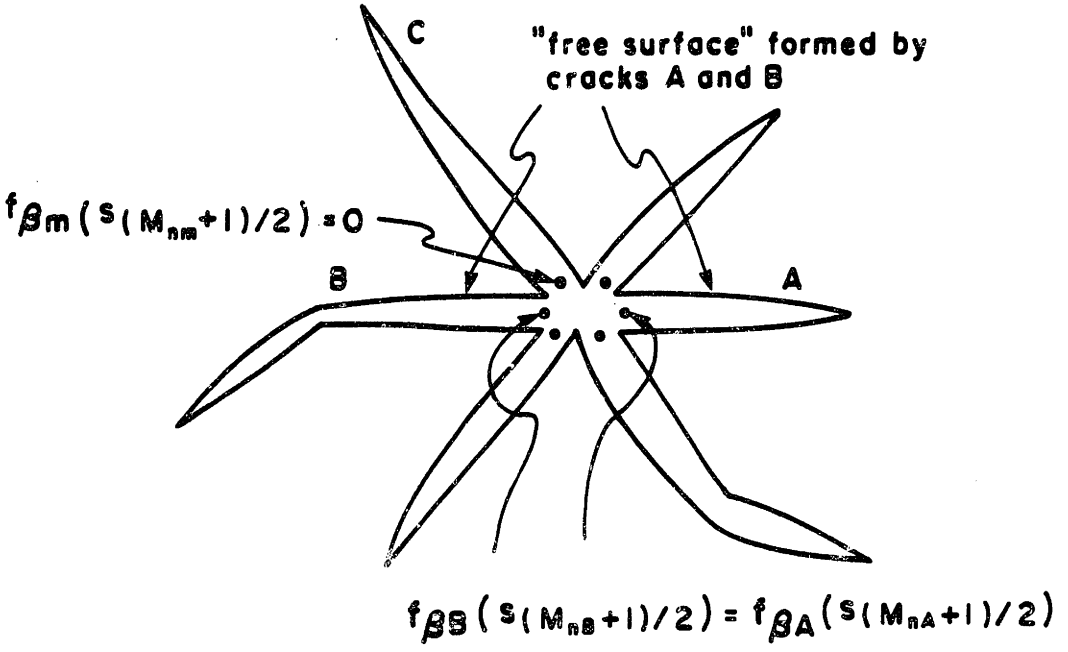


Figure 2.14 Matching conditions for a radial crack system.

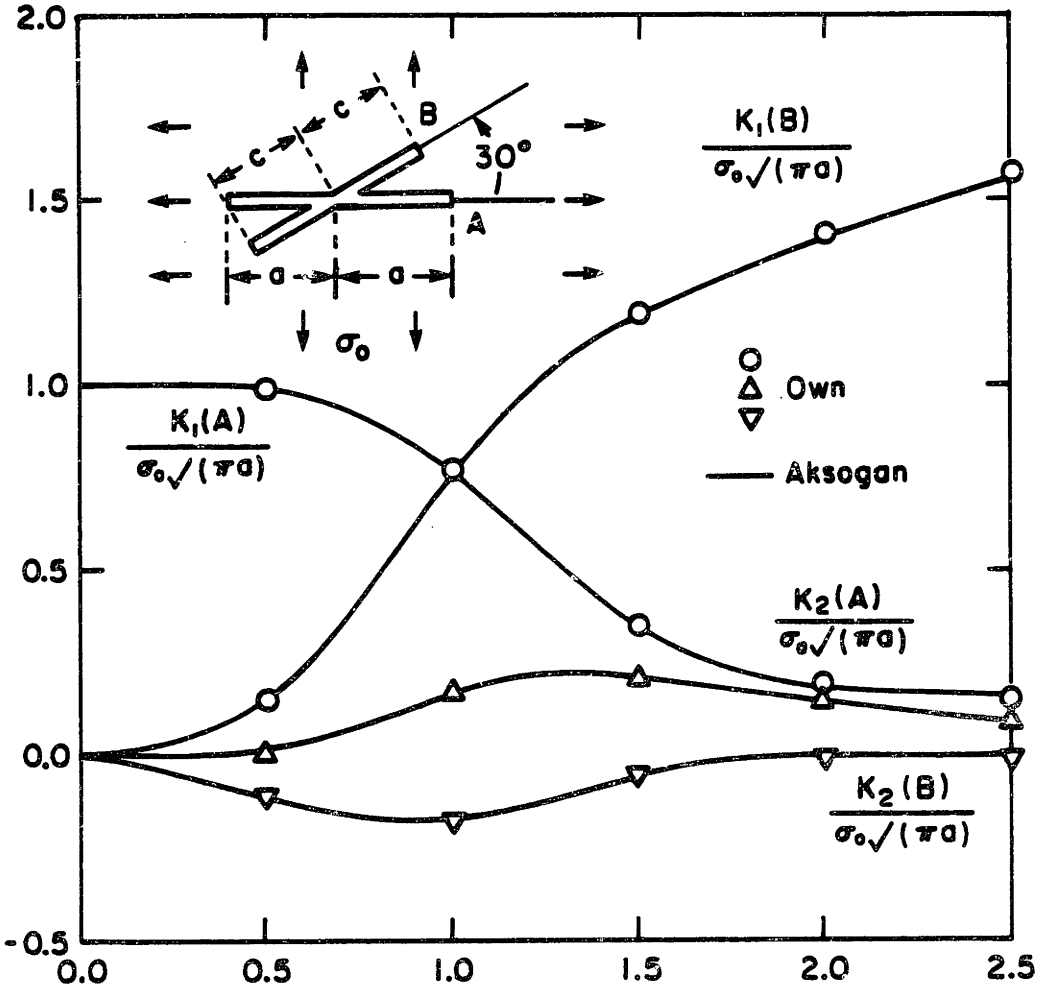


Figure 2.15 Comparison of stress intensity factors for an X-shaped crack with those from Aksogan [21].

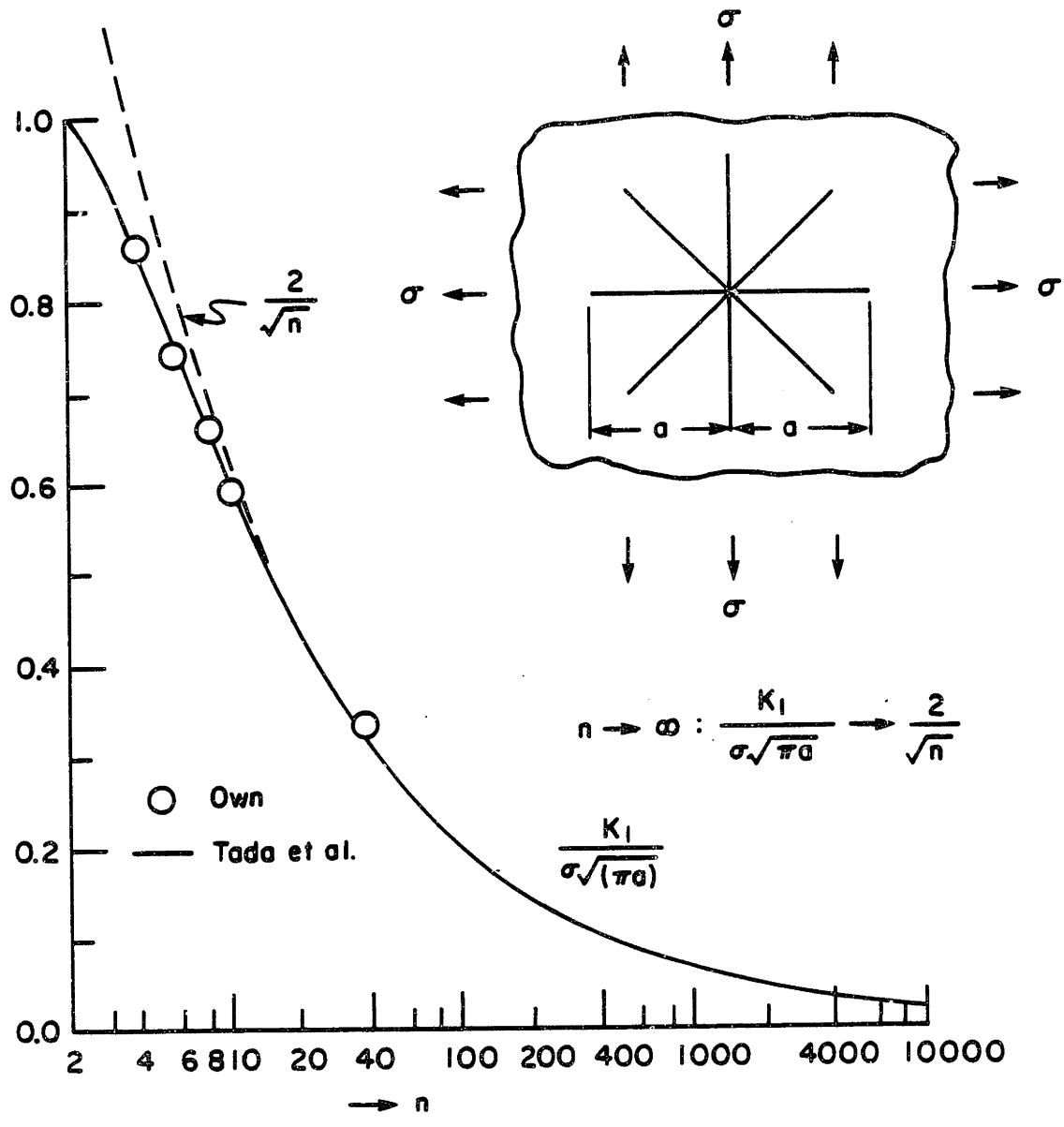


Figure 2.16 Stress intensity factors for a star crack as a function of the crack number. Comparison with those from Tada et al. [17].

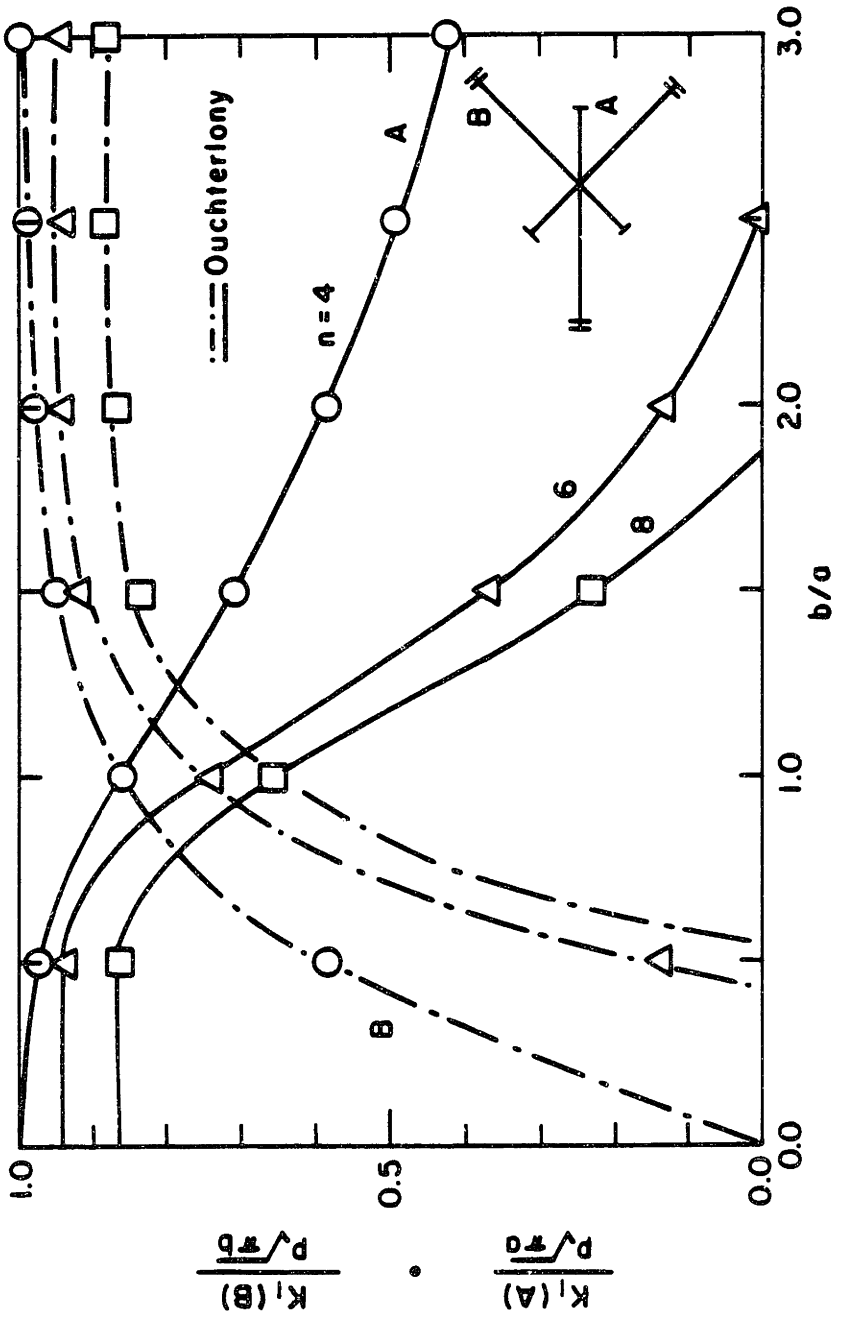


Figure 2.17 Stress intensity factors for pressurized radial crack with arms of alternating lengths, as a function of the crack length ratio for various crack numbers. Comparison with those from Ouchterlony [22].

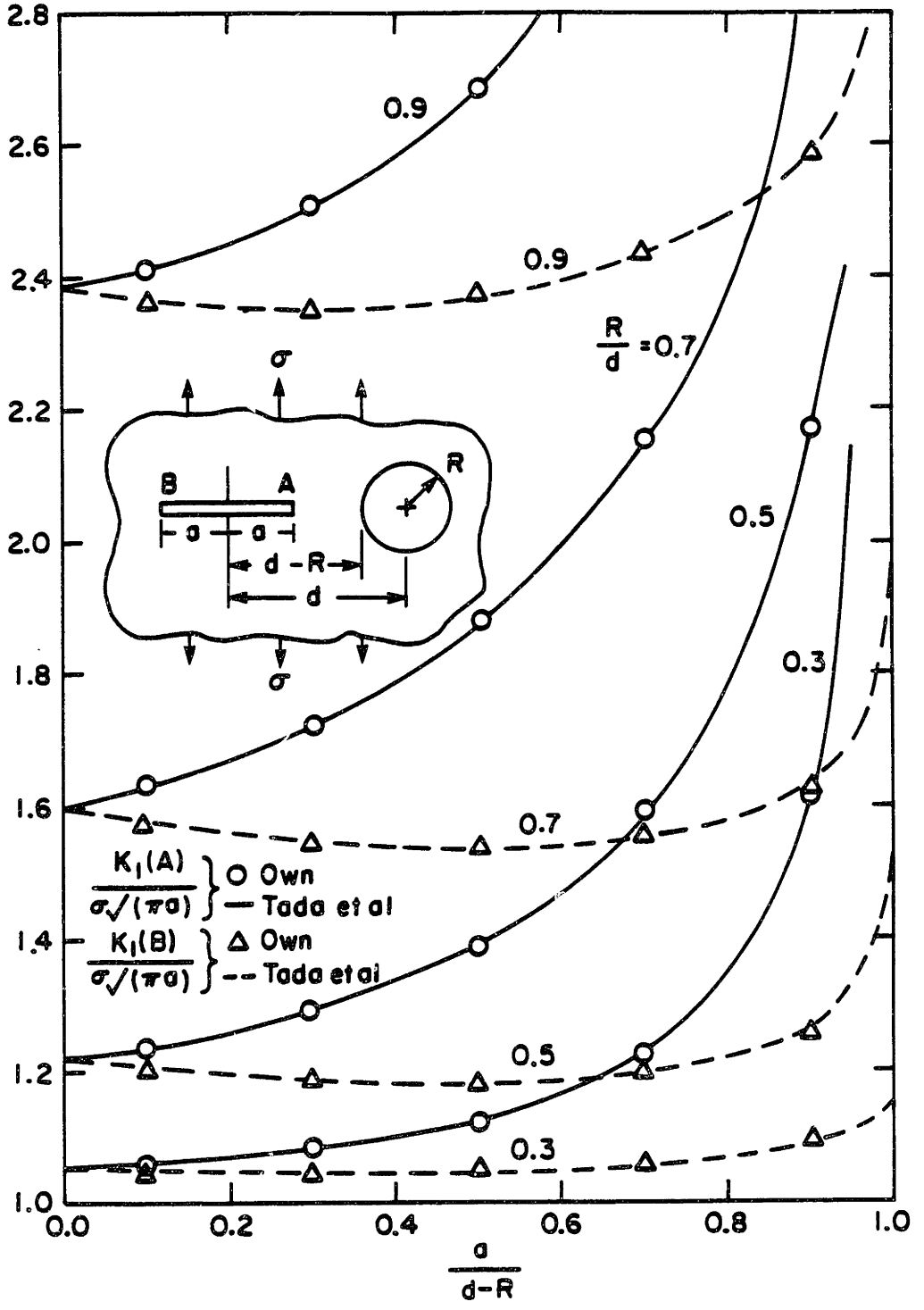


Figure 2.18 Comparison of stress intensity factors for a crack near a hole with those from Tada et al. [17].

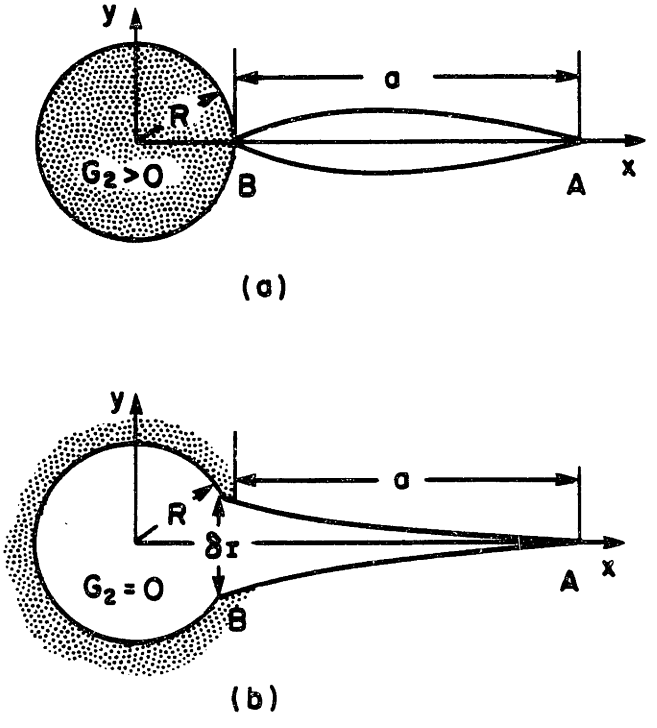
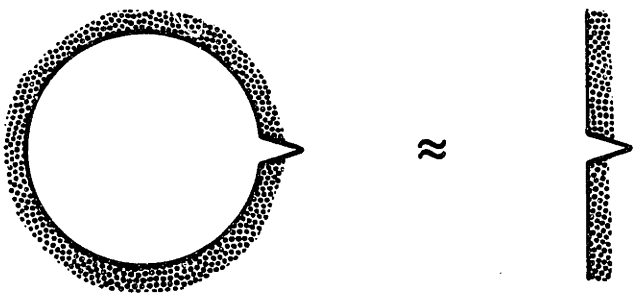


Figure 2.19 Effect of inclusion modulus on crack tip displacement. (a) Tip closed for non-zero shear modulus; (b) tip has displacement δ_I when inclusion is a hole.



(a) Surface crack approximation

(b) Completely embedded crack approximation

Figure 2.20 Approximations for a crack emanating from a hole. (a) If the crack length to hole radius is small, the crack may be considered as a surface crack; (b) If the tip opening displacement δ_I is small relative to the maximum opening displacement δ_{max}, the tip terminating at the hole may be approximated to be closed.

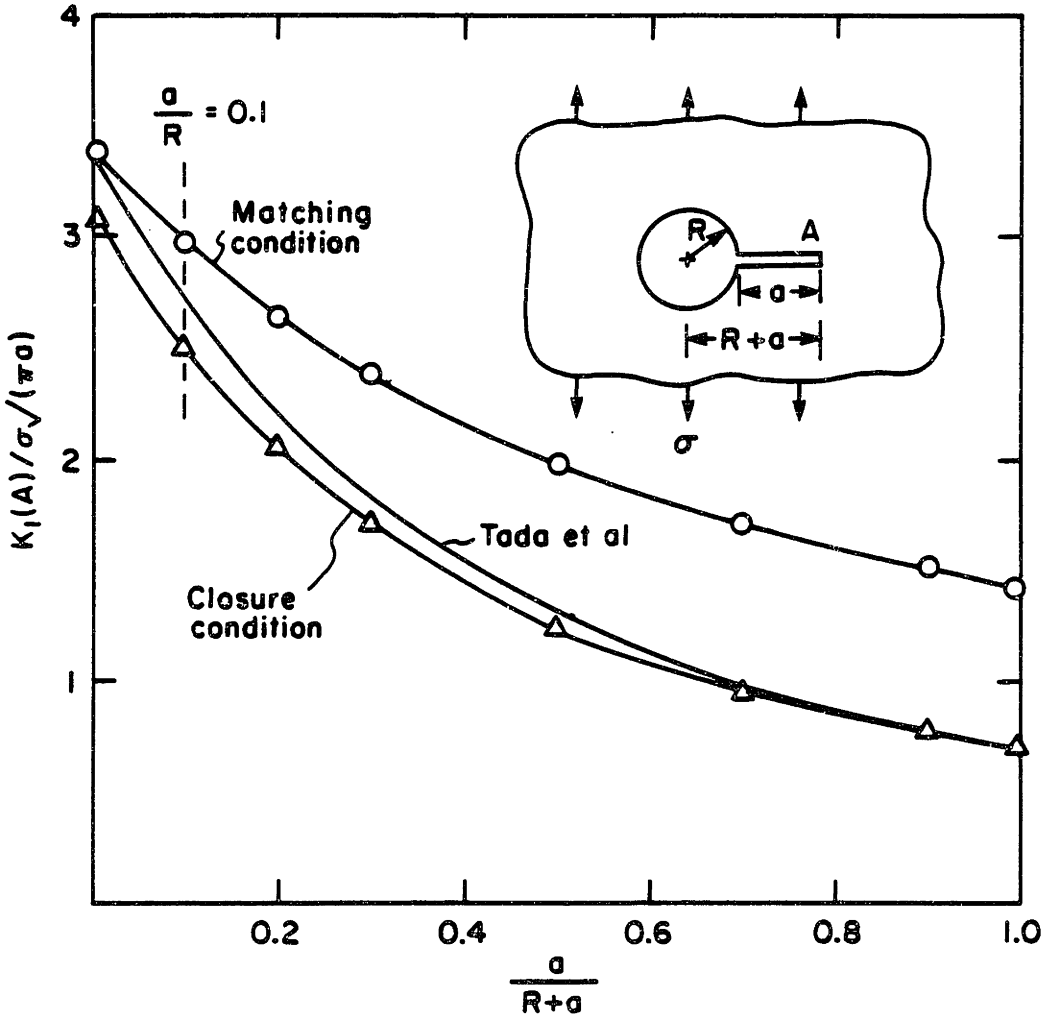


Figure 2.21 Comparison of the stress intensity factors obtained by using the closure condition (Eqn. 2.6) and the free-surface matching condition (Eqn. 2.7) with those from Tada et al. [17]. The closure condition gives more accurate results for $a/R \leq 0.1$, while the matching condition is more appropriate when $a/R > 0.1$.

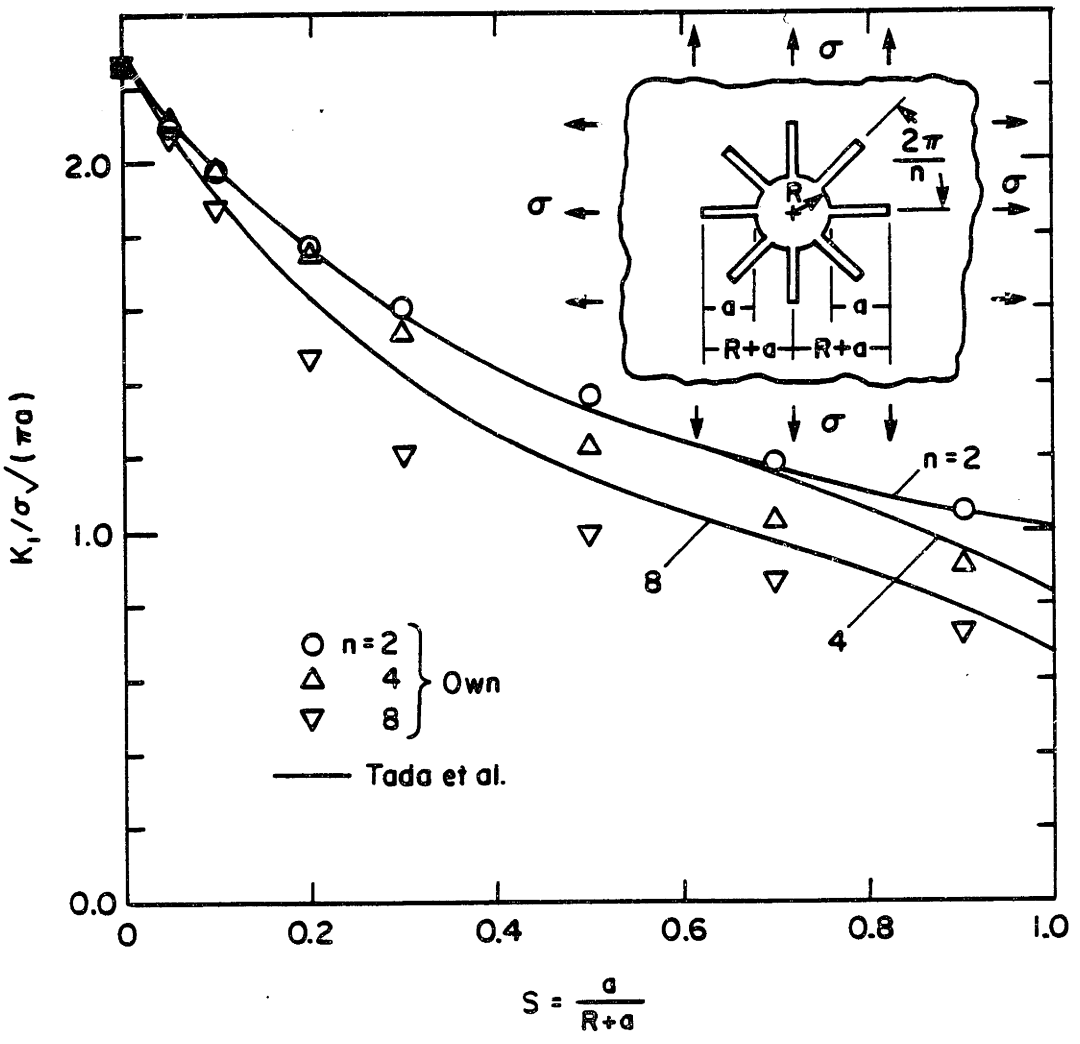


Figure 2.22 Stress intensity factors for radial cracks emanating from a hole, under uniform all-round tension, as a function of crack length for various crack numbers. Comparison with those from Tada et al. [17].

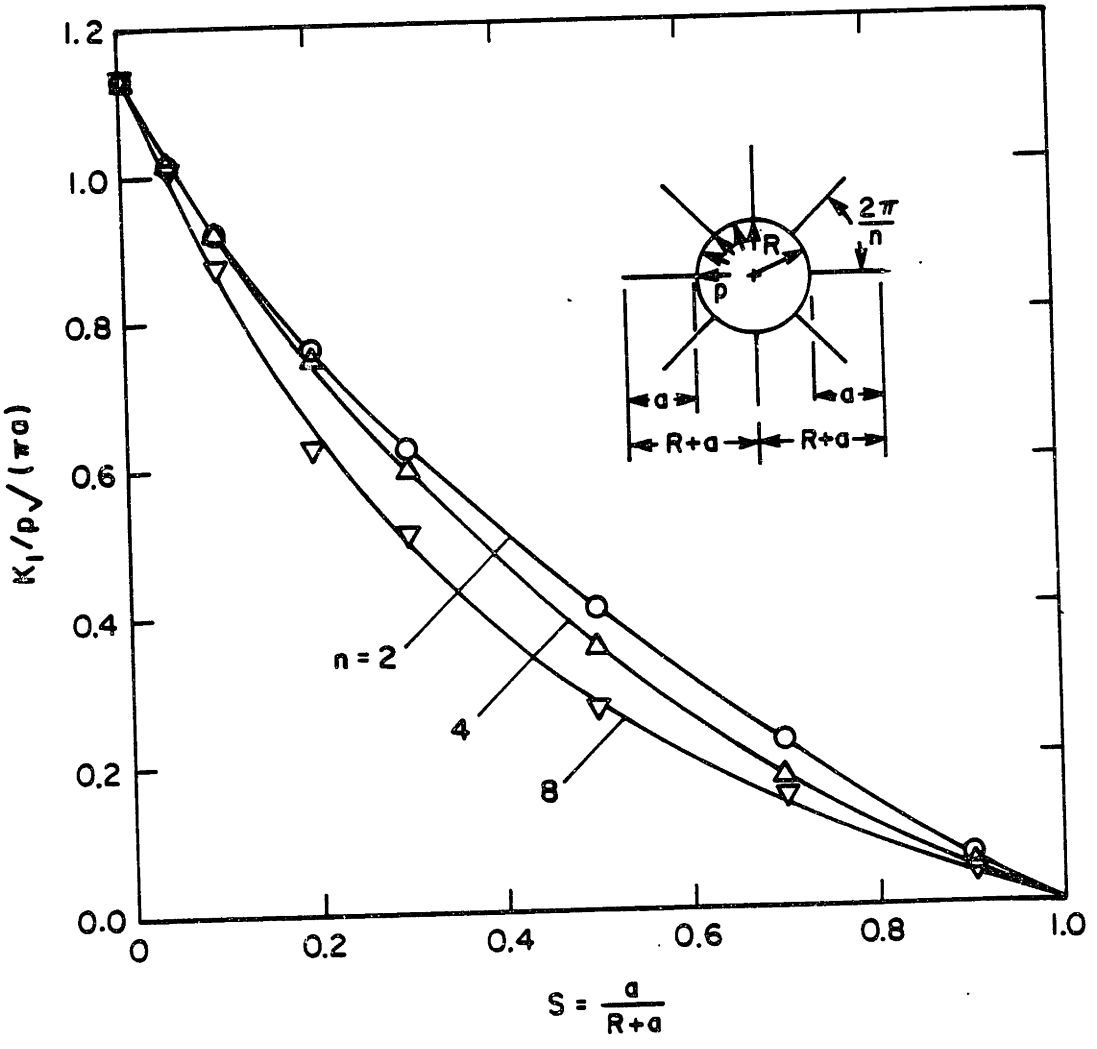


Figure 2.23 Stress intensity factors for empty radial cracks emanating from a pressurized circular hole, as a function of crack length for various crack numbers.

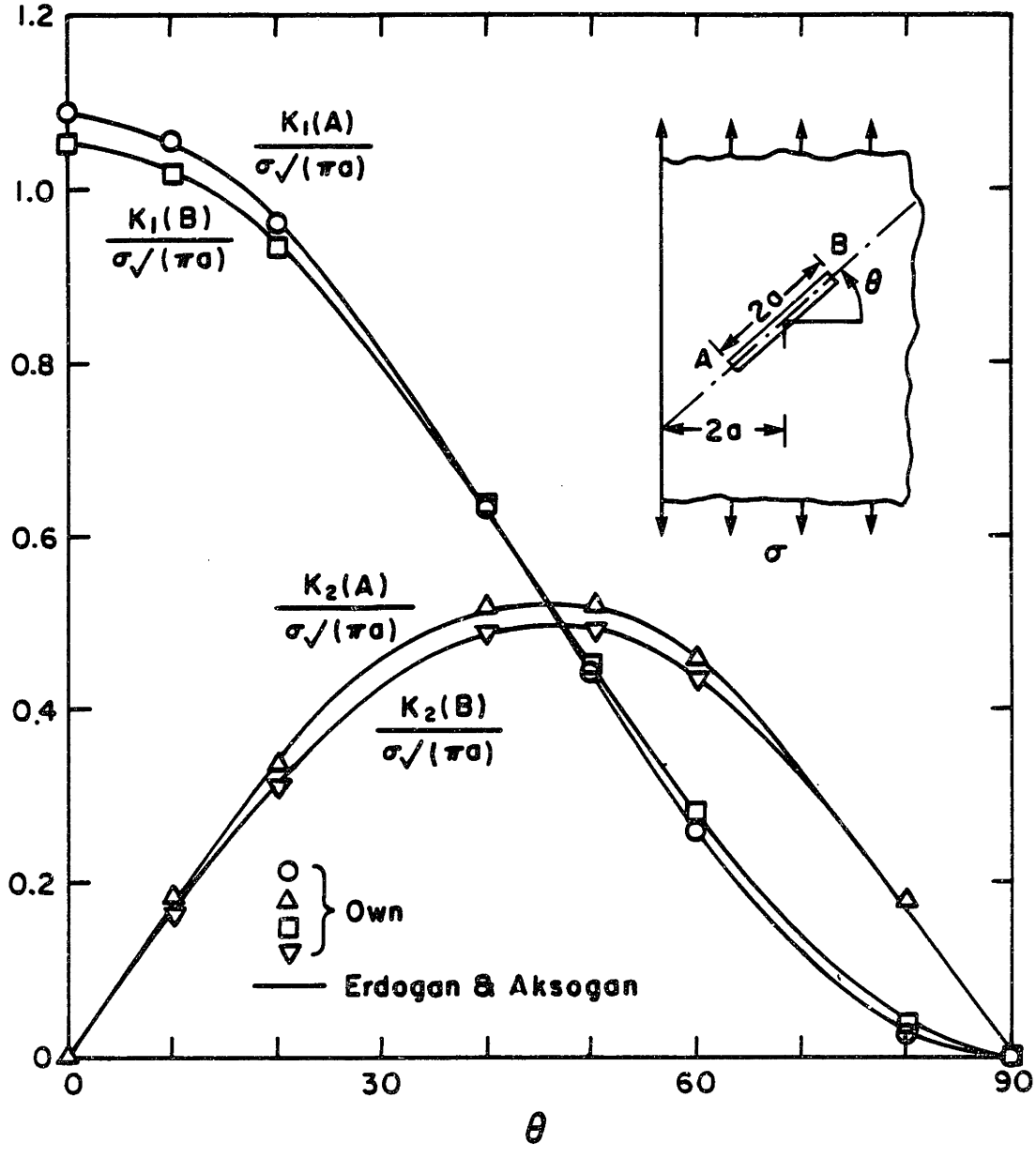


Figure 2.24 Stress intensity factors for an arbitrarily oriented crack in a half plane loaded parallel to the free boundary. Comparison with those from Erdogan and Aksogan [4].

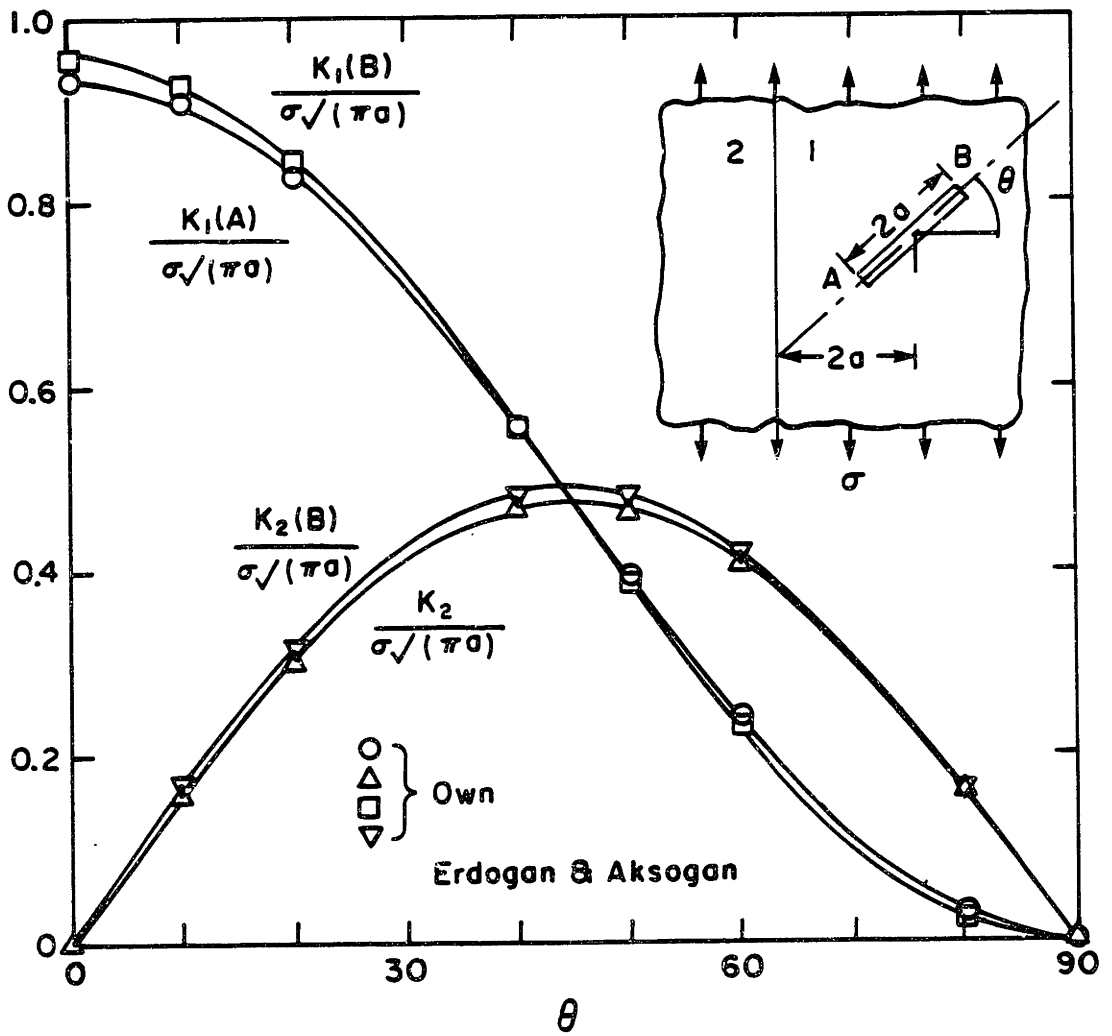


Figure 2.25 Stress intensity factors for an arbitrarily oriented crack in a bounded half plane loaded parallel to the interface ($G_2/G_1=22.22$, $\nu_1=0.35$, $\nu_2=0.3$). Comparison with those from Erdogan and Aksogan [4].

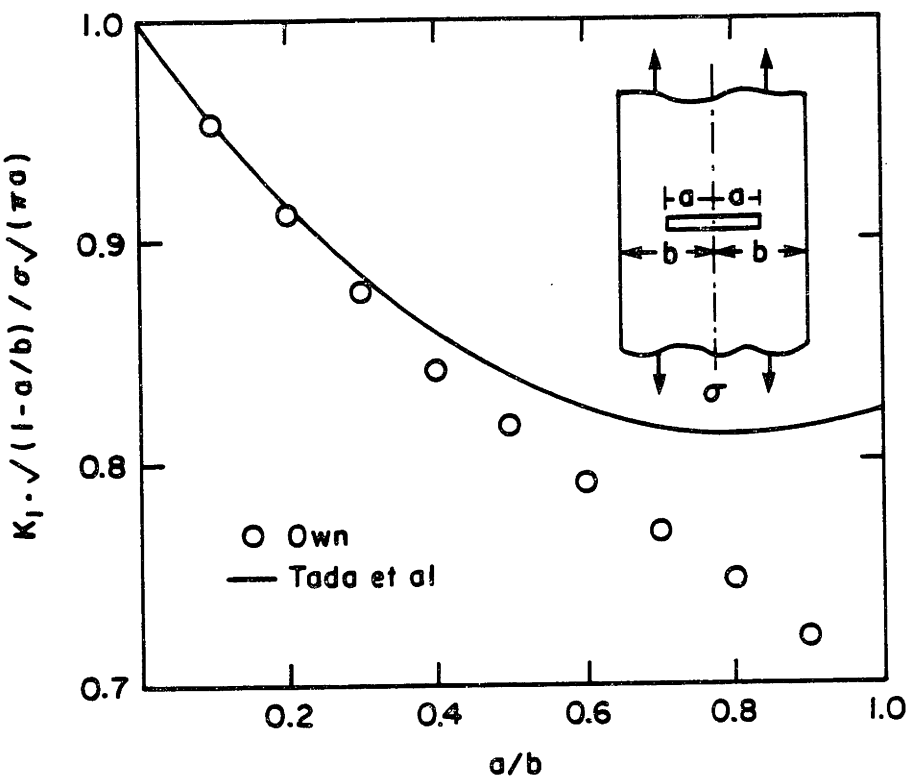


Figure 2.26 Comparison of stress intensity factors for center cracked infinite strip with those from Tada et al. [17]. The approximate double interface influence function is more accurate for smaller ratios of crack length to strip width.

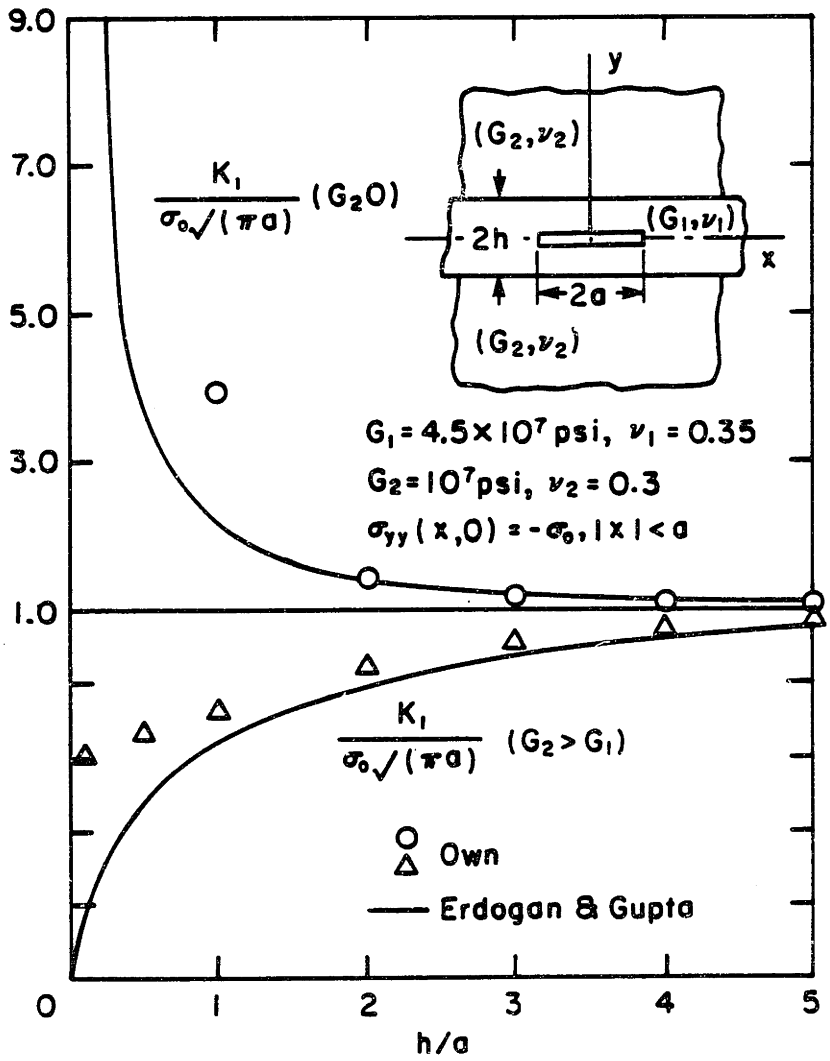


Figure 2.27 Comparison of stress intensity factors for center crack in an elastic layer bonded to two identical half planes with those from Erdogan and Gupta [5].

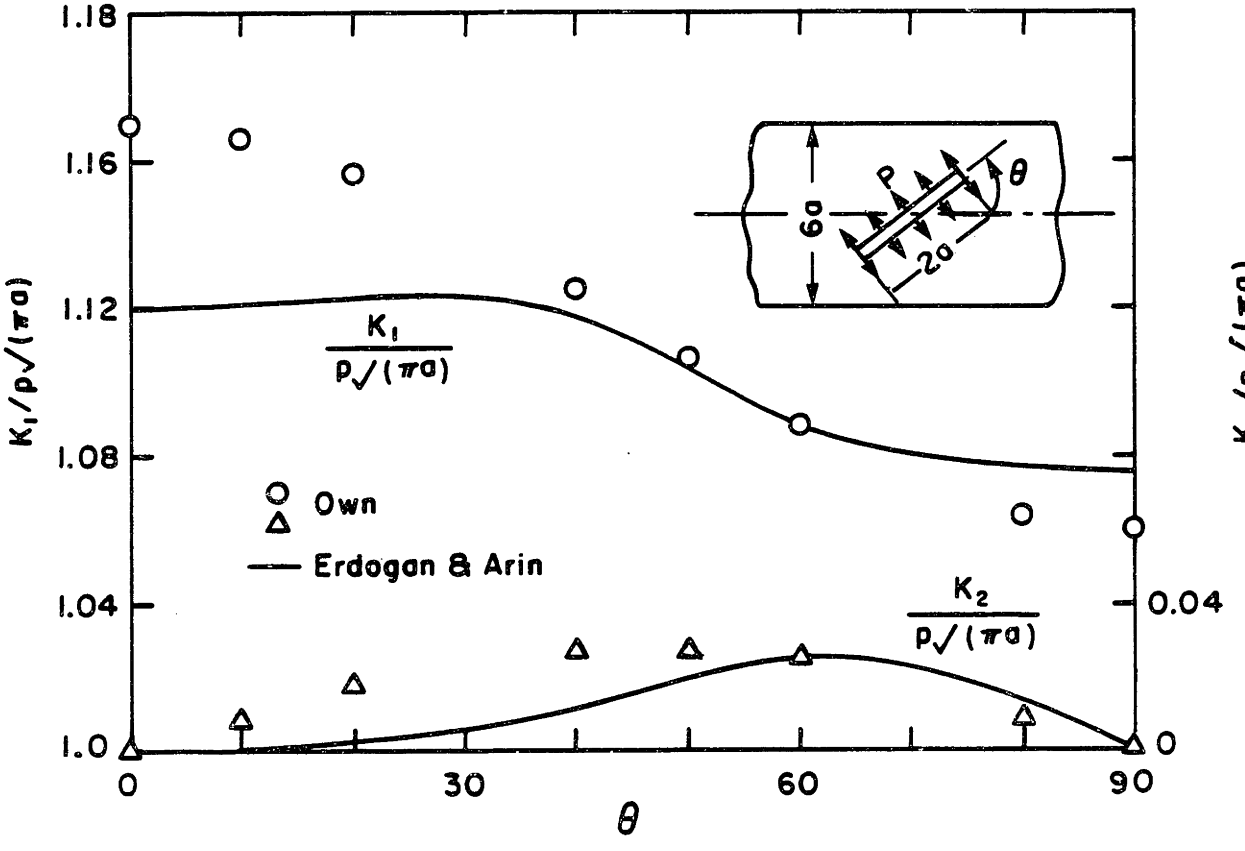


Figure 2.28 Comparison of stress intensity factors for a crack in an infinite strip under uniform internal pressure with those from Erdogan and Arin [6].

CONCLUSIONS

A general purpose numerical scheme has been developed for the elastostatic analysis of multiple arbitrarily shaped cracks, in plane inhomogeneous regions and under arbitrarily applied stresses. The scheme uses distributions of edge dislocations to formulate integral expressions for the crack tractions. For two-dimensional problems, approximate integration formulas employing Chebyshev polynomials were used to reduce these equations to a system of linear algebraic equations for the crack tractions in terms of the unknown dislocation densities. Appropriate conditions were developed for closing the set of equations governing cracks intersecting inhomogeneity boundaries; these closure/matching conditions allow us to employ the Chebyshev scheme even when the strength of the stress singularity at a crack tip is no longer square-root. The stress and displacement fields associated with the cracked medium can be calculated once the density distributions have been determined. The scheme is efficient, physically tractable, and more general than other available formulations and techniques. A listing of the FORTRAN program used to implement the numerical techniques described in this study in a single framework is available in Appendix D.

Besides providing a powerful tool for elastostatic fracture mechanics, this numerical scheme can also form

the starting point for the study of crack initiation and propagation in inhomogeneous media. In particular, we have used the scheme in developing a general purpose numerical simulator to describe the growth and interaction of multiple hydraulic fractures [33]; it is also the basis for a finite element surface integral hybrid scheme [34], that will allow us to analyze problems of curvilinear crack growth and interaction in finite bounded regions.

REFERENCES

[1] Barr, D.T. and M.P. Cleary, Thermoelastic fracture solutions using distributions of singular influence functions-I, to appear in Int. J. Solids Structures.

[2] Erdogan, F, G.D. Gupta and M. Ratwani, Interaction between a circular inclusion and an arbitrarily oriented crack, J. Appl. Mech., 41, 1007-1013 (1974).

[3] Erdogan, F. and G.D. Gupta, The inclusion problem with a crack crossing the boundary, Int. J. Fracture, 11, 13-27 (1975).

[4] Erdogan, F. and O. Aksogan, Bonded half planes containing an arbitrarily oriented crack, Int. J. Solids Structures, 10, 569-585 (1974).

[5] Erdogan, F. and G.D. Gupta, The stress analysis of multi-layered composites with a flaw, Int. J. Solids Structures, 7 , 39-61 (1971).

[6] Erdogan, F. and K. Arin, A half plane and a strip with an arbitrarily located crack, Int. J. Fracture, 11, 191-204 (1975).

[7] Segall, P. and D.D. Pollard, Mechanics of discontinuous faults, J. Geophys. Res., 85, 4337-4370 (1980).

[8] Banichuk, N.V. Determination of the form of a curvilinear crack by small parameter technique, Izv. AN SSSR, MTT 7, 130-137 (1970).

[9] Goldstein, R.V. and R.L. Salganik, Planar problem of curvilinear cracks in an elastic solid, Izv. AN SSSR, MTT 5, 69-82 (1970).

[10] Goldstein, R.V. and R.L. Salganik, Determination of crack opening and stress intensity coefficients for a smooth curvilinear crack in an elastic plane, Izv. AN SSSR, MTT 7, 69-78 (1972).

[11] Cotterell, B. and J.R. Rice, Slightly curved or kinked cracks, Int. J. Fracture, 16, 155-169 (1980).

REFERENCES (cont'd)

[12] Hanson, M.E., G.D. Anderson, R.J. Schaffer, D.N. Montan, B. Haimson and M.P. Cleary, LLL Gas Stimulation Program, Quarterly Progress Reports No. URCL 536-78-1,2 from Lawrence Livermore Laboratories, January and April (1978).

[13] Wong, S.K., Numerical analysis of axisymmetric and other crack problems related to hydraulic fracturing, S.M. Thesis, MIT (1981).

[14] Cleary, M.P., Primary factors governing hydraulic fractures in heterogeneous stratified porous formations, ASME Paper 78-Pet-47 (1978).

[15] Erdogan, F., and G.D. Gupta, On the numerical integration of singular integral equations, Quart. Appl. Math., 525-534 (1972).

[16] Cleary, M.P., Moving singularities in elasto-diffusive solids with applications to fracture propagation, Int. J. Solids Structures, 14, 81-97 (1978).

[17] Tada, H., P.C. Paris and G.R. Irwin, The stress analysis of cracks handbook, Del Research Corp., Hellertown, PA (1973).

[18] Yokobori, T.M. Uozumi and M. Ichikawa, Interaction between non-coplanar parallel staggered elastic cracks, in Rep. Inst. Strength and Fracture of Materials, 7, 25-47, Tohoku University (1971).

[19] Lam, K.Y., General branching and frictional slippage at crack tips with aplications to hydraulic fracturing, S.M. Thesis, MIT (1982).

[20] Ramachandra Rao, B.S., Infinite elastic plane with a parabolic arc cut, Appl. Sci. Res. Ser. A., 12, 86-90 (1963).

[21] Aksogan, O., The stress intensity factors for x-formed array of cracks, in Fracture 1977, Advances in Research on the Strength and Fracture of Materials (ed., D.M.R. Taplin), vol. 3A, Pergamon Press (1978).

[22] Ouchterlony, F., Fracture mechanics applied to rock blasting, in Proc. 3rd Intl. Cong. of Soc. for Rock Mech., Denver (1973), pub. by Natl. Acad. Sci., Wash. D.C., vol. II, part B, 1374-1383 (1974).

REFERENCES (cont'd)

[23] Bhargava, R.D., and R.J. Bhargava, Elastic circular inclusion in an infinite plane containing two cracks, Int. J. Engng. Sci., 11, 437-449 (1973).

[24] Sheng, C.F. and L. Wheeler, Crack path prediction for a kinked crack in the neighborhood of a circular inclusion in an infinite medium, ASME Paper 81-APM-31 (1981).

[25] Dundurs, J., and T. Mura, Interaction between an edge dislocation and a circular inclusion, J. Mech. Phys. Solids, 12, 177-189 (1964).

[26] Dundurs, J. and G.P. Sendeckyj, Edge dislocation inside a circular inclusion, J. Mech. Phys. Solids, 13, 141-147 (1965).

[27] Muskhelishvili, N.I., Some basic problems of the mathematical theory of elasticity, Noordhoff (1953).

[28] Hsu, Y.C., The infinite sheet with cracked cylindrical hole under inclined tension or in-plane shear, Int. J. Fracture, 11, 571-581 (1975).

[29] Grandt, A.F., Jr., Stress intensity factors for some through-cracked fastener holes, Int. J. Fracture, 11, 283-294 (1975).

[30] Ashbaugh, N., Stress solution for a crack at an arbitrary angle to an interface, Int. J. Fracture, 11, 205-219 (1975).

[31] Abramowitz, M. and I.A. Stegun (eds.), Handbook of mathematical functions, National Bureau of Standards, Appl. Math. Series 55 (1964).

[32] Narendran, V.M. and M.P. Cleary, Growth and interaction of multiple hydraulic fractures, to be submitted for publication.

[33] Annigeri, B., V.M. Narendran and M.P. Cleary, "A finite element surface integral hybrid scheme for crack growth and interaction in bounded plane regions," to be submitted for publication.

APPENDIX A

Coordinate Transformation and Surface Discretization Schemes

Consider the system of N cracks in the plane (Fig. A.1) where the n-th crack surface is described by

$$y_n = f_n(x_n); \quad x_{na} \leq x_n \leq x_{nb} \tag{A.1}$$

between the crack tips at $(x_{na}, f_n(x_{na}))$ and $(x_{nb}, f_n(x_{nb}))$. A dimensional curvilinear coordinate, t_n, may be defined along the surface as

$$t_n = \int_{x_{na}}^{x_{nk}} \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{1/2} dx$$

$$= \int_{x_{na}}^{x_{nk}} \left[1 + (f'_n)^2\right]^{1/2} dx \tag{A.2}$$

Note that the curvilinear coordinate has been defined to be zero at the tip $(x_{na}, f_n(X_{na}))$. The curvilinear length of the crack can be found from

$$L_n = \int_{x_{na}}^{x_{nb}} \left[1 + (f'_n)^2\right]^{1/2} dx \tag{A.3}$$

Using eqns. (A.2) and (A.3), a dimensionless curvilinear coordinate s_n is defined as

$$s_n = 2(t_n/L_n) - 1 \tag{A.4}$$

The dimensionless coordinate thus transforms the crack surface which lies between $(x_{na}, f_n(x_{na}))$ and $(x_{nb}, f_n(x_{nb}))$ into the interval (+1,-1). The dimensionless curvilinear coordinate ξ_m is analogous to s_n and is derived similarly.

Since most crack surfaces cannot be analytically expressed as required by eqn. (A.1), it is more convenient to specify a surface by a discrete number of piecewise linear segments (Fig. A.2). The i-th linear segment lies between (x_i, y_i) and (x_{i+1}, y_{i+1}), and the surface is discretized by specifying k end points. The crack surface length becomes

$$L_n = \sum_{i=1}^{k-1} \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right]^{1/2} \qquad (A.5)$$

while the piecewise linear distance of the j-th end point is

$$\ell_{nj} = \sum_{i=1}^{j-1} \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right]^{1/2} \qquad (A.6)$$

The dimensionless coordinates s_n and ξ_m are found as the zeros of the Chebyshev polynomials of the first and second kind respectively. For a given s_n or ξ_m then, an inverse transformation provides the corresponding Cartesian coordinates $(x(s_n), y(s_n))$ or $(x(\xi_m), y(\xi_m))$. Hence

$$x(s_n) = x_{j+1} - \frac{\ell_{(nj+1)} - L_n \cdot s_n}{\ell_{(nj+1)} - \ell_{nj}} \quad (x_{j+1} - x_j) \qquad \text{(A.7a)}$$

when

$$\ell_{nj} < L_n \cdot s_n < \ell_{n(j+1)} \quad . \qquad \text{(A.7b)}$$

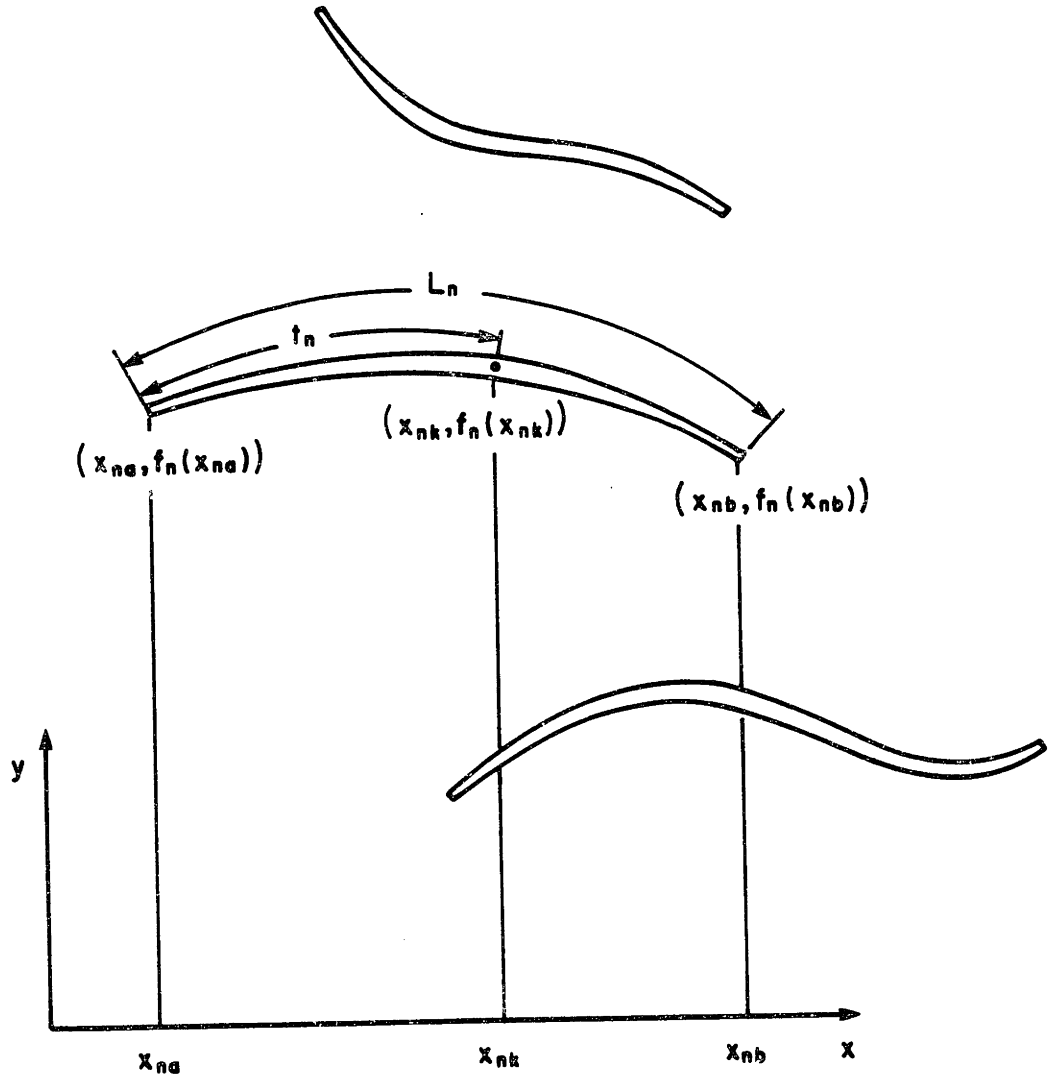
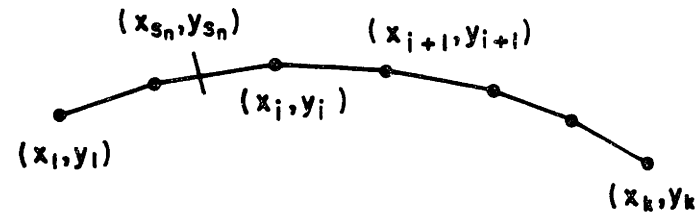


Figure A.1 Schematic for coordinate transformation scheme.



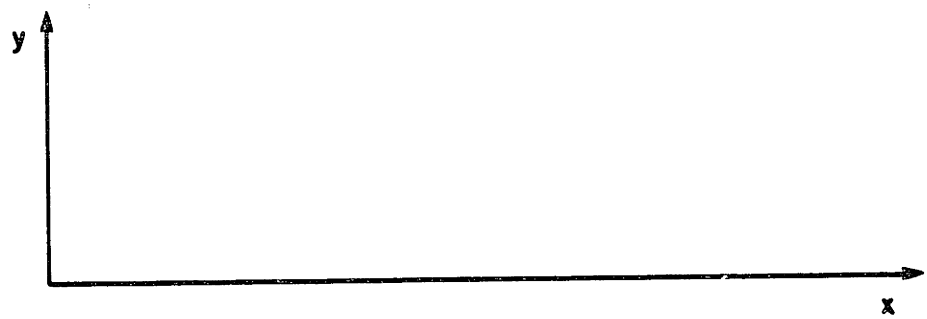


Figure A.2 Schematic for crack surface discretization scheme.

APPENDIX B

Edge Dislocation Influence Functions

The stresses σ_α at point (x,y) due to an edge dislocation of strength b_β at (x_d,y_d) in an isotropic, linearly elastic medium is given by the influence functions Γ_α^β. The solutions for an arbitrarily oriented dislocation in an infinite medium, or near a straight interface, or in a layer of finite width bonded between two half-planes, can be specialized from the influence functions for an edge dislocation near a circular inclusion.

I. Edge Dislocation near a Circular Inclusion

The most general form of the influence function is when the second material phase occurs as a long cylindrical inclusion (Fig. B.1a), but this may be specialized to the case of a straight interface (Fig. B.1b), where $\lambda \equiv d/a \to 1$, $r/a \to 1$. The case of an infinite medium may be obtained by simply inputting the same material properties for the inclusion and the matrix, hence A = B = 0. The influence functions are normalized with respect to the "crack-opening modulus" $\bar{E}$ ($\equiv E_1/4(1-\nu_1^2)$), derived from the Young's modulus E_1 and Poisson's ratio ν_1. Employing the parameters indicated in Figs. B.1-B.2, the influence functions for the stresses in the matrix may be written as (with the terms in the curley parentheses disappearing in the limit of a

straight interface):

$$\Gamma^{x}_{xx} = \frac{\bar{E}b_x}{2\pi}\left[- \frac{2y_1}{r_1^2}\left(\frac{2x_1^2}{r_1^2} + 1\right) + \left(A + B + 4A\frac{x_2^2}{r_2^2}\right)\frac{y_2}{r_2^2}\right.$$

$$- 2A\frac{(\lambda^2-1)a}{\lambda^3 r_2^2}\left[\frac{2x_2y_2}{r_2^2}\left(1 - \frac{4x_2^2}{r_2^2}\right) - \frac{(\lambda^2-1)ay_2}{\lambda r_2^2}\left(1 - \frac{4x_2^2}{r_2^2}\right)\right]$$

$$\left. - \left\{\left(A + B + 4A\frac{x^2}{r^2}\right)\frac{y}{r^2} + 2(B - A)\frac{a}{\lambda}\frac{xy}{r^4} - 2A\frac{a^2y}{r^4}\left(1 - \frac{4x^2}{r^2}\right)\right\}\right]$$

$$\Gamma^{y}_{xx} = \frac{\bar{E}b_y}{2\pi}\left[\frac{2x_1}{r_1^2}\left(\frac{2x_1^2}{r_1^2} - 1\right) + \left(B + A - 4A\frac{x_2^2}{r_2^2}\right)\frac{x_2}{r_2^2}\right.$$

$$+ A\frac{(\lambda^2-1)a}{\lambda^3 r_2^2}\left[2\lambda^2\left(\frac{2x_2^2}{r_2^2} - 1\right) - \left(3 - \frac{4x_2^2}{r_2^2}\right)\frac{4x_2^2}{r_2^2} + \frac{(\lambda^2-1)ax_2}{\lambda r_2^2}\left(6 - 8\frac{x_2^2}{r_2^2}\right)\right]$$

$$\left. - \left\{\frac{x}{r^2}\left(B + A - 4A\frac{x^2}{r^2}\right) + \left[A(2\lambda^2 - 1) + M(\kappa_2 + 1) - 1\right]\frac{a}{\lambda r^2}\left(2\frac{x^2}{r^2} - 1\right)\right.\right.$$

$$\left.\left. + A\frac{a^2x}{r^4}\left(6 - 8\frac{x^2}{r^2}\right)\right\}\right],$$

$$\Gamma^{x}_{yy} = \frac{\bar{E}b_x}{2}\left[\frac{2y_1}{r_1^2}\left(\frac{2x_1^2}{r_1^2} - 1\right) + \left(3A - B - 4A\frac{x_2^2}{r_2^2}\right)\frac{y_2}{r_2^2}\right.$$

$$- 2A\frac{(\lambda^2-2)a}{\lambda^3 r_2^2}\left[\frac{2x_2y_2}{r_2^2}\left(\frac{4x_2^2}{r_2^2} - 3\right) - \frac{(\lambda^2-1)ay_2}{\lambda r_2^2}\left(\frac{4x_2^2}{r_2^2} - 1\right)\right]$$

$$\left. - \left\{\left(3A - B - 4A\frac{x^2}{r^2}\right)\frac{y}{r^2} - 2(B - A)\frac{a}{\lambda}\frac{xy}{r^4} - 2A\frac{a^2y}{r^4}\left(\frac{4x^2}{r^2} - 1\right)\right\}\right] ,$$

$$\Gamma^{y}_{yy} = \frac{\bar{E}b_y}{2\pi}\left[2\left(3 - \frac{2x_1^2}{r_1^2}\right)\frac{x_1}{r_1^2} - \left(5A + B - 4A\frac{x_2^2}{r_2^2}\right)\frac{x_2}{r_2^2}\right.$$

$$- 2A\frac{(\lambda^2-1)a}{\lambda^3 r_2^2}\left[(2 - \lambda^2) - 2(5 - \lambda^2)\frac{x_2^2}{r_2^2} + 8\frac{x_2^4}{r_2^4} + \frac{(\lambda^2-1)ax_2}{\lambda r_2^2}\left(3 - 4\frac{x_2^2}{r_2^2}\right)\right]$$

$$+ \left\{\left(5A + B - 4A\frac{x^2}{r^2}\right)\frac{x}{r^2} - [A(2\lambda^2 - 1) + M(\kappa_2 + 1) - 1]\frac{a}{\lambda r^2}\left(1 - \frac{2x^2}{r^2}\right)\right.$$

$$\left. \left. - \frac{2Aa^2x}{r^4}\left(3 - \frac{4x^2}{r^2}\right)\right\}\right] .$$

$$\Gamma^{x}_{xy} = \frac{\bar{E}b_x}{2\pi}\left[\frac{2x_1}{r_1^2}\left(\frac{2x_1^2}{r_1^2} - 1\right) + \left(3A - B - 4A\,\frac{x_2^2}{r_2^2}\right)\frac{x_2}{r_2^2}\right.$$

$$- 2A\,\frac{(\lambda^2-1)a}{\lambda^3 r_2^2}\left[1 - 8\,\frac{x_2^2}{r_2^2} + 8\,\frac{x_2^4}{r_2^4} + \frac{(\lambda^2-1)ax_2}{\lambda r_2^2}\left(3 - \frac{4x_2^2}{r_2^2}\right)\right]$$

$$\left. - \left\{\left(3A - B - 4A\,\frac{x^2}{r^2}\right)\frac{x}{r^2} + (B - A)\left(1 - \frac{2x^2}{r^2}\right)\frac{a}{\lambda r^2} + \frac{2Aa^2x}{r^4}\left(3 - 4\,\frac{x^2}{r^2}\right)\right\}\right]$$

$$\Gamma^{y}_{xy} = \frac{\bar{E}b_y}{2\pi}\left[\frac{2y_1}{r_1^2}\left(\frac{2x_1^2}{r_1^2} - 1\right) + \left(B + A - 4A\,\frac{x_2^2}{r_2^2}\right)\frac{y_2}{r_2^2}\right.$$

$$+ A\,\frac{(\lambda^2-1)a}{\lambda^3 r_2^2}\left[4\lambda^2\,\frac{x_2y_2}{r_2^2} - 8\,\frac{x_2y_2}{r_2^2}\left(1 - \frac{2x_2^2}{\lambda r_2^2}\right) + 2\,\frac{(\lambda^2-1)a}{\lambda r_2^2}\,y_2\left(1 - \frac{4x_2^2}{r_2^2}\right)\right]$$

$$- \left\{\frac{y}{r^2}\left(B + A - 4A\,\frac{x^2}{r^2}\right) + 2\,[A(2\lambda^2 - 1) + M(\kappa_2 + 1) - 1]\,\frac{a}{\lambda}\,\frac{xy}{r^4}\right.$$

$$\left. \left. + Aa^2\,\frac{2y}{r^4}\left(1 - \frac{4x^2}{r^2}\right)\right\}\right] .$$

In the case of an arbitrarily oriented dislocation (Fig. B.2), the influence functions in the global x-y coordinate system are obtained from the transformation

$$\begin{Bmatrix} \Gamma^X \\ \Gamma^Y \end{Bmatrix} = \begin{bmatrix} -\sin\alpha & -\cos\alpha \\ \cos\alpha & -\sin\alpha \end{bmatrix} \begin{Bmatrix} \Gamma^x \\ \Gamma^y \end{Bmatrix}$$

II. Edge Dislocation Interacting with Two Interfaces

Based on the solutions for an edge dislocation near a single interface (see above), Wong [13] has proposed an approximation to the influence functions for an edge dislocation interacting with two interfaces (Fig. B.3).

For an edge dislocation in a homogeneous medium, the influence functions $\Gamma_H(\xi_m, s_n)$ give the stresses at ξ_m due to a dislocation at s_n. To account for the first nearby interface I, however, the appropriate influence function is $\Gamma_I(\xi_m, s_n)$. Let us assume for the moment that the medium with the interface I is homogeneous and bring in a second interface II. If the influence function for an edge dislocation near interface II is $\Gamma_{II}(\xi_m, s_n)$, the overall influence function $\Gamma_L(\xi_m, s_n)$ for an edge dislocation interacting with two interfaces may be approximated as

$$\Gamma_L(\xi_m, s_n) = \frac{\Gamma_I(\xi_m, s_n)\,\Gamma_{II}(\xi_m, s_n)}{\Gamma_H(\xi_m, s_n)}$$

In effect, the influence function Γ_I has been modified to represent a homogeneous material by dividing it with Γ_H, and the effect of the second interface is then introduced via a multiplicative procedure. An alternate approximation can be obtained from an additive procedure, whereby

$$\Gamma_L(\xi_m, s_n) = \Gamma_I(\xi_m, s_n) - \Gamma_H(\xi_m, s_n) + \Gamma_{II}(\xi_m, s_n)$$

Note that both these approximate solutions have the desired symmetry of allowing the interfaces I and II to be interchanged. They can also be specialized to yield the influence functions for an edge dislocation in a homogeneous medium or near a single interface. The homogeneous case is achieved by simply inputting same material properties for all three regions separated by the two interfaces. On the other hand, by having the material constants for region 2 (or 3) the same as those for region 1 and merging the locations of the two interfaces, both the approximate solutions reduce to the influence function for an edge dislocation near a single interface.

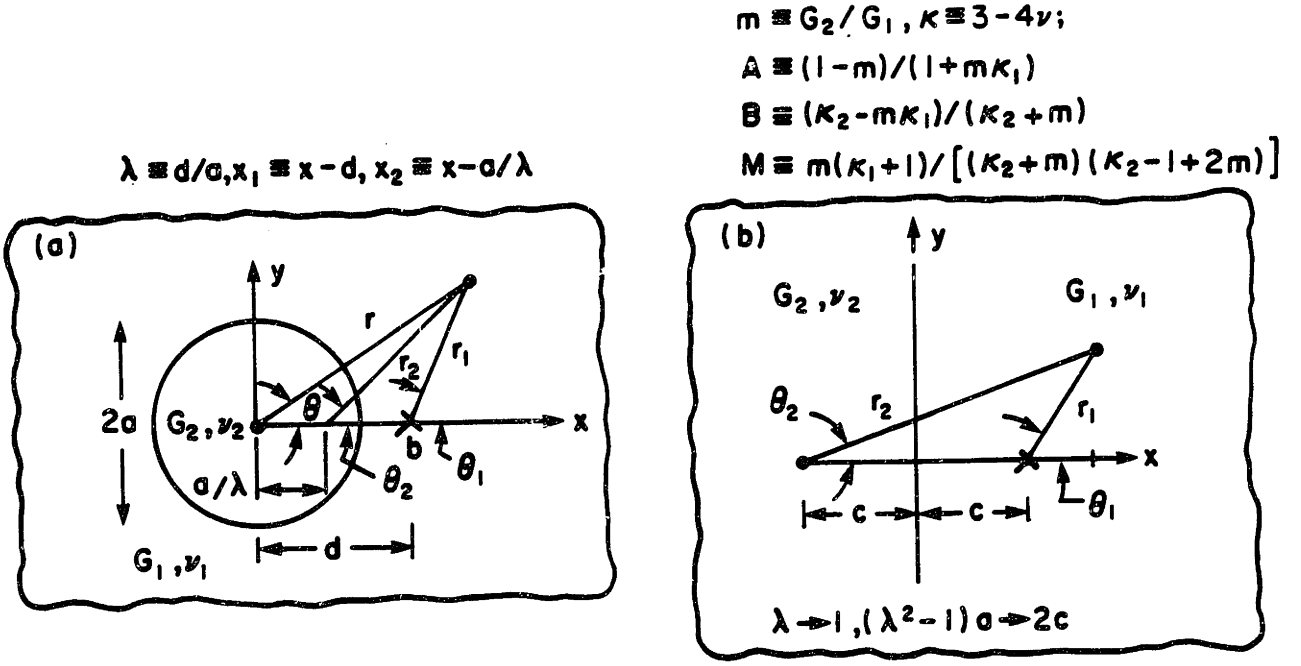


Figure B.1 (a) Edge dislocation $b=b_x+ib_y$ near a cylindrical inclusion;
(b) Edge dislocation near a straight interface between inhomogeneous strata.

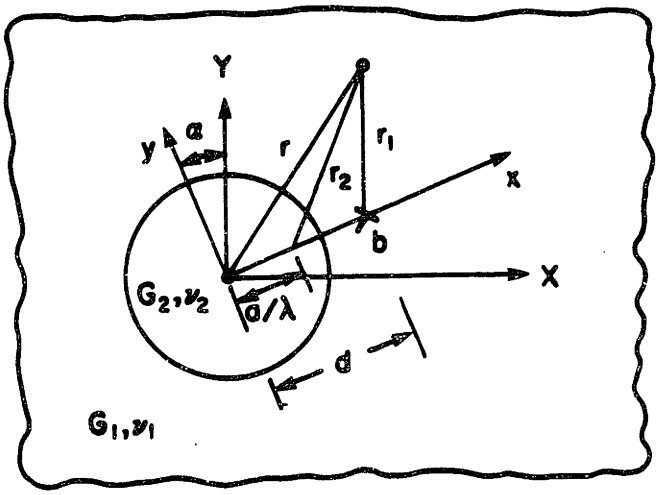


Figure B.2 Arbitrarily oriented dislocation near a cylindrical inclusion.

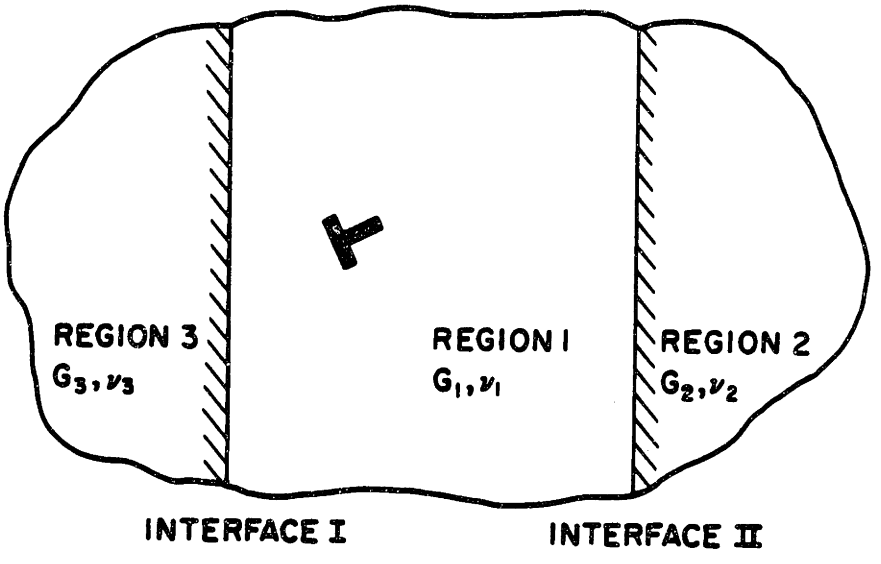


Figure B.3 Edge dislocation in an isotropic layer of finite width bonded between two isotropic half-planes.

APPENDIX C

Numerical Integration Techniques

I. Gauss-Chebyshev Integration Formula

This section describes the solution of the singular integral equation of the kind in eqn. (1.8). The solution employs interpolation functions which are appropriate to the domain in question and to the singularity in the kernel.

Consider first the integral relations given by Abramowitz and Stegun [31]:

$$\text{P.V.}\int_{-1}^{+1} \frac{T_n(s)\,ds}{(s-\xi)\sqrt{(1-s^2)}} = \pi U_{n-1}(\xi) \quad , \; n > 0 \tag{C.1}$$

$$\int_{-1}^{+1} \frac{h(s)\,ds}{\sqrt{(1-s^2)}} \simeq \frac{\pi}{M} \sum_{k=1}^{M} \frac{h(s_k)}{M} \quad , \; s_k = -\cos\frac{\pi(2k-1)}{2M} \tag{C.2}$$

where T_n, U_n are the Chebyshev polynomials defined as

$$T_n(s) = \cos n\theta, \; U_n(s) = \frac{\sin(n+1)\theta}{\sin\theta} \quad , \; s \equiv \cos\theta \tag{C.3}$$

and the nodal points s_k are the zeros for the polynomials of the first kind $T_M(s)$.

The integral equation of interest here is of the form

$$\sigma(\xi) = \frac{1}{\pi} \int_{-1}^{+1} \frac{f(s)}{\sqrt{(1-s^2)}} [(s-\xi)^{-1} + k(\xi,s)]ds, \quad -1 \leq \xi \leq 1 \qquad (C.4)$$

where $k(\xi,s)$ is a bounded smooth kernel and $\delta'(s) = f(s)/\sqrt{(1-s^2)}$ is the dislocation density. The unknown function $f(s)$ may be approximated by a series of the Chebyshev polynomials

$$f(s) \simeq \sum_{n=1}^{N} B_n T_n(s) \qquad (C.5)$$

The integral equation then takes the form

$$\sum_{k=1}^{M} k(\xi,s_k) f(s_k) + M \sum_{n=1}^{N} B_n U_{n-1}(\xi) = M\sigma(\xi)/\pi \qquad (C.6)$$

which may be further reduced to linear algebraic equations for the unknowns B_n,

$$\sum_{n=1}^{N} \left[\sum_{k=1}^{M} k(\xi_r,s_k) T_n(s_k) + M U_{n-1}(\xi_r) \right] B_n = M\sigma(\xi r)/\pi$$
$$(r = 1,\ldots,N) \qquad (C.7)$$

Use is now made of the formula established by Erdogan and Gupta [15], namely

$$U_{n-1}(\xi_r) = \sum_{k=1}^{M} \frac{T_n(s_k)}{M(s_k - \xi_r)}, \quad \xi_r = -\cos\left(\frac{\pi r}{M+1}\right) \qquad (C.8)$$

which simplifies (C.7) to the form

$$\sum_{k=1}^{M}\left[(s_k-\xi_r)^{-1}+k(\xi_r,s_k)\right] f(s_k) = M\sigma(\xi_r)/\pi \ , \qquad r = 1,\ldots,M-1 \quad \text{(C.9)}$$

If the value M_n is specified for each of the N cracks, eqns. (C.9, 1.5) may be combined to give eqn. (1.9).

II. Integration Formula Based on Chebyshev Series Approximation of the Integrand

An integration formula is developed for integrals whose integrands contain a $(1-s^2)^{-1/2}$ singularity, i.e.,

$$\int_a^b \frac{f(s)ds}{\sqrt{(1-s^2)}} \qquad \text{(C.10)}$$

where f(s) is defined in the domain (-1,+1), and a and b, both being inside the interval (-1,+1), are the lower and upper limits of the integration respectively.

As in eqn. (C.5), the nonsingular part f(s) can be approximated by a Chebyshev series

$$f(s) \simeq \sum_{m=0}^{M_n} a_m T_m(s) \qquad \text{(C.11)}$$

where $T_m(s)$ is the m-th order Chebyshev polynomial

$$T_m(s) \equiv -\cos(m\cos^{-1}s) \qquad \text{(C.12)}$$

The coefficients a_m are determined from the orthogonal property of the Chevyshev polynomial, namely

$$a_m = \frac{A}{\pi} \int_{-1}^{+1} \frac{f(s)T_m(s)}{\sqrt{(1-s^2)}}\, ds \qquad (C.13)$$

where A = 1 for m = 0 and A = 2 for m > 0. Following eqn. (C.2), these coefficients may be approximated from

$$\int_{-1}^{+1} \frac{f(s)ds}{\sqrt{(1-s^2)}} \simeq \frac{\pi}{M_n} \sum_{k=1}^{M_n} f(s_k) \quad , \quad s_k = -\cos\frac{(2k-1)\pi}{2M_n} \qquad (C.14)$$

Combining eqns. (C.10-C.12), and transforming the variable of integration from s to θ through $s \equiv \cos\theta$, leads to

$$\int_a^b \frac{f(s)ds}{\sqrt{(1-s^2)}} \simeq \sum_{m=0}^{M_n} \frac{a_m}{m} [\sin(m\cos^{-1}a) - \sin(m\cos^{-1}b)] \qquad (C.15)$$

The approximation in eqn. (C.14) finally gives

$$\int_a^b \frac{f(s)}{\sqrt{(1-s^2)}} \simeq \frac{2}{M_n} \sum_{k=1}^{M_m} \Big\{ 1/2[\cos^{-1}a - \cos^{-1}b] + \sum_{m=1}^{M_n-1} \frac{T_m(s_k)}{m} [\sin(m\cos^{-1}a) - \sin(m\cos^{-1}b] \Big\} f(s_k) \qquad (C.16)$$

where

$$T_m(s_k) = \cos\left(\frac{m(2M_n-2k+1)}{2M_n}\right) \qquad (C.17)$$

Notice that when $a = +1$ and $b = s_n$, eqn. (C.17) transforms eqn. (1.14) into (1.15).

APPENDIX D

Listing of Computer Program

```
C     ****************************************************************
C
C                        E L A S T I C . F O R T R A N
C
C                     A GENERAL PURPOSE PROGRAM FOR
C                     PLANE STATIC ELASTICITY ANALYSIS
C
C     ****************************************************************
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &               TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA3/ TK(200),ZR(200)
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
C
      PARAMETER IRTELL=90
C
      REWIND IRTELL
      READ (IRTELL,900) IRDATA,IWRITE,IRCONFIG,
     &                  IRINCLU,IRTWOFACE
  900 FORMAT (I3)
C
      REWIND IR
      REWIND IRCONFIG
      REWIND IRINCLU
      REWIND IRTWOFACE
C
      CALL INIT
C
      CALL INPUT (IR,IRINCLU,IRTWOFACE)
C
      CALL CONFIGRTN (IRCONFIG)
C
      CALL CALCDATA
C
      CALL ZRTK
C
      CALL PHYSMAP
C
      CALL COEFMTX
C
      CALL TRACNVTR
C
      CALL INTMTX
```

```
C
      CALL CXFXD
C
      CALL PRINT (IWRITE)
C
      END
```

```
      SUBROUTINE INIT
C     ------------------------------------------------------------
C     This subroutine initializes the relevant vectors and
C     arrays to zero.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA3/ TK(200),ZR(200)
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
C
      DO 30 I=1,16
         DO 10 J=1,9
10          WINGDATA(I,J)=0.
         DO 20 J=1,2
20          XSIF(I,J)=0.
30    CONTINUE
C
      DO 40 J=1,2
         DO 40 I=1,400
40          SLTHANG(I,J)=0.
C
      DO 80 I=1,200
         TK(I)=0.
         ZR(I)=0.
         DO 50 J=1,5
            PHYSTK(I,J)=0.
50          PHYSZR(I,J)=0.
         DO 60 J=1,50
60          VTRINT(I,J)=0.
         DO 70 J=1,2
            XF(I,J)=0.
70          XDELTA(I,J)=0.
80    CONTINUE
C
      DO 90 I=1,400
         TRAC(I)=0.
         DO 90 J=1,400
            COEF(I,J)=0.
90          ARRINT(I,J)=0.
C
      END
```

```
      SUBROUTINE INPUT (IR1,IR2,IR3)
C     ------------------------------------------------------------
C     This subroutine reads input data from a data file.  The
C     data determines the initial configuration of the
C     fracture(s), the pressure(s) at the wellbore(s), and the
C     applied stress state.  If the configuration includes a
C     circular inclusion, or one or two material interfaces, the
C     subroutine reads relevant material constants and the
C     inclusion pressure from a separate data file.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      INTEGER GWGN,POINTCNT
      REAL KAP1,KAP2,KAP3
C
C     NWELL       =  Total number of wells in configuration
C     NWING       =  Total number of crack wings in configuration
C     NPOINT      =  Maximum number of specification points allowed
C                    on each wing
C     XO,YO       =  x- and y- coordinates of each well
C     LNWG        =  Local number of wings at each well
C     NWGIMG      =  Global wing number of "image" wing at each well
C                    ( =0 if NINCLUTYPE=2 )
C     NSP         =  Number of specification points on each wing
C     X,Y         =  x- and y- coordinates of each specification point
C     NFACE       =  0 if crack configuration is located in
C                    homogeneous infinite region
C                 =  1 if cracks located in one of the materials of a
C                    bimaterial composite.  The straight interface
C                    lies along the y-axis (X=0) and the cracks lie
C                    in the right-half plane (X.GE.0)
C                 =  2 if cracks located in the middle layer of a
C                    three-material composite.  The first interface
C                    lies along the y-axis (X=0) and the second
C                    interface is at X=XFACE2
C     NINCLUTYPE  =  0 if no circular inclusion in configuration or
C                    if NFACE=1 or NFACE=2
C                 =  1 if cracks located near a circular inclusion in
C                    an infinite region
C                 =  2 if cracks emanating from a borehole in an
C                    infinite region
C     The input stresses and pressures are normalized with
C     phi = E/(4*pi*(1-nu**2)), nu is Poisson's ratio
C     PWELL       =  Pressure at each well
C
```

```
c     SGMAXX      =  ]
c     SGMAYY      =  ] Normal and shear stresses at infinity
c     SGMAXY      =  ]
c     RADINCLU    =  Radius of the circular inclusion
c     For the following, 1 refers to the region outside the circular
c     inclusion and 2 to the region inside the inclusion.  In the
c     case of a bimaterial composite, 1 refers to the right-half
c     plane (where the cracks are located) and 2 to the left-half
c     plane.
c     G1,G2       =  Shear modulus
c     KAP1,KAP2   =  (3-4*nu) for plane strain
c     PINCLU      =  Pressure inside the inclusion
c     For the following, 1 refers to the region between the two
c     material interfaces (0<X<XFACE2) where the cracks are located.
c     2 refers to the left-half plane, and 3 to the region right of
c     the second interface (X>XFACE2).
c     XFACE2      =  x-coordinate of second material interface.
c     G1,G2,G3    =  Shear modulus
c     KAP1,KAP2   =  ] (3-4*nu) for plane strain
c     KAP3        =  ]
c     INCLUTYPE   =  1 if multiplicative approximation,
c                 =  2 if additive approximation to the influence
c                    function for an edge dislocation interacting with
c                    two interfaces.
c

      GWGN=0
      POINTCNT=0
      READ (IR1,900)
      READ (IR1,900) NWELL
      READ (IR1,900)
      READ (IR1,900) NWING
      READ (IR1,900)
      READ (IR1,900) NPOINT
      DO 30 NWL=1,NWELL
         READ (IR1,900)
         READ (IR1,910) XO,YO
         WELLDATA(NWL,1)=XO
         WELLDATA(NWL,2)=YO
         READ (IR1,900)
         READ (IR1,900) LNWG
         WELLDATA(NWL,3)=LNWG
         READ (IR1,900)
         READ (IR1,900) NWGIMG
         WELLDATA (NWL,4)=NWGIMG
         DO 20 NWG=1,LNWG
            GWGN=GWGN+1
            READ (IR1,920)
            READ (IR1,920) NSP
```

```
            WINGDATA(GWGN,1)=NSP
            DO 10 I=1,NSP
               POINTCNT=POINTCNT+1
               READ (IR1,930) X,Y
               POINTDATA(POINTCNT,1)=X
               POINTDATA(POINTCNT,2)=Y
10          CONTINUE
            POINTCNT=GWGN*NPOINT
20       CONTINUE
30    CONTINUE
C
      READ (IR1,900)
      READ (IR1,900) NFACE
      READ (IR1,900)
      READ (IR1,900) NINCLUTYPE
      CINCLUDATA(1)=NINCLUTYPE
      READ (IR1,940)
      READ (IR1,940) (PWELL(NWL),NWL=1,NWELL)
      READ (IR1,940)
      READ (IR1,940) SGMAXX
      READ (IR1,940)
      READ (IR1,940) SGMAYY
      READ (IR1,940)
      READ (IR1,940) SGMAXY
C
      IF ((NFACE.NE.1).AND.(NINCLUTYPE.EQ.0)) GO TO 40
      READ (IR2,940)
      READ (IR2,940) RADINCLU
      READ (IR2,910)
      READ (IR2,910) G1,G2
      READ (IR2,910)
      READ (IR2,910) KAP1,KAP2
      READ (IR2,940)
      READ (IR2,940) PINCLU
      CINCLUDATA(2)=RADINCLU
      CINCLUDATA(3)=G1
      CINCLUDATA(4)=G2
      CINCLUDATA(5)=KAP1
      CINCLUDATA(6)=KAP2
      CINCLUDATA(7)=PINCLU
C
40    IF (NFACE.NE.2) GO TO 50
      READ (IR3,940)
      READ (IR3,940) XFACE2
      READ (IR3,950)
      READ (IR3,950) G1,G2,G3
      READ (IR3,950)
      READ (IR3,950) KAP1,KAP2,KAP3
```

```
      READ (IR3,900)
      READ (IR3,900) INCLUTYPE
      TWOFACEDATA(1)=XFACE2
      TWOFACEDATA(2)=G1
      TWOFACEDATA(3)=G2
      TWOFACEDATA(4)=G3
      TWOFACEDATA(5)=KAP1
      TWOFACEDATA(6)=KAP2
      TWOFACEDATA(7)=KAP3
      TWOFACEDATA(8)=INCLUTYPE
C
900   FORMAT (I2)
910   FORMAT (2F10.8)
920   FORMAT (T11,I2)
930   FORMAT (T10,2F10.8)
940   FORMAT (F10.4)
950   FORMAT (3F10.8)
C
50    RETURN
      END
```

```
      SUBROUTINE CONFIGRTN (IR)
C     ------------------------------------------------------------
C     This subroutine reads in the parameters that specify the
C     configuration of the fractures to be considered, and the
C     associated number of nodal points.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      INTEGER TOTNNODE
C
C     NBRCH    =  Number of branch wings on each main crack
C     NSPMN    =  Number of specification points on each main crack
C     NNODEMN  =  Number of nodal points on each main crack
C     NSPBR    =  Number of specification points on each branch wing
C     NNODEBR  =  Number of nodal points on each branch wing
C
      DO 30 NWG=1,NWING
         READ (IR,910) NBRCH
         WINGDATA(NWG,3)=NBRCH
         READ (IR,920) NSPMN,NNODEMN
         WINGDATA(NWG,4)=NSPMN
         NODEDATA(NWG,2)=NNODEMN
         TOTNNODE=NNODEMN
         IF (NBRCH.EQ.0) GO TO 20
         DO 10 NBR=1,NBRCH
            READ (IR,920) NSPBR,NNODEBR
            WINGDATA(NWG,(4+2*NBR))=NSPBR
            NODEDATA(NWG,2+NBR)=NNODEBR
            TOTNNODE=TOTNNODE+NNODEBR
10       CONTINUE
20       NODEDATA(NWG,1)=TOTNNODE
30    CONTINUE
C
910   FORMAT (I2)
920   FORMAT (I2,8X,I2)
C
      RETURN
      END
```

```
      SUBROUTINE CALCDATA
c     ------------------------------------------------------------
c     This subroutine finds the length and angle of the
c     individual wing segments, calling the subroutine
c     LENGTHANGLE for the actual computations.  It also calls
c     the subroutine TOTALENGTH to find the lengths of the wings.
c     ------------------------------------------------------------
c
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      INTEGER GWGN
c
      GWGN=0
      DO 30 NWL=1,NWELL
         XO=WELLDATA(NWL,1)
         YO=WELLDATA(NWL,2)
         LNWG=WELLDATA(NWL,3)
         DO 20 NWG=1,LNWG
            GWGN=GWGN+1
            NSPPT1=1+(GWGN-1)*NPOINT
            DX=POINTDATA(NSPPT1,1)-XO
            DY=POINTDATA(NSPPT1,2)-YO
            CALL LENGTHANGLE (DX,DY,SLTH,SANG)
            SLTHANG(NSPPT1,1)=SLTH
            SLTHANG(NSPPT1,2)=SANG
            LNSP=WINGDATA(GWGN,1)
            IF (LNSP.EQ.1) GO TO 20
            NSPPT2=NSPPT1+1
            NSPTIP=NSPPT1+LNSP-1
            DO 10 NPT=NSPPT2,NSPTIP
               NPTM1=NPT-1
               DX=POINTDATA(NPT,1)-POINTDATA(NPTM1,1)
               DY=POINTDATA(NPT,2)-POINTDATA(NPTM1,2)
               CALL LENGTHANGLE (DX,DY,SLTH,SANG)
               SLTHANG(NPT,1)=SLTH
               SLTHANG(NPT,2)=SANG
10          CONTINUE
20       CONTINUE
30    CONTINUE
c
      CALL TOTALENGTH
c
      RETURN
      END
```

```
      SUBROUTINE LENGTHANGLE (DX,DY,SLTH,SANG)
c     ----------------------------------------------------------
c     Called by CALCDATA and LOCATE, this subroutine returns the
c     length (or distance) and angle of a crack segment (or
c     individual nodal or collocation point)
c     ----------------------------------------------------------
c
      DATA PI /3.141592653589793238/
c
      SLTH=(DX*DX+DY*DY)**0.5
c
      PIH=PI*0.5
      IF (DX) 10,20,30
10    PSI=PIH+ATAN(DY/DX)
      GO TO 40
20    IF (DY.GT.0.) PSI=0.
      IF (DY.LT.0.) PSI=PI
      GO TO 40
30    PSI=ATAN(DY/DX)-PIH
c
40    SANG=PSI
c
      RETURN
      END
```

```
      SUBROUTINE TOTALENGTH
c     ------------------------------------------------------------
c     Called by CALCDATA, this subroutine finds the length of
c     the main wing(s), and branch wing(s) if any.
c     ------------------------------------------------------------
c
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      REAL LTH
c
      NINCLUTYPE=CINCLUDATA(1)
      RADINCLU=CINCLUDATA(2)
      DO 50 NWG=1,NWING
         NSPPT1=1+(NWG-1)*NPOINT
         NSPMN=WINGDATA(NWG,4)
         NSPTIP=NSPPT1+NSPMN-1
         LTH=0.
         TOTLTH=0.
         DO 10 NPT=NSPPT1,NSPTIP
10          LTH=LTH+SLTHANG(NPT,1)
         IF (NINCLUTYPE.EQ.2) LTH=(LTH-RADINCLU)*0.5
         WINGDATA(NWG,5)=LTH
         TOTLTH=LTH
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 40
         DO 30 NBR=1,NBRCH
            NSPPT1=NSPTIP+1
            NSPBR=WINGDATA(NWG,(4+2*NBR))
            NSPTIP=NSPPT1+NSPBR-1
            LTH=0.
            DO 20 NPT=NSPPT1,NSPTIP
20             LTH=LTH+SLTHANG(NPT,1)
            WINGDATA(NWG,(5+2*NBR))=LTH*0.5
            TOTLTH=TOTLTH+LTH
30       CONTINUE
40       WINGDATA(NWG,2)=TOTLTH
50    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE ZRTK
C     ------------------------------------------------------------
C     This subroutine finds the zeros of the Rth degree
C     Chebyshev polynomials of the first and second kind,
C     defined respectively as TK and ZR.  The degree R is equal
C     to the number of nodals points specified.  The actual
C     computation of the zeros is done in subroutine ZEROS.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      INTEGER COLLCNT,WGTYPE
C
      NODECNT=0
      COLLCNT=0
      NINCLUTYPE=CINCLUDATA(1)
      DO 20 NWG=1,NWING
         WGTYPE=0
         IF (NINCLUTYPE.EQ.2) WGTYPE=1
         NNODEMN=NODEDATA(NWG,2)
         CALL ZEROS (NNODEMN,NODECNT,COLLCNT,WGTYPE)
         NODECNT=NODECNT+NNODEMN
         COLLCNT=COLLCNT+NNODEMN-1
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 20
         WGTYPE=1
         DO 10 NBR=1,NBRCH
            NNODEBR=NODEDATA(NWG,2+NBR)
            CALL ZEROS (NNODEBR,NODECNT,COLLCNT,WGTYPE)
            NODECNT=NODECNT+NNODEBR
            COLLCNT=COLLCNT+NNODEBR-1
10       CONTINUE
20    CONTINUE
C
      NCOEFSZ=2*NODECNT
C
      RETURN
      END
```

```
      SUBROUTINE ZEROS (NNODE,NODECNT,COLLCNT,WGTYPE)
C     ------------------------------------------------------------
C     Called by ZRTK.
C     ------------------------------------------------------------
C
      COMMON / DATA3/ TK(200),ZR(200)
      DATA PI /3.141592653589793238/
      INTEGER COLLCNT,TOTNNODE,WGTYPE
C
      NFIRST=NNODE
      TOTNNODE=(2*NNODE)-1
      IF (WGTYPE.NE.1) GO TO 10
      NFIRST=1
      TOTNNODE=NNODE
C
10    FNODE=FLOAT(TOTNNODE)
      PIN=PI/FNODE
      PI2N=PIN*0.5
      I=0
      DO 20 J=NFIRST,TOTNNODE
         I=I+1
         TK(NODECNT+I)=-COS((2*J-1)*PI2N)
         ZR(COLLCNT+I)=-COS(J*PIN)
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PHYSMAP
c     ------------------------------------------------------------
c     This subroutine "maps" the Chebyshev zeros (lying in either
c     the [0,+1] or [-1,+1] range) onto the actual (physical)
c     fracture configuration.
c     ------------------------------------------------------------
c
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      INTEGER COLLCNT,GWGN,WGTYPE
c
      GWGN=0
      NODECNT=0
      COLLCNT=0
      NINCLUTYPE=CINCLUDATA(1)
      DO 30 NWL=1,NWELL
         XO=WELLDATA(NWL,1)
         YO=WELLDATA(NWL,2)
         LNWG=WELLDATA(NWL,3)
         DO 20 NWG=1,LNWG
            GWGN=GWGN+1
            NNODEMN=NODEDATA(GWGN,2)
            WGLTH=WINGDATA(GWGN,5)
            NSPPT1=1+(GWGN-1)*NPOINT
            NSPMN=WINGDATA(GWGN,4)
            NSPTIP=NSPPT1+NSPMN-1
            WGTYPE=0
            IF (NINCLUTYPE.EQ.2) WGTYPE=2
            CALL LOCATE (NNODEMN,WGLTH,NSPPT1,NSPTIP,
     &                   NODECNT,COLLCNT,WGTYPE,XO,YO)
            NODECNT=NODECNT+NNODEMN
            COLLCNT=COLLCNT+NNODEMN-1
            NBRCH=WINGDATA(GWGN,3)
            IF (NBRCH.EQ.0) GO TO 20
            WGTYPE=1
            DO 10 NBR=1,NBRCH
               NNODEBR=NODEDATA(GWGN,2+NBR)
               WGLTH=WINGDATA(GWGN,(5+2*NBR))
               NSPPT1=NSPTIP+1
               NSPBR=WINGDATA(GWGN,(4+2*NBR))
               NSPTIP=NSPPT1+NSPBR-1
               CALL LOCATE (NNODEBR,WGLTH,NSPPT1,NSPTIP,
     &                      NODECNT,COLLCNT,WGTYPE,XO,YO)
               NODECNT=NODECNT+NNODEBR
               COLLCNT=COLLCNT+NNODEBR-1
10          CONTINUE
20       CONTINUE
```

```
30    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE LOCATE (NNODE,WGLTH,NSPPT1,NSPTIP,
     &                   NODECNT,COLLCNT,WGTYPE,XO,YO)
C     ------------------------------------------------------------
C     Called by PHYSMAP, this subroutine locates the position of
C     the Chebyshev zeros on the physical fracture configuration
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA3/ TK(200),ZR(200)
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      INTEGER COLLCNT,COLLPT,WGTYPE

C
      NINCLUTYPE=CINCLUDATA(1)
      RADINCLU=CINCLUDATA(2)
      NODEPT=NODECNT
      DO 50 I=1,NNODE
         NODEPT=NODEPT+1
         CTK=TK(NODEPT)
         STK=WGLTH*CTK
         IF (WGTYPE.EQ.1) STK=WGLTH*(1.+CTK)
         IF (WGTYPE.EQ.2) STK=WGLTH*(1.+CTK)+RADINCLU
         S=0.
         DO 10 NPT=NSPPT1,NSPTIP
            S=S+SLTHANG(NPT,1)
            IF (STK.LE.S) GO TO 20
10       CONTINUE
20       IF ((WGTYPE.EQ.1).OR.(NPT.GT.NSPPT1)) GO TO 30
         XA=XO
         YA=YO
         GO TO 40
30       IF (NPT.GT.NSPTIP) NPT=NSPTIP
         NPTM1=NPT-1
         XA=POINTDATA(NPTM1,1)
         YA=POINTDATA(NPTM1,2)
40       XB=POINTDATA(NPT,1)
         YB=POINTDATA(NPT,2)
         SCALE=(S-STK)/SLTHANG(NPT,1)
         XTK=XB-SCALE*(XB-XA)
         YTK=YB-SCALE*(YB-YA)
         PHYSTK(NODEPT,1)=XTK
         PHYSTK(NODEPT,2)=YTK
         PHYSTK(NODEPT,3)=SLTHANG(NPT,2)
         IF (NINCLUTYPE.EQ.0) GO TO 50
         CALL LENGTHANGLE (XTK,YTK,SLTH,SANG)
         PHYSTK(NODEPT,4)=SLTH
```

```
         PHYSTK(NODEPT,5)=SANG
50     CONTINUE
c
c
       NCOLL=NNODE-1
       COLLPT=COLLCNT
       DO 100 I=1,NCOLL
          COLLPT=COLLPT+1
          CZR=ZR(COLLPT)
          SZR=WGLTH*CZR
          IF (WGTYPE.EQ.1) SZR=WGLTH*(1.+CZR)
          IF (WGTYPE.EQ.2) SZR=WGLTH*(1.+CZR)+RADINCLU
          S=0.
          DO 60 NPT=NSPPT1,NSPTIP
             S=S+SLTHANG(NPT,1)
             IF (SZR.LE.S) GO TO 70
60        CONTINUE
70        IF ((WGTYPE.EQ.1).OR.(NPT.GT.NSPPT1)) GO TO 80
          XA=XO
          YA=YO
          GO TO 90
80        IF (NPT.GT.NSPTIP) NPT=NSPTIP
          NPTM1=NPT-1
          XA=POINTDATA(NPTM1,1)
          YA=POINTDATA(NPTM1,2)
90        XB=POINTDATA(NPT,1)
          YB=POINTDATA(NPT,2)
          SCALE=(S-SZR)/SLTHANG(NPT,1)
          XZR=XB-SCALE*(XB-XA)
          YZR=YB-SCALE*(YB-YA)
          PHYSZR(COLLPT,1)=XZR
          PHYSZR(COLLPT,2)=YZR
          PHYSZR(COLLPT,3)=SLTHANG(NPT,2)
          IF (NINCLUTYPE.EQ.0) GO TO 100
          CALL LENGTHANGLE (XZR,YZR,SLTH,SANG)
          PHYSZR(COLLPT,4)=SLTH
          PHYSZR(COLLPT,5)=SANG
100    CONTINUE
c
       RETURN
       END
```

```
      SUBROUTINE COEFMTX
C     ------------------------------------------------------------
C     This subroutine sets up the elasticity coefficient matrix
C     [ COEF ].
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      INTEGER COLLCNT,GWGN,FIRSTWG,PREVROWCNT,ROWCNT
C
      GWGN=0
      COLLCNT=0
      ROWCNT=0
      DO 30 NWL=1,NWELL
         PREVROWCNT=ROWCNT
         FIRSTWG=GWGN+1
         LNWG=WELLDATA(NWL,3)
         DO 20 NWG=1,LNWG
            GWGN=GWGN+1
            NNODEMN=NODEDATA(GWGN,2)
            CALL COEFSCRTH (NNODEMN,COLLCNT,ROWCNT)
            COLLCNT=COLLCNT+NNODEMN-1
            ROWCNT=ROWCNT+(2*NNODEMN)
            NBRCH=WINGDATA(GWGN,3)
            IF (NBRCH.EQ.0) GO TO 20
            DO 10 NBR=1,NBRCH
               NNODEBR=NODEDATA(GWGN,2+NBR)
               CALL COEFSCRTH (NNODEBR,COLLCNT,ROWCNT)
               COLLCNT=COLLCNT+NNODEBR-1
               ROWCNT=ROWCNT+(2*NNODEBR)
10          CONTINUE
20       CONTINUE
C
         CALL CLSRE (NWL,FIRSTWG,LNWG,PREVROWCNT)
C
30    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE COEFSCRTH (NNODE,COLLCNT,ROWCNT)
c     ------------------------------------------------------------
c     Called by COEFMTX, this subroutine is the second stage in
c     setting up [ COEF ].  It calls the subroutine INFLU for
c     the necessary calculations.
c     ------------------------------------------------------------
c
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      REAL LTH
      INTEGER COLLCNT,COLLPT,COLUMNCNT,ROWCNT,WGTYPE
c
      COLLPT=COLLCNT
      NROW=ROWCNT
      NCOLL=NNODE-1
      DO 30 I=1,NCOLL
         COLLPT=COLLPT+1
         NROW=NROW+1
         NODECNT=0
         COLUMNCNT=0
         NINCLUTYPE=CINCLUDATA(1)
         DO 20 NWG=1,NWING
            WGTYPE=0
            IF (NINCLUTYPE.EQ.2) WGTYPE=1
            NNODEMN=NODEDATA(NWG,2)
            LTH=WINGDATA(NWG,5)
            CALL INFLU (COLLPT,NCOLL,NROW,NNODEMN,
     &                  NODECNT,LTH,COLUMNCNT,WGTYPE)
            NODECNT=NODECNT+NNODEMN
            COLUMNCNT=COLUMNCNT+(2*NNODEMN)
            NBRCH=WINGDATA(NWG,3)
            IF (NBRCH.EQ.0) GO TO 20
            WGTYPE=1
            DO 10 NBR=1,NBRCH
               NNODEBR=NODEDATA(NWG,2+NBR)
               LTH=WINGDATA(NWG,(5+2*NBR))
               CALL INFLU (COLLPT,NCOLL,NROW,NNODEBR,
     &                     NODECNT,LTH,COLUMNCNT,WGTYPE)
               NODECNT=NODECNT+NNODEBR
               COLUMNCNT=COLUMNCNT+(2*NNODEBR)
10          CONTINUE
20       CONTINUE
30    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE INFLU (COLLPT,NCOLL,NROW,NNODE,
     &                  NODECNT,LTH,COLUMNCNT,WGTYPE)
C     ------------------------------------------------------
C     This subroutine computes the fracture tractions at point
C     (XZR,YZR) due to unit dislocations in the x- and y-
C     directions at point (XTK,YTK).
C     ------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      DATA PI /3.141592653589793238/
      INTEGER COL1,COL2,COLLPT,COLUMNCNT,
     &        ROW1,ROW2,TOTNNODE,WGTYPE
      REAL KAP1,KAP2,KAP3,LM,LM2,LTH,LTHINV,L2,M
      COMPLEX GAMDX,GAMDY,GX,GY,EI2PSI,HDX,HDY,HX,HY,EI2THETA
C
      TOTNNODE=(2*NNODE)-1
      IF (WGTYPE.EQ.1) TOTNNODE=NNODE
      FNODE=FLOAT(TOTNNODE)
      TINV=1./FNODE
      LTHINV=1./LTH
      L2=2./LTH
C
      NODEPT=NODECNT
      NCOLUMN=COLUMNCNT
      NINCLUTYPE=CINCLUDATA(1)
      IF (NINCLUTYPE.GT.0) GO TO 20
      IF (NFACE.EQ.1) GO TO 40
      IF (NFACE.EQ.2) GO TO 60
C
C     ----------------------------------------
C     Influence functions for an edge dislocation
C     in an infinite region
C     ----------------------------------------
C
      XZR=PHYSZR(COLLPT,1)
      YZR=PHYSZR(COLLPT,2)
      PSIZR=PHYSZR(COLLPT,3)
      DO 10 I=1,NNODE
         NODEPT=NODEPT+1
         XTK=PHYSTK(NODEPT,1)
         YTK=PHYSTK(NODEPT,2)
         NCOLUMN=NCOLUMN+1
C
         XI=(XTK-XZR)*LTHINV
         LM=(YZR-YTK)*LTHINV
```

```
      XI2=XI*XI
      LM2=LM*LM
      D1=(LM2+XI2)**2
      D2=1./D1
C
      GAMXXX=-LM*(LM2+3.*XI2)*D2
C
      GAMYYX=-LM*(LM2-XI2)*D2
C
      GAMXYX=-XI*(XI2-LM2)*D2
C
      GAMXXY= XI*(LM2-XI2)*D2
C
      GAMYYY= XI*-1.*(3.*LM2+XI2)*D2
C
      GAMXYY= LM*(XI2-LM2)*D2
C
      GAMMX=0.5*(GAMXXX+GAMYYX)
      GAMDX=CMPLX(0.5*(GAMXXX-GAMYYX),-1.*GAMXYX)
      GAMMY=0.5*(GAMXXY+GAMYYY)
      GAMDY=CMPLX(0.5*(GAMXXY-GAMYYY),-1.*GAMXYY)
C
      EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))

C
      GX=GAMMX+EI2PSI*GAMDX
      GY=GAMMY+EI2PSI*GAMDY
      GAMNNX=REAL(GX)
      GAMNSX=AIMAG(GX)
      GAMNNY=REAL(GY)
      GAMNSY=AIMAG(GY)
C
      ROW1=NROW
      ROW2=NROW+NCOLL
      COL1=NCOLUMN
      COL2=NCOLUMN+NNODE
      COEF(ROW1,COL1)=TINV*GAMNNX
      COEF(ROW1,COL2)=TINV*GAMNNY
      COEF(ROW2,COL1)=TINV*GAMNSX
      COEF(ROW2,COL2)=TINV*GAMNSY
C
10    CONTINUE
      RETURN
C
C     ------------------------------------------------
C     Influence functions for an edge dislocation
C     near a circular inclusion in an infinite region
C     ------------------------------------------------
```

```
c
20    PIH=PI*0.5
      RINC=CINCLUDATA(2)*L2
      G1=CINCLUDATA(3)
      G2=CINCLUDATA(4)
      KAP1=CINCLUDATA(5)
      KAP2=CINCLUDATA(6)
c
      RZR=PHYSZR(COLLPT,4)*L2
      R2=RZR*RZR
      PSIGZR=PHYSZR(COLLPT,5)
      PSIMZR=PHYSZR(COLLPT,3)
c
      SM=G2/G1
      A=(1.-SM)/(1.+SM*KAP1)
      B=(KAP2-SM*KAP1)/(KAP2+SM)
      M=(SM*(1.+KAP1))/((SM+KAP2)*(KAP2-1.+2.*SM))
c
      DO 30 I=1,NNODE
         NODEPT=NODEPT+1
         RTK=PHYSTK(NODEPT,4)*L2
         PSIGTK=PHYSTK(NODEPT,5)
         NCOLUMN=NCOLUMN+1
c
         BT=RTK/RINC
         BT2=BT*BT
         BT2M1=BT2-1.
         BT3=BT2*BT
         PSILZR=PSIGZR-(PIH+PSIGTK)
         X=-RZR*SIN(PSILZR)
         Y= RZR*COS(PSILZR)
c
         X1=(X-RTK)
         X2=X-(RINC/BT)
         X12=X1*X1
         X22=X2*X2
         R12=X12+Y*Y
         R22=(X2*X2)+(Y*Y)
         XR2=X/R2
         X1R12=X1/R12
         X2R22=X2/R22
         X2R2=X*X/R2
         X12R12=X12/R12
         X22R22=X22/R22
         YR2=Y/R2
         YR12=Y/R12
         YR22=Y/R22
         RR2=RINC/R2
```

```
      RR22=RINC/R22
C
      HYY1 =   2.*YR12*(2.*X12R12-1.)
     &       + (3.*A-B-4.*A*X22R22)*YR22
     &       - 2.*A*(BT2M1/BT3)*RR22*( 2.*X2R22*Y*(4.*X22R22-3.)
     &       - (BT2M1/BT)*RR22*Y*(4.*X22R22-1.) )
     &       - ( (3.*A-B-4.*A*X2R2)*YR2
     &       - 2.*((B-A)/BT)*XR2*YR2*RINC
     &       + 2.*A*RINC*RR2*YR2*(4.*X2R2-1.) )
C
      HXX1 = - 2.*YR12*(2.*X12R12+1.)
     &       + (A+B+4.*A*X22R22)*YR22
     &       - 2.*A*(BT2M1/BT3)*RR22*( 2.*X2R22*Y*(1.-4.*X22R22)
     &       - (BT2M1/BT)*RR22*Y*(1.-4.*X22R22) )
     &       - ( (A+B+4.*A*X2R2)*YR2
     &       + 2.*((B-A)/BT)*RINC*XR2*YR2
     &       + 2.*A*RINC*RR2*YR2*(1.-4.*X2R2) )
C
      HXY1 =   2.*X1R12*(2.*X12R12-1.)
     &       + (3.*A-B-4.*A*X22R22)*X2R22
     &       - 2.*A*(BT2M1/BT3)*RR22*( 1. - 8.*X22R22
     &       + 8.*X22R22*X22R22
     &       + (BT2M1/BT)*RINC*X2R22*(3.- 4.*X22R22) )
     &       - ( (3.*A-B-4.*A*X2R2)*XR2
     &       + ((B-A)/BT)*(1.-2.*X2R2)*RR2
     &       - 2.*A*RINC*RR2*XR2*(3.-4.*X2R2) )
C
      HYY2 =   2.*(3.-2.*X12R12)*X1R12
     &       - (5.*A+B-4.*A*X22R22)*X2R22
     &       - 2.*A*(BT2M1/BT3)*RR22*( (2.-BT2)
     &       - 2.*(5.-BT2)*X22R22 + 8.*X22R22*X22R22
     &       + (BT2M1/BT)*RINC*X2R22*(3.- 4.*X22R22) )
     &       + ( (5.*A+B-4.*A*X2R2)*XR2 - ( A*(2.*BT2-1.)
     &       + M*(1.+KAP2)-1. )*(RR2/BT)*(1.-2.*X2R2)
     &       - 2.*A*RINC*RR2*XR2*(3.-4.*X2R2) )
C
      HXX2 =   2.*X1R12*(2.*X12R12-1.)
     &       + ( B+A-4.*A*X22R22 )*X2R22
     &       + A*(BT2M1/BT3)*RR22*( 2*BT2*(2.*X22R22-1.)
     &       - (3.-4.*X22R22)*4.*X22R22
     &       + (BT2M1/BT)*RINC*X2R22*(6.- 8.*X22R22) )
     &       - ( XR2*(B+A-4.*A*X2R2) + ( A*(2.*BT2-1.)
     &       + M*(1.+KAP2)-1. )*(RR2/BT)*(2.*X2R2-1.)
     &       - A*RINC*RR2*XR2*(6.-8.*X2R2) )
C
      HXY2 =   2.*YR12*(2.*X12R12-1.)
     &       + (B+A-4.*A*X22R22)*YR22
     &       + A*(BT2M1/BT3)*RR22*( 4.*BT2*X2R22*Y
```

```
     &          - 8.*X2R22*Y*(1.- 2.*(X22R22/BT))
     &          + 2.*(BT2M1/BT)*RR22*Y*(1.- 4.*X22R22) )
     &          - ( YR22*(B+A-4.*A*X2R2)+ 2.*( A*(2.*BT2-1.)
     &          + M*(1.+KAP2)-1. )*(RINC/BT)*XR2*YR2
     &          - 2.*A*RINC*RR2*YR2*(1.-4.*X2R2) )
c
      SPSI=SIN(PSIGTK)
      CPSI=COS(PSIGTK)
c
      HXXX=-HXX1*SPSI-HXX2*CPSI
      HXXY= HXX1*CPSI-HXX2*SPSI
      HYYX=-HYY1*SPSI-HYY2*CPSI
      HYYY= HYY1*CPSI-HYY2*SPSI
      HXYX=-HXY1*SPSI-HXY2*CPSI
      HXYY= HXY1*CPSI-HXY2*SPSI
c
      HMX=0.5*(HXXX+HYYX)
      HDX=CMPLX(0.5*(HXXX-HYYX),-1.*HXYX)
      HMY=0.5*(HXXY+HYYY)
      HDY=CMPLX(0.5*(HXXY-HYYY),-1.*HXYY)
c
      THETA=PSIMZR-(PIH+PSIGTK)
      EI2THETA=CMPLX(COS(2.*THETA),SIN(2.*THETA))
c
      HX=HMX+EI2THETA*HDX
      HY=HMY+EI2THETA*HDY
      HNNX=REAL(HX)
      HNSX=AIMAG(HX)
      HNNY=REAL(HY)
      HNSY=AIMAG(HY)
c
      ROW1=NROW
      ROW2=NROW+NCOLL
      COL1=NCOLUMN
      COL2=NCOLUMN+NNODE
      COEF(ROW1,COL1)=TINV*HNNX
      COEF(ROW1,COL2)=TINV*HNNY
      COEF(ROW2,COL1)=TINV*HNSX
      COEF(ROW2,COL2)=TINV*HNSY
c
30    CONTINUE
      RETURN
c
c     ----------------------------------------
c     Influence functions for an edge dislocation
c     near a straight interface in a bimaterial
c     composite
c     ----------------------------------------
```

```
c
40    G1=CINCLUDATA(3)
      G2=CINCLUDATA(4)
      KAP1=CINCLUDATA(5)
      KAP2=CINCLUDATA(6)
c
      SM=G2/G1
      A=(1.-SM)/(1.+SM*KAP1)
      B=(KAP2-SM*KAP1)/(KAP2+SM)
c
      XZR=PHYSZR(COLLPT,1)*L2
      YZR=PHYSZR(COLLPT,2)*L2
      PSIZR=PHYSZR(COLLPT,3)
      DO 50 I=1,NNODE
         NODEPT=NODEPT+1
         XTK=PHYSTK(NODEPT,1)*L2
         YTK=PHYSTK(NODEPT,2)*L2
         NCOLUMN=NCOLUMN+1
c
         X1=XZR-XTK
         X2=XZR+XTK
         Y=YZR-YTK
         Y2=Y*Y
         X12=X1*X1
         X22=X2*X2
         R12=X12+Y2
         R22=X22+Y2
         XR22=XTK/R22
         X1R12=X1/R12
         X2R22=X2/R22
         X12R12=X12/R12
         X22R22=X22/R22
         YR12=Y/R12
         YR22=Y/R22
c
         HYYX =   2.*YR12*(2.*X12R12-1.)
     &          + (3.*A-B-4.*A*X22R22)*YR22
     &          - 4.*A*XR22*( 2.*X2R22*Y*(4.*X22R22-3.)
     &          - 2.*XR22*Y*(4.*X22R22-1.) )
c
         HXXX = - 2.*YR12*(2.*X12R12+1.)
     &          + (A+B+4.*A*X22R22)*YR22
     &          - 4.*A*XR22*( 2.*X2R22*Y*(1.-4.*X22R22)
     &          - 2.*XR22*Y*(1.-4.*X22R22) )
c
         HXYX =   2.*X1R12*(2.*X12R12-1.)
     &          + (3.*A-B-4.*A*X22R22)*X2R22
     &          - 4.*A*XR22*( 1. - 8.*X22R22
```

```
     &           + 8.*X22R22*X22R22
     &           + 2.*XR22*X2*(3.-4.*X22R22) )
c
          HYYY =   2.*(3.-2.*X12R12)*X1R12
     &           - (5.*A+B-4.*A*X22R22)*X2R22
     &           - 4.*A*XR22*( 1. - 8.*X22R22
     &           + 8.*X22R22*X22R22
     &           + 2.*XR22*X2*(3.-4.*X22R22) )
c
          HXXY =   2.*X1R12*(2.*X12R12-1.)
     &           + ( B+A-4.*A*X22R22 )*X2R22
     &           + 2.*A*XR22*( 2.*(2.*X22R22-1.)
     &           - (3.-4.*X22R22)*4.*X22R22
     &           + 2.*XR22*X2*(6.-8.*X22R22) )
c
          HXYY =   2.*YR12*(2.*X12R12-1.)
     &           + (B+A-4.*A*X22R22)*YR22
     &           + 2.*A*XR22*( 4.*X2R22*Y
     &           - 8.*X2R22*Y*(1.- 2.*X22R22)
     &           + 4.*XR22*Y*(1.-4.*X22R22) )
c
          HMX=0.5*(HXXX+HYYX)
          HDX=CMPLX(0.5*(HXXX-HYYX),-1.*HXYX)
          HMY=0.5*(HXXY+HYYY)
          HDY=CMPLX(0.5*(HXXY-HYYY),-1.*HXYY)
c
          EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))
c
          HX=HMX+EI2PSI*HDX
          HY=HMY+EI2PSI*HDY
          HNNX=REAL(HX)
          HNSX=AIMAG(HX)
          HNNY=REAL(HY)
          HNSY=AIMAG(HY)
c
          ROW1=NROW
          ROW2=NROW+NCOLL
          COL1=NCOLUMN
          COL2=NCOLUMN+NNODE
          COEF(ROW1,COL1)=TINV*HNNX
          COEF(ROW1,COL2)=TINV*HNNY
          COEF(ROW2,COL1)=TINV*HNSX
          COEF(ROW2,COL2)=TINV*HNSY
c
50     CONTINUE
       RETURN
c
c
```

```
c     ------------------------------------------
c     Influence functions for an edge dislocation
c     near two straight interfaces in a three-
c     material composite
c     ------------------------------------------
c
60    XFACE2=TWOFACEDATA(1)
      G1=TWOFACEDATA(2)
      G2=TWOFACEDATA(3)
      G3=TWOFACEDATA(4)
      KAP1=TWOFACEDATA(5)
      KAP2=TWOFACEDATA(6)
      KAP3=TWOFACEDATA(7)
      INCLUTYPE=TWOFACEDATA(8)
c
      SMF1=G2/G1
      SMF2=G3/G1
      AF1=(1.-SMF1)/(1.+SMF1*KAP1)
      AF2=(1.-SMF2)/(1.+SMF2*KAP1)
      BF1=(KAP2-SMF1*KAP1)/(KAP2+SMF1)
      BF2=(KAP3-SMF2*KAP1)/(KAP3+SMF2)
c
      XZR=PHYSZR(COLLPT,1)
      YZR=PHYSZR(COLLPT,2)
      XZRF1=XZR*L2
      YZRF1=YZR*L2
      XZRF2=(XZR-XFACE2)*L2
      YZRF2=YZRF1
      PSIZR=PHYSZR(COLLPT,3)
c
      DO 90 I=1,NNODE
         NODEPT=NODEPT+1
         XTK=PHYSTK(NODEPT,1)
         YTK=PHYSTK(NODEPT,2)
         XTKF1=XTK*L2
         YTKF1=YTK*L2
         XTKF2=(XTK-XFACE2)*L2
         YTKF2=YTKF1
         NCOLUMN=NCOLUMN+1
c
         XI=(XTK-XZR)*LTHINV
         LM=(YZR-YTK)*LTHINV
         XI2=XI*XI
         LM2=LM*LM
         D1=(LM2+XI2)**2
         D2=1./D1
c
         HXXXHM=-LM*(LM2+3.*XI2)*D2
```

```
c
      HYYXHM=-LM*(LM2-XI2)*D2
c
      HXYXHM=-XI*(XI2-LM2)*D2
c
      HXXYHM= XI*(LM2-XI2)*D2
c
      HYYYHM= XI*-1.*(3.*LM2+XI2)*D2
c
      HXYYHM= LM*(XI2-LM2)*D2
c
      HMX=0.5*(HXXXHM+HYYXHM)
      HDX=CMPLX(0.5*(HXXXHM-HYYXHM),-1.*HXYXHM)
      HMY=0.5*(HXXYHM+HYYYHM)
      HDY=CMPLX(0.5*(HXXYHM-HYYYHM),-1.*HXYYHM)
c
      EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))
c
      HX=HMX+EI2PSI*HDX
      HY=HMY+EI2PSI*HDY
      HNNXHM=REAL(HX)
      HNSXHM=AIMAG(HX)
      HNNYHM=REAL(HY)
      HNSYHM=AIMAG(HY)
c
      X1=XZRF1-XTKF1
      X2=XZRF1+XTKF1
      Y=YZRF1-YTKF1
      Y2=Y*Y
      X12=X1*X1
      X22=X2*X2
      R12=X12+Y2
      R22=X22+Y2
      XR22=XTKF1/R22
      X1R12=X1/R12
      X2R22=X2/R22
      X12R12=X12/R12
      X22R22=X22/R22
      YR12=Y/R12
      YR22=Y/R22
c
      HYYXF1 =   2.*YR12*(2.*X12R12-1.)
     &         + (3.*AF1-BF1-4.*AF1*X22R22)*YR22
     &         - 4.*AF1*XR22*( 2.*X2R22*Y*(4.*X22R22-3.)
     &         - 2.*XR22*Y*(4.*X22R22-1.) )
c
      HXXXF1 = - 2.*YR12*(2.*X12R12+1.)
```

```
     &            + (AF1+BF1+4.*AF1*X22R22)*YR22
     &            - 4.*AF1*XR22*( 2.*X2R22*Y*(1.-4.*X22R22)
     &            - 2.*XR22*Y*(1.-4.*X22R22) )
C
          HXYXF1 =   2.*X1R12*(2.*X12R12-1.)
     &            + (3.*AF1-BF1-4.*AF1*X22R22)*X2R22
     &            - 4.*AF1*XR22*( 1. - 8.*X22R22
     &            + 8.*X22R22*X22R22
     &            + 2.*XR22*X2*(3.-4.*X22R22) )
C
          HYYYF1 =   2.*(3.-2.*X12R12)*X1R12
     &            - (5.*AF1+BF1-4.*AF1*X22R22)*X2R22
     &            - 4.*AF1*XR22*( 1. - 8.*X22R22
     &            + 8.*X22R22*X22R22
     &            + 2.*XR22*X2*(3.-4.*X22R22) )
C
          HXXYF1 =   2.*X1R12*(2.*X12R12-1.)
     &            + ( BF1+AF1-4.*AF1*X22R22 )*X2R22
     &            + 2.*AF1*XR22*( 2.*(2.*X22R22-1.)
     &            - (3.-4.*X22R22)*4.*X22R22
     &            + 2.*XR22*X2*(6.-8.*X22R22) )
C
          HXYYF1 =   2.*YR12*(2.*X12R12-1.)
     &            + (BF1+AF1-4.*AF1*X22R22)*YR22
     &            + 2.*AF1*XR22*( 4.*X2R22*Y
     &            - 8.*X2R22*Y*(1.- 2.*X22R22)
     &            + 4.*XR22*Y*(1.-4.*X22R22) )
C
          HMX=0.5*(HXXXF1+HYYXF1)
          HDX=CMPLX(0.5*(HXXXF1-HYYXF1),-1.*HXYXF1)
          HMY=0.5*(HXXYF1+HYYYF1)
          HDY=CMPLX(0.5*(HXXYF1-HYYYF1),-1.*HXYYF1)
C
          EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))

C
          HX=HMX+EI2PSI*HDX
          HY=HMY+EI2PSI*HDY
          HNNXF1=REAL(HX)
          HNSXF1=AIMAG(HX)
          HNNYF1=REAL(HY)
          HNSYF1=AIMAG(HY)
C
          X1=XZRF2-XTKF2
          X2=XZRF2+XTKF2
          Y=YZRF2-YTKF2
          Y2=Y*Y
          X12=X1*X1
```

```
      X22=X2*X2
      R12=X12+Y2
      R22=X22+Y2
      XR22=XTKF2/R22
      X1R12=X1/R12
      X2R22=X2/R22
      X12R12=X12/R12
      X22R22=X22/R22
      YR12=Y/R12
      YR22=Y/R22
C
      HYYXF2 =   2.*YR12*(2.*X12R12-1.)
     &         + (3.*AF2-BF2-4.*AF2*X22R22)*YR22
     &         - 4.*AF2*XR22*( 2.*X2R22*Y*(4.*X22R22-3.)
     &         - 2.*XR22*Y*(4.*X22R22-1.) )
C
      HXXXF2 = - 2.*YR12*(2.*X12R12+1.)
     &         + (AF2+BF2+4.*AF2*X22R22)*YR22
     &         - 4.*AF2*XR22*( 2.*X2R22*Y*(1.-4.*X22R22)
     &         - 2.*XR22*Y*(1.-4.*X22R22) )
C
      HXYXF2 =   2.*X1R12*(2.*X12R12-1.)
     &         + (3.*AF2-BF2-4.*AF2*X22R22)*X2R22
     &         - 4.*AF2*XR22*( 1. - 8.*X22R22
     &         + 8.*X22R22*X22R22
     &         + 2.*XR22*X2*(3.-4.*X22R22) )
C
      HYYYF2 =   2.*(3.-2.*X12R12)*X1R12
     &         - (5.*AF2+BF2-4.*AF2*X22R22)*X2R22
     &         - 4.*AF2*XR22*( 1. - 8.*X22R22
     &         + 8.*X22R22*X22R22
     &         + 2.*XR22*X2*(3.-4.*X22R22) )
C
      HXXYF2 =   2.*X1R12*(2.*X12R12-1.)
     &         + ( BF2+AF2-4.*AF2*X22R22 )*X2R22
     &         + 2.*AF2*XR22*( 2.*(2.*X22R22-1.)
     &         - (3.-4.*X22R22)*4.*X22R22
     &         + 2.*XR22*X2*(6.-8.*X22R22) )
C
      HXYYF2 =   2.*YR12*(2.*X12R12-1.)
     &         + (BF2+AF2-4.*AF2*X22R22)*YR22
     &         + 2.*AF2*XR22*( 4.*X2R22*Y
     &         - 8.*X2R22*Y*(1.- 2.*X22R22)
     &         + 4.*XR22*Y*(1.-4.*X22R22) )
C
      HMX=0.5*(HXXXF2+HYYXF2)
      HDX=CMPLX(0.5*(HXXXF2-HYYXF2),-1.*HXYXF2)
      HMY=0.5*(HXXYF2+HYYYF2)
```

```
      HDY=CMPLX(0.5*(HXXYF2-HYYYF2),-1.*HXYYF2)
C
      EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))

C
      HX=HMX+EI2PSI*HDX
      HY=HMY+EI2PSI*HDY
      HNNXF2=REAL(HX)
      HNSXF2=AIMAG(HX)
      HNNYF2=REAL(HY)
      HNSYF2=AIMAG(HY)
C
      IF (INCLUTYPE.EQ.2) GO TO 70
      HNNX=HNNXF1*HNNXF2/HNNXHM
      HNSX=HNSXF1*HNSXF2/HNSXHM
      HNNY=HNNYF1*HNNYF2/HNNYHM
      HNSY=HNSYF1*HNSYF2/HNSYHM
      GO TO 80
C
70    HNNX=HNNXF1+HNNXF2-HNNXHM
      HNSX=HNSXF1+HNSXF2-HNSXHM
      HNNY=HNNYF1+HNNYF2-HNNYHM
      HNSY=HNSYF1+HNSYF2-HNSYHM
C
80    ROW1=NROW
      ROW2=NROW+NCOLL
      COL1=NCOLUMN
      COL2=NCOLUMN+NNODE
      COEF(ROW1,COL1)=TINV*HNNX
      COEF(ROW1,COL2)=TINV*HNNY
      COEF(ROW2,COL1)=TINV*HNSX
      COEF(ROW2,COL2)=TINV*HNSY
C
90    CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE CLSRE (NWL,FIRSTWG,LNWG,PREVROWCNT)
C     ------------------------------------------------------------
C     Called by COEFMTX, this subroutine fills in the
C     closure and matching conditions necessary to complete the
C     system of equations. The closure condition is that
C     there is no net entrapped dislocation at a wellbore,
C     while the matching conditions at a wellbore are that
C     1)  the dislocation density is continuous for the first
C     wing and its image wing, and 2)  the dislocation density
C     for any other wing is zero.  The matching conditions for
C     branch wings is that the dislocation density is zero at
C     the point of intersection with the main wing or
C     preceding branch wing.  Note that the conditions for
C     cracks intersecting an inclusion depend on the inclusion
C     modulus and the crack length/borehole radius ratio.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      INTEGER COLUMNCNT,FIRSTWG,GWGN,PREVROWCNT,ROWCNT,WGTYPE
      REAL LTH
C
      NINCLUTYPE=CINCLUDATA(1)
      RADINCLU=CINCLUDATA(2)
      G2=CINCLUDATA(4)
      ROWCNT=0
      COLUMNCNT=PREVROWCNT
      LTH=WINGDATA(FIRSTWG,5)
      WGTYPE=0
      IF (NINCLUTYPE.EQ.2) WGTYPE=5
      IF ((NINCLUTYPE.EQ.2).AND.(LNWG.EQ.1).AND.
     &    (G2.EQ.0).AND.(LTH.LE.(0.1*RADINCLU))) WGTYPE=4
      NNODEMN=NODEDATA(FIRSTWG,2)
      NROWCK=PREVROWCNT+(2*NNODEMN)
      ROWCNT=NROWCK
      CALL CLSRESCRTH (WGTYPE,NNODEMN,LTH,NROWCK,
     &                 COLUMNCNT,ROWCNT,PREVROWCNT)
      COLUMNCNT=ROWCNT
      NBRCH=WINGDATA(FIRSTWG,3)
      IF (NBRCH.EQ.0) GO TO 20
      WGTYPE=3
      DO 10 NBR=1,NBRCH
         NNODEBR=NODEDATA(FIRSTWG,2+NBR)
         LTH=WINGDATA(FIRSTWG,(5+2*NBR))
         ROWCNT=ROWCNT+(2*NNODEBR)
         CALL CLSRESCRTH (WGTYPE,NNODEBR,LTH,NROWCK,
     &                    COLUMNCNT,ROWCNT,PREVROWCNT)
```

```
         COLUMNCNT=ROWCNT
10     CONTINUE
20     IF ((NINCLUTYPE.EQ.2).AND.(LNWG.EQ.1)) GO TO 50
       NWGIMG=WELLDATA(NWL,4)
       GWGN=FIRSTWG
       DO 40 NWG=2,LNWG
          GWGN=GWGN+1
          WGTYPE=1
          IF (GWGN.EQ.NWGIMG) WGTYPE=2
          IF (NINCLUTYPE.EQ.2) WGTYPE=6
          NNODEMN=NODEDATA(GWGN,2)
          LTH=WINGDATA(GWGN,5)
          ROWCNT=ROWCNT+(2*NNODEMN)
          CALL CLSRESCRTH (WGTYPE,NNODEMN,LTH,NROWCK,
     &                     COLUMNCNT,ROWCNT,PREVROWCNT)
          COLUMNCNT=ROWCNT
          NBRCH=WINGDATA(GWGN,3)
          IF (NBRCH.EQ.0) GO TO 40
          WGTYPE=3
          DO 30 NBR=1,NBRCH
             NNODEBR=NODEDATA(GWGN,2+NBR)
             LTH=WINGDATA(GWGN,(5+2*NBR))
             ROWCNT=ROWCNT+(2*NNODEBR)
             CALL CLSRESCRTH (WGTYPE,NNODEBR,LTH,NROWCK,
     &                        COLUMNCNT,ROWCNT,PREVROWCNT)
             COLUMNCNT=ROWCNT
30        CONTINUE
40     CONTINUE
C
50     RETURN
       END
```

```
      SUBROUTINE CLSRESCRTH (WGTYPE,NNODE,LTH,NROWCK,
     &                       COLUMNCNT,NROW,PREVROWCNT)
C     ------------------------------------------------------------
C     Called by CLSRE, this subroutine calculates the values of
C     the elements to be inserted into [ COEF ] as the
C     appropriate closure conditions; it also fills in the
C     matching conditions by inserting appropriate values into
C     [ COEF ] and [ TRAC ].
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      DATA PI /3.141592653589793238/
      REAL LTH
      INTEGER COLUMNCNT,PREVROWCNT,TOTNNODE,WGTYPE
C
      NROWCKM1=NROWCK-1
      NCOL1=COLUMNCNT
      NCOL2=NCOL1+NNODE
C
C     ------------------
C     Closure conditions
C     ------------------
C
      IF ((WGTYPE.NE.0).AND.(WGTYPE.NE.1).AND.
     &    (WGTYPE.NE.2)) GO TO 30
      NFIRST=NNODE
      TOTNNODE=(2*NNODE)-1
      NM1=TOTNNODE-1
      FNODE=FLOAT(TOTNNODE)
      PIH=PI*0.5
      PI2N=PIH/FNODE
      CONST=2./FNODE
      K=0
      DO 20 I=NFIRST,TOTNNODE
         K=K+1
         ARG=PI-(2*I-1)*PI2N
         SUM=0.
         DO 10 J=1,NM1
            FJ=FLOAT(J)
10          SUM=SUM+COS(J*ARG)*SIN(J*PIH)/FJ
         CFNT=LTH*(PI2N+CONST*SUM)
         COEF(NROWCKM1,NCOL1+K)=CFNT
         COEF(NROWCK,NCOL2+K)=CFNT
20    CONTINUE
C
```

```
30    IF ((WGTYPE.NE.3).AND.(WGTYPE.NE.5).AND.
     &    (WGTYPE.NE.6)) GO TO 50
      FNODE=FLOAT(NNODE)
      CFNT=LTH*PI/FNODE
      DO 40 I=1,NNODE
         COEF(NROWCKM1,NCOL1+I)=CFNT
         COEF(NROWCK,NCOL2+I)=CFNT
40    CONTINUE
C
C     -------------------
C     Matching conditions
C     -------------------
C
50    IF ((WGTYPE.NE.1).AND.(WGTYPE.NE.3).AND.
     &    (WGTYPE.NE.4).AND.(WGTYPE.NE.6)) GO TO 60
      NROWM1=NROW-1
      COEF(NROWM1,NCOL1+1)=1.
      TRAC(NROWM1)=0.
      COEF(NROW,NCOL2+1)=1.
      TRAC(NROW)=0.
C
60    IF (WGTYPE.NE.2) GO TO 70
      NNODECK=(NROWCK-PREVROWCNT)/2
      NROWM1=NROW-1
      COEF(NROWM1,PREVROWCNT+1)=1.
      COEF(NROWM1,NCOL1+1)=-1.
      TRAC(NROWM1)=0.
      COEF(NROW,PREVROWCNT+NNODECK+1)=1.
      COEF(NROW,NCOL2+1)=-1.
      TRAC(NROW)=0.
C
70    RETURN
      END
```

```
      SUBROUTINE TRACNVTR
c     ------------------------------------------------------
c     This subroutine determines the tractions required on the
c     fracture surfaces.  Superposition of these tractions with
c     the stresses gives only normal stresses on the fracture
c     faces, equal to the internal fracture pressures.
c     ------------------------------------------------------
c
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      INTEGER COLLCNT,GWGN,ROWCNT
c
      GWGN=0
      COLLCNT=0
      ROWCNT=0
      DO 30 NWL=1,NWELL
         PRESSR=PWELL(NWL)
         LNWG=WELLDATA(NWL,3)
         DO 20 NWG=1,LNWG
            GWGN=GWGN+1
            NNODEMN=NODEDATA(GWGN,2)
            CALL TRACNSCRTH (NNODEMN,COLLCNT,ROWCNT,PRESSR)
            COLLCNT=COLLCNT+NNODEMN-1
            ROWCNT=ROWCNT+(2*NNODEMN)
            NBRCH=WINGDATA(GWGN,3)
            IF (NBRCH.EQ.0) GO TO 20
            DO 10 NBR=1,NBRCH
               NNODEBR=NODEDATA(GWGN,2+NBR)
               CALL TRACNSCRTH (NNODEBR,COLLCNT,ROWCNT,PRESSR)
               COLLCNT=COLLCNT+NNODEBR-1
               ROWCNT=ROWCNT+(2*NNODEBR)
10          CONTINUE
20       CONTINUE
30    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE TRACNSCRTH (NNODE,COLLCNT,ROWCNT,PRESSR)
C     ------------------------------------------------------
C     Called by TRACNVTR, this subroutine calculates the
C     required tractions and assigns them to the traction, or
C     excess pressure, vector [ TRAC ].
C     ------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      DATA PI /3.141592653589793238/
      REAL KAP1,KAP2
      INTEGER COLLCNT,COLLPT,ROWCNT
      COMPLEX SGMAD,SIGMA,EI2PSI
C
      COLLPT=COLLCNT
      NROW=ROWCNT
      NCOLL=NNODE-1
C
      NINCLUTYPE=CINCLUDATA(1)
      IF (NINCLUTYPE.GT.0) GO TO 20
C
C     ---------------------------------------
C     Stresses in a homogenous infinite region
C     ---------------------------------------
C
      SGMAM=0.5*(SGMAXX+SGMAYY)
      SGMAD=CMPLX(0.5*(SGMAXX-SGMAYY),-1.*SGMAXY)
C
      DO 10 I=1,NCOLL
         COLLPT=COLLPT+1
         PSIZR=PHYSZR(COLLPT,3)
         NROW=NROW+1
C
         EI2PSI=CMPLX(COS(2.*PSIZR),SIN(2.*PSIZR))
         SIGMA=SGMAM+EI2PSI*SGMAD
         SGMANN=REAL(SIGMA)
         SGMANS=AIMAG(SIGMA)
C
         TRAC(NROW)=-SGMANN-PRESSR
         TRAC(NROW+NCOLL)=-SGMANS
C
10    CONTINUE
      RETURN
C
C
C
```

```
c     ------------------------------------------------------
c     Stresses near a circular inclusion in an infinite region
c     ------------------------------------------------------
c
20    RINC=CINCLUDATA(2)
      G1=CINCLUDATA(3)
      G2=CINCLUDATA(4)
      KAP1=CINCLUDATA(5)
      KAP2=CINCLUDATA(6)
      PINCLU=CINCLUDATA(7)
      SM=G2/G1
      A=2.*(1.-SM)/(1.+SM*KAP1)
      B=(KAP2-1.-SM*(KAP1-1.))/(2.*SM+KAP2-1.)
      C=(SM-1.)/(SM*KAP1+1.)
      SXPSY=0.5*(SGMAXX+SGMAYY)
      SXMSY=0.5*(SGMAXX-SGMAYY)
      PIH=PI*0.5
c
      DO 30 I=1,NCOLL
         COLLPT=COLLPT+1
         PSIGZR=PHYSZR(COLLPT,5)
         PSIMZR=PHYSZR(COLLPT,3)
         PSI=PSIMZR-PSIGZR
         RCOLL=PHYSZR(COLLPT,4)
         R2R2=(RINC*RINC)/(RCOLL*RCOLL)
         R4R4=R2R2*R2R2
         PR2R2=PINCLU*R2R2
         THETA=PIH+PSIGZR
         C2THA=COS(2.*THETA)
         S2THA=SIN(2.*THETA)
c
         SGMARR=SXPSY*(1.-B*R2R2)-PR2R2+SXMSY*(1.-2.*A*R2R2
     &          - 3.*C*R4R4)*C2THA
c
         SGMATH=SXPSY*(1.+B*R2R2)+PR2R2-SXMSY*(1.-3.*C*R4R4)*C2THA
c
         SGMART=SXMSY*(1.+A*R2R2+3.*C*R4R4)*S2THA
c
         SGMAM=0.5*(SGMATH+SGMARR)
         SGMAD=CMPLX(0.5*(SGMATH-SGMARR),-1.*SGMART)
         EI2PSI=CMPLX(COS(2.*PSI),SIN(2.*PSI))
         SIGMA=SGMAM+EI2PSI*SGMAD
         SGMANN=REAL(SIGMA)
         SGMANS=AIMAG(SIGMA)
c
         NROW=NROW+1
         TRAC(NROW)=-SGMANN-PRESSR
         TRAC(NROW+NCOLL)=-SGMANS
```

```
C
30    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTMTX
C     ------------------------------------------------------------
C     This subroutine sets up the integration matrix [ ARRINT ]
C     to obtain the crack opening displacements from the
C     dislocation densities.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      REAL LTH
      INTEGER FIRSTROWBR(2),FIRSTROWMN,NNODEBR(2),
     &        RWCNTARR,RWCNTVTR
C
      CALL INTVTR
C
      RWCNTVTR=0
      RWCNTARR=0
      DO 30 NWG=1,NWING
         NNODEMN=NODEDATA(NWG,2)
         LTH=WINGDATA(NWG,5)
         FIRSTROWMN=RWCNTARR+1
         CALL INTMTXSCRTH (RWCNTARR,RWCNTVTR,NNODEMN,LTH)
         RWCNTVTR=RWCNTVTR+NNODEMN
         RWCNTARR=RWCNTARR+(2*NNODEMN)
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 30
         DO 20 NBR=1,NBRCH
            NNODEBR(NBR)=NODEDATA(NWG,2+NBR)
            LTH=WINGDATA(NWG,(5+2*NBR))
            FIRSTROWBR(NBR)=RWCNTARR+1
            CALL INTMTXSCRTH (RWCNTARR,RWCNTVTR,NNODEBR(NBR),LTH)
            RWCNTVTR=RWCNTVTR+NNODEBR(NBR)
            RWCNTARR=RWCNTARR+(2*NNODEBR(NBR))
            CALL ADDCTRBN (FIRSTROWMN,NNODEMN,
     &                     FIRSTROWBR(NBR),NNODEBR(NBR),LTH)
            IF (NBR.EQ.1) GO TO 20
            NBRM1=NBR-1
            DO 10 I=1,NBRM1
               CALL ADDCTRBN (FIRSTROWBR(I),NNODEBR(I),
     &                        FIRSTROWBR(NBR),NNODEBR(NBR),LTH)
10          CONTINUE
20       CONTINUE
30    CONTINUE
C
      RETURN
```

END

```
      SUBROUTINE INTVTR
C     ------------------------------------------------------------
C     This subroutine performs some of the calculations
C     necessary in setting up the integration matrix [ ARRINT ]
C     (the actual calculations are done in the subroutine
C     INTVTRSCRTH).  The resulting values are stored as elements
C     of a vector, each vector corresponding to a nodal point on
C     either a main or branch wing.  These vectors are stored as
C     rows in the array [ VTRINT ].
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      INTEGER ROWCNT,WGTYPE
C
      ROWCNT=0
      NINCLUTYPE=CINCLUDATA(1)
      DO 20 NWG=1,NWING
         WGTYPE=0
         IF (NINCLUTYPE.EQ.2) WGTYPE=1
         NNODEMN=NODEDATA(NWG,2)
         CALL INTVTRSCRTH (WGTYPE,NNODEMN,ROWCNT)
         ROWCNT=ROWCNT+NNODEMN
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 20
         WGTYPE=1
         DO 10 NBR=1,NBRCH
            NNODEBR=NODEDATA(NWG,2+NBR)
            CALL INTVTRSCRTH (WGTYPE,NNODEBR,ROWCNT)
            ROWCNT=ROWCNT+NNODEBR
10       CONTINUE
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTVTRSCRTH (WGTYPE,NNODE,ROWCNT)
C     ------------------------------------------------------------
C     Called by INTVTR, this subroutine performs the
C     appropriate calculations and assigns the resulting values
C     to [ VTRINT ].
C     ------------------------------------------------------------
C
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      DATA PI /3.141592653589793238/
      INTEGER ROWCNT,TOTNNODE,WGTYPE
      REAL ITRNCONST
C
      NROW=ROWCNT
      NFIRST=NNODE
      TOTNNODE=(2*NNODE)-1
      IF (WGTYPE.NE.1) GO TO 10
      NFIRST=1
      TOTNNODE=NNODE
10    NM1=TOTNNODE-1
      PIH=PI*0.5
      FNODE=FLOAT(TOTNNODE)
      PI2N=PIH/FNODE
      CONST=2./FNODE
C
      DO 40 J=NFIRST,TOTNNODE
         NROW=NROW+1
         ACOSLIM=PI-(2*J-1)*PI2N
         ITRNCONST=ACOSLIM*0.5
         NCOL=0
         DO 30 K=NFIRST,TOTNNODE
            NCOL=NCOL+1
            ARG=PI-(2*K-1)*PI2N
            SUM=0.
            DO 20 L=1,NM1
               FL=FLOAT(L)
               SUM=SUM+COS(FL*ARG)*SIN(FL*ACOSLIM)/FL
20          CONTINUE
            VALUE=CONST*(ITRNCONST+SUM)
            VTRINT(NROW,NCOL)=VALUE
30       CONTINUE
40    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTMTXSCRTH (RWCNTARR,RWCNTVTR,NNODE,LTH)
c     ------------------------------------------------------------
c     Called by INTMTX, this subroutine sets up [ ARRINT ] by
c     multiplying the appropriate rows of [ VTRINT ] by the
c     corresponding wing lengths.
c     ------------------------------------------------------------
c
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      INTEGER RWCNTARR,RWCNTVTR
      REAL LTH
c
      NROWARR=RWCNTARR
      NROWVTR=RWCNTVTR
c
      DO 20 I=1,NNODE
         NROWARR=NROWARR+1
         NROWVTR=NROWVTR+1
         NCOLARR=RWCNTARR
         NCOLVTR=0
         DO 10 J=1,NNODE
            NCOLARR=NCOLARR+1
            NCOLVTR=NCOLVTR+1
            CFNT=VTRINT(NROWVTR,NCOLVTR)
            VALUE=LTH*CFNT
            ARRINT(NROWARR,NCOLARR)=VALUE
            ARRINT(NROWARR+NNODE,NCOLARR+NNODE)=VALUE
10       CONTINUE
20    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE ADDCTRBN (FIRSTROW1,NNODE1,
     &                     FIRSTROW2,NNODE2,LTH)
C     ------------------------------------------------------
C     Called by INTMTX, this subroutine calculates and inserts
C     appropriate values into [ ARRINT ] to account for the
C     fact that the crack opening displacement at any point
C     along a crack depends on the displacement at its tip.
C     ------------------------------------------------------
C
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      DATA PI /3.14159265358979323B/
      INTEGER FIRSTROW1,FIRSTROW2
      REAL LTH
C
      NROW=FIRSTROW1
      FNODE2=FLOAT(NNODE2)
      PIN=PI/FNODE2
      CONST=-1.*PIN*LTH
      DO 20 I=1,NNODE1
         NCOL=FIRSTROW2
         DO 10 J=1,NNODE2
            ARRINT(NROW,NCOL)=CONST
            ARRINT(NROW+NNODE2,NCOL+NNODE2)=CONST
            NCOL=NCOL+1
10       CONTINUE
         NROW=NROW+1
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CXFXD
C     ------------------------------------------------------------
C     This subroutine calls the IMSL matrix solver LINV3F to
C     compute the dislocation densities.  It also finds the
C     crack opening displacements by multiplying { F } with
C     [ ARRINT ].
C     ------------------------------------------------------------
C
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA6/ COEF(400,400),NCOEFSZ
      COMMON / DATA7/ VTRINT(200,50),ARRINT(400,400)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
      REAL XFTEMP(400),WKAREA(800),DTEMP(400)
C
      DO 10 I=1,NCOEFSZ
10       XFTEMP(I)=TRAC(I)
C
      IJOB=2
      IA=400
      D1=-1.0
      CALL LINV3F (COEF,XFTEMP,IJOB,NCOEFSZ,IA,D1,D2,WKAREA,IER)
C
      CALL ORDERF (XFTEMP)
C
      CALL SIFACTOR (XFTEMP)
C
      CALL MULT (ARRINT,XFTEMP,DTEMP,NCOEFSZ,NCOEFSZ,400)
C
      CALL DISPLACEMENT (DTEMP,XDELTA)
C
      RETURN
      END
```

```
      SUBROUTINE ORDERF (XFTEMP)
C     ------------------------------------------------------------
C     Called by CXFXD, this subroutine orders the dislocation
C     densities.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      REAL XFTEMP(400)
      INTEGER ROWCNT
C
      ROWCNT=0
      DO 20 NWG=1,NWING
         NNODEMN=NODEDATA(NWG,2)
         CALL ORDERFSCRTH (NNODEMN,ROWCNT,XFTEMP)
         ROWCNT=ROWCNT+NNODEMN
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 20
         DO 10 NBR=1,NBRCH
            NNODEBR=NODEDATA(NWG,2+NBR)
            CALL ORDERFSCRTH (NNODEBR,ROWCNT,XFTEMP)
            ROWCNT=ROWCNT+NNODEBR
10       CONTINUE
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ORDERFSCRTH (NNODE,ROWCNT,XFTEMP)
c     ------------------------------------------------------------
c     Called by ORDERF
c     ------------------------------------------------------------
c
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
      REAL XFTEMP(400)
      INTEGER ROWCNT
c
      NROW=ROWCNT
      NROW1=2*ROWCNT
      NROW2=NROW1+NNODE
c
      DO 10 I=1,NNODE
         NROW=NROW+1
         NROW1=NROW1+1
         NROW2=NROW2+1
         XF(NROW,1)=XFTEMP(NROW1)
         XF(NROW,2)=XFTEMP(NROW2)
10    CONTINUE
c
      RETURN
      END
```

```
      SUBROUTINE SIFACTOR (XFTEMP)
C     ------------------------------------------------------------
C     This subroutine computes the stress intensity factors at
C     the crack tips.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      REAL XFTEMP(400)
      INTEGER NSPPT1,NSPTIP,TOTNNODE,ROWCNT
C
      ROWCNT=0
      DO 10 NWG=1,NWING
         NSPPT1=1+(NWG-1)*NPOINT
         NSP=WINGDATA(NWG,1)
         NSPTIP=NSPPT1+NSP-1
         TOTNNODE=NODEDATA(NWG,1)
         ROWCNT=ROWCNT+(2*TOTNNODE)
         NBRCH=WINGDATA(NWG,3)
         NNODE=NODEDATA(NWG,2+NBRCH)
         NROWFX=ROWCNT-NNODE
         NROWFY=ROWCNT
         CALL SIFSCRTH (NWG,NROWFX,NROWFY,NSPTIP,XFTEMP)
10    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SIFSCRTH (NWG,NROWFX,NROWFY,NSPTIP,XFTEMP)
C     ------------------------------------------------------------
C     Called by  SIFACTOR, this subroutine computes the stress
C     intensity factors and assigns them to the vector SIF.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
      REAL XFTEMP(400)
      REAL K1,K2
C
      FX=XFTEMP(NROWFX)
      FY=XFTEMP(NROWFY)
      PSI=SLTHANG(NSPTIP,2)
      SPSI=SIN(PSI)
      CPSI=COS(PSI)
C
      SCALE=1.
      NBRCH=WINGDATA(NWG,3)
      IF (NBRCH.EQ.0) GO TO 10
      BRCHLTH=WINGDATA(NWG,(5+2*NBRCH))
      WINGLTH=WINGDATA(NWG,2)
      SCALE=(BRCHLTH/WINGLTH)**0.5
C
10    K1 = (-FX*CPSI-FY*SPSI)*SCALE
      K2 = ( FX*SPSI-FY*CPSI)*SCALE
C
      XSIF(NWG,1)=K1
      XSIF(NWG,2)=K2
C
      RETURN
      END
```

```
      SUBROUTINE MULT (A,B,C,L,M,MDIM)
C     ------------------------------------------------------
C     This is a matrix multiplication subroutine : given [ A ]
C     and [ B ], it finds [ C ] = [ A ] [ B ].
C     ------------------------------------------------------
C
      REAL A(MDIM,MDIM),B(MDIM),C(MDIM)
C
      DO 20 I=1,L
         SUM=0.
         DO 10 K=1,M
10             SUM=SUM+A(I,K)*B(K)
20       C(I)=SUM
C
      RETURN
      END
```

```
      SUBROUTINE DISPLACEMENT (DTEMP,DELTATEMP)
C     ------------------------------------------------------------
C     This subroutine resolves the opening and shear
C     displacements from the global coordinate frame into the
C     local frame at each TK point.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      REAL DTEMP(400),DELTATEMP(200,2)
      INTEGER ROWCNT
C
      NODECNT=0
      ROWCNT=0
      DO 20 NWG=1,NWING
         NNODEMN=NODEDATA(NWG,2)
         CALL DISPLSCRTH (NNODEMN,NODECNT,ROWCNT,DTEMP,DELTATEMP)
         NODECNT=NODECNT+NNODEMN
         ROWCNT=ROWCNT+(2*NNODEMN)
         NBRCH=WINGDATA(NWG,3)
         IF (NBRCH.EQ.0) GO TO 20
         DO 10 NBR=1,NBRCH
            NNODEBR=NODEDATA(NWG,2+NBR)
            CALL DISPLSCRTH (NNODEBR,NODECNT,ROWCNT,DTEMP,DELTATEMP)
            NODECNT=NODECNT+NNODEBR
            ROWCNT=ROWCNT+(2*NNODEBR)
10       CONTINUE
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DISPLSCRTH (NNODE,NODECNT,ROWCNT,DTEMP,DELTATEMP)
C     ------------------------------------------------------------
C     Called by DISPLACEMENT
C     ------------------------------------------------------------
C
      COMMON / DATA4/ PHYSTK(200,5),PHYSZR(200,5)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
      REAL DTEMP(400),DELTATEMP(200,2)
      INTEGER ROWCNT
C
      NODEPT=NODECNT
      NROWDX=ROWCNT
      NROWDY=ROWCNT+NNODE
      DO 10 I=1,NNODE
         NODEPT=NODEPT+1
         NROWDX=NROWDX+1
         NROWDY=NROWDY+1
         DX=DTEMP(NROWDX)
         DY=DTEMP(NROWDY)
         PSI=PHYSTK(NODEPT,3)
         SPSI=SIN(PSI)
         CPSI=COS(PSI)
C
         DELTATEMP(NODEPT,1)=-DX*CPSI-DY*SPSI
         DELTATEMP(NODEPT,2)=-DX*SPSI+DY*CPSI
C
10    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PRINT (IW)
C     ------------------------------------------------------------
C     This subroutine prints all the input data and the computed
C     values of dislocation densities, stress intensity factors,
C     and crack opening displacements.
C     ------------------------------------------------------------
C
      COMMON / DATA1/ WELLDATA(8,4),WINGDATA(16,9),CINCLUDATA(7),
     &                TWOFACEDATA(8),NWELL,NWING,NPOINT,NFACE
      COMMON / DATA2/ POINTDATA(400,2),SLTHANG(400,2),NODEDATA(16,4)
      COMMON / DATA3/ TK(200),ZR(200)
      COMMON / DATA5/ SGMAXX,SGMAYY,SGMAXY,PWELL(8),TRAC(400)
      COMMON / DATA8/ XF(200,2),XSIF(16,2),XDELTA(200,2)
C
      WRITE (IW,900)
900   FORMAT (/,' APPLIED STRESS STATE')
      WRITE (IW,901) SGMAXX
901   FORMAT (/,'   SIGMA XX = ',F10.6)
      WRITE (IW,902) SGMAYY
902   FORMAT ('   SIGMA YY = ',F10.6)
      WRITE (IW,903) SGMAXY
903   FORMAT ('   SIGMA XY = ',F10.6)
      WRITE (IW,913) NFACE
913   FORMAT (//,' NFACE =  ',I2)
      NINCLUTYPE=CINCLUDATA(1)
      WRITE (IW,904) NINCLUTYPE
904   FORMAT (' NINCLUTYPE =  ',I2)
      IF ((NINCLUTYPE.EQ.0).AND.(NFACE.EQ.0)) GO TO 30
      IF (NFACE.EQ.2) GO TO 40
      WRITE (IW,905) CINCLUDATA(2)
905   FORMAT (/,' RADINCLU =  ',F10.4)
      WRITE (IW,906) CINCLUDATA(3),CINCLUDATA(4)
906   FORMAT (' G1 = ',F10.4,'  G2 = ',F10.4)
      WRITE (IW,907) CINCLUDATA(5),CINCLUDATA(6)
907   FORMAT (' KAP1 = ',F10.4,'  KAP2 = ',F10.4)
      WRITE (IW,908) CINCLUDATA(7)
908   FORMAT (' PRESSURE IN INCLUSION = ',F10.4)
      GO TO 30
40    WRITE (IW,909) TWOFACEDATA(1)
909   FORMAT (/,' XFACE2 =  ',F10.4)
      WRITE (IW,910) TWOFACEDATA(2),TWOFACEDATA(3),TWOFACEDATA(4)
910   FORMAT (' G1 = ',F10.4,'  G2 = ',F10.4,'  G3 = ',F10.4)
      WRITE (IW,911) TWOFACEDATA(5),TWOFACEDATA(6),TWOFACEDATA(7)
911   FORMAT (' KAP1 = ',F10.4,'  KAP2 = ',F10.4,'  KAP3 = ',F10.4)
      INCLUTYPE=TWOFACEDATA(8)
      WRITE (IW,912) INCLUTYPE
912   FORMAT (' INCLUTYPE =  ',I2)
C
```

```
30    GWGN=0
      NROW=0
      N=0
      DO 20 NWL=1,NWELL
         WRITE (IW,914) NWL
         XO=WELLDATA(NWL,1)
         YO=WELLDATA(NWL,2)
         WRITE (IW,915) XO,YO
         WRITE (IW,916) PWELL(NWL)
         LNWG=WELLDATA(NWL,3)
         WRITE (IW,917) LNWG
         DO 10 NWG=1,LNWG
            WRITE (IW,918) NWG
            GWGN=GWGN+1
            NSP=WINGDATA(GWGN,1)
            NSPPT1=1+(GWGN-1)*NPOINT
            NSPTIP=NSPPT1+NSP-1
            WRITE (IW,919)
            WRITE (IW,920) (K,POINTDATA(K,1),K,POINTDATA(K,2),
     &                      K=NSPPT1,NSPTIP)
            WRITE (IW,927) WINGDATA(GWGN,2)
            WRITE (IW,926) XSIF(GWGN,1),XSIF(GWGN,2)
            NNODE=NODEDATA(GWGN,1)
            WRITE (IW,921)
            WRITE (IW,922) (TK(N+K),XF(NROW+K,1),XF(NROW+K,2),
     &                      XDELTA(N+K,1),XDELTA(N+K,2),K=1,NNODE)
            N=N+NNODE
            NROW=NROW+NNODE
10       CONTINUE
20    CONTINUE
C
914   FORMAT (//,' WELLBORE',I3)
915   FORMAT (/,' xO = ',F10.6,'   yO = ',F10.6)
916   FORMAT (' Pressure = ',F10.4)
917   FORMAT (' Number of wings = ',I3)
918   FORMAT (//,' WING',I3)
919   FORMAT (/,' Specification points')
920   FORMAT (' x(',I3,')= ',F10.4,'   y(',I3,')= ',F10.4)
921   FORMAT (/,T6,'TK',T21,'XFx',T36,'XFy',T51,'XDopening',T66,
     &        'XDshear',T81)
922   FORMAT (5E15.6)
926   FORMAT (/,'  K1 = ',F10.4,' K2 = ',F10.4)
927   FORMAT (' Wing Length = ',F10.4)
C
      RETURN
      END
```