1983-04-01

Development of a Computer-aided Analysis Package for Linear Elastic Fracture Mechanics

William E. Warkentin
Brigham Young University - Provo

BYU ScholarsArchive Citation

Warkentin, William E., "Development of a Computer-aided Analysis Package for Linear Elastic Fracture Mechanics" (1983). *Theses and Dissertations*. 3455.
https://scholarsarchive.byu.edu/etd/3455

DEVELOPMENT OF A COMPUTER-AIDED ANALYSIS PACKAGE FOR LINEAR ELASTIC FRACTURE MECHANICS

WILLIAM E. WARKENTIN

DEVELOPMENT OF A COMPUTER-AIDED ANALYSIS PACKAGE FOR LINEAR ELASTIC FRACTURE MECHANICS

A Thesis
Presented to the
Department Of Civil Engineering
Brigham Young University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
William E. Warkentin
April 1983

This thesis, by William E. Warkentin, is accepted in its present form by the department of Civil Engineering of Brigham Young University as satisfying the thesis requirement for the degree of Master of Science.

Steven E Benzley

Steven E. Benzley
Committee Chairman

S. Olani Durrant
Committee Member

17 Aug 83
Date

Henry N. Christiansen
Department Chairman

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude for the patience, guidance, and encouragement of Dr. Steven E. Benzley and Dr. S. Olani Durrant during the research and preparation of this thesis. Much appreciation also goes to the entire Civil Engineering Faculty at Brigham Young University for their professional example and instruction.

The financial assistance received through Dr. Benzley from Sandia Laboratories was appreciated, as was the initial coding of the finite element program by Dr. Benzley and Zelma E. Beisinger.

Lastly, the author wishes to express thanks to his mother for her support and love during the preparation of this thesis.

LIST OF FIGURES

Figure

Figure

CHAPTER ONE

Introduction

Mathematical modeling of linear elastic structures often gives solutions that have stress and strain singularities at particular points in the body. Specifically these points are crack tips, re-entrant corners, or discontinuities. These features are of major concern to the stress analyst because they can lead to catastrophic failure.

An interactive computer-aided analysis package that would allow for the easy modification of the size and location of crack tips would be very beneficial to the engineer. With minimal effort, the engineer could model a structure with many different cracked configurations. These models could be compared and the most critical one for that particular structure would be determined. Such a process is important to ensure that a structure is designed for its most critical condition.

The work done in this thesis is a part of a research project sponsored by Sandia National Laboratories to model failure of Glass Ceramic components. The specific work done in this thesis is broken down into three areas: 1) a library search on the subjects of fracture mechanics and glass ceramics, 2) the modification of an existing finite element fracture mechanics program (CHILES 2) to make it interactive, and 3) the analysis of three test specimen designs to demonstrate the use of the modified finite element program.

The test specimens will be analyzed according to the following procedure. Meshes of the different specimens will be made up using the mesh generation program QMESH. Material properties and loading conditions will be assigned to the specimens and they will be analyzed using the modified version of CHILES 2. Graphical displays of the results of the loading conditions on the specimens will be shown using the MOVIE.BYU computer graphics system. These results will then be discussed.

The computer package consisting of the programs QMESH.BYU, CHILES.BYU, and MOVIE.BYU allows an engineer to easily modify the size and location of cracks within a structure. Critical points in the structure can then be rapidly investigated.

CHAPTER TWO

Fracture Mechanics

"The fundamental principle of fracture mechanics is that the stress field ahead of a sharp crack in a structural member can be characterized in terms of a single parameter, K, the stress intensity factor, that has units of psi. This parameter, K, is related to both the nominal stress level (σ) in the member and the size of the crack present."[6] Whereas unflawed members can be loaded to various stress levels, σ, so can structural members or test specimens which have flaws be loaded to various levels of K.

"Linear-elastic fracture mechanics technology is based on an analytical procedure that relates the stress-field magnitude and distribution in the vicinity of a crack tip to the nominal stress applied to the structure, to the size, shape, and orientation of the crack-like discontinuity, and to material properties."[6] Three types of relative movement can be defined to show the movement of two crack surfaces. This movement is defined as either Mode I, II, or III which correspond to the opening, shear, or tearing modes (see Fig. 1). The stress field at a crack tip can be treated as either one of these modes or any combination of them.

The equations for the stress and displacement fields "show that the distribution of the elastic-stress fields and of the deformation fields in the vicinity of the crack tip are invariant in all components subjected to a given mode of deformation and that the magnitude of the elastic-stress field can be described by single-term parameters,K1, K2, K3, that correspond to Modes I, II, III, respectively. Consequently, the applied stress, the crack shape and size, and the structural configuration associated with structural components subjected to a given mode of deformation affect the value of the stress-intensity factor but do not alter the stress-field distribution."[6]

The magnitude of the stress intensity factor, K, is directly related to the nominal stress (σ) and the square root of the crack length (a).

$$K=(f(g))(\sigma)(\sqrt{a})$$

where f(g) is a parameter that depends on the specimen and crack geometry and has been the subject of extensive investigations and research. Barker has suggested material testing to determine this.

A review of the literature pertinent to the tasks of this thesis has been done. This survey incorporated a computer assisted search of the "Engineering Index" [1 and 2] to find recently published articles. This review used the following combinations of key words in searching abstracts listed in References 1 and 2.

1. Glass Ceramic - fracture mechanics
2. Glass Ceramic - finite element
3. Glass Ceramic - bimaterial interface

The significant articles of this survey are included in the bibliography.

References 3 and 4 provide very basic and complete descriptions of brittle fracture of ceramic materials and References 5 and 6 treat the basics of linear elastic fracture mechanics. A recent summary of the state-of-the art in the application of fracture mechanics to glass ceramics is given in Reference 7. Information on fracture mechanics data of glass ceramics is given in References 8-24.

The problem of fracture of a bimaterial bond is covered in general in Reference 25. Of particular interest is the nature of the oscillatory character of the stresses in the near crack tip region [26]. The real effect of these oscillating stresses are covered in References 27-31. Fracture toughness data for the bond strength of ceramic to metal joints is discussed by Pabst [32]. The problem of a crack perpendicular to a bimaterial interface is treated in References 33-39.

The actual mathematical modeling of a bimaterial interface has received relatively little attention. Lin and Mar [40] successfully developed a hybrid crack tip finite element for the bimaterial problem. Recently, Flemming, et. al. [41], compared both finite element and edge function methods to problems of bimaterial interfaces. Here both conventional and special elements were successfully used to solve the crack tip problems. The "quarter point" element as presented by Henshall and Shaw [47] and the generalized enriched element derived by Benzley [46] both both have possibilities for use with bimaterial interfaces.

In this thesis two calculations will be done concerning fracture mechanics. The first will determine K between a S-Glass/Inconnel interface using the bimaterial interface option of CHILES. The second, a classical analysis, will analyze the S-Glass in a specimen, assuming that surface cracks exist, and using a measured value of K for the S-Glass.

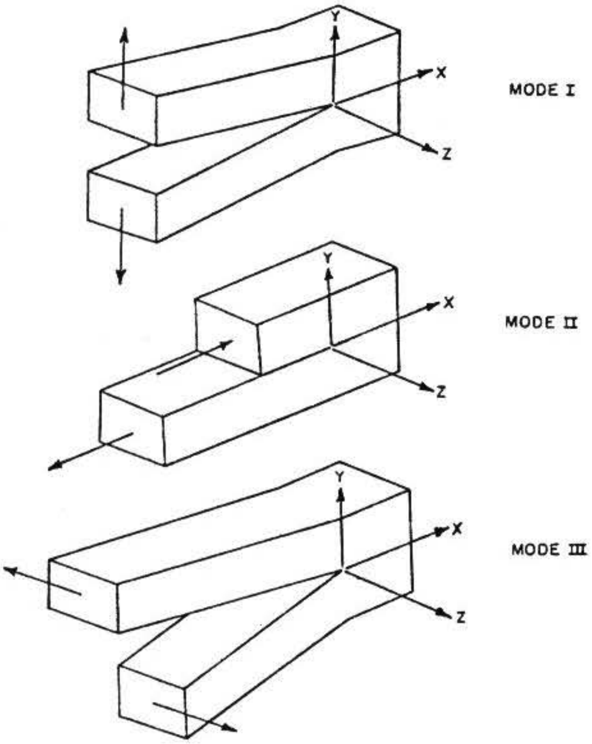

Figure 1. Types of relative crack movement

CHAPTER THREE

CHILES.BYU

Chiles 2 is a FORTRAN finite element computer program that calculates the intensities of linear elastic singularities in isotropic and orthotropic materials. It was authored by Steven E. Benzley and Zelma E. Beisinger, and prepared by Sandia Laboratories, Albuquerque, New Mexico and Livermore, California for the United States Department of Energy. The abstract for the program reads as follows:

> CHILES 2 is a finite element computer program that calculates the strength of singularities in linear elastic bodies. A generalized quadrilateral finite element that includes a singular point at a corner node is incorporated in the code. The displacement formulation is used and interelement compatibility is maintained so that monotone convergence is preserved. Plane stress, plain strain, and axisymmetric conditions are treated. Isotropic and orthotropic crack tip singularity problems are solved by this version of the code but any type of singularity may be properly modeled by modifying selected subroutines in the program.[46]

The above referenced version of CHILES 2 is not interactive. To run this version the user has to make a file consisting of a set of problem identification lines. The information in the lines has to appear in certain columns making the task of creating the file cumbersome, time consuming, and prone to error. An interactive version of CHILES 2 named CHILES.BYU was developed as a part of this thesis. An interactive program allows a user to be questioned by the computer program such that the necessary input data may be generated. The modifications to make CHILES 2 interactive are discussed in what follows.

In modifying CHILES it was essential that the user of the program be able to easily change data if erroneous information was input. Therefore, questions asking if changes need to be made are placed at the end of sections dealing with common information. For example, after the questions dealing with the initialization process are asked, the user is then asked if any changes are to be made. If no changes are to be made the program continues on to the next block of questions to be asked, namely, questions dealing with the material properties of the specimen. The program continues through the different blocks of information until the user is satisfied that all the information is input correctly.

The program then goes through the solution of the problem in question. To inform the user that the program is in the calculation stage, the message

"CHILES IS EXECUTING" appears on the terminal. When calculations cease, the message that two files, CHILES.LIS and CHILES.MOV, have been made appears. CHILES.LIS is a file listing information such as stresses, strains, connectivity, and node point coordinates of a problem. CHILES.MOV is a file containing plotting information that is to be used in conjunction with a graphics program in displaying computed data. The program then tells the user that calculations are complete and signs off. The limitations and capabilities of CHILES.BYU are explained in Appendix A.

CHAPTER FOUR

Actuator Specimen Design Analysis

I. DESCRIPTION OF TEST SPECIMENS

As a part of this thesis the modeling of fracture of glass ceramics and glass ceramic/metal interfaces will be demonstrated using CHILES.BYU.

Three test specimens will be designed. The specimens will be composed of three materials. The properties of these materials are given as follows:

Inconnel 718	Modulus of Elasticity:	29.0 E6 psi
	Poisson's Ratio:	0.294
	Yield Stress:	13.0 E4 psi
	Expansion:	0.00975
S-Glass	Modulus of Elasticity:	13.0 E6 psi
	Poisson's Ratio:	0.200
	Yield Stress:	None
	Expansion:	0.00882
Hastalloy C-276	Modulus of Elasticity:	29.8 E6 psi
	Poisson's Ratio:	0.300
	Yield Stress:	57.8 E3 psi
	Expansion:	0.00930

The three specimens are all variations of the same basic shape with different dimensions to produce failures at different locations in the specimen. The goal is to have three different designs that will produce failures at:

a) the S-Glass/Inconnel interface
b) the S-Glass alone
c) the S-Glass/Hastalloy interface

The basic test specimen from which the others were modeled is shown in Fig. 2.

The S-Glass/Inconnell interface specimen is designed to produce a critical stress area between the Inconnel and S-Glass. The S-Glass specimen is designed to produce failure on the S-Glass surfaces. This is done by maximizing the bending that will occur in the S-Glass and minimizing the effect of the S-Glass/Inconnel interface. The S-Glass/Pin failure specimen is designed to produce a critical stress area between the S-Glass and Hastalloy. This occurs at the point of maximum bending at the bottom of the specimen between the S-Glass/Hastalloy interface.

Since the specimens are symmetrical about their central axis only half of each specimen is modeled. The finite element meshes were created by keeping each element as square as possible though a length to width ratio of 2:1 was allowed. Critical areas were defined more completely by a finer mesh while areas of lesser interest and importance were modeled more coarsely. Three colors were assigned to the three different materials that make up the specimens. Hastalloy is gold, S-Glass is red, and Inconnel is turquoise. Pictures of the specimen designs, with the finite element grid overlaying each specimen, are shown in Figs. 3 thru 5.

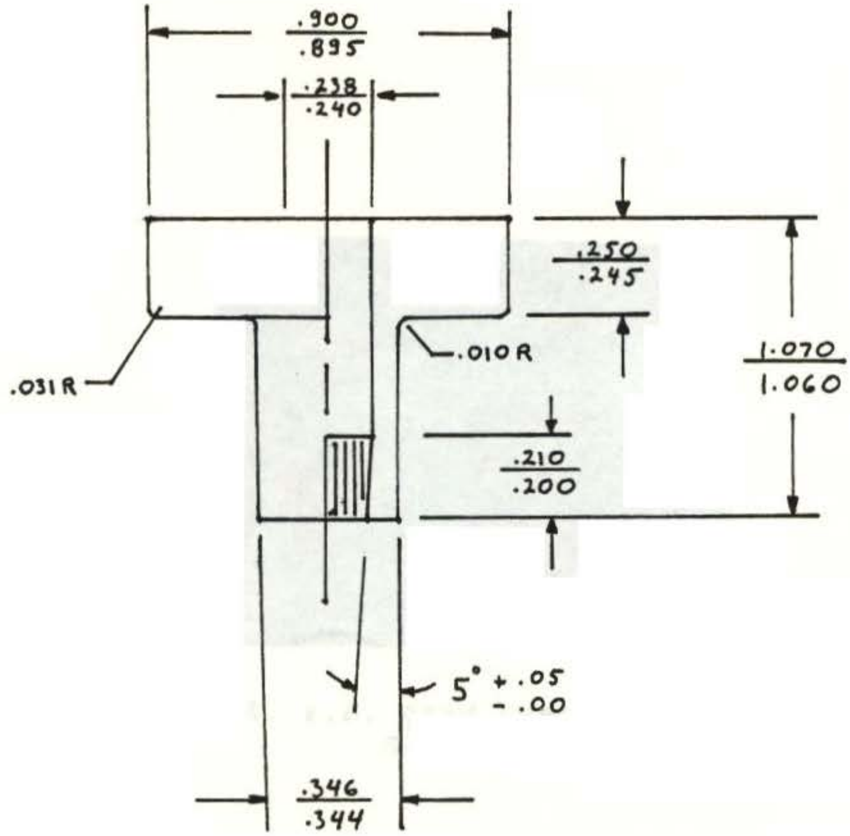

Figure 2. Basic Test Specimen

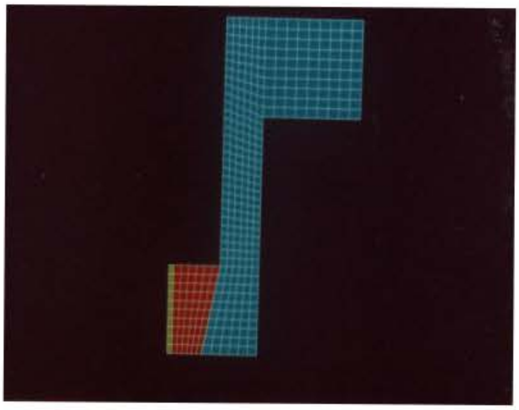

Figure 3. F.E. grid over color parts
S-Glass/Inconnel

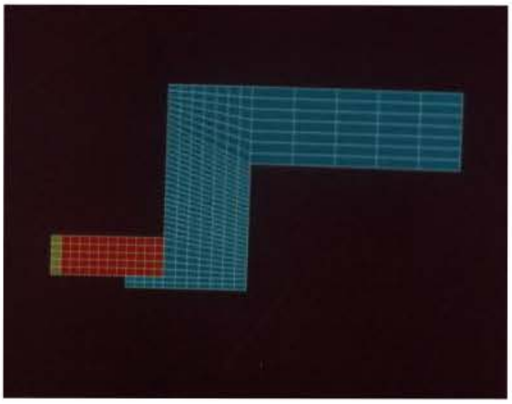

Figure 4. F.E. grid over color parts
S-Glass

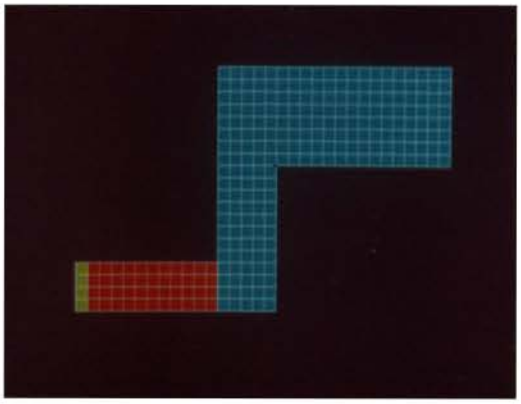

Figure 5. F.E. grid over color parts
S-Glass/Pin

II. RESULTS OF TEST SPECIMENS ANALYSES

Boundary conditions are applied to the finite element models to allow them to behave in a manner which simulates their real behavior. A pressure of 50,000 psi is applied to the inner surface of the specimens. The models are constrained, and the pressure is applied, according to the basic test specimen shown in Fig. 6.

The deformed geometries of the models must be magnified many times to allow the viewing of the deformed shape. The deformation of each model, produced by the 50,000 psi load, is shown in Figs. 7 thru 9.

The stresses developed in the specimens can be thought of as being distributed in the model according to contours or fringes. The fringes will be modeled in five colors with blue being the lowest stress and red being the highest stress. The Von Mises color stress fringes in the S-Glass of each model are shown in Figs. 10 thru 12.

The radial color stress fringes in the S-Glass of each model are likewise important. These stresses are shown in Figs. 13 thru 15.

The critical area of the S-Glass/Inconnel interface specimen is the region where the S-Glass meets the Inconnel. This interface is modeled more accurately by defining a "near field" mesh of the region. A section of the S-Glass/Inconnel mesh is, in essence, cut out and this becomes the near field region. The displacements of the nodes of the S-Glass/Inconnel mesh, caused by the 50,000 psi load, are applied to the nodes of the near field mesh. This near field mesh, removed from the main specimen, is shown in Fig. 16.

Two cracks are then modeled into the near field mesh to determine the strength of the crack tip singularity that forms between the two materials. The size of the two cracks modeled are .001" and .005" respectively. The meshes for these two cracks are shown in Figs. 17 and 18.

As stated previously, displacements are applied to the nodes of the near field mesh which are common to the main mesh. The pressure of 50,000 psi is applied to the two near field meshes and this results in the deformed shapes shown in Figs. 19 and 20.

Each of these cracks has two stress intensity factors, KI and KII. The values of these two factors for each crack length are shown in the following table.

Crack Size	KI	KII
.001"	11,720	-13,473
.005"	13,881	-20,907

A classical analysis of a surface crack on the S-Glass will now be described. Data supplied [50] indicates that machining flaws on the surface of the S-Glass could range from 50-100 microns deep and be 100-200 microns long. Such flaws could be assumed to be semi-circular. For such a flaw, the stress intensity factor, KI, can be determined from the following analysis [5].

For a semi-circular flaw

$$KI=1.1(\sigma)^{1/2}(\sigma)(a/Q)^{1/2}$$

where KI=Stress Intensity Factor
σ=Applied Stress
a=Crack depth
Q=Flaw Shape parameter (apprx.=2.4)

Thus $KI=1.259(\sigma)(a)^{1/2}$

Using the measured value of K for S-Glass [50] of 1620 psi; we obtain

$$1287/\sigma=(a)^{1/2}$$

Applying this equation the graph in Figure 21 can be drawn. This figure can be used to determine the critical flaw depth for a semi-circular crack along the surface of the S-Glass. Note that the expected flaw size of 100 microns (i.e. .003937") corresponds to a stress level of approximately 20,000 psi stress.

Plots of hoop and radial stresses along the surface of the S-Glass for the three specimens are shown in Figures 22 thru 24. The value of tensile stress in the hoop direction (i.e. circumferential), for all three specimens, is significantly lower than the tensile stress in the radial direction, thus failures originating from surface flaws on the S-Glass are most likely to occur from radial stresses. Figures 22 thru 24 show that the highest values of tensile stress and the greatest S-Glass surface area above 20,000 psi tension occur in the S-Glass specimen. Consequently this specimen is most likely to produce failures along the S-Glass surface away from the S-Glass/Inconnel interface.

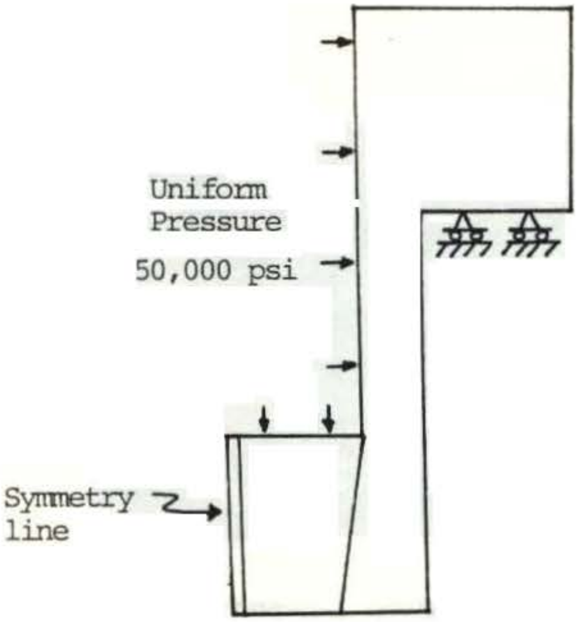

Figure 6. Boundary conditions on basic test specimen

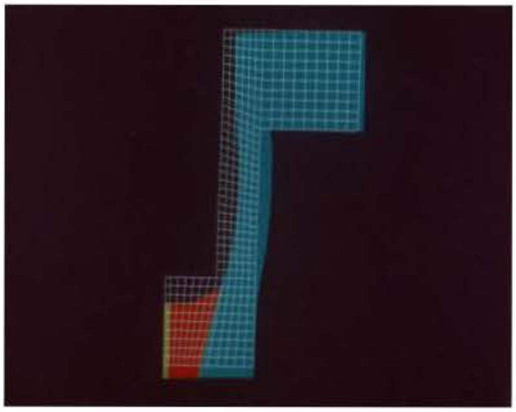

Figure 7. Deformed geometry (S-Glass/Inconnel)

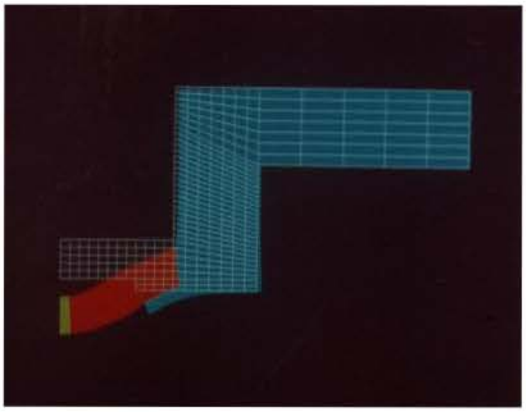

Figure 8. Deformed geometry (S-Glass)

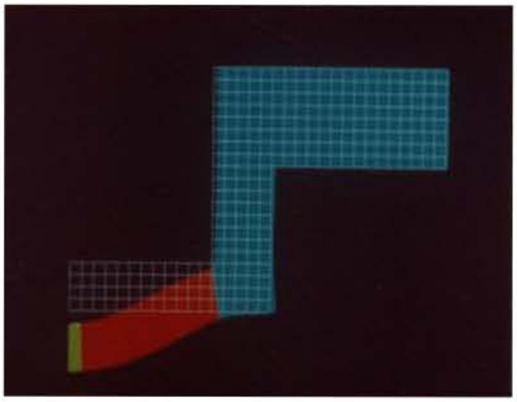

Figure 9. Deformed geometry (S-Glass/Pin)

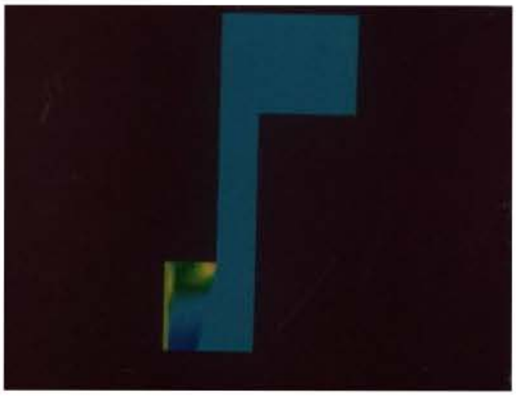

Figure 10. Von Mises Stress Fringes (S-Glass/Inconnel)
Range: 2,755 psi - 77,000 psi

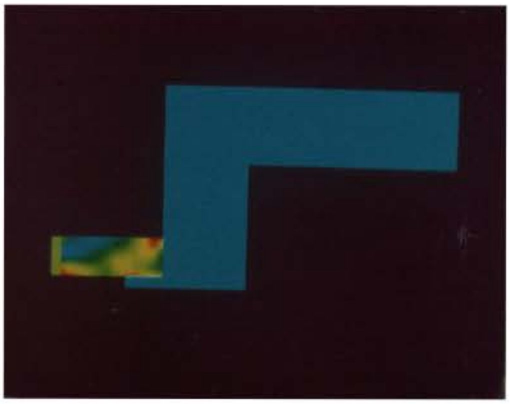

Figure 11. Von Mises Stress Fringes (S-Glass)
Range: 5,506 psi - 170,000 psi

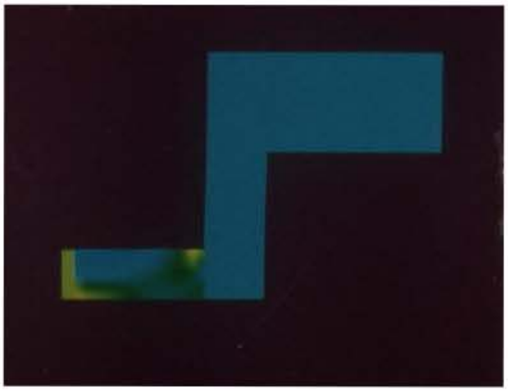

Figure 12. Von Mises Stress Fringes (S-Glass/Pin)
Range: 13,800 psi - 319,000 psi

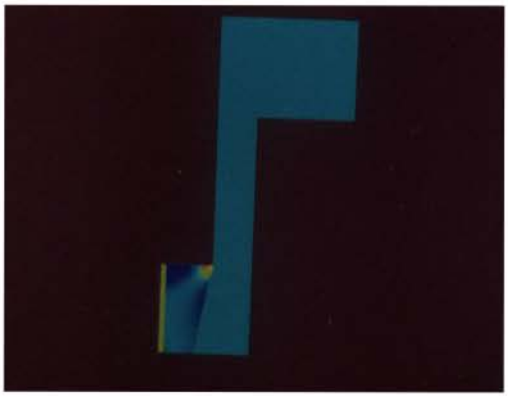

Figure 13. Radial Stress Fringes (S-Glass/Inconnel)
Range: -400 psi - 23,750 psi

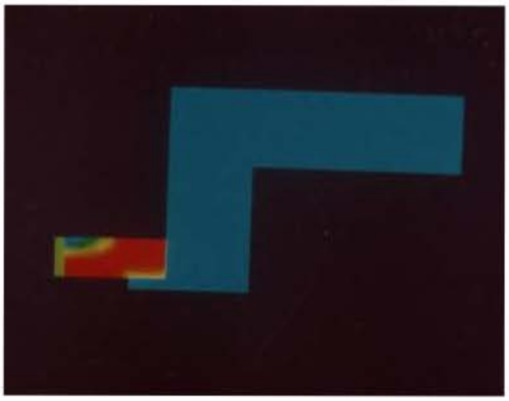

Figure 14. Radial Stress Fringes (S-Glass)
Range: -171,430 psi - 100,000 psi

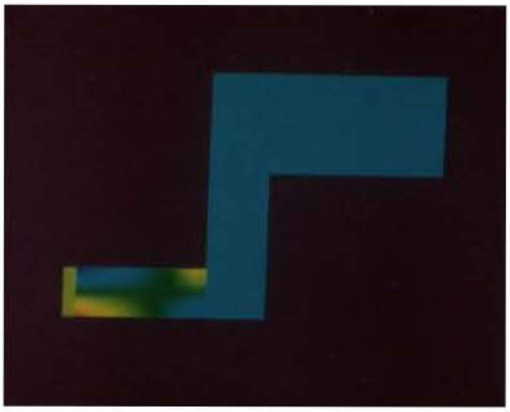

Figure 15. Radial Stress Fringes (S-Glass/Pin)
Range: -189,000 psi - 309,000 psi

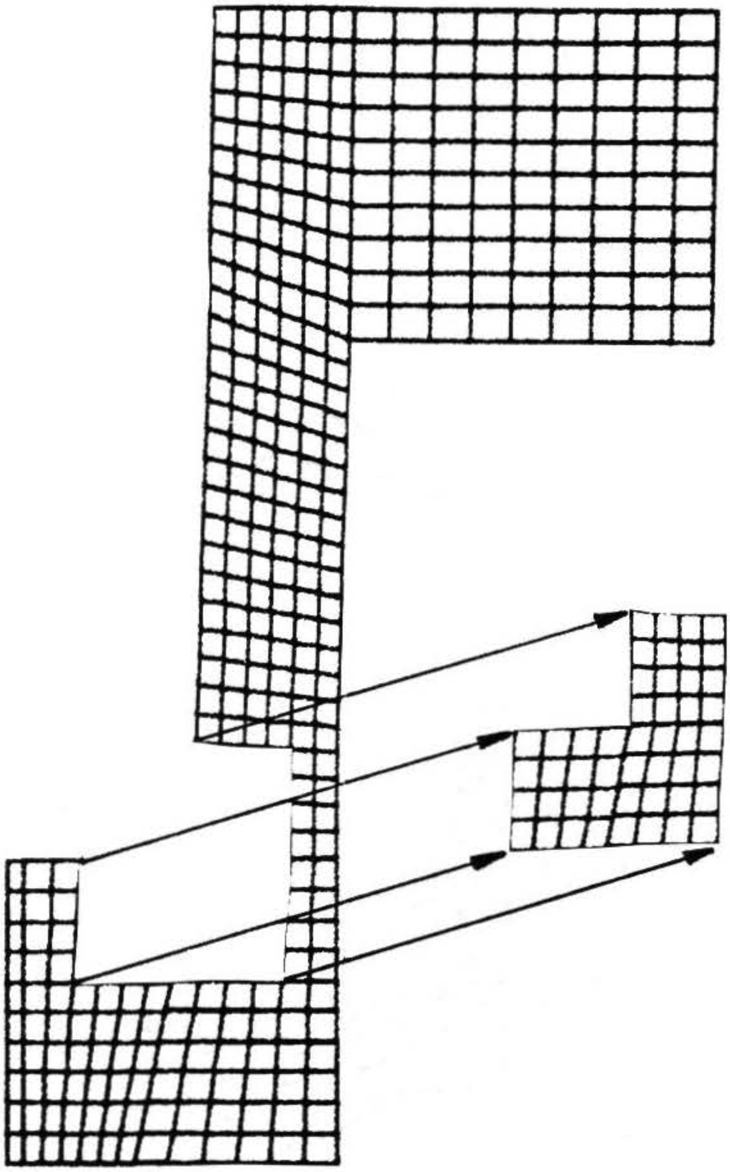

Figure 16. Near field mesh removed from main specimen

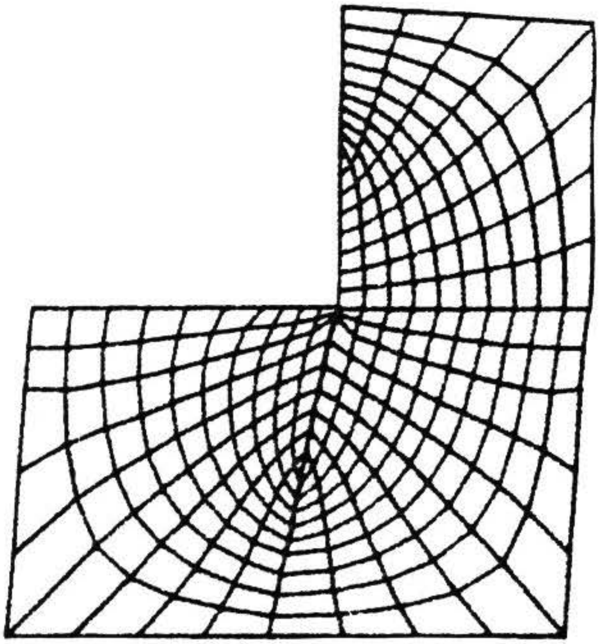

Figure 17. Near field mesh - .001" crack

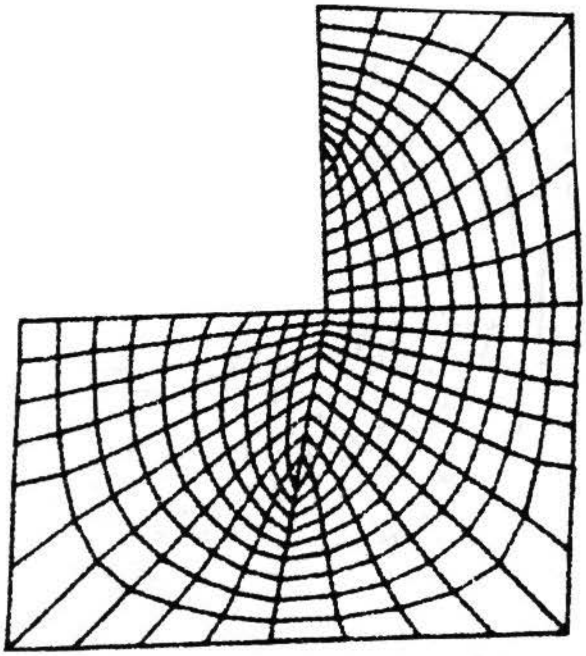

Figure 18. Near field mesh - .005" crack

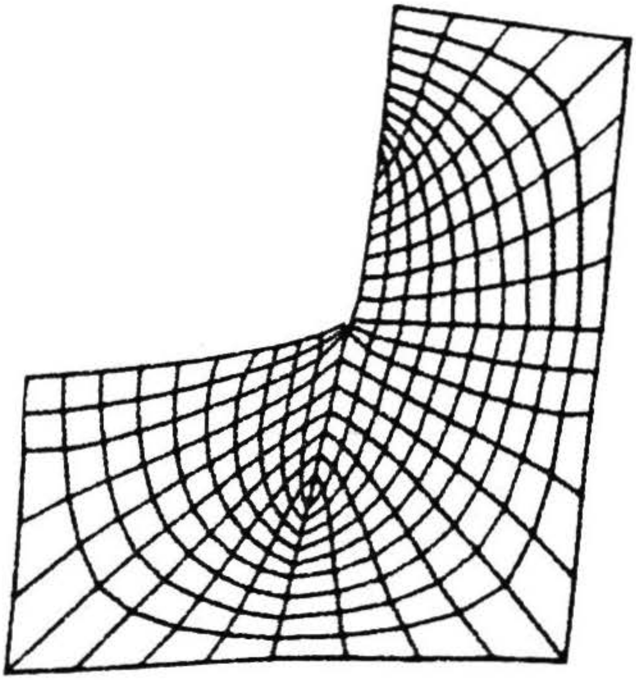

Figure 19. Deformed shape - .001" crack

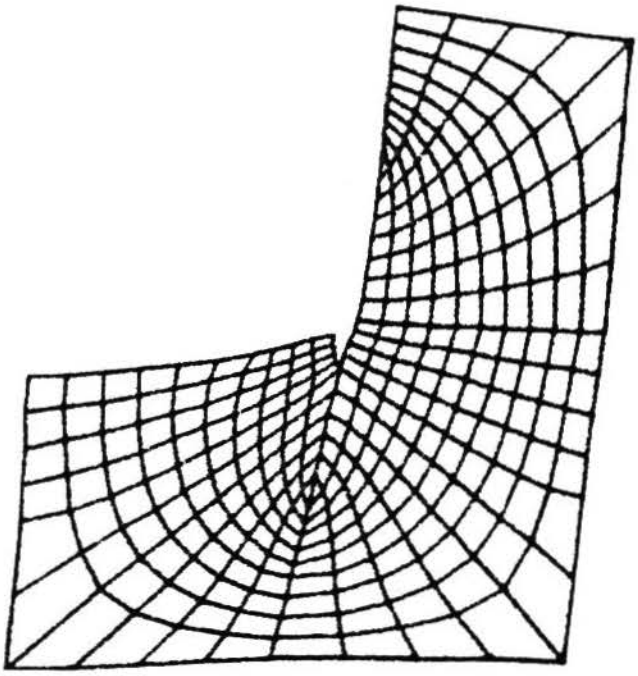

Figure 20. Deformed shape - .005" crack

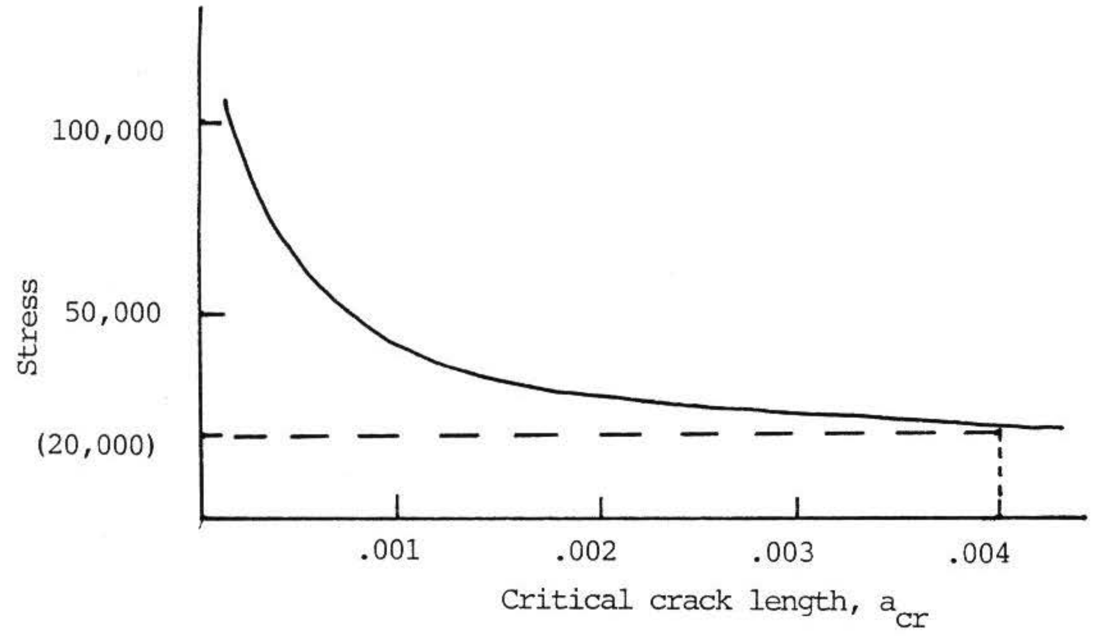

Figure 21. Stress vs. crack size

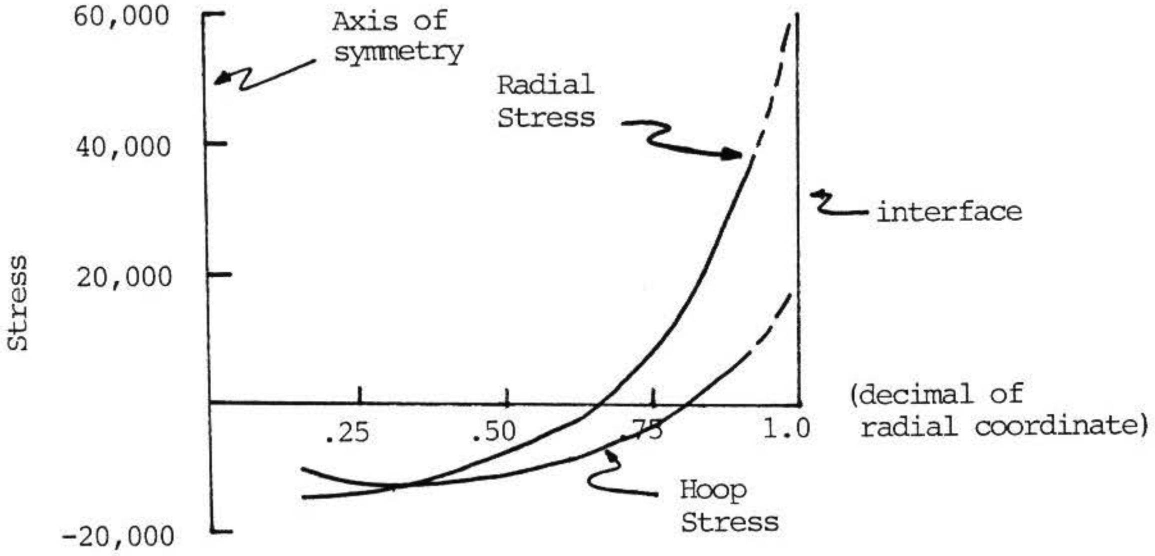

Figure 22. Hoop and radial stress
S-Glass/Inconnel

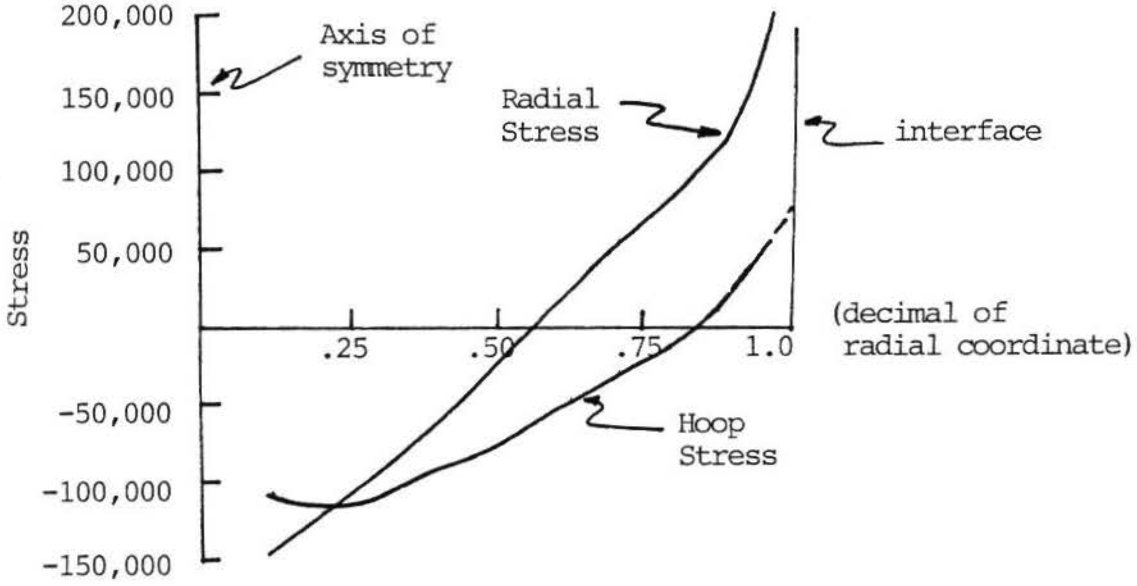

Figure 23. Hoop and radial stress
S-Glass

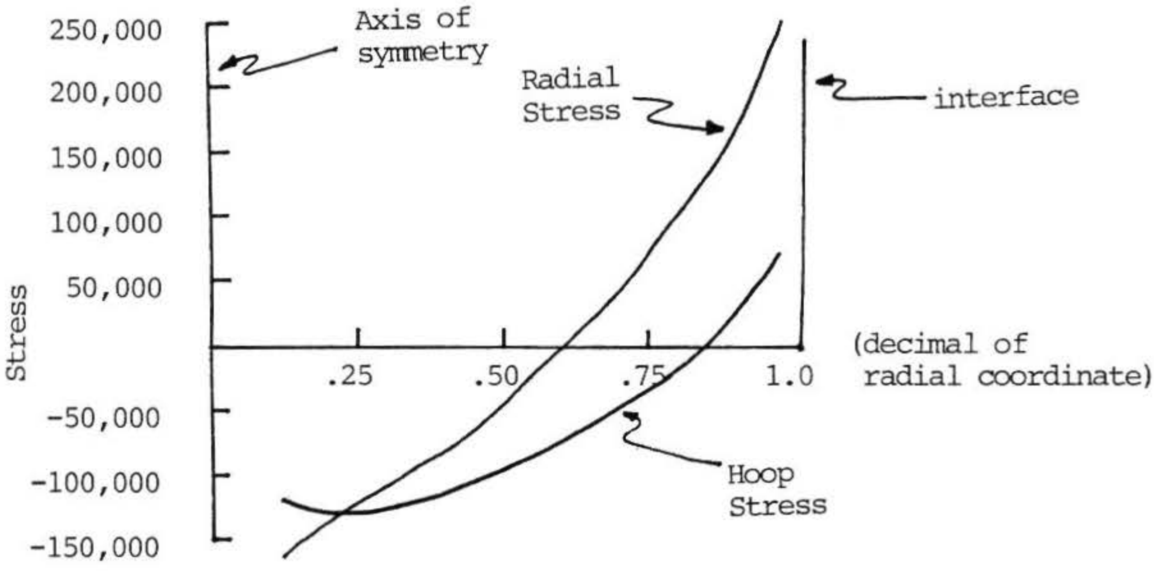

Figure 24. Hoop and radial stress
S-Glass/Pin

III. EVALUATION OF RESULTS

The specimen designs were all derived from a basic specimen and modeled to produce different critical areas. The results obtained from the analysis of the three test specimens indicate that each specimen will fail at a different location.

The S-Glass/Inconnel interface failure specimen was designed to produce a critical region at the S-Glass and Inconnel interface. To further analyze this region a near field mesh was defined and two different size cracks were introduced between the two materials. A fracture mechanics analysis using the near field meshes shows that as the size of the crack increases, the value of KI increases. Material testing as suggested by Barker is required at this point to determine a value of toughness for such a bimaterial interface.

The S-Glass failure specimen was modeled with a geometric discontinuity along the lower surface. An area of high stress should be produced at this discontinuity. Analysis of this problem using CHILES.BYU shows that high stresses exist at the discontinuity but also that significant bending is caused by the discontinuity. These bending regions produce high tensile stresses at the top and bottom of the S-Glass on opposite ends of each other. Therefore, this specimen is very likely to produce failures from cracks existing on the S-Glass surface.

The S-Glass/Pin specimen was designed to produce high stresses at the area of maximum bending. This specimen's dimensions produced this maximum condition at the bottom of the specimen at the Hastalloy pin. Analysis using CHILES.BYU proved this to be correct. The bond between the pin and the S-Glass is, therefore, the critical region in this specimen.

The designs of the three test specimens has demonstrated the use of a computer aided analysis package for linear elastic fracture mechanics. By using a basic specimen shape and modifying certain dimensions, different regions in each specimen become critical. Finite element analysis of these specimens has shown that failures can initiate at material interfaces, geometric discontinuities, and regions of maximum bending. By analyzing these regions with fracture mechanics principles critical areas can be completely evaluated.

Essentially two fracture mechanics analyses have been used. The first analysis used the bimaterial interface crack option in CHILES.BYU. The stress intensity factors for different crack sizes between the S-Glass and Inconnel were computed. To determine if these interface cracks are critical, the computed stress intensity factor must be compared to laboratory measured values of fracture toughness of S-Glass/Inconnel bonds. By comparing the computed values and measured values critical crack size can be computed.

The second analysis assumed machining flaws would produce surface cracks from 50-100 microns deep and 100-200 microns long. Assuming the flaws to be semi-circular a stress intensity value KI can be determined to be $KI=1.259(\sigma)(\sqrt{a})$. Using a value of KI=1620 psi a graph of stress vs. critical length can be drawn. By plotting the hoop and radial stresses along the surface of the S-Glass the area where stress corresponds to critical crack size can be found.

CHAPTER FIVE

Conclusions and Recommendations

The principal contribution of this thesis was the development of an interactive version of the CHILES Fracture Mechanics program. With this program an engineer can easily modify the size and location of cracks. A structure's shape can also easily be changed to analyze different situations. Fracture mechanics analyses can then be performed on these situations.

Two methods have been used to perform fracture mechanics analyses. The single or bimaterial interface option in CHILES.BYU can be used to calculate the stress intensity factors of a crack. A classical surface flaw analysis can also be performed by assuming a crack size and shape and using a computed stress field. Critical stresses for specific crack sizes and shapes can then be calculated.

The package of QMESH.BYU ---- CHILES.BYU ---- MOVIE.BYU allows an engineer to rapidly investigate different crack configurations and display the results using color graphics. This visual aid allows the engineer to analyze and investigate the results of critical areas.

Although CHILES.BYU is interactive and "user-friendly" the following modifications could be incorporated into the software. 1- Error checks could be used such that if a number is input that does not follow the prescribed format, the program would not terminate. 2- More selective editing could be used than is available. Instead of having to modify a whole block of entries it would be useful to edit just one individual entry at a time and not be required to input all entries to correct a single error. 3- An echo of data input should be provided to allow the user to see all the input data. These above improvements would all be very useful to improve the "user-friendliness" of CHILES.BYU.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Compendex, (Copr. Engineering Index Inc.)

[2] ISMEC: Mechanical Engineering, (Copr. Cambridge Sci. Abs)

[3] Liebowitz H., Fracture, An Advanced Treatise, VII, 1972, Academic Press, New York.

[4] Jayatilaka, Ayal de S., Fracture of Engineering Brittle Materials. 1979, Applied Science Publishers, Ltd., London.

[5] Broek, D., Elementary Engineering Fracture Mechanics, 1974, Noordhoff, Leyden.

[6] Rolfe, S.T., and J. M. Barson, Fracture and Fatigue Control in Structures, 1977, Prentice Hall, Englewood Cliffs, New Jersey.

[7] Mecholsky, J.J., "Fracture Mechanics Analysis of Glass Ceramics," Special Pub., Advances in Ceramics, American Ceramic Society, 1982.

[8] Szendi-Horvath, G., "Fracture Toughness Determination of Brittle Materials Using Small to Extremely Small Specimens," Eng. Fract. Mech., v13, no. 4, 1980.

[9] Hofmann, V., O. Henkel, and D. Schulze, Mechanical Behavior of Solids in Dependence on Their Structure - Glass, Ceramics, Polymers, 1976," Arag-Verlag, Berlin, 1976.

[10] Marion, R.H., "Use of Indentation Fracture to Determine Fracture Toughness," ASTM STP 678, 1979.

[11] Ritter, J.E., "Engineering Design and Fatigue Failure of Brittle Materials," Fract Mec. of Ceram, v.4, Symp on Fract Mech of Ceram, Proc, Pa. State Univ., University Park, 1977.

[12] Ritter, J.E., "Strength and Failure Predictions for Glass and Ceramics," J. Am. Ceram. Soc., v.59, no. 11, 1976.

[13] Wiederhorn, S.M., E.R. Fuller, J. Mandel, and A.G. Evans, "Error Analysis of Failure Prediction Techniques Derived from Fracture Mechanics," J. Am. Ceram. Soc., v. 59, no. 9-10, 1976, p. 403-411.

[14] Pisarenkg, G.S., Y.I. Kozub, U.G. Soluganou, and A.P. Poleshko, "Evaluation of the Strength of Brittle Materials by Studying the Fracture Surface," Acad. of Sci., Kiev, UKr SSR.

[15] Mecholsky, J.J., and S.W. Freiman, "Fracture Surface Analysis of Glass Ceramics," Eleventh International Glass Congress, Prague, Czechosolvakia, 1977.

[16] Mecholsky, J.J., S.W. Freiman, and R.W. Rice, "Fracture Surface Analysis of Ceramics," J. Mat. Sci. 11, 1310-19, 1976.

[17] Bansal, G.K., W. Duckworth, and D.E. Niesz, Bull. Am. Ceram. Soc. 55, (3), 298-92, 1976.

[18] Lewis III, D.L., "Fracture Strength and Mirror Size in A Commercial Glass-Ceramic," J. Am. Ceram. Soc. 64, (2) 82-86, 1981.

[19] Mecholsky, J.J., C.T. Moynihan, P.B. Macedo, and G.R. Srinivasn, "Microstructure and Properties of an Infrared Transmitting Chalcogenide Glass Ceramic," J. Mat. Sci. 11, 1952-60, 1976.

[20] Govila, R.K., K.R. Kinsman, and P. Beardmore, "Fracture Phenomenology of a Lithium-Aluminum-Silicate Glass Ceramic," J. Mat. Sci. 13, 2081-91, 1978.

[21] Swearengen, J.C., E.K. Beauchamp, and R.J. Eagan, "Fracture Toughness of Reinforced Glasses," Fracture Mechanics of Ceramics op.cit., v. 4, 973-987, 1978.

[22] Evans, A.G., A.H. Hever, D.L. Porter, "The Fracture Toughness of Ceramics," Fracture 1977, v. 1, ICF4, Waterloo, June 1977.

[23] Henshall, J.L., D.J. Roucliffe, and J.W. Edington, "The Measurement of K and Subcritical Crack Propagation Rates in Hot Pressed Sic and Si N ." Fracture 1977, v. 3, ICF4, Waterloo, June 1977.

[24] Abdel-Latif, A I.A., R.E. Tressler, and R.C., Bradt., "Fracture Mirror Formation in Single Crystal Alumina," Fracture 1977, v. 3, ICF4, Waterloo, June 1977.

[25] Anderson, G.P., S.J. Bennett, and K.L. DeVries, Analysis and Testing of Adhesive Bonds, Academic Press, New York, 1977.

[26] Williams, M.L., "The Stresses Around a Fault or Crack in Dissimilar Media," Bull. Seis. Soc. Am., V. 49, 1959, pp 199-204.

[27] Rice, J.R., G.G. Sih, "Plane Problems of Cracks in Dissimilar Media," J. of App. Mech., ASME, V. 32, 1965, pp. 418-423.

[28] Erdogan, F., "Stress Disbribution in Bonded Dissimilar Materials with Cracks," J. of App. Mech., ASME, V. 32, 1965, pp. 403-410.

[29] England, "A Crack Between Dissimilar Media," J. of App. Mech., ASME, v. 32, 1965, pp. 400-402.

[30] Mak, A.F., and L.M. Keer, "No-Slip Edge Crack on a Bimaterial Interface," J. of App. Mech., ASME, v. 47, 1980, pp 816-820.

[31] Theocaris, P.S., and E.E. Gdoutas, "Stress Singularities in Cracked Composite Full-Planes," Int, J. Fracture, v. 13, n. 6, 1977, pp 763-773.

[32] Pabst, R., and G. Elssner, "Bond Fracture Strength in Ceramic-to-Metal Joints," Fracture 1977, ICF4, v. 3, pp 1025-1029.

[33] Tracey, D.M., and T.S. Cook, "STRESS Distribution in a Cracked Bimaterial Plate," Fracture 1977, ICF4, v. 3, 1977, pp 1055-1058.

[34] Theocaris, P.S., J. Milios, "Crack-Arrest at a Bimaterial Interface," Int J. Solids Struct., v. 17, n. 2, 1981, pp 217-230

[35] Ioakimidia, N.I., P.S. Theocaris, "Practical Evaluation of STRESS Intensity Factors at Semi-Infinite Crack Tips," Eng. Fract. Mechj., v. 13, n. 1, 1980, pp 31-42.

[36] Theocaris, P.S., and E.E. Gdoutos, "STRESS Singularities in Cracked Composite Full-Planes," Int. J. Fracture, V. 13, N. 6, Dec 1977, pp 763-773.

[37] Erodogan, F., "Fracture of Composite Materials," Prospects of Fract. Mech. Int. Conf., Proc. Delft, Neth. June 1974, pp 477-492.

[38] Ashbaugh, N., "STRESS Solution for a Crack at an Arbitrary Angle to an Interface," Int. J. Fract., v. 11, n. 2, April 1975, pp 205-219.

[39] Ashbaugh, N., "On the Opening of a Finite Crack Normal to an Interface," J. All. Mech., ASME, v. 40, n. 2, June 1973, pp 626-628.

[40] Lin, K.Y., and J.W. Mar, "Finite Element ANalysis of Stress Intnesity Factors for Cracks at a BiMaterials Interface," Int. J. Fract., V. 12, n.4, Aug 1976.

[41] Flemming, J.F., J.R. Guydisk, J.R. Penta, C.E. Ronnion, "The Finite Element Method vs the Edge Function Method for Linear Fracture Analysis," Eng. Fract. Mech., V. 13, pp 42-55, 1980.

[42] Randall, P.N., Plain Strain Crack Toughness Testing of High Strength Metallic Materials, ASTM 410, ed. W.F. Brown, Jr., and J.E. Srawley, 88-126, 1966.

[43] Benchmark Editorial Committee of the SESA Fracture Committee, "A Critical Evaluation of Numerical Solutions to the 'Benchmark' Surface Flaw Problem," Experimental Mechanics, 253-64, Aug 1980.

[44] Tada, H., P. Paris, and G. Irwin, "The Stress Analysis of Cracks Handbook," Del Research Corporation, Hellectown, Pa., 1973.

[45] Parmerter, R. Reid, "Stress Intensity Factor for Three-Dimensional Problems," AFRPL-TR-76-30, Air Force Rocket Propulsion Laboratory, Edwards AFB, California, 1976.

[46] Benzley, S.E., and Z.E. Beisinger, "CHILES---A Finite Element Computer Program That Calculates the Intensities of Linear Elastic Singularities", Sandia Lab., Rep. No. SLA-73-0894, Albuquerque, New Mexico (September 1973).

[47] Henshall, R.D., and K.G. Shaw, "Crack Tip Finite Elements are Unnecessary", IJNME, Vol. 9, No. 3, 1975, pp. 495-507.

[48] Grandt, A.F., "Two Dimensional Stress Intensity Factor Solutions for Radially Cracked Rings," Technical Report, AFML-TR-75-121, Wright-Patterson AFB, Ohio, 1975.

[49] Warkington, W.E., "Development of a Computer Aided Analysis Package for Linear Elastic Fracture Mechanics," Masters Thesis, Brigham Young University Provo, Utah, 1983.

[50] Private Communication with J. J. Mecholsky.

[51] Parks, V.J., Sanford, R.J. "Photoelastic Stress and Fracture Analysis of Two Neutron Tube Designs," preliminary report to SLA.

APPENDIX A

CHILES.BYU USER'S GUIDE

CHILES.BYU

(CHILES . BRIGHAM YOUNG UNIVERSITY)

AN INTERACTIVE FINITE ELEMENT COMPUTER PROGRAM THAT CALCULATES THE INTENSITIES OF LINEAR ELASTIC SINGULARITIES IN ISOTROPIC OR ORTHOTROPIC MATERIALS AND ALONG BIMATERIAL INTERFACES

AUGUST 1982 EDITION

The computer program described in this document is available from Brigham Young University. Neither Brigham Young University nor their employees makes any warranty, expressed or implied, or assumes any legal responsibility for the accuracy, completeness or usefulness of this program and document.

ABSTRACT

CHILES.BYU is a finite element computer program that calculates the strength of singularities in linear elastic bodies. A generalized quadrilateral finite element that includes a singular point at a corner node is incorporated in the code. The displacement formulation is used and interelement compatibility is maintained so that monotone convergence is preserved. Plane stress, plane strain and axisymmetric conditions are treated. Isotropic and orthotropic crack tip singularity problems are solved by this version of the code but any type of singularity may be properly modeled by modifying selected subroutines in the program. This program also calculates the stress intensity factor of crack tip problems using a bimaterial crack option.

PROGRAM CAPABILITIES AND LIMITATIONS

1. CHILES performs a linear elastic stress analyses of any two-dimensional body in a plane stress, plane strain or axisymmetric state. Singular points are treated with enriched finite elements.
2. Up to three singular nodes may be defined in the body.
3. 1000 nodal points may be used.
4. 1000 elements may be used.
5. Bandwidth is limited to 54 (i.e., difference between node numbers in any one element must be < 27).
6. Mechanical and thermal loads are accepted.
7. A pre-created mesh and boundary condition scheme must be read from a file.
8. Displacements, stresses, and strains are output on the file CHILES.MOV for plotting.
9. CHILES.BYU automatically surrounds a singularity with type A and type B elements.
10. Small strains are assumed, a condition that is violated at the crack.
11. Up to 10 different materials can be defined.
12. Special elements are compatible with conventional elements.
13. Users may replace subroutines (CALQ and CALQI) to model singularities other than crack tips.

INTRODUCTION

This user's guide describes how to use the program CHILES.BYU. Each block of required input data is described in detail in the order requested by CHILES.BYU. In this way, the user can follow the guide in the same order that the data is requested. This guide is not intended to describe the theory of finite element analysis and the specific fracture mechanics elements in the program but is a step-by-step guide to allow someone inexperienced with CHILES.BYU to be able to use the program with ease. For a more thorough description of CHILES.BYU and its development the user should refer to CHILES 2 by Benzley and Beisinger; available through Sandia Labs. After each block of information is input a question asks if any changes are to be made. If changes are to be made input "yes" and re-input that block of information. If no changes are to be made input "no" and the program will continue on to the next block of information.

I. PROBLEM IDENTIFICATION INFORMATION

The first block that is required in the program is the initial problem identification data. This data identifies the scope of the problem to be analyzed. The specific input prompt is given followed by an explanation of the data requested by the prompt. All numerical values of this block must be entered in integer format (i.e. no decimal).

```
<ENTER TITLE OF PROBLEM>
```

This request requires a statement of identification for the problem. A maximum of 80 characters is allowed.

```
<ENTER NUMBER OF SINGULAR POINTS (3 MAX.)>
```

A value of zero to three is to be entered for this question. CHILES can handle up to three singular points.

```
<ENTER 1,2, OR 3 FOR AXISYMMETRIC, PLANE STRESS,
 OR PLANE STRAIN GEOMETRY>
```

The geometry selector question asks for either 1 to be entered for axisymmetric geometry, 2 to be entered for plane stress geometry, or 3 to be entered for plane strain geometry.

```
<ENTER NUMBER OF MATERIALS (10 MAX.)>
```

This asks for the number of materials which make up the problem being analyzed. Enter any number from one thru ten with ten being the maximum number of materials the program can handle.

```
<ENTER 0 FOR REDUCED OUTPUT OR 1 FOR
 EXTENDED OUTPUT>
```

This deals with the amount of information contained in the CHILES.LIS file. Reduced output contains stresses, strains, and displacements. Extended output contains loading information, connectivity, stresses, strains, and displacements.

```
<ENTER 0 FOR NO BOUNDARY CONDITIONS ON INTENS-
 ITIES OR 1 TO ALLOW INTENSITY BOUNDARY CONDIT-
 ION TO BE READ>
```

This gives the user the option of putting boundary conditions on intensities. Zero for none or one to allow.

```
<ENTER 0 TO READ QMESH FILE OR 1 TO INPUT
 GEOMETRY POINT BY POINT>
```

If a finite element mesh file has already been created input zero; otherwise input one and the geometry of the problem will input point by point.

```
<ENTER NUMBER OF BOUNDARY FLAGS AS SET
 IN QMESH>
```

This asks for the number of boundary flags which will be applied to the problem. Input the actual number that will be used.

```
<ENTER 0 FOR ISOTROPIC MATERIAL OR 1 FOR
 ORTHOTROPIC MATERIAL>
```

Isotropic material properties will be specified if 0 is input; orthotropic properties if 1.

II. MATERIAL PROPERTY INFORMATION (ISOTROPIC)

If there were no changes to be made for the first block of information the program continues to the material properties section of the program. For isotropic materials the program will ask three questions for each material. It asks these questions for one material then continues on to the next material in chronological order until all material properties are defined. All numerical values of this block must be entered in real format (i.e. a number including a decimal).

<ENTER YOUNGS MODULUS FOR MATERIAL (n)>

This entry specifies Young's modulus for the material number defined by the (n).

<ENTER POISSONS RATIO FOR MATERIAL (n)>

This entry specifies Poisson's ratio for the material number defined by the (n).

<ENTER COEF. OF THER. EXPANSION FOR MATERIAL (n)>

This entry specifies the coefficient of thermal expansion for the material number defined by the (n).

II. MATERIAL PROPERTY INFORMATION (ORTHOTROPOIC)

If orthotropic materials are used this block of material property questions will be asked. For each material seven entries will be required.

<ENTER E11 FOR MATERIAL (n)>

This entry specifies Young's modulus in the first principle direction of material orthotropy.

<ENTER V12 FOR MATERIAL (n)>

This entry specifies Poisson's ratio in the 1-2 plane.

<ENTER E22 FOR MATERIAL (n)>

This entry specifies Young's modulus in the second principle direction of material orthotropy.

<ENTER E33 FOR MATERIAL (n)>

This entry specifies Young's modulus normal to the plane of analysis.

<ENTER V31 FOR MATERIAL (n)>

This entry specifies Poisson's ratio in the 1-3 plane.

<ENTER V32 FOR MATERIAL (n)>

This entry specifies Poisson's ratio in the 2-3 plane.

<ENTER SHEAR MODULUS FOR MATERIAL (n)>

This entry specifies the shear modulus for the material in question.

<ENTER ANGLE FOR PRINCIPAL AXES>

This entry specifies the angle the principal axes of orthotropy make with respect to r-z coordinates.

III. FINITE ELEMENT MESH FILE INFORMATION

This block deals with the finite element mesh file that will be read by CHILES.BYU. Currently this is the data file written by the QMESH.BYU mesh generator. Any file name can be used . It is important to insure that the difference between node numbers in any one element be less than 27 for bandwidth considerations.

<SPECIFY QMESH FILE>

This asks to input the name of the finite element mesh file.

IV. BOUNDARY CONDITION INFORMATION

This block of information defines the different boundary conditions (i.e. displacements, forces, shear tractions, and normal tractions) that can be placed on sides or nodes of a mesh. The boundary flags that specify the location of the intended condition are defined in the QMESH file. A table describing the various codes is given below. All numerical values except the boundary flag number must be entered in real format.

CODE	Radial (r)	Axial (z)
0.0	Force	Force
1.0	Displ	Force
2.0	Force	Displ
3.0	Displ	Displ

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>

This entry defines the boundary flag number that has been defined in the QMESH file.

<ENTER BOUNDARY CODE>

This entry specifies the boundary code that will be applied to all nodes with a boundary flag as defined above. If the boundary code is greater than or equal to zero the boundary codes are defined in the table above and the next two entries are nodal forces or displacements. If the boundary code is less than zero the next two entries are the normal and shear tractions on an element face.

<ENTER XR OR PN VALUE OF DISPLACEMENT, FORCE OR NORMAL TRACTION>

Enter the proper value according to the boundary code used.

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR SHEAR TRACTION>

Enter the proper value according to the boundary code used.

V. DEFINITION OF A SINGULAR POINT

This block of information defines the singular points. Fracture mechanics analyses of single material interfaces and bimaterial interfaces use this option of CHILES.BYU. A maximum of three singular points can be defined. All numerical values except the angle phi must be entered in integer format. Phi must be entered as a real number.

DO YOU WISH TO DEFINE A SINGULAR POINT?

This asks if a singular point is to be defined. If not, input no.

<ENTER SINGULAR REGION>

Input the number which corresponds to the specific singular point, (i.e. 1,2, or 3).

<ENTER NODE NUMBER OF CRACK TIP>

The node number of the crack tip is entered here. This can be determined from a listing of the finite element mesh or displaying the mesh using a graphics program to determine what node number corresponds to the crack tip.

<ENTER ANGLE PHI OF CRACK>

Enter angle, in degrees, that the crack makes with the r-axis.

<ENTER REFERENCE MATERIAL FOR SINGULAR REGION>

Enter the number of the material that surrounds the singular region defined above.

<ENTER ADJACENT MATERIAL FOR SINGULAR REGION>

If a single material crack interface is being defined this entry is the same material number as the reference material entered above. If a bimaterial crack interface is being defined, this entry is the material on the opposite side of the interface from the reference material entered above. The stress intensity factors are normalized with respect to the reference material.

VI. DEFINITION OF BOUNDARY CODE FOR A NODE

This block of information allows the definition of a boundary code for a node. This option is useful if a boundary condition is needed to be specified for a node that does not have an associated boundary flag. Instead of using a boundary flag number as a reference the node number is used. All numerical values except the node number must be entered as a real number.

DO YOU WISH TO DEFINE A BOUNDARY CODE FOR A NODE?

If this is not needed input no.

<ENTER NODE NUMBER>

Enter the specific node number to be constrained. This number can be obtained from either a listing of the finite element mesh or from a graphics display.

<ENTER BOUNDARY CODE>

Enter this according to the chart in section IV.

<ENTER XR VALUE>

Enter this as described in section IV.

<ENTER XZ VALUE>

Enter this as described in section IV.

APPENDIX B

CHILES.BYU PROGRAM LISTING

```
      PROGRAM CHILES2V
        OPEN (UNIT=6, TYPE='NEW',ACCESS='SEQUENTIAL',NAME='CHILES.LIS')
        OPEN (UNIT=10,TYPE='NEW',ACCESS='SEQUENTIAL',NAME='CHILES.MOV',
     1        FORM='UNFORMATTED')
        OPEN (UNIT=1, TYPE='SCRATCH',ACCESS='SEQUENTIAL',
     1        FORM='UNFORMATTED')
        OPEN (UNIT=2, TYPE='SCRATCH',ACCESS='SEQUENTIAL',
     1        FORM='UNFORMATTED')
        OPEN (UNIT=4,TYPE='SCRATCH',ACCESS='SEQUENTIAL')
        OPEN (UNIT=12,TYPE='SCRATCH',ACCESS='SEQUENTIAL',
     1        FORM='UNFORMATTED')
      CALL CHILES2
        CLOSE (UNIT=1)
        CLOSE (UNIT=2)
        CLOSE (UNIT=4)
        CLOSE (UNIT=6)
        CLOSE (UNIT=9)
        CLOSE (UNIT=10)
        CLOSE (UNIT=12)
        CLOSE (UNIT=13)
      END
      SUBROUTINE CHILES2
C
C
C                               CHILES 2V
C
C                             S. E. BENZLEY
C                            W. E. WARKENTIN
C                      CIVIL ENGINEERING DEPARTMENT
C               BRIGHAM YOUNG UNIVERSITY, PROVO, UTAH, 84602
C                    CHILES RELEASED SEPTEMBER 1973
C                     CHILES2 RELEASED AUGUST 1977
C                    CHILES 2V RELEASED AUGUST 1982
C        CHILES IS A FINITE ELEMENT COMPUTER PROGRAM THAT CALCULATES
C        THE STRENGTH OF SINGULARITIES IN LINEAR ELASTIC BODIES.  A
C        GENERALIZED QUADRILATERAL FINITE ELEMENT THAT INCLUDES A
C        SINGULAR POINT AT A CORNER NODE IS INCORPORATED IN THE CODE.
C        THE DISPLACEMENT FORMULATION IS USED AND INTERELEMENT
C        COMPATIBILITY IS MAINTAINED SO THAT MONOTONE CONVERGENCE IS
C        PRESERVED.  PLANE STRESS, PLANE STRAIN AND AXISYMMETRIC
C        CONDITIONS ARE TREATED.  ISOTROPIC AND ORTHOTROPIC CRACK TIP
C        SINGULARITY PROBLEMS ARE SOLVED BY THIS VERSION OF THE CODE AS
C        WELL AS A SPECIAL APPROXIMATE TREATMENT OF A BIMATERIAL INTER-
C        FACE CRACK.  ANY TYPE OF SINGULARITY MAY BE PROPERLY MODELED
C        BY MODIFYING SELECTED SUBROUTINES IN THE PROGRAM.
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *         CHILES WAS ISSUED BY SANDIA LABORATORIES,               *
C     *                 A PRIME CONTRACTOR TO THE                       *
C     *           UNITED STATES ATOMIC ENERGY COMMISSION                *
C     * * * * * * * * * * * * *  NOTICE    * * * * * * * * * * * * * * * *
C     THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE
C     UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE
C     UNITED STATES ATOMIC ENERGY COMMISSION, NOR ANY OF THEIR
C     EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR
C     EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY
C     LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS
C     OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT OR PROCESS
C     DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE
C     PRIVATELY OWNED RIGHTS.
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *THE BASIC REFERENCE DOCUMENT FOR THIS CODE IS SLA-73-0894,       *
C     *SEPTEMBER 1973.                                                  *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
C          TAPE1---BINARY TAPE FOR REDUCED BLOCKS OF EQUATIONS IN SOLV.
C          TAPE2---BINARY TAPE FOR UNREDUCED BLOCKS OF EQUATIONS IN
C                  STIFF FOR USE IN SOLV.
C          TAPE4---BCD TAPE USED TO OUTPUT CARD IMAGES.
C          TAPE5---STANDARD INPUT TAPE.
C          TAPE6---STANDARD OUTPUT TAPE.
C          TAPE9---BINARY INPUT TAPE--WRITTEN BY A MESH GENERATOR.
C          TAPE10--BINARY DATA TAPE TO BE USED FOR PLOTTING.
C          TAPE12--BINARY TAPE STORES STRESS AND STRAIN MATRICES IN
C                  ELSTIF FOR LATER CALCULATIONS.
C          THE MAIN PROGRAM CALLS SUBROUTINES ZONE, MESHC, TYPE, STIFF,
C          AND SOLV.
C
C
      COMMON          IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1                XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2                PR(200),IS(200),JS(200),SH(200)
      COMMON /ROOT/E1S(10),E2S(10),XNU1S(10),XNU2S(10),G2S(10),ANGS(10),
     1                E3S(10),XNU31S(10),XNU32S(10)
C     COMMON /MAT/ D(4,4,10),HED(8)
      COMMON /MAT/ D(4,4,10),HED(20)
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1                NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /EL1/ XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /GLB/ XF(108),XBM(108,54),XC(108,6),XCT(6,6),XBT(6)
      COMMON /ORTO/ KORTSW
      COMMON /HEAT/ TSTR(1000),ALPHA(10)
      DIMENSION       SIG(4,1000),STRN(4,1000),U(10),ES(4),EST(4),
     1                STRAIN(4,10),YM(10),NSINN(3),CARD(20)
C    1                STRAIN(4,10),YM(10),NSINN(3),CARD(9)
      DIMENSION       T(4,4),TD(4,4)
      DIMENSION       RMI(1000),ZMI(1000)
      EQUIVALENCE     (R(1),SIG(1)),(XZ(1),STRN(1))
      INTEGER         BETA
      DATA            ENDDAT/4HEND /
C
      TIM1=SECNDS(0.0)
      REWIND 12
C      CALL FTNBIN (1,0,IIDUM)
C
C          ECHO OF INPUT CARDS
C
C
C          HOROLOG IS A LIBRARY SUBROUTINE AVAILABLE ONLY AT THE
C          SANDIA LABORATORIES CDC 6600 INSTALLATION.
C
C     CALL HOROLOG (IIDUM,IIDUM,IDATE)
      CALL DATE(IDATE)
      WRITE (6,580) IDATE
C
      TYPE 14
   14 FORMAT(//////////,28X,' <<WELCOME TO CHILES.BYU>>'//)
      TYPE 21
   21 FORMAT(35X,'AN INTERACTIVE')
      TYPE 31
   31 FORMAT(33X,'TWO-DIMENSIONAL OR')
      TYPE 41
   41 FORMAT(32X,'AXISYMMETRIC FINITE')
      TYPE 51
   51 FORMAT(34X,'ELEMENT PROGRAM'/////)
      DO 20 I=1,1000
      R(I)=0.0
      Z(I)=0.0
      IX(1,I)=0
      IX(2,I)=0
```

```
      IX(2,I)=0
      IX(3,I)=0
      IX(4,I)=0
      IX(5,I)=0
      CODE(I)=0.0
      XR(I)=0.0
      XZ(I)=0.0
      ISP(I)=0
      BETA(I)=6
   20 CONTINUE
      DO 30 I=1,200
      IP(I)=0
      JP(I)=0
      PR(I)=0.0
      IS(I)=0
      JS(I)=0
      SH(I)=0.0
   30 CONTINUE
      DO 40 I=1,4
      DO 40 J=1,10
      D(I,1,J)=0.0
      D(I,2,J)=0.0
      D(I,3,J)=0.0
      D(I,4,J)=0.0
   40 CONTINUE
      DO 50 I=1,6
      ISMAT(I)=0
      XBT(I)=0.0
      DO 50 J=1,6
   50 XCT(J,I)=0.0
      KODE(1)=0
      KODE(2)=0
      KODE(3)=0
C
C         BEGINNING OF DATA INPUT/OUTPUT
   52 CONTINUE
      TYPE 920
      ACCEPT 620,HED
      WRITE (6,630) HED
      TYPE 931
      ACCEPT 941,NSP
      TYPE 951
      ACCEPT 941,NPAR
      TYPE 961
      ACCEPT 941,NMAT
      TYPE 971
      ACCEPT 941,IPTSW
      TYPE 980
      ACCEPT 941,KBSW
      TYPE 990
      ACCEPT 941,KGEOSW
      TYPE 1000
      ACCEPT 941,NUMTB
      TYPE 1100
      ACCEPT 941,KORTSW
      TYPE 54
   54 FORMAT(//,' DO YOU WISH TO CHANGE ANY OF THE ABOVE ENTRIES? ' $)
      ACCEPT 56, ANS
   56 FORMAT (A1)
      IF(ANS.EQ.'Y') GO TO 52
      WRITE (6,650) NSP,NPAR,NMAT,NMESHC,IPTSW,KBSW,KGEOSW
      WRITE(6,660) NUMTB,KORTSW,NSSURF
      NST=NSP+NSP
C
      IF (KORTSW.GT.0) GO TO 110
C         SET MATERIAL CONSTANTS FOR D MATRIX
   58 CONTINUE
```

```
   58 CONTINUE
      DO 100 I=1,NMAT
      TYPE 1200, I
      ACCEPT 945,E
      TYPE 1300, I
      ACCEPT 945,XNU
      TYPE 1400, I
      ACCEPT 945,ALPHA(I)
      YM(I)=E
      XNUS(I)=XNU
      CON=(E*(1.0-XNU))/((1.0+XNU)*(1.0-2.0*XNU))
      C2=(CON*XNU)/(1.0-XNU)
C
C         NPAR = 1  FOR AXISYMMETRIC SOLUTION
C         NPAR = 2  FOR PLANE STRESS SOLUTION
C         NPAR = 3  FOR PLANE STRAIN SOLUTION
C
      GO TO (60,70,80), NPAR
   60 D(2,2,I)=CON
      D(1,2,I)=C2
      D(2,1,I)=C2
      D(2,3,I)=C2
      D(3,2,I)=C2
      D(4,4,I)=CON*(1.-2.*XNU)/(2.*(1.-XNU))
      GO TO 90
   70 CON=E/(1.-XNU*XNU)
      C2=CON*XNU
      D(4,4,I)=CON*(1.-XNU)/2.
      GO TO 90
   80 D(4,4,I)=E/(2.0*(1.0+XNU))
   90 CONTINUE
      D(1,1,I)=CON
      D(3,3,I)=CON
      D(1,3,I)=C2
      D(3,1,I)=C2
  100 CONTINUE
      TYPE 105
  105 FORMAT(//,' DO YOU WISH TO CHANGE ANY MATERIAL PROPERTY ENTRIES? '
     1 $)
      ACCEPT 107, ANS
  107 FORMAT (A1)
      IF(ANS.EQ.'Y') GO TO 58
      GO TO 230
C
  110 CONTINUE
      DO 220 I=1,NMAT
      TYPE 1500, I
      ACCEPT 945,E1
      TYPE 1600, I
      ACCEPT 945,XNU1
      TYPE 1700, I
      ACCEPT 945,E2
      TYPE 1800, I
      ACCEPT 945,E3
      TYPE 1900, I
      ACCEPT 945,XNU31
      TYPE 2000, I
      ACCEPT 945,XNU32
      TYPE 2100, I
      ACCEPT 945,G2
      TYPE 2200, I
      ACCEPT 945,ANG
      XNU2=XNU1*E2/E1
      E3S(I)=E3
      XNU31S(I)=XNU31
      XNU32S(I)=XNU32
      E1S(I)=E1
```

```
      E1S(I)=E1
      E2S(I)=E2
      XNU1S(I)=XNU1
      XNU2S(I)=XNU2
      G2S(I)=G2
      ANGS(I)=ANG
      XN=E1/E2
      XM=G2/E2
      CON=E2/((1.0+XNU1)*(1.0-XNU1-2.0*XN*XNU2*XNU2))
      DO 120 N=1,4
      DO 120 J=1,4
      D(N,J,I)=0.0
  120 CONTINUE
C
C          TRANSFORM IF ORTHOTROPIC SKEWED TO R-Z COORDINATES
C
      GO TO (130,140,150), NPAR
  130 CONTINUE
      D(1,1,I)=CON*XN*(1.0-XN*XNU2*XNU2)
      D(2,2,I)=D(1,1,I)
      D(1,2,I)=CON*XN*(XNU1+XN*XNU2*XNU2)
      D(2,1,I)=D(1,2,I)
      D(1,3,I)=CON*XN*XNU2*(1.0+XNU1)
      D(3,1,I)=D(1,3,I)
      D(2,3,I)=D(1,3,I)
      D(3,2,I)=D(1,3,I)
      D(3,3,I)=CON*(1.0-XNU1*XNU1)
      D(4,4,I)=CON*XM*(1.0+XNU1)*(1.0-XNU1-2.0*XN*XNU2*XNU2)
      GO TO 160
  140 CONTINUE
      CON=E2/(1.0-XN*XNU2*XNU2)
      D(1,1,I)=CON*XN
      D(1,3,I)=CON*XN*XNU2
      D(3,1,I)=D(1,3,I)
      D(3,3,I)=CON
      D(4,4,I)=CON*XM*(1.0-XN*XNU2*XNU2)
      GO TO 160
  150 CONTINUE
      D(1,1,I)=CON*XN*(1.0-XN*XNU2*XNU2)
      D(1,3,I)=CON*XN*XNU2*(1.0+XNU1)
      D(3,1,I)=D(1,3,I)
      D(3,3,I)=CON*(1.0-XNU1*XNU1)
      D(4,4,I)=CON*XM*(1.0+XNU1)*(1.0-XNU1-2.0*XN*XNU2*XNU2)
  160 CONTINUE
      IF (ANG.EQ.0.0) GO TO 220
      DO 170 J=1,4
      T(2,J)=0.0
      T(J,2)=0.0
  170 CONTINUE
      T(1,1)=COS(ANG)*COS(ANG)
      T(1,3)=SIN(ANG)*SIN(ANG)
      T(3,1)=T(1,3)
      T(2,2)=1.0
      T(4,1)=SIN(ANG)*COS(ANG)
      T(1,4)=-2.0*T(4,1)
      T(3,3)=T(1,1)
      T(3,4)=-T(1,4)
      T(4,3)=-T(4,1)
      T(4,4)=T(1,1)-T(1,3)
C          TD=T*D
      DO 190 NR=1,4
      DO 190 NC=1,4
      Q1=0.0
      DO 180 J=1,4
      Q1=Q1+T(NR,J)*D(J,NC,I)
  180 CONTINUE
  190 TD(NR,NC)=Q1
```

```
  190 TD(NR,NC)=Q1
C         D=TD*TT
      DO 210 NR=1,4
      DO 210 NC=1,4
      Q1=0.0
      DO 200 J=1,4
  200 Q1=Q1+TD(NR,J)*T(NC,J)
  210 D(NR,NC,I)=Q1
  220 CONTINUE
      TYPE 222
  222 FORMAT(//,' DO YOU WISH TO CHANGE ANY MATERIAL PROPERTY ENTRIES? '
     1 $)
      ACCEPT 224, ANS
  224 FORMAT (A1)
      IF(ANS.EQ.'Y') GO TO 110
  230 CONTINUE
C
C         TEST (KGEOSW) SWITCH TO DETERMINE IF MESHING INFORMATION
C         IS INPUT FROM TAPE9 OR FROM CARDS
      IF (KGEOSW.NE.0) CALL ZONE
      IF (KGEOSW.EQ.0) CALL QMESH (NUMTB)
      WRITE (6,690) NEL,NODES,NFORCE,NUMSC
C         OUTPUT PRESSURE AND SHEAR LOADING INFO
      IF (NFORCE.EQ.0) GO TO 270
      WRITE (6,240)
  240 FORMAT (//9X,1HI,9X,1HJ,5X,8HPRESSURE,5X,5HSHEAR/)
      DO 260 I=1,NFORCE
      WRITE (6,250) IP(I),JP(I),PR(I),SH(I)
  250 FORMAT (2I10,2E15.5)
  260 CONTINUE
  270 CONTINUE
C
C         OUTPUT MATERIAL PROPERTIES
C
      IF (KORTSW.GT.0) GO TO 280
      WRITE (6,700)
      WRITE(6,730) (I,YM(I),XNUS(I),ALPHA(I),I=1,NMAT)
      GO TO 290
  280 CONTINUE
      WRITE (6,710)
      WRITE (6,720) (I,E1S(I),XNU1S(I),E2S(I),E3S(I),XNU31S(I),XNU32S(I)
     1,G2S(I),ANGS(I),I=1,NMAT)
  290 CONTINUE
      IF (NPAR-2) 300,310,320
  300 WRITE (6,740)
      GO TO 330
  310 WRITE (6,750)
      GO TO 330
  320 WRITE (6,760)
  330 CONTINUE
C
C         TEST (NMESHC) SWITCH TO DETERMINE IF MESH CHANGES ARE NEEDED
C
      IF (NST.EQ.0) GO TO 350
C
C         INPUT/OUTPUT SINGULAR POINTS CARD
C
      WRITE (6,770)
C
C        INTERACTIVE INPUT OF SINGULAR POINT INFORMATION
C
 5001 CONTINUE
      TYPE 5000
 5000 FORMAT(//,' DO YOU WISH TO DEFINE A SINGULAR POINT? ' $)
      ACCEPT 5010, ANS
 5010 FORMAT (A1)
      IF(ANS.EQ.'N') GO TO 6000
```

```
      IF(ANS.EQ.'N') GO TO 6000
      TYPE 5020
 5020 FORMAT(//,' <ENTER SINGULAR REGION> ' $)
      ACCEPT 5030,I
 5030 FORMAT(I)
      TYPE 5040
 5040 FORMAT(//,' <ENTER NODE NUMBER OF CRACK TIP> ' $)
      ACCEPT 5030, NODEN
      NSINN(I)=NODEN
      TYPE 5050
 5050 FORMAT(//,' <ENTER ANGLE PHI OF CRACK> ' $)
      ACCEPT 5060, ANGPHI
 5060 FORMAT(E)
      PHI(I)=ANGPHI
      TYPE 5070
 5070 FORMAT(//,' <ENTER REFERENCE MATERIAL FOR SINGULAR REGION> ' $)
      ACCEPT 5030, MATREF
      I2=2*I
      ISMAT(I2)=MATREF
      TYPE 5080
 5080 FORMAT(//,' <ENTER ADJACENT MATERIAL FOR SINGULAR REGION> ' $)
      ACCEPT 5030, MATADJ
      I1=I2-1
      ISMAT(I1)=MATADJ
      GO TO 5001
 6000 CONTINUE
      TYPE 583
  583 FORMAT(//,' DO YOU WISH TO MAKE ANY CHANGES? ' $)
      ACCEPT 584, ANS
  584 FORMAT (A1)
      IF (ANS.EQ.'Y') GO TO 5001
      DO 340 I=1,NSP
      I1=NSINN(I)
      CALL TYPE (I,I1)
      RC(I)=R(I1)
      ZC(I)=Z(I1)
      I2=I+I
      WRITE (6,790) I,I1,RC(I),ZC(I),PHI(I),ISMAT(I2)
      PHI(I)=PHI(I)*0.01745329251994
  340 CONTINUE
  350 CONTINUE
      IF (KBSW.EQ.0) GO TO 360
C
C         INPUT/OUTPUT BOUNDARY CODE CARD FOR SINGULAR POINTS
C
      DO 355 I11=1,NSP
      TYPE 2500,I11
      ACCEPT 945,KODE(I11)
  355 CONTINUE
      WRITE (6,810) (I,KODE(I),I=1,NSP)
  360 CONTINUE
C
C       READ PRESSURE CARDS ON SINGULAR SURFACES AND COMPUTE PSEUDO LOADS
C
C
C       EXACT INTEGRATION FOR PLANE PROBLEMS, APPROXIMATE INTEGRATION FOR
C       AXISYMMETRIC PROBLEMS
C
      RMEAN=1.
      IF(NSSURF.EQ.0) GO TO 3612
      DO 3651 I=1,NSSURF
      READ(4,960)   IMM,I1,I2,I3,ISPN,XPRES
      WRITE(6,970)I,IMM,I1,I2,I3,ISPN,XPRES
960   FORMAT(5I5,E10.3)
970   FORMAT(1H ,' SINGULAR SURFACE NUMBER = ',I5/
     1             ' MATERIAL NUMBER         = ',I5/
     2             ' CRACK TIP NODAL POINT   = ',I5/
```

```
     2                ' CRACK TIP NODAL POINT  = ',I5/
     3                ' FIRST NODAL POINT      = ',I5/
     4                ' SECOND NODAL POINT     = ',I5/
     5                ' SINGULAR POINT NUMBER = ',I5/
     6                ' APPLIED PRESSURE       = ',E15.4///)
      IF(NPAR.EQ.1) RMEAN=.5*(R(I1)+R(I2))
      RHO1=SQRT((R(I2)-R(I1))**2+(Z(I2)-Z(I1))**2)
      RHO2=SQRT((R(I3)-R(I1))**2+(Z(I3)-Z(I1))**2)
      XL=RHO2-RHO1
      XNU=XNUS(IMN)
      XKAP=3.-4.*XNU
      IF(NPAR.EQ.2) XKAP=(3.-XNU)/(1.+XNU)
      INDEX=2*ISPN-1
      XBT(INDEX)=XBT(INDEX)+RMEAN*(RHO1**1.5)*(XKAP+1.)/12.
      R1P=RHO2
      R2P=RHO1
      IF(NPAR.EQ.1) RMEAN=.5*(R(I2)+R(I3))
      XTEM=0.
      XTEM=XTEM+(R1P**1.5-R2P**1.5)*4./3.
      XTEM=XTEM-(R1P**2.5)*4./(5.*XL)
      XTEM=XTEM-(R2P**2.5)*8./(15.*XL)
      XTEM=XTEM+(R2P*R1P**1.5)*4./(3.*XL)
      XTEM=XTEM-XL*R1P**.5/3.-XL*R2P**.5*2./3.
      XTEM=XTEM*(XKAP+1.)*.5*RMEAN
      XBT(INDEX)=XBT(INDEX)+XTEM
      XBT(INDEX)=XPRES*XBT(INDEX)
3651  CONTINUE
 3612 CONTINUE
      DO 361 I=1,1000
361   TSTR(I)=0.0
      IF(NMESHC.EQ.0) GO TO 365
      IF(NMESHC.LT.0) GO TO 363
      DO 362 I=1,NODES
      READ(4,940) NODT,TEMP
362   TSTR(NODT)=TEMP
      GO TO 364
363   CONTINUE
C
C     READ NODAL POINT TEMPERATURE FROM TAPE
C
      DO 773 I= 1,NODES
773   TSTR(I)=-100.
364   CONTINUE
      WRITE(6,930)
      DO 3641 I=1,NODES
3641  WRITE(6,950) I,TSTR(I)
365   CONTINUE
930   FORMAT(1H0,4X,'NODE',10X,'TEMPERATURE DIFFERENTIAL'//)
940   FORMAT(I10,E10.0)
950   FORMAT(I10,E30.5)
C
C      INTERACTIVE INPUT OF POINT BOUNDARY CODES
C
 3001 CONTINUE
      TYPE 3000
 3000 FORMAT(//,' DO YOU WISH TO DEFINE A BOUNDARY CODE FOR A NODE ? ' $
     1)
      ACCEPT 3010, ANS
 3010 FORMAT(A1)
      IF(ANS.EQ.'Y') GO TO 3003
      GO TO 4000
 3003 TYPE 3020
 3020 FORMAT(//,' <ENTER NODE NUMBER> ' $)
      ACCEPT 3030, NUMBER
 3030 FORMAT(I)
      TYPE 3040
 3040 FORMAT(//,' <ENTER BOUNDARY CODE> ' $)
```

```
 3040 FORMAT(//,' <ENTER BOUNDARY CODE> ',$)
      ACCEPT 3050, BCODE
 3050 FORMAT (E)
      CODE(NUMBER)=BCODE
      GO TO 3001
 4000 CONTINUE
      TYPE 666
  666 FORMAT(////,28X' <<CHILES IS EXECUTING>>'////)
C         TEST (IPTSW) SWITCH TO DECIDE OUTPUT--IF IPTSW"0 ELEMENT DATA
C         INCLUDING SINGULAR REGION AND R-Z COORDINATES WILL BE OUTPUT
C
      IF (IPTSW.EQ.0) GO TO 410
      MPRINT=0
      DO 380 I=1,NEL
      IF (MPRINT.NE.0) GO TO 370
      WRITE (6,820) HED
      MPRINT=50
  370 MPRINT=MPRINT-1
      WRITE (6,830) I,(IX(J,I),J=1,5)
      IF (ISP(I).NE.0) WRITE (6,840) BETA(I),ISP(I)
  380 CONTINUE
      MPRINT=0
      DO 400 I=1,NODES
      IF (MPRINT.NE.0) GO TO 390
      WRITE (6,850) HED
      MPRINT=50
  390 MPRINT=MPRINT-1
      WRITE (6,860) I,CODE(I),R(I),Z(I),XR(I),XZ(I)
  400 CONTINUE
  410 CONTINUE
C
C         END OF DATA INPUT/OUTPUT
C
C         CALCULATE BANDWIDTH
C
      MBAND=0
      DO 420 I=1,NEL
      NNMAX=MAX0(IX(1,I),IX(2,I),IX(3,I),IX(4,I))
      NNMIN=MIN0(IX(1,I),IX(2,I),IX(3,I),IX(4,I))
      NBW1=(NNMAX-NNMIN+1)*2
      IF (NBW1.GT.MBAND) MBAND=NBW1
  420 CONTINUE
      WRITE (6,870) MBAND
      IF (MBAND.LE.54) GO TO 430
      WRITE (6,880)
      STOP
  430 CONTINUE
C
C         CALL ROUTINE TO FORM STIFFNESS MATRIX
C
      CALL STIFF (NMESHC)
C
C         CALL ROUTINE TO SOLVE BANDED STIFFNESS MATRIX AND
C         OUTPUT DISPLACEMENT SOLUTION AND SINGULAR INTENSITIES
C
      CALL SOLV
C
C         OUTPUT GEOMETRY DATA AND DISPLACEMENTS ON BINARY DATA TAPE10
C
      REWIND 10
      WRITE (10) HED,NEL,NODES
      WRITE (10) (R(I),I=1,NODES),(Z(I),I=1,NODES),((IX(I,J),J=1,NEL),I=
     11,5)
      TDUM=0.0
      WRITE (10) TDUM
      NDOF=2*NODES
      WRITE (10) (XR(I),I=1,NDOF),(XR(I),I=1,NDOF),(XR(I),I=1,NDOF)
```

```
      WRITE (10) (XF(I),I=1,NDOF),(XF(I),I=1,NDOF),(XF(I),I=1,NDOF)
C
C        CALCULATE AND OUTPUT STRESSES AND STRAINS FOR EACH ELEMENT
C
      MPRINT=0
      REWIND 12
      DO 570 I=1,NEL
      READ (12) NRN,RM,ZM,IMAT,IEL
      RMI(IEL)=RM
      ZMI(IEL)=ZM
      READ (12) ST
      READ (12) STRAIN
      DO 440 K=1,8
      IND=NRN(K)
  440 U(K)=XF(IND)
      IF (NST.EQ.0) GO TO 460
      DO 450 J=1,2
      K=J+8
      IND=NRN(K)
  450 U(K)=XBT(IND)
  460 CONTINUE
      DO 470 J=1,4
      ES(J)=0.0
      EST(J)=0.0
      KT=10
      IF (NST.EQ.0) KT=8
      DO 470 K=1,KT
      EST(J)=EST(J)+STRAIN(J,K)*U(K)
  470 ES(J)=ES(J)+ST(J,K)*U(K)
C
C    CALCULATE STRESS AND STRAINS
C
      I1=IX(1,IEL)
      I2=IX(2,IEL)
      I3=IX(3,IEL)
      I4=IX(4,IEL)
      ET1=TSTR(I1)*ALPHA(IMAT)
      ET2=TSTR(I2)*ALPHA(IMAT)
      ET3=TSTR(I3)*ALPHA(IMAT)
      ET4=TSTR(I4)*ALPHA(IMAT)
      ET=.25*(ET1+ET2+ET3+ET4)
      ES(1)=ES(1)-(D(1,1,IMAT)+D(1,2,IMAT)+D(1,3,IMAT))*ET
      ES(2)=ES(2)-(D(2,1,IMAT)+D(2,2,IMAT)+D(2,3,IMAT))*ET
      ES(3)=ES(3)-(D(3,1,IMAT)+D(3,2,IMAT)+D(3,3,IMAT))*ET
      IF (NPAR-2) 540,510,480
C
C        CALCULATE T-STRESS FOR PLANE STRAIN SOLUTION
C
  480 CONTINUE
      IF (KORTSW.GT.0) GO TO 490
      ES(2)=(ES(1)+ES(3))*XNUS(IMAT)-YM(IMAT)*ET
      GO TO 500
  490 ES(2)=ES(1)*XNU32S(IMAT)+ES(3)*XNU32S(IMAT)
  500 CONTINUE
      GO TO 540
C
C        CALCULATE T-STRAIN FOR PLANE STRESS SOLUTION
C
  510 CONTINUE
      IF (KORTSW.GT.0) GO TO 520
      EST(2)=-(ES(1)+ES(3))*XNUS(IMAT)/YM(IMAT)+ET
      GO TO 530
  520 EST(2)=-ES(1)*XNU31S(IMAT)/E1S(IMAT)-ES(3)*XNU32S(IMAT)/E2S(IMAT)
  530 CONTINUE
  540 CONTINUE
      DO 550 J=1,4
```

```
     IF (J.EQ.2) J1=3
     IF (J.EQ.3) J1=2
     STRN(J,IEL)=EST(J1)
     SIG(J,IEL)=ES(J1)
550  CONTINUE
 570 CONTINUE
     MPRINT=0
     DO 575 I=1,NEL
     IF (MPRINT.NE.0) GO TO 560
     WRITE (6,890) HED
     MPRINT=50
 560 MPRINT=MPRINT-1
     WRITE(6,900) I,RMI(I),ZMI(I),(SIG(K,I),K=1,4)
575  CONTINUE
     MPRINT=0
     DO 577 I=1,NEL
     IF (MPRINT.NE.0) GO TO 578
```

```
      WRITE (6,891) HED
      MPRINT=50
  578 MPRINT=MPRINT-1
      WRITE(6,900) I,RMI(I),ZMI(I),(STRN(K,I),K=1,4)
  577 CONTINUE
C        OUTPUT STRESSES AND STRAINS ON BINARY DATA TAPE
C
      WRITE (10) ((SIG(I,J),J=1,NEL),I=1,4)
      WRITE (10) ((STRN(I,J),J=1,NEL),I=1,4)
      REWIND 10
C
C     CALL SECOND(TIM2)
      TIM2=SECNDS(0.0)
      TIM=TIM2-TIM1
      WRITE (6,910) TIM
      IF(NMESHC.EQ.0.OR.KORTSW.EQ.0) WRITE(6,917)
      TYPE 2655
 2655 FORMAT(25X,'EXECUTION COMPLETED;'/)
      TYPE 2656
 2656 FORMAT(30X,'YOUR PRINTED OUTPUT FILE IS CHILES.LIS')
      TYPE 2657
 2657 FORMAT(30X,'YOUR PLOT DATA FILE IS CHILES.MOV'///)
      TYPE 2658
 2658 FORMAT(25X,'SEE YOU AGAIN NEXT TIME'///)
      TYPE 2659
 2659 FORMAT(25X,'YOUR USER FRIENDLY FINITE ELEMENT PROGRAM'///)
      TYPE 2660
 2660 FORMAT(30X,'SIGNED,'/)
      TYPE 2661
 2661 FORMAT(33X,'CHILES.')
      RETURN
  917 FORMAT(1H ,'   PROGRAM STOPPED, ONLY ISOTROPIC THERMAL EXPANSION
     1ALLOWED  ')
C
  580 FORMAT (100H1 CHILES2--A FINITE ELEMENT PROGRAM THAT CALCULATES TH
     1E INTENSITIES OF LINEAR ELASTIC SINGULARITIES.,//10X,24HRELEASED S
     2EPTEMBER 1973.,/10X,19HREVISED AUGUST 1977,//10X,23HTHIS PROBLEM W
     3AS RUN ON,4X,A10//)
  590 FORMAT (1H1,35X,18HECHO OF INPUT DATA//9X,2H10,8X,2H20,8X,2H30,8X,
     12H40,8X,2H50,8X,2H60,8X,2H70,8X,2H80/81H 12345678901234567890123456
     267890123456789012345678901234567890123456789012345678901234567890/)
C 600 FORMAT (A3,A7,7A10)
  600 FORMAT (20A4)
C 610 FORMAT (1H ,A3,A7,7A10)
  610 FORMAT (1H ,20A4)
C 620 FORMAT (8A10)
  620 FORMAT (20A4)
  630 FORMAT (11X,20A4/)
  640 FORMAT (10I5)
  650 FORMAT (4X,54HNUMBER OF SINGULAR POINTS (NSP)---------------------
     1--,I5/,58H    GEOMETRY PARAMETER (NPAR)---------------------------
     2--,I5/,58H    NUMBER OF MATERIALS (NMAT)--------------------------
     3--,I5/4X,54HTHERMAL LOADING INPUT SWITCH (NMESHC)------------------
     4,I5/4X,54HGEOMETRY OUTPUT OPTION (IPTSW)------------------------,I
     55/4X,54HSTRESS INTENSITY BOUNDARY CONDITIONS (KBSW)-----------,I5/
     64X,54HMESH GENERATION SWITCH (KGEOSW)-----------------------,I5)
  660 FORMAT (4X,54HB.C. TABLE CARDS (NUMTB)----------------------------
     1--,I5/,4X,54HORTHOTROPIC MATERIAL SWITCH (KORTSW)-----------------
     2-,I5/  4X,54HNUMBER OF LOADED SINGULAR SURFACES (NSSURF)----------
     2-,I5)
  670 FORMAT (2E10.0)
  680 FORMAT (8E10.0)
  690 FORMAT (4X,54HNUMBER OF ELEMENTS (NEL)----------------------------
     1-- I5/4X 54HNUMBER OF NODAL POINTS (NODES)------------------------
```

```
     1--,I5/4X,54HNUMBER OF NODAL POINTS (NODES)------------------------,I
     2,I5/4X,54HNUMBER OF PRESSURE CARDS (NFORCE)---------------------,I
     35/4X,54HNUMBER OF SHEAR CARDS (NUMSC)------------------------,I5)
  700 FORMAT (//20X,19HMATERIAL PROPERTIES//4X,12HMATERIAL NO.,5X,14HYOU
     1NGS MODULUS,5X,14HPOISSONS RATIO,3X,24HCOEFFICIENT OF EXPANSION)
  710 FORMAT (//5X,3HMAT,8X,2HE1,12X,4HNU12,12X,2HE2,13X,2HE3,12X,4HNU31
     1,11X,4HNU32,12X,2HG2,11X,3HANG/)
  720 FORMAT (I7,8E15.5)
  730 FORMAT(I10,E23.5,F18.5,E26.5)
  740 FORMAT (//4X,54HTHIS PROBLEM SOLVES EQUATIONS FOR AXISYMETRIC GEOM
     1ETRY)
  750 FORMAT (//4X,55HTHIS PROBLEM SOLVES EQUATIONS FOR PLANE STRESS GEO
     1METRY)
  760 FORMAT (//4X,55HTHIS PROBLEM SOLVES EQUATIONS FOR PLANE STRAIN GEO
     1METRY)
  770 FORMAT (///4X,12HSINGULAR PT.,1X,4HNODE,4X,10HR-ORDINATE,5X,10HZ-O
     1RDINATE,4X,11HCRACK ANGLE,2X,8HMATERIAL/)
  780 FORMAT (3(I5,E10.0))
  790 FORMAT (2I10,2E15.5,F13.2,I9)
  800 FORMAT (3I5)
  810 FORMAT (///4X,36HSTRESS INTENSITY BOUNDARY CONDITIONS//9X,6HREGION
     1,15X,4HKODE//(I13,I19))
  820 FORMAT (1H1,5X,20A4///,10X,29HTABLE OF ELEMENT CONNECTIVITY,//,6X,
     17HELEMENT,5X,1HI,5X,1HJ,5X,1HK,5X,1HL,3X,8HMATERIAL,2X,'ELEMENT',
     21X,'TYPE',2X,15HSINGULAR REGION/)
  830 FORMAT (1I13,4I6,I7)
  840 FORMAT (1H+,46X,I11,I14)
  850 FORMAT (1H1,5X,20A4///12H NODAL POINT,2X,4HCODE,2X,'R-ORDINATE',5X
     1,'Z-ORDINATE  R LOAD OR DISPLACEMENT  Z LOAD OR DISPLACEMENT')
  860 FORMAT (6X,I6,F6.0,2F12.3,E20.5,E24.5)
  870 FORMAT (//4X,12HBANDWIDTH IS,I5)
  880 FORMAT (42H0***FATAL ERROR    MAXIMUM BANDWITH IS 54.)
  890 FORMAT (1H1,25X,20A4/003X,2HEL,6X,1HR,8X,1HZ,6X,8HR-STRESS,5X,8HZ-
     1STRESS,5X,8HT-STRESS,4X,9HRZ-STRESS/)
  891 FORMAT (1H1,25X,20A4/003X,2HEL,6X,1HR,8X,1HZ,5X,8HR-STRAIN,5X,8
     1HZ-STRAIN,5X,8HT-STRAIN,4X,9HRZ-STRAIN/)
  900 FORMAT (I5,2F9.3,8E13.5)
  910 FORMAT (//10X,14HEND OF PROBLEM,10X,F10.2,32H CPU SECONDS WERE USE
     1D BY CHILES)
  920 FORMAT (//,' <ENTER TITLE OF PROBLEM> ' $)
  931 FORMAT (//,' <ENTER NUMBER OF SINGULAR POINTS (3 MAX.)> ' $)
  941 FORMAT (I)
  945 FORMAT (E)
  951 FORMAT (//,' <ENTER 1,2, OR 3 FOR AXISYMMETRIC, PLANE STRESS, OR P
     1LAIN STRAIN GEOMETRY> ' $)
  961 FORMAT(//,' <ENTER NUMBER OF MATERIALS (10 MAX.)> ' $)
  971 FORMAT(//,' <ENTER 0 FOR REDUCED OUTPUT OR 1 FOR EXTENDED OUTPUT>
     1' $)
  980 FORMAT(//,' <ENTER 0 FOR NO BOUNDARY CONDITIONS ON INTENSITIES OR
     11 TO ALLOW INTENSITY',/,' BOUNDARY CONDITION TO BE READ> ' $)
  990 FORMAT(//,' <ENTER 0 TO READ QMESH FILE OR 1 TO INPUT GEOMETRY POI
     1NT BY POINT> ' $)
 1100 FORMAT(//,' <ENTER 0 FOR ISOTROPIC MATERIAL OR 1 FOR ORTHOTROPIC M
     1ATERIAL> ' $)
 1000 FORMAT (//,' <ENTER NUMBER OF BOUNDARY FLAGS SET IN QMESH> ' $)
 1200 FORMAT (//,' <ENTER YOUNGS MODULUS FOR MATERIAL',I3,'> ' $)
 1300 FORMAT (//,' <ENTER POISSONS RATIO FOR MATERIAL',I3,'> ' $)
 1400 FORMAT (//,' <ENTER COEF. OF THER. EXPANSION FOR MATERIAL',I3,'> '
     1 $)
 1500 FORMAT (//,' <ENTER E11 FOR MATERIAL',I3,'> ' $)
 1600 FORMAT (//,' <ENTER v12 FOR MATERIAL',I3,'> ' $)
 1700 FORMAT (//,' <ENTER E22 FOR MATERIAL',I3,'> ' $)
 1800 FORMAT (//,' <ENTER E33 FOR MATERIAL',I3,'> ' $)
 1900 FORMAT (//,' <ENTER v31 FOR MATERIAL',I3,'> ' $)
 2000 FORMAT (//,' <ENTER v32 FOR MATERIAL',I3,'> ' $)
 2100 FORMAT (//,' <ENTER SHEAR MODULUS FOR MATERIAL',I3,'> ' $)
 2200 FORMAT (//,' <ENTER ANGLE FOR SINGULAR POINT',I3,'> ' $)
```

```
 2200 FORMAT (//,' <ENTER ANGLE FOR SINGULAR POINT',I3,'> ' $)
 2500 FORMAT (//,' <ENTER KODE FOR POINT ',I2,'> ' $)
      END
      SUBROUTINE ZONE
C
C         THIS SUBROUTINE READS MESHING CARDS INTERNALLY AND CREATES A
C         MESH.
C         THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM.
C
      COMMON        IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1              XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2              PR(200),IS(200),JS(200),SH(200)
      COMMON /PAR/  NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
C
C         READ ELEMENT PROPERTIES FROM CARDS
C
      IBOMB=0
      READ (4,160) NEL,NODES,NUMPC
      WRITE (6,170) NEL,NODES,NUMPC
      NFORCE=NUMPC
      NUMSC=NUMPC
      N=0
   10 READ (4,160) M,(IX(I,M),I=1,5)
   20 N=N+1
      IF (M-N) 60,40,30
   30 IX(1,N)=IX(1,N-1)+1
      IX(2,N)=IX(2,N-1)+1
      IX(3,N)=IX(3,N-1)+1
      IX(4,N)=IX(4,N-1)+1
      IX(5,N)=IX(5,N-1)
   40 IF (M-N) 60,50,20
   50 IF (NEL-N) 70,70,10
   60 WRITE (6,180) M
      IBOMB=1
   70 CONTINUE
C
C         READ NODAL POINT DATA FROM CARDS
C
      N=0
   80 READ (4,190) M,CODE(M),R(M),Z(M),XR(M),XZ(M)
      NNL=N+1
      IF (NNL.EQ.1) GO TO 90
      ZX=FLOAT(M-N)
```

```
      DR=(R(M)-R(N))/ZX
      DZ=(Z(M)-Z(N))/ZX
   90 N=N+1
      IF (M-N) 120,110,100
  100 IF (CODE(NNL-1).EQ.CODE(M)) CODE(N)=CODE(M)
      R(N)=R(N-1)+DR
      Z(N)=Z(N-1)+DZ
      XR(N)=0.0
      XZ(N)=0.0
      GO TO 90
  110 IF (NODES-M) 120,130,80
  120 WRITE (6,200) M
      IBOMB=1
  130 CONTINUE
C
C         READ PRESSURE AND/OR SHEAR BOUNDARY STRESSES FROM CARDS
C
      IF (NUMPC.EQ.0) GO TO 150
      DO 140 N=1,NUMPC
      READ (4,210) IP(N),JP(N),PR(N),SH(N)
      IS(N)=IP(N)
  140 JS(N)=JP(N)
  150 CONTINUE
      IF (IBOMB.NE.0) CALL EXIT
      RETURN
C
  160 FORMAT (6I5)
  170 FORMAT (4X,54HNUMBER OF ELEMENTS (NEL)----------------------------
     1--,I5/4X,54HNUMBER OF NODAL POINTS (NODES)-----------------------
     2,I5/4X,54HNUMBER OF PRESSURE CARDS (NUMPC)---------------------,I
     35)
  180 FORMAT (36H0***FATAL ERROR    ELEMENT CARD, M =,I5)
  190 FORMAT (I5,F5.0,4E10.0)
  200 FORMAT (40H0***FATAL ERROR    NODAL POINT CARD, M =,I5)
  210 FORMAT (2I5,2E10.0)
      END
      SUBROUTINE QMESH (NUMTB)
C
C         READ MESH GENERATED BY QMESH (TAPE9).  READ CARDS CONTAINING
C         B.C. INFORMATION.  COMPLETE THE B.C. ARRAYS.
C         THE BASIC REFERENCE DOCUMENT FOR THE MESH GENERATOR IS--
C         R.E. JONES,  USERS MANUAL FOR QMESH, A SELF-ORGANIZING MESH
C         GENERATING PROGRAM  SLA-74-0239, JULY 1974.
C
      COMMON        IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1              XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2              PR(200),IS(200),JS(200),SH(200)
      COMMON /PAR/  NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      DIMENSION     HEDQ(8),IFLAG(1),IBC(100),BCODE(100),PNOR(100)
      DIMENSION     PTOZ(100),XN(12)
      EQUIVALENCE   (IX(1,1),IFLAG(1))
C
      TYPE 200
      ACCEPT 210,XN(1),XN(2),XN(3),XN(4),XN(5),XN(6),XN(7),XN(8),XN(9),
     1XN(10),XN(11)
      OPEN(UNIT=9,FILE=XN,TYPE='OLD',ACCESS='SEQUENTIAL',
     1FORM='UNFORMATTED',ERR=10)
      REWIND 9
      READ (9) HEDQ
      READ (9) NEL,NODES,NFF
C
C         READ NODAL POINT AND ELEMENT DATA FROM QMESH-RENUM TAPE
C
      READ (9) (R(N),Z(N),N=1,NODES)
```

```
      READ (9) (R(N),Z(N),N=1,NODES)
      READ (9) ((IX(I,N),I=1,5),N=1,NEL)
      DO 10 N=1,NODES
      CODE(N)=0.0
      XR(N)=0.0
      XZ(N)=0.0
   10 CONTINUE
      NFORCE=0
      NUMSC=0
C
      IF (NFF.EQ.0) GO TO 140
      IF (NUMTB.GT.0) GO TO 30
      WRITE (6,20)
   20 FORMAT (52H0NO B.C. TABLE.  B.C. FLAGS FROM QMESH TAPE IGNORED.)
      GO TO 140
   30 CONTINUE
C
C         READ BOUNDARY CONDITION DATA FROM QMESH-RENUM TAPE
C
      READ (9) (IFLAG(I),I=1,NFF)
C
C         READ CARDS TO FILL IN B.C. DATA FOR FLAGS FROM QMESH-RENUM TAPE
   35 CONTINUE
      DO 45 J=1,NUMTB
      TYPE 150
      ACCEPT 146,IBC(J)
      TYPE 160
      ACCEPT 145,BCODE(J)
      TYPE 170
      ACCEPT 145,PNOR(J)
      TYPE 180
      ACCEPT 145,PTOZ(J)
   45 CONTINUE
      TYPE 42
   42 FORMAT(//,' DO YOU WISH TO MAKE ANY CHANGES? ' $)
      ACCEPT 44, ANS
   44 FORMAT (A1)
      IF(ANS.EQ.'Y') GO TO 35
      DO 90 II=1,NFF
      IF (IFLAG(II).LE.0) GO TO 50
      NNN=IFLAG(II)
      GO TO 60
   50 NFLAG=IABS(IFLAG(II))
      GO TO 90
   60 DO 80 I=1,NODES
      IF (NNN.NE.I) GO TO 80
      DO 70 J=1,NUMTB
      IF (IBC(J).NE.NFLAG) GO TO 70
      IF (BCODE(J).LT.0.0) GO TO 70
      CODE(I)=BCODE(J)
      XR(I)=PNOR(J)
      XZ(I)=PTOZ(J)
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
C
C         OBTAIN PRESSURE AND SHEAR LOADING INFORMATION FROM FLAG INFO
C
      DO 130 II=1,NFF
      IF (IFLAG(II).LE.0) GO TO 100
      IF (IFLAG(II-1).LE.0) GO TO 130
      NNN=IFLAG(II-1)
      NNN1=IFLAG(II)
      GO TO 110
  100 NFLAG=IABS(IFLAG(II))
      GO TO 130
  110 DO 120 J=1,NUMTB
```

```
  110 DO 120 J=1,NUMIB
      IF (IBC(J).NE.NFLAG) GO TO 120
      IF (BCODE(J).GE.0.0) GO TO 120
      NFORCE=NFORCE+1
      IP(NFORCE)=NNN
      JP(NFORCE)=NNN1
      PR(NFORCE)=PNOR(J)
      NUMSC=NUMSC+1
      IS(NUMSC)=NNN
      JS(NUMSC)=NNN1
      SH(NUMSC)=PTOZ(J)
  120 CONTINUE
  130 CONTINUE
      REWIND 9
      READ (9)
      READ (9)
      READ (9)
      READ (9) ((IX(I,N),I=1,5),N=1,NEL)
  140 CONTINUE
  145 FORMAT (E)
  146 FORMAT (I)
  150 FORMAT (//,' <ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH> ' $)
  160 FORMAT (//,' <ENTER BOUNDARY CODE> ' $)
  170 FORMAT (//,' <ENTER XR OR PN VALUE OF DISPLACEMENT, FORCE OR NORMA
     1L TRACTION> ' $)
  180 FORMAT (//,' <ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR SHEAR
     1 TRACTION> ' $)
  200 FORMAT(//,' <SPECIFY QMESH FILE> ' $)
  210 FORMAT(11A1)
      END
      SUBROUTINE TYPE (IND,NSIN)
C
C        THIS SUBROUTINE IDENTIFIES THE ELEMENT TYPE AND CATALOGS
C        THE DIFFERENT SINGULAR POINTS.  (BETA AND ISP ARRAYS)
C        THIS SUBROUTINE IS CALLED FROM THE MAIN PROGRAM.
C
      COMMON        IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1              XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2              PR(200),IS(200),JS(200),SH(200)
      COMMON /PAR/  NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      INTEGER       BETA
C
      IBOMB=0
      DO 100 I=1,NEL
      IF (NSIN.NE.IX(1,I)) GO TO 10
      IB=1
      K=3
      K1=2
      K2=4
      IBET=7
      IBETL=9
      GO TO 40
   10 IF (NSIN.NE.IX(2,I)) GO TO 20
      IB=2
      K=4
      K1=3
      K2=1
      IBET=9
      IBETL=11
      GO TO 40
   20 IF (NSIN.NE.IX(3,I)) GO TO 30
      IB=3
      K=1
      K1=4
      K2=2
      IBET=11
      IBETL=13
```

```
      IBETL=13
      GO TO 40
   30 IF (NSIN.NE.IX(4,I)) GO TO 100
      IB=4
      K=2
      K1=1
      K2=3
      IBET=13
      IBETL=7
   40 BETA(I)=IB
      IF (ISP(I).NE.0) IBOMB=I
      ISP(I)=IND
      MAT=IX(5,I)
      I1=IND+IND
      IF (MAT.EQ.ISMAT(I1)) GO TO 50
      IF (ISMAT(I1).NE.0) IBOMB=I
      I2=I1-1
   50 CONTINUE
      INX=IX(K,I)
      DO 90 II=1,NEL
      IF (INX.NE.IX(K2,II)) GO TO 60
      IF (IX(K1,I).NE.IX(IB,II)) GO TO 60
      BETA(II)=IBET
      GO TO 80
   60 IF (INX.NE.IX(IB,II)) GO TO 70
      BETA(II)=IBET+1
      GO TO 80
   70 IF (INX.NE.IX(K1,II)) GO TO 90
      IF (IX(K2,I).NE.IX(IB,II)) GO TO 90
      BETA(II)=IBETL
   80 IF (ISP(II).NE.0) IBOMB=I
      ISP(II)=IND
      MAT=IX(5,II)
      I1=IND+IND
      IF (MAT.EQ.ISMAT(I1)) GO TO 90
      IF (ISMAT(I1).NE.0) IBOMB=I
      I2=I1-1
   90 CONTINUE
  100 CONTINUE
      IF (IBOMB.EQ.0) GO TO 110
      WRITE (6,120) IBOMB
  110 RETURN
C
  120 FORMAT (77H0***FATAL ERROR    MATERIAL OR ELEMENT INCONSISTENCY IN
     1 SINGULAR REGION, EL =,I5)
      END
      SUBROUTINE STIFF(NMESHC)
C
C          THIS SUBROUTINE FORMS THE GLOBAL STIFFNESS MATRIX AND
C          LOADING VECTOR IN BLOCKS.
C          THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM.
C          THIS SUBROUTINE CALLS ELSTIF AND MODIFY.
C          THIS SUBROUTINE IS A VERSION OF SUBROUTINE STIFF FROM THE WORK
C          BY E. L. WILSON--A DIGITAL COMPUTER PROGRAM FOR THE FINITE
C          ELEMENT ANALYSIS OF SOLIDS WITH NONLINAR MATERIAL PROPERTIES,
C          JULY 1965, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA.
C
      COMMON         IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1               XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2               PR(200),IS(200),JS(200),SH(200)
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1               NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /EL1/ XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /GLB/ XF(108),XBM(108,54),XC(108,6),XCT(6,6),XBT(6)
      DIMENSION    A(108,54),B(108),C(108,6)
      DIMENSION    B(10),LM(4)
```

```
      DIMENSION      F(10),LM(4)
      EQUIVALENCE   (A(1,1),XBM(1,1)),(B(1),XF(1)),(C(1,1),XC(1,1))
      INTEGER        BETA
C
C        INITIALIZATION
C
      IF (NPAR.EQ.2) NPP=1
      IF (NPAR.EQ.1) NPP=2
      IF (NPAR.EQ.3) NPP=3
      NUMEL=NEL
      NUMNP=NODES
      NUMPC=NFORCE
      REWIND 2
      NB=27
      ND=2*NB
      ND2=2*ND
      STOP=0.
      NUMBLK=0
      DO 5 I=1,6
5     XBT(I)=0.0
      DO 30 M=1,108
      B(M)=0.0
      DO 10 N=1,54
   10 A(M,N)=0.0
      DO 20 N=1,6
   20 C(M,N)=0.0
   30 CONTINUE
C
C        FORM STIFFNESS MATRIX IN BLOCKS
C
   40 NUMBLK=NUMBLK+1
      NH=NB*(NUMBLK+1)
      NM=NH-NB
      NL=NM-NB+1
      KSHIFT=2*NL-2
C
      DO 140 N=1,NUMEL
      IF (IX(5,N).LE.0) GO TO 140
      DO 50 I=1,4
      IF (IX(I,N).LT.NL) GO TO 50
      IF (IX(I,N).LE.NM) GO TO 60
   50 CONTINUE
      GO TO 140
   60 CONTINUE
      DO 70 I=1,4
      NOD=IX(I,N)
      NN(I)=NOD
      RI(I)=R(NOD)
   70 ZI(I)=Z(NOD)
      IMAT=IX(5,N)
      IX(5,N)=-IX(5,N)
      ITYPE=BETA(N)
      NSING=ISP(N)
      IF (NSING.EQ.0) GO TO 80
      RCN=RC(NSING)
      ZCN=ZC(NSING)
      PHIN=PHI(NSING)
      SINPHI=SIN(PHIN)
      COSPHI=COS(PHIN)
   80 CONTINUE
      CALL ELSTIF(ITYPE,NSING,N)
C
C        ADD ELEMENT STIFFNESS MATRIX TO GLOBAL STIFFNESS MATRIX
C
C
C      COMPUTE THERMAL LOADS
C
```

```
      CALL TLOADS(N,P,XKF1,XKF2,ITYPE,NSING)
      DO 90 I=1,4
   90 LM(I)=2*IX(I,N)-2
      DO 110 I=1,4
      DO 110 K=1,2
      II=LM(I)+K-KSHIFT
      KK=2*I-2+K
C
C        BODY FORCES SET TO ZERO
C
C       THERMAL LOADS PUT IN P
      B(II)=B(II)+P(KK)
      DO 110 J=1,4
      DO 110 L=1,2
      JJ=LM(J)+L-II+1-KSHIFT
      LL=2*J-2+L
      IF (JJ.LE.0) GO TO 110
      IF (ND.GE.JJ) GO TO 100
      WRITE (6,410) N
      STOP=1.
      GO TO 140
  100 A(II,JJ)=A(II,JJ)+XK(KK,LL)
  110 CONTINUE
      IF (ITYPE.EQ.6) GO TO 140
      JT=NSING+NSING
      JB=JT-1
      DO 120 J=JB,JT
      J8=J-(NSING-1)*2+8
      DO 120 I=1,4
      DO 120 K=1,2
      JJ=2*I-2+K
      II=LM(I)+K-KSHIFT
  120 XC(II,J)=XC(II,J)+XK(JJ,J8)
      DO 130 I=JB,JT
      DO 130 J=JB,JT
      I1=I-(NSING-1)*2
      J1=J-(NSING-1)*2
  130 XCT(I,J)=XCT(I,J)+XK(I1+8,J1+8)
      IF(NMESHC.NE.0) XBT(JB)=XBT(JB)+XKF1
      IF(NMESHC.NE.0) XBT(JT)=XBT(JT)+XKF2
  140 CONTINUE
C
C        ADD CONCENTRATED FORCES
C
      DO 150 N=NL,NM
      IF (N.GT.NUMNP) GO TO 160
      K=2*N-KSHIFT
      B(K)=B(K)+XZ(N)
  150 B(K-1)=B(K-1)+XR(N)
C
C        ADD PRESSURE BOUNDARY CONDITIONS
C
  160 IF (NUMPC.EQ.0) GO TO 220
      DO 210 L=1,NUMPC
      I=IP(L)
      J=JP(L)
      PP=PR(L)/6.
      DZ=(Z(I)-Z(J))*PP
      DR=(R(J)-R(I))*PP
      RX=2.*R(I)+R(J)
      ZX=R(I)+2.*R(J)
      IF (NPP.EQ.2.OR.NPP.EQ.5) GO TO 170
      RX=3.0
      ZX=3.0
  170 CONTINUE
      II=2*I-KSHIFT
```

```
      II=2*I-KSHIFT
      JJ=2*J-KSHIFT
      IF (II.LE.0.OR.II.GT.ND) GO TO 190
      SINA=0.
      COSA=1.
      IF (CODE(I).GE.0.) GO TO 180
      SINA=SIN(CODE(I)/57.29578)
      COSA=COS(CODE(I)/57.29578)
  180 B(II-1)=B(II-1)+RX*(COSA*DZ+SINA*DR)
      B(II)=B(II)-RX*(SINA*DZ-COSA*DR)
  190 IF (JJ.LE.0.OR.JJ.GT.ND) GO TO 210
      SINA=0.
      COSA=1.
      IF (CODE(J).GE.0.) GO TO 200
      SINA=SIN(CODE(J)/57.29578)
      COSA=COS(CODE(J)/57.29578)
  200 B(JJ-1)=B(JJ-1)+ZX*(COSA*DZ+SINA*DR)
      B(JJ)=B(JJ)-ZX*(SINA*DZ-COSA*DR)
  210 CONTINUE
C
C        ADD SHEAR BOUNDARY CONDITIONS
C
  220 IF (NUMSC.EQ.0) GO TO 280
      DO 270 L=1,NUMSC
      I=IS(L)
      J=JS(L)
      SS=SH(L)/6.
      DZ=(Z(I)-Z(J))*SS
      DR=(R(J)-R(I))*SS
      RX=2.*R(I)+R(J)
      ZX=R(I)+2.*R(J)
      IF (NPP.EQ.2.OR.NPP.EQ.5) GO TO 230
      RX=3.0
      ZX=3.0
  230 CONTINUE
      II=2*I-KSHIFT
      JJ=2*J-KSHIFT
      IF (II.LE.0.OR.II.GT.ND) GO TO 250
      SINA=0.
      COSA=1.
      IF (CODE(I).GE.0.) GO TO 240
      SINA=SIN(CODE(I)/57.29578)
      COSA=COS(CODE(I)/57.29578)
  240 B(II-1)=B(II-1)+RX*(SINA*DZ+COSA*DR)
      B(II)=B(II)-RX*(COSA*DZ-SINA*DR)
  250 IF (JJ.LE.0.OR.JJ.GT.ND) GO TO 270
      SINA=0.
      COSA=1.
      IF (CODE(J).GE.0.) GO TO 260
      SINA=SIN(CODE(J)/57.29578)
      COSA=COS(CODE(J)/57.29578)
  260 B(JJ-1)=B(JJ-1)+ZX*(SINA*DZ+COSA*DR)
      B(JJ)=B(JJ)-ZX*(COSA*DZ-SINA*DR)
  270 CONTINUE
C
C        ADD DISPLACEMENT BOUNDARY CONDITIONS
C
  280 DO 330 M=NL,NH
      IF (M.GT.NUMNP) GO TO 330
      UX=XR(M)
      N=2*M-1-KSHIFT
      IF (CODE(M)) 310,330,290
  290 IF (CODE(M).EQ.1.) GO TO 320
      IF (CODE(M).EQ.2.) GO TO 310
      IF (CODE(M).EQ.3.) GO TO 300
      GO TO 310
  300 CALL MODIFY (ND2,N,UX)
```

```
  300 CALL MODIFY (ND2,N,UX)
  310 UX=XZ(M)
      N=N+1
  320 CALL MODIFY (ND2,N,UX)
  330 CONTINUE
      IF (NSP.EQ.0) GO TO 370
      DO 360 J=1,NSP
      KOD=KODE(J)
      IF (KOD.EQ.0) GO TO 360
      ICOL=2*(J-1)+KOD
      DO 340 I=1,108
  340 XC(I,ICOL)=0.0
      DO 350 I=1,6
      XCT(ICOL,I)=0.0
  350 XCT(I,ICOL)=0.0
      XCT(ICOL,ICOL)=1.0
      XBT(ICOL)=0.0
  360 CONTINUE
  370 CONTINUE
C
C        WRITE BLOCK OF EQUATIONS ON FILE 2 AND SHIFT UP LOWER BLOCK
C
      WRITE (2) (B(N),(A(N,M),M=1,MBAND),N=1,ND)
      IF (NST.GT.0) WRITE (2) ((C(N,L),L=1,NST),N=1,ND)
      DO 380 N=1,ND
      K=N+ND
      B(N)=B(K)
      B(K)=0.
      DO 380 M=1,ND
      A(N,M)=A(K,M)
  380 A(K,M)=0.
      IF (NST.EQ.0) GO TO 400
      DO 390 N=1,ND
      DO 390 M=1,NST
      K=N+ND
      C(N,M)=C(K,M)
      C(K,M)=0.0
  390 CONTINUE
  400 CONTINUE
C
C        CHECK FOR LAST BLOCK
C
      IF (NM.LT.NUMNP) GO TO 40
      IF (STOP.NE.0.) STOP
      RETURN
C
  410 FORMAT (46H  BAND WIDTH EXCEEDS ALLOWABLE FOR ELEMENT NO.,I4)
      END
      SUBROUTINE MODIFY (NEQ,N,U)
C
C        THIS SUBROUTINE SETS THE BOUNDARY CONDITIONS IN THE GLOBAL
C        STIFFNESS MATRIX.
C        THIS SUBROUTINE IS CALLED BY STIFF.
C        THIS SUBROUTINE IS A VERSION OF SUBROUTINE MODIFY FROM THE WORK
C        BY E. L. WILSON--A DIGITAL COMPUTER PROGRAM FOR THE FINITE
C        ELEMENT ANALYSIS OF SOLIDS WITH NONLINAR MATERIAL PROPERTIES,
C        JULY 1965, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA.
C
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /GLB/ XF(108),XBM(108,54),XC(108,6),XCT(6,6),XBT(6)
      DIMENSION    A(108,54),B(108),C(108,6)
      EQUIVALENCE  (A(1,1),XBM(1,1)),(B(1),XF(1)),(C(1,1),XC(1,1))
C
      DO 20 M=2,MBAND
      K=N-M+1
      IF (K.LE.0) GO TO 10
      B(K)=B(K)-A(K,M)*U
```

```
      B(K)=B(K)-A(K,M)*U
      A(K,M)=0.
   10 K=N+M-1
      IF (NEQ.LT.K) GO TO 20
      B(K)=B(K)-A(N,M)*U
      A(N,M)=0.
   20 CONTINUE
      DO 30 I=1,NST
   30 C(N,I)=0.0
      A(N,1)=1.
      B(N)=U
      RETURN
      END
      SUBROUTINE ELSTIF(ITYPE,NSING,IEL)
C
C        THIS SUBROUTINE CONTROLS THE CALCULATION OF THE ELEMENT
C        STIFFNESS MATRIX AND COMPUTES ELEMENT STRESS AND STRAIN
C        MATRICES.
C        THIS SUBROUTINE IS CALLED BY SUBROUTINE STIFF.
C        THIS SUBROUTINE CALLS ADSING AND CALB11.
C
      COMMON /SNG/  RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1              NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /MAT/  D(4,4,10),HED(8)
      COMMON /EL1/  XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /EL2/  RAJ(50),F(4),GAM(4),TAU(4),B11(4,8),B12(4,2)
      COMMON /EL3/  R,Z,RA,RB,ZA,ZB,OJ
      DIMENSION     A(2),H(2),BS(4,10)
      DATA          H(1),H(2)/1.0,1.0/
```

```
      DATA         A(1),A(2)/-0.5773502691,0.5773502691/
C
      ISTRMT=0
      DO 10 J=1,8
      B11(1,J)=0.0
      B11(2,J)=0.0
      B11(3,J)=0.0
      B11(4,J)=0.0
   10 CONTINUE
      DO 20 I=1,10
      DO 20 J=1,10
   20 XK(I,J)=0.0
      K=0
      DO 40 K1=1,2
      AU=A(K1)
      DO 40 K2=1,2
      K=K+1
      BU=A(K2)
      CALL CALB11 (AU,BU,K)
      DO 30 M=1,8
      BD1=B11(1,M)*D(1,1,IMAT)+B11(2,M)*D(2,1,IMAT)+B11(3,M)*D(3,1,IMAT)
     1+B11(4,M)*D(4,1,IMAT)
      BD2=B11(1,M)*D(1,2,IMAT)+B11(2,M)*D(2,2,IMAT)+B11(3,M)*D(3,2,IMAT)
     1+B11(4,M)*D(4,2,IMAT)
      BD3=B11(1,M)*D(1,3,IMAT)+B11(2,M)*D(2,3,IMAT)+B11(3,M)*D(3,3,IMAT)
     1+B11(4,M)*D(4,3,IMAT)
      BD4=B11(1,M)*D(1,4,IMAT)+B11(2,M)*D(2,4,IMAT)+B11(3,M)*D(3,4,IMAT)
     1+B11(4,M)*D(4,4,IMAT)
      DO 30 N=1,8
      B1DB1K=BD1*B11(1,N)+BD2*B11(2,N)+BD3*B11(3,N)+BD4*B11(4,N)
C
C         PERFORM GAUSSIAN 2 BY 2 INTEGRATION (WEIGHT FACTORS H(K1)*H(K2)
C         ARE 1.0*1.0 THEREFORE ARE OMITTED)
C
      XK(M,N)=XK(M,N)+B1DB1K*RAJ(K)
   30 CONTINUE
   40 CONTINUE
C
C         FORM ENRICHED STIFFNESS MATRIX COMPONENTS IF ELEMENT
C         IS IN A SINGULAR FIELD
      IF (ITYPE.NE.6) CALL ADSING (ITYPE,ISTRMT,NSING)
      DO 50 I=1,4
      I2=I+I
      I1=I2-1
      NE2=NN(I)
      NRN(I2)=NE2+NE2
   50 NRN(I1)=NRN(I2)-1
      NRN(10)=NSING+NSING
      NRN(9)=NRN(10)-1
C
C         COMPUTE STRESS AND STRAIN MATRICES AND OUTPUT ON TAPE12.
C
      ISTRMT=1
      DO 60 I=1,4
      B12(I,1)=0.0
   60 B12(I,2)=0.0
      AU=0.0
      BU=0.0
      K=1
      CALL CALB11 (AU,BU,K)
      IF (ITYPE.NE.6) CALL ADSING (ITYPE,ISTRMT,NSING)
      DO 80 I=1,4
      DO 70 J=1,8
   70 BS(I,J)=B11(I,J)
```

```
   70 BS(I,J)=B11(I,J)
      BS(I,9)=B12(I,1)
   80 BS(I,10)=B12(I,2)
      DO 90 J=1,10
      ST(1,J)=D(1,1,IMAT)*BS(1,J)+D(1,2,IMAT)*BS(2,J)+D(1,3,IMAT)*BS(3,J
     1)+D(1,4,IMAT)*BS(4,J)
      ST(2,J)=D(2,1,IMAT)*BS(1,J)+D(2,2,IMAT)*BS(2,J)+D(2,3,IMAT)*BS(3,J
     1)+D(2,4,IMAT)*BS(4,J)
      ST(3,J)=D(3,1,IMAT)*BS(1,J)+D(3,2,IMAT)*BS(2,J)+D(3,3,IMAT)*BS(3,J
     1)+D(3,4,IMAT)*BS(4,J)
      ST(4,J)=D(4,1,IMAT)*BS(1,J)+D(4,2,IMAT)*BS(2,J)+D(4,3,IMAT)*BS(3,J
     1)+D(4,4,IMAT)*BS(4,J)
   90 CONTINUE
      WRITE (12) NRN,R,Z,IMAT,IEL
      WRITE (12) ST
      WRITE (12) BS
      RETURN
      END
      SUBROUTINE TLOADS(IEL,P,XKF1,XKF2,ITYPE,NSING)
C
C      THIS SUBROUTINE COMPUTES THE ELEMENT THERMAL LOAD VECTOR
C
      COMMON/KLD/ ET1,ET2,ET3,ET4
      COMMON /HEAT/ TSTR(1000),ALPHA(10)
      DIMENSION XA(4),XB(4),FA(4),FB(4)
      DIMENSION P(8)
      COMMON        IX(5,1000),R(1000),Z(1000),CODE(1000),XR(1000),
     1              XZ(1000),ISP(1000),BETA(1000),IP(200),JP(200),
     2              PR(200),IS(200),JS(200),SH(200)
      COMMON /SNG/  RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1              NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /MAT/  D(4,4,10),HED(8)
      COMMON /EL1/  XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /EL2/  RAJ(50),F(4),GAM(4),TAU(4),B11(4,8),B12(4,2)
      DIMENSION     A(2),H(2),BS(4,10)
      DATA          H(1),H(2)/1.0,1.0/
      DATA          A(1),A(2)/-0.5773502691,0.5773502691/
      DATA          (XA(I),I=1,4)/-1.0,1.0,1.0,-1.0/
      DATA          (XB(I),I=1,4)/-1.0,-1.0,1.0,1.0/
      DO 3 I=1,8
3     P(I)=0.0
C
      ISTRMT=0
      DO 10 J=1,8
      B11(1,J)=0.0
      B11(2,J)=0.0
      B11(3,J)=0.0
      B11(4,J)=0.0
   10 CONTINUE
      K=0
      DO 40 K1=1,2
      AU=A(K1)
      DO 40 K2=1,2
      K=K+1
      BU=A(K2)
      CALL CALB11 (AU,BU,K)
C
C      COMPUTE THERMAL STRAINS AT INTEGRATION POINTS
C
      I1=IX(1,IEL)
      I2=IX(2,IEL)
      I3=IX(3,IEL)
      I4=IX(4,IEL)
      ET1=TSTR(I1)*ALPHA(IMAT)
      ET2=TSTR(I2)*ALPHA(IMAT)
      ET3=TSTR(I3)*ALPHA(IMAT)
      ET4=TSTR(I4)*ALPHA(IMAT)
```

```
      ET4=TSTR(I4)*ALPHA(IMAT)
      DO 5 I=1,4
5      F(I)=.25*(1.+AU*XA(I))*(1.+BU*XB(I))
      ET=F(1)*ET1+F(2)*ET2+F(3)*ET3+F(4)*ET4
      DO 30 M=1,8
      BD1=B11(1,M)*D(1,1,IMAT)+B11(2,M)*D(2,1,IMAT)+B11(3,M)*D(3,1,IMAT)
     1+B11(4,M)*D(4,1,IMAT)
      BD2=B11(1,M)*D(1,2,IMAT)+B11(2,M)*D(2,2,IMAT)+B11(3,M)*D(3,2,IMAT)
     1+B11(4,M)*D(4,2,IMAT)
      BD3=B11(1,M)*D(1,3,IMAT)+B11(2,M)*D(2,3,IMAT)+B11(3,M)*D(3,3,IMAT)
     1+B11(4,M)*D(4,3,IMAT)
      BD4=B11(1,M)*D(1,4,IMAT)+B11(2,M)*D(2,4,IMAT)+B11(3,M)*D(3,4,IMAT)
     1+B11(4,M)*D(4,4,IMAT)
      B1DE=(BD1+BD2+BD3)*ET
      P(M)=P(M)+B1DE*RAJ(K)
30    CONTINUE
40    CONTINUE
      IF(ITYPE.NE.6) CALL KTLOAD(ITYPE,XKF1,XKF2,NSING)
      RETURN
      END
      SUBROUTINE ADSING (ITYPE,ISTRMT,NSING)
C
C         THIS SUBROUTINE ADDS THE SINGULAR ROWS AND COLUMNS TO THE
C         ELEMENT STIFFNESS MATRIX.
C         THIS SUBROUTINE IS CALLED BY ELSTIF.
C         THIS SUBROUTINE CALLS CALB11, CALKIL, CALQ, AND CALQI.
C
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /MAT/ D(4,4,10),HED(8)
      COMMON /EL1/ XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /EL2/ RAJ(50),F(4),GAM(4),TAU(4),B11(4,8),B12(4,2)
      COMMON /EL3/ R,Z,RA,RB,ZA,ZB,OJ
      COMMON /QUE/ QI(4,4),Q(4),QR(4),QZ(4)
      COMMON /ORTO/ KORTSW
      DIMENSION    A(7),H(7)
      DIMENSION    B1DB2(8,2,50),B2DB2(2,2,50)
      DATA         (H(I),I=1,7)/.1294849662,.2797053915,.3818300505,
     1             .4179591837,.3818300505,.2797053915,.1294849662/
      DATA         (A(I),I=1,7)/-.9491079123,-.7415311856,-.4058451514,
     1             0.0,.4058451514,.7415311856,.9491079123/
      DATA         NINTP/7/
C
      IF (KORTSW.GT.0) CALL ROOTS (IMAT)
      CALL CALQI (NSING)
      ASSIGN 30 TO IST1
      ASSIGN 160 TO IST2
      IF (ISTRMT.EQ.1) GO TO 10
      ASSIGN 20 TO IST1
      ASSIGN 80 TO IST2
   10 CONTINUE
C
      DO 120 K1=1,NINTP
      DO 110 K2=1,NINTP
      K=1
      AU=0.0
      BU=0.0
      GO TO IST1, (30,20)
   20 CONTINUE
      K=K2+(K1-1)*NINTP
      AU=A(K1)
      BU=A(K2)
   30 CONTINUE
      CALL CALB11 (AU,BU,K)
      CALL CALKIL (ITYPE,AU,BU,ALPHA,BETA,FCR,FCZ,FKL)
      CALL CALQ (NSING)
      DO 40 I=1,4
```

```
      DO 40 I=1,4
      B12(I,1)=0.0
      B12(I,2)=0.0
   40 CONTINUE
      FJ=FKL*OJ
      BJ=BETA*OJ
      ALJ=ALPHA*OJ
      FR=FKL/R
      DO 60 I=1,4
      B12(1,1)=B12(1,1)+QI(1,I)*GAM(I)*FJ+QI(1,I)*F(I)*BJ
      B12(1,2)=B12(1,2)+QI(2,I)*GAM(I)*FJ+QI(2,I)*F(I)*BJ
      IF (NPAR.NE.1) GO TO 50
      B12(2,1)=B12(2,1)-QI(1,I)*F(I)*FR
      B12(2,2)=B12(2,2)-QI(2,I)*F(I)*FR
   50 B12(3,1)=B12(3,1)+QI(3,I)*TAU(I)*FJ+QI(3,I)*F(I)*ALJ
      B12(3,2)=B12(3,2)+QI(4,I)*TAU(I)*FJ+QI(4,I)*F(I)*ALJ
      B12(4,1)=B12(4,1)+(QI(1,I)*TAU(I)+QI(3,I)*GAM(I))*FJ+ALJ*QI(1,I)*F
     1(I)+BJ*QI(3,I)*F(I)
   60 B12(4,2)=B12(4,2)+(QI(2,I)*TAU(I)+QI(4,I)*GAM(I))*FJ+ALJ*QI(2,I)*F
     1(I)+BJ*QI(4,I)*F(I)
      B12(1,1)=B12(1,1)+FKL*QR(1)+FCR*Q(1)
      B12(1,2)=B12(1,2)+FKL*QR(2)+FCR*Q(2)
      IF (NPAR.NE.1) GO TO 70
      B12(2,1)=B12(2,1)+FR*Q(1)
      B12(2,2)=B12(2,2)+FR*Q(2)
   70 B12(3,1)=B12(3,1)+FKL*QZ(3)+FCZ*Q(3)
      B12(3,2)=B12(3,2)+FKL*QZ(4)+FCZ*Q(4)
      B12(4,1)=B12(4,1)+FKL*QZ(1)+FCZ*Q(1)+FKL*QR(3)+FCR*Q(3)
      B12(4,2)=B12(4,2)+FKL*QZ(2)+FCZ*Q(2)+FKL*QR(4)+FCR*Q(4)
      GO TO IST2, (160,80)
   80 CONTINUE
      DO 100 N=1,2
      D1B=D(1,1,IMAT)*B12(1,N)+D(1,2,IMAT)*B12(2,N)+D(1,3,IMAT)*B12(3,N)
     1+D(1,4,IMAT)*B12(4,N)
      D2B=D(2,1,IMAT)*B12(1,N)+D(2,2,IMAT)*B12(2,N)+D(2,3,IMAT)*B12(3,N)
     1+D(2,4,IMAT)*B12(4,N)
      D3B=D(3,1,IMAT)*B12(1,N)+D(3,2,IMAT)*B12(2,N)+D(3,3,IMAT)*B12(3,N)
     1+D(3,4,IMAT)*B12(4,N)
      D4B=D(4,1,IMAT)*B12(1,N)+D(4,2,IMAT)*B12(2,N)+D(4,3,IMAT)*B12(3,N)
     1+D(4,4,IMAT)*B12(4,N)
      DO 90 M=1,8
      B1DB2(M,N,K)=B11(1,M)*D1B+B11(2,M)*D2B+B11(3,M)*D3B+B11(4,M)*D4B
   90 CONTINUE
      B2DB2(1,N,K)=B12(1,1)*D1B+B12(2,1)*D2B+B12(3,1)*D3B+B12(4,1)*D4B
      B2DB2(2,N,K)=B12(1,2)*D1B+B12(2,2)*D2B+B12(3,2)*D3B+B12(4,2)*D4B
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
C
C        GAUSSIAN INTEGRATION OF K12 AND K22 TERMS
C
      DO 150 I=1,10
      XK2=0.0
      XK1=0.0
      DO 140 K1=1,NINTP
      HK1=H(K1)
      DO 140 K2=1,NINTP
      K=K2+(K1-1)*NINTP
      HHR=HK1*H(K2)*RAJ(K)
      IF (I.GT.8) GO TO 130
      XK1=XK1+B1DB2(I,1,K)*HHR
      XK2=XK2+B1DB2(I,2,K)*HHR
      GO TO 140
  130 XK1=XK1+B2DB2(I-8,1,K)*HHR
      XK2=XK2+B2DB2(I-8,2,K)*HHR
  140 CONTINUE
      XK(I,9)=XK1
```

```
      XK(I,9)=XK1
      XK(9,I)=XK1
      XK(I,10)=XK2
      XK(10,I)=XK2
  150 CONTINUE
  160 CONTINUE
      RETURN
      END
      SUBROUTINE KTLOAD(ITYPE,XKF1,XKF2,NSING)
C
C      THIS SUBROUTINE ADDS THERMAL FORCES ASSOCIATED WITH THE SINGULAR TERMS
C      IT IS SIMILAR TO SUBROUTINE ADSING
C
      COMMON/KLD/ ET1,ET2,ET3,ET4
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /MAT/ D(4,4,10),HED(8)
      COMMON /EL1/ XK(10,10),NRN(10),NN(4),ST(4,10)
      COMMON /EL2/ RAJ(50),F(4),GAM(4),TAU(4),B11(4,8),B12(4,2)
      COMMON /EL3/ R,Z,RA,RB,ZA,ZB,OJ
      COMMON /QUE/ QI(4,4),Q(4),QR(4),QZ(4)
      COMMON /ORTO/ KORTSW
      DIMENSION    A(7),H(7)
      DIMENSION    B1DB2(8,2,50),B2DB2(2,2,50)
      DATA         (H(I),I=1,7)/.1294849662,.2797053915,.3818300505,
     1             .4179591837,.3818300505,.2797053915,.1294849662/
      DATA         (A(I),I=1,7)/-.9491079123,-.7415311856,-.4058451514,
     1             0.0,.4058451514,.7415311856,.9491079123/
      DATA         NINTP/7/
C
      ISTRMT=0
      IF (KORTSW.GT.0) CALL ROOTS (IMAT)
      CALL CALQI (NSING)
      ASSIGN 30 TO IST1
      ASSIGN 160 TO IST2
      IF (ISTRMT.EQ.1) GO TO 10
      ASSIGN 20 TO IST1
      ASSIGN 80 TO IST2
   10 CONTINUE
C
      DO 120 K1=1,NINTP
      DO 110 K2=1,NINTP
      K=1
      AU=0.0
      BU=0.0
      GO TO IST1, (30,20)
   20 CONTINUE
      K=K2+(K1-1)*NINTP
      AU=A(K1)
      BU=A(K2)
   30 CONTINUE
      CALL CALB11 (AU,BU,K)
      CALL CALKIL (ITYPE,AU,BU,ALPHA,BETA,FCR,FCZ,FKL)
      CALL CALQ(NSING)
      DO 40 I=1,4
      B12(I,1)=0.0
      B12(I,2)=0.0
   40 CONTINUE
      FJ=FKL*OJ
      BJ=BETA*OJ
      ALJ=ALPHA*OJ
      FR=FKL/R
      DO 60 I=1,4
      B12(1,1)=B12(1,1)+QI(1,I)*GAM(I)*FJ+QI(1,I)*F(I)*BJ
      B12(1,2)=B12(1,2)+QI(2,I)*GAM(I)*FJ+QI(2,I)*F(I)*BJ
      IF (NPAR.NE.1) GO TO 50
      B12(2,1)=B12(2,1)-QI(1,I)*F(I)*FR
```

```
      B12(2,1)=B12(2,1)-QI(1,I)*F(I)*FR
      B12(2,2)=B12(2,2)-QI(2,I)*F(I)*FR
   50 B12(3,1)=B12(3,1)+QI(3,I)*TAU(I)*FJ+QI(3,I)*F(I)*ALJ
      B12(3,2)=B12(3,2)+QI(4,I)*TAU(I)*FJ+QI(4,I)*F(I)*ALJ
      B12(4,1)=B12(4,1)+(QI(1,I)*TAU(I)+QI(3,I)*GAM(I))*FJ+ALJ*QI(1,I)*F
     1(I)+BJ*QI(3,I)*F(I)
   60 B12(4,2)=B12(4,2)+(QI(2,I)*TAU(I)+QI(4,I)*GAM(I))*FJ+ALJ*QI(2,I)*F
     1(I)+BJ*QI(4,I)*F(I)
      B12(1,1)=B12(1,1)+FKL*QR(1)+FCR*Q(1)
      B12(1,2)=B12(1,2)+FKL*QR(2)+FCR*Q(2)
      IF (NPAR.NE.1) GO TO 70
      B12(2,1)=B12(2,1)+FR*Q(1)
      B12(2,2)=B12(2,2)+FR*Q(2)
   70 B12(3,1)=B12(3,1)+FKL*QZ(3)+FCZ*Q(3)
      B12(3,2)=B12(3,2)+FKL*QZ(4)+FCZ*Q(4)
      B12(4,1)=B12(4,1)+FKL*QZ(1)+FCZ*Q(1)+FKL*QR(3)+FCR*Q(3)
      B12(4,2)=B12(4,2)+FKL*QZ(2)+FCZ*Q(2)+FKL*QR(4)+FCR*Q(4)
      GO TO IST2, (160,80)
   80 CONTINUE
      ET=F(1)*ET1+F(2)*ET2+F(3)*ET3+F(4)*ET4
      D1B=(D(1,1,IMAT)+D(1,2,IMAT)+D(1,3,IMAT))*ET
      D2B=(D(2,1,IMAT)+D(2,2,IMAT)+D(2,3,IMAT))*ET
      D3B=(D(3,1,IMAT)+D(3,2,IMAT)+D(3,3,IMAT))*ET
      B2DB2(1,1,K)=B12(1,1)*D1B+B12(2,1)*D2B+B12(3,1)*D3B
      B2DB2(2,1,K)=B12(1,2)*D1B+B12(2,2)*D2B+B12(3,2)*D3B
110   CONTINUE
120   CONTINUE
      XKF1=0.0
      XKF2=0.0
      DO 140 K1=1,NINTP
      HK1=H(K1)
      DO 140 K2=1,NINTP
      K=K2+(K1-1)*NINTP
      HHR=HK1*H(K2)*RAJ(K)
      XKF1=XKF1+B2DB2(1,1,K)*HHR
140   XKF2=XKF2+B2DB2(2,1,K)*HHR
160   CONTINUE
      RETURN
      END
      SUBROUTINE ROOTS (IMAT)
C
C      THIS SUBROUTINE COMPUTES THE COMPLEX ROOTS OF THE CAHRACTERISTIC
C
      COMMON /ROOT/E1S(10),E2S(10),XNU1S(10),XNU2S(10),G2S(10),ANGS(10),
     1               E3S(10),XNU31S(10),XNU32S(10)
      COMMON /SNG/ DUM(30),NPAR
      COMMON /MAT/ D(4,4,10),HED(8)
      COMMON /IMAG/ S1,S2,P1,P2,Q1,Q2
      DIMENSION COEF(5),WR(4),WI(4)
      COMPLEX  S1,S2,P1,P2,Q1,Q2
      A11=1./E1S(IMAT)
      A12=-XNU2S(IMAT)/E2S(IMAT)
      A22=1./E2S(IMAT)
      A66=1./G2S(IMAT)
      IF (NPAR.EQ.2) GO TO 10
      A33=1./E3S(IMAT)
      A13=-XNU31S(IMAT)/E3S(IMAT)
      A23=-XNU2S(IMAT)/E2S(IMAT)
      B11=(A11*A33-A13*A13)/A33
      B12=(A12*A33-A13*A23)/A33
      B22=(A22*A33-A23*A23)/A33
      B66=A66
      A11=B11
      A12=B12
      A22=B22
      A66=B66
   10 CONTINUE
```

```
   10 CONTINUE
      COEF(1)=A11
      COEF(2)=0.
      COEF(3)=2.*A12+A66
      COEF(4)=0.
      COEF(5)=A22
      CALL RPQR (4,COEF,WR,WI,IERROR)
      S1=CMPLX(WR(1),WI(1))
      S2=CMPLX(WR(3),WI(3))
      P1=CMPLX(WR(2),WI(2))
      P2=CMPLX(WR(4),WI(4))
      P1=A11*S1*S1+A12
      Q1=(A12*S1*S1+A22)/S1
      P2=A11*S2*S2+A12
      Q2=(A12*S2*S2+A22)/S2
      END
      SUBROUTINE CALB11 (AU,BU,K)
C
C        THIS SUBROUTINE CALCULATES THE B MATRIX WHICH RELATES ELEMENT
C        STRAIN TO NODAL POINT DISPLACEMENTS.
C        THIS SUBROUTINE IS CALLED BY ELSTIF, AND ADSING.
C
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /EL2/ RAJ(50),F(4),GAM(4),TAU(4),B11(4,8),B12(4,2)
      COMMON /EL3/ R,Z,RA,RB,ZA,ZB,OJ
      DIMENSION    XA(4),XB(4),FA(4),FB(4)
      DATA         (XA(I),I=1,4)/-1.0,1.0,1.0,-1.0/
      DATA         (XB(I),I=1,4)/-1.0,-1.0,1.0,1.0/
C
      DO 10 I=1,4
      F(I)=.25*(1.+AU*XA(I))*(1.+BU*XB(I))
      FA(I)=.25*XA(I)*(1.+XB(I)*BU)
   10 FB(I)=.25*XB(I)*(1.+XA(I)*AU)
      ZA=0.0
      ZB=0.0
      RA=0.0
      RB=0.0
      R=0.0
      Z=0.0
      DO 20 I=1,4
      ZB=ZB+FB(I)*ZI(I)
      RA=RA+FA(I)*RI(I)
      ZA=ZA+FA(I)*ZI(I)
      RB=RB+FB(I)*RI(I)
      Z=Z+F(I)*ZI(I)
   20 R=R+F(I)*RI(I)
      RINTK=R
      IF (NPAR.GT.1) RINTK=1.0
      AJK=RA*ZB-RB*ZA
      RAJ(K)=RINTK*ABS(AJK)
      OJ=1.0/AJK
      DO 30 J=1,4
      L=J+J
      I=L-1
      B11(1,I)=(ZB*FA(J)-ZA*FB(J))*OJ
      B11(2,I)=F(J)/R
      IF (NPAR.GT.1) B11(2,I)=0.0
      B11(3,L)=(RA*FB(J)-RB*FA(J))*OJ
      B11(4,I)=B11(3,L)
   30 B11(4,L)=B11(1,I)
      DO 40 I=1,4
      GAM(I)=ZA*FB(I)-ZB*FA(I)
   40 TAU(I)=RB*FA(I)-RA*FB(I)
      RETURN
      END
      SUBROUTINE CALK11 (ITYPE AU BU ALPHA BETA FCR FCZ FKL)
```

```
      SUBROUTINE CALKIL (ITYPE,AU,BU,ALPHA,BETA,FCR,FCZ,FKL)
C
C        THIS SUBROUTINE CALCULATES THE KILL FUNCTION R USED IN ELEMENT
C        TYPE B TO ELIMINATE THE INCOMPATIBILITY.
C        THIS SUBROUTINE IS CALLED BY ADSING.
C
      COMMON /EL3/ R,Z,RA,RB,ZA,ZB,OJ
C
      FKL=1.0
      FCR=0.0
      FCZ=0.0
      ALPHA=0.0
      BETA=0.0
      IF (ITYPE.LT.6) GO TO 100
      IF (ITYPE.NE.7) GO TO 10
      FKL=.5*(1.-AU)
      FCA=-.5
      FCB=0.
      GO TO 90
   10 IF (ITYPE.NE.8) GO TO 20
      FKL=.25*(1.-AU)*(1.-BU)
      FCA=-.25*(1.-BU)
      FCB=-.25*(1.-AU)
      GO TO 90
   20 IF (ITYPE.NE.9) GO TO 30
      FKL=.5*(1.-BU)
      FCA=0.
      FCB=-.5
      GO TO 90
   30 IF (ITYPE.NE.10) GO TO 40
      FKL=.25*(1.+AU)*(1.-BU)
      FCA=.25*(1.-BU)
      FCB=-.25*(1.+AU)
      GO TO 90
   40 IF (ITYPE.NE.11) GO TO 50
      FKL=.5*(1.+AU)
      FCA=.5
      FCB=0.0
      GO TO 90
   50 IF (ITYPE.NE.12) GO TO 60
      FKL=.25*(1.+AU)*(1.+BU)
      FCA=.25*(1.+BU)
      FCB=.25*(1.+AU)
      GO TO 90
   60 IF (ITYPE.NE.13) GO TO 70
      FKL=.5*(1.+BU)
      FCA=0.
      FCB=.5
      GO TO 90
   70 IF (ITYPE.NE.14) GO TO 80
      FKL=.25*(1.-AU)*(1.+BU)
      FCA=-.25*(1.+BU)
      FCB=.25*(1.-AU)
      GO TO 90
   80 WRITE (6,110) ITYPE
      STOP
   90 ALPHA=RB*FCA-RA*FCB
      BETA=ZA*FCB-ZB*FCA
      FCR=(ZB*FCA-ZA*FCB)*OJ
      FCZ=(-RB*FCA+RA*FCB)*OJ
  100 CONTINUE
      RETURN
C
  110 FORMAT (31H0***FATAL ERROR    ELEMENT TYPE,I5,20H IS NOT A VALAD T
     1YPE)
      END
      SUBROUTINE CALO(NSING)
```

```
      SUBROUTINE CALQ(NSING)
C
C         THIS SUBROUTINE CALCULATES THE Q TERMS USED IN FORMULATING
C         THE SINGULAR PART OF THE ELEMENT STIFFNESS MATRIX.
C         THIS SUBROUTINE IS CALLED BY ADSING.
C
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /EL3/ R,Z,RA,RB,ZA,ZB,OJ
      COMMON /QUE/ QI(4,4),Q(4),QR(4),QZ(4)
      COMMON /ORTO/ KORTSW
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /MAT/ D(4,4,10),HED(20)
      DIMENSION    QT(4)
C
      ZD=Z-ZCN
      RD=R-RCN
      RHO=SQRT(RD*RD+ZD*ZD)
      R2=SQRT(RHO)
      IR=2*NSING
      IA=IR-1
      IMATREF=ISMAT(IR)
      IMATADJ=ISMAT(IA)
      GAVG=.5*(D(4,4,IMATREF)+D(4,4,IMATADJ))
      RATIO=GAVG/D(4,4,IMAT)
      IF(IMAT.EQ.IMATADJ.OR.IMAT.EQ.IMATREF) GO TO 5
      TYPE 3,IMAT,IMATADJ,IMATREF
  3   FORMAT(' BIMATERIAL INTERFACE DOES NOT MATCH ELEMENT MATERIAL ',
     1' IMAT = ',I5,' IMATADJ = ',I5,' IMATREF = ',I5)
      TYPE 33, NSING,IR,IA,ISMAT(IR),ISMAT(IA)
 33   FORMAT(' NSING,IR,IA,ISMAT(IR),ISMAT(IA) ',5I5)
      CALL EXIT
  5   CONTINUE
      RHO1=1.0/RHO
      COST=(COSPHI*RD+SINPHI*ZD)*RHO1
      SI=(COSPHI*ZD-SINPHI*RD)*RHO1
      SI=SI/ABS(SI)
      THEDA=SI*ACOS(COST)
      CTP=COS(THEDA+PHIN)
      STP=SIN(THEDA+PHIN)
      T2=THEDA*0.5
      C2=COS(T2)
      S2=SIN(T2)
      XNU=XNUS(IMAT)
      SP=SINPHI
      CP=COSPHI
      IF (KORTSW.GT.0) CALL CALG (0,G1,G2,G3,G4,G5,G6,G7,G8,THEDA)
      IF (KORTSW.GT.0) GO TO 10
      XKAP=3.-4.*XNU
      IF (NPAR.EQ.2) XKAP=(3.-XNU)/(1.+XNU)
      XKAP1=(XKAP-1.)/2.
      XKAP2=(XKAP+1.)/2.
      G1=C2*(XKAP1+S2*S2)
      G2=S2*(XKAP2+C2*C2)
      G3=S2*(XKAP2-C2*C2)
      G4=-C2*(XKAP1-S2*S2)
      G5=0.5*(-S2*XKAP1+2.0*C2*C2*S2-S2*S2*S2)
      G6=0.5*(C2*XKAP2-2.0*C2*S2*S2+C2*C2*C2)
      G7=0.5*(C2*XKAP2+2.0*C2*S2*S2-C2*C2*C2)
      G8=-0.5*(-S2*XKAP1-2.0*C2*C2*S2+S2*S2*S2)
C
C          MODIFY SINGULAR FACTORS TO ACCOMIDATE BIMATERIAL INTERFACE
C
      G2=G2*RATIO
      G3=G3*RATIO
      G6=G6*RATIO
      G7=G7*RATIO
```

```
      G7=G7*RATIO
   10 CONTINUE
      Q(1)=R2*(G1*CP-G3*SP)
      Q(2)=R2*(G2*CP-G4*SP)
      Q(3)=R2*(G1*SP+G3*CP)
      Q(4)=R2*(G2*SP+G4*CP)
      R3=0.5/R2
      QR(1)=R3*(G1*CP-G3*SP)
      QR(2)=R3*(G2*CP-G4*SP)
      QR(3)=R3*(G1*SP+G3*CP)
      QR(4)=R3*(G2*SP+G4*CP)
      QT(1)=R2*(G5*CP-G7*SP)
      QT(2)=R2*(G6*CP-G8*SP)
      QT(3)=R2*(G5*SP+G7*CP)
      QT(4)=R2*(G6*SP+G8*CP)
      DO 20 I=1,4
      QRHO=QR(I)
      QTHEDA=QT(I)
      QR(I)=QRHO*CTP-QTHEDA*STP*RHO1
      QZ(I)=QRHO*STP+QTHEDA*CTP*RHO1
   20 CONTINUE
      RETURN
      END
      SUBROUTINE CALQI (NSING)
C
C         THIS SUBROUTINE CALCULATES THE Q BAR TERMS USED IN
C         FORMULATING THE SINGULAR PART OF THE ELEMENT STIFFNESS MATRIX.
C         THIS SUBROUTINE IS CALLED BY ADSING.
C         THIS SUBROUTINE CALLS CALSI.
C
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
      COMMON /QUE/ QI(4,4),Q(4),QR(4),QZ(4)
      COMMON /ORTO/ KORTSW
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /MAT/ D(4,4,10),HED(20)
C
      XNU=XNUS(IMAT)
      XKAP=3.-4.*XNU
      IF (NPAR.EQ.2) XKAP=(3.-XNU)/(1.+XNU)
      XKAP1=(XKAP-1.)/2.
      XKAP2=(XKAP+1.)/2.
      DO 30 I=1,4
      ZD=ZI(I)-ZCN
      RD=RI(I)-RCN
      RHO=SQRT(RD*RD+ZD*ZD)
      IF (RHO.GT.1.0E-30) GO TO 10
      QI(1,I)=0.0
      QI(2,I)=0.0
      QI(3,I)=0.0
      QI(4,I)=0.0
      GO TO 30
C
C         CALCULATE THEDA BY VECTOR MULTIPLICATION.
C
   10 CONTINUE
      RHO1=1.0/RHO
      COST=(COSPHI*RD+SINPHI*ZD)*RHO1
      SI=(COSPHI*ZD-SINPHI*RD)*RHO1
      IF (ABS(SI).LT.5.0E-4) CALL CALSI (I,SI)
      SI=SI/ABS(SI)
      THEDA=SI*ACOS(COST)
      T2=THEDA*0.5
      C2=COS(T2)
```

```
      S2=SIN(T2)
      R2=SQRT(RHO)
C
C      COMPUTE COMPLEX FUNCTIONS FOR ORTHOTROPIC MATERIALS
C
      IF (KORTSW.GT.0) CALL CALG (1,G1,G2,G3,G4,G5,G6,G7,G8,THEDA)
      IF (KORTSW.GT.0) GO TO 20
      IR=2*NSING
      IA=IR-1
      IMATREF=ISMAT(IR)
      IMATADJ=ISMAT(IA)
      GAVG=.5*(D(4,4,IMATREF)+D(4,4,IMATADJ))
      RATIO=GAVG/D(4,4,IMAT)
      G1=C2*(XKAP1+S2*S2)
      G2=S2*(XKAP2+C2*C2)
      G3=S2*(XKAP2-C2*C2)
      G4=C2*(-XKAP1+S2*S2)
      G2=G2*RATIO
      G3=G3*RATIO
   20 CONTINUE
      QI(1,I)=R2*(G1*COSPHI-G3*SINPHI)
      QI(2,I)=R2*(G2*COSPHI-G4*SINPHI)
      QI(3,I)=R2*(G1*SINPHI+G3*COSPHI)
      QI(4,I)=R2*(G2*SINPHI+G4*COSPHI)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE CALG (ICHK,G1,G2,G3,G4,G5,G6,G7,G8,THEDA)
      COMMON /IMAG/ S1,S2,P1,P2,Q1,Q2
      COMPLEX S1,S2,P1,P2,Q1,Q2,SMS,GI,CS2,CS1
      CT=COS(THEDA)
      ST=SIN(THEDA)
      SMS=S1-S2
```

```
      SMS=1./SMS
      CS2=CSQRT(CT+S2*ST)
      CS1=CSQRT(CT+S1*ST)
      GI=SMS*(S1*P2*CS2-S2*P1*CS1)
      G1=REAL(GI)
      GI=SMS*(P2*CS2-P1*CS1)
      G2=REAL(GI)
      GI=SMS*(S1*Q2*CS2-S2*Q1*CS1)
      G3=REAL(GI)
      GI=SMS*(Q2*CS2-Q1*CS1)
      G4=REAL(GI)
      IF (ICHK.EQ.1) RETURN
      CS2=.5*(-ST+S2*CT)/CS2
      CS1=.5*(-ST+S1*CT)/CS1
      GI=SMS*(S1*P2*CS2-S2*P1*CS1)
      G5=REAL(GI)
      GI=SMS*(P2*CS2-P1*CS1)
      G6=REAL(GI)
      GI=SMS*(S1*Q2*CS2-S2*Q1*CS1)
      G7=REAL(GI)
      GI=SMS*(Q2*CS2-Q1*CS1)
      G8=REAL(GI)
      RETURN
      END
      SUBROUTINE CALSI (I,SI)
C
C        THIS SUBROUTINE CALCULATES THE SIGN OF AND ANGLE BETWEEN
C        TWO VECTORS.
C        THIS SUBROUTINE IS CALLED BY CALQI.
C
      COMMON /SNG/ RI(4),ZI(4),XNUS(10),PHI(3),RC(3),ZC(3),KODE(3),
     1             NPAR,IMAT,RCN,ZCN,PHIN,SINPHI,COSPHI
C
      IF (I.EQ.3) KI=1
      IF (I.EQ.4) KI=2
      IF (I.EQ.1) KI=3
      IF (I.EQ.2) KI=4
      ZD=ZI(KI)-ZCN
      RD=RI(KI)-RCN
      RHO=1.0/SQRT(RD*RD+ZD*ZD)
      SI=(COSPHI*ZD-SINPHI*RD)*RHO
      RETURN
      END
      SUBROUTINE SOLV
C
C        THIS SUBROUTINE REDUCES THE GLOBAL SET OF EQUATIONS BLOCK BY
C        BLOCK, PERFORMS THE BACK SUBSTITUTION, OUTPUTS THE
C        DISPLACEMENT SOLUTION, CALCULATES AND OUTPUTS THE SINGULAR
C        INTENSITIES.
C        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM.
C        THIS SUBROUTINE CALLS SYMSOL.
C        THIS SUBROUTINE IS A VERSION OF SUBROUTINE SOLVE FROM THE WORK
C        BY E. L. WILSON--A DIGITAL COMPUTER PROGRAM FOR THE FINITE
C        ELEMENT ANALYSIS OF SOLIDS WITH NONLINAR MATERIAL PROPERTIES,
C        JULY 1965, UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA.
C
      COMMON /MAT/ D(4,4,10),HED(8)
      COMMON /PAR/ NODES,NEL,NFORCE,NUMSC,NST,NSP,ISMAT(6),MBAND,NUMBLK
      COMMON /GLB/ XF(108),XBM(108,54),XC(108,6),XCT(6,6),XBT(6)
      COMMON /ORTO/ KORTSW
      DIMENSION    A(108,54),B(108),C(108,6)
      DIMENSION    XBCT(6,6)
      EQUIVALENCE  (A(1,1),XBM(1,1)),(B(1),XF(1)),(C(1,1),XC(1,1))
```

```
      ISMASH=0
      NUMNP=NODES
      MM=MBAND
      NN=54
      NL=NN+1
      NH=NN+NN
      REWIND 1
      REWIND 2
      NB=0
      DO 10 I=1,6
      DO 10 J=1,6
   10 XBCT(J,I)=0.0
      GO TO 60
C
C         REDUCE EQUATIONS BY BLOCKS
C
C         SHIFT BLOCK OF EQUATIONS
C
   20 NB=NB+1
      DO 30 N=1,NN
      NM=NN+N
      B(N)=B(NM)
      B(NM)=0.
      DO 30 M=1,MM
      A(N,M)=A(NM,M)
   30 A(NM,M)=0.
      IF (NST.EQ.0) GO TO 50
      DO 40 N=1,NN
      NM=NN+N
      DO 40 M=1,NST
      C(N,M)=C(NM,M)
   40 C(NM,M)=0.
   50 CONTINUE
C
C         READ NEXT BLOCK OF EQUATIONS INTO CORE
C
      IF (NUMBLK.EQ.NB) GO TO 70
   60 READ (2) (B(N),(A(N,M),M=1,MM),N=NL,NH)
      IF (NST.GT.0) READ (2) ((C(N,L),L=1,NST),N=NL,NH)
      IF (NB.EQ.0) GO TO 20
C
C         REDUCE BLOCK OF EQUATIONS
C
   70 DO 170 N=1,NN
      IF (A(N,1).EQ.0.) GO TO 170
      NEQ=N+(NB-1)*NN
      IF (ISMASH.EQ.0.AND.A(N,1).LE.1.0E-30) ISMASH=NEQ
      IF (A(N,1).GT.0.) GO TO 80
      WRITE (6,360) NEQ
   80 B(N)=B(N)/A(N,1)
      DO 120 L=2,MM
      IF (A(N,L).EQ.0.) GO TO 120
      DA=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 90 K=L,MM
      J=J+1
   90 A(I,J)=A(I,J)-DA*A(N,K)
      IF (NST.EQ.0) GO TO 110
      DO 100 M=1,NST
      IF (C(N,M).EQ.0.) GO TO 100
      C(I,M)=C(I,M)-DA*C(N,M)
  100 CONTINUE
  110 CONTINUE
      B(I)=B(I)-A(N,L)*B(N)
      A(N,L)=DA
```

```
      A(N,L)=BA
  120 CONTINUE
      IF (NST.EQ.0) GO TO 160
      DO 140 M=1,NST
      DO 130 MN=M,NST
  130 XCT(M,MN)=XCT(M,MN)-C(N,M)*C(N,MN)/A(N,1)
  140 XBT(M)=XBT(M)-B(N)*C(N,M)
      DO 150 M=1,NST
      IF (C(N,M).EQ.0.0) GO TO 150
      C(N,M)=C(N,M)/A(N,1)
  150 CONTINUE
  160 CONTINUE
  170 CONTINUE
C
C        WRITE BLOCK OF REDUCED EQUATIONS ON FILE1
C
      IF (NUMBLK.EQ.NB) GO TO 180
      WRITE (1) (B(N),(A(N,M),M=2,MM),N=1,NN)
      IF (NST.GT.0) WRITE (1) ((C(N,L),L=1,NST),N=1,NN)
      GO TO 20
C
C        BACK-SUBSTITUTION
C
  180 CONTINUE
      IF (NST.EQ.0) GO TO 200
      DO 190 I=1,NST
      DO 190 J=I,NST
      J1=J-I+1
  190 XBCT(I,J1)=XCT(I,J)
      CALL SYMSOL (XBT,XBCT,NST,NST)
  200 DO 240 M=1,NN
      N=NN+1-M
      DO 210 K=2,MM
      L=N+K-1
  210 B(N)=B(N)-A(N,K)*B(L)
      IF (NST.EQ.0) GO TO 230
      DO 220 I=1,NST
  220 B(N)=B(N)-C(N,I)*XBT(I)
  230 CONTINUE
      NM=N+NN
      B(NM)=B(N)
  240 A(NM,NB)=B(N)
      NB=NB-1
      IF (NB.EQ.0) GO TO 250
      BACKSPACE 1
      IF (NST.GT.0) BACKSPACE 1
      READ (1) (B(N),(A(N,M),M=2,MM),N=1,NN)
      IF (NST.GT.0) READ (1) ((C(N,L),L=1,NST),N=1,NN)
      BACKSPACE 1
      IF (NST.GT.0) BACKSPACE 1
      GO TO 200
C
C        ORDER FORMER UNKNOWNS IN B ARRAY
C
  250 K=0
      DO 260 NB=1,NUMBLK
      DO 260 N=1,NN
      NM=N+NN
      K=K+1
  260 B(K)=A(NM,NB)
C
C        WRITE SOLUTION
C
      MPRINT=0
      K=0
      DO 280 N=1,NUMNP
      K=K+1
```

```
      K=K+1
      IF (MPRINT.NE.0) GO TO 270
      IF (NUMNP.LT.K+50.AND.K.GT.1) GO TO 290
      IF (NUMNP.GT.K+49.AND.K.GT.1) K=K+50
      WRITE (6,370) HED
      WRITE (6,380)
      MPRINT=50
  270 MPRINT=MPRINT-1
      NN=K+50
      IF (NUMNP.LT.NN) NN=NUMNP
      IF (K.GT.NUMNP) GO TO 290
  280 WRITE (6,390) (I,B(2*I-1),B(2*I),I=K,NN,50)
  290 CONTINUE
      IF (NST.EQ.0) GO TO 330
C
C         CALCULATE AND OUTPUT SINGULAR INTENSITIES
C
      WRITE (6,340) HED
      K=0
      DO 320 I=1,NST,2
      K=K+1
C
C         SINP=XBT*(D(4,4,ISMAT)*2.0*SQRT(3.14159)/SQRT(2.0))
C
C       FOR OTRHOTROPIC
C       SINP=XBT(SQRT(3.14159)/SQRT(2)
C
      I1=ISMAT(I)
      I2=ISMAT(I+1)
      IF (KORTSW.EQ.0) GO TO 300
      SINP1=XBT(I)*1.253314
      SINP2=XBT(I+1)*1.253314
      GO TO 310
  300 CONTINUE
C       SINP1=XBT(I)*D(4,4,I1)*2.506628274631
C       SINP2=XBT(I+1)*D(4,4,I2)*2.506628274631
      GAVG=.5*(D(4,4,I1)+D(4,4,I2))
      SINP1=XBT(I)*GAVG*2.506628274631
      SINP2=XBT(I+1)*GAVG*2.506628274631
  310 CONTINUE
      WRITE (6,350) K,SINP1,SINP2
  320 CONTINUE
  330 CONTINUE
      RETURN
C
  340 FORMAT (1H1,8X,8A10///30X,26HINTENSITY OF SINGULARITIES//10X,14HSI
     1NGULAR POINT,14X,3HK I,23X,4HK II/)
  350 FORMAT (I17,2E27.5)
  360 FORMAT (34H NEGATIVE DIAGONAL AT EQUATION NO.,I5)
  370 FORMAT (1H1,30X,8A10//53X,13HDISPLACEMENTS)
  380 FORMAT (1H0,5X,2HNP,9X,2HUR,14X,2HUZ,12X,2HNP,9X,2HUR,14X,2HUZ/)
  390 FORMAT (1X,2(2X,I5,2E16.6,2X))
      END
      SUBROUTINE SYMSOL (B,A,NN,MM)
C
C         THIS SUBROUTINE SOLVES A SET OF BANDED EQUATIONS.
C         THIS SUBROUTINE IS CALLED BY SOLV.
C
      DIMENSION A(6,6), B(6)
C
C         REDUCE MATRIX
C
      DO 50 N=1,NN
      B(N)=B(N)/A(N,1)
      DO 50 L=2,MM
      IF (A(N,L)) 10,50,10
   10 C=A(N,L)/A(N,1)
```

```
   10 C=A(N,L)/A(N,1)
      I=N+L-1
      IF (NN-I) 40,20,20
   20 J=0
      DO 30 K=L,MM
      J=J+1
   30 A(I,J)=A(I,J)-C*A(N,K)
      B(I)=B(I)-A(N,L)*B(N)
   40 A(N,L)=C
   50 CONTINUE
C
C         BACK SUBSTITUTION
C
      N=NN
   60 N=N-1
      IF (N) 70,100,70
   70 DO 90 K=2,MM
      L=N+K-1
      IF (NN-L) 90,80,80
   80 B(N)=B(N)-A(N,K)*B(L)
   90 CONTINUE
      GO TO 60
  100 RETURN
      END
```

APPENDIX C

SAMPLE QMESH INPUT FILE

This is a sample QMESH input file for the S-Glass/Pin specimen. It defines node point coordinates, boundary codes, line strings, regions, and mesh renumbering.

```
COMMEN      S GLASS-PIN FAILURE
POINT           1/   0.0000     0.0000                 100
POINT           2/   0.2500     0.0000                   0
POINT           3/   0.2500     1.0000                   0
POINT           4/   0.0000     1.0000                   0
POINT           5/   3.0000     0.0000                   0
POINT           6/   4.2500     0.0000                   0
POINT           7/   4.2500     3.0000                   0
POINT           8/   4.2500     5.0000                   0
POINT           9/   3.0000     5.0000                   0
POINT          10/   3.0000     1.0000                   0
POINT          11/   8.0000     3.0000                   0
POINT          12/   8.0000     5.0000                   0
POINT          13/   3.0000     3.0000                   0
LINE   STR      1/   1      2    0     1     1.0000      0
LINE   STR      2/   2      3    0     4     1.0000      0
LINE   STR      3/   3      4    0     1     1.0000    300
LINE   STR      4/   4      1    0     4     1.0000    200
LINE   STR      5/   2      5    0    11     1.0000      0
LINE   STR      6/   5     10    0     4     1.0000      0
LINE   STR      7/  10      3    0    11     1.0000    400
LINE   STR      8/   5      6    0     5     1.0000      0
LINE   STR      9/   6      7    0    12     1.0000      0
LINE   STR     10/   7     13    0     5     1.0000      0
LINE   STR     11/  13     10    0     8     1.0000    500
LINE   STR     12/   7      8    0     8     1.0000      0
LINE   STR     13/   8      9    0     5     1.0000      0
LINE   STR     14/   9     13    0     8     1.0000    600
LINE   STR     15/   7     11    0    15     1.0000    700
LINE   STR     16/  11     12    0     8     1.0000      0
LINE   STR     17/  12      8    0    15     1.0000      0
REGION    1     1/  -1     -2   -3    -4
REGION    2     2/  -5     -6   -7    -2
REGION    3     3/  -8     -9  -10   -11     -6
REGION    3     4/ -12    -13  -14   -10
REGION    3     5/ -15    -16  -17   -12
SCHEME          1M
SCHEME          2M
SCHEME          3M
SCHEME          4M
SCHEME          5M
BODY             /   1      2    3     4     5
END             4
P-L-           12   17      8   13     9
END
```

APPENDIX D

SAMPLE CHILES.BYU INTERACTIVE SESSION

This is a sample interactive session of CHILES.BYU. The prompts and answers that follow are what are displayed on the terminal screen when running the program. These input values correspond to the analysis of the S-Glass/Pin problem of Chapter 4. The QMESH input file for this problem was created from the data given in Appendix C.

```
<ENTER TITLE OF PROBLEM>  S-Glass/Pin

<ENTER NUMBER OF SINGULAR POINTS (3 MAX.)>  0

<ENTER 1,2, OR 3 FOR AXISYMMETRIC, PLANE STRESS,
OR PLANE STRAIN GEOMETRY>  1

<ENTER NUMBER OF MATERIALS (10 MAX.)>  3

<ENTER 0 FOR REDUCED OUTPUT OR 1 FOR EXTENDED
OUTPUT>  1

<ENTER 0 FOR NO BOUNDARY CONDITIONS ON INTENSITIES
OR 1 TO ALLOW INTENSITY BOUNDARY CONDITION
TO BE READ>  0

<ENTER 0 TO READ QMESH FILE OR 1 TO INPUT GEOMETRY
POINT BY POINT>  0

<ENTER NUMBER OF BOUNDARY FLAGS AS SET IN QMESH>  6

<ENTER 0 FOR ISOTROPIC MATERIAL OR 1 FOR ORTHO-
TROPIC MATERIAL>  0

DO YOU WISH TO CHANGE ANY OF THE ABOVE ENTRIES?  No

<ENTER YOUNGS MODULUS FOR MATERIAL  1>  29.8E+6

<ENTER POISSONS RATIO FOR MATERIAL  1>  0.3

<ENTER COEF. OF THER. EXPANSION FOR MATERIAL  1>  0

<ENTER YOUNGS MODULUS FOR MATERIAL  2>  13.0E+6

<ENTER POISSONS RATIO FOR MATERIAL  2>  0.2

<ENTER COEF. OF THER. EXPANSION FOR MATERIAL  2>  0

<ENTER YOUNGS MODULUS FOR MATERIAL  3>  29.0E+6
```

```
<ENTER POISSONS RATIO FOR MATERIAL  3>  .294

<ENTER COEF. OF THER. EXPANSION FOR MATERIAL  3>  0

DO YOU WISH TO CHANGE ANY MATERIAL PROPERTY
ENTRIES?  No

<SPECIFY QMESH FILE>  PIN9.DAT

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  200

<ENTER BOUNDARY CODE>  1.0

<ENTER XR OR PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  300

<ENTER BOUNDARY CODE>  -1

<ENTER XR OF PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>  50000.

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>  50000.

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  400

<ENTER BOUNDARY CODE>  -1

<ENTER XR OR PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>  50000.

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>  50000.

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  500

<ENTER BOUNDARY CODE>  -1

<ENTER XR OF PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>  50000.

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>  50000.

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  600
```

```
<ENTER BOUNDARY CODE>  -1

<ENTER XR OF PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>  50000.

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>  50000.

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>  700

<ENTER BOUNDARY CODE>  2.0

<ENTER XR OF PN VALUE OF DISPLACEMENT, FORCE OR
NORMAL TRACTION>

<ENTER XZ OR SH VALUE OF DISPLACEMENT, FORCE OR
SHEAR TRACTION>

DO YOU WISH TO MAKE ANY CHANGES?  No

DO YOU WISH TO DEFINE A SINGULAR POINT?  No

DO YOU WISH TO DEFINE A BOUNDARY CODE FOR A NODE? No


            <<CHILES IS EXECUTING>>


        EXECUTION COMPLETED;

             YOUR PRINTED OUTPUT FILE IS CHILES.LIS
             YOUR PLOT DATA FILE IS CHILES.MOV


        SEE YOU AGAIN NEXT TIME


        YOUR USER FRIENDLY FINITE ELEMENT PROGRAM


             SIGNED,

                CHILES
```

DEVELOPMENT OF A COMPUTER-AIDED ANALYSIS PACKAGE FOR LINEAR ELASTIC FRACTURE MECHANICS

William E. Warkentin

Department of Civil Engineering

M.S. Degree, April 1983

ABSTRACT

This thesis develps an interactive computer-aided analysis package for solving linear elastic fracture mechanics problems. CHILES 2 (an existing finite element fracture mechanics program) was modified to be interactive. The modified program is named CHILES.BYU and is used in conjunction with QMESH.BYU and MOVIE.BYU to form the total software package. An analysis of glass ceramics and glass ceramic/metal interfaces was performed demonstrating the capabilities of the package.

This study demonstrates the usefulness of an interactive package of computer programs to allow an engineer to rapidly investigate different crack configurations.

COMMITTEE APPROVAL:

Steven E Benzley

Steven E. Benzley
Committee Chairman

Olani Durrant

S. Olani Durrant
Committee Memeber

H. N. Christiansen

Henry N. Christiansen
Department Chairman