UNCLASSIFIED

AD NUMBER

ADB108123

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies only; Critical Technology; Mar 1986. Other requests shall be referred to Air Force Wright Aeronautical Laboratories, Attn: FIBRA, Wright-Patterson AFB, OH 45433.

AUTHORITY

WL/DOOS ltr, 24 Apr 1992

THIS PAGE IS UNCLASSIFIED

(2)

AD-B108 123

AFWAL-TR-86-3035

DESIGN GUIDE FOR BOLTED JOINTS IN COMPOSITE STRUCTURES

R. L. RAMKUMAR
E. S. SAETHER
D. CHENG

Northrop Corporation, Aircraft Division
One Northrop Avenue
Hawthorne, California 90250

March 1986

Final Report for Period January 1983 to November 1985

Distribution limited to U.S. government agencies only, test and evaluation, March 1986. Other requests for this document must be referred to AFWAL/FIBRA, WPAFB, OHIO 45433

WARNING—This document contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or Executive Order 12470. Violation of these export laws is subject to severe criminal penalties.

DESTRUCTION NOTICE - Destroy by any method that will prevent disclosure of contents or reconstruction of the document.

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

87 1 13 041

DISCLAIMER NOTICE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADB108123L

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: T/E
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Distribution limited to U.S. Government Agencies only; ~~critical technology~~; 3.86. Other requests must be referred to AFWAL/FIBRA, WPAFB, OH 45433
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): NOR 86-211	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFWAL-TR-86-3035
6a. NAME OF PERFORMING ORGANIZATION: Northrop Corporation, Aircraft Division; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Flight Dynamics Laboratory (AFWAL/FIBRA) Air Force Wright Aeronautical Laboratories
6c. ADDRESS (City, State and ZIP Code): One Northrop Avenue, Hawthorne, CA 90250	7b. ADDRESS (City, State and ZIP Code): Wright-Patterson Air Force Base, Dayton, OH 45433-6553
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: Contract F33615-82-C-3217
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
62201F	2401	02	55

11. TITLE (Include Security Classification): Design Guide for Bolted Joints in Composite Structures

12. PERSONAL AUTHOR(S): R. L. Ramkumar, E. S. Saether and D. Chang

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 1/1/83 TO 11/30/85	March 1986	264

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS
FIELD	GROUP	SUB. GR.	Bolted joints, composite structures, design guidelines, analytical design tools ~~(SASCJ and SAMCJ)~~, user instructions, samples, program listing.
01	03		
13	05		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A design guide was developed for bolted composite structural joints. The guide includes general design guidelines for the various joint parameters, an analytical design methodology, a description of the analytical design tools, an illustration of the use of corresponding computer codes (SASCJ and SAMCJ), and a listing of the computer codes. The proposed design procedure is purely analytical, and enables the user to rapidly evaluate many different joint concepts for a selected application. When the bolted structure is fabricated using existing (fully characterized) materials, the design requires no complementary test results. Presented analytical design tools are currently restricted to primarily uniaxially loaded joints and fastener arrangements that are currently used in aircraft structures. Also, the bolted joint is assumed to be strength-critical. However, sample fatigue test results are presented to illustrate a durability check on the joint, assuming a simplified fatigue analysis and assuming that fatigue failure is induced by excessive hole elongation. Despite its current restrictions, this guide is the first government document that provides guidance and analytical tools for the design of bolted composite structures.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: V. B. Venkayya	22b. TELEPHONE NUMBER (Include Area Code): (513) 255-6992; 22c. OFFICE SYMBOL: AFWAL/FIBRA

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

PREFACE

This report was prepared under Contract F33615-82-C-3217, titled "Bolted Joints in Composite Structures: Design, Analysis and Verification," and administered by the Air Force Wright Aeronautical Laboratories. Dr. V. B. Venkayya was the Air Force project engineer, and was assisted by Capt. M. Sobota and Lt. D. L. Graves as program co-monitors. The program manager and principal investigator at Northrop was Dr. R. L. Ramkumar.

This report is a guide for the design of bolted joints in composite structures, and was prepared under Task 4 in the referenced program (Project 2401).

The authors gratefully acknowledge the support provided by the following personnel:

Graphics	R. Cordero
Typing	C. Harris

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

SECTION 1

INTRODUCTION

Bolted joints are a prime means of load transfer between structural parts in aircraft. Compared to other joining methods (bonding, welding, etc.), mechanical fastening is more reliable, with a potential for improved structural efficiency, maintainability and cost effectiveness. However, bolted joints are a source of stress concentration and could precipitate structural failures if they are designed improperly.

Prior to the initiation of this Northrop/AFWAL program, no analysis was available to be used as an exclusive design tool for bolted parts, especially if they were laminated composites. Consequently, their design has hitherto been based on extensive testing, empirical data and approximate analyses. The analysis developed in this Northrop/AFWAL program eliminates the need for extensive testing and provides a tool for the rapid evaluation of a bolted joint concept. If the structural part is to be fabricated using a characterized material, it eliminates the need for experimental information.

In the following sub-sections, the scope of this design guide is stated, sample bolted concepts are presented, criteria for the design of bolted joints in composite structures are discussed, the proposed design procedure is described, the analytical and experimental requirements for the design procedure are outlined, and its current restrictions are mentioned. In Section 2, general guidelines for the design of a bolted joint in composite structures are presented, along with summary statements on the effects of critical joint parameters. Section 3 presents the computer codes developed in this program for the strength analyses of single and multiple fastener joints in composites (SASCJ and SAMCJ, respectively). Section 4 demonstrates the use of the developed analysis in predicting the strength of a realistic structural element.

1.1 Scope of the Design Guide

This design guide summarizes the effects of critical parameters on the strength and lifetime of bolted joints in composite structures, and presents general design guidelines. It also describes a test-independent analytical procedure for the strength evaluation of a bolted concept, based on the analyses developed in this program. The reader is familarized with the computer codes (SASCJ and SAMCJ) that perform these analyses, and an application to a realistic structural bolted joint is demonstrated. This design guide will enable one to perform a rapid analytical evaluation of many joint configurations, and to select an efficient bolting concept. The described computer codes are currently restricted to uniaxial loading, conventionally used fastener spacing and protruding head fasteners.

1.2 Sample Joint Configurations

Figure 1 presents six composite-to-metal bolted joint configurations used in the F/A-18A aircraft wing (Reference 1). Figures 2 and 3 present joint configurations used in a typical fuselage structure (Reference 2). A skin-to-root fitting bolted joint in the F-20 horizontal stabilizer is shown in Figure 4. Many bolted joint concepts have been studied recently as potential alternative joining concepts for the F/A-18A wing root section and the F/A-18A vertical tail root section (Figures 5 and 6, respectively). The sample bolted configurations in Figures 1 to 6 illustrate the possible variety in this joining concept.

1.3 Overview of Design Methodology

There are many variables in the design of a bolted joint in composite structures. These include the geometry and the

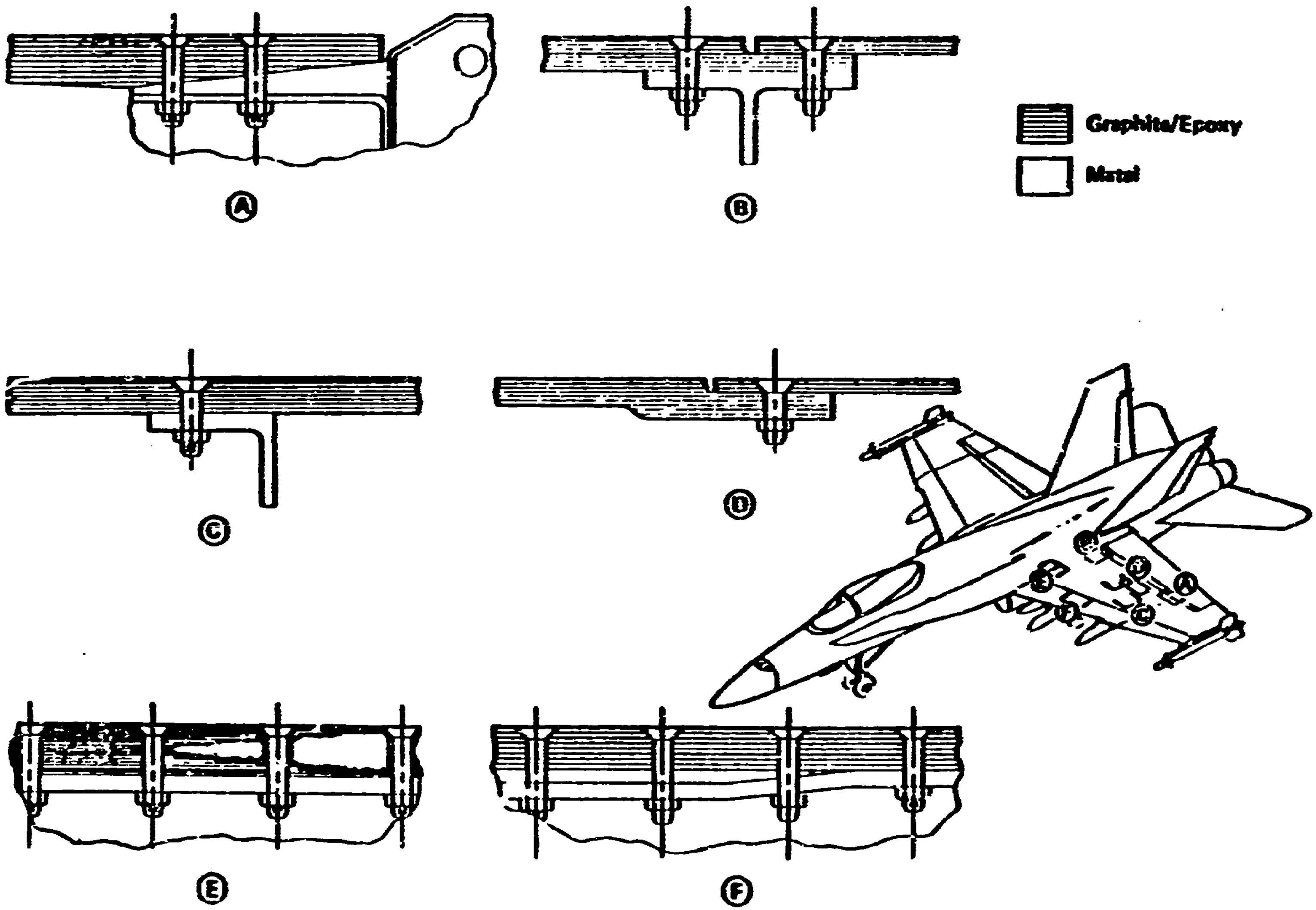


Figure 1. Sample Bolted Joints in the F/A-18A Aircraft Wing (Reference 1).

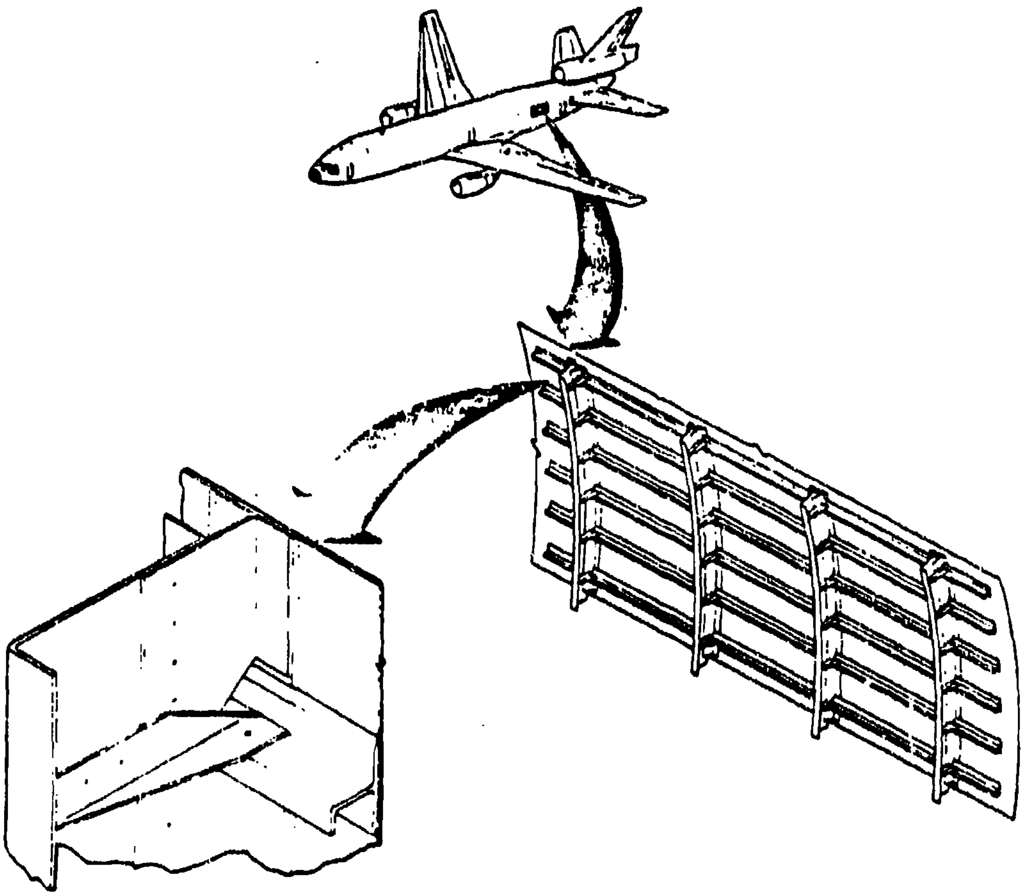


Figure 2. A Bolted Joint Concept for a Fuselage Structure (Reference 2).

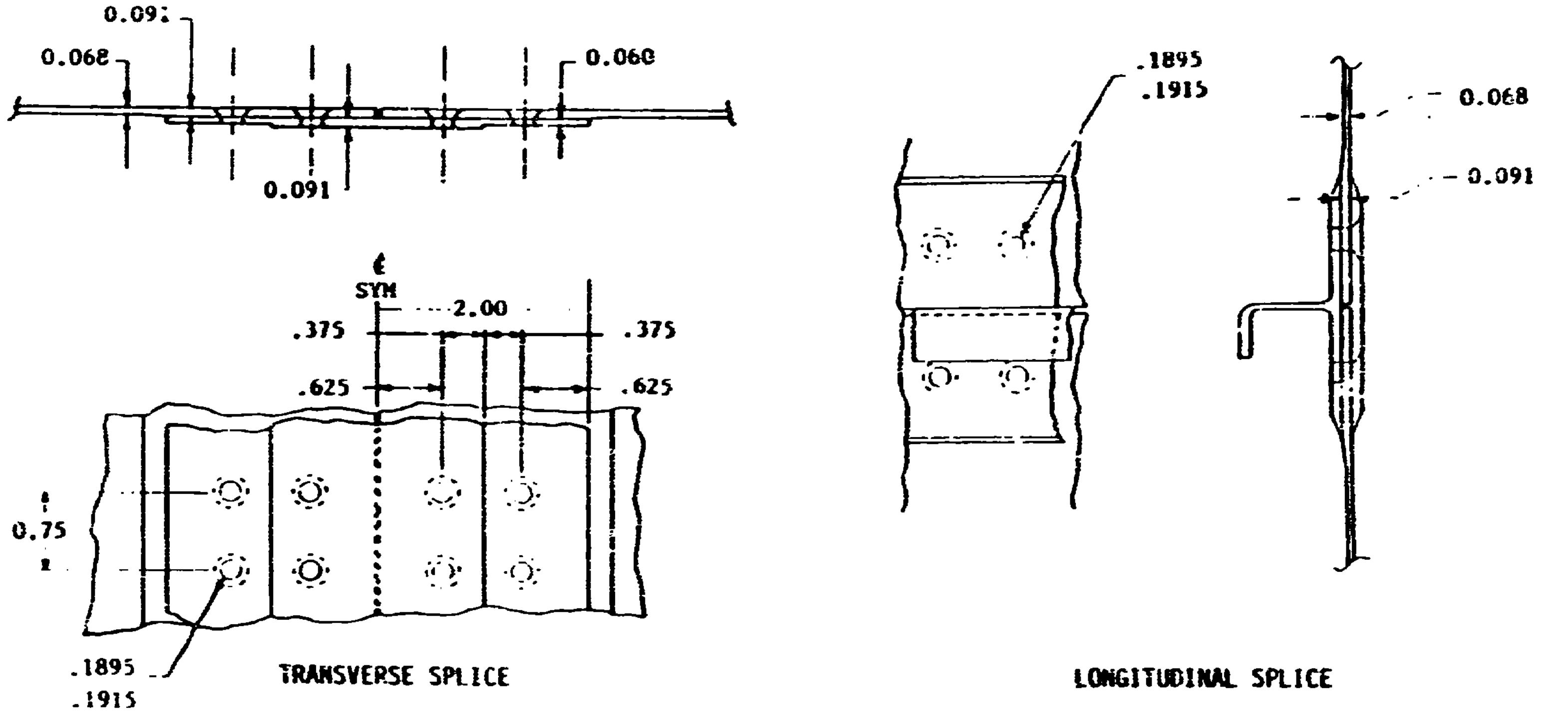


Figure 3. Bolted Joint Concepts for Composite Fuselage Structures (Reference 2).

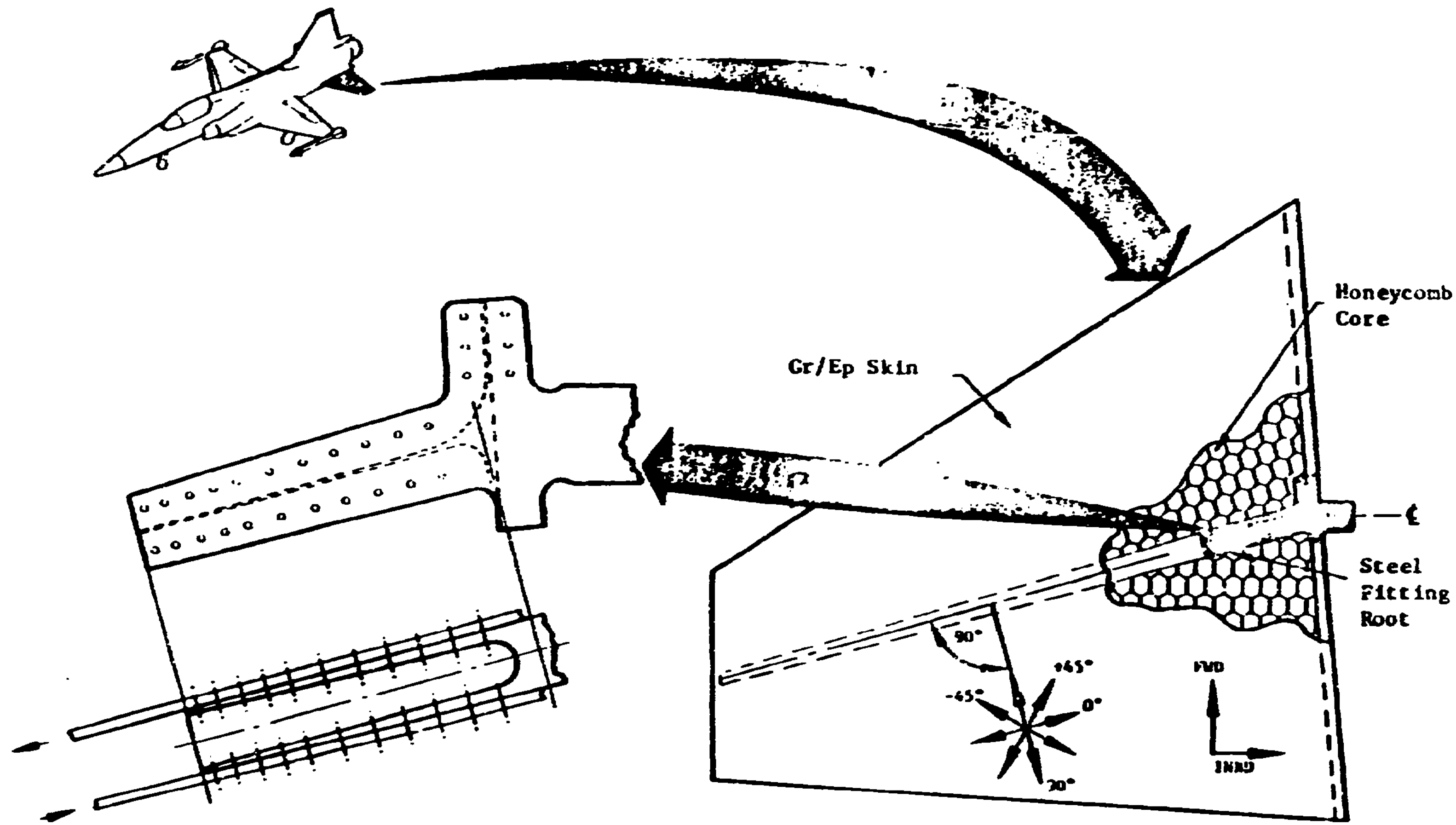


Figure 4. Skin-to-Root Fitting Joint in the F-20 Horizontal Stabilizer.

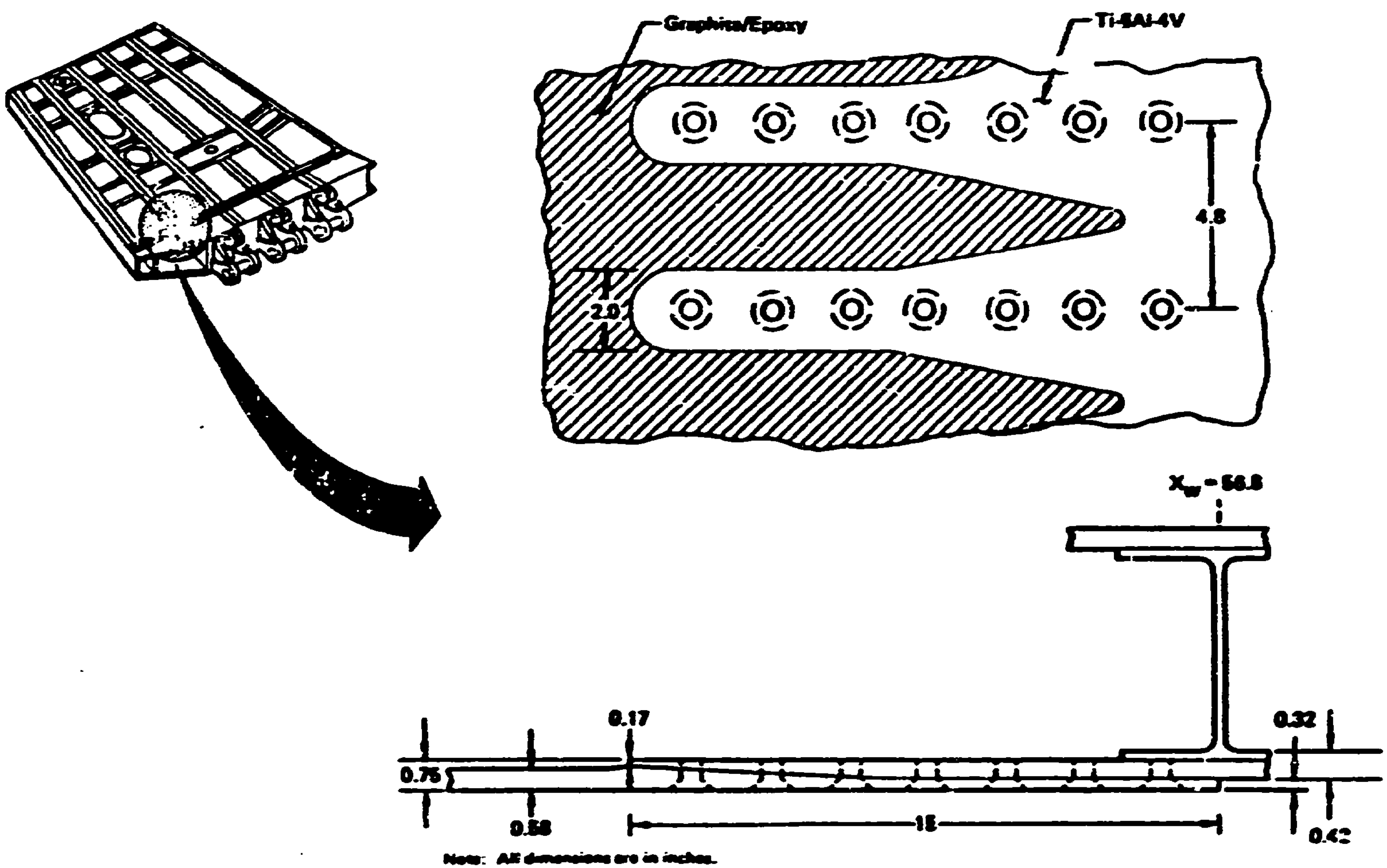


Figure 5 . Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section (Reference 1).

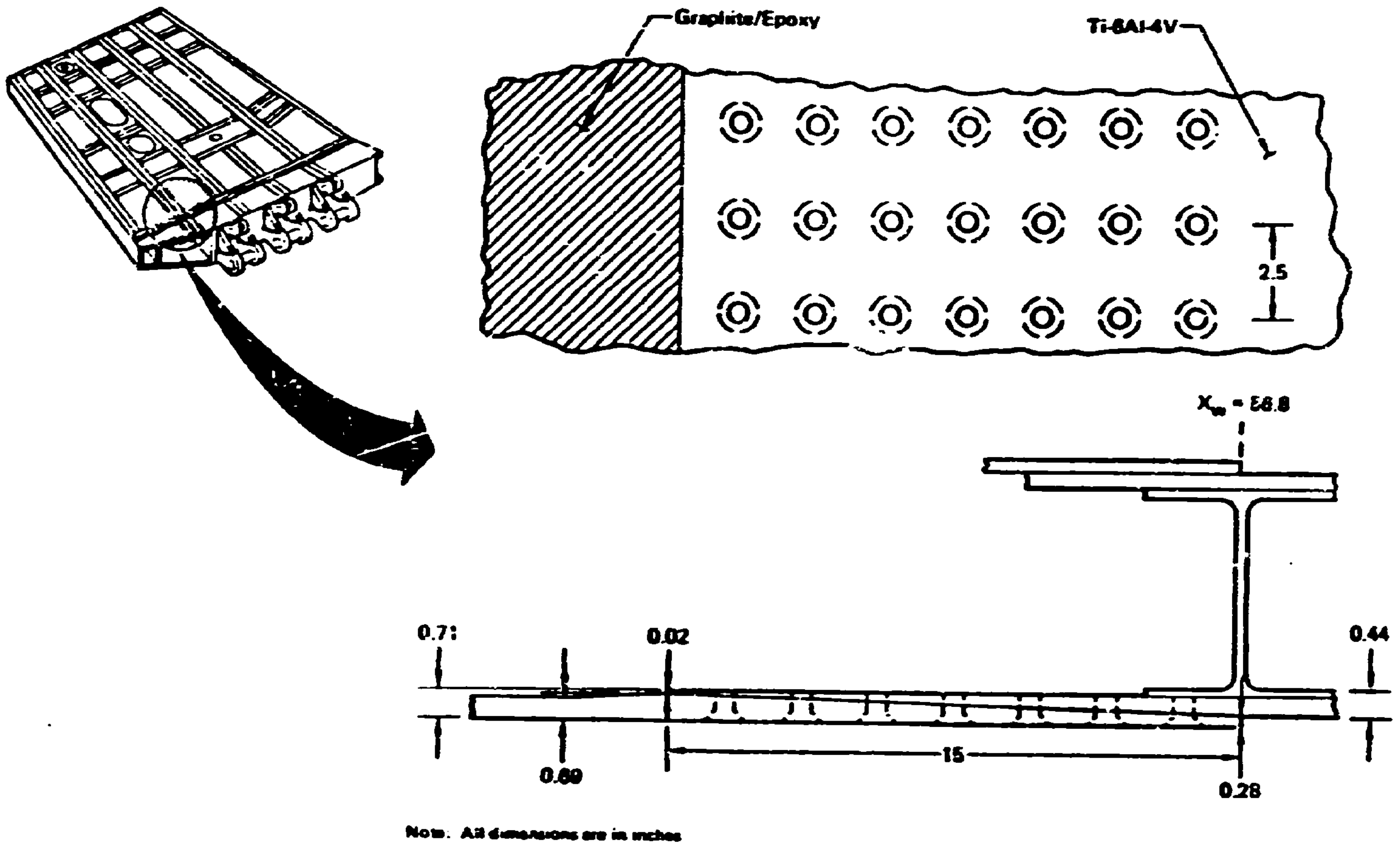


Figure 5 . Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section. (Reference 1; Concluded).

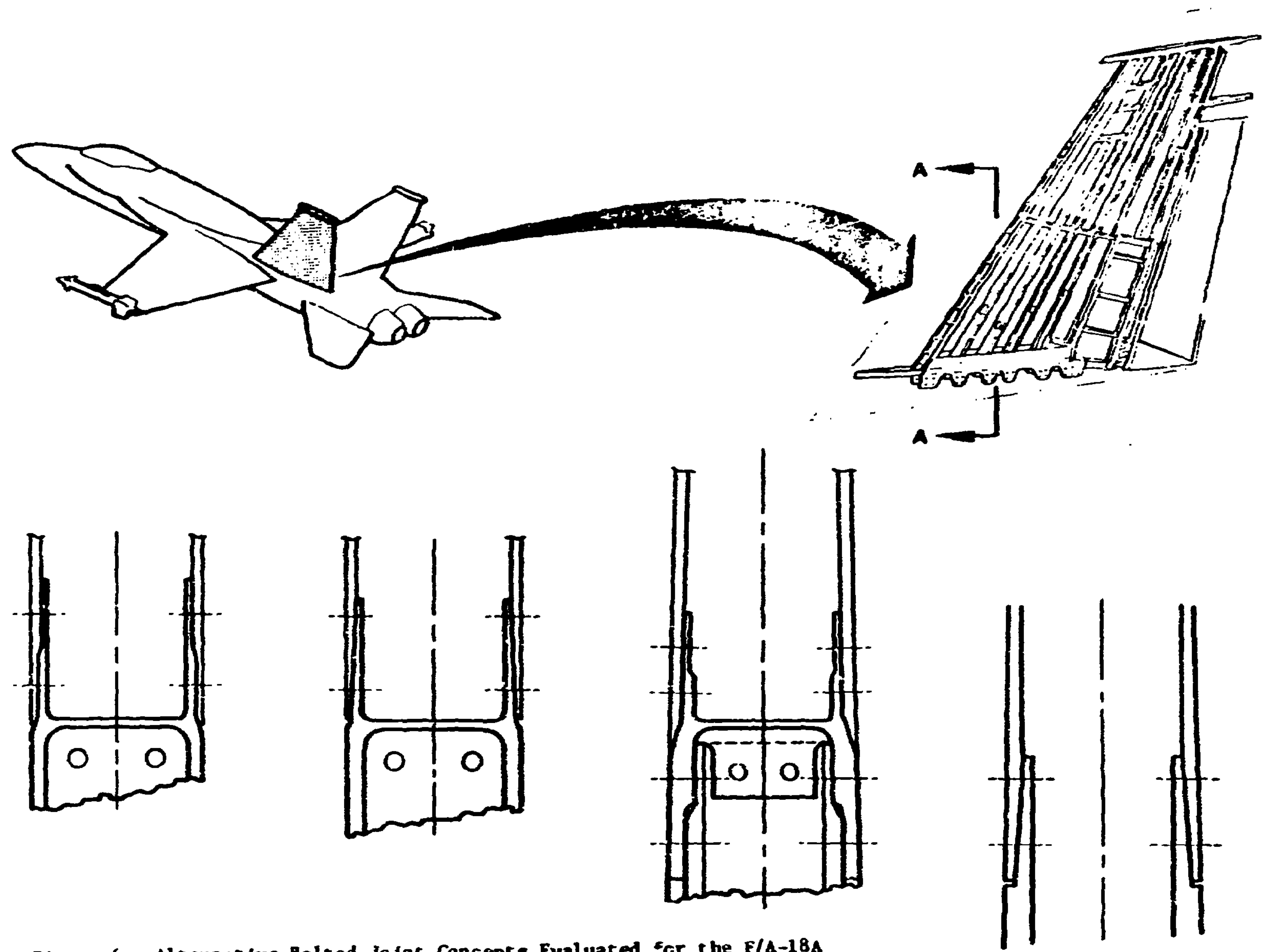


Figure 6. Alternative Bolted Joint Concepts Evaluated for the F/A-18A Vertical Tail Root Section (References 3 to 5).

material properties of the bolted parts, the size and arrangement of the fasteners, the fastener material properties and torque, applied loading and the load transfer mechanism (single versus double shear), etc. The design of a bolted joint involves a parametric study of the effects of the above variables on the joint efficiency, for a specified loading condition. A preliminary analysis of a structural component, based on conventional assumptions, yields the general biaxial loading transferred at the joint location (see Figure 7). The design procedure recommended in this guide assumes a predominantly uniaxial loading at the joint location.

The design of a uniaxially loaded joint in composite structures may be performed using the analyses developed in this Northrop/AFWAL program. Section 3 describes the use of the SASCJ and the SAMCJ computer codes for the strength prediction of single and multiple fastener joints in composites, respectively. The SASCJ code predicts the strength of joints when a single fastener transfers the applied load between the bolted plates. This analysis accounts for material nonlinearity in the bolted plates, the non-uniform fastener load distribution in the thickness direction of the bolted plates, and the progression of ply-level failures based on a choice among a few failure criteria. The SAMCJ code predicts the strength of plates bolted together by one or many fasteners. It computes the magnitude and orientation of the load at every fastener location, the applied load level for averaged stress components to reach critical levels at fastener and cut-out locations, the failure value of the applied load, the failure location and the failure mode (net section, shear-out or bearing). Failure predictions are made at the laminate level using average stress failure criteria.

The proposed design procedure involves the use of the developed analyses to evaluate the effect of joint variables on joint efficiency. If the bolted plates are fabricated using characterized materials, the joint design is tested-independent.

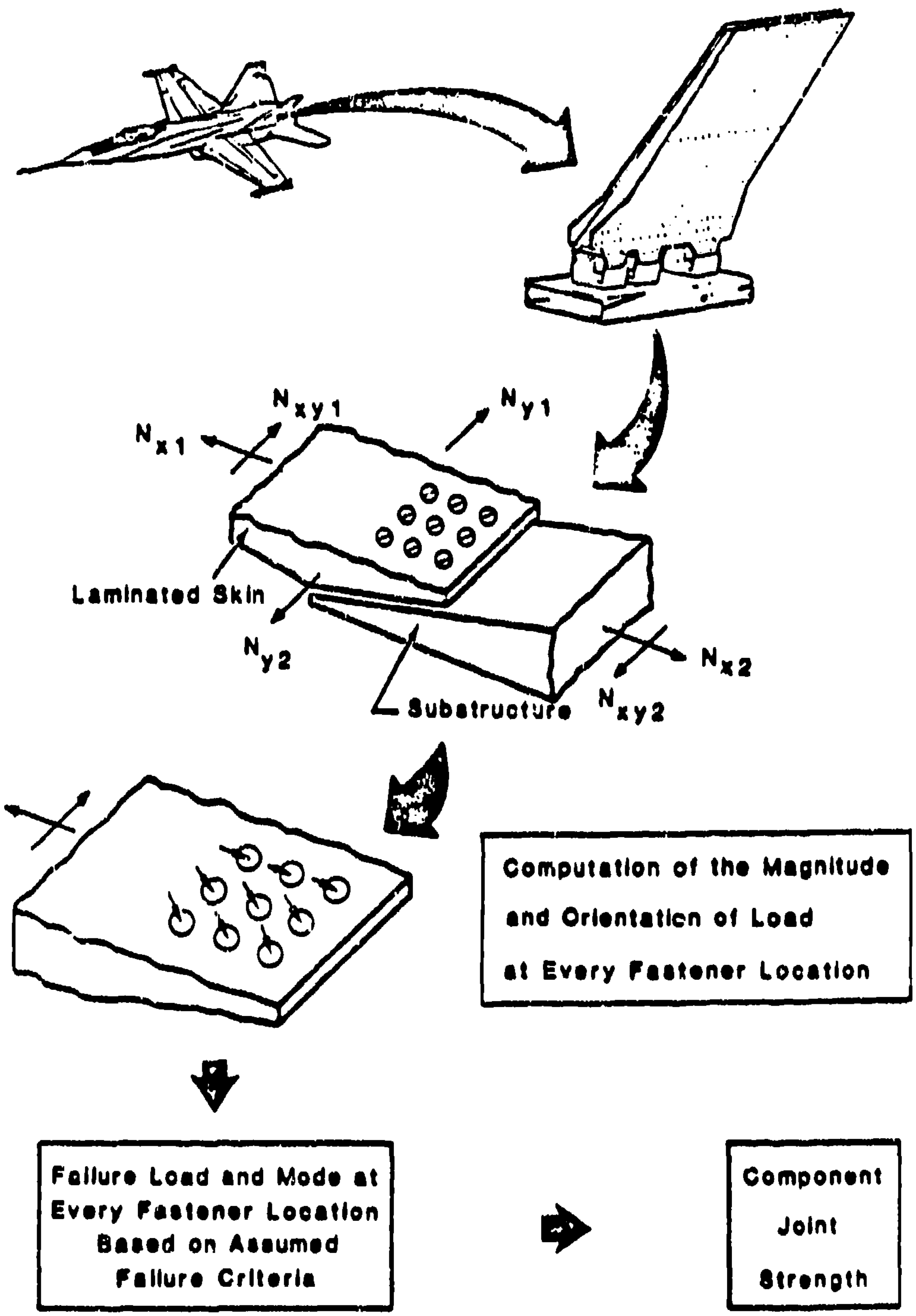


Figure 7. Overview of the Strength Analysis of Bolted Structures.

Candidate bolted joint concepts are selected following the general guidelines outlined in Section 2. The fastener size and arrangement (spacing between fasteners), the geometry of the bolted plates, the load transfer mechanism, etc. are varied without violating the constraints imposed by the structural application. The strength and durability of each bolted joint concept, along with its impact on manufacturing costs and maintenance, are evaluated to establish joint efficiency. An efficient bolted joint concept can thus be designed using a purely analytical tool on a finite number of concepts that are selected in accordance with established guidelines.

1.4 Analytical Requirements

The design of a bolted joint for composite structures requires the analyses developed in this Northrop/AFWAL program (References 6 and 7). The analysis of plates bolted together by a single fastener may be performed using the SASCJ (Strength Analysis of Single Fastener Composite Joints) or the SAMCJ (Strength Analysis of Multiple Fastener Composite Joints) computer code. Plates bolted together by many fasteners are analyzed using SAMCJ computer code. Section 3 presents a brief description of these analyses. The reader is referred to References 6 and 7 for further details.

1.5 Test Requirements

A test-independent, purely analytical design tool has been developed to design a bolted joint for composite structures that are fabricated using characterized materials. The engineering properties (Young's moduli in the fiber and transverse directions, major Poisson's ratio and the shear modulus in the fiber coordinate system), the strengths or failure strains (under tension, compression and shear), and the failure parameters for the assumed failure criteria (characteristic distances for net section, shear-out and bearing failure predictions using the average stress failure criteria, for example) are known for a characterized composite material (lamina). Tests required to obtain the above material properties must be performed on a new (uncharacterized) material system, prior to designing bolted joints for structural parts made from this material. When previously characterized materials are used in the bolted plates, the test requirements are nil for the design of an efficient bolted joint concept.

1.6 Current Restrictions

The design of bolted joints in composite structures is influenced by the current restrictions in the developed analytical tools. The primary restrictions are listed below:

(1) The developed strength analyses (SASCJ and SAMCJ computer codes) do not account for countersunk fastener effects.

(2) SASCJ and SAMCJ contain a stress analysis that approximates the fastener/plate contact problem by an assumed radial stress distribution.

(3) SASCJ AND SAMCJ are restricted to a uniaxial applied loading, in tension or in compression.

(4) The prediction of the durability of a joint is restricted to the incorporation of the bearing stress at critical fastener locations into experimentally obtained curves for joint life.

(5) SAMCJ restricts the user to rectangular element geometries and currently used fastener spacing and arrangement.

Despite the above restrictions, the developed analyses and the proposed design procedure mark a significant improvement over the state-of-the-art with respect to the design and analysis of bolted joints in composite structures.

SECTION 2

GENERAL DESIGN GUIDELINES AND JOINT VARIABLES

The design of boted joints in composite structures involves the definition of many variables. The major design considerations are listed below:

(a) The loads that must be transferred from one part to another.

(b) The load transfer location in the structure.

(c) Geometric constraints, if any, at the load transfer location.

(d) Fastener type, size and arrangement.

(e) The environmental range the joint will be exposed to.

(f) The effect of the joint concept on structural efficiency and reliability.

The following sub-sections discuss the primary variables that influence the design of bolted joints in composite structures. Design guidelines corresponding to the discussed joint parameters are highlighted within the sub-sections.

2.1 Joint Location in the Structure

The location of the joint in a structure influences the selection of the joint variables significantly. Design guidelines pertaining to selected joint locations are presented below:

(a) When aerodynamic surfaces in an aircraft structure are joined to substructural parts, or segments of a surface are joined together, the requirement of a smooth outer moldline should not be violated. The use of protruding head fasteners on such surfaces, or the presence of any other geometric discontinuity (step) at the joint location, will adversely affect the lift distribution on these surfaces and their aerodynamic performance.

On aerodynamic surfaces, fasteners must be installed to be flush with the surface, without exposed fastener heads, and joined members must retain a smooth outer moldline.	(1)

(b) When structural members are joined together in fuel-containment areas, measures must be taken to preclude leakage of the fuel and service-related hazards. The use of metallic fasteners on the outer surface, for instance, introduce the threat of arcing within the fuel cell in the event of a lightning strike. In designing joints for these locations, special consideration must be given to the mentioned sealing requirements.

In fuel containment areas, joints must be sealed to be leak-proof. Fasteners must also be sealed to prevent arcing within the fuel cell in the event of a lightening strike.	(2)

(c) When bolted joints are designed for structural regions with limited or restricted access, special fastener types have to be used.

In areas of restricted accessibility, blind fasteners must be used.	(3)

(d) When a laminated part is bolted to a metallic substructure, the threat of joint corrosion must be considered.

In composite-to-metal joint locations. corrosion barriers like fiberglass layers must be used.	(4)

2.2 Joint Configurations

Selected joint configurations are significantly influenced by their structural locations. Figures 1 to 6 present typical structural joint configurations in current aircraft. Figure 8 presents the localized structural joint configurations along with their equivalent configurations that are analyzed. The configurations that transfer loads in single shear introduce localized bending effects that could adversely affect the strength and durability of the joint. Stepped lap and scarf configurations involve thickness changes that provide an additional design variable (layup) in bolted laminates.

2.3 Joint Loading

Structural joints are designed to be effective over their design lifetime, when subjected to the anticipated design spectrum fatigue loading. The durability considerations for structural joints are discussed in Section 4. This design guide emphasizes the strength analysis of a bolted joint, and presents computer codes that perform it. The reader must supplement the joint design based on a strength analysis with a durability check, using information similar to that presented in Section 2.9. The effect of joint loading is discussed further below, at three levels -- structural, among fastener rows, and at an isolated fastener location.

2.3.1 Joint Loads at the Structural Level

Joint loads at the structural level fall into two basic categories -- inplane loads and out-of-plane or bending loads. Figure 9 presents some possible inplane load conditions in typical wing skin-to-substructure attachments. The analyses developed in this Northrop/AFWAL program, and described in Section 3, assume that the joint at each location is subjected to a predominant unidirectional load. Figure 9 illustrates that this assumption will

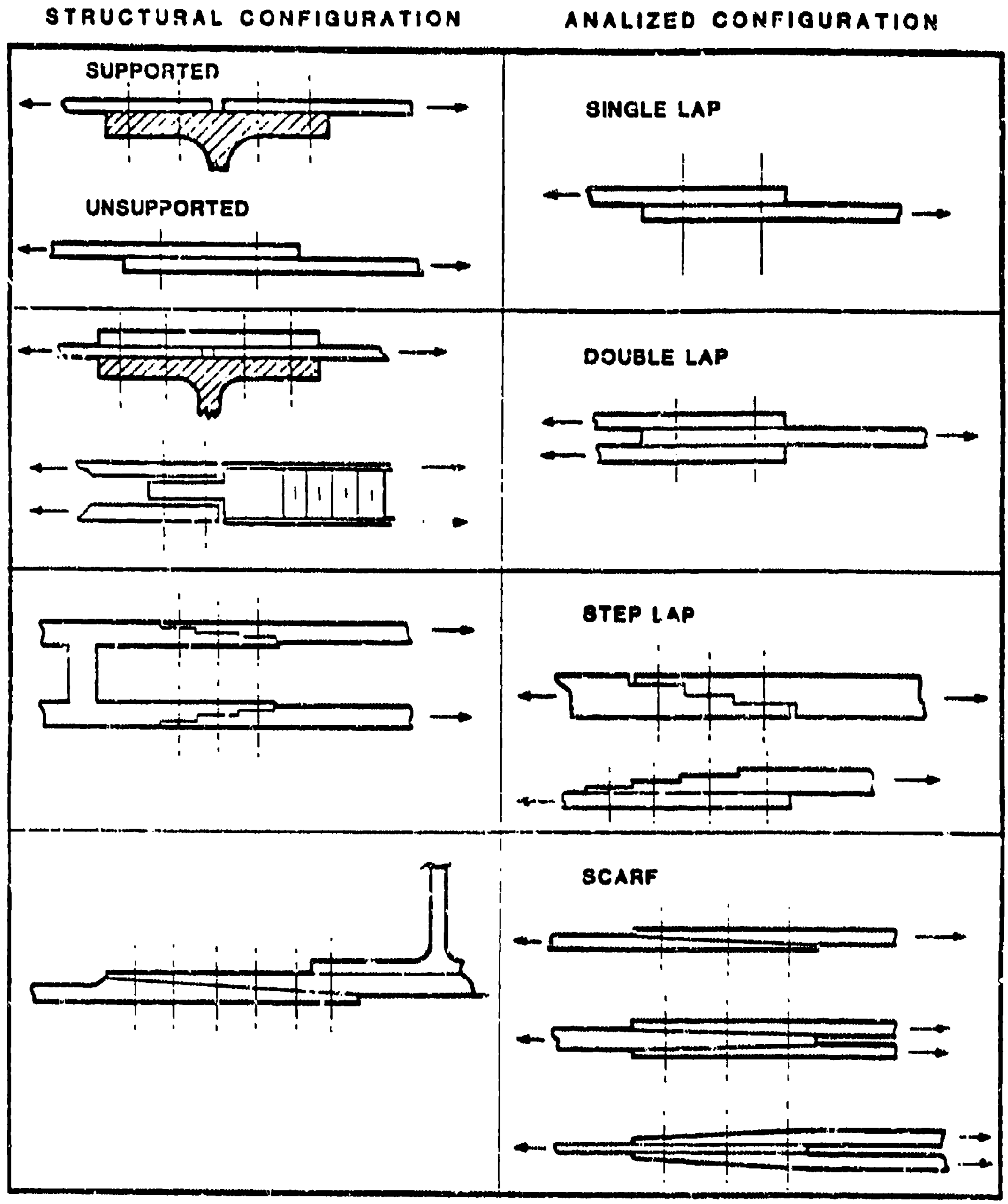
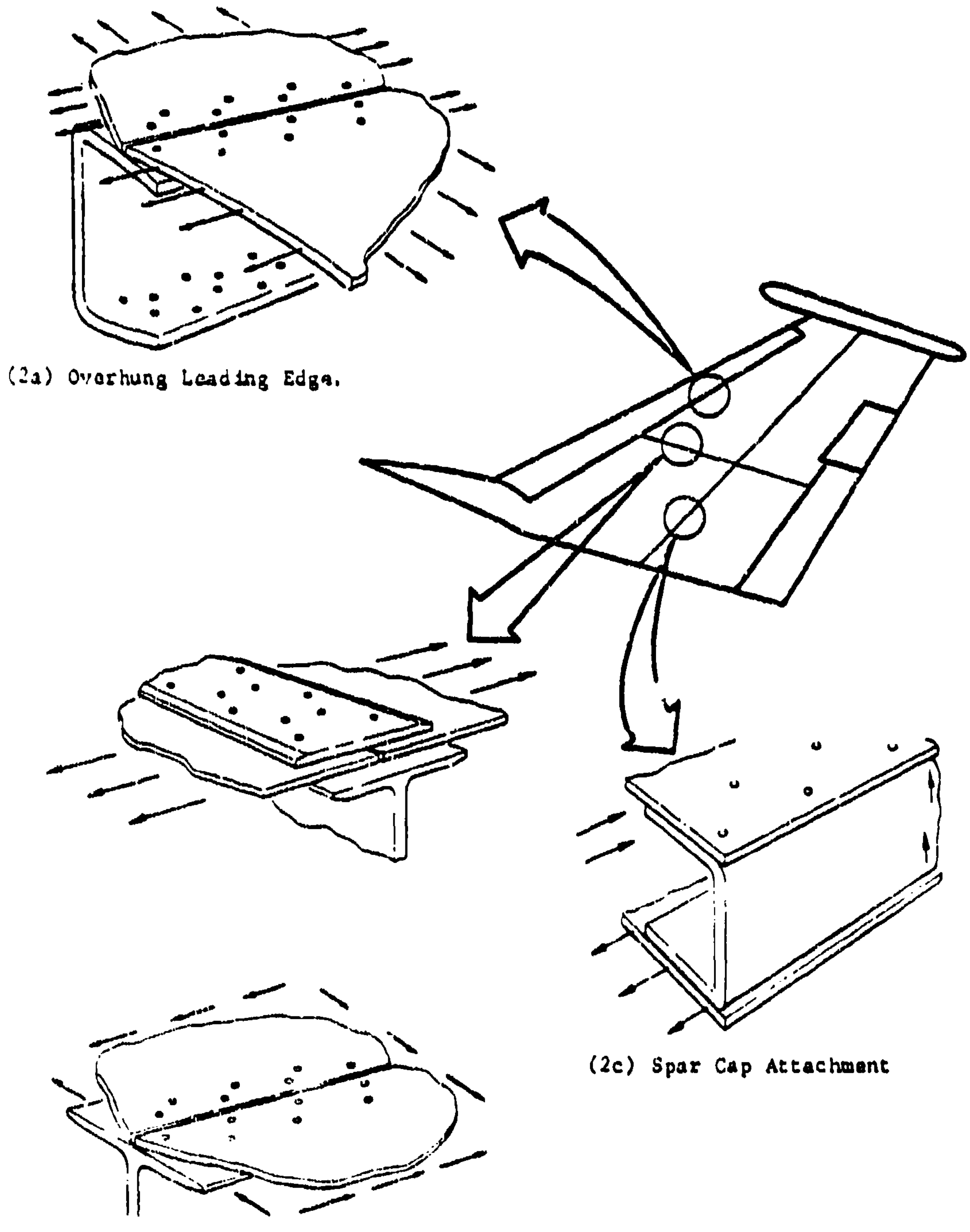


Figure 8. Structural and Analyzed Bolted Joint Configurations.



(2a) Overhung Leading Edge.

(2b) Chordwise Wing Skin Splice.

(2c) Spar Cap Attachment

Figure 9. Inplane Loads in Typical Wing Skin-to-Substructure Attachments.

not be valid at some locations.

Figure 10 presents sample situations where considerable out-of-plane (bending) loads are introduced at the joint location. This is inherent in single shear load transfer configurations (see Figure 8), and adversely affects joint strength and durability. If one of the bolted plates is very stiff compared to the other, the deleterious effects of load eccentricity in a single shear configuration are minimized. In double-shear load transfer configurations (see Figure 8), the out-of-plane loads are reduced to a negligible level.

Single - shear load transfer joint configurations introduce out-of-plane (bending) loads that could significantly reduce the strength of the joint. When one of the bolted members is very stiff, the effect of the out-of-plane loads is minimized.

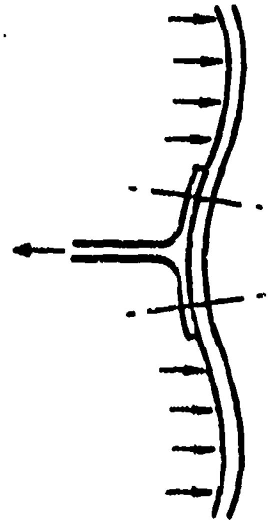
(5)

Double - shear load transfer joint configurations essentially introduce inplane loads in the bolted plates.

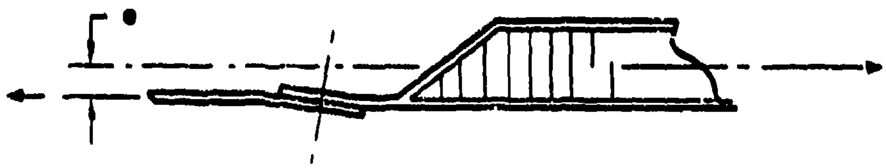
(6)

2.3.2 Load Distribution Among Rows of Fasteners

Assuming a unidirectional applied load, the fasteners in a row are arranged perpendicular to the load direction. Joint configurations affect the distribution of the applied load among the various rows of fasteners in a joint, and the distribution of the row-wise load fraction among the fasteners in any row. Hitherto, the fasteners in a row have been assumed to carry equal loads, and only the row-wise load distribution has been analytically predicted. The SAMCJ code developed in this Northrop/AFWAL program overcomes this limitation, and predicts the two-dimensional load distribution (magnitude and orientation of fastener loads at all locations) for a selected fastener pattern.



A. OUT-OF-PLANE JOINT LOADING DUE TO INTERNAL PRESSURE (e.g., FUEL PRESSURE, FUSELAGE CABIN PRESSURE, ETC.)



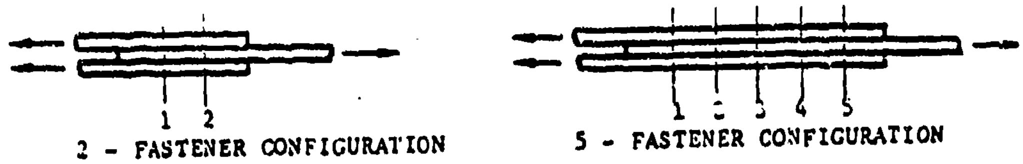
B. OUT-OF-PLANE JOINT LOAD DUE TO LOAD PATH ECCENTRICITY

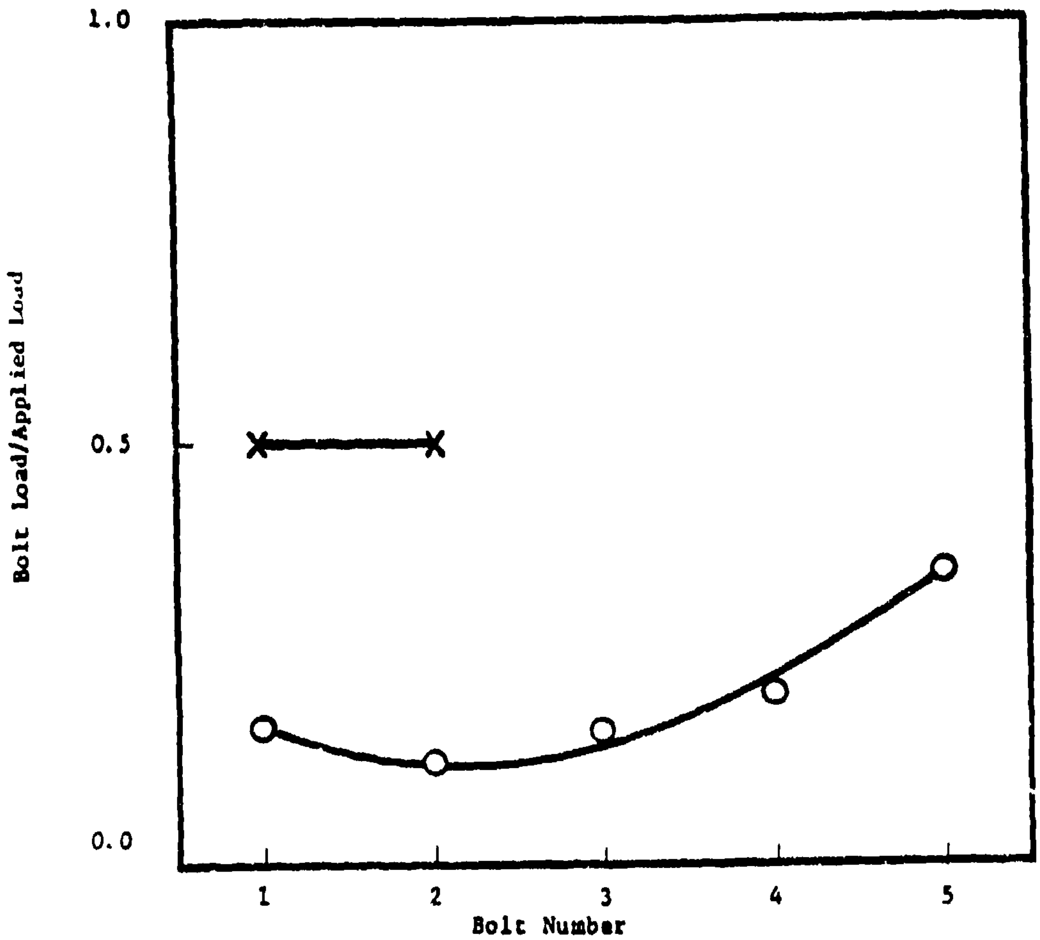
Figure 10. Sample Joint Configurations that Introduce Significant Out-of-Plane Loads at the Joint Location.

Figure 11 presents the load distributions for two and five fastener, double shear joint configurations tested in this program (References 7 and 8). The bolted plates in Figure 11 were uniform in thickness. Figure 12 illustrates how the load distribution among four rows of fasteners can be varied by changing the joint configuration. In the strongest configuration (4), a combination of tapering and reinforcing of the splice plates minimizes the bearing load where the by-pass load is the largest (station 1), and maximizes the bearing load where there is no by-pass load (station 4). The plate width-to-bolt diameter ratio (W/D) is 5 at station 1, and 4 at stations 2 and 3. A larger bolt is used at station 4 (W/D=3). This results in a reduction of the bearing stresses at stations 2 to 4, and the strongest configuration (see References 9 and 10).

In bolted metallic plates, the fastener load distribution is similar to those shown in Figure 11 for low values of the applied load. But, as the applied joint load increases toward the failure value, yielding will occur at peak fastener load locations. This causes the incremental applied load to be carried by the remaining fasteners, generally resulting in a uniform fastener load distribution near failure. For the five fastener configuration in Figure 11, for example, every fastener will carry one-fifth of the applied load at failure. However, laminated plates generally exhibit a linear elastic and brittle behavior, with negligible ductility or yielding. The non-uniform load distribution among rows of fasteners in composite laminates, therefore, remains non-uniform at failure. This reduces the failure load level if the peaks in the load distribution are not accompanied by appropriate thickness tapering and other changes in the joint configuration. Joint efficiency is determined by the overall load-carrying capability of the joint.

The load distribution among rows of fasteners in a bolted laminate generally remains non-uniform at the failure load level, in contrast to what is





Note: Double-shear load transfer between 50/40/10, AS1/3501-6 graphite/epoxy laminate and aluminum using 5/16-inch diameter, protruding head steel fasteners torqued to 100 in-lb ; static tension; RTD.

Figure 11. Fastener Load Distribution in the Laminated Plate for Two Double-Shear Configurations (References 7, 8).

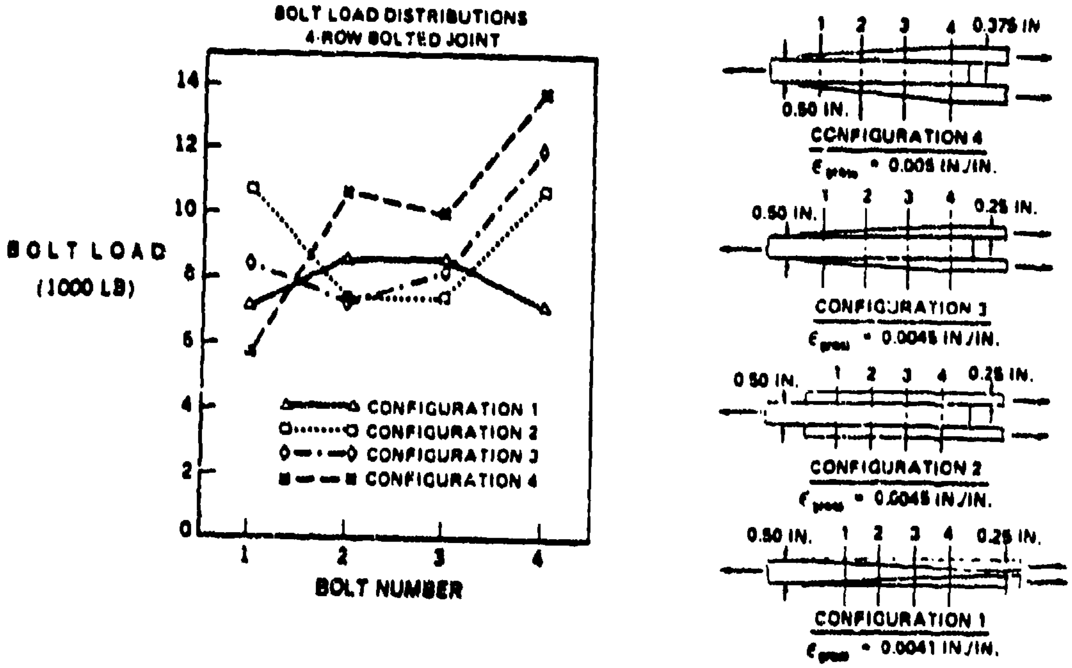


Figure 12. Effect of Joint Configuration on Fastener Load Distribution (Reference 9).

assumed in bolted ductile metals. This adversely influences the failure load for bolted laminates, unless thickness tapering or other configuration changes are introduced.

(7)

2.3.3 Bearing and By-Pass Loads at an Isolated Fastener Location

Figure 13 illustrates the bearing and by-pass loads, and the interaction between them, at an isolated fastener location in a bolted laminate. The failure of the bolted plate is generally assumed to coincide with the failure at the most critical fastener location. The identification of the most critical fastener location requires a knowledge of the load distribution among the fasteners, and an understanding of the interaction between the bearing and by-pass loads at a fastener location (Figure 13).

In ductile metals, minimal interaction is assumed between the bearing load and the by-pass load. However, in composites, a significant interaction has been demonstrated between the two loads under tensile loading (see Figure 13). Only a minimal interaction is observed under compression (see Figure 13). The open hole and bearing strengths of laminates (under tension and compression) are dependent on the laminate layup. The bearing stress at failure is also dependent on the edge distance (geometry) of the bolted laminate when its layup contains more than 40% of 0-degree plies.

Under tensile loading, an increase in the bearing stress reduces the by-pass stress value at failure in bolted laminates.

(8)

Under compressive loading, a minimal or negligible interaction between the bearing and by-pass loads is observed in bolted laminates

(9)

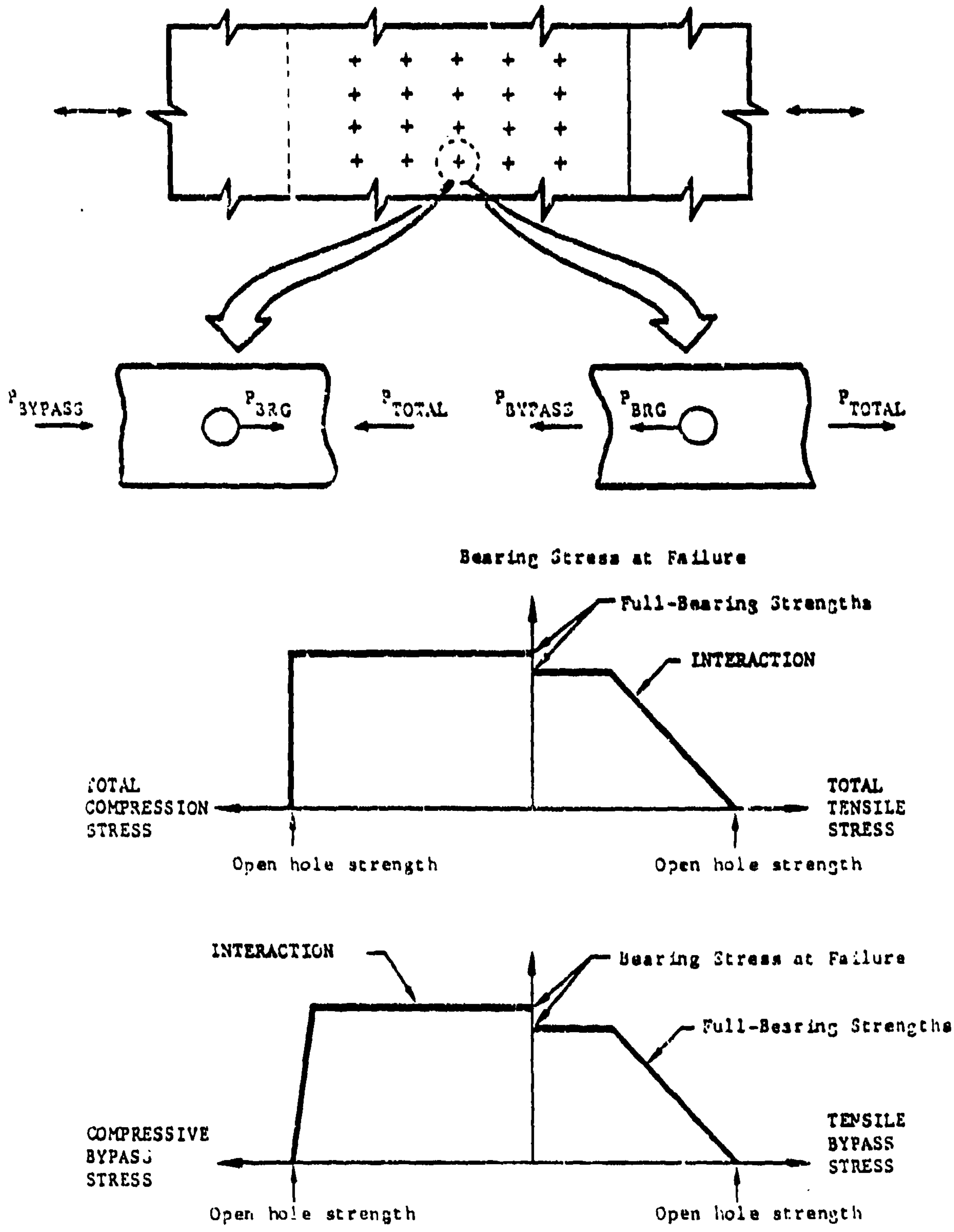


Figure 13. Interaction Between Bearing and By-Pass Loads at a Fastener Location.

2.4 Failure Modes in Bolted Laminates

Bolted laminates exhibit one or more among a variety of failure modes, depending on their layup and geometry, the fastener type and the loading configuration. Figure 14 presents the basic failure modes observed in bolted laminates and possible fastener or fastener-induced failures. In the design of bolted laminates using the SAMCJ computer code, only the net section, shear-out and bearing modes of failures in the laminate are considered, and fastener-related failures are assumed to be preclud *a priori.* Net section and shear-out failures lead to catastrophic joint failures, while bearing failure is generally non-catastrophic. Critical, highly-loaded structural joints should, therefore, be designed to fail in a bearing mode.

Ensuring that fastener-related failures are predicted, highly-loaded structural joints must be designed to fail in a bearing mode to avoid the catastrophic failures induced by net section and shear-out mdes of failure.

(10)

2.5 Fastener Type, Material and Installation Variables

In selecting fasteners for bolted composite structures, many variables have to be considered. These are briefly discussed below.

2.5.1 Fastener Type

Fasteners are available in different forms for different applications, and are broadly classified as protruding head fasteners or countersunk (flush head) fasteners. Countersunk fasteners generally have a 100 degree head angle, and are referred to as tension head or shear head fasteners based on the countersunk depth. Special fastener types include hi-lok, big foot, Jo-bolt, Eddie-bolt, k-Lobe, composite fasteners, etc. (Reference 11).

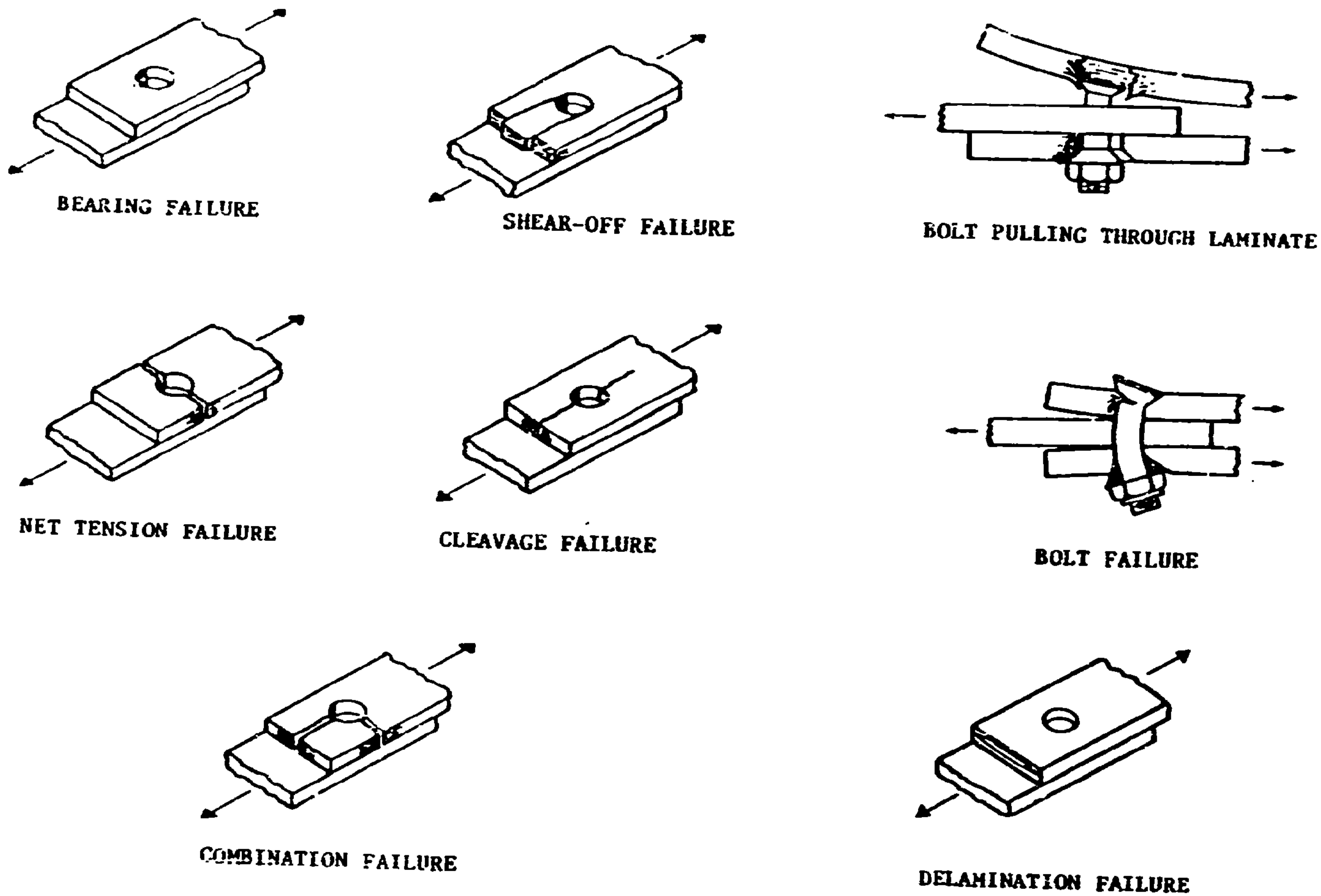


Figure 14. Basic Failure Modes in Bolted Laminates and Fastener-Related Failures.

The joint location influences the selected fastener type and introduces sealing requirements (see Section 2.1). The three guidelines corresponding to this are repeated below:

> Flush head (countersunk) fasteners should be used on aerodynamic surfaces to maintain contour smoothness. (1)

> In fuel containment areas, the fastener locations must be sealed to be leak-proof and to prevent arcing in the fuel cell in the event of a lightning strike. (2)

> In areas of restricted accessibility, blind fasteners must be used. (3)

Tension head countersunk fasteners have a larger countersunk depth than shear head countersunk fasteners. Tension head fasteners, therefore, rest over a larger area of the bolted plate, and carry the load primarily in tension along the fastener axis. Shear head fasteners have a smaller countersunk depth, and carry the load primarily in shear over the fastener cross-section. Consequently, tension head fasteners are capable of carrying larger loads than shear head fasteners. But, when the countersunk depth exceeds approximately 70% of the bolted plate thickness, the fastener effectiveness is reduced due to the local "knife edge" effect, influencing the selection of the fastener type.

Tension head fasteners are preferred over shear head fasteners when the countersunk depth is below approximately 70% of the bolted plate thickness.

(11)

2.5.2 Fastener Material

The main considerations in the selection of the fastener material are its compatibility with the bolted plate material and its mechanical properties. Galvanic corrosion is a problem when steel or aluminum is used adjacent to graphite/epoxy composites, especially in a salt spray atmosphere (see Table 1, Figure 15 and Reference 12). Titanium does not corrode when it is in contact with graphite/epoxy composites. The compatibility of other materials with graphite/epoxy composites is rated in Table 1. Consequently, titanium fasteners are preferred for use in bolted composite structures. Also, a corrosion barrier is generally introduced between bolted composite and metallic parts, if the metal is steel or aluminum (see Figure 15).

Titanium fasteners are preferred for use with graphite-reinforced composites. Steel and aluminum fasteners are not recommended for use with these composites due to their corrosion susceptibility.

(12)

2.5.3 Fastener Size

The fastener size is generally selected to preclude excessive fastener bending effects that could reduce its load transfer capability and induce premature fastener failure. As a general rule, the ratio of the fastener diameter (D) to the bolted plate thickness (t) should be greater than 1 (see Figure 16).

The fastener diameter must be larger than the thickness of either bolted plate.

(13)

2.5.4 Fastener Fit and Hole Quality

Structural parts that are mechanically fastened together are drilled in accordance with established process specifications. Nevertheless, the presence of flaws at fastener locations is commonplace. These flaws include improper fastener seating,

TABLE 1. GALVANIC COMPATIBILITY OF FASTENER MATERIALS WITH COMPOSITES (REFERENCE 12).

Fastener Material	Compatibility with Graphite/Epoxy Composites
Titanium and its alloys	Very Good
MP-35N, INCO 600 (Nickel, Cobolt alloys)	Good
A286, PH13-8MO (Molybdenum alloys)	Acceptable
Monel	Marginal
Low Alloy Steel	Not Compatible
Silver Plate, Chrom. Plate	Adequate with/A286, PH13 13-8MO
Cadmium or Zinc Plate	Not Compatible
Aluminum of Magnesium Alloys	Not Compatible

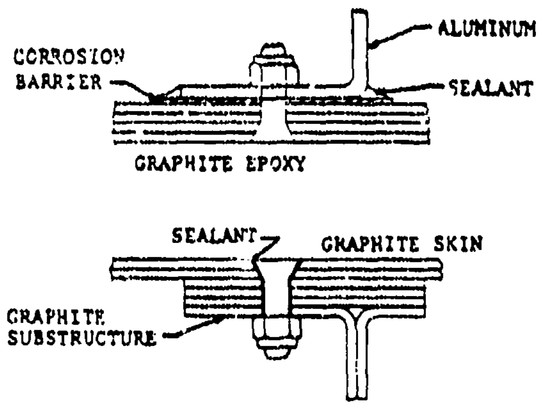


Figure 15. Galvanic Compatibility and Corrosion Prevention.

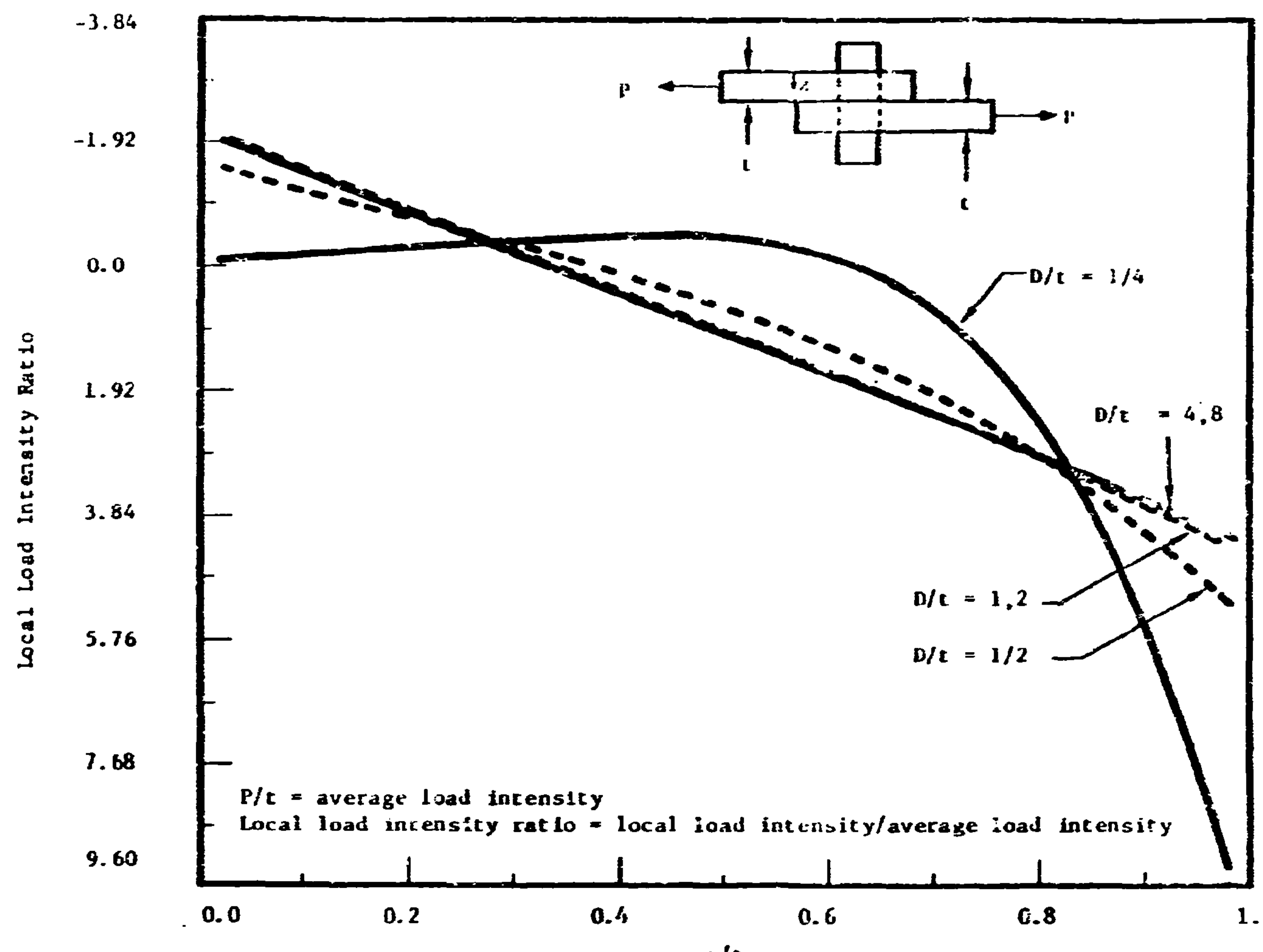


Figure 16. Effect of Fastener Size on the Load Distribution (Reference 6).

cratering of the hole boundary, broken and separated fibers at the drill exit side, delaminations near the exit surface, and a slight tilt (<10 degrees) in the hole axis away from the normal to the bolted plate (Reference 13). Interference fit of fasteners will also affect hole quality and influence the efficiency of the joint. The effects of interference fits and fastener hole flaws were studied in Reference 13 (see Table 2). A summary of the results is presented below:

Interference fastener fits (up to 0.008 inch of interference) induce negligible tensile strength losses. Nevertheless, they are generally not recommended due to installation problems and their effect on hole quality.

(14)

If the countersunk fastener seating (assuming 50% of the bolted plate thickness to be the nominal countersunk depth) is increased beyond 80% of the bolted plate thickness, the joint strength is decreased considerably (20 to 50%).

(15)

If the countersunk hole axis is at least 10 degrees away from the normal to the bolted plate, significant joint strength losses result (over 20% for a 10 degree tilt).

(16)

Other flaws (exit side broken fibers and delaminations, less than a moderate level of porosity in bolted laminates, holes offset by less than 0.005 inch, etc.) at fastener locations introduce negligible joint strength losses (<10%).

(17)

2.5.5 Fastener Torque-Up

Static and fatigue tests on composite-to-metal joints

TABLE 2. EFFECTS OF FLAWS AT FASTENER HOLE LOCATIONS (REFERENCE 13).

	Percent Change in Strength*			
	COMPRESSION		TENSION	
	RTW	250W	RTD	250W
OUT-OF-ROUND HOLES				
• 50/40/10 LAMINATE	–	–	<2.0	–
• 30/60/10 LAMINATE	–	–	-4.8	–
BROKEN FIBERS EXIT SIDE OF HOLE				
• SEVERE	-3.5	-12.2	-9.5	–
• MODERATE	-6.2	<2.0	-4.9	–
POROSITY AROUND HOLE				
• SEVERE	-12.1	-32.8	<2.0	–
• SEVERE WITH FREEZE-THAW	-13.3	–	–	–
• MODERATE	-5.4	-17.9	–	–
• MODERATE WITH FREEZE THAW	-7.9	–	–	–
IMPROPER FASTENER SEATING DEPTH (50% OF NOMINAL)				
• 80% THICKNESS	–	–	-23.2	–
• 100% THICKNESS	–	–	-56.9	–
TILTED COUNTERSINKS				
• AWAY FROM BEARING SURFACE	–	-20.0	<2.0	–
• TOWARD BEARING SURFACE	–	-22.7	-23.9	–
INTERFERENCE FIT TOLERANCES (INCH)				
• 50/40/10 @ 0.003	–	–	<2.0	+14.7
@ 0.008	–	–	<2.0	+11.2
• 30/60/10 @ 0.003	–	–	<2.0	+2.4
@ 0.008	–	–	<2.0	<2.0
FASTENER REMOVAL AND REINSTALLATION				
• 100 CYCLES	–	-7.4	<2.0	

* RTD – Room Temperature, Dry; RTW – Room Temperature, Wet; 250W – 250°F, Wet;
Wet – 0.86% Moisture by Weight.

were conducted in Reference 13, varying the fastener torque-up value from 0 in-lb to 160 in-lbs. Fastener torque-up significantly improved the static strength of the joint (15 to 30%), and its fatigue life at a selected stress level. Similar results were observed in Reference 14. Under fatigue loading, the torque-up inhibits the initial growth of local failures in the joint, and the results in a more abrupt fatigue failure due to excessive hole elongation than a joint with no applied torque.

Fastener torque-up increases the static strength of a joint and its fatigue life at a selected stress level.

(18)

2.6 Bolted Laminate Properties

The basic material and its layup (stacking sequence) in bolted laminates influence the joint performance considerably. When graphite/epoxy laminates are bolted to metallic substructures, galvanic corrosion must be addressed (see Figure 15 and Table 1). For example, a corrosion barrier like a glass/epoxy layer must be used between graphite-reinforced composites and aluminum substructures.

When graphite-reinforced composites are bolted to metallic substructures, corrosion barriers must be introduced if the metal is not compatible with the composite material (see Table 1).

(19)

The bolted laminate layup is generally denoted by the percentages of plies with fiber orientations of 0, + or -45 and 90 degrees, with respect to the primary loading direction, for most structural laminates. The envelope within which a bearing failure mode and the maximum bearing strength are realized is shown in Figure 17. Within this envelope, the strength is independent of the actual stacking sequence. This assumes a laminate width-to-fastener diameter ratio (W/D) of at least 4, and an edge distance (E) of at

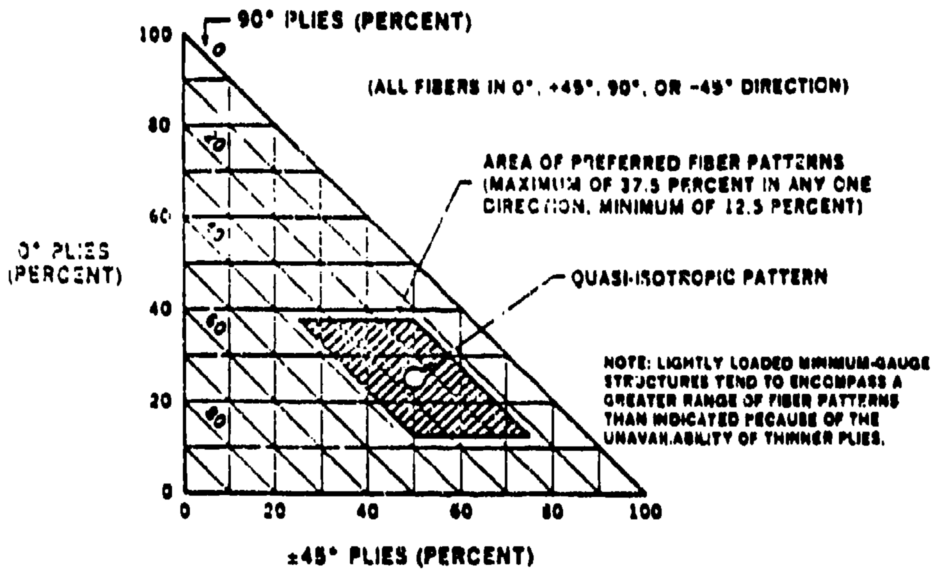


Figure 17. The Envelop of Bolted Laminate Layups for Realizing a Bearing Mode of Failure and the Maximum Bearing Strength (Reference 10).

least 3D. When the percentage of 0 degree plies exceeds 40, a shear-out mode of failure is introduced, reducing the bearing stress value at failure. Section 2.7 presents the effects of fastener spacing and the geometry of a bolted plate on its strength.

The bearing strength of a laminate is maximum when its layup contains less than 40% each of 0, + or -45 and 90 degree plies. The corresponding failure occurs in a bearing mode.	(20)

In addition, the individual plies must be arranged such that adjacent plies have different fiber orientations. If the stacking sequence contains groups of plies with identical fiber orientations, delamination-related failures will occur and reduce the joint strength.

Plies with different fiber orientations should be interspersed within the laminate, to the maximum possible extent, to minimize delamination-induced strength losses. Group of identical plies should not exceed 0.02 inch in thickness.	(21)

2.7 Fastener Spacing and Arrangement

The geometrical parameters that define the fastener spacing and the fastener arrangement in a bolted plate are illustrated in Figure 18. E is the edge distance, S_L and S_T are the fastener spacings in the loading and transverse directions, and W = S_T for a single fastener joint. The effects of these geometrical parameters were studied in References 8, 13 and 14. The results are summarized below:

The bearing and net section strengths decrease when the fastener size increases (see Figure 19).	(22)

The bearing stress at failure decreases

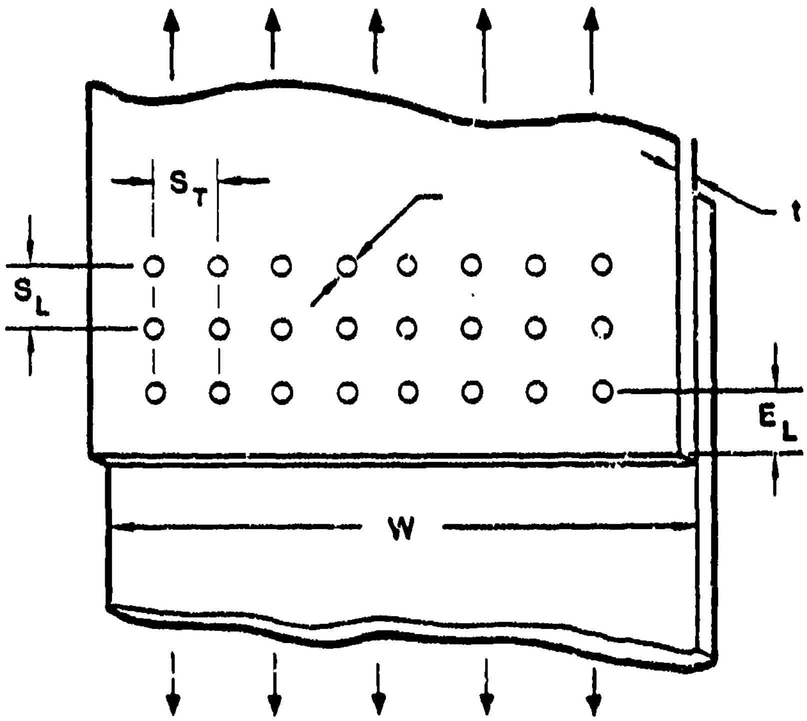
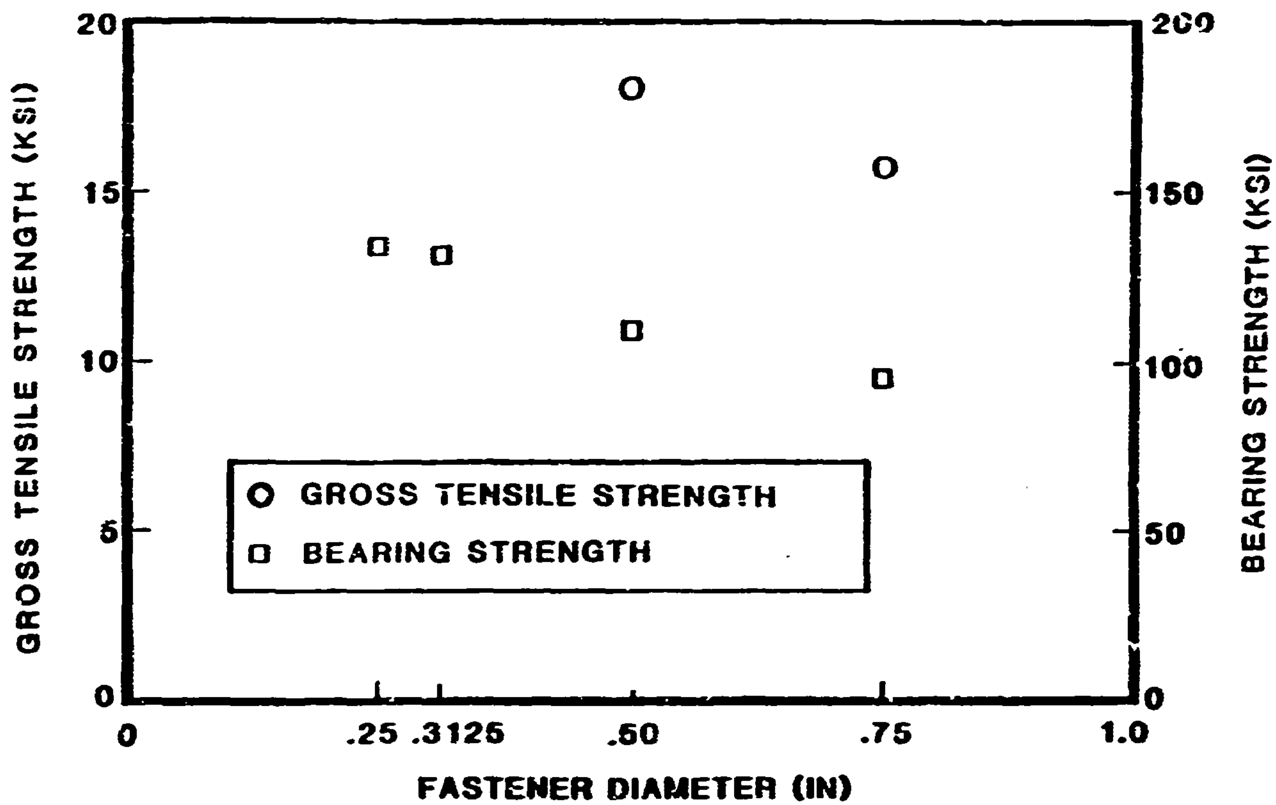


Figure 18. Geometrical Parameters for a Bolted Plate.



NOTE: 20-Ply, 50/40/10, AS1/3501-6 Layup; 0.31 In. Aluminum Plate; Torque = 100 in-lbs, E/D = 3, W/D = 6, Protruding Head Steel Fasteners; RTD; Net Section Strength = 1.2 x Gross Strength

Figure 19. Effect of Fastener Size on the Tensile Response of Composite-to- Metal Joints in Single Shear.

significantly when E/D is reduced below 3 (see Figure 20). A bearing mode of failure is observed only when E/D>4, and the percentage of 0 degree plies is less than 40. A shear-out mode of failure results when E/D<3, or when the percentage of 0 degree plies is >40.

(23)

The bearing stress at failure decreases significantly when S /D (W/D for a single-fastener joint) is reduced below 4 (see Figure 21). When E/D>3, W/D>4, and the percentage of 0 degree plies is below 40, a bearing mode of failure occurs. When W/D<4, a net section failure occurs in the same laminate.

(24)

When the fastener spacing in the loading direction (S_L/D) is decreased below 4, the joint strength decreases due to stress concentration interaction (see Figure 22). The same effect is observed with S_T/D (see Figure 21).

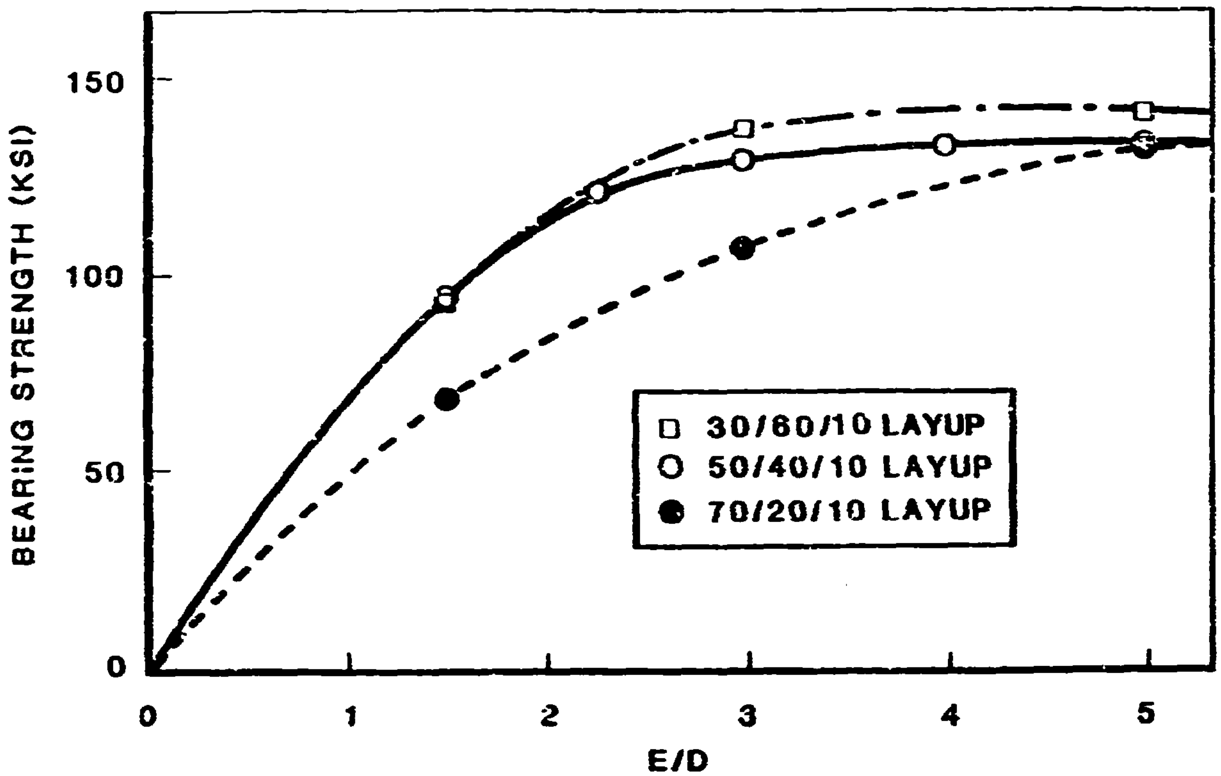
(25)

In summary, ensure that D/t>1, E/D>3, W/D (S_T/D)>4, S_L/D>4, and the percentage of plies in any orientation is <40, to achieve a bearing failure mode and to realize the maximum joint strength.

(26)

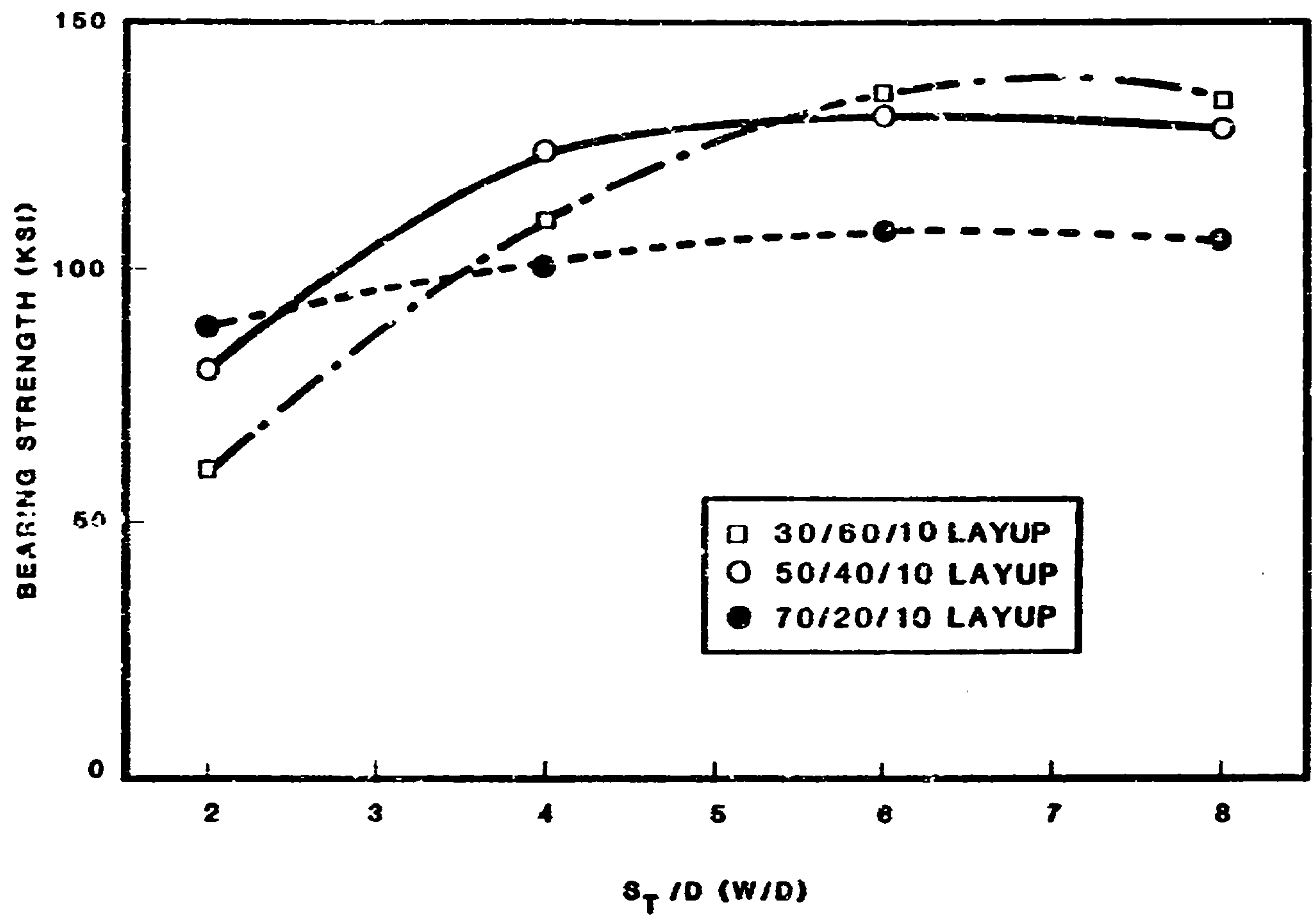
2.8 Joint Tailoring for Maximum Efficiency

The design of a joint should achieve the following objectives to be considered efficient: (1) It should be capable of transferring the design ultimate loads without failing any member; (2) It should possess the design life when subjected to the design spectrum fatigue loading; (3) It should be the least weight design that meets (1) and (2); and (4) The complexity of the design concept



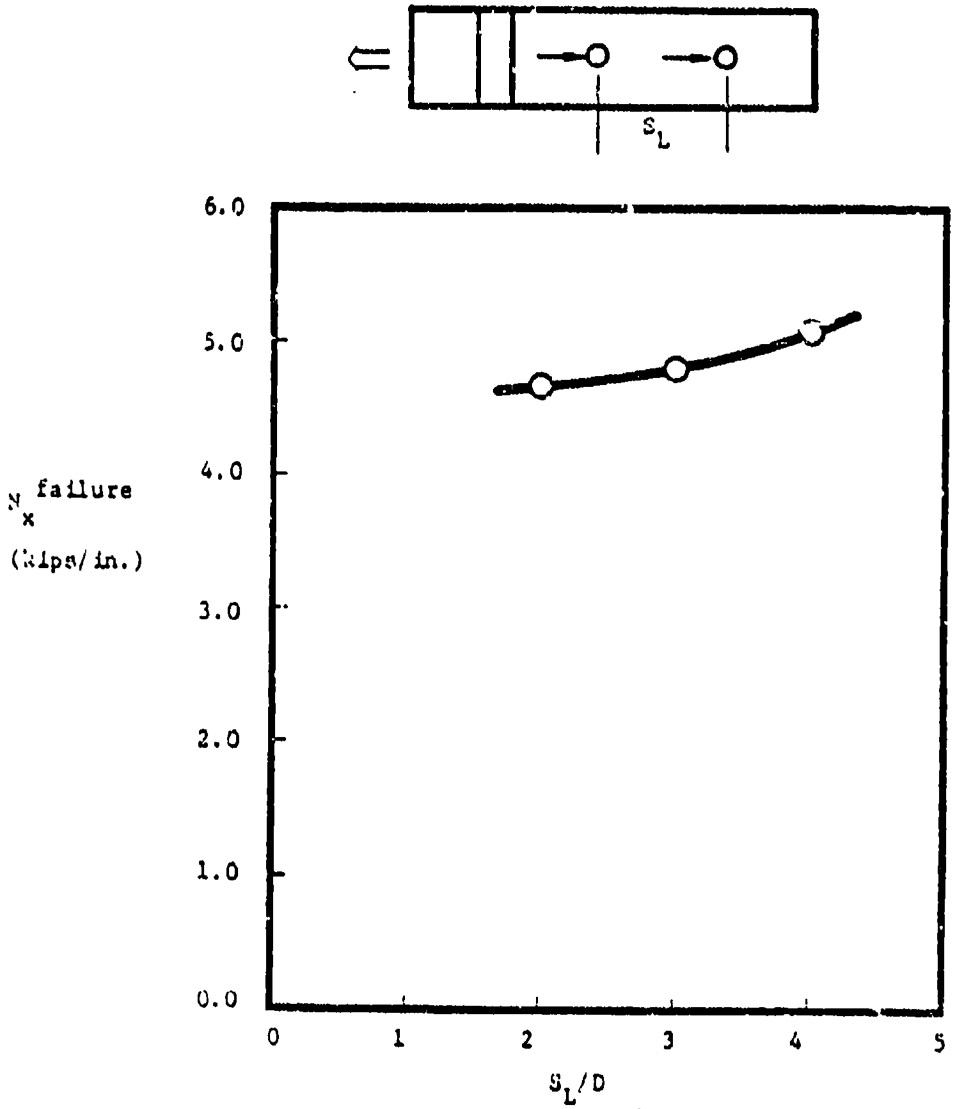
NOTE: Composite-to-metal joints in single shear; 20-ply AS1/3501-6 layups; 0.31 in. aluminum plate; W/D = 6, Torque = 100 in-lbs; Protruding Head, steel fastener; D = 5/16 in.

Figure 20. Effect of E/D on the Bearing Strength of Bolted Laminates.



NOTE: Composite-to-metal joints in single shear; 20-ply, AS1/3501-6 layups; 0.31 in. aluminum plate; E/D = 3; Protruding head steel fastener; D = 5/16 in.; Torque = 100 in-lbs

Figure 21. Effect of W/D on the Bearing Strength of Bolted Laminates.



Note: Composote-to-metal, two fasteners-in-a-row joint; 20-ply, 50/40/10 layup; AS1/3501-6 graphite/epoxy; 0.31 in. aluminum plate, single shear; RTD; static tension; S_T/D=W/D=6; protruding head steel fastener; D=5/16 in.; T=100 in-lbs.

Figure 22. Effect of S_L/D on the Strength of Bolted Laminates.

should be controlled to aid producibility and maintainability of the structural joint.

A joint can be tailored to improve its efficiency. For example, when the number of fastener rows (a row being perpendicular to the primary loading direction) is increased, the peak load fraction is generally carried by the innermost or outermost fastener row (see Figures 11 and 12). If the failure mode at the critical fastener location is bearing or net section, the thickness and width of the bolted plate at that location will influence the joint failure load. In an efficient design, the width and the thickness of the bolted plates will be tailored such that every fastener location is equally critical (see Figure 5). The peak bearing stress at the design ultimate load level will be lowered to a level that ensures a minimal bearing/by-pass interaction, if possible (see Figure 13).

Some experimental concepts have also been demonstrated to be efficient joint tailoring concepts, despite the difficulty they introduce in applying the concept at the production level. An example is shown in Figure 23, where the 0 degree plies in the bolted skin are replaced by + and -45 degree plies in the joint region (Reference 15). This causes a smaller fraction of the running load to be transferred at the joint location, and also increases the local bearing strength. An alternative, equivalent concept would be to replace the stiffer material by a tougher material at the joint location. For example, graphite/epoxy plies can be replaced by aramid fiber/epoxy plies at the joint location. It is reiterated, though, that these validated tailoring concepts are difficult to implement in a production environment.

The geometry of bolted laminates must be tailored, in the width and thickness directions, to render every fastener location equally critical.

(27)

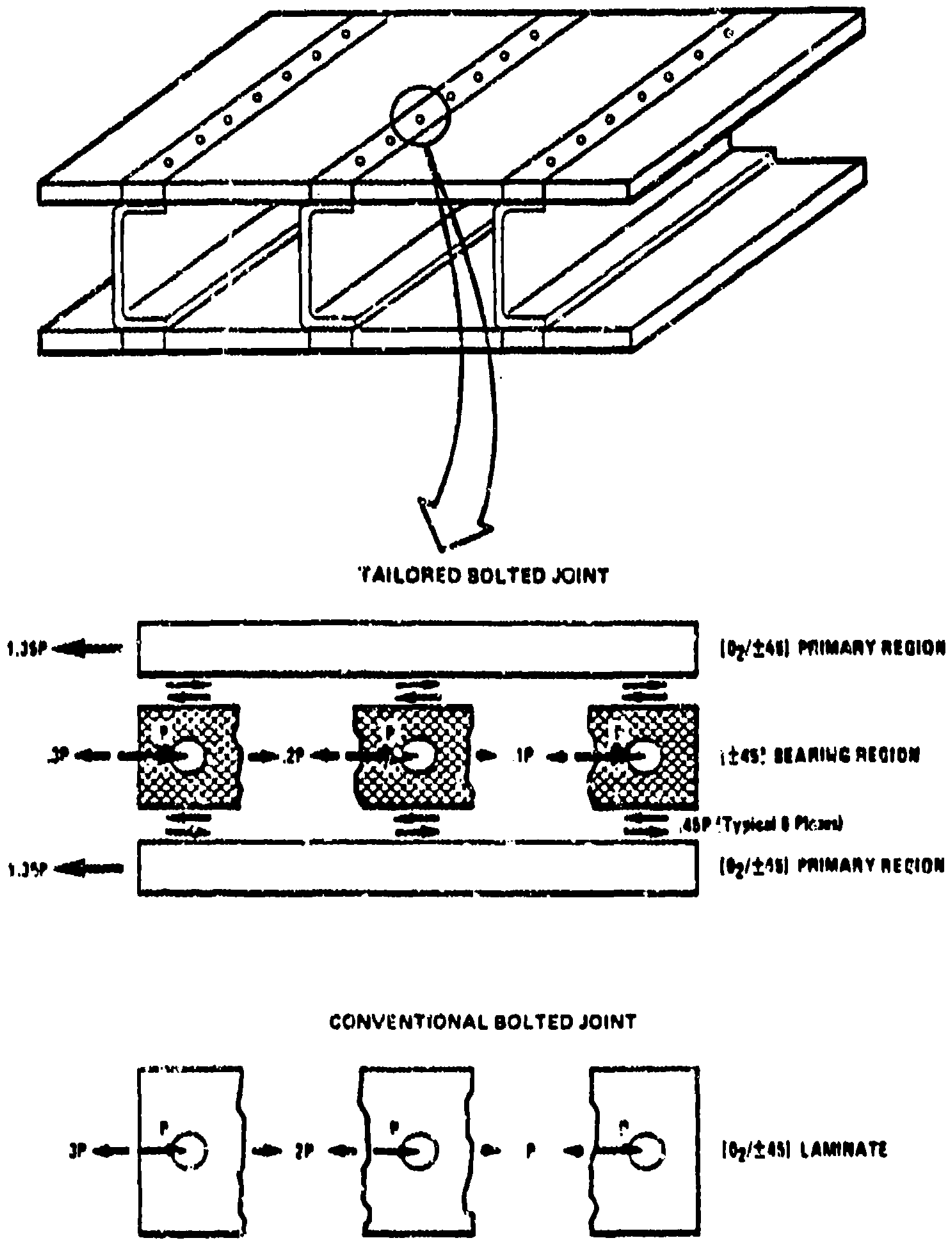


Figure 23. A Sample Tailored Joint.

2.9 Durability Considerations

The design of a bolted joint is currently based on an assumed design ultimate load level and a static strength analysis (see Section 3). The assumed design ultimate load level should account for durability considerations also. Generally, irrespective of the static failure mode, a bolted joint suffers fatigue failure via excessive hole elongation (bearing). This possible change in the failure mode from the static loading case to the fatigue loading case has been observed by many in the literature (see References 13 and 14).

If the joint statically fails in a bearing mode, it could suffer premature excessive hole elongation (fatigue failure) when subjected to the spectrum fatigue loading. Figures 24 and 25 present sample constant amplitude fatigue test results from Reference 14 for a fully reversed loading case (R=-1). Similar results should be used to approximately and conservatively estimate the fatigue life of a joint using a fatigue analysis (Miner's rule, for example). Based on the fatigue analysis, the bearing stress at the critical fastener location should be designed to be sufficiently lower than the static bearing strength, to ensure the design life of the joint. The final joint design, therefore, will be capable of statically transferring the design ultimate load, with the peak bearing stress value ensuring the design fatigue life.

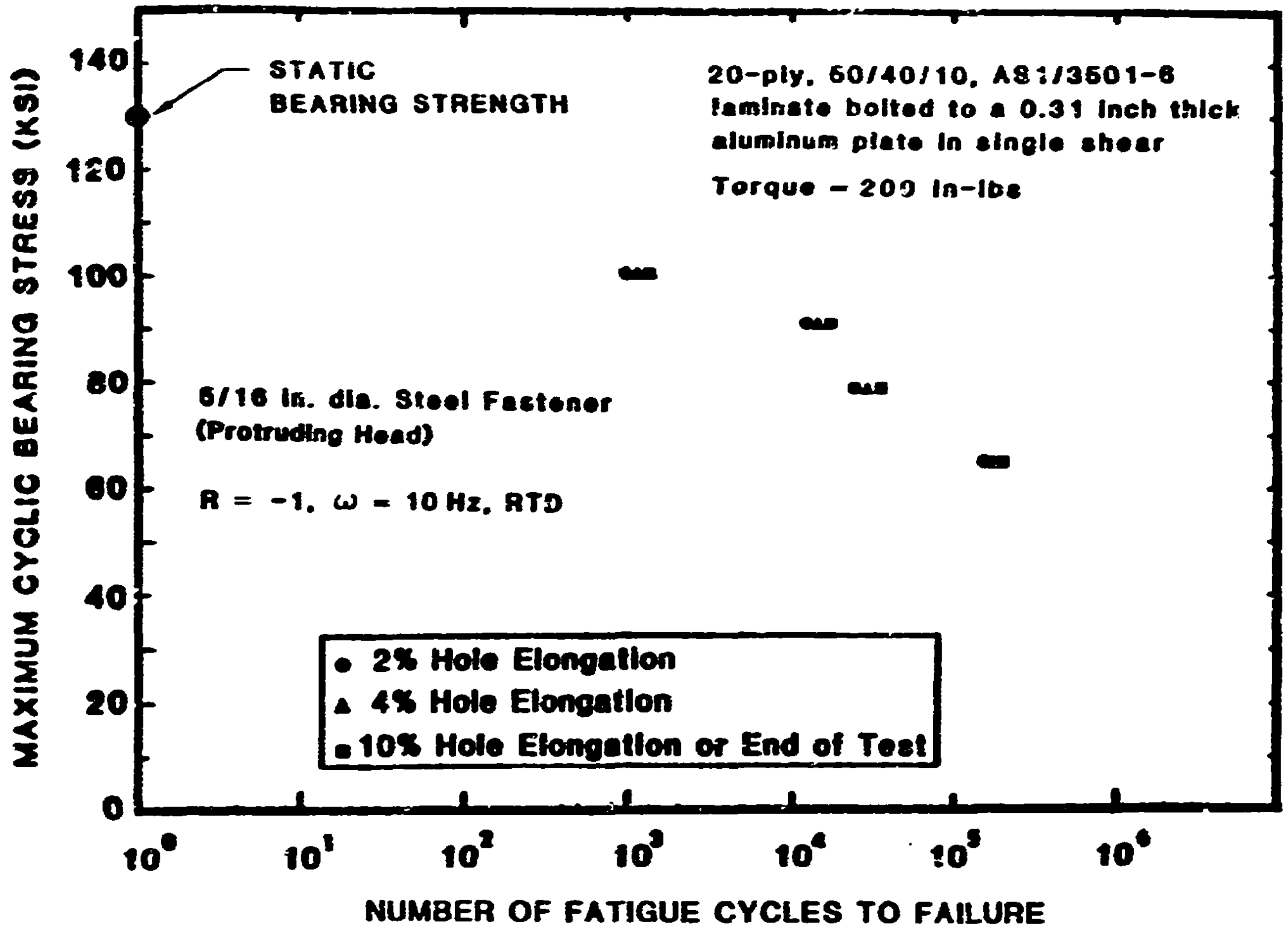


Figure 24. Effect of Maximum Cyclic Bearing Stress on the Number of R=-1 Fatigue Cycles to Cause Specified Hole Elongations in a Bolted Laminate.

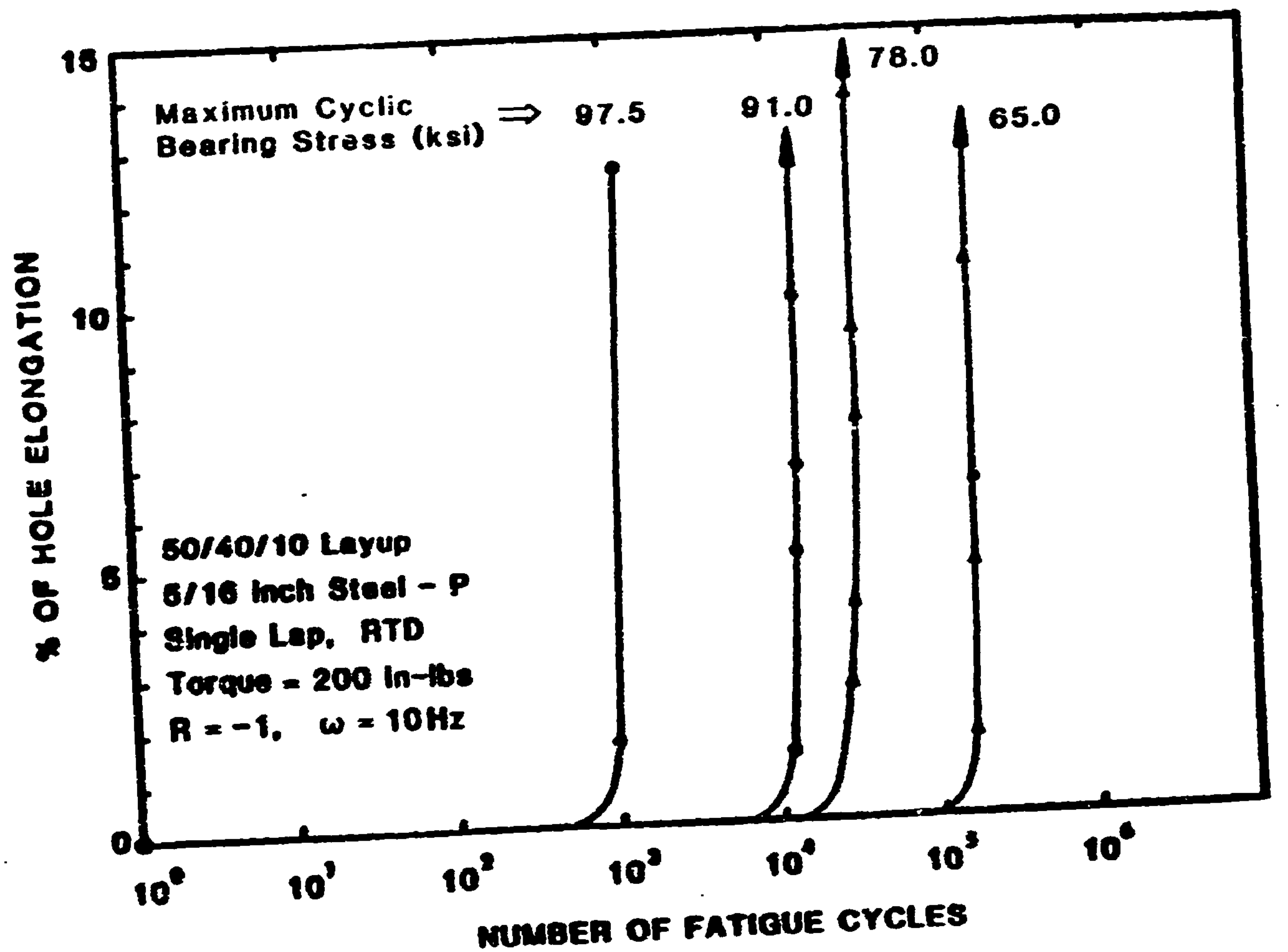


Figure 25. Effect of Maximum Cyclic Bearing Stress on the Hole Elongation Rate for a Bolted Laminate under R=-1 Loading.

SECTION 3

STRENGTH ANALYSIS OF BOLTED COMPOSITE STRUCTURES

As mentioned in Section 1.4, two computer codes were developed in this Northrop/AFWAL program to predict the strength of bolted joints containing a single fastener (SASCJ and SAMCJ) or multiple fasteners (SAMCJ). Most of the structural joints contain multiple fasteners, and SAMCJ is adequate for the design of these joints. SAMCJ is also capable of predicting the strength of single fastener joints, without accounting for the nonlinear joint load versus deflection behavior introduced by ply level failures. However, if the user wishes to interrogate an isolated fastener location, accounting for the nonlinear joint behavior due to progressive (two-stage) ply failures, the SASCJ code is useful. The reader is referred to References 6 and 7 for detailed descriptions of the SASCJ and SAMCJ analyses, respectively.

In the following sub-sections, brief descriptions of the analyses in the SASCJ and SAMCJ computer codes are presented, along with detailed instructions for the use of these analytical design tools.

3.1 Description of SASCJ Analysis

A two-dimensional anisotropic plate analysis that accounts for finite plate dimensions (FIGEOM), and a finite difference fastener analysis (FDFA), are incorporated into a progressive failure procedure to develop a strength analysis for single fastener joints in composite structures (SASCJ). An isolated fastener location in a bolted structures (see Figure 7) is primarily subjected to the loading shown in Figure 26. The general bolt bearing/by-pass situation can be analyzed as a superposition of an unloaded hole situation and a fully loaded hole situation (see Figure 26). The unloaded hole case is analyzed using the two-dimensional plate analysis (FIGEOM), and does not involve the

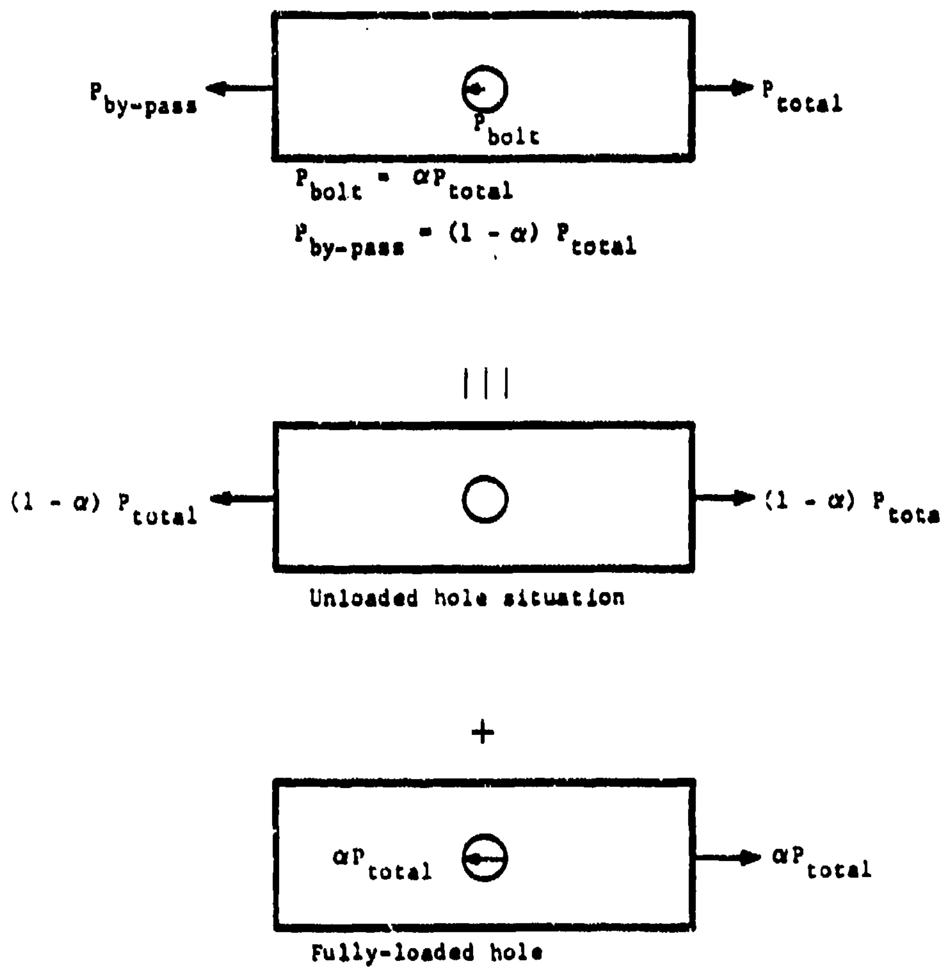


Figure 26. Schematic Representation of a General Single-Fastener Situation as a Superposition of Unloaded and Fully Loaded Hole Situations.

fastener analysis (FDFA). The fully-loaded hole situation is analyzed using a progressive failure procedure that predicts local ply failures and delaminations until the bolted plate cannot carry any additional applied load. The employed ply-failure criteria and the delamination criterion are discussed in Sections 3.1.3 and 3.1.4.

3.1.1 Strength Analysis Procedure for Fully-Loaded Holes

The strength of laminates with fully loaded holes is predicted using the procedure outlined in Figure 27. A two-dimensional stress analysis (FIGEOM), accounting for finite dimensions of the bolted plates, is initially performed on each bolted plate. Computed plate stresses are used to calculate the effective moduli of the various ply types in each bolted plate (see Reference 6). The inplane strains computed by the FIGEOM code are used to obtain the stress state in each ply. The ply stresses around the hole boundary are integrated to yield the bearing load in each ply (see Reference 6). The inplane stresses in each ply, per unit bearing load, are incorporated into selected failure criteria to compute the ply (bearing) loads corresponding to the various inplane failure modes.

The effective moduli and the ply bearing loads corresponding to the various failure modes, for all the plies in each bolted plate, are incorporated into the fastener analysis. The initial fastener analysis on the undamaged plates computes the distribution of the applied bearing load among the various plies. Comparing these ply loads with the stored failure values for inplane ply failures, the joint load corresponding to the earliest ply failure is obtained. The fastener analysis also computes approximate shear strain values at the interfacial locations between adjacent plies. Incorporating these into an interlaminar failure criterion, the joint load corresponding to the earliest interlaminar failure (delamination) is obtained. The smaller of the two joint loads, corresponding to the earliest inplane and interlaminar

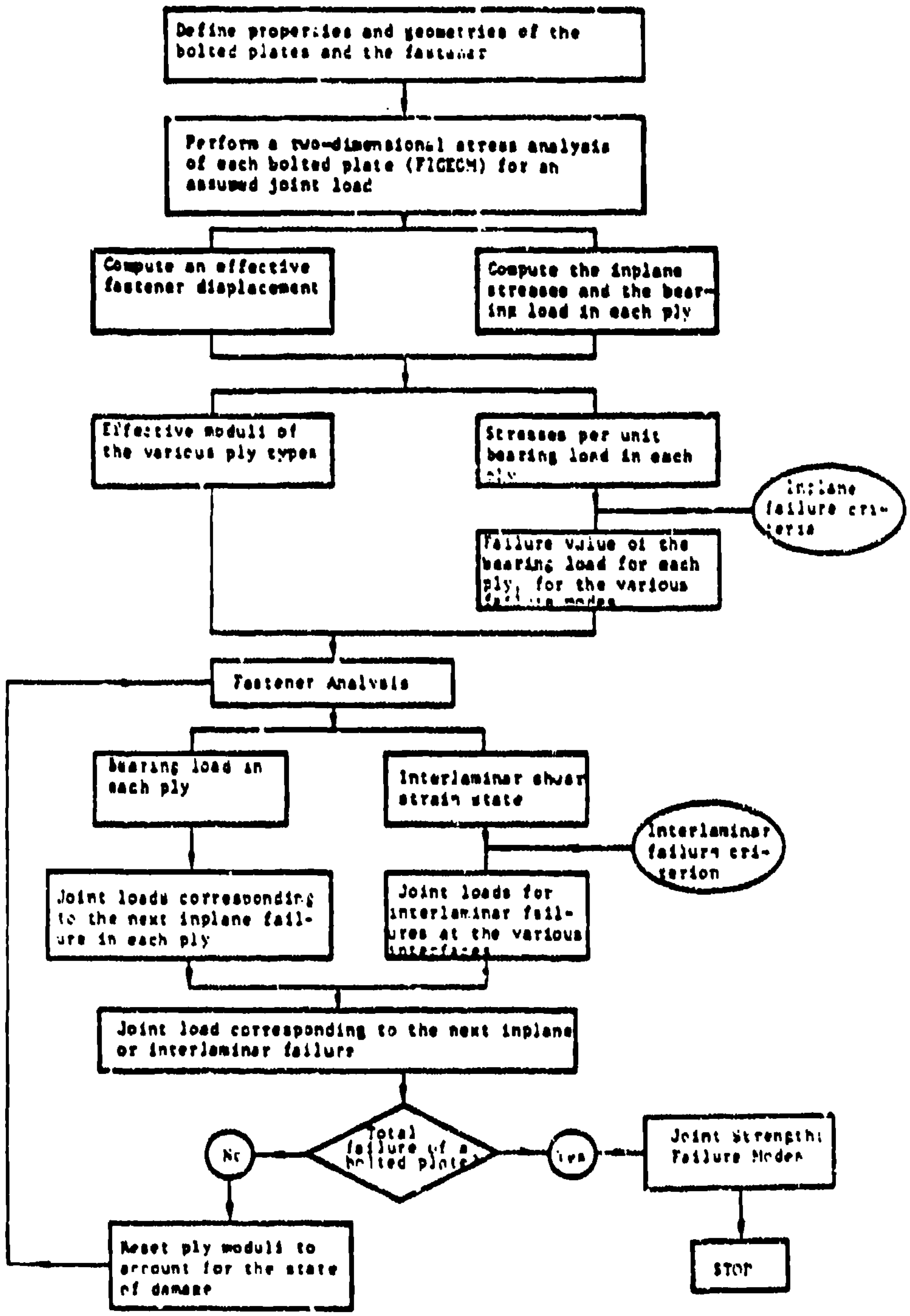


Figure 27. Flowchart for the Strength Analysis of Laminates with Fully Loaded Holes.

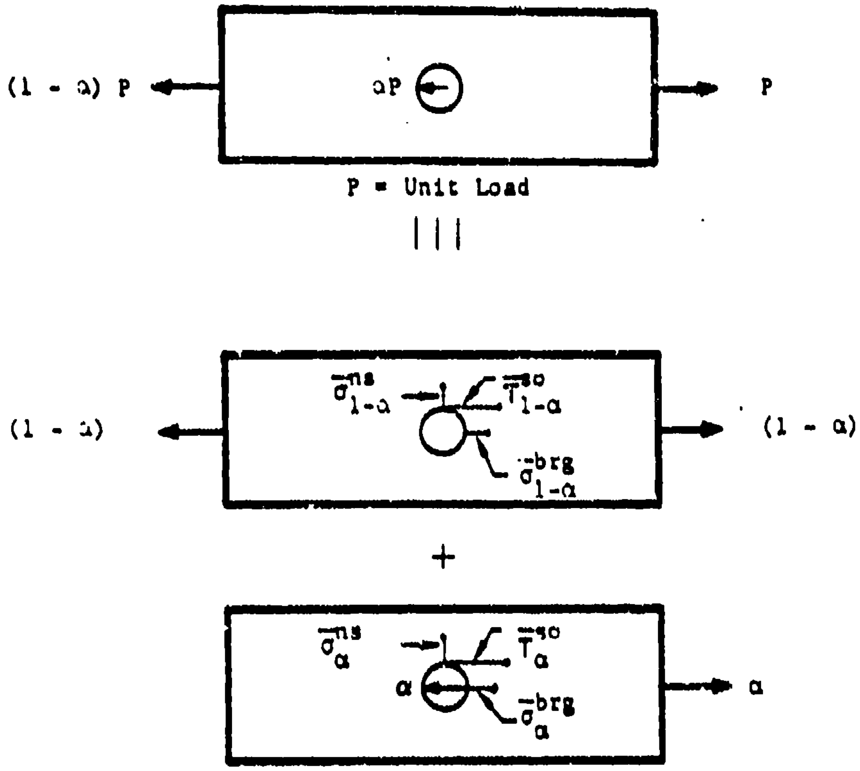
failures, determines the first failure in a bolted plate and the corresponding joint load value.

The effective moduli of the damaged plies are reset to appropriately represent the predicted failure modes. The revised moduli are incorporated into the fastener analysis, and the procedure is repeated to predict the next failure mode and the corresponding joint load. When any ply is predicted to fail totally, the analysis computes the redistribution of the corresponding joint load among the remaining effective plies, and determines if any other concomitant ply failure is precipitated. This process is repeated until one of the bolted plates becomes ineffective in transferring the applied load (joint failure).

The SASCJ computer code is restricted to protruding head fasteners, and assumes that fastener failure is precluded. However, when a countersunk fastener is specified, SASCJ assumes an appropriate boundary condition at the head location, and expects the user to input an equivalent (larger) uniform fastener diameter. It can analyze any combination of laminated and metallic plates, bolted together in a single-lap or double-lap configuration.

3.1.2 Strength Analysis Procedure for Partially-Loaded Holes

A general fastener location in a bolted plate transfers a fraction (α) of the total applied load via the fastener, the remainder ($1-\alpha$) being by-passed to the next fastener location (see Figures 7 and 26). In this case, the stress state at the fastener location is computed as a superposition of the stress states corresponding to the unloaded and fully-loaded hole situations. Figure 28, for example, presents a schematic representation of how the averaged stresses are obtained to predict net section, shear-out and bearing failures in the plies using average stress failure criteria. For a unit applied load, the averaged stresses in the laminate with an unloaded hole, when subjected to a load of ($1-\alpha$), and the averaged stresses in the laminate with a fully loaded hole,



$$\bar{\sigma}^{ns} = \int_{D/2}^{D/2 + d_{ons}} \sigma_x(o,y)\, dy = \bar{\sigma}^{ns}_{1-\alpha} + \bar{\sigma}^{ns}_{\alpha}$$

$$\bar{\tau}^{so} = \int_{0}^{d_{oso}} \tau_{xy}\,(x,D/2)\, dx = \bar{\tau}^{so}_{1-\alpha} + \bar{\tau}^{so}_{\alpha}$$

$$\bar{\sigma}^{brg} = \int_{D/2}^{D/2 + d_{obrg}} \sigma_x(x,o)dx = \bar{\sigma}^{brg}_{1-\alpha} + \bar{\sigma}^{brg}_{\alpha}$$

Figure 28. Strength Analysis of Laminates with Partially-Loaded Holes using Average Stress Failure Criteria.

when subjected to a load of α, are computed separately and added. Incorporating the combined averaged stresses into the appropriate failure criteria, the applied load corresponding to a ply failure is computed.

In the case of fully loaded holes, progressive failure prediction involves the repetition of the fastener analysis with revised ply properties after every ply failure. The two-dimensional analysis (FIGEOM) is only carried out once. But, in the case of partially loaded holes, a ply failure will affect the unloaded and the fully loaded hole contributions to the local stresses. Hence, progressive failure prediction in the partially loaded case involves repeating FIGEOM and FDFA analyses after total ply failures.

3.1.3 Inplane Failure Criteria

The SASCJ code permits the user to select any of the following five failure criteria for the prediction of ply failures based on inplane stresses and strains: (1) point stress failure criterion, (2) average stress failure criterion, (3) maximum (fiber directional) strain criterion, (4) Hoffman criteron, and (5) Tsai-Hill criterion. The first two criteria predict three modes of failure in each ply--net section, shear-out and bearing. The maximum strain criterion predicts ply failure based on fiber failure. The Hoffman and Tsai-Hill criteria predict ply failure accounting for biaxial stress interaction that is ignored by the first three criteria.

The point stress failure criterion predicts net section, shear-out and bearing failures when the appropriate stress components at selected locations attain unnotched specimen failure values (see Figure 29). a_{ons}, a_{osc} and a_{obrg} are called characteristic distances. When σ_x $(0, D + a_{ons})$ exceeds the unnotched tensile or compressive strength of the ply, as appropriate, a net section ply failure is predicted. When σ_x $(D + a_{obrg}, 0)$ exceeds the unnotched compressive strength of the ply, a bearing mode of ply failure is

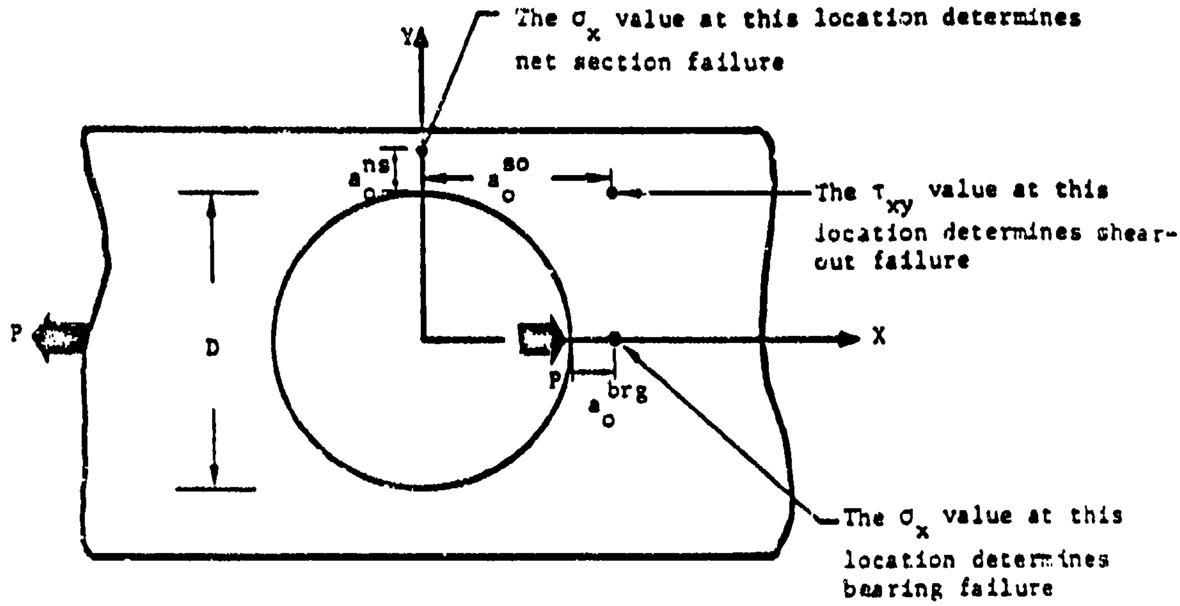


Figure 29. The Characteristic Distances used in the Point Stress Failure Criteria

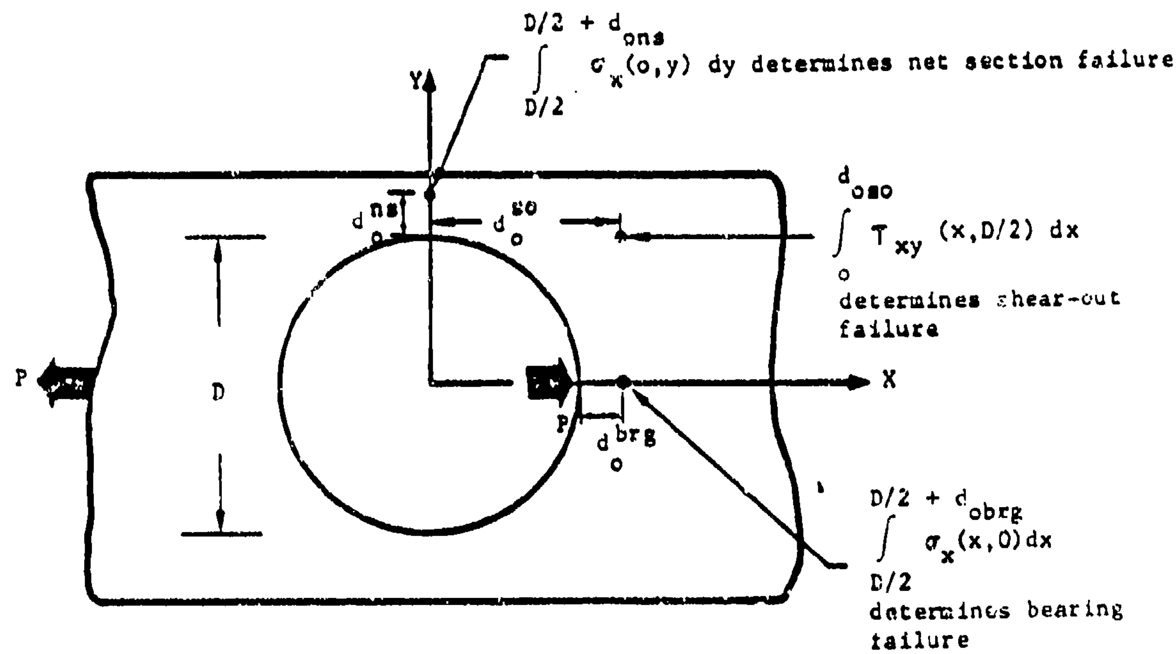


Figure 30. The Characteristic Distances Used in the Average Stress Failure Criteria.

predicted. When τ_{xy} (a_{oso}, D/2) exceeds the unnotched ply shear strength, a shear-out mode of ply failure is predicted. The averag stress failure criterion predicts these failures based on averaged values of the mentioned stress components over selected characteristic distances (d_{ons}, d_{oso}, and d_{obrg}) that are larger in magnitude compared to those used in conjunction with the point stress criterion(see Figure 30).

Of the three ply failure modes, only the net section mode causes the ply to become almost ineffective (total failure). The bearing node of failure causes the ply to suffer a reduction in its effective modulus without losing its load-carrying capacity. The shear-out mode of failure causes a ply to become ineffective only when it is delaminated from the adjacent plies. When a ply suffers any of the above failures, its load versus deflection response is at the knee of the bilinear representation in Figure 3 The damaged ply can carry additional load until total ply failure precipitated. The SASCJ computer code automatically stores the damage state in every ply in the bolted plates, and reassigns value for ply moduli to appropriately represent predicted ply failures. When a ply suffers total failure, its modulus is set equal to zero, and the redistribution of the joint load among the remaining plies is computed. A typical overall load versus deflection behavior of the joint is shown in Figure 32, indicating the effects of local an total ply failures.

The maximum strain (fiber directional), Hoffman and Tsai-Hill criteria are applied along a path that is concentric to the fastener hole, at a characteristic distance (a_o) from the hole boundary (see Figure 33). The location along this path where the selected criterion is satisfied determines the failure location. The maximum strain criterion predicts fiber failure in a ply (total ply failure) when its fiber directional strain exceeds the failure values (ϵ_{11}^{tu} or ϵ_{11}^{cu}).

The Hoffman failure criterion, based on inplane ply

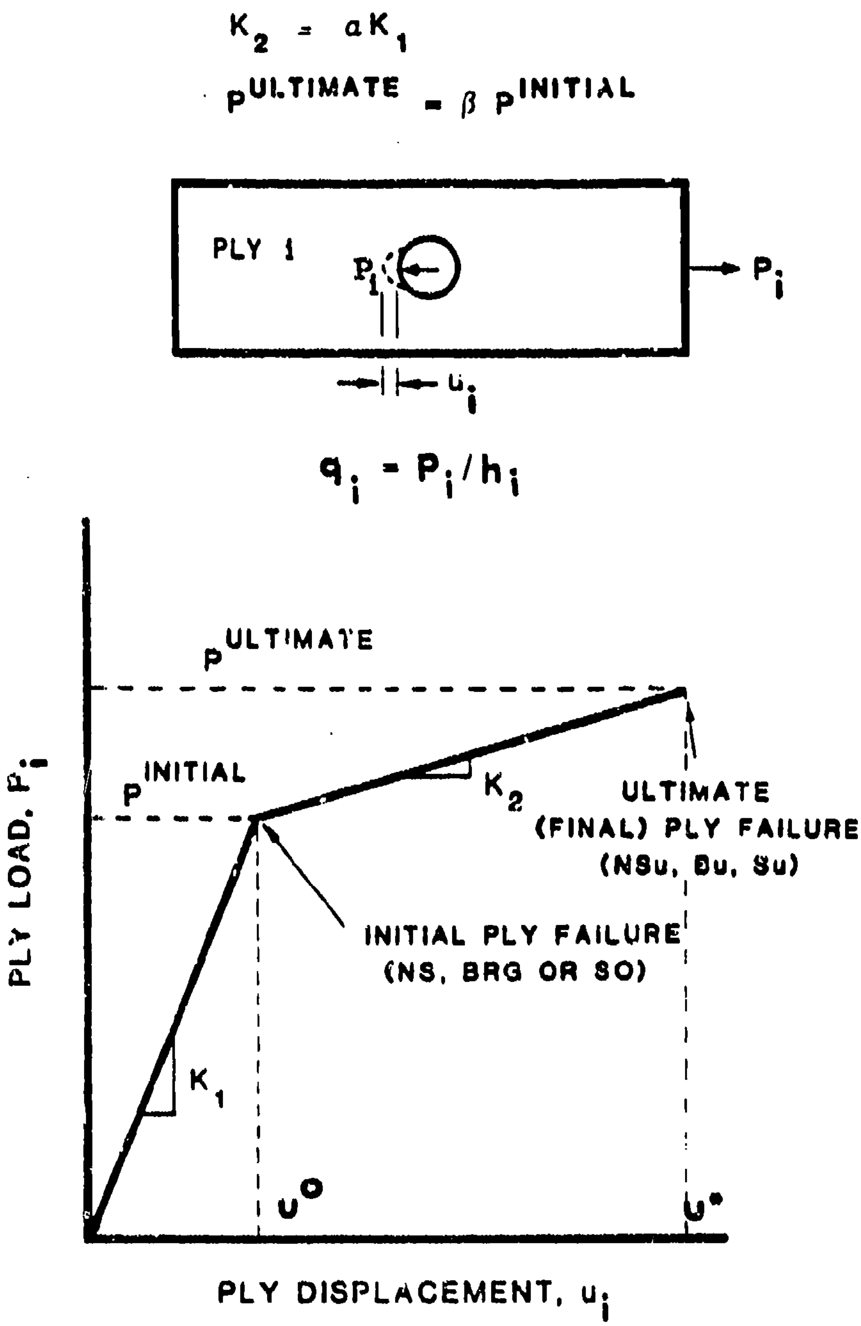


Figure 31. Bilinear Elastic Behavior of a Ply.

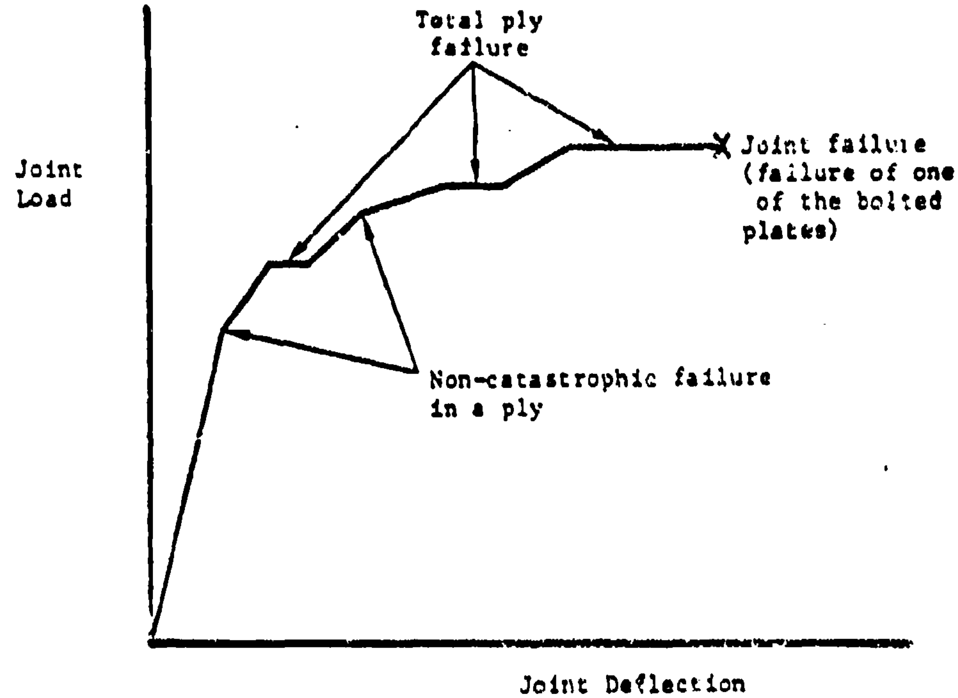


Figure 32. A Schematic Representation of the Overall Load Versus Deflection Response of the Joint.

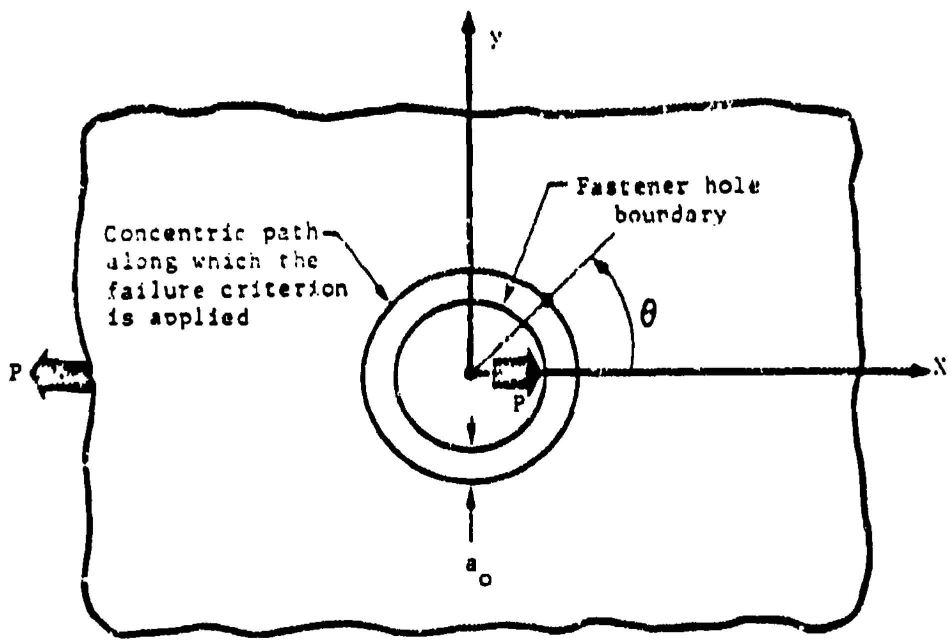


Figure 33. The Characteristic Distance (a_o) Defining the Region Where the Maximum Strain, Hoffman or Hill Criterion is Applied.

stresses, states that total ply failure will occur when the failure index (H) in the following equation reaches a value of unity:

$$(\sigma_1^2 - \sigma_1\sigma_2)/X_cX_t + \sigma_1(X_c-X_t)/X_cX_t + \sigma_2^2/Y_cY_t + \sigma_2(Y_c-Y_t)/Y_cY_t +$$
$$\sigma_6^2/S^2 = H$$

In the above equation, σ_1, σ_2 and σ_6 are the ply stresses in the fiber coordinate system, X_t and X_c are the uniaxial tensile and compressive material strengths along the fiber direction (1), Y_t and Y_c are the uniaxial tensile and compressive material strengths perpendicular to the fiber direction (2), and S is the material shear strength in the 1-2 plane.

In the SASCJ code, the Hoffman criterion is applied along a path that is concentric to the fastener hole, defined by the characteristic distance a_o (see Figure 33). At selected points along this path, the following expressions for the failure values of the ply load (P_f) are computed:

$$P_f = (-b \pm \sqrt{b^2-4ac})/2a$$

where

$$a = \left[(\sigma_1^2-\sigma_1\sigma_2)/X_cX_t + \sigma_2^2/Y_cY_t + \sigma_6^2/S^2\right]/P_i^2$$

$$b = \left[(X_c-X_t)\sigma_1/X_cX_t + (Y_c-Y_t)\sigma_2/Y_cY_t\right]/P_i$$

$c = -1$, and

P_i = ply load at which σ_1, σ_2 and σ_6 are computed

The location where the smallest non-negative value for P_f is computed identifies the failure initiation point.

The Hoffman criterion predicts total ply failure and the failure location, but does not identify the mode of failure. The failure location, though, generally indicates the possible failure mode. Referring to Figure 33, if failure is predicted near $\theta=0°$, a bearing mode of failure is suspected. If the failure location is near $\theta=90°$, a net section mode of failure is suspected. And, intermediate values of θ indicate a shear-out mode of failure. The Tsai-Hill criterion can be obtained from the Hoffman criterion by setting $X_t = X_c$ and $Y_t = Y_c$. This criterion, therefore, does not account for different strengths under tension and compression. The ply failure load (P_f) in this case is computed to be $1/\sqrt{a}$.

3.1.4 Interlaminar Failure Criterion

Delamination between plies is predicted by incorporating computed shear strains at the interfacial locations into a maximum shear strain criterion. At the interface between plies i and j, for example, the shear strain is computed to be:

$$\gamma_{xz}^{i-j} = (u_i - u_j) / h_\alpha$$

where h_α is the ply thickness in the plate containing plies i and j. This expression for the shear strain is approximate. Plies i and j are assumed to delaminate when γ_{xz}^{i-j} exceeds a failure value. The failure value for γ_{xz} is determined by correlating predictions with observations for a sample test case.

3.2 SASCJ Input Description

SASCJ assumes a uniaxial tensile or compressive load to be applied to a single fastener bolted joint, in a single or a double shear configuration (see Figures 34 and 35). The code requests information for a general bearing/by-pass situation. If the joint is a symmetric double shear configuration, only half the joint is analyzed (see Figure 35). For example, if plate 2 in Figure 35 is metallic, the input thickness should be half the actual value, and if plate 2 is a laminate, only the layup from the surface to its midplane should be input. The analysis accounts for the joint symmetry through appropriate symmetry conditions at the midplane location (see Figure 35).

A sample SASCJ problem is now presented to describe the input requirements for the code. It addresses a steel-to-composite joint in a single shear configuration (see Figure 34). The input is requested by SASCJ in an interactive mode. Figure 36 presents the code requests and the user replies for the sample joint. Though the information in Figure 36 is self-explanatory, a description of the input quantites is presented below.

The first input quantity specifies that the problem addresses a bearing/by-pass situation with a by-pass ratio of 0.99 --nearly an open hole situation. The second and third input quantities specify that a static tensile load is applied in a single shear configuration. Subsequently, the two bolted plates are specified to be either a composite laminate or a metal. If the bolted plate is a laminate, SASCJ requests the user to specify the number of plies in that plate (20). Note again that, for a double shear configuration, only half the thickness of the second plate should be defined (see Figure 35). SASCJ then requests the user to specify the thickness of the metallic plate (0.25). For the laminated plate, SASCJ requests, in sequence, the average cured ply thickness (0.006), the number of distinct ply orientations (4), definition of the four orientations (0.0, +45.0, -45.0 and 90.0), and the laminate stacking sequence -- $[(45/0/-45/0)_2/0/90]_s$. SASCJ automatically assumes a metallic plate to be divided into thirty identical layers. The number of layers in a laminate is controlled

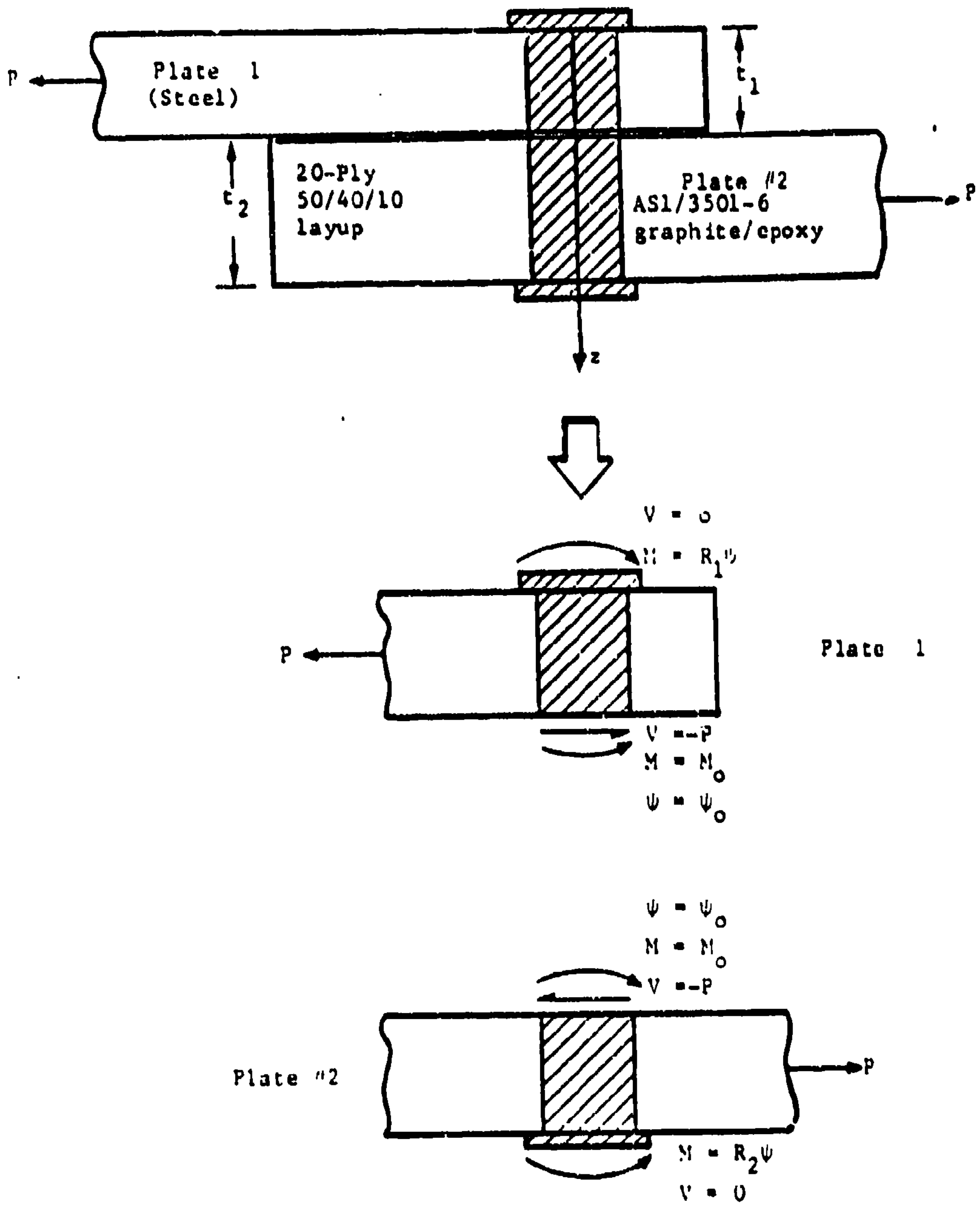


Figure 34. SASCJ Analysis of a Joint in Single Shear.

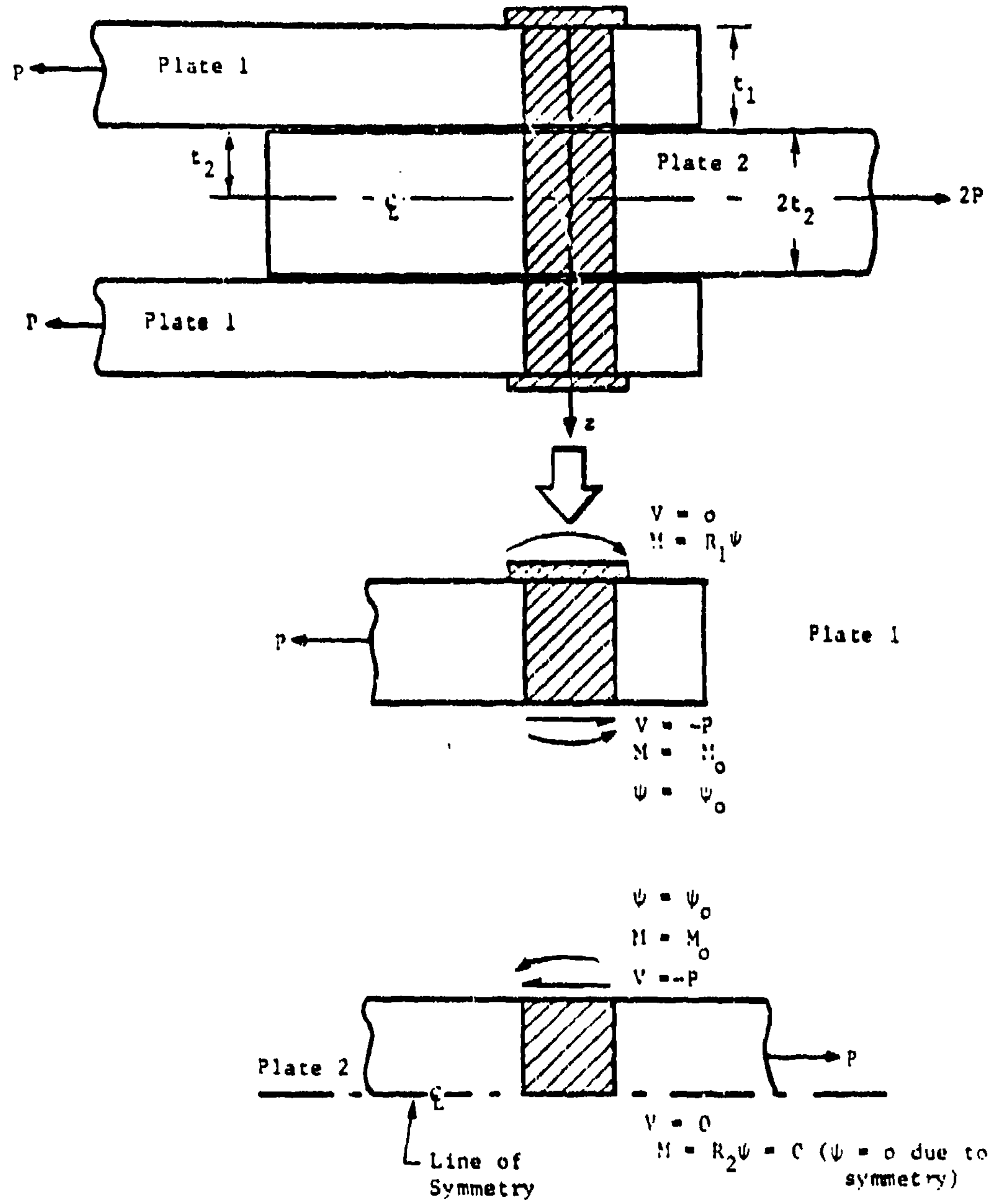


Figure 35. SASCJ Analysis of a Joint in Double Shear.

```
IKJ56006I TEMPNAME ASSUMED AS MEMBERNAME


          PROGRAM SASCJ

PROGRAM SASCJ PREDICTS FAILURE LOADS OF
MECHANICALLY FASTENED, COMPOSITE LAMINATE,
SINGLE OR DOUBLE LAP SHEAR JOINTS.

PROGRAM ASSUMES THAT INPUT PARAMETERS ARE
IN ENGLISH UNITS - LENGTHS ARE INPUT
IN INCHES AND MODULI AND STRENGTHS ARE
EXPRESSED IN PSI.


ENTER BYPASS RATIO ALPHA:

     ALPHA=0 FOR FULL BEARING
     ALPHA=1 FOR OPEN HOLE
     0<ALPHA<1 FOR GENERAL BYPASS
?
0.99
ENTER:
      1 FOR STATIC TENSION
      2 FOR STATIC COMPRESSION

?
1
ENTER:
      1 FOR SLS (SINGLE LAP SHEAR)
      2 FOR DLS (DOUBLE LAP SHEAR)

?
1
IS THE TOP PLATE A COMPOSITE OR A METAL ?
ENTER C OR M IN THE FIRST FIELD
m
INPUT MATERIAL DESCRIPTION OF THIS PLATE
     EX: AGA/3501-6
steel
IS THE BOTTOM PLATE A COMPOSITE OR A METAL ?
ENTER C OR M IN THE FIRST FIELD
c
INPUT MATERIAL DESCRIPTION OF THIS PLATE
     EX: AGA/3501-6
as1/3501-6  ((45/0/-45/0)2/0/90)s

NOTE: FOR COMPUTATIONAL PURPOSES A
METALLIC PLATE IS MODELED AS A 30 PLY
LAMINATE OF 0 DEGREE PLIES WITH ISOTROPIC
MATERIAL PROPERTIES

NOTE: NUMERICAL DESIGNATIONS FOR THE
PLATES ARE:
                 TOP PLATE    = NO 1
                 BOTTOM PLATE = NO 2

INPUT NUMBER OF PLIES IN PLATE NO    2
 (N ) OR = 2)
?
20
INPUT THICKNESS OF PLATE NO    1
?
0.25
INPUT PLY THICKNESS IN PLATE NO    2
?
0.005
INPUT NUMBER OF DISTINCT PLY ORIENTATIONS
IN PLATE NO    2
?
4

FOR PLATE NUMBER    2 :

INPUT ORIENTATION OF PLY TYPE NO    1
?
0.0
INPUT ORIENTATION OF PLY TYPE NO    2
?
45.0
INPUT ORIENTATION OF PLY TYPE NO    3
?
-45.0
INPUT ORIENTATION OF PLY TYPE NO    4
?
90.0
INPUT TYPE OF PLY IN PLATE NO    2 FROM TOP
TO BOTTOM

    PLY TYPE          ORIENTATION
       1                 0.0  DEGREES
       2               45.00 DEGREES
       3              -45.00 DEGREES
       4               90.00 DEGREES


INPUT TYPE OF PLY FOR PLY NO    1
?
2
INPUT TYPE OF PLY FOR PLY NO    2
?
1
INPUT TYPE OF PLY FOR PLY NO    3
?
3
INPUT TYPE OF PLY FOR PLY NO    4
?
1
INPUT TYPE OF PLY FOR PLY NO    5
?
2
INPUT TYPE OF PLY FOR PLY NO    6
?
1
INPUT TYPE OF PLY FOR PLY NO    7
?
3
INPUT TYPE OF PLY FOR PLY NO    8
?
1
INPUT TYPE OF PLY FOR PLY NO    9
?
1
INPUT TYPE OF PLY FOR PLY NO   10
?
```

Figure 36. Sample SASCJ Input.

```
INPUT TYPE OF PLY FOR PLY NO   11
?
4
INPUT TYPE OF PLY FOR PLY NO   12
?
1
INPUT TYPE OF PLY FOR PLY NO   13
?
1
INPUT TYPE OF PLY FOR PLY NO   14
?
3
INPUT TYPE OF PLY FOR PLY NO   15
?
1
INPUT TYPE OF PLY FOR PLY NO   16
?
2
INPUT TYPE OF PLY FOR PLY NO   17
?
1
INPUT TYPE OF PLY FOR PLY NO   18
?
3
INPUT TYPE OF PLY FOR PLY NO   19
?
1
INPUT TYPE OF PLY FOR PLY NO   20
?
2
INPUT THE ENGINEERING PROPERTIES OF PLATE NO     1
INPUT YOUNGS MODULUS AND POISSONS RATIO
?
30.0E6 0.2
INPUT THE ENGINEERING PROPERTIES OF PLATE NO     2
INPUT YOUNGS MODULI, E1 AND E2
?
18.5E6 1.9E6
INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO
?
0.85E6 0.3
INPUT MATERIAL DESCRIPTION FOR FASTENER
steel
INPUT YOUNGS MODULUS AND POISSONS RATIO FOR
THE FASTENER
?
30.0E6 0.3
INPUT THE DIAMETER OF THE FASTENER
?
0.3125

FASTENER TYPE
     ENTER:    1 FOR PROTRUDING HEAD
               2 FOR COUNTERSUNK HEAD
?
1
PLATE GEOMETRIES ARE SPECIFIED BY
INPUTTING THE COORDINATES OF THE CORNER
VERTICIES. NOTE: THE ORIGIN IS AT THE FASTENER
CENTER ; INPUT COORDINATES ACCORDINGLY

        U3                      U2
                  HOLE

                 CENTROID
          U4                     U1

APPLIED LOAD CONVENTION:
FOR PLATE NO 1 (TOP) NORMAL LOADS ARE APPLIED
BETWEEN U3 AND U4
FOR PLATE NO 2 (BOTTOM) NORMAL LOADS ARE APPLIED
BETWEEN U1 AND U2

FOR PLATE NUMBER     1 :

ENTER X,Y COORDINATES OF U   1
?
3.0 -0.9375
ENTER X,Y COORDINATES OF U   2
?
3.0 0.9375
ENTER X,Y COORDINATES OF U   3
?
-3.0 0.9375
ENTER X,Y COORDINATES OF U   4
?
-3.0 -0.9375
FOR PLATE NUMBER     2 :

ENTER X,Y COORDINATES OF U   1
?
3.0 -0.9375
ENTER X,Y COORDINATES OF U   2
?
3.0 0.9375
ENTER X,Y COORDINATES OF U   3
?
-3.0 0.9375
ENTER X,Y COORDINATES OF U   4
?
-3.0 -0.9375
SELECT FAILURE CRITERION:

  ENTER 1 FOR POINT STRESS CRITERION
  ENTER 2 FOR AVERAGE STRESS CRITERION
?
2

AVERAGE STRESS CRITERION

AO IS THE CHARACTERISTIC DISTANCE OVER WHICH
STRESSES ARE AVERAGED AND COMPARED WITH UNNOTCHED
STRENGTHS TO PREDICT FAILURE
ENTER AO VALUES CORRESPONDING TO THE THREE
PLY FAILURE MODES IN PLATE NO          1
        AONT = NET SECTION
        AOBR = BEARING
        AOSO = SHEAR OUT
INPUT AONT, AOBR, AOSO
?
0.8 0.4 0.8
```

Figure 36. Sample SASCJ Input (Continued).

```
TO AVOID LENGTHY RUN TIMES DUE TO
STRESS FIELD RECOMPUTATION SPECIFY THE
NUMBER OF ULTIMATE PLY FAILURES AFTER
WHICH JOINT FAILURE WILL BE PREDICTED
ENTER: NO OF ULTIMATE FAILURES
?
20
ENTER A0 VALUES CORRESPONDING TO THE THREE
PLY FAILURE MODES IN PLATE NO         2
           A0NT - NET SECTION
           A0BR - BEARING
           A0SO - SHEAR OUT
INPUT A0NT, A0BR, A0SO
?
0.1 0.025 0.08

FOR PLATE NUMBER      1 ENTER THE THREE STRENGTHS
REQUIRED TO PREDICT THE THREE FAILURE MODES
FNST-UNNOTCHED STRENGTH IN TENSION
FNSC-UNNOTCHED STRENGTH IN COMPRESSION
FSO-UNNOTCHED STRENGTH IN SHEAR-OUT

INPUT FNST,FNSC,FSO
?
250.0E3 300.0E3 200.0E3
FOR PLATE NO      2 ENTER FIBER ULTIMATE
STRAIN VALUES

                  EPSILON ULT IN COMPRESSION
                  EPSILON ULT IN TENSION
                  GAMMA ULT IN SHEAR
(UNITS: IN/IN)
?
0.014 0.011 0.012
SASCJ ASSUMES A BILINEAR PLY BEHAVIOR. THE
INITIAL MODULUS, K1, IS COMPUTED BY THE CODE.
THE REDUCED MODULUS, K2, FOR INITIAL FAILURE
IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED
BY THE FORMULA K2=ALPHA*K1.

FOR PLATE NUMBER      1 INPUT ALPHA VALUES FOR
NET SECTION, SHEAROUT AND BEARING FAILURE
?
001 0.1 0.1
INPUT SCALE FACTORS FOR P ULTIMATE
CALCULATION SUCH THAT P(ULT)=BETA*P(INITIAL)
INPUT BETA1 FOR NET SECTION ULTIMATE
      BETA2 FOR BEARING ULTIMATE
      BETA3 FOR SHEAROUT ULTIMATE
?
1.02 1.5 1.12
SASCJ ASSUMES A BILINEAR PLY BEHAVIOR. THE
INITIAL MODULUS, K1, IS COMPUTED BY THE CODE.
THE REDUCED MODULUS, K2, FOR INITIAL FAILURE
IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED
BY THE FORMULA K2=ALPHA*K1.

FOR PLATE NUMBER      2 INPUT ALPHA VALUES FOR
NET SECTION, SHEAROUT AND BEARING FAILURE
?
0.1 0.1 0.1
INPUT SCALE FACTORS FOR P ULTIMATE
CALCULATION SUCH THAT P(ULT)=BETA*P(INITIAL)

INPUT BETA1 FOR NET SECTION ULTIMATE
      BETA2 FOR BEARING ULTIMATE
      BETA3 FOR SHEAROUT ULTIMATE
?
1.02 1.5 1.12
```

Figure 36. Sample SASCJ Input (Concluded).

by the user. In Figure 36, each physical ply is modeled as one layer. For this sample problem, for example, the user could also specify each physical ply to be divided into two identical plies, by setting the number of plies in the laminate to be 40, the cured ply thickness to be 0.003 inch, and repeating each ply orientation in the stacking sequence twice.

Subsequent to the above information, SASCJ requests the material properties for plates 1 and 2 (Young's modulus and Poisson's ratio for a metal, and Young's modulus, shear modulus and the major Poisson's ratio for each lamina, in the fiber coordinate system). The fastener modulus, Poisson's ratio, diameter and head type (protruding head or countersunk) are requested next. Following that, the geometry of the bolted plates is defined by specifying the coordinates for the plate corners, assuming that the origin is located at the center of the fastener hole.

The last block of data addresses the selected failure criterion and the corresponding failure parameters. In the sample problem in Figure 36, the average stress failure criteria are selected for failure prediction (4). The characteristic distances for net section, bearing and shear-out modes of failure are then specified for the two plates. This is followed by the unnotched strengths for the two plates under tension, compression and inplane shear. Next, SASCJ requests the parameters that define the bilinear material behavior. These are the factors that define the modulus change after initial failure, and the ratio of the ultimate ply failure load to the initial ply failure load. Different factors may be specified for the three failure modes. Finally, the approximate ultimate shear strain value is requested for delamination prediction. A large value is generally specified for a metallic plate, to preclude the prediction of delaminations that are not applicable to these materials.

3.3 SASCJ Output Description

For the sample problem defined in Figure 36, SASCJ provides the output shown in Figure 37. The input data for the bolted plates is initially reproduced for user verification. Subsequently, the sequence of failures in the bolted laminate and the corresponding joint load levels are printed. Note that the ultimate failure of a ply (shear-out of the 45 degree plies) does not necessarily imply joint failure. In the considered sample problem, shear-out of the 0 degree plies limits the load-carrying capacity of the joint. Every ply suffers a two-stage failure as described before (Figure 31).

When executed in some systems, SASCJ could yield underflow messages after many plies have suffered total failure. This may occur when the double precision format is not followed in entering input data. Nevertheless, the user is advised to ignore these messages.

3.4 Description of SAMCJ Analysis

This section presents an overview of the strength analysis in the SAMCJ computer code, a description of the developed special finite elements, and the analytical procedure used in SAMCJ to predict fastener loads, the critical fastener or cut-out location, the corresponding joint strength and the failure mode.

A flow chart of SAMCJ operations is presented in Figure 38. As input, SAMCJ requires the user to specify how the bolted plates are divided into plain elements and elements with loaded or unloaded holes. The bolted plates are currently assumed by SAMCJ to be subjected to uniaxial tensile or compressive loading, in a single or double shear configuration. Additional input requirements for the SAMCJ code include the material properties of the bolted plates and fasteners, and the fastener size, location and torque. The material properties of the bolted laminates include the tensile and compressive failure strains in the fiber direction of the lamina, and the characteristic distances over which stresses are averaged to

```
PROGRAM SASCJ

A SINGLE LAP SHEAR JOINT WILL BE ANALYZED

WITH A PARTIALLY LOADED HOLE
BYPASS RATIO - 0.900D+00

LOADED IN STATIC TENSION

        PLATE NO     1 :

STEEL

T - 0.250D+00 INCHES

MATERIAL PROPERTIES

        E1   -0.300D+08 PSI
        E2   -0.300D+08 PSI
        G12  -0.115D+08 PSI
        NU12-0.300D+00
        NU21-0.300D+00

        PLATE NO     2 :

AS1/3501-6  ((45/0/-45/0)2/0/90)S

T - 0.120D+00 INCHES

MATERIAL PROPERTIES

        E1   -0.185D+08 PSI
        E2   -0.190D+07 PSI
        G12  -0.850D+06 PSI
        NU12-0.300D+00
        NU21-0.308D-01

        FASTENER:

STEEL

 DIAMETER - 0.313D+00 INCHES

 MATERIAL PROPERTIES

        E -0.300D+08 PSI
        NU-0.300D+00


        FAILURE ANALYSIS

AN AVERAGE STRESS CRITERION WILL BE USED

PLATE NUMBER     1

LAMINATE STRENGTH

NET SECTION ULTIMATE (TEN) -0.250D+06 PSI
NET SECTION ULTIMATE (COMP)-0.300D+06 PSI
BEARING ULTIMATE           -0.380D+06 PSI
SHEAROUT ULTIMATE          -0.200D+06 PSI

CHARACTERISTIC DISTANCES

 AONT - 0.200D+00 INCHES
 AOBR - 0.400D+00 INCHES
 AOSO - 0.500D+00 INCHES

PLATE NUMBER     2

LAMINATE STRENGTH

NET SECTION ULTIMATE (TEN) -0.123D+06 PSI
NET SECTION ULTIMATE (COMP)-0.156D+06 PSI
BEARING ULTIMATE           -0.156D+06 PSI
SHEAROUT ULTIMATE          -0.540D+05 PSI

CHARACTERISTIC DISTANCES

 AONT - 0.180D+00 INCHES
 AOBR - 0.250D-01 INCHES
 AOSO - 0.880D-01 INCHES

      GEOMETRY OF PLATE NO     1 :

COORDINATES OF CORNER VERTEXES

  -3.000,  0.938              3.000,  0.938

  -3.000, -0.938              3.000, -0.938

FASTENER HOLE DIAMETER - 0.313D+00 INCHES

E/D RATIO - 0.960D+01

W/D RATIO - 0.600D+01

      GEOMETRY OF PLATE NO     2 :

COORDINATES OF CORNER VERTEXES
```

Figure 37. SASCJ Output for the Problem Defined in Figure 36.

-3.000, 0.938 3.000, 0.938

3.000, -0.938 3.000, -0.938

FASTENER HOLE DIAMETER = 0.313D+00 INCHES

E/D RATIO = 0.960D+01

W/D RATIO = 0.600D+01

FAILURE MODE ABBREVIATIONS:

ND - NO ADDITIONAL DAMAGE AT CURRENT JOINT LOAD
DL - DELAMINATION
SO - SHEAR-OUT
BR - BEARING
NS - NET SECTION
SUD- ULTIMATE FAILURE AFTER SO AND DL
SU - ULTIMATE FAILURE IN SO
BU - ULTIMATE FAILURE IN BR
NSU- ULTIMATE FAILURE IN NS
ULT- ULTIMATE FAILURE

INCREMENT NO	JOINT LOAD	NODE	PLY TYPE		MODE
1	0.858D+04	32	0.0	DEGREE	NS
2	0.870D+04	34	0.0	DEGREE	NS
3	0.871D+04	36	0.0	DEGREE	NS
4	0.871D+04	38	0.0	DEGREE	NS
5	0.872D+04	39	0.0	DEGREE	NS
6	0.873D+04	42	0.0	DEGREE	NS
7	0.873D+04	43	0.0	DEGREE	NS
8	0.873D+04	45	0.0	DEGREE	NS
9	0.873D+04	47	0.0	DEGREE	NS
10	0.874D+04	49	0.0	DEGREE	NS
11	0.886D+04	32	0.0	DEGREE	NSU
12	0.817D+04	34	0.0	DEGREE	NSU
13	0.747D+04	36	0.0	DEGREE	NSU
14	0.676D+04	38	0.0	DEGREE	NSU
15	0.605D+04	39	0.0	DEGREE	NSU
16	0.534D+04	42	0.0	DEGREE	NSU
17	0.461D+04	43	0.0	DEGREE	NSU
18	0.388D+04	45	0.0	DEGREE	NSU
19	0.315D+04	47	0.0	DEGREE	NSU
20	0.246D+04	49	0.0	DEGREE	NSU
21	0.821D+04	31	45.000	DEGREE	NS
22	0.821D+04	35	45.000	DEGREE	NS
23	0.822D+04	46	45.000	DEGREE	NS
24	0.822D+04	50	45.000	DEGREE	NS
25	0.824D+04	33	-45.000	DEGREE	NS
26	0.824D+04	37	-45.000	DEGREE	NS
27	0.825D+04	44	-45.000	DEGREE	NS
28	0.625D+04	48	-45.000	DEGREE	NS
29	0.833D+04	31	45.000	DEGREE	NSU
30	0.627D+04	35	45.000	DEGREE	NSU
31	0.412D+04	46	45.000	DEGREE	NSU
32	0.287D+04	50	45.000	DEGREE	NSU
33	0.373D+04	33	-45.000	DEGREE	NSU
34	0.305D+04	37	-45.000	DEGREE	NSU
35	0.229D+04	44	45.000	DEGREE	NSU
36	0.143D+04	48	45.000	DEGREE	NSU

THE PREDICTED JOINT FAILURE
LOAD IS 0.886270D+04 LBS

READY

Figure 37. SASCJ Output for the Problem Defined in Figure 36 (Concluded).

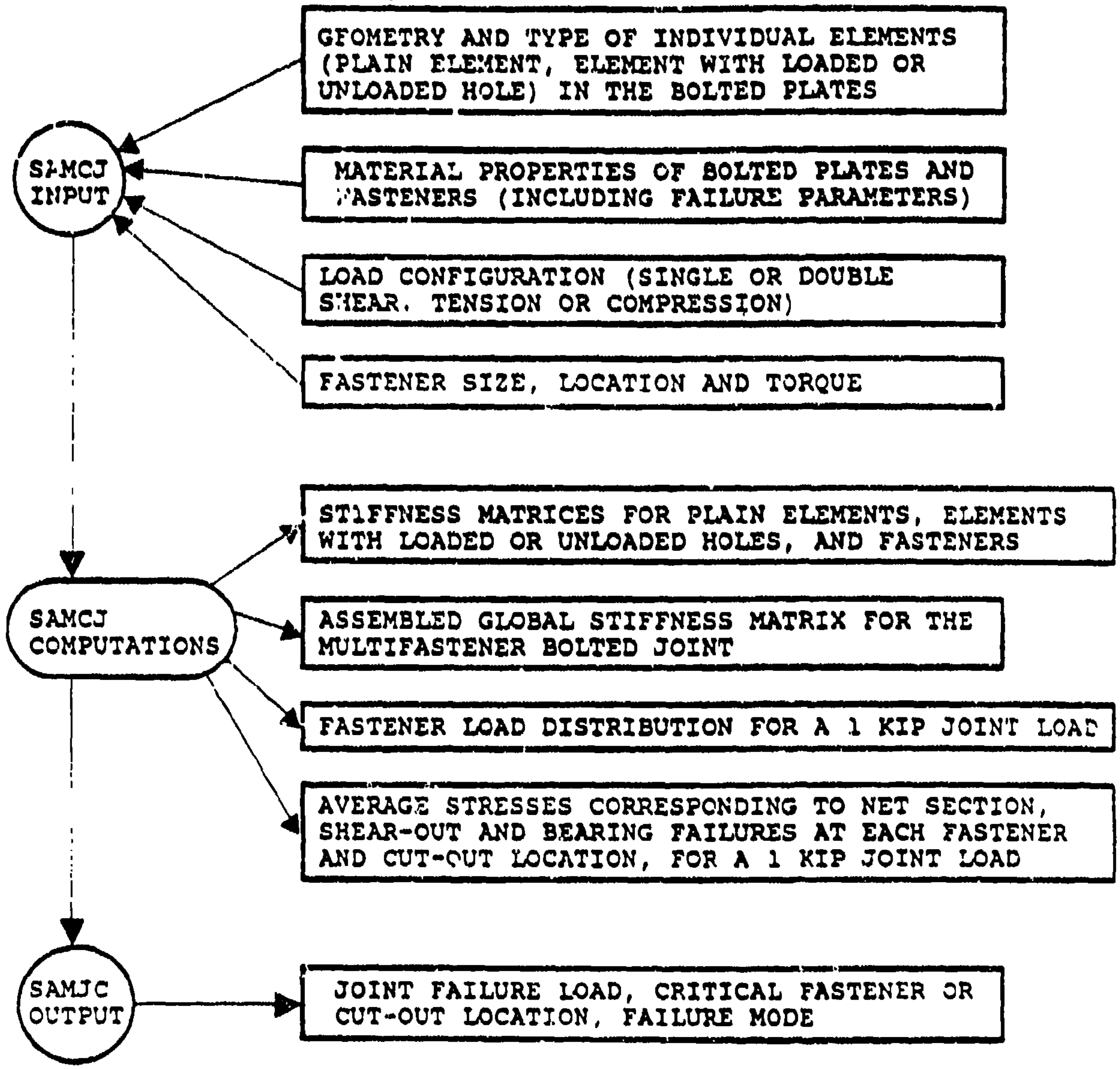


Figure 38. Flow Chart of SAMCJ Operations.

predict net section, shear-out and bearing failures at the fastener or cut-out location.

With the above input, SAMCJ performs the following computations. It initially generates stiffness matrices for all the special finite elements, namely, plain elements, elements with loaded or unloaded holes, and effective fastener elements (see Reference 7). The individual stiffness matrices are subsequently assembled to obtain the global stiffness matrix for the bolted joint. A 1 kip uniaxial tensile or compressive joint load is imposed on the left end of the top plate, in accordance with the input instructions (see Figure 33). The nodes at the right end of the bottom plate are constrained from translating in the load direction, and one of these nodes is also constrained in the transverse direction, to preclude all rigid body translations. The solution to this finite element formulation of the bolted joint provides the axial and transverse components of the load at every fastener location, corresponding to a 1 kip joint load. Also computed are the average net section, shear-out and bearing stresses at every fastener and cut-out location, corresponding to a 1 kip joint load.

SAMCJ provides, as output, the failure value of the uniaxial joint load, the critical fastener or cut-out location, and the joint failure mode. These are obtained as follows. The tensile, compressive and shear strengths of the plain laminates are computed based on the input tensile and compressive failure strains in the fiber direction of the lamina. The ratios of the averaged stresses to the corresponding unnotched laminate strengths, at selected locations around each fastener and cut-out boundary, are compared to predict the failure mode, the critical fastener or cut-out location and the joint failure load. SAMCJ predicts net section, shear-out and bearing modes of failure at the laminate level. In the SASCJ code, similar failure predictions for single fastener joints in composites are made at the lamina level. Consequently, the failure parameters (characteristic distances for

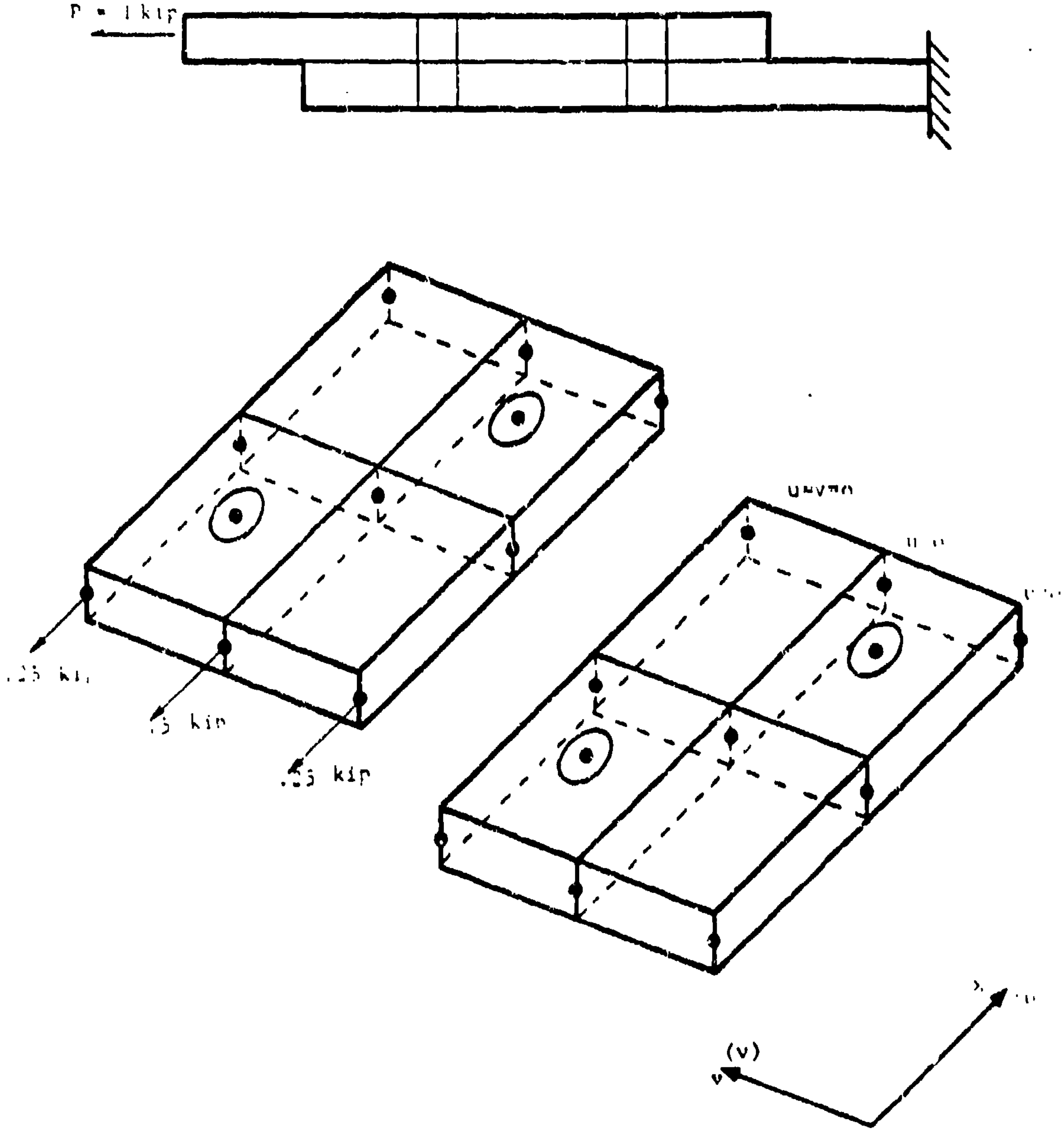


Figure 39. Application of Load and Displacement Boundary Conditions in the SAMCJ Code.

the three failure modes) used with SAMCJ are different from those used with SASCJ.

The incorporation of the transverse effective fastener stiffness values provides SAMCJ the capability to account for fastener flexibility, torque, and load eccentricity (single versus double shear load transfer). The FDFA code, developed in Reference 6, is used to compute the effective fastener transverse stiffnesses, along and perpendicular to the load direction (see Reference 7). The effect of the laminate stacking sequence is also accounted for in this analysis. SAMCJ executes FDFA twice to account for the layup variation (by 90 degrees) from the loading direction to the perpendicular direction.

SAMCJ accounts for stress concentration interaction effects introduced by neighboring cut-outs, free edges and proximate fastener locations. This is made possible by the use of the FIGEOM stress analysis, developed in Reference 6, to generate element stiffness matricess (see Reference 7). FIGEOM accounts for finite planform plate dimensions through a boundary collocation solution procedure (see Reference 6).

SAMCJ computes the magnitude and the orientation of the load at each fastener location. It is a two-dimensional load distribution analysis that does not rely on an experimental measurement of "joint stiffness." In a design situation, many fastener arrangements can be analytically and economically evaluated by SAMCJ to arrive at the best fastener pattern for the assumed loading conditions.

When the bolted plates are tapered, the SAMCJ user can input equivalent uniform thickness elements to approximate the tapering effect (see Figure 40). Adjacent elements in the tapered plate will have different thickness values. This feature is essential in the analysis of practical structural joints.

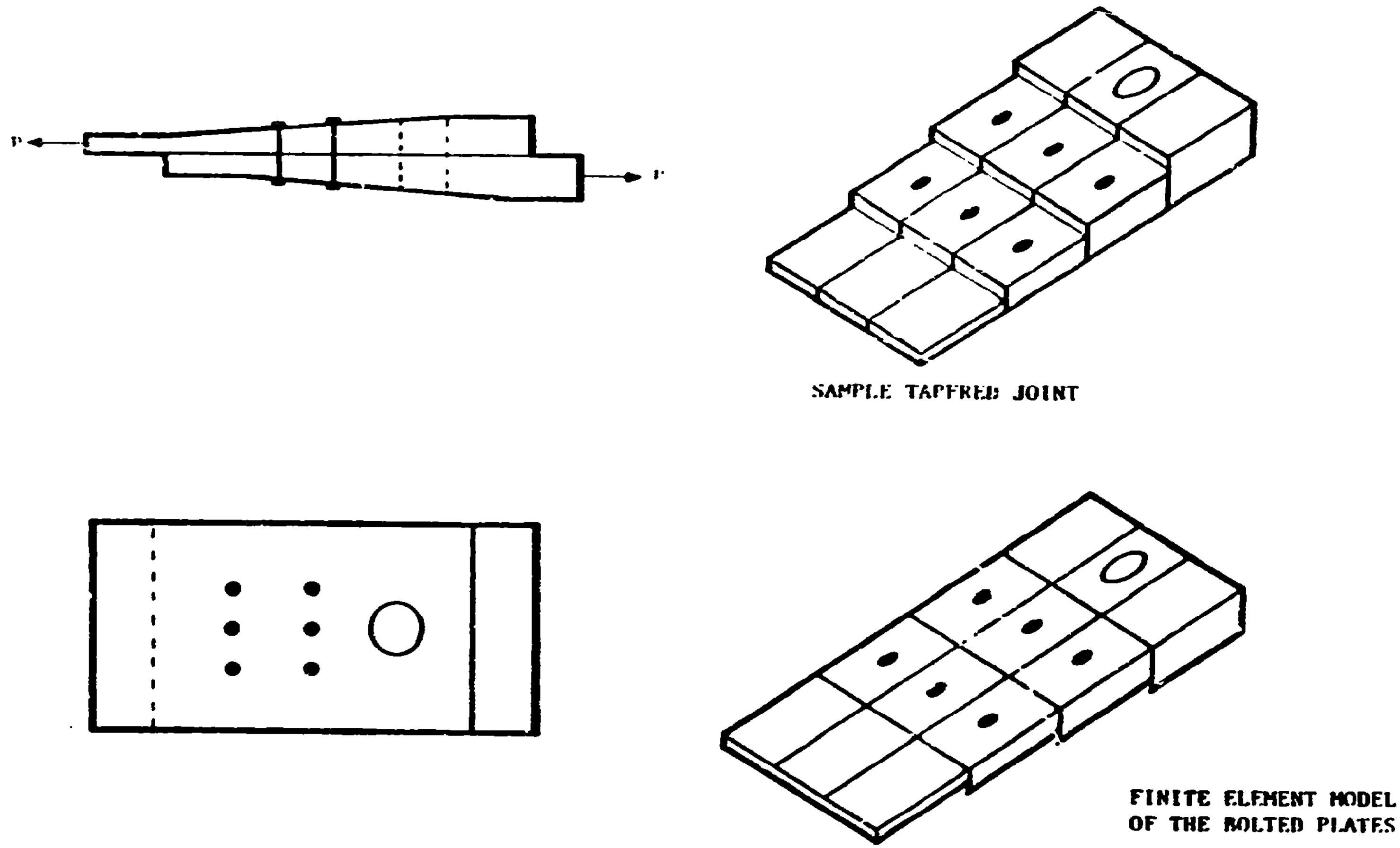
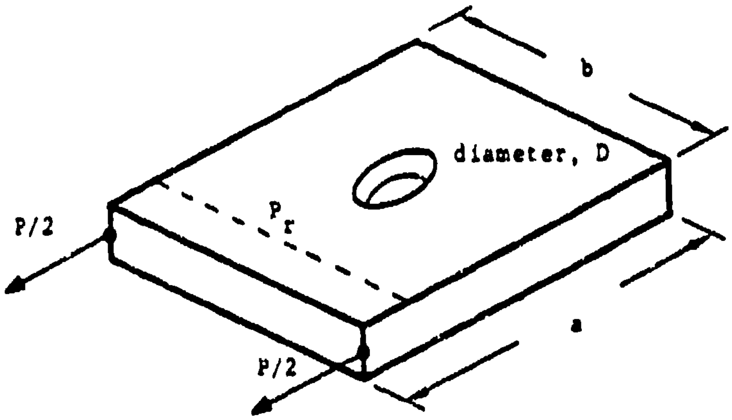


Figure 40. Finite Element Model of a Sample Tapered Bolted Joint.

SAMCJ has been developed for the strength prediction of bolted laminated structural parts. It currently assumes that the selected fasteners preclude fastener failure. Also, it applies the same failure procedure to both the bolted plates, accounting for net section, shear-out and bearing failures via the averaged stress failure criteria applied at the laminate level. Joint failure is assumed to be a one-step (catastrophic) process. The strength of a bolted plate corresponds to the initial failure at a fastener or a cut-out location, in the bearing, shear-out or net section failure.

The unnotched laminate strengths, under tension, compression and inplane shear, are computed by SAMCJ based on input fiber-directional failure strain values (tensile and compressive). Laminate strengths under N_x and N_{xy} loadings (inplane normal and shear stress resultants, respectively) are assumed to correspond to first fiber failure in a ply. This simplistic strength prediction procedure introduces inaccuracies that have been acknowledged and discussed in the literature. Nevertheless, SAMCJ adopts this procedure for lack of a validated alternative.

Despite its versatility, SAMCJ has limitations that the user should be aware of. Reference 7 discusses the limitations of the five-noded (10 degrees of freedom) loaded hole element and the four-noded (8 degrees of freedom) unloaded hole element. In addition, when dividing a bolted plate into many elements (loaded or unloaded hole elements, as well as plain elements), it is advisable to maintain element geometries that do not render the generated stiffness matrices inaccurate. Figure 41 presents results from a study conducted on a singly-fastened metallic plate. P_r is the recovered load that is obtained by integrating the stresses along a line transverse to the load direction as shown in Figure 41. P is the applied load or the sum of the nodal loads (especially in the interior elements in a general multifastened plate). The recovered load (P_r) approaches the applied load value (P) when the plate aspect ratio (a/b) increases beyond unity, and when a/D and b/D have a minimum value of approximately three. In predicting failure in



D = 0.3125 inch

a/D	b/D	P_r/P
1.6	1.6	5.38
3.2	1.6	2.27
6.4	1.6	1.57
16.0	1.6	1.29
1.6	3.2	1.24
3.2	3.2	1.76
6.4	3.2	1.37
16.0	3.2	1.16
1.6	6.4	-0.0995
3.2	6.4	0.989
6.4	6.4	1.23
16.0	6.4	1.16
3.2	16.0	-0.46
6.4	16.0	0.029
16.0	16.0	1.23

Figure 41. Element Load Recovery for Various a/D and b/D Ratios.

the net section, bearing and shear-out modes, the computed average stress values are multiplied by P/P_r, to remove geometry (modeling) effects from the computed stresses.

3.5 SAMCJ Input Description

To familiarize the user with SAMCJ input requirements, a sample problem is presented here (see Figure 42). The sample problem considers a six fastener composite-to-metal joint, with a one inch diameter circular cut-out adjacent to the first row of fasteners. Figure 42 presents the assumed nine element model of each of the two bolted plates, analyzed by SAMCJ. Figure 43 presents SAMCJ requests and user input in response to these requests, for the sample problem in Figure 42.

Though self-explanatory, the interactively entered SAMCJ input in Figure 43, for the sample problem in Figure 42, is described here for completeness. The first entry (1) identifies the loading configuration to be a single shear configuration. The second entry (1) identifies the load to be in static tension. The next two entries say that the top plate is a metal (M), identified as "Aluminum." The two entries following these say that the bottom plate is a composite laminate (C), identified as follows: "(45/0/-45/0)2/0/90)2s." Subsequently, the Young's modulus (10.0D6) and Poisson's ratio (0.3) for aluminum, and the fiber-directional, transverse and shear moduli and Poisson's ratio (18.5D6, 0.85D6 and 0.3, respectively) for the composite lamina are input. The next five entries specify that four (4) different fiber orientations are present in the laminate (0, 45, -45 and 90 degrees with respect to the loading direction). The following three entries say that the elements in the bottom plate contain one (1) layup of forty (40) plies, of 0.006175 inch thickness each. The stacking sequence for this layup is input next, where 1, 2, 3 and 4 refer to 0, 45, -45 and 90 degree fiber orientations, respectively. Subsequently, the fastener is identified as "Steel," and its Young's modulus, Poisson's ratio, and head type (30.0D6, 0.3, 0.3125 and

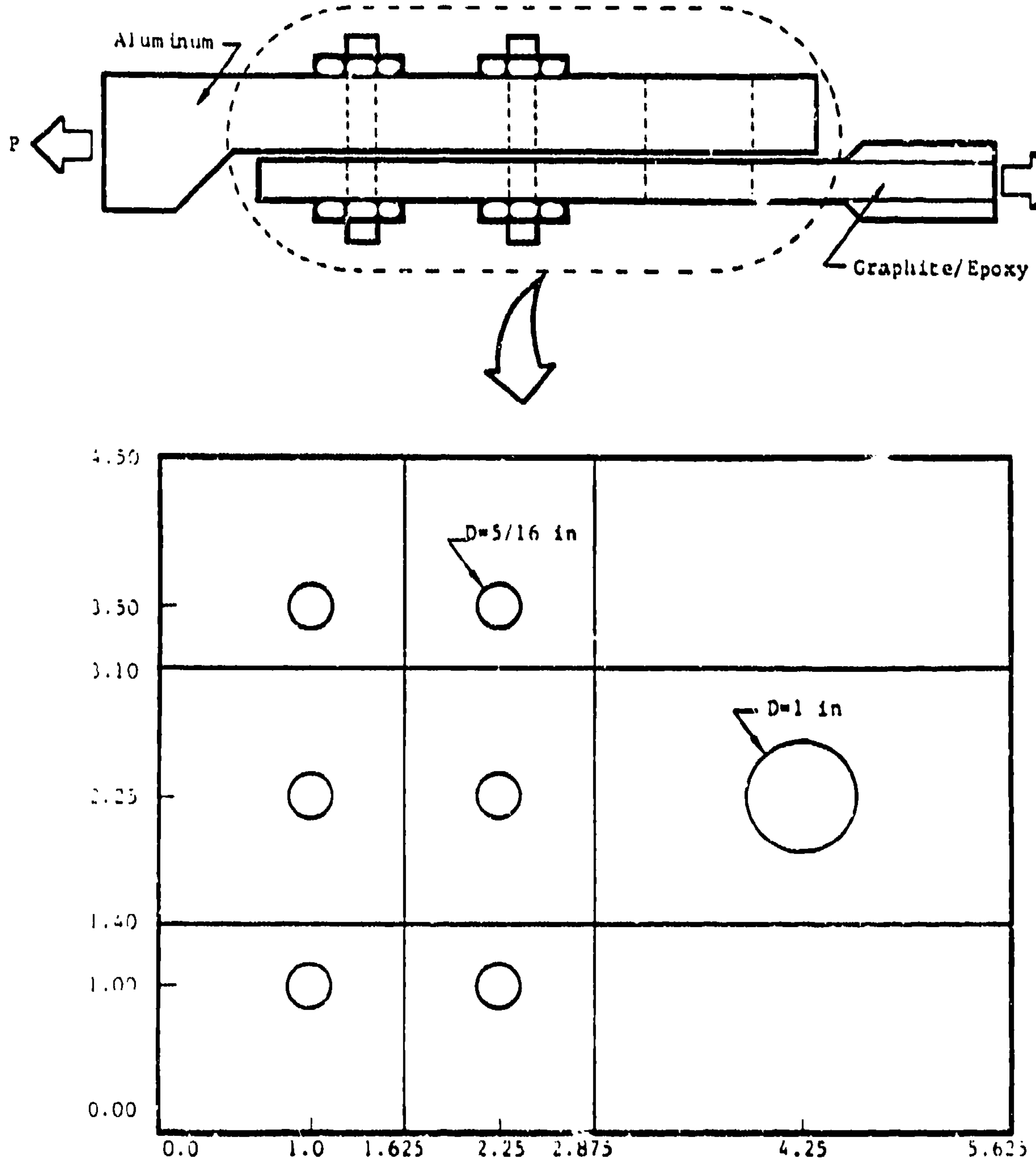


Figure 42. Nine Element Model of Each of the two Bolted Plates in the Sample Joint.

```
call samcj
 1  56006I TEMPNAME ASSUMED AS MEMBERNAME


          PROGRAM SAMCJ

PROGRAM SAMCJ PREDICTS THE FAILURE LOAD, FAILURE
LOCATION, AND FAILURE MODE IN MULTIPLY-FASTENED,
SINGLE OR DOUBLE LAP COMPOSITE SHEAR JOINTS.

THE ANALYSIS ASSUMES THAT INPUT PARAMETERS ARE
SPECIFIED IN ENGLISH UNITS - LENGTH IN INCHES,
MODULI AND STRENGTHS IN PSI.


ENTER:
      1 FOR SLS (SINGLE LAP SHEAR)
      2 FOR DLS (DOUBLE LAP SHEAR)

?
1
ENTER:
      1 FOR STATIC TENSION
      2 FOR STATIC COMPRESSION

?
1
IS THE TOP PLATE A COMPOSITE OR A METAL?
ENTER C OR M IN THE FIRST FIELD
m
INPUT MATERIAL DESCRIPTION OF THIS PLATE
     EX: AS4/3501-6
aluminum
IS THE BOTTOM PLATE A COMPOSITE OR A METAL?
ENTER C OR M IN THE FIRST FIELD
c
INPUT MATERIAL DESCRIPTION OF THIS PLATE
     EX: AS4/3501-6
((45/0/-45/0)2/0/90)s

NOTE: FOR COMPUTATIONAL PURPOSES A
METALLIC PLATE IS MODELED AS A 30 PLY
LAMINATE OF 0 DEGREE PLIES WITH ISOTROPIC
MATERIAL PROPERTIES

INPUT THE ENGINEERING PROPERTIES OF THE TOP PLATE
INPUT YOUNGS MODULUS AND POISSONS RATIO
?
10.0E6 0.3
INPUT THE ENGINEERING PROPERTIES OF THE BOTTOM PLATE
INPUT YOUNGS MODULI, E1 AND E2
?
18.5E6 1.9E6
INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO
?
0.85E6 0.3
INPUT TOTAL NUMBER OF DISTINCT PLY
ORIENTATIONS IN THE BOTTOM PLATE
?
4
INPUT ORIENTATION OF PLY TYPE NO     1
?
0.0
INPUT ORIENTATION OF PLY TYPE NO     2
?
45.0
INPUT ORIENTATION OF PLY TYPE NO     3
?
-45.0
INPUT ORIENTATION OF PLY TYPE NO     4
?
90.0

THICKNESS VARIATIONS MAY BE APPROXIMATED
BY ASSIGNING DIFFERENT LAYUPS TO ELEMENTS
IN A COMPOSITE PLATE OR BY SPECIFYING DIFFERENT
THICKNESSES TO ELEMENTS IN A METALLIC PLATE


ENTER NUMBER OF DIFFERENT LAYUPS IN THE
BOTTOM PLATE
?
1
ENTER NUMBER OF PLIES IN LAYUP NO        1
?
40
ENTER PLY THICKNESS FOR THIS LAYUP
?
0.006175
ENTER SEQUENCE OF PLY TYPES FROM TOP TO BOTTOM
?
2
?
1
?
3
?
1
?
2
?
1
?
3
?
1
?
1
?
4
?
2
?
1
?
2
?
1
```

Figure 43. SAMCJ Input for the Sample Problem in Figure 42.

```
0.3125

FASTENER TYPE
      ENTER:      1 FOR PROTRUDING HEAD
                  2 FOR COUNTERSUNK HEAD
?
1

GRID LAYOUT:

ENTER NUMBER OF GRIDS IN TOP PLATE
?
22

ENTER       22 GRID POINTS

FORMAT: GRID ID, X AND Y COORDINATES
?
101 0.0 0.0
?
102 0.0 1.40
?
103 0.0 3.1
?
104 0.0 4.5
?
105 1.0 1.0
?
106 1.0 2.25
?
107 1.0 3.5
?
108 1.625 0.0
?
109 1.625 1.4
?
110 1.625 3.1
?
111 1.625 4.5
?
112 2.25 1.0
?
113 2.25 2.25
?
114 2.25 3.5
?
115 2.875 0.0
?
116 2.875 1.4
?
117 2.875 3.1
?
118 2.875 4.5
?
119 5.025 0.0
?
120 5.025 1.4
```

```
FASTENER DESCRIPTION:

INPUT MATERIAL DESCRIPTION FOR FASTENER
steel
INPUT YOUNGS MODULUS AND POISSONS RATIO FOR
THE FASTENER
?
30.0E6 0.3
INPUT THE DIAMETER OF THE FASTENER
?
```

Figure 43. SAMCJ Input for the Sample Problem in Figure 42 (Continued).

```
?
121 5.625 3.1
?
122 5.625 4.5
ENTER NUMBER OF GRIDS IN BOTTOM PLATE
?
22
ENTER       22 GRID POINTS
FORMAT: GRID ID, X AND Y COORDINATES
?
201 0.0 0.0
?
202 0.0 1.4
?
203 0.0 3.1
?
204 0.0 4.5
?
205 1.0 1.0
?
206 1.0 2.25
?
207 1.0 3.5
?
208 1.625 0.0
?
209 1.625 1.4
?
210 1.625 3.1
?
211 1.625 4.5
?
212 2.25 1.0
?
213 2.25 2.25
?
214 2.25 3.5
?
215 2.875 0.0
?
216 2.875 1.4
?
217 2.875 3.1
?
218 2.875 4.5
?
219 5.625 0.0
?
220 5.625 1.4
?
221 5.625 3.1
?
222 5.625 4.5
ELEMENT DESCRIPTION:

PLANAR ELEMENTS ARE NUMBERED
CLOCKWISE AS SHOWN:

      N2    N3
         N5
      N1    N4

ELEMENT TYPES ARE DESIGNATED AS FOLLOWS:

 4 NODE PLAIN ELEMENT           TYPE NO. 1
 5 NODE LOADED HOLE ELEMENT     TYPE NO. 2
 4 NODE OPEN HOLE ELEMENT       TYPE NO. 3

(NOTE: ENTER N5=0 FOR FOUR NODE ELEMENTS)

ENTER NUMBER OF ELEMENTS IN TOP PLATE
?
9
FOR ELEMENT NO    1 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
101 101 102 109 108 105 2
ENTER ELEMENT THICKNESS
?
0.5
FOR ELEMENT NO    2 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
102 102 103 110 109 106 2
ENTER ELEMENT THICKNESS
?
0.5
FOR ELEMENT NO    3 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
103 103 104 111 110 107 2
ENTER ELEMENT THICKNESS
?
0.5
FOR ELEMENT NO    4 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
104 108 109 116 115 112 2
ENTER ELEMENT THICKNESS
?
0.5
FOR ELEMENT NO    5 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
105 109 110 117 116 113 2
ENTER ELEMENT THICKNESS
?
0.5
FOR ELEMENT NO    6 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
106 110 111 118 117 114 2
ENTER ELEMENT THICKNESS
?
0.5
```

Figure 43. SAMCJ Input for the Sample Problem in Figure 42 (Continued).

```
FOR ELEMENT NO    7 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE

107 115 116 120 119 0 1
ENTER ELEMENT THICKNESS
?
0.8
FOR ELEMENT NO    8 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
108 116 117 121 120 0 3
ENTER ELEMENT THICKNESS, X AND Y COORDINATES
OF OPEN HOLE AND HOLE RADIUS
?
0.5 4.25 2.25 0.5
FOR ELEMENT NO    9 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
109 117 118 122 121 0 1
ENTER ELEMENT THICKNESS
?
0.5

ENTER NUMBER OF ELEMENTS IN BOTTOM PLATE
?
9
FOR ELEMENT NO   10 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
201 201 202 209 208 205 2
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   11 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
202 202 203 210 209 206 2
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   12 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
203 203 204 211 210 207 2
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   13 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
204 208 209 216 215 212 2
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   14 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
205 209 210 217 216 213 2
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   15 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
206 210 211 218 217 214 2
ENTER ELEMENT LAYUP NO
?
1

FOR ELEMENT NO   16 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
207 215 216 220 219 0 1
ENTER ELEMENT LAYUP NO
?
1
FOR ELEMENT NO   17 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
208 216 217 221 220 0 3
ENTER ELEMENT LAYUP NUMBER, X AND Y
COORDINATES OF THE OPEN HOLE AND THE HOLE
RADIUS
?
1 4.25 2.25 0.5
FOR ELEMENT NO   18 ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE
?
209 217 218 222 221 0 1
ENTER ELEMENT LAYUP NO
?
1

FASTENERS ARE MODELED BY EFFECTIVE
FASTENER ELEMENTS WHICH PROVIDE THE
ELASTIC LINK BETWEEN THE TOP AND
BOTTOM PLATES

ENTER NUMBER OF FASTENERS IN JOINT
?
6


EFFECTIVE FASTENER ELEMENTS ARE
NUMBERED AS SHOWN:
                          N1 (TOP PLATE)

                          N2 (BOTTOM PLATE)

WHERE N1 AND N2 CORRESPOND TO THE CENTRAL
NODES IN LOADED HOLE ELEMENTS

FORMAT: ELEMENT ID, N1, N2
ENTER ELEMENT NO    1
?
101 105 205
ENTER ELEMENT NO    2
?
102 106 206
ENTER ELEMENT NO    3
?
103 107 207
ENTER ELEMENT NO    4
?
104 112 212
ENTER ELEMENT NO    5
?
105 113 213
ENTER ELEMENT NO    6
?
106 114 214
```

Figure 43. SAMCJ Input for the Sample Problem in Figure 42 (Continued).

```
TO REDUCE RUN TIMES, ELEMENTS MAY BE
GROUPED INTO SETS WHICH WILL BE ASSIGNED
IDENTICAL STIFFNESS MATRICIES
ENTER:      1 TO USE THIS OPTION
            2 OTHERWISE
?
1
FOR THE TOP PLATE INPUT NUMBER OF GROUPS
FOR THE EFFECTIVE FASTENER, LOADED HOLE, UNLOADED
HOLE AND PLAIN ELEMENT
(INPUT 0 IF ELEMENT TYPE IS NOT USED)
?
1 6 1 1
GROUPING OF EFFECTIVE FASTENER ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
6
ENTER        6 ELEMENT IDS
?
101 102 103 104 105 106

GROUPING OF LOADED HOLE ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
1
INPUT        1 ELEMENT IDS
?
101
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        2
?
1
INPUT        1 ELEMENT IDS
?
102
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        3
?
1
INPUT        1 ELEMENT IDS
?
103
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        4
?
1
INPUT        1 ELEMENT IDS
?
104
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        5
?
1
INPUT        1 ELEMENT IDS
?
105
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        6
?
1
INPUT        1 ELEMENT IDS
?
106
GROUPING OF UNLOADED HOLE ELEMENTS
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
1
ENTER        1 ELEMENT IDS
?
108
GROUPING OF PLAIN ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
2
ENTER        2 ELEMENT IDS
?
107 108

FOR THE BOTTOM PLATE INPUT NUMBER OF GROUPS
FOR THE LOADED HOLE,UNLOADED HOLE, AND PLAIN
ELEMENTS
(INPUT 0 IF AN ELEMENT TYPE IS NOT USED)
?
6 1 1

GROUPING OF LOADED HOLE ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
1
INPUT        1 ELEMENT IDS
?
201
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        2
?
1
INPUT        1 ELEMENT IDS
?
202
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        3
?
1
INPUT        1 ELEMENT IDS
?
203
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        4
?
1
INPUT        1 ELEMENT IDS
?
204
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        5
?
1
INPUT        1 ELEMENT IDS
?
205
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        6
?
1
INPUT        1 ELEMENT IDS
?
206
```

Figure 43. SAHCJ Input for the Sample Problem in Figure 42 (Continued).

```
GROUPING OF UNLOADED HOLE ELEMENTS
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
1
ENTER        1 ELEMENT IDS
248
GROUPING OF PLAIN ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER        1
?
2
ENTER        2 ELEMENT IDS
207 240

INPUT DATA FOR FAILURE ANALYSIS:

ENTER METALLIC STRENGTHS:
  TENSILE STRENGTH
  COMPRESSIVE STRENGTH
  SHEAR STRENGTH
?
250.003 250.003 250.003

AN AVERAGE STRESS CRITERIA IS USED TO
PREDICT FAILURE. AO VALUES ARE REQUIRED AS
CHARACTERISTIC DISTANCES OVER WHICH STRESSES
ARE TO BE AVERAGED AND COMPARED TO UNNOTCHED
LAMINATE STRENGTHS TO PREDICT FAILURE

ENTER AO VALUES FOR STRESS AVERAGING
FOR EACH FAILURE MODE IN PLATE NO     1
        AONT = NET SECTION
        AOBR = BEARING
        AOSO = SHEAROUT
?
0.5 0.5 0.5
ENTER FIBER ULTIMATE STRAIN VALUES
IN PLATE NO       2
          EPSILON ULT IN COMPRESSION
          EPSILON ULT IN TENSION
          GAMMA ULT IN SHEAR

?
0.0175 0.012 0.012

AN AVERAGE STRESS CRITERIA IS USED TO
PREDICT FAILURE. AO VALUES ARE REQUIRED AS
CHARACTERISTIC DISTANCES OVER WHICH STRESSES
ARE TO BE AVERAGED AND COMPARED TO UNNOTCHED
LAMINATE STRENGTHS TO PREDICT FAILURE

ENTER AO VALUES FOR STRESS AVERAGING
FOR EACH FAILURE MODE IN PLATE NO     2
        AONT = NET SECTION
        AOBR = BEARING
        AOSO = SHEAROUT
?
0.1 0.05 0.05
```

Figure 43. SAMCJ Input for the Sample Problem in Figure 42 (Concluded).

protruding head) are input.

Twenty-two (22) grid points each are specified in the top and bottom plates (101 to 122 and 201 to 222, respectively), along with their x and y coordinates (see Figure 42). Following this, nine (9) elements are specified in each plate, along with their nodal connectivity and element type information. Nodal connectivity is specified starting from the bottom left node, going clockwise around the element boundary, and ending at the fastener (internal) node. Element 101 in the top plate, for example, has 101, 102, 109 and 108 as its corner nodes, and 105 as its fastener node. The fifth node will be entered as 0 for plain and unloaded hole elements. The element type information follows the fifth node identification. It is 1, 2 and 3 for plain, loaded hole and unloaded hole elements, respectively. The element definitions are succeeded by the definition of six (6) effective fasteners (101 to 106). Fastener 101, for example, is identified as a fastener that connects node 105 in the top plate to node 205 in the bottom plate.

Following the above input, additional element data are specified for the two plates. These include the element thicknesses (for metallic plates) or layup identification number (for laminated plates), for plain and loaded hole elements, with additional information (x and y coordinates of the hole center and the hole radius) for unloaded hole elements. For the sample problem in Figure 42, all the elements in the top plate (metal) are specified to be 0.50 inch thick, and all the elements in the bottom plate (composite) are specified to contain the stacking sequence identified as one (1). Elements 108 and 208 specify the cut-out size and location. The one (1) following this states that groups of identical elements will be specified in the two plates. If two (2) is entered here, all elements will be assumed to be different from one another, resulting in larger computational costs. The entry "1 6 1 1" refers to the number of groups of effective fasteners, loaded hole, unloaded hole and plain elements, respectively, in the top plate. A zero (0) specifies the absence of an element type.

The number of elements in each group, and the corresponding element numbers, are input subsequently. Following this, the number of groups of loaded hole, unloaded hole and plain elements in the bottom plate (6, 1 and 1, respectively) is entered.

The last four lines of input introduce the failure parameters for the materials in the two plates. For metallic plates, the tensile, compressive and shear strengths (250.0D3 each), and the averaging distances for the net section, bearing and shear-out modes of failure (0.5 each) are input. Since the joints were designed to fail the laminated plates, and SAMCJ was developed primarily for the prediction of the strength of bolted laminates, the failure parameters for the metallic plates were input to be arbitrairly high. This information is followed by the failure parameters for the bottom (composite) plate. The first line specifies the fiber directional failure strains for the material under tension (0.012) and compression (0.0175). These values are used by SAMCJ to compute the unnotched laminate tensile, compressive and shear strengths, based on laminated plate theory and the assumption of laminate failure corresponding to the first fiber failure in any of its plies. The last line in Figure 43 specifies the distance over which the longitudinal (0.10 and 0.25) and shear (0.25) stress components are averaged, to predict net section, bearing and shear-out modes of failure, respectively.

3.6 SAMCJ Output Description

For the sample problem introduced in Section 3.5, the SAMCJ code yields the output presented in Figure 44. The initial part of the output reprints critical user-supplied information for verification purposes. Subsequently, SAMCJ prints the x and y components of the element nodal forces for all the elements in the bolted plates. This is followed by a list of the computed joint load levels that correspond to the three failure modes (net section, shear-out and bearing) at every loaded and unloaded hole element location. The smallest among these loads yields the joint failure

```
        PROGRAM SAMCJ

A SINGLE LAP SHEAR PANEL WILL BE ANALYZED
LOADED IN STATIC TENSION
        PLATE NO     1 :
ALUMINUM
MATERIAL PROPERTIES
        E1  =0.1000+08 PSI
        E2  =0.1000+08 PSI
        G12 =0.1670+08 PSI
        MU12=0.3000+00
        MU21=0.3000+00
        PLATE NO     2 :
AS1 3501-6
MATERIAL PROPERTIES
        E1  =0.1850+08 PSI
        E2  =0.1800+07 PSI
        G12 =0.8540+06 PSI
        MU12=0.3000+00
        MU21=0.3000-01
        FASTENER DESCRIPTION:
STEEL
 DIAMETER = 0.3130+00 INCHES
 MATERIAL PROPERTIES
        E =0.3000+08 PSI
        MU=0.3000+00

        FAILURE ANALYSIS
AN AVERAGE STRESS CRITERION WILL BE USED
PLATE NUMBER    1
METALLIC STRENGTHS
TENSILE STRENGTH      = 0.8000+05
COMPRESSIVE STRENGTH  = 0.8600+05
SHEAR STRENGTH        = 0.8500+05

CHARACTERISTIC DISTANCES
AONT = 0.5000+00 INCHES
AOBR = 0.5000+00 INCHES
AOSG = 0.5000+00 INCHES
 PLATE NUMBER    2

 FIBER STRAIN ULTIMATES
 EPSILON ULT COMP = 0.175D-01
 EPSILON ULT TEN  = 0.162D-01
 GAMMA ULT SHEAR  = 0.120D-01

 CHARACTERISTIC DISTANCES
AONT = 0.1000+00 INCHES
AOBR = 0.2500+00 INCHES
AOSG = 0.2500+00 INCHES

PAUSE FOR STIFFNESS MATRIX CALCULATIONS

        ELEMENT FORCES
     (P APPLIED = 10,640 LBS)

ELEMENT ID     101
     GRID          FX          FY
      101     -.1560+03   0.173D-12
      102     -.1840+03   -.284D+02
      109     0.895D+02   0.156D+02
      108     0.757D+02   0.126D+02
      105     0.165D+03   0.105D+00
ELEMENT ID     102
     GRID          FX          FY
      102     -.1000+03   0.884D+00
      103     -.1660+03   -.284D+00
      110     0.612D+02   -.5:6D+01
      109     0.654D+02   0.479D+01
      106     0.181D+03   0.418D+00
ELEMENT ID     103
     GRID          FX          FY
      103     -.100D+03   0.884D+00
      104     -.1580+03   -.864D-13
      111     0.786D+02   -.133D+02
```


Figure 44. SAMCJ Output for the Problem Defined in Figure 43.

GRID	FX	FY
110	0.100D-01	-.158D+02
107	0.165D+03	-.154D+00

ELEMENT ID 104

GRID	FX	FY
108	-.757D+02	-.126D+02
109	-.918D+02	-.148D+02
116	0.245D+01	0.188D+02
115	0.713D+01	0.787D+01
112	0.158D+03	0.245D+01

ELEMENT ID 105

GRID	FX	FY
109	-.776D+02	-.553D+01
110	-.767D+02	0.387D-01
117	-.106D+02	0.646D+01
116	-.969D+01	-.815D+01
113	0.175D+03	0.334D+01

ELEMENT ID 106

GRID	FX	FY
110	-.925D+02	0.163D+02
111	-.756D+02	0.133D+02
118	0.705D+01	-.739D+0[illegible]
117	0.365D+01	-.164D+02
114	0.157D+03	-.619D+01

ELEMENT ID 107

GRID	FX	FY
115	-.713D+01	-.703D+01
116	0.342D+01	0.343D+01
120	0.371D+01	0.363D+01
119	0.792D-12	-.341D-12

ELEMENT ID 108

GRID	FX	FY
116	0.378D+01	-.132D+02
117	0.368D+01	0.133D+02
121	-.373D+01	0.353D+01
120	-.371D+01	-.363D+01

ELEMENT ID 109

GRID	FX	FY
117	0.333D+01	-.350D+01
118	-.705D+01	0.739D+01
122	-.152D-11	-.117D-11
121	0.373D+01	-.359D+01

ELEMENT ID 201

GRID	FX	FY
221	0.165D [illegible]	0.[illegible]D-13
222	0.[illegible]	0.6[illegible]D+01
203	0.1[illegible]	0.32[illegible]D+01
208	0.544[illegible]	-.93[illegible]D+01
225	-.165[illegible]	-.185D+00

ELEMENT ID 202

GRID	FX	FY
202	-.906D+0[illegible]	-.614D+01
203	-.106D+02	0.676D+01
210	0.101D+03	0.81[illegible]D+01
209	0.994D+02	-.577D+01
206	-.181D+03	-.408D+00

ELEMENT ID 203

GRID	FX	FY
203	0.106D+02	-.619D+01
204	0.238D-12	-.462D-13
211	0.540D+02	0.851D+01
210	0.100D+03	-.261D+01
227	-.165D+03	0.138D+00

ELEMENT ID 204

GRID	FX	FY
208	-.540D+02	0.919D+01
209	-.803D+02	0.247D+02
216	0.174D+03	-.589D+01
215	0.128D+03	-.256D+02
212	-.158D+03	-.245D+01

ELEMENT ID 205

GRID	FX	FY
209	-.111D+03	-.[illegible]D+02
210	-.116D+03	0.[illegible]D+02
217	0.202D+03	0.453D+02
216	0.199D+03	-.445D+02
213	-.175D+03	-.334D+01

ELEMENT ID 206

GRID	FX	FY
210	-.855D+02	-.283D+03
211	-.540D+02	-.851D+01
218	0.186D+03	0.253D+02
217	0.168D+03	0.523D+01
214	-.157D+03	0.619D+01

Figure 44. SAMCJ Output for the Problem Defined in Figure 43 (Continued).

ELEMENT ID 207

GRID	FX	FY
215	.120D+03	0.256D+02
216	-.160D+03	-.888E+01
220	0.126D+03	-.175D-12
219	0.163D+03	-.739D-12

ELEMENT ID 208

GRID	FX	FY
216	-.212D+03	0.585D+02
217	-.211D+03	-.591D+02
221	0.212D+03	-.169D+02
220	0.211D+03	0.175D+02

ELEMENT ID 209

GRID	FX	FY
217	-.159D+03	0.841D+01
218	-.129D+03	-.253D+02
222	0.162D+03	-.178D-13
221	0.126D+03	0.169D+02

JOINT LOAD LEVELS CORRESPONDING TO NET SECTION (NS), SHEAR-OUT (SO) AND BEARING (BR) FAILURES AT EVERY LOADED AND UNLOADED HOLE ELEMENT ARE PREDICTED AS FOLLOWS:

ELEMENT	NS	SO	BR
101	0.585D+06	0.883D+06	0.459D+06
102	0.546D+06	0.700D+06	0.501D+06
103	0.417D+06	0.626D+06	0.492D+06
104	0.292D+07	0.151D+07	0.738D+06
105	0.311D+07	0.149D+07	0.837D+06
106	0.283D+07	0.171D+07	0.737D+06
108	0.148D+08	0.899D+08	0.927D+08
201	0.340D+06	0.123D+06	0.120D+06
202	0.295D+06	0.976D+05	0.105D+06
203	0.513D+06	0.120D+06	0.132D+06
204	0.764D+05	0.644D+05	0.777D+05
205	0.529D+05	0.411D+05	0.577D+05
206	0.808D+05	0.555D+05	0.813D+05
208	0.373D+05	0.174D+06	0.342D+07

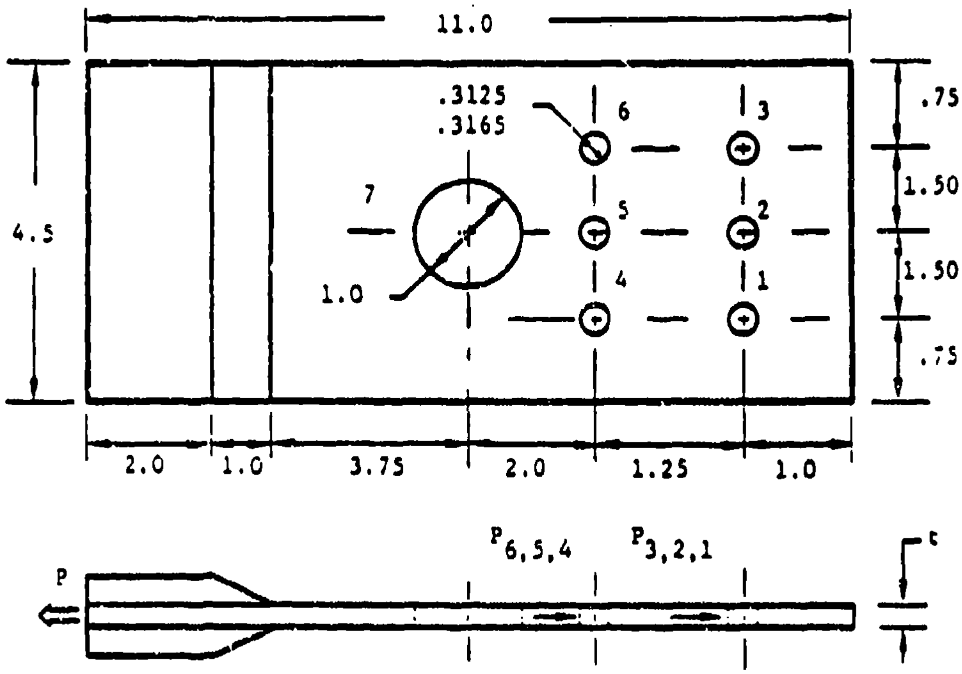
FAILURE IS PREDICTED TO OCCUR IN ELEMENT NUMBER 208 AT AN APPLIED JOINT LOAD VALUE OF 0.3729000D+05 LBS

THE PREDICTED FAILURE MODE IS NET SECTION

Figure 44. SAMCJ Output for the Problem Defined in Figure 43 (Concluded).

load, the failure location and the failure mode. For the considered sample problem, a net section failure is predicted across the one inch diameter cut-out (element 208) in the graphite/epoxy plate, at a joint load level of 37.3 kips. Figure 45 compares SAMCJ predictions with test results from Reference 8.

Test Case 243, Static Tension, Single Lap
40-Ply, 50/40/10 Laminate, t=0.247 in., t_{AL}=0.50 in.
D=5/16 in., H_D=1 in., S_L/D=S_T/D=4, W/D=14.4, E/D=3.2



	SAMCJ PREDICTION	TEST RESULTS (Ref. 2)
P_1/P	0.165	0.162
P_2/P	0.151	0.150
P_3/P	0.165	0.168
P_4/P	0.167	0.177
P_5/P	0.175	0.161
P_6/P	0.158	0.165
$P_{failure}$ (kips)	37.3 (52.7)*	42.0
FAILURE LOCATION	7 (5)	7 and 4, 5, 6
FAILURE MODE(S)	NET SECTION (NET SECTION)	NET SECTION

* Next possible failure mode and location at a higher load level

Figure 45. Comparison of SAMCJ Predictions for the Sample Problem with Test Results from Reference 8.

SECTION 4

DESIGN VERIFICATION OF A BOLTED STRUCTURAL ELEMENT

The design of a highly-loaded structural bolted joint is verified in this section using the analytical tool (SAMCJ computer code) proposed for the recommended design methodology (Section 1.3).

4.1 Description of the Bolted Structural Element

In Reference 5, a bolted joint concept was studied as an alternative to a highly loaded composite-to-titanium, step lap bonded joint. The vertical tail structure of the F/A-18A was used as the baseline for this study. A preliminary design of the bolted structural element, representative of the critical F/A-18A vertical tail root section, was performed based on approximate analyses and available test results. The test element was designed to transfer a design ultimate load of 70.2 kips (obtained from the F/A-18A empennage stress analysis report), and to survive two lifetimes of a representative design spectrum fatigue loading.

The design of the bolted structural element studied in Reference 5 differs from the existing F/A-18A vertical tail root joint significantly. It eliminates the graphite/epoxy skin-to-titanium bonded joint, and directly attaches the skins to the fuselage frame. In doing so, it also uses a light root rib, in contrast to the highly-loaded attachment root rib used in Reference 4. The AS4/3501-6 graphite/epoxy skins of the element have a 41-ply layup away from the attachment location. The skins increase in thickness to a 60-ply layup near the tab region that bolts the vertical tail skin to the fuselage frame. The graphite/epoxy tabs are machined, prior to assembly, to introduce a taper at the joint location. In Reference 5, the fuselage attachment fitting was made out of steel, and the skins were bolted to it using 3/8 inch diameter, countersunk high strength steel bolts. Figure 46 shows a

Figure 46. Photograph of an Assembled Test Element.

photograph of an assembled test element. The element spar and the root rib were fabricated using an aluminum alloy.

4.2 <u>Test Results</u>

Elements fabricated based on this preliminary design were subjected to static and fatigue loads in Reference 5. They survived two lifetimes of a spectrum fatigue load that was significantly more severe than the actual F/A-18A vertical tail design spectrum load, and their static strengths were approximately 30% larger than the design ultimate load. During the static test, failure occured in the graphite/epoxy skin tab in a combined mode (see Figure 47). The observed failure modes were significantly influenced by the tilting or "digging in" of the countersunk fasteners - a phenomenon that cannot be accounted for by the fastener analysis in the SAMCJ computer code.

4.3 <u>Design Verification of the Element Using SAMCJ</u>

The critical vertical tail skin-to-fuselage joint region is analyzed below using the SAMCJ code that is recommended as an analytical design tool. Though the analysis was performed retrospectively, the assumed material and failure parameters are identical to those used in Reference 7.

Figure 48 presents the dimensions of the analyzed graphite/epoxy skin tabs and the fuselage attachment frame. The tapered skin has a $[0_{28}/\pm45_{12}/90_7]_C$ layup at the top of the tab region. Across the top row of fasteners, it has an average of 58 plies, and across the bottom row of fasteners, it has an average of 52 plies. For analytical purposes, the tapered tab region is modeled as two uniform regions of different thicknesses. The top region is modeled to contain a $[0_{28}/\pm45_{12}/90_6]_C$ layup, and the bottom region is assumed to be a $[0_{26}/\pm45_{10}/90_6]_C$ laminate. The average thickness of a ply in the skin was measured to be 0.0049 inch. The fuselage attachment frame is, likewise, divided into a

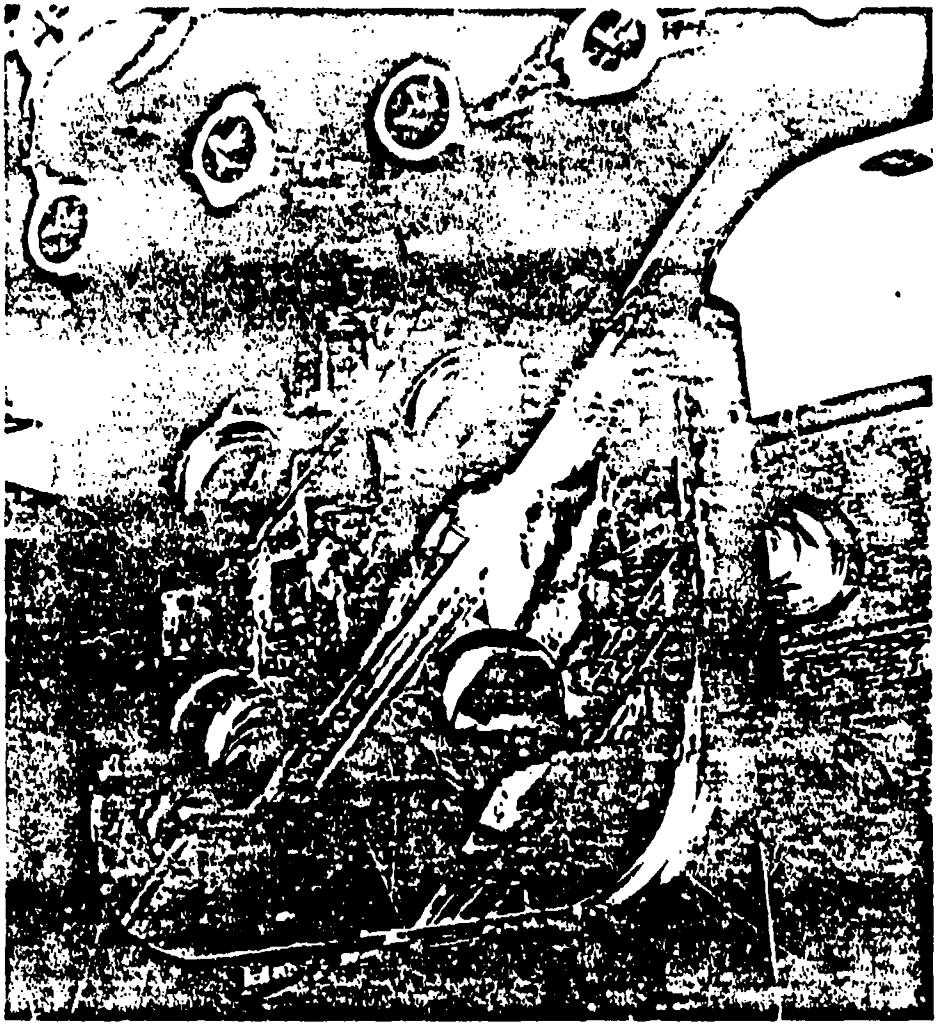


Figure 47. Photograph of the Tab Region of the Failed Element.

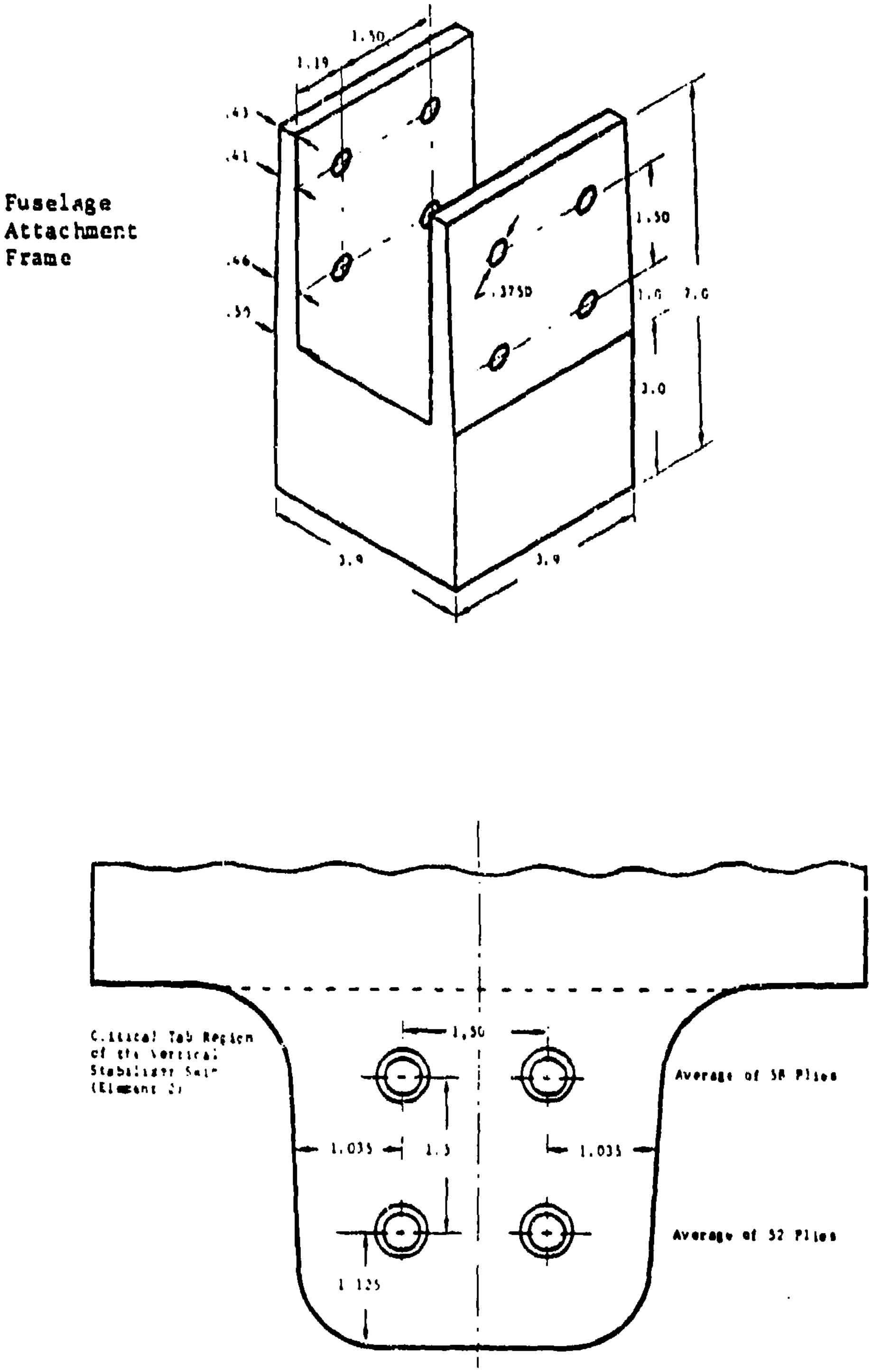


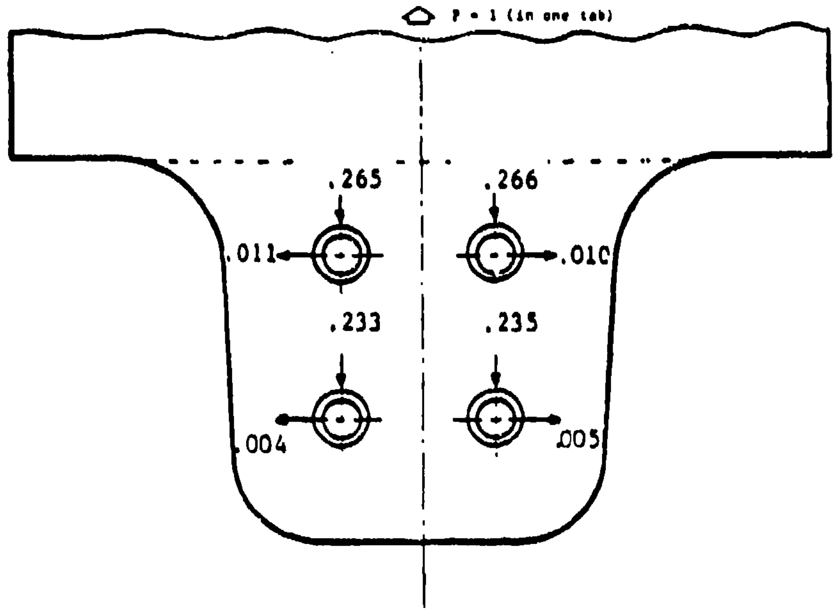
Figure 48. Dimensions of the Critical Skin Tab and the Fuselage Attachment Frame.

0.41 inch thick region and a 0.46 inch thick region (see Figure 48).

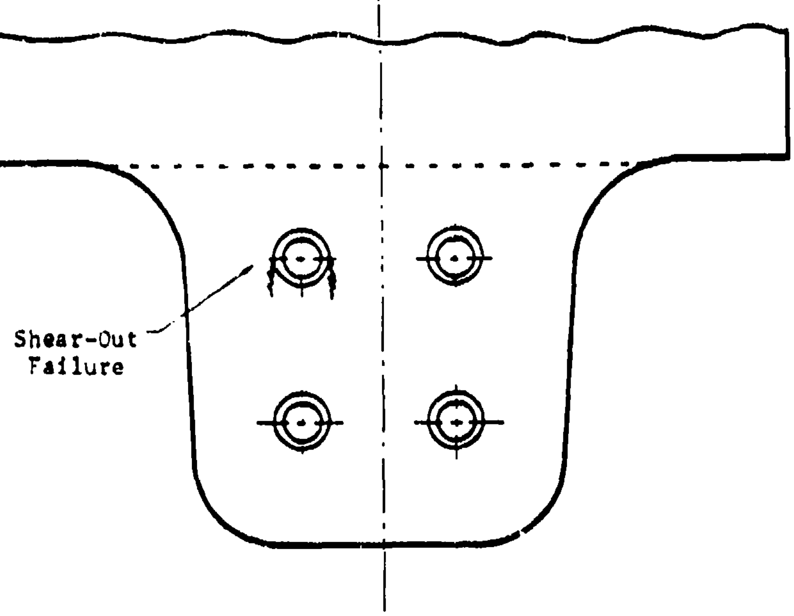
The modeled joint segment is half of the symmetric skin tab-to-fuselage attachment. The total joint failure load is, therefore, twice the predicted load. A single shear load transfer between the AS4/3501-6 graphite/epoxy skin tab and the steel attachment frame is analyzed. The graphite/epoxy tab and the steel plate are divided into four elements each. The average width of the slightly tapered tab is used in the analytical model (3.57 in.). The fiber-directional tensile and compressive failure strains for AS4/3501-6 graphite/epoxy are assumed to be 0.012 and 0.0175, respectively (References 7, 13). The characteristic distances for net section, bearing and shear-out failure modes are assumed to be 0.10, 0.25 and 0.25 inch, respectively (Reference 7). The basic AS4/3501-6 lamina properties are assumed to be 18.5 Msi, 1.9 Msi and 0.85 Msi for E_{11}, E_{22} and G_{12}, respectively, and 0.3 for the major Poisson's ratio.

The skins are attached to the fuselage frame by 3/8 inch diameter, countersunk fasteners (100 degree tension head). The fastener analysis in SAMCJ cannot accurately account for the effects of the countersunk head geometry. However, it approximates the actual effects by assuming free rotation at the fastener head location, and requires the user to input an equivalent protruding head fastener diameter. In the discussed element analysis, the average fastener diameter is assumed to be 0.458 inch, to account for the 100 degree tension head geometry.

Analytically predicted load distribution among the fasteners in each tab is presented in Figure 49. The symmetry in the fastener arrangement results in low values for the transverse components of fastener loads (perpendicular to the load direction). Also, the loads in the top row of fasteners are approximately 14% larger than those in the bottom row of fasteners. This leads to a prediction of failure initiation from the top row of fasteners (see Figure 49). The predicted failure site (critical location) is in



Fastener Load Distribution



Failure Location and Failure Mode

Figure 49. Load Distribution Among Fasteners, Failure Location and Failure Mode in the Graphite/Epoxy Tabs.

agreement with experimental observation.

Figure 50 presents the analytically predicted element load levels to precipitate net section, bearing and shear-out modes of failure at the various fastener locations. The lowest among these provides the element failure load, the failure location and the failure mode. SAMCJ predicts element failure to be caused by a shear-out mode of failure at the top left fastener location in Figure 50. The failure mode observed in Reference 5, however, was severe damage around the fastener hole, introduced by the tilting of the countersunk fasteners (see Figure 47). This included some amount of shear-out and local bearing, and severe delaminations around the fastener hole boundaries. Since SAMCJ cannot account for the severe local three-dimensional stress state introduced by the countersunk fasteners, the predicted failure mode (shear-out) does not correlate well with the observed combined failure mode (partial shear-out, local bearing, and severe delaminations).

Despite the approximate failure mode prediction, however, SAMCJ correctly predicts the failure location, and the failure load predicted by SAMCJ (98.0 kips) is only 7% larger than the measured value (91.8 kips). The approximation of the countersunk fasteners by equivalent protruding head fasteners (larger diameter, unconstrained at the head location), therefore, predicts the element failure load with adequate accuracy. The SAMCJ analysis and the test results in Reference 5 indepenently verify the 30% margin of safety in the static strength of the test element, due to the approximate analyses used in its preliminary design.

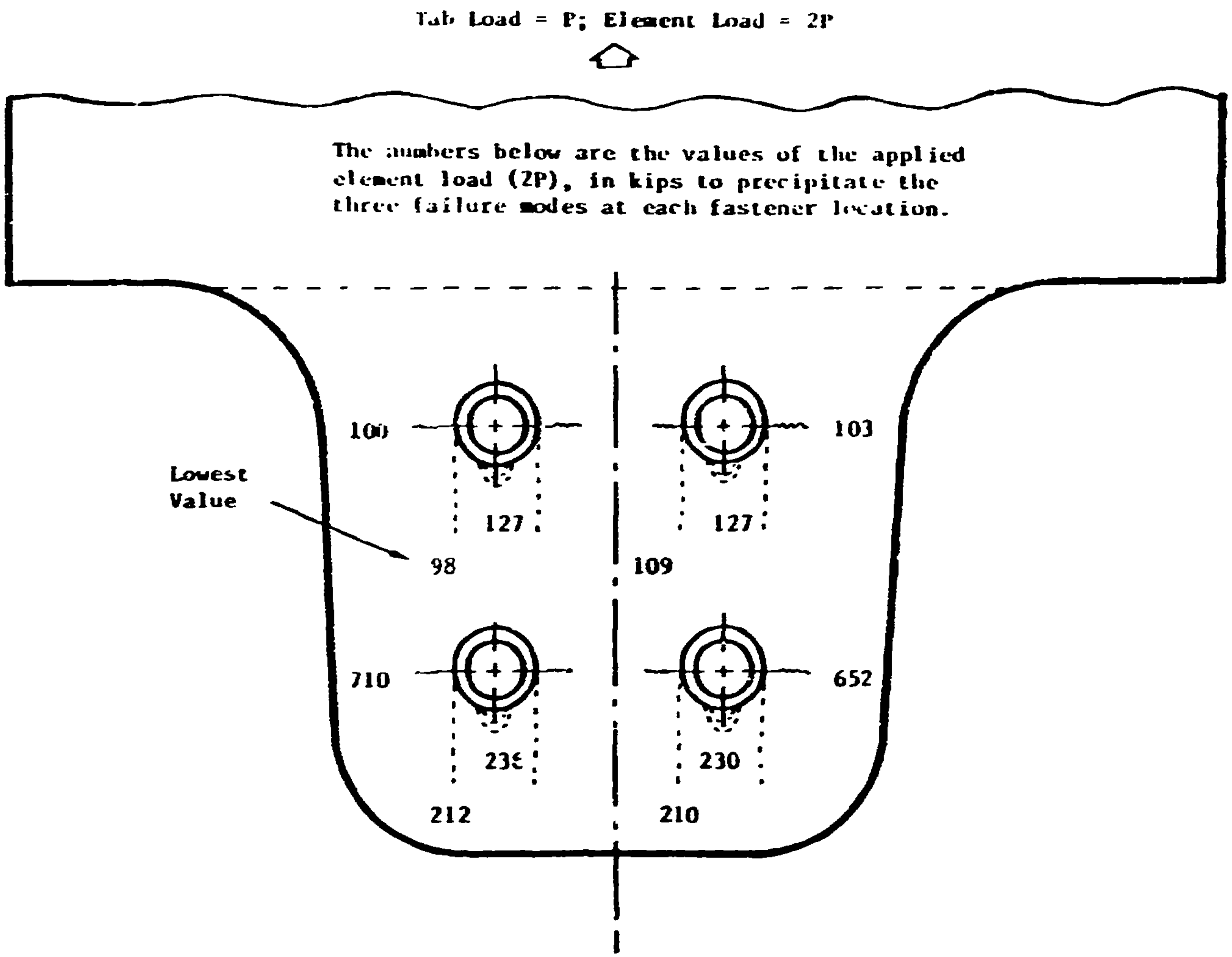


Figure 50. Analytically Predicted Element Load Levels to Precipitate Net Section, Bearing and Shear-Out Modes of Tab Failure at Each Fastener Location.

SECTION 5

CONCLUSIONS

A design guide was developed to enable the user in designing efficient bolted joints in composite structures. The guide highlights general design guidelines for the various parameters that are to be considered in selecting a bolted joint concept. A purely analytical design methodology is presented. It is devoid of complementary test requirements when a previously characterized material is used to fabricate the bolted structure. The design guide also illustrates the use of two computer codes (SASCJ and SAMCJ) that were developed in this Northrop/AFWAL program and are required for design purposes. A listing of these computer codes is appended to this report.

REFERENCES

1. Buchanan, D. L. and Garbo, S. P., "Design of Highly Loaded Composite Joints and Attachment for Wing Structures," NADC-81194-60, August 1981.

2. Watts, D. J., et al., "Development of Composites Technology for Joints and Cutouts in Fuselage Structure of Large Transport Aircraft," Semi-Annual Technical Report Number 1, ACEE-34-PR-3419, October 1984.

3. Kong, S. J., "Conseptual Design of a High Load Transfer Mechanical Attachemnt for Tail Structures," NADC-81216-60, August 1981.

4. Averill, S. W. and Zamani, H. R., "Development of High Load Joints and Attachments for Composite Structure," NADC-81220-60, August 1983.

5. Ramkumar, R. L., Saether, E. S. and Tossavainen, E., "Design, Fabrication, Testing and Analysis of Bolted Structural Elements," AFWAL-TR-86-XXXX, January 1986.

6. Ramkumar, R. L., et al., "Strength Analysis of Composite and Metallic Plates Bolted Together by a Single Fastener," AFWAL-TR-85-3064, August 1985.

7. Ramkumar, R. L., Saether, E. S., and Appa, K., "Strength Analysis of Laminated and Metallic Plates Bolted Together by Many Fasteners," AFWAL-TR-86-XXXX, January 1986.

8. Ramkumar, R. L. and Tossavainen, E., "Bolted Joints in Composite Structures: Design, Analysis and Verification; Task II Test Results," AFWAL-TR-85-3065, August 1985.

REFERENCES (Concluded)

9. ACEE Composite Structures Technology, Papers by Douglas Aircraft Company, edited by M. Klotzsche, NASA-CR-172359, August 1984.

10. Hart-Smith, L. J., "Mechanically-Fastened Joints for Advanced Composite-Phenomenological Considerations and Simple Analysis," Proceedings of the Fourth Conference on Fibrous Composites in Structural Design, November 1978.

11. Cole, B. and Batch, E. J., "Special Fastener Development for Composite Structure," AFWAL-TR-82-3049, July 1982.

12. Wall, F. T., "Fasteners for Use with Composite Materials," NASC Report Number 78-01, Aerospace Industries Association of America, Inc, January 1978.

13. Garbo, S. P. and Ogonowski, J. M., "Effect of Variances and Manufacturing Tolerances on the Design Strength and Life of Mechanically Fastened Joints," Volume 1, AFWAL-TR-81-3041, April 1981.

14. Ramkumar, R. L. and Tossavainen, E., "Bolted Joints in Composite Structures: Design, Analysis and Verification; Task I Test Results," AFWAL-TR-84-3047, August 1984.

15. Eisenmann, J. R. and Leonhardt, J. L., "Improving Composite Bolted Joint Efficiency by Laminate Tailoring," Joining of Composite Materials, ASTM STP 749, edited by K. T. Kedward, American Society for Testing and Materials, 1981, pp. 117-130.

APPENDIX A

SASCJ Program Listing

```
C*****************************************************************
C*****************************************************************
C**                                                           ***
C**                    PROGRAM SASCJ                          ***
C**                                                           ***
C**    STRENGTH ANALYSIS OF SINGLE-FASTENER COMPOSITE JOINTS  ***
C**                                                           ***
C*****************************************************************
C*****************************************************************
C
C      SASCJ PREDICTS LOAD-DEFORMATION CURVES AND FAILURE LOADS OF
C      MECHANICALLY FASTENED, COMPOSITE LAMINATE, SINGLE LAP OR
C      SYMMETRICAL DOUBLE LAP SHEAR JOINTS. THE BASIS OF THE FAILURE
C      ANALYSIS IS A NONLINEAR FINITE DIFFERENCE SOLUTION OF A BEAM
C      (FASTENER) ON AN ELASTIC FOUNDATION (COMPOSITE LAMINATE).
C      SELECTED FAILURE CRITERIA ARE USED TO PREDICT INDIVIDUAL PLY
C      FAILURES AND MODES (INCLUDING INTERLAMINAR SHEAR). THE LOAD
C      IS AUTOMATINCALLY INCREMENTED TO FINAL FAILURE TO ACCOUNT FOR
C      THE NONLINEAR JOINT BEHAVIOR.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NPLY(2),PLYK(100),WASHD(2),ES1(2),ES2(2),ES3(2)
      DIMENSION MTL(3,15),R(2),F(100),U(100)
      DIMENSION NPNM(100,2),NUMPLY(2),GAMDL(2),CM(2)
      DIMENSION DELNS(5,2),PNS(5,2),PBR(5,2),PSO(5,2),PALT(3,2)
      DIMENSION DELBR(5,2),DELSO(5,2),ANGK(5,2)
      DIMENSION ANG(5,2),IPLY(100,2),DONT(2),DOBR(2),DOSO(2)
      DIMENSION FNT(5,2),FBR(5,2),FSO(5,2)
      DIMENSION BARK(100),BARU(100),HFMC(5,2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2),H(2)
      DIMENSION UN(100),MDAMP(100),MDAMI(100),PN(100),GAMN(100)
      DIMENSION XOUT(400),YOUT(400),PSTC(5,5,2),RCA(2),RCB(2),NRCOUT(2)
      DIMENSION XC(2,5),YC(2,5),SALOW(2)
      DIMENSION AONT(2,2),AOBR(2,2),AOSO(2,2)
      COMMON/AOV/AONT,AOBR,AOSO
      COMMON/COUNT/NPNM
      COMMON/MOD/E1,E2,G12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/RC/RCA,RCB,NRCOUT
      COMMON/PSC1/DONT,DOBR,DOSO
      COMMON/PSC2/PSTC
      COMMON/FAL1/PNS,PBR,PSO,PALT
      COMMON/FAL2/FNT,FBR,FSO
      COMMON/FAL3/DELNS,DELBR,DELSO
      COMMON/FAL4/UN,GAMN,MDAMP,MDAMI,PN
      COMMON/PBB/PLYK,BARK,BARU
      COMMON/ELP/AX,BX,NOUT
      COMMON/HFF/HFMC
      COMMON/SER/NT,NB
      DATA CMC/'C'/
C
C
C     READ IN REQUIRED INPUT DATA
C
C
```

```
      WRITE(6,876)
  876 FORMAT(///,10X,' PROGRAM SASCJ',//,
     *' PROGRAM SASCJ PREDICTS FAILURE LOADS OF ',/,
     *' MECHANICALLY FASTENED, COMPOSITE LAMINATE,',/,
     *' SINGLE OR DOUBLE LAP SHEAR JOINTS.         ',//,
     *' PROGRAM ASSUMES THAT INPUT PARAMETERS ARE ',/,
     *' IN ENGLISH UNITS - LENGTHS ARE INPUT ',/,
     *' IN INCHES AND MODULI AND STRENGTHS ARE      ',/,
     *' EXPRESSED IN PSI                           ',///)
      WRITE(6,401)
  401 FORMAT(' ENTER BYPASS RATIO ALPHA:      ',//,
     *'       ALPHA=0 FOR FULL BEARING     ',/,
     *'       ALPHA=1 FOR OPEN HOLE       ',/,
     *'       0<ALPHA<1 FOR GENERAL BYPASS   ')
      READ(5,*) BPR
      WRITE(6,911)
  911 FORMAT(' ENTER:                        ',/,
     *'          1 FOR STATIC TENSION     ',/,
     *'          2 FOR STATIC COMPRESSION',/)
      READ(5,*) LTNCM
      NLIM=1
      IF(BPR.EQ.1.0) GO TO 380
      NLIM=2
      WRITE(6,400)
  400 FORMAT(' ENTER:                        ',/,
     *'          1 FOR SLS (SINGLE LAP SHEAR)',/,
     *'          2 FOR DLS (DOUBLE LAP SHEAR)',/)
      READ(5,*) NSDLS
  106 FORMAT(A1)
  380 CONTINUE
      DO 300 K=1,NLIM
      IF(K.EQ.1) WRITE(6,611)
  611 FORMAT(' IS THE TOP PLATE A COMPOSITE OR A METAL ?',/,
     *' ENTER C OR M IN THE FIRST FIELD')
      IF(K.EQ.2) WRITE(6,789)
  789 FORMAT(' IS THE BOTTOM PLATE A COMPOSITE OR A METAL ?',/,
     *' ENTER C OR M IN THE FIRST FIELD')
      READ(5,106) CM(K)
      WRITE(6,203)
  203 FORMAT(' INPUT MATERIAL DESCRIPTION OF THIS PLATE ',/,
     *'        EX: AS4/3501-6')
      READ(5,204) (MTL(K,I),I=1,15)
  204 FORMAT(15A4)
  300 CONTINUE
      IF(CM(1).NE.CMC.OR.CM(2).NE.CMC) WRITE(6,721)
  721 FORMAT(/,' NOTE: FOR COMPUTATIONAL PURPOSES A ',/,
     *' METALLIC PLATE IS MODELED AS A 30 PLY ',/,
     *' LAMINATE OF 0 DEGREE PLIES WITH ISOTROPIC',/,
     *' MATERIAL PROPERTIES',/)
      IF(BPR.NE.1.0) WRITE(6,494)
      IF(BPR.EQ.1.0) WRITE(6,495)
  494 FORMAT(' NOTE: NUMERICAL DESIGNATIONS FOR THE ',/,
     *' PLATES ARE:                              ',/,
     *'                     TOP PLATE    = NO 1          ',/,
     *'                     BOTTOM PLATE = NO 2          ',/)
  495 FORMAT(' NOTE: A SINGLE PLATE WITH AN OPEN ',/,
     *' HOLE IS DESIGNATED AS PLATE NUMBER 1',/)
      DO 301 K=1,NLIM
      IF(CM(K).EQ.CMC) GO TO 15
      NPLY(K)=30
```

```
      GO TO 301
   15 CONTINUE
      IF(NSDLS.EQ.2.AND.K.EQ.1) WRITE(6,932)
  932 FORMAT(/,' NOTE: FOR THE DOUBLE LAP SHEAR CASE HAVING ',/,
     *' A COMPOSITE PLATE NUMBER 2, ENTER ONLY HALF',/,
     *' OF THE LAYUP - IE HALF THE NUMBER OF ACTUAL',/,
     *' PLIES            ')
      WRITE(6,205) K
  205 FORMAT(' INPUT NUMBER OF PLIES IN PLATE NO',I5,/,
     *'  (N > OR = 2)')
      READ(5,*) NPLY(K)
      NLIM2=2*NPLY(K)+1
  301 CONTINUE
      DO 302 K=1,NLIM
      IF(CM(K).EQ.CMC) GO TO 25
      IF(NSDLS.EQ.2.AND.K.EQ.2) WRITE(6,933)
  933 FORMAT(/,' FOR THE DOUBLE LAP SHEAR CASE HAVING',/,
     *' A METALLIC PLATE NUMBER TWO, ENTER HALF THE ',/,
     *' ACTUAL PLATE THICKNESS   ')
      WRITE(6,35) K
   35 FORMAT(' INPUT THICKNESS OF PLATE NO',I5)
      READ(5,*) A1
      H(K)=A1/NPLY(K)
      GO TO 302
   25 CONTINUE
      WRITE(6,260) K
  260 FORMAT(' INPUT PLY THICKNESS IN PLATE NO',I5)
      READ(5,*) H(K)
  302 CONTINUE
      DO 303 K=1,NLIM
      IF(CM(K).EQ.CMC) GO TO 45
      NUMPLY(K)=1
      GO TO 303
   45 CONTINUE
      WRITE(6,207) K
  207 FORMAT(' INPUT NUMBER OF DISTINCT PLY ORIENTATIONS',/,
     *' IN PLATE NO',I5)
      READ(5,*) NUMPLY(K)
  303 CONTINUE
      DO 209 K=1,NLIM
      IF(CM(K).EQ.CMC) GO TO 55
      ANG(1,K)=0.
      GO TO 209
   55 CONTINUE
      WRITE(6,487) K
  487 FORMAT(/,' FOR PLATE NUMBER',I5,' :',/)
      N=NUMPLY(K)
      DO 209 L=1,N
      WRITE(6,206) L
  206 FORMAT(' INPUT ORIENTATION OF PLY TYPE NO',I5)
      READ(5,*) ANG(L,K)
  209 CONTINUE
      DO 305 K=1,NLIM
      IF(CM(K).EQ.CMC) GO TO 65
      NN=NPLY(K)
      DO 75 IJ=1,NN
   75 IPLY(IJ,K)=1
      GO TO 305
   65 CONTINUE
      WRITE(6,210) K
```

```
 210 FORMAT(' INPUT TYPE OF PLY IN PLATE NO',I5,' FROM TOP',/,
     *' TO BOTTOM                                         ',//,
     *5X,'PLY TYPE',10X,'ORIENTATION')
     N=NUMPLY(K)
     DO 212 L=1,N
     WRITE(6,213) L,ANG(L,K)
 213 FORMAT(5X,I5,10X,F7.2,' DEGREES')
 212 CONTINUE
     WRITE(6,711)
711  FORMAT(/)
     N=NPLY(K)
     DO 215 I=1,N
     WRITE(6,214) I
 214 FORMAT(' INPUT TYPE OF PLY FOR PLY NO',I5)
     READ(5,*) IPLY(I,K)
 215 CONTINUE
 305 CONTINUE
     DO 306 K=1,NLIM
     WRITE(6,216) K
 216 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF PLATE NO',I5)
     IF(CM(K).EQ.CMC) GO TO 85
     WRITE(6,95)
  95 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO')
     READ(5,*) E1(K),V12(K)
     E2(K)=E1(K)
     V21(K)=V12(K)*E2(K)/E1(K)
     G12(K)=E1(K)/(2.*(1.+V12(K)))
     GO TO 306
  85 CONTINUE
     WRITE(6,217)
 217 FORMAT(' INPUT YOUNGS MODULI, E1 AND E2')
     READ(5,*) E1(K),E2(K)
     WRITE(6,218)
 218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO')
     READ(5,*) G12(K),V12(K)
     V21(K)=V12(K)*E2(K)/E1(K)
 306 CONTINUE
     IF(BPR.NE.1.0) GO TO 930
     WRITE(6,844)
 844 FORMAT(' INPUT HOLE DIAMETER')
     READ(5,*) FASD
     GO TO 360
 930 CONTINUE
     WRITE(6,250)
 250 FORMAT(' INPUT MATERIAL DESCRIPTION FOR FASTENER')
     READ(5,251) (MTL(3,I),I=1,15)
 251 FORMAT(15A4)
     WRITE(6,252)
 252 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO FOR',/,
     *' THE FASTENER')
     READ(5,*) FASE,FASV
     WRITE(6,253)
 253 FORMAT(' INPUT THE DIAMETER OF THE FASTENER')
     READ(5,*) FASD
     WRITE(6,888)
 888 FORMAT(/,' FASTENER TYPE     ',/,
     *'        ENTER:      1 FOR PROTRUDING HEAD ',/,
     *'                    2 FOR COUNTERSUNK HEAD ')
     READ(5,*) NFTYP
     R(1)=1.0D10
```

```
      R(2)=1.0D0
      IF(NFTYP.EQ.1) GO TO 360
      WRITE(6,889)
  889 FORMAT(/,' ENTER PLATE WHICH CONTAINS THE COUNTERSUNK',/,
     *' HEAD (OPPOSITE PLATE ASSUMES THE NUT HEAD)     ',//,
     *'       ENTER:     1 FOR TOP PLATE          ',/,
     *'                  2 FOR BOTTOM PLATE       ')
      READ(5,*) N
      R(N)=0.0D0
  360 CONTINUE
C
C     READ IN GEOMETRY AND BOUNDARY DATA
C
      AX=FASD/2.0D0
      BX=AX
      WRITE(6,856)
  856 FORMAT(' PLATE GEOMETRIES ARE SPECIFIED BY ',/,
     *' INPUTTING THE COORDINATES OF THE CORNER',/,
     *' VERTICIES. NOTE: THE ORIGIN IS AT THE FASTENER',/,
     *' CENTER ; INPUT COORDINATES ACCORDINGLY',/,
     *'                                             ',/,
     *'          V3                      V2         ',/,
     *'                   HOLE                      ',/,
     *'                 CENTROID                    ',/,
     *'          V4                      V1         ',//,
     *' APPLIED LOAD CONVENTION:                    ',/,
     *' FOR PLATE NO 1 (TOP) NORMAL LOADS ARE APPLIED     ',/,
     *' BETWEEN V3 AND V4                           ',/,
     *' FOR PLATE NO 2 (BOTTOM) NORMAL LOADS ARE APPLIED ',/,
     *' BETWEEN V1 AND V2                           ',//)
      DO 480 K=1,NLIM
      WRITE(6,734) K
 734  FORMAT(' FOR PLATE NUMBER ',I5,' :',/)
      DO 110 I=1,4
      WRITE(6,290) I
  290 FORMAT(' ENTER X,Y COORDINATES OF V',I4)
      READ(5,*) XC(K,I),YC(K,I)
  110 CONTINUE
      IF(K.EQ.2) GO TO 841
      A1=XC(1,1)
      B1=YC(1,1)
      A2=XC(1,2)
      B2=YC(1,2)
      XC(1,1)=XC(1,4)
      YC(1,1)=YC(1,4)
      XC(1,2)=XC(1,3)
      YC(1,2)=YC(1,3)
      XC(1,4)=A1
      YC(1,4)=B1
      XC(1,3)=A2
      YC(1,3)=B2
  841 CONTINUE
      WTH=YC(K,2)-YC(K,1)
  480 CONTINUE
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 567
      WRITE(6,741)
  741 FORMAT(' SELECT FAILURE CRITERION:     ',//,
     *'    ENTER 1 FOR POINT STRESS CRITERION    ',/,
     *'    ENTER 2 FOR AVERAGE STRESS CRITERION ')
      READ(5,*) NPT
```

```
      IF(NPT.EQ.1) NOPT4=2
      IF(NPT.EQ.2) NOPT4=4
      GO TO 601
  567 CONTINUE
      WRITE(6,220)
  220 FORMAT(' SELECT FAILURE CRITERION    :',//,
     *'    ENTER 1 FOR HOFFMAN/TSAI-HILL CRITERION  ',/,
     *'    ENTER 2 FOR POINT STRESS CRITERION       ',/,
     *'    ENTER 3 FOR MAXIMUM STRAIN CRITERION     ',/,
     *'    ENTER 4 FOR AVERAGE STRESS CRITERION')
      READ(5,*) NOPT4
  601 CONTINUE
      IF(NOPT4.EQ.2.OR.NOPT4.EQ.4) GO TO 221
      DO 412 K=1,NLIM
      WRITE(6,222) K
  222 FORMAT(' FOR PLATE NUMBER ',I5,' ENTER RADIUS OF      ',/,
     *' CHARACTERISTIC CIRCLE AT WHICH STRESSES ARE',/,
     *' TO BE COMPUTED TO PREDICT FAILURE')
      READ(5,*) RCA(K)
      RCB(K)=RCA(K)
      NRCOUT(K)=50
      IF(NOPT4.EQ.3) GO TO 591
      WRITE(6,834)
  834 FORMAT(' ENTER THE FAILURE INDEXES FOR THE ',/,
     *' HOFFMAN/TSAI-HILL CRITERIA                 ',//,
     *' NOTE: FOR USING TSAI-HILL SET EQUAL THE COMPRESSION ',/,
     *'         AND TENSION ULTIMATES IN SIGMA X AND Y ',//
     *'      ENTER:  SIGMA X ULTIMATE-COMPRESSION      ',/,
     *'              SIGMA X ULTIMATE-TENSION          ',/,
     *'              SIGMA Y ULTIMATE-COMPRESSION      ',/,
     *'              SIGMA Y ULTIMATE-TENSION            ',/,
     *'              SIGMA XY ULTIMATE                 ')
      READ(5,*)(HFMC(I,K),I=1,5)
      GO TO 412
  591 CONTINUE
      WRITE(6,393) K
  393 FORMAT(' ENTER MAXIMUM STRAIN ALLOWABLE FOR',/,
     *' PLATE NUMBER ',I7,' (UNITS: IN/IN)')
      READ(5,*) SALOW(K)
  412 CONTINUE
      IF(NOPT4.EQ.3) GO TO 391
      IF(NOPT4.EQ.1) GO TO 391
      GO TO 262
  221 CONTINUE
      IF(NOPT4.EQ.2) WRITE(6,555)
      IF(NOPT4.EQ.4) WRITE(6,556)
  555 FORMAT(/,' POINT STRESS CRITERION ',/)
  556 FORMAT(/,' AVERAGE STRESS CRITERION  ',//,
     *' AO IS THE CHARACTERISTIC DISTANCE OVER WHICH',/,
     *' STRESSES ARE AVERAGED AND COMPARED WITH UNNOTCHED',/,
     *' STRENGTHS TO PREDICT FAILURE')
      DO 226 K=1,NLIM
      IF(BPR.NE.0.0.AND.BPR.NE.1.0) GO TO 531
      WRITE(6,225) K
  225 FORMAT(' INPUT AO FOR EACH OF THE THREE PLY FAILURE',/,
     *' MODES OF PLATE NO',I5,/,
     *'              AONT = NET SECTION                      ',/,
     *'              AOBR = BEARING                          ',/,
     *'              AOSO = SHEAR OUT                        ',/)
      N=NUMPLY(K)
```

```
      WRITE(6,227)
  227 FORMAT(' INPUT AONT,AOBR,AND AOSO')
      READ(5,*) DONT(K),DOBR(K),DOSO(K)
      GO TO 226
  531 CONTINUE
      WRITE(6,532) K
  532 FORMAT(' ENTER AO VALUES CORRESPONDING TO THE THREE',/,
     *' PLY FAILURE MODES IN PLATE NO ',I2,/,
     *'          AONT = NET SECTION      ',/,
     *'          AOBR = BEARING          ',/,
     *'          AOSO = SHEAR OUT        ')
      WRITE(6,533)
  533 FORMAT(' INPUT AONT, AOBR, AOSO ')
      READ(5,*) AONT(1,K),AOBR(1,K),AOSO(1,K)
      AONT(2,K)=AONT(1,K)
      AOBR(2,K)=AOBR(1,K)
      AOSO(2,K)=AOSO(1,K)
      IF(K.EQ.1) WRITE(6,554)
  554 FORMAT(/,' TO AVOID LENGTHY RUN TIMES DUE TO ',/,
     *' STRESS FIELD RECOMPUTATION SPECIFY THE ',/,
     *' NUMBER OF ULTIMATE PLY FAILURES AFTER ',/,
     *' WHICH JOINT FAILURE WILL BE PREDICTED ',/,
     *' ENTER: NO OF ULTIMATE FAILURES    ')
      IF(K.EQ.1) READ(5,*) NULTF
  226 CONTINUE
  291 CONTINUE
      NOPT1=1
      IF(BPR.EQ.1.0) NOPT1=2
      DO 229 K=1,NLIM
      IF(BPR.NE.1.0.OR.(BPR.EQ.1.0.AND.NOPT1.EQ.2)) GO TO 670
      GO TO 671
  670 CONTINUE
      IF(CM(K).EQ.CMC) GO TO 672
      WRITE(6,228) K
  228 FORMAT(/' FOR PLATE NUMBER ',I5,' ENTER THE THREE STRENGTHS ',/,
     *' REQUIRED TO PREDICT THE THREE FAILURE MODES ',/,
     *' FNST=UNNOTCHED STRENGTH IN TENSION ',/,
     *' FNSC=UNNOTCHED STRENGTH IN COMPRESSION',/,
     *' FSO=UNNOTCHED STRENGTH IN SHEAR-OUT',//,
     *' INPUT FNST,FNSC,FSO             ')
      READ(5,*) AF1,AF2,AF4
      GO TO 673
  672 WRITE(6,674) K
  674 FORMAT(' FOR PLATE NO ',I5,' ENTER FIBER ULTIMATE',/,
     *' STRAIN VALUES                                ',/,
     *'                   EPSILON ULT IN COMPRESSION ',/,
     *'                   EPSILON ULT IN TENSION     ',/,
     *'                   GAMMA ULT IN SHEAR         ',/,
     *' (UNITS: IN/IN)')
      READ(5,*) ES1(K),ES2(K),ESS(K)
      CALL STRTH(H,ES1,ES2,ESS,AF1,AF2,AF4,K)
  673 CONTINUE
      AF3=AF2
      NP=NUMPLY(K)
      DO 666 IL=1,NP
      PSTC(1,IL,K)=AF1
      PSTC(2,IL,K)=AF2
      PSTC(3,IL,K)=AF3
      PSTC(4,IL,K)=AF4
  666 CONTINUE
```

```
  229 CONTINUE
  671 CONTINUE
      IF(NOPT4.NE.4) GO TO 261
C
C     NUMBER OF DIVISIONS FOR STRESS AVERAGING
C     IS SET EQUAL TO 50
C
      NAVD=50
  261 CONTINUE
      IF(BPR.EQ.1.0) GO TO 262
      DO 319 K=1,NLIM
      N=NUMPLY(K)
      WRITE(6,320) K
  320 FORMAT(' SASCJ ASSUMES A BILINEAR PLY BEHAVIOR. THE ',/,
     *' INITIAL MODULUS, K1, IS COMPUTED BY THE CODE. ',/,
     *' THE REDUCED MODULUS, K2, FOR INITIAL FAILURE',/,
     *' IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED',/,
     *' BY THE FORMULA K2=ALPHA*K1.                ',//,
     *' FOR PLATE NUMBER ',I5,' INPUT ALPHA VALUES FOR ',/,
     *' NET SECTION, SHEAROUT AND BEARING FAILURE     ')
      READ(5,*) AF1,AF2,AF3
      DO 321 I=1,N
      DELNS(I,K)=AF1
      DELBR(I,K)=AF2
      DELSO(I,K)=AF3
  321 CONTINUE
      WRITE(6,389)
  389 FORMAT(' INPUT SCALE FACTORS FOR P ULTIMATE ',/,
     *' CALCULATION SUCH THAT P(ULT)=BETA*P(INITIAL)',/,
     *' INPUT BETA1 FOR NET SECTION ULTIMATE      ',/,
     *'       BETA2 FOR BEARING ULTIMATE       ',/,
     *'       BETA3 FOR SHEAROUT ULTIMATE ')
      READ(5,*) PALT(3,K),PALT(2,K),PALT(1,K)
  319 CONTINUE
  391 CONTINUE
      IF(BPR.NE.0.0) GO TO 262
      DO 312 K=1,NLIM
      GAMDL(K)=10.0
      IF(CM(K).NE.CMC) GO TO 312
      WRITE(6,231) K
  231 FORMAT(/,' INPUT THE APPROXIMATE INTERLAMINAR SHEAR STRAIN',/,
     *' ULTIMATE FOR DELAMINATION PREDICTION IN PLATE NO ',I5,/,
     *' (UNITS: IN/IN)                              ')
      READ(5,*) GAMDL(K)
  312 CONTINUE
  262 CONTINUE
C
C     CASE HEADING
C
      WRITE(6,143)
  143 FORMAT(///,10X,'PROGRAM SASCJ',//)
      IF(NSDLS.EQ.1.AND.BPR.NE.1.0) WRITE(6,633)
      IF(NSDLS.EQ.2.AND.BPR.NE.1.0) WRITE(6,634)
  633 FORMAT(2X,'A SINGLE LAP SHEAR JOINT WILL BE ANALYZED',/)
  634 FORMAT(2X,'A DOUBLE LAP SHEAR JOINT WILL BE ANALYZED',/)
      IF(BPR.EQ.0.0) WRITE(6,881)
      IF(BPR.EQ.1.0) WRITE(6,882)
      IF(BPR.NE.0.0.AND.BPR.NE.1.0) WRITE(6,883) BPR
  881 FORMAT(2X,'WITH A LOADED HOLE',/)
  882 FORMAT(2X,'WITH AN OPEN HOLE',/)
```

```
883 FORMAT(2X,'WITH A PARTIALLY LOADED HOLE',/,
   *2X,'BYPASS RATIO = ',D9.3,/)
    IF(LTNCM.EQ.1) WRITE(6,823)
    IF(LTNCM.EQ.2) WRITE(6,824)
823 FORMAT(2X,'LOADED IN STATIC TENSION',/)
824 FORMAT(2X,'LOADED IN STATIC COMPRESSION',/)
    DO 605 I=1,NLIM
    WRITE(6,600) I
600 FORMAT(10X,'PLATE NO ',I5,' :',/)
    WRITE(6,689) (MTL(I,J),J=1,15)
689 FORMAT(2X,15A4,/)
    HT=NPLY(I)*H(I)
    WRITE(6,602) HT
602 FORMAT(2X,'T = ',D9.3,' INCHES',/)
    WRITE(6,603) E1(I),E2(I),G12(I),V12(I),V21(I)
603 FORMAT(2X,'MATERIAL PROPERTIES',/,/,
   *10X,'E1  =',D9.3,' PSI',/,
   *10X,'E2  =',D9.3,' PSI',/,
   *10X,'G12 =',D9.3,' PSI',/,
   *10X,'NU12=',D9.3,/,
   *10X,'NU21=',D9.3,/)
605 CONTINUE
    IF(BPR.EQ.1.0) GO TO 708
    WRITE(6,606)
606 FORMAT(10X,'FASTENER:',/)
    WRITE(6,607) (MTL(3,J),J=1,15)
607 FORMAT(2X,15A4,/)
    WRITE(6,608) FASD
608 FORMAT(2X,' DIAMETER = ',D9.3,' INCHES',/)
    WRITE(6,609) FASE,FASV
609 FORMAT(2X,' MATERIAL PROPERTIES',/,/
   *10X,'E =',D9.3,' PSI',/,
   *10X,'MU=',D9.3,/)
708 CONTINUE
    WRITE(6,923)
923 FORMAT(//,10X,'FAILURE ANALYSIS',/)
    IF(NOPT4.EQ.2.OR.NOPT4.EQ.4) GO TO 621
    IF(NOPT4.EQ.3) GO TO 821
    WRITE(6,622)
622 FORMAT(2X,'THE HOFFMAN/TSAI-HILL CRITERION WILL BE USED',/)
    DO 623 J=1,NLIM
    WRITE(6,624) J,RCA(J)
624 FORMAT(2X,'PLATE NUMBER ',I5,//,
   *2X,'CHARACTERISTIC RADIUS =',D9.3,' INCHES')
    WRITE(6,790)
790 FORMAT(/,16X,' ULTIMATE STRESSES:                ',/,/,
   *10X,'TENSION',18X,'COMPRESSION')
    WRITE(6,625) (HFMC(I,J), I=1,5),HFMC(5,J)
625 FORMAT(/,2X,'SIGMA X = ',D9.3,' PSI',5X,'SIGMA X = ',
   *D9.3,' PSI',/,
   *2X,'SIGMA Y = ',D9.3,' PSI',5X,'SIGMA Y = ',
   *D9.3,' PSI',/,
   *2X,'SIGMA S = ',D9.3,' PSI',5X,'SIGMA S = ',
   *D9.3,' PSI'/)
623 CONTINUE
    GO TO 627
621 CONTINUE
    IF(NOPT4.EQ.2) WRITE(6,628)
628 FORMAT(2X,'A POINT STRESS CRITERION WILL BE USED',/)
    IF(NOPT4.EQ.4) WRITE(6,558)
```

```
558 FORMAT(2X,'AN AVERAGE STRESS CRITERION WILL BE USED',/)
    DO 631 I=1,NLIM
    WRITE(6,632) I
632 FORMAT(2X,'PLATE NUMBER',I5,/)
    NP=NUMPLY(I)
    WRITE(6,713)
713 FORMAT(/,2X,'LAMINATE STRENGTH',/)
776 WRITE(6,677) (PSTC(LL,1,I),LL=1,4)
677 FORMAT(2X,'NET SECTION ULTIMATE (TEN) =',D9.3,' PSI',/,
   *'  NET SECTION ULTIMATE (COMP)=',D9.3,' PSI',/,
   *2X,'BEARING ULTIMATE           =',D9.3,' PSI',/,
   *2X,'SHEAROUT ULTIMATE          =',D9.3,' PSI',/)
    IF(BPR.NE.0.0.AND.BPR.NE.1.0) GO TO 561
    WRITE(6,644)
644 FORMAT(2X,'CHARACTERISTIC DISTANCES ',/)
    WRITE(6,645) DONT(I),DOBR(I),DOSO(I)
645 FORMAT(2X,' DONT = ',D9.3,' INCHES',/,
   *2X,' DOBR = ',D9.3,' INCHES',/,
   *2X,' DOSO = ',D9.3,' INCHES',/)
    GO TO 631
561 WRITE(6,562)
562 FORMAT(2X,'CHARACTERISTIC DISTANCES',/)
    WRITE(6,564) AONT(1,I),AOBR(1,I),AOSO(1,I)
564 FORMAT(2X,' AONT = ',D9.3,' INCHES',/,
   *'   AOBR = ',D9.3,' INCHES',/,
   *'   AOSO = ',D9.3,' INCHES',/)
631 CONTINUE
    GO TO 627
821 CONTINUE
    WRITE(6,822)
822 FORMAT(2X,'MAXIMUM STRAIN CRITERION WILL BE USED',/)
    DO 887 II=1,NLIM
    WRITE(6,858) II,RCA(II)
858 FORMAT(2X,'PLATE NUMBER ',I5,//,
   *2X,'CHARACTERISTIC RADIUS =',D9.3,' INCHES')
    WRITE(6,825) SALOW(II)
825 FORMAT(/,8X,'STRAIN ULTIMATE = ',D9.3,' IN/IN',/)
887 CONTINUE
627 CONTINUE
C
C     CALCULATE THE PLY FOUNDATION MODULI AND
C     FAILURE LOADS
C
      NBP=1
      IF(BPR.NE.0.0.AND.BPR.NE.1.0) NBP=2
      IF(NBP.EQ.1) NLIM2=1
      DO 71 LOM=1,NLIM2
      DO 22 IL=1,NBP
      DO 20 K=1,NLIM
C
C     INITIALIZE PARAMETERS FOR COLLOCATION
C
      NT=7
      NOUT=50
      NCOL=10
      NB=NOUT+4*NCOL
C
C     CONTINUE CASE HEADING
C
      IF(LOM.GT.1) GO TO 23
```

```
      IF(IL.EQ.2) GO TO 23
      WRITE(6,871) K
  871 FORMAT(/,5X,' GEOMETRY OF PLATE NO ',I5,'-',/)
      WRITE(6,872)
  872 FORMAT(' COORDINATES OF CORNER VERTEXES ',/)
      IF(K.EQ.1) WRITE(6,873) XC(K,2),YC(K,2),XC(K,3),YC(K,3)
      IF(K.EQ.1) WRITE(6,874) XC(K,1),YC(K,1),XC(K,4),YC(K,4)
      IF(K.EQ.2) WRITE(6,873) XC(K,3),YC(K,3),XC(K,2),YC(K,2)
      IF(K.EQ.2) WRITE(6,874) XC(K,4),YC(K,4),XC(K,1),YC(K,1)
  873 FORMAT(2X,F7.3,',',F7.3,10X,F7.3,',',F7.3,/)
  874 FORMAT(2X,F7.3,',',F7.3,10X,F7.3,',',F7.3,/)
      AXD=AX*2.
      WRITE(6,875) AXD
  875 FORMAT(' FASTENER HOLE DIAMETER = ',D9.3,' INCHES',/)
      ED=DABS(XC(K,3)/AXD)
      WD=DABS((YC(K,3)-YC(K,4))/AXD)
      WRITE(6,755) ED
  755 FORMAT(' E/D RATIO = ',D9.3,/)
      WRITE(6,879) WD
  879 FORMAT(' W/D RATIO = ',D9.3,/)
   23 CONTINUE
C
C     PROCESS INPUT DATA ON PLATE GEOMETRIES
C
      WTH=YC(K,2)-YC(K,1)
      LM1=LOM
      CALL POLY(JK,K,XC,YC,H,AST,NCOL,LTNCM,BPR,IL)
      CALL CIRC(H,AST,JK,K,LTNCM,BPR,IL)
      IF(NOPT4.EQ.1.OR.NOPT4.EQ.3) CALL RCOUT(K)
      IF(NOPT4.EQ.2) CALL PSTRSS(K,LTNCM,BPR,IL)
      IF(NOPT4.EQ.4) CALL AVSTRS(K,LTNCM,NAVD,BPR,IL)
C
C     PERFORM FINITE GEOMETRY ANALYSIS FOR STRESS/DISPLACEMENT
C     STATE, COMPUTE FOUNDATION MODULI AND FAILURE VALUES
C
      CALL FIGEOM(H,K,NOPT4,ITT)
      IF(BPR.NE.0.0.AND.BPR.NE.1.0.AND.IL.EQ.1) GO TO 21
      IF(BPR.NE.1.0.AND.LOM.LE.1) CALL FBOLT(ANGK,H,K,NOPT1,LM1)
   21 CALL FCRIT(SALOW,H,WTH,AST,K,NOPT1,NOPT4,BPR,NAVD,IL)
   20 CONTINUE
   22 CONTINUE
      IF(BPR.EQ.1.0) GO TO 410
C
C     PREPARE INPUT FOR SEQUENTIAL PLY FAILURE
C     PREDICTION
C
      IF(LOM.GT.1) GO TO 61
      N=NPLY(1)
      DO 30 I=1,N
      M=IPLY(I,1)
   30 PLYK(I)=ANGK(M,1)
      N=NPLY(2)
      DO 60 I=1,N
      N1=I+NPLY(1)
      N2=IPLY(I,2)
   60 PLYK(N1)=ANGK(N2,2)
   61 CONTINUE
C
C
C     CALCULATION OF FASTENER STIFFNESSES.
```

```
C
      FASO=FASE/(2.*(1.+FASV))
      FASLAM=5.*(1.0+FASV)/(7.+5.*FASV)
      FASR=FASD/2.
      FASA=ACOS(-1.)*FASR**2
      FASI=ACOS(-1.)*FASR**4/4.
      FASSS=FASLAM*FASO*FASA
      FASBS=FASE*FASI
C
C     INITIALIZATION
C
      IF(LOM.GT.1) GO TO 72
      ITT=0
      NTFL=0
      JNT=1
      P=0.
      DELP=1000.
      DO 5012 I=1,100
      NPNM(I,1)=I
      NPNM(I,2)=I+NPLY(1)
      UN(I)=0.
      DAMN(I)=0.
      MDAMP(I)=0.
      MDAMI(I)=0.
      PN(I)=0.
      BARK(I)=0.
 5012 BARU(I)=0.
   72 CONTINUE
C
C     INCREMENTAL LOADS TO PLY FAILURE, PLY FAILURE
C     MODES, AND FRACTIONAL STIFFNESS LOSSES ARE
C     CALCULATED FOR EACH PLY FROM TOP TO BOTTOM
C     UNTIL FINAL JOINT FAILURE
C
   90 CONTINUE
      ITT=ITT+1
      CALL CENTD(R,H,FASSS,FASBS,P,DELP,ITT)
      CALL SOLVE(U,H,P,DELP,NSDLS,ITT)
      CALL FAIL(DAMDL,U,H,P,DELP,BPR,AST,WTH,PFAIL,ANGLE,NODE,
     *IROUT,NOPT4,NULTF,JNT,ITT,NTFL)
      CALL PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT,
     *NP,NSDLS,ITT)
      IF(JNT.EQ.0) GO TO 410
      IF(NLIM2.EQ.1) GO TO 90
      IF(NTFL.EQ.0.AND.NLIM2.GT.1) GO TO 90
   71 CONTINUE
  410 STOP
      END
C
C
C
      SUBROUTINE STRTH(H,ES1,ES2,ESS,AF1,AF2,AF4,K)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AINV(3,3),AVN(3),H(2),NV(3)
      DIMENSION NPLY(2),NUMPL(2),ANG(5,2),IPLY(100,2)
      DIMENSION WK(25),PSMX(3),ES1(2),ES2(2),ESS(2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2)
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/MCD/E1,E2,G12,V12,V21
      COMMON/AMI/A
```

```
C
C     COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM
C     FIBER STRAINS FOR EACH FAILURE MODE
C
      CALL AMATRX(H,K)
      N=3
      IDOT=4
      IA=3
      CALL LINV2F(A,N,IA,AINV,IDOT,WK,IER)
      DO 100 KK=1,3
      DO 10 II=1,3
      NV(II)=0
   10 AVN(II)=0.0D0
      IF(KK.EQ.1) NV(1)=1
      IF(KK.EQ.2) NV(1)=-1
      IF(KK.EQ.3) NV(3)=1
      DO 15 II=1,3
      DO 15 JJ=1,3
      AVN(II)=AVN(II)+AINV(II,JJ)*NV(JJ)
   15 CONTINUE
      NP=NUMPLY(K)
      SMX=0.0D0
      RAD=DARCOS(-1.0D0)/180.0D0
      DO 25 II=1,NP
      TH=ANG(II,K)*RAD
      E11=DCOS(TH)**2*AVN(1)+AVN(2)*DSIN(TH)**2+
     *DCOS(TH)*DSIN(TH)*AVN(3)
      IF(KK.NE.1) GO TO 65
      EPRT=E11/ES2(K)
      GO TO 50
   65 IF(KK.NE.2) GO TO 75
      EPRT=E11/ES1(K)
      GO TO 50
   75 EPRT=E11/ES2(K)
   50 CONTINUE
      IF(DABS(SMX).LT.DABS(EPRT)) SMX=EPRT
   25 CONTINUE
      IF(DABS(SMX).GT.1.0D-10) GO TO 555
      PSMX(KK)=ESS(K)*G12(K)
      GO TO 100
  555 CONTINUE
      PSMX(KK)=DABS(1.0D0/SMX)
  100 CONTINUE
      AF1=PSMX(1)
      AF2=PSMX(2)
      AF4=PSMX(3)
      RETURN
      END
C
C
C
      SUBROUTINE POLY(J,K,XC,YC,H,AST,NCOL,LTNCM,BPR,IL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XC(2,5),YC(2,5),A1(400),A2(400),XB(400)
      DIMENSION YB(400),T(400),A1A(4),A2A(4)
      COMMON/CMT1/XB,YB,A1,A2,T
C
C     ARRAY COLLOCATION POINTS AROUND EXTERIOR BOUNDARY
C     AND APPLY STRESS BOUNDARY CONDITIONS
C
```

```
      DO 120 I=1,4
      A1A(I)=0.
      A2A(I)=0.
  120 CONTINUE
      W=DABS(YC(K,2)-YC(K,1))
      IF(LTNCM.EQ.1) A1A(1)=1000.0
      IF(LTNCM.EQ.2) A1A(1)=-1000.0
      IF(BPR.NE.0.0) A1A(3)=A1A(1)
      IF(IL.EQ.2) A1A(3)=0.0
      AST=DABS(A1A(1))
      J=0
      XC(K,5)=XC(K,1)
      YC(K,5)=YC(K,1)
      PI=DARCOS(-1.0D0)
      DAT=PI/NCOL
      DO 10 I=1,4
      X=XC(K,I)-XC(K,I+1)
      Y=YC(K,I+1)-YC(K,I)
      IF(X.EQ.0.) X=1.D-6
      IF(Y.EQ.0.) Y=1.D-6
      TH=DATAN2(X,Y)
      TH=TH*180./DARCOS(-0 1D1)
      DX=(XC(K,I+1)-XC(K,I))/(NCOL+1)
      DY=(YC(K,I+1)-YC(K,I))/(NCOL+1)
      DO 20 II=1,NCOL
      J=J+1
      IF(I.EQ.1.OR.I.EQ 3) GO TO 23
      YB(J)=YC(K,I)
      XB(J)=XC(K,I)+DX*(II- .5)
      IF(II.EQ.1) XB(J)=XC(K,I)+(DX/2.)
      GO TO 24
   23 CONTINUE
      IF(XC(K,3).NE.0.0) GO TO 26
      IF(I.NE.3) GO TO 26
      ADT=DAT*II
      XB(J)=YC(K,3)*DCOS((PI/2.)+ADT)
      YB(J)=YC(K,3)*DSIN((PI/2.)+ADT)
      TH=((PI/2.)+ADT)*180./PI
      GO TO 24
   26 CONTINUE
      YB(J)=YC(K,I)+DY*(II- .5)
      IF(II.EQ.1) YB(J)=YC(K,I)+(DY/2.)
      XB(J)=XC(K,I)
 24   T(J)=TH
      A1(J)=A1A(I)
      A2(J)=A2A(I)
20    CONTINUE
10    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CIRC(W,AST,JK,K,LTNCM,BPR,IL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(400),Y(400),THTA(400),A1(400),A2(400)
      DIMENSION XB(400),YB(400)
      COMMON/FB1/BSTR,XSTR
      COMMON/CMT1/XB,YB,A1,A2,THTA
      COMMON/CMT2/X,Y,NPST,NAST
```

```
      COMMON/ELP/A,B,N
C
C     ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY
C     AND APPLY BEARING STRESS IN A COSINUSOIDAL
C     DISTRIBUTION
C
      CON=-1.0
      XSTR=AST
      BSTR=(2.*W*XSTR)/(DARCOS(CON)*B)
      IF(BPR.NE.0.0.AND.BPR.NE.1.0.AND.IL.EQ.1) BSTR=0.0
      IF(BPR.EQ.1.0) BSTR=0.
      NM4=N-4
      NQ=NM4/4
      DO 20 I=1,N
      JK=JK+1
      TH=((I-1)*2+1)*DARCOS(CON)/N
      X(I)=A*DCOS(TH)
      Y(I)=B*DSIN(TH)
      CS=-X(I)*B*B/(Y(I)*A*A)
      IF(Y(I).GT.0)THTA(JK)=DATAN(CS)-DARCOS(CON)/2.
      IF(Y(I).LT.0)THTA(JK)=DATAN(CS)+DARCOS(CON)/2.
      THTA(JK)=THTA(JK)*180./DARCOS(CON)
      IF(LTNCM.EQ.2) GO TO 25
      IF(I.GT.(NQ+1).AND.I.LT.(N-NQ)) GO TO 204
      GO TO 30
   25 IF(I.LE.(NQ+2).OR.I.GE.(N-NQ-1)) GO TO 204
   30 CONTINUE
      A1(JK)=0.
      A2(JK)=0.
      XB(JK)=X(I)
      YB(JK)=Y(I)
      GO TO 20
204   IF(Y(I).GT.0.) TETA=ARCOS(-1.)-DATAN(Y(I)/X(I))
      IF(Y(I).LT.0.) TETA=ARCOS(-1.)+DATAN(Y(I)/X(I))
      A1(JK)=-1.0*BSTR*DABS(DCOS(TETA))
      A2(JK)=0.
      XB(JK)=X(I)
      YB(JK)=Y(I)
   20 CONTINUE
      RETURN
      END
C
      SUBROUTINE RCOUT(K)
C
C     SPECIFY COORDINATES AROUND CHARACTERISTIC CIRCLE
C     AN WHICH STRESSES ARE NEEDED FOR THE HOFFMAN/
C     TSAI-HILL FAILURE CRITERIA
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(400),Y(400),RCA(2),RCB(2),NRC(2)
      COMMON/CMT2/X,Y,NPST,NAST
      COMMON/RC/RCA,RCB,NRC
      COMMON/ELP/AX,BX,NOUT
      RAD=DARCOS(-0.1D1)/180.
      N1=NRC(K)
      DO 40 I=1,N1
      TINCR=360./NRC(K)
      THETA=(I-1)*TINCR*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
```

```
      R=DSQRT(1./((C**2/RCA(K)**2)+(S**2/RCB(K)**2)))
      X(I+NOUT)=R*DCOS(THETA)
      Y(I+NOUT)=R*DSIN(THETA)
   40 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE PSTRSS(K,NCS,BPR,IL)
C
C     SPECIFYDISCRETE COORDINATES OF POINTS AT WHICH
C     STRESSES ARE REQUIRED FOR THE POINT STRESS
C     CRITERION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(400),Y(400),DONT(2),DOBR(2)
      DIMENSION DOSO(2),NPLY(2),NUMPLY(2),ANO(5,2)
      DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSO(2,2)
      COMMON/ELP/AX,BX,NOUT
      COMMON/CMT2/X,Y,NPST,NAST
      COMMON/PSC1/DONT,DOBR,DOSO
      COMMON/LYP/NPLY,NUMPLY,ANO,IPLY
      COMMON/AOV/AONT,AOBR,AOSO
      ANT=DONT(K)
      ABR=DOBR(K)
      ASO=DOSO(K)
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 25
      ANT=AONT(IL,K)
      ABR=AOBR(IL,K)
      ASO=AOSO(IL,K)
   25 CONTINUE
      L=NOUT+1
      SO=1.0
      IF(NCS.EQ.1) SO=-1.0
      X(L)=0.
      Y(L)=ANT+BX
      X(L+1)=SO*(AX+ABR)
      Y(L+1)=0
      X(L+2)=SO*(AX+ASO)
      Y(L+2)=BX
      NPST=3
C      DO 555 I=1,3
      L=NOUT+1+(I-1)
      RETURN
      END
C
C
C
      SUBROUTINE AVSTRS(K,NCS,NAVD,BPR,IL)
C
C     SPECIFY COORDINATES OF POINTS ALONG WHICH
C     STRESSES WILL BE AVERAGED FOR THE AVERAGE
C     STRESS CRITERION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(400),Y(400),DONT(2),DOBR(2)
      DIMENSION DOSO(2),NPLY(2),NUMPLY(2),ANO(5,2)
      DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSO(2,2)
      COMMON/AOV/AONT,AOBR,AOSO
```

```
      COMMON/EL. /AX,BX,NOUT
      COMMON/CMT2/X,Y,NPST,NAST
      COMMON/PSC1/DONT,DOBR,DOSO
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      ANT=DONT(K)
      ABR=DOBR(K)
      ASO=DOSO(K)
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 23
      ANT=AONT(IL,K)
      ABR=AOBR(IL,K)
      ASO=AOSO(IL,K)
   25 CONTINUE
      L=NOUT
      SG=1.0
      IF(NCS.EQ.1) SG=-1.0
      ANDO=ANT/FLOAT(NAVD)
      DO 20 I=1,NAVD
      L=L+1
      X(L)=0.
   20 Y(L)=BX+ANDO/2.+(I-1)*ANDO
      ANSO=ASO/FLOAT(NAVD)
      DO 30 I=1,NAVD
      L=L+1
      X(L)=SG*(BX+ANSO/2.+(I-1)*ANSO)
   30 Y(L)=BX
      ANBR=ABR/FLOAT(NAVD)
      DO 40 I=1,NAVD
      L=L+1
      X(L)=SG*(AX+ANBR/2.+(I-1)*ANBR)
   40 Y(L)=0.
      NAST=3*NAVD
      N1=NOUT+1
      N2=N1+NAST
      NN=NOUT+3*NAVD
      RETURN
      END
C
C
C
      SUBROUTINE FIGEOM(H,KJ,NOPT4,ITT)
C
C
C
C     FIGEOM PERFORMS A FINITE GEOMETRY ANALYSIS
C     USING THE BOUNDARY COLLOCATION TECHNIQUE
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),WK(25),AI(3,3),AZ(3),WKK(121),BC(400)
      DIMENSION CH(4),H(2)
      COMPLEX*16 GRHS(122)
      COMPLEX*16 CM(196,124),CMC(196,121),CMCTCM(121,121),RHS(121)
      COMMON/ROOTS/R1,R2
      COMMON/TERMS/P1,Q1,P2,Q2
      COMMON/ELP/AX,BX,NOUT
      COMMON/SER/NT,NB
      COMMON/AMT/A
      COMPLEX*16 Z(4),Z1,Z2,Q1,Q2,P1,P2,R1,R2,WA(1488S)
C
```

```
C     AMATRX CALCULATES THE LAMINATE 'A' MATRIX
C
      CALL AMATRX(H,KJ)
      N=3
      IDGT=4
      IA=3
C
C     LINV2F INVERTS THE 'A' MATRIX
C
      CALL LINV2F(A,N,IA,AI,IDGT,WK,IER)
      NDEG=4
      AZ(1)=AI(1,1)/AI(2,2)
      AZ(2)=-2.*AI(1,3)/AI(2,2)
      AZ(3)=(2.*AI(1,2)+AI(3,3))/AI(2,2)
      AZ(4)=-2.*AI(2,3)/AI(2,2)
      AZ(5)=1.0D0
C
C     ZRPOLY FINDS THE ROOTS OF THE CHARACTERISTIC EQUATION
C
      CALL ZRPOLY(AZ,NDEG,Z,IER)
C
C     Z(2) AND Z(4) ARE THE COMPLEX CONJUGATES OF Z(1)
C     AND Z(3) RESPECTIVELY
C
      R1=Z(1)
      R2=Z(3)
C
C     THE TWO ROOTS MUST B. CHECKED FOR A UNITARY COMPONENT
C     IN EITHER THE REAL OR IMAGINARY PART; SUCH AN
C     OCCURANCE SIGNIFIE. A QUASI-ISOTROPIC LAYUP AND
C     THE VALUE MUST BE PERTURBED SLIGHTLY IN ORDER TO
C     AVOID A SINGULAR MATRIX
C
      CH(1)=R1
      CH(2)=(0.0,-1.0)*R1
      CH(3)=R2
      CH(4)=(0.0,-1.0)*R2
      DO 30 IJK=1,4
      IF(DABS(CH(IJK)).LT.1.0D-10) CH(IJK)=1.0D-10
      AR=DABS(CH(IJK))
      IF(AR.LE.1.0) GO TO 31
      GO TO 32
   31 IF((1.0-AR).LT.0.02) CH(IJK)=0.98
      GO TO 30
   32 IF((AR-1.0).LT.0.02) CH(IJK)=1.02
   30 CONTINUE
      R1=DCMPLX(CH(1),CH(2))
      R2=DCMPLX(CH(3),CH(4))
C
C     CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS
C
      P1=AI(1,1)*R1**2+AI(1,2)-AI(1,3)*R1
      P2=AI(1,1)*R2**2+AI(1,2)-AI(1,3)*R2
      Q1=AI(2,2)/R1+AI(1,2)*R1-AI(2,3)
      Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)
C
C     INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS
C
      NT4=4*NT
      NT8=8*NT
```

```
      NT8P4=8*NT+4
      NT8P2=8*NT+2
      NT8P1=8*NT+1
      NB2=2*NB
      NWK=NT8P1*(NT8P1+2)
      CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,
     1NT8P1,NB2,NWK,WA,WKK,AI,NOPT4,KJ,ITT)
      RETURN
      END
C
C
C
      SUBROUTINE AMATRX(H,K)
C
C     ASSEMBLE THE A MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),ANG(5,2),H(2),NPLY(2),NUMPLY(2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2)
      DIMENSION IPLY(100,2)
      COMMON/MOD/E1,E2,G12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/AMT/A
      THKNES=NPLY(K)*H(K)
      DENO=1.-E2(K)*V12(K)**2/E1(K)
      Q11=E1(K)/DENO
      Q22=E2(K)/DENO
      Q12=V12(K)*Q22
      Q21=Q12
      Q33=G12(K)
      DO 10 I=1,3
      DO 10 J=1,3
   10 A(I,J)=0.
      NN=NPLY(K)
      DO 20 I=1,NN
      T=H(K)
      LP=IPLY(I,K)
      THTAI=ANG(LP,K)*DARCOS(-1.D0)/180.D0
      C=DCOS(THTAI)
      S=DSIN(THTAI)
      A(1,1)=(Q11*C**4+2.*(Q12+2.*Q33)*C*C*S*S+Q22*S**4)*T+A(1,1)
      A(2,2)=(Q11*S**4+2.*(Q12+2.*Q33)*C*C*S*S+Q22*C**4)*T+A(2,2)
      A(1,2)=((Q11+Q22-4.*Q33)*C*C*S*S+Q12*(C**4+S**4))*T+A(1,2)
      A(2,1)=A(1,2)
      A(3,3)=((Q11+Q22-2.*Q12-2.*Q33)*C*C*S*S+Q33*(C**4+S**4))*T+A(3,3)
      A(1,3)=((Q11-Q12-2.*Q33)*C**3*S+(Q12-Q22+2.*Q33)*S**3*C)*T+A(1,3)
      A(2,3)=((Q11-Q12-2.*Q33)*S**3*C+(Q12-Q22+2.*Q33)*C**3*S)*T+A(2,3)
      A(3,2)=A(2,3)
      A(3,1)=A(1,3)
   20 CONTINUE
      DO 53 I=1,3
      DO 53 J=1,3
      A(I,J)=A(I,J)/THKNES
   53 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,
```

```
     1NT8P1,NB2,NWK,WA,WKK,AMAT,NOPT4,KJ,ITT)
C
C
C      CMAT OUTPUTS STRESSES, STRAINS, AND DISPLACEMENTS
C      AT SPECIFIED COORDINATES
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION RCA(2),RCB(2),NRCOUT(2)
       DIMENSION ASX(400),ASXY(400)
       COMMON/XXY1/ASX,ASXY
       COMMON/ROOTS/R1,R2
       COMMON/TERMS/P1,Q1,P2,Q2
       COMMON/CMT1/X,Y,AIN1,AIN2,THTA
       COMMON/CMT2/XOUT,YOUT,NPST,NAST
       COMMON/FB2/FUR,FTHT,FSMR
       COMMON/QMT/RTHT,REPX,REPY,REPXY
       COMMON/RC/RCA,RCB,NRCOUT
       COMMON/ELP/AX,BX,NOUT
       COMMON/SER/NT,NB
       DIMENSION  THTA(400),X(400),Y(400),AMAT(3,3)
       DIMENSION  AIN1(400),AIN2(400),BC(NB2)
       DIMENSION  XOUT(400),YOUT(400),WKK(NT8P1)
       DIMENSION FUR(400),FTHT(400),FSMR(400)
       DIMENSION RTHT(400),REPX(400),REPY(400),REPXY(400)
       COMPLEX*16 CMCTCM(NT8P1,NT8P1),RHS(NT8P1),PHI1D,PHI2D,XETA1,XETA2
       COMPLEX*16 CM(NB2,NT8P4),CMC(NB2,NT8P1),Z1,Z2,Z11,Z22,R1,R2
       COMPLEX*16 T11,T12,T21,T22,P11,P12,P21,P22
       COMPLEX*16 P1,P2,Q1,Q2,DCMPLX,CO,CSUM,GRHS(NT8P2)
       COMPLEX*16 PHI1DP,PHI2DP,PHI1DN,PHI2DN
       COMPLEX*16 PHI1P,PHI2P,PHI1N,PHI2N,PHI1,PHI2
       COMPLEX*16 SV11,SV12,SV21,SV22,RB11,RB21,RB11B,RB21B
       COMPLEX*16 R1B,R2B,P1B,P2B,Q1B,Q2B,WA(NWK)
       A=AX
       B=BX
       CO=(0.0,1.0)
       RB11=(Q1-P1*R1)/(A CO*R1*B)
       RB21=(Q2-P2*R2)/(A CO*R2*B)
       REALR1=R1
       REALR2=R2
       REALP1=P1
       REALP2=P2
       REALQ1=Q1
       REALQ2=Q2
       RRB11=RB11
       RRB21=RB21
       AIMGR1=CO*R1
       AIMGR2=CO*R2
       AIMGP1=CO*P1
       AIMGP2=CO*P2
       AIMGQ1=CO*Q1
       AIMGQ2=CO*Q2
       ARB11=CO*RB11
       ARB21=CO*RB21
       R1B=DCMPLX(REALR1,AIMGR1)
       R2B=DCMPLX(REALR2,AIMGR2)
       P1B=DCMPLX(REALP1,AIMGP1)
       P2B=DCMPLX(REALP2,AIMGP2)
       Q1B=DCMPLX(REALQ1,AIMGQ1)
       Q2B=DCMPLX(REALQ2,AIMGQ2)
```

```
      RB11B=DCMPLX(RRB11,ARB11)
      RB21B=DCMPLX(RRB21,ARB21)
      JJJ=0
      DO 1000 I=1,NB
      J=I*2
      THTAI=THTA(I)*DARCOS(-1.0D0)/180.D0
      C=DCOS(THTAI)
      S=DSIN(THTAI)
      P11=C*P1+S*Q1
      P12=C*P2+S*Q2
      P21=-S*P1+C*Q1
      P22=-S*P2+C*Q2
      T11=(C*C*R1*R1+S*S-2.*C*S*R1)
      T12=(C*C*R2*R2+S*S-2.*C*S*R2)
      T21=(-C*S*R1*R1+C*S-(C*C-S*S)*R1)
      T22=(-C*S*R2*R2+C*S-(C*C-S*S)*R2)
      Z1=X(I)+R1*Y(I)
      Z2=X(I)+R2*Y(I)
      Z11=CDSQRT(Z1*Z1-A*A-R1*R1*B*B)
      Z22=CDSQRT(Z2*Z2-A*A-R2*R2*B*B)
      REAL1=Z11
      AIMO1=-CO*Z11
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMO1).LE.1.D-16)AIMO1=0.0D0
      Z11=DCMPLX(REAL1,AIMO1)
      REAL2=Z22
      AIMO2=-CO*Z22
      IF(DABS(REAL2).LE.1.D-16)REAL2=0.0D0
      IF(DABS(AIMO2).LE.1.D-16)AIMO2=0.0D0
      Z22=DCMPLX(REAL2,AIMO2)
      XETA1=(Z1+Z11)/(A-CO*R1*B)
      IF(CDABS(XETA1).LT.0.999) GO TO 300
      GO TO 310
  300 Z11=-Z11
      XETA1=(Z1+Z11)/(A-CO*R1*B)
  310 XETA2=(Z2+Z22)/(A-CO*R2*B)
      IF(CDABS(XETA2).LT.0.999) GO TO 320
      GO TO 330
  320 Z22=-Z22
      XETA2=(Z2+Z22)/(A-CO*R2*B)
  330 CONTINUE
      JJJ=JJJ+1
C
C     NORMAL & TANGENTIAL STRESS BOUNDARY CONDITIONS ARE IMPOSED
C
      DO 5 N=1,NT
      NP=N
      CM(J-1,N)=NP*XETA1**NP*T11/Z11
      CM(J-1,2*NT+N)=NP*XETA2**NP*T12/Z22
      CM(J,N)=NP*XETA1**NP*T21/Z11
      CM(J,2*NT+N)=NP*XETA2**NP*T22/Z22
      NN=-N
      CM(J-1,NT+N)=NN*XETA1**NN*T11/Z11
      CM(J-1,3*NT+N)=NN*XETA2**NN*T12/Z22
      CM(J,NT+N)=NN*XETA1**NN*T21/Z11
      CM(J,3*NT+N)=NN*XETA2**NN*T22/Z22
    5 CONTINUE
      CM(J-1,NT8+1)=T11/Z11
      CM(J-1,NT8+2)=T12/Z22
      CM(J,NT8+1)=T21/Z11
```

```
      CM(J,NT8+2)=T22/Z22
 1000 CONTINUE
      DO 195 I=1,NB2
      DO 196 J=1,NT4
      REAL1=CM(I,J)
      AIMG1=-CO*CM(I,J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
      CM(I,J)=DCMPLX(REAL1,AIMG1)
      AIMG2=-AIMG1
      CM(I,NT4+J)=DCMPLX(REAL1,AIMG2)
  196 CONTINUE
  195 CONTINUE
      DO 295 I=1,NB2
      DO 296 J=1,2
      REAL1=CM(I,NT8+J)
      AIMG1=-CO*CM(I,NT8+J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
      CM(I,NT8+J)=DCMPLX(REAL1,AIMG1)
      AIMG2=-AIMG1
      CM(I,NT8+2+J)=DCMPLX(REAL1,AIMG2)
  296 CONTINUE
  295 CONTINUE
      SV11=(P2*Q1B-Q2*P1B)/(Q1*P2-Q2*P1)
      SV12=(P2*Q2B-Q2*P2B)/(Q1*P2-Q2*P1)
      SV21=(Q1*P1B-Q1B*P1)/(Q1*P2-Q2*P1)
      SV22=(Q1*P2B-Q2B*P1)/(Q1*P2-Q2*P1)
      DO 139 I=1,NB2
C
C     IMPOSE RIGID BODY ROTATION CONDITION
C
      CM(I,2*NT+1)=-CM(I,1)*RB21/RB11+CM(I,2*NT+1)
      CM(I,4*NT+1)=-CM(I,1)*RB11B/RB11+CM(I,4*NT+1)
      CM(I,6*NT+1)=-CM(I,1)*RB21B/RB11+CM(I,6*NT+1)
      CM(I,1)=(0.0,0.0)
C
C     IMPOSE SINGLE-VALUEDNESS CONDITION
C
      CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+1)*SV12+CM(I,NT8+4)
      CM(I,NT8+3)=CM(I,NT8+2)*SV21+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+2)*SV22+CM(I,NT8+4)
      CM(I,NT8+1)=(0.0,0.0)
      CM(I,NT8+2)=(0.0,0.0)
  139 CONTINUE
      DO 141 I=1,NB2
      DO 142 J=2,NT8
  142 CM(I,J-1)=CM(I,J)
      CM(I,NT8)=CM(I,NT8+3)
      CM(I,NT8+1)=CM(I,NT8+4)
  141 CONTINUE
      DO 95 I=1,NB2
      DO 96 J=1,NT8P1
      REAL1=CM(I,J)
      AIMG1=-CO*CM(I,J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
      CM(I,J)=DCMPLX(REAL1,AIMG1)
      AIMG2=-AIMG1
```

```
      CMC(I,J)=DCMPLX(REAL1,AIMG2)
   96 CONTINUE
   95 CONTINUE
      DO 120 I=1,NB
      J=I*2
      BC(J-1)=AIN1(I)
  120 BC(J)=AIN2(I)
      DO 100 I=1,NT8P1
      DO 100 J=1,NT8P1
      CSUM=(0.0,0.0)
      DO 110 K=1,NB2
  110 CSUM=CMC(K,I)*CM(K,J)+CSUM
      CMCTCM(I,J)=CSUM
  100 CONTINUE
      DO 130 I=1,NT8P1
      CSUM=(0.0,0.0)
      DO 140 K=1,NB2
  140 CSUM=CMC(K,I)*BC(K)+CSUM
  130 RHS(I)=CSUM
      IJOB=0
      M=1
      CALL LEQ2C(CMCTCM,NT8P1,NT8P1,RHS,M,NT8P1,IJOB,WA,WKK,IER)
      ORHS(1)=-(RHS(2*NT)*RB21+RHS(4*NT)*RB11B+RHS(6*NT)*RB21B)/RB11
      ORHS(8*NT+1)=RHS(8*NT)*SV11+RHS(8*NT+1)*SV12
      ORHS(8*NT+2)=RHS(8*NT)*SV21+RHS(8*NT+1)*SV22
      DO 151 I=2,NT8
  151 ORHS(I)=RHS(I-1)
C
C     STRESS AND STRAIN CALCULATION
C
      NRCS=NOUT+1
      IF(NOPT4.EQ.1.OR.NOPT4.EQ.3) NRCF=NOUT+NRCOUT(KJ)
      IF(NOPT4.EQ.2) NRCF=NOUT+NPST
      IF(NOPT4.EQ.4) NRCF=NOUT+NAST
      DO 190 K=1,NRCF
      Z1=XOUT(K)+R1*YOUT(K)
      Z2=XOUT(K)+R2*YOUT(K)
      Z11=CDSQRT(Z1*Z1-A*A -R1*R1*B*B)
      Z22=CDSQRT(Z2*Z2-A*A-R2*R2*B*B)
      XETA1=(Z1+Z11)/(A-CO*R1*B)
      IF(CDABS(XETA1).LT.0.999) GO TO 400
      GO TO 410
  400 Z11=-Z11
      XETA1=(Z1+Z11)/(A-CO*R1*B)
  410 XETA2=(Z2+Z22)/(A-CO*R2*B)
      IF(CDABS(XETA2).LT.0.999) GO TO 420
      GO TO 430
  420 Z22=-Z22
      XETA2=(Z2+Z22)/(A-CO*R2*B)
  430 CONTINUE
      PHI1DP=(0.0,0.0)
      PHI2DP=(0.0,0.0)
      PHI1DN=(0.0,0.0)
      PHI2DN=(0.0,0.0)
      PHI1P=(0.0,0.0)
      PHI2P=(0.0,0.0)
      PHI1N=(0.0,0.0)
      PHI2N=(0.0,0.0)
      DO 170 N=1,NT
      NP=N
```

```
      NN=-N
      PHI1DP=NP*XETA1**NP*GRHS(N)/Z11+PHI1DP
      PHI1DN=NN*XETA1**NN*GRHS(NT+N)/Z11+PHI1DN
      PHI2DP=NP*XETA2**NP*GRHS(2*NT+N)/Z22+PHI2DP
      PHI2DN=NN*XETA2**NN*GRHS(3*NT+N)/Z22+PHI2DN
      PHI1P=XETA1**NP*GRHS(N)+PHI1P
      PHI1N=XETA1**NN*GRHS(NT+N)+PHI1N
      PHI2P=XETA2**NP*GRHS(2*NT+N)+PHI2P
      PHI2N=XETA2**NN*GRHS(3*NT+N)+PHI2N
  170 CONTINUE
      PHI1D=PHI1DP+PHI1DN+GRHS(8*NT+1)/Z11
      PHI2D=PHI2DP+PHI2DN+GRHS(8*NT+2)/Z22
      PHI1=PHI1P+PHI1N+GRHS(8*NT+1)*CDLOG(XETA1)
      PHI2=PHI2P+PHI2N+GRHS(8*NT+2)*CDLOG(XETA2)
      SGMAX=2.*(R1*R1*PHI1D+R2*R2*PHI2D)
      SGMAY=2.*(PHI1D+PHI2D)
      SGMAXY=-2.*(R1*PHI1D+R2*PHI2D)
      EPSX=AMAT(1,1)*SGMAX+AMAT(1,2)*SGMAY+AMAT(1,3)*SGMAXY
      EPSY=AMAT(2,1)*SGMAX+AMAT(2,2)*SGMAY+AMAT(2,3)*SGMAXY
      EPSXY=AMAT(3,1)*SGMAX+AMAT(3,2)*SGMAY+AMAT(3,3)*SGMAXY
      U=2.*(P1*PHI1+P2*PHI2)
      V=2.*(Q1*PHI1+Q2*PHI2)
      PI=DARCOS(-1.D0)
      IF(XOUT(K).GT.0..AND.YOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI
      IF(XOUT(K).LT.0..AND.YOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180.
      IF(XOUT(K).LT.0..AND.YOUT(K).LT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180.
      IF(YOUT(K).LT.0..AND.XOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+360.
      C=DCOS(TETAA*PI/180.)
      S=DSIN(TETAA*PI/180.)
      SGMAR=C**2*SGMAX+S**2*SGMAY+2.*C*S*SGMAXY
      SGMAT=S**2*SGMAX+C**2*SGMAY-2.*C*S*SGMAXY
      SGMART=-C*S*SGMAX+C*S*SGMAY+(C**2-S**2)*SGMAXY
      EPSR=C**2*EPSX+S**2*EPSY+C*S*EPSXY
      EPST=S**2*EPSX+C**2*EPSY-C*S*EPSXY
      EPSRT=2.*(-C*S*EPSX+C*S*EPSY+(C**2-S**2)*(EPSXY/2.))
      UR=U*C+V*S
      RTHT(K)=TETAA
      REPX(K)=EPSX
      REPY(K)=EPSY
      REPXY(K)=EPSXY
      ASX(K)=SGMAX
      ASXY(K)=SGMAXY
      FUR(K)=UR
      FTHT(K)=TETAA
      FSMR(K)=SGMAR
  190 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FBOLT(ANOK,H,K,NOPT1,IM1)
C
C
C     FBOLT CALCULATES THE INDIVIDUAL PLY FOUNDATION
C     MODULI AND THE INDIVIDUAL PLY LOADS
```

```
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ATETAA(400),ANG(5,2),ASIGR(400),ASIGRT(400),H(2)
      DIMENSION ASIG1(400),ASIG2(400),ASIG6(400),UR(400),ANGK(5,2)
      DIMENSION FSMR(400),PLXPT(100)
      DIMENSION IPLY(100,2),NPLY(2),NUMPLY(2)
      DIMENSION FKI(100),PLX(100)
      DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2)
      DIMENSION RCA(2),RCB(2),NRC(2)
      COMMON/STRESS/ASIGR,ASIGRT,ASIG1,ASIG2,ASIG6
      COMMON/ELP/AX,BX,NOUT
      COMMON/FB1/BSTR,XSTR
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/FB2/UR,ATETAA,FSMR
      COMMON/MOD/E11,E22,ESS,PMU12,PMU21
      COMMON/RC/RCA,RCB,NRC
      COMMON/FCT/PLXPT
      RAD=DARCOS(-0.1D1)/180.
      THKTOT=NPLY(K)*H(K)
      NN=NUMPLY(K)
C
C     CALCULATE DELEFF
C
      WORK=0.
      PLOADX=0.
      IF(K.EQ.1) PLD=0.
      DO 210 KK=1,NOUT
      TH1=ATETAA(KK+1)-ATETAA(KK)
      TH2=(ATETAA(KK)+ATETAA(KK+1))/2.
      THETA=TH2*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      R=DSQRT(1./(C**2/AX**2+S**2/BX**2))
      FORCE=((FSMR(KK)+FSMR(KK+1))/2.)*R*TH1*RAD*THKTOT
      WORK=WORK+FORCE*.5*((UR(KK)+UR(KK+1))/2.)
      PLOADX=PLOADX+FORCE*C
  210 CONTINUE
      PLD=PLD+PLOADX
      DELEFF=WORK/PLOADX
C
C
C
C
C     COMPUTE PLY STRESSES FROM LAMINATE STRAINS
C
C     (SIGMA)R,O,RO = (Q)*(EPS)R,O,RO
C
C
      NN=NPLY(K)
      DO 100 J=1,NN
      LP=IPLY(J,K)
      THETA=ANG(LP,K)*RAD
      LI1=1
      LI2=NOUT
      NCAS=1
      CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
C
C
C     INTEGRATE AROUND CIRCULAR BOUNDARY FOR
```

```
C     INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION
C     MODULI
C
C
      NNN=LI2-1
      PLOADX=0.
      WK=0.
      DO 70 I=LI1,NNN
      TH1=ATETAA(I+1)-ATETAA(I)
      TH2=(ATETAA(I)+ATETAA(I+1))/2.
      THETA=TH2*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      R=DSQRT(1./(C**2/AX**2+S**2/BX**2))
      FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*R*TH1*RAD*H(K)
      FORCRT=((ASIGRT(I)+ASIGRT(I+1))/2.)*R*TH1*RAD*H(K)
      PLOADX=PLOADX+FORCR*C-FORCRT*S
   70 CONTINUE
      FKI(J)=DABS(PLOADX/(H(K)*DELEFF))
      PLX(J+(K-1)*NPLY(1))=PLOADX
  100 CONTINUE
      NT=NUMPLY(K)
      NN=NPLY(K)
      DO 310 I=1,NT
      DO 310 II=1,NN
      IF(IPLY(II,K).EQ.I) ANOK(I,K)=FKI(II)
      IF(IPLY(II,K).EQ.I) PLXPT(I)=PLX(II+(K-1)*NPLY(1))
  310 CONTINUE
      NP=NUMPLY(K)
C
C
C     COMPUTE TOTAL BEARING LOAD
C
C
      IF(K.EQ.1) GO TO 611
      PLXTOT=0.
      TH=H(1)*NPLY(1)+H(2)*NPLY(2)
      BLOAD=(BSTR*DARCOS(-1.0D0)*BX*TH)/2.
      NN=NPLY(1)+NPLY(2)
      DO 212 I=1,NN
      PLXTOT=PLXTOT+PLX(I)
  212 CONTINUE
  611 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FCRIT(SALOW,H,WTH,AST,K,NOPT1,NOPT4,BPR,NAVD,IL)
C
C
C
C     FAILURE LOAD CALCULATION
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASIO1(400),ASIO2(400),ASIO6(400),ASIOR(400)
      DIMENSION UR(400),FSMR(400),ATETAA(400),ASIORT(400),NUMPLY(2)
      DIMENSION ANO(5,2),IPLY(100,2),HFMC(5,2),PLXPT(100),NPLY(2)
      DIMENSION PNS(5,2),PBR(5,2),PSO(5,2),PALT(3,2)
```

```
      DIMENSION W(2),SALOW(2),SX(400),SXY(400),RCA(2),RCB(2),NRC(2)
      DIMENSION AEPS1(400),PFL(5,2),PSTC(5,5,2)
      DIMENSION BPSTS(2,10,2,3)
      COMMON/BP1/BPSTS
      COMMON/RC/RCA,RCB,NRC
      COMMON/FB1/BSTR,XSTR
      COMMON/STRSS2/AEPS1
      COMMON/FB2/UR,ATETAA,FSMR
      COMMON/FCT/PLXPT
      COMMON/ELP/AX,BX,NOUT
      COMMON/HFF/HFMC
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/FAL1/PNS,PBR,PSO,PALT
      COMMON/FAL3/PFL
      COMMON/STRESS/ASIOR,ASIORT,ASIO1,ASIO2,ASIO6
      COMMON/PSC2/PSTC
      COMMON/PSC3/SX,SXY
      RAD=DARCOS(-0.1D1)/180.
      IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,39) K
   39 FORMAT(//,' ANALYSIS OF PLATE NO',I5,' ',//)
      IF(NOPT4.EQ.2) GO TO 20
      IF(NOPT4.EQ.3) GO TO 90
      IF(NOPT4.EQ.4) GO TO 40
C
C
C     HOFFMAN/TSAI-HILL CRITERIA
C
C
      WRITE(6,10)
   10 FORMAT(//,' HOFFMAN/TSAI-HILL CRITERION',//)
      LI1=NOUT+1
      LI2=LI1+NRC(K)
      NCAS=2
      NN=NUMPLY(K)
      DO 402 I=1,NN
      THETA=ANG(I,K)*RAD
      CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
      PFAIL=1.0D10
      N1=NRC(K)
      IF(BPR.EQ.0.0) DSB=DABS(PLXPT(I))
      IF(BPR.EQ.1.0) DSB=DABS(XSTR)
      DO 404 J=1,N1
      S1=ASIO1(J)/DSB
      S2=ASIO2(J)/DSB
      S6=ASIO6(J)/DSB
      CALL HOFF(S1,S2,S6,A,B,K)
      NNN=NOUT+J
C
C
C     FOR EACH PLY TYPE FIND THE LOCATION AND MAGNITUDE
C     OF THE HIGHEST HOFFMAN/TSAI-HILL FAILURE INDEX VALUE
C
C
      P1=(-B+DSQRT(B**2+4*A))/(2.*A)
      P2=(-B-DSQRT(B**2+4*A))/(2.*A)
      IF(P1.LT.0.D0) PF=P2
      IF(P2.LT.0.D0) PF=P1
      IF(P1.LT.P2.AND.P1.GT.0.D0) PF=P1
      IF(P2.LT.P1.AND.P2.GT.0.D0) PF=P2
      IF(DABS(PF).GT.PFAIL) GO TO 480
```

```
      PFAIL=DABS(PF)
      AMAX=A
      BMAX=B
      LOC=J
  480 CONTINUE
  404 CONTINUE
      A=AMAX
      B=BMAX
      THTA=ATETAA(NOUT+LOC)
C
C
C     THE CORRESPONDING FAILURE LOAD IS OBTAINED FROM THE INDEX VALUE
C
C
      XULT=HFMC(1,K)
      IF(ASIO1(LOC).LT.0.) XULT=-HFMC(2,K)
      YULT=HFMC(3,K)
      IF(ASIC2(LOC).LT.0.) YULT=-HFMC(4,K)
      SULT=HFMC(5,K)
      IF(ASIO6(LOC).LT.0.) SULT=-HFMC(5,K)
      SR1=ASIO1(LOC)/XULT
      SR2=ASIO2(LOC)/YULT
      SR6=ASIO6(LOC)/SULT
      IF(BPR.EQ.0.0) WRITE(6,405) I,ANO(I,K),THTA,PFAIL,SR1,SR2,SR6
  405 FORMAT(/,' FOR PLY TYPE NO ',I5,' ( ',D9.3,' DEGREES )',/,
     *' THE HIGHEST FAILURE INDEX WAS FOUND AT  ',D9.3,' DEGREES',/,
     *' THE CORRESPONDING FAILURE LOAD =',D9.3,' LBS',/,
     *' THE STRESS RATIOS AT THIS LOCATION ARE:',/,
     *' SIG1/XULT = ',D9.3,/,
     *' SIG2/YULT = ',D9.3,/,
     *' SIG6/SULT = ',D9.3,//)
      PFL(I,K)=PFAIL
  402 CONTINUE
      IF(BPR.EQ.0.0) GO TO 80
      SFAIL=1.0D10
      DO 110 I=1,NN
      IF(SFAIL.GT.PFL(I,K)) NPY=I
  110 IF(SFAIL.GT.PFL(I,K)) SFAIL=PFL(I,K)
      PLFL=SFAIL*WTH*H(K)*NPLY(K)
      WRITE(6,771) PLFL
  771 FORMAT(//,' FOR THE OPEN HOLE LAMINATE, FAILURE',/,
     *' IS PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
      GO TO 80
   20 CONTINUE
C
C
C     POINT STRESS CRITERION
C
C
      IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,50)
   50 FORMAT(//,' POINT STRESS CRITERION ',//)
      NN=NUMPLY(K)
      NCAS=2
      LI1=NOUT+1
      LI2=LI1+2
      IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) NN=1
      DO 100 I=1,NN
      THETA=ANO(I,K)*RAD
      CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 705
```

```
      IF(IL.EQ.1) FAC=BPR
      IF(IL.EQ.2) FAC=(1.-BPR)/PLXPT(I)
      BPSTS(K,I,IL,1)=SX(1)*FAC/(AST*WTH*H(K)*NPLY(K))
      BPSTS(K,I,IL,2)=SX(2)*FAC/(AST*WTH*H(K)*NPLY(K))
      BPSTS(K,I,IL,3)=SXY(3)*FAC/(AST*WTH*H(K)*NPLY(K))
      GO TO 100
  705 CONTINUE
      IF(BPR.EQ.0.0) DSN=DABS(PLXPT(I))
      IF(BPR.EQ.1.0) DSN=XSTR
      PNT=DSN*PSTC(1,I,K)/DABS(SX(1))
      IF(SX(1).LT.0.) PNT=DSN*PSTC(2,I,K)/DABS(SX(1))
      PBN=DSN*PSTC(3,I,K)/DABS(SX(2))
      PSH=DSN*PSTC(4,I,K)/DABS(SXY(3))
      IF(BPR.EQ.0.0) WRITE(6,70) I,ANG(I,K),PNT,PBN,PSH
   70 FORMAT(/,' FOR PLY TYPE NUMBER ',I5,' WITH ',/,
     *' A PLY ORIENTATION OF ',D9.3,' DEGREES     ',/,
     *' NET SECTION FAILURE LOAD = ',D9.3,' LBS  ',/,
     *' BEARING FAILURE LOAD     = ',D9.3,' LBS  ',/,
     *' SHEAROUT FAILURE LOAD    = ',D9.3,' LBS  ',/)
      PNS(I,K)=PNT
      PBR(I,K)=PBN
      PSO(I,K)=PSH
  100 CONTINUE
      IF(BPR.EQ.0.0) GO TO 80
      N=NUMPLY(K)
      IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) N=1
      PFAIL1=1.0D10
      PFAIL2=1.0D10
      PFAIL3=1.0D10
      DO 781 I=1,N
      IF(PFAIL1.GT.PNS(I,K)) NPY1=I
      IF(PFAIL1.GT.PNS(I,K)) PFAIL1=PNS(I,K)
      IF(PFAIL2.GT.PBR(I,K)) NPY2=I
      IF(PFAIL2.GT.PBR(I,K)) PFAIL2=PBR(I,K)
      IF(PFAIL3.GT.PSO(I,K)) NPY3=I
  781 IF(PFAIL3.GT.PSO(I,K)) PFAIL3=PSO(I,K)
      IF(PFAIL1.GE.PFAIL2.OR.PFAIL1.GE.PFAIL3) GO TO 813
      PFAIL1=PFAIL1*WTH*H(K)*NPLY(K)
      WRITE(6,982) PFAIL1
      WRITE(6,814)
      GO TO 811
  813 IF(PFAIL2.GE.PFAIL1.OR.PFAIL2.GE.PFAIL3) GO TO 812
      PFAIL2=PFAIL2*WTH*H(K)*NPLY(K)
      WRITE(6,982) PFAIL2
      WRITE(6,815)
      GO TO 811
  812 IF(PFAIL3.GE.PFAIL1.OR.PFAIL3.GE.PFAIL2) GO TO 811
      PFAIL3=PFAIL3*WTH*H(K)*NPLY(K)
      WRITE(6,982) PFAIL3
      WRITE(6,816)
  811 CONTINUE
  982 FORMAT(//,' FOR THE LAMINATE WITH AN OPEN HOLE, FAILURE ',/,
     *' IS PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS ',/)
  814 FORMAT(' PREDICTED FAILURE MODE IS NET SECTION',//)
  815 FORMAT(' PREDICTED FAILURE MODE IS BEARING FAILURE',//)
  816 FORMAT(' PREDICTED FAILURE MODE IS SHEAR-OUT FAILURE',//)
      GO TO 80
C
C     AVERAGE STRESS CRITERION
C
```

```
   40 CONTINUE
C
C
      IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,55)
   55 FORMAT(//,' AVERAGE STRESS CRITERION ',//)
      LI1=NOUT+1
      NN=NUMPLY(K)
      NCAS=2
      IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) NN=1
      DO 105 I=1,NN
      LI2=NOUT+3*NAVD
      THETA=ANG(I,K)*RAD
      CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
C
C     CALCULATE AVERAGE STRESS
C
      SUM=0.
      N1=1
      N2=NAVD
      DO 200 IJ=N1,N2
  200 SUM=SUM+SX(IJ)
      AS1=SUM/FLOAT(NAVD)
      N1=NAVD+1
      N2=2*NAVD
      SUM=0.
      DO 215 IJ=N1,N2
  215 SUM=SUM+SXY(IJ)
      AS2=SUM/FLOAT(NAVD)
      SUM=0.
      N1=2*NAVD+1
      N2=3*NAVD
      DO 220 IJ=N1,N2
  220 SUM=SUM+SX(IJ)
      AS3=SUM/FLOAT(NAVD)
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 720
      IF(IL.EQ.1) FAC=BPR
      IF(IL.EQ.2) FAC=(1.-BPR)/PLXPT(I)
      BPSTS(K,I,IL,1)=AS1*FAC/(AST*WTH*H(K)*NPLY(K))
      BPSTS(K,I,IL,2)=AS3*FAC/(AST*WTH*H(K)*NPLY(K))
      BPSTS(K,I,IL,3)=AS2*FAC/(AST*WTH*H(K)*NPLY(K))
      GO TO 105
  720 CONTINUE
      IF(BPR.EQ.0.0) DSN=DABS(PLXPT(I))
      IF(BPR.EQ.1.0) DSN=XSTR
      PNT=DSN*PSTC(1,I,K)/DABS(AS1)
      IF(AS1.LT.0.) PNT=DSN*PSTC(2,I,K)/DABS(AS1)
      PBN=DSN*PSTC(3,I,K)/DABS(AS3)
      PSH=DSN*PSTC(4,I,K)/DABS(AS2)
      IF(BPR.EQ.0.0) WRITE(6,75) I,ANG(I,K),PNT,PBN,PSH
   75 FORMAT(/,' FOR PLY TYPE NUMBER ',I5,' WITH ',/,
     *' A PLY ORIENTATION OF ',D..3,' DEGREES    ',/,
     *' NET SECTION FAILURE LOAD = ',D9.3,' LBS   ',/,
     *' BEARING FAILURE LOAD     = ',D9.3,' LBS   ',/,
     *' SHEAROUT FAILURE LOAD    = ',D9.3,' LBS   ',/)
      PNS(I,K)=PNT
      PBR(I,K)=PBN
      PSO(I,K)=PSH
  105 CONTINUE
      IF(BPR.EQ.0.0) GO TO 80
      N=NUMPLY(K)
```

```
      IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) N=1
      PFAIL1=1.0D10
      PFAIL2=1.0D10
      PFAIL3=1.0D10
      DO 718 I=1,N
      IF(PFAIL1.GT.PNS(I,K)) NPY1=I
      IF(PFAIL1.GT.PNS(I,K)) PFAIL1=PNS(I,K)
      IF(PFAIL2.GT.PBR(I,K)) NPY2=I
      IF(PFAIL2.GT.PBR(I,K)) PFAIL2=PBR(I,K)
      IF(PFAIL3.GT.PSO(I,K)) NPY3=I
  718 IF(PFAIL3.GT.PSO(I,K)) PFAIL3=PSO(I,K)
      IF(PFAIL1.GE.PFAIL2.OR.PFAIL1.GE.PFAIL3) GO TO 883
      PFAIL1=PFAIL1*WTH*H(K)*NPLY(K)
      WRITE(6,478) PFAIL1
      WRITE(6,884)
      GO TO 881
  883 IF(PFAIL2.GE.PFAIL1.OR.PFAIL2.GE.PFAIL3) GO TO 882
      PFAIL2=PFAIL2*WTH*H(K)*NPLY(K)
      WRITE(6,478) PFAIL2
      WRITE(6,885)
      GO TO 881
  882 IF(PFAIL3.GE.PFAIL1.OR.PFAIL3.GE.PFAIL2) GO TO 881
      PFAIL3=PFAIL3*WTH*H(K)*NPLY(K)
      WRITE(6,478) PFAIL3
      WRITE(6,886)
  881 CONTINUE
  478 FORMAT(//,' FOR THE LAMINATE WITH THE OPEN HOLE, FAILURE',/,
     *' IS PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',/)
  884 FORMAT(' PREDICTED FAILURE MODE IS NET SECTION',//)
  885 FORMAT(' PREDICTED FAILURE MODE IS BEARING FAILURE',//)
  886 FORMAT(' PREDICTED FAILURE MODE IS SHEAR-OUT FAILURE',//)
      GO TO 80
C
C     MAXIMUM STRAIN CRITERION
C
   90 CONTINUE
      PAPP=XSTR*H(K)*NPLY(K)*WTH
      WRITE(6,772)
  772 FORMAT(//,' MAXIMUM STRAIN CRITERION ',//)
      LI1=NOUT+1
      LI2=LI1+NRC(K)
      NCAS=2
      NN=NUMPLY(K)
      DO 210 I=1,NN
      THETA=ANG(I,K)*RAD
      CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
      STMAX=-1.0D10
      N1=NRC(K)
      DO 510 J=1,N1
      IF(STMAX.LT.DABS(AEPS1(J))) LOC=J
  510 IF(STMAX.LT.DABS(AEPS1(J))) STMAX=DABS(AEPS1(J))
      THTA=ATETAA(NOUT+LOC)
      IF(BPR.EQ.1.0) GO TO 511
      PFL(I,K)=DABS(PLXPT(I)*SALOW(K)/STMAX)
      WRITE(6,774) I,ANG(I,K),THTA,PFL(I,K)
  774 FORMAT(' FOR PLY TYPE NUMBER ',I5,' WITH ',/,
     *' A PLY ORIENTATION OF ',D9.3,' DEGREES  ',/,
     *' FAILURE IS PREDICTED AT ',D9.3,' DEGREES ',/,
     *' AT A PLY LOAD OF ',D9.3,' LBS',//)
      GO TO 210
```

```
  511 PFL(I,K)=DABS(PAPP*SALOW(K)/STMAX)
  210 CONTINUE
      IF(BPR.EQ.0.0) GO TO 80
      A=1.0D10
      NN=NUMPLY(K)
      DO 514 I=1,NN
      IF(A.GT.PFL(I,K)) NPY=I
  514 IF(A.GT.PFL(I,K)) A=PFL(I,K)
      WRITE(6,778) A
  778 FORMAT(//,' FOR THE OPEN HOLE LAMINATE, FAILURE IS',/,
     *' PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
   80 RETURN
      END
C
C
C
      SUBROUTINE QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
C
C     QMATX PERFORMS BASIC STRESS AND STRAIN
C     TRANSFORMATIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASIGR(400),ASIGRT(400),ASIG1(400),ASIG2(400),ASIG6(400)
      DIMENSION ATETAA(400),AEPSX(400),AEPSY(400),AEPSXY(400)
      DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400)
      DIMENSION AEPS1(400)
      DIMENSION ASX(400),ASXY(400)
      COMMON/XXY1/ASX,ASXY
      COMMON/MOD/E11,E22,ESS,PMU12,PMU21
      COMMON/STRSS2/AEPS1
      COMMON/STRESS/ASIGR,ASIGRT,ASIG1,ASIG2,ASIG6
      COMMON/QMT/ATETAA,AEPSX,AEPSY,AEPSXY
      COMMON/PSC3/SX,SXY
      J=0
      Q11=E11(K)/(1.0-PMU12(K)*PMU21(K))
      Q12=(PMU21(K)*E11(K))/(1.0-PMU12(K)*PMU21(K))
      Q22=E22(K)/(1.0-PMU12(K)*PMU21(K))
      Q66=ESS(K)
      C=DCOS(THETA)
      S=DSIN(THETA)
      BQ11=(Q11*(C**4))+(2.*(Q12+(2.*Q66))*(C**2)*(S**2))+(Q22*(S**4))
      BQ12=((Q11+Q22-(4.*Q66))*(S**2)*(C**2))+(Q12*(S**4+C**4))
      BQ16=((Q11-Q12-(2.*Q66))*(S*(C**3)))+((Q12-Q22+(2.*Q66))*((S**3)*C
     *))
      BQ22=(Q11*(S**4))+(2.*(Q12+(2.*Q66))*(S**2)*(C**2))+(Q22*(C**4))
      BQ26=((Q11-Q12-(2.*Q66))*C*(S**3))+((Q12-Q22+(2.*Q66))*S*(C**3))
      BQ66=((Q11+Q22-(2.*(Q12+Q66)))*(S**2)*(C**2))+(Q66*((C**4)+(S**4)
     *))
      DO 40 I=LI1,LI2
      J=J+1
      IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      SIGX=BQ11*AEPSX(I)+BQ12*AEPSY(I)+BQ16*AEPSXY(I)
      SIGY=BQ12*AEPSX(I)+BQ22*AEPSY(I)+BQ26*AEPSXY(I)
      SIGXY=BQ16*AEPSX(I)+BQ26*AEPSY(I)+BQ66*AEPSXY(I)
      SX(J)=SIGX
      SXY(J)=SIGXY
      IF(NOPT1.EQ.2) SX(J)=ASX(I)
      IF(NOPT1.EQ.2) SXY(J)=ASXY(I)
```

```
      ASIGR(I)=SIGX*C**2+SIGY*S**2+2.*SIGXY*S*C
      ASIGRT(I)=-SIGX*S*C+SIGY*C*S+SIGXY*(C**2-S**2)
      ASIG1(J)=SIGX*C**2+SIGY*S**2+2.*S*C*SIGXY
      ASIG2(J)=SIGX*S**2+SIGY*C**2-2.*S*C*SIGXY
      ASIG6(J)=-C*S*SIGX+SIGY*C*S+(C**2-S**2)*SIGXY
      AEPS1(J)=AEPSX(I)*C**2+AEPSY(I)*S**2+AEPSXY(I)*S*C
   40 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE HOFF(S1,S2,S6,A,B,K)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION HFMC(5,2)
      COMMON/HFF/HFMC
C
C     COMPUTE THE HOFFMAN/TSAI-HILL FAILURE INDEX
C
      A=0.0D0
      B=0.0D0
      XC=HFMC(1,K)
      XT=HFMC(2,K)
      YC=HFMC(3,K)
      YT=HFMC(4,K)
      STC=HFMC(5,K)
      A=(S1**2-S1*S2)/(XC*XT)+(S2**2)/(YC*YT)+(S6/STC)**2
      IF(XC.EQ.XT.AND.YC.EQ.YT) GO TO 10
      B=((XC-XT)/(XC*XT))*S1+((YC-YT)/(YC*YT))*S2
      GO TO 20
   10 CONTINUE
      B=0.0D0
   20 CONTINUE
      RETURN
      END
C
C
C
C
C
C
      SUBROUTINE CENTD(RF,H,FASSS,FASBS,P,DELP,ITT)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PLYK(100),BARK(100),BARU(100),F(100)
      DIMENSION H(2),RF(2)
      DIMENSION AII(100,100),A(2),B(2)
      DIMENSION NPLY(2)
      COMMON/PBB/PLYK,BARK,BARU
      COMMON/AFM/AII,F
      COMMON/LYP/NPLY
C
C     SET UP THE CENTRAL DIFFERENCE EQUATIONS
C
      DO 3 I=1,100
      DO 3 J=1,100
    3 AII(I,J)=0.
C
C     NECESSARY CONSTANTS ARE FORMED
```

```
C
      DO 7 I=1,2
      A(I)=H(I)**2/FASSS
    7 B(I)=H(I)**4/FASBS
      H12=H(1)/H(2)
      A1=H(1)**2/FASSS
      A2=H(2)**2/FASSS
      NP=NPLY(1)+NPLY(2)
C
C
C     SHEAR AT TOP OF JOINT EQUALS ZERO
C
      AII(1,1)=1.
      AII(1,2)=-(2.+A1*PLYK(2))
      AII(1,4)=2.+A1*PLYK(2)
      AII(1,5)=-1.
      F(1)=0.0
C
C     MOMENT CONDITION AT TOP
C
      IF(RF(1).GE.1.D10) GO TO 50
      Z=1.
      R=RF(1)
      GO TO 60
   50 Z=0.
      R=1.
   60 AII(2,1)=R
      AII(2,2)=(Z*2.*H(1)*FASSS)+R*(-2.-A1*PLYK(2)+(H(1)**2
     **FASSS)/FASBS)
      AII(2,3)=-Z*(4.*H(1)*FASSS+(2*H(1)**2*PLYK(1)*H(1)))
      AII(2,4)=Z*2.*H(1)*FASSS+R*(2.+A1*PLYK(2)-(H(1)**2
     **FASSS)/FASBS)
      AII(2,5)=-R
      F(2)=Z*2.*H(1)**3*BARK(1)*BARU(1)
C
C     GOVERNING EQUATIONS FOR THE TOP PLATE
C
      N2=NPLY(1)
      DO 55 J=1,N2
      I=J+2
      AII(I,J)=1.
      IF(J.EQ.1) GO TO 56
      AII(I,J+1)=-4.-A(1)*PLYK(J-1)
      GO TO 57
   56 AII(I,J+1)=-4.-A(1)*PLYK(2)
   57 AII(I,J+2)=6.+(2.*A(1)+B(1))*PLYK(J)
      IF(J.EQ.N2) GO TO 61
      AII(I,J+3)=-4.-A(1)*PLYK(J+1)
      GO TO 62
   61 AII(I,J+3)=-4.-A(1)*PLYK(NPLY(1)-1)
   62 AII(I,J+4)=1.
      IF(J.EQ.1) GO TO 58
      IF(J.EQ.N2) GO TO 63
      F(I)=A(1)*BARK(J-1)*BARU(J-1)
     *-(2.*A(1)+B(1))*BARK(J)*BARU(J)
     *+A(1)*BARK(J+1)*BARU(J+1)
      GO TO 59
   58 F(I)=2.*A(1)*BARK(2)*BARU(2)
     *-(2.*A(1)+B(1))*BARK(1)*BARU(1)
      GO TO 59
```

```
   63 F(I)=2.*[illegible])*BARK(NPLY(1)-1)*BARU(NPLY(1)-1)
     *-(2.*A(1)+B(1))*BARK(J)*BARU(J)
   59 CONTINUE
   55 CONTINUE
C
C     INTERFACE SHEAR ON TOP PLATE = P+DELP
C
      I=NPLY(1)+3
      J=NPLY(1)
      AII(I,J)=1.
      AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1))
      AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)
      AII(I,J+4)=-1.
      F(I)=(-2.*H(1)**3*(P+DELP))/FASBS
C
C     SLOPE CONTINUITY
C
      I=NPLY(1)+4
      J=NPLY(1)
      AII(I,J)=1.
      AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS)
      AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS
      AII(I,J+4)=-1.
      AII(I,J+5)=-H12**3
      AII(I,J+6)=H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSS/FASBS)
      AII(I,J+8)=-H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSS/FASBS)
      AII(I,J+9)=H12**3
      F(I)=0.
C
C     MOMENT CONTINUITY
C
      I=NPLY(1)+5
      J=NPLY(1)+1
      AII(I,J)=1.
      AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)))
      AII(I,J+2)=1.
      AII(I,J+5)=-H12**2
      AII(I,J+6)=H12**2*(2.+A2*PLYK(NPLY(1)+1))
      AII(I,J+7)=-H12**2
      F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*
     *BARU(NPLY(1)+1))
C
C     INTERFACE SHEAR ON BOTTOM PLATE
C
      I=NPLY(1)+6
      J=NPLY(1)+5
      AII(I,J)=-1.
      AII(I,J+1)=(2.+A2*PLYK(NPLY(1)+2))
      AII(I,J+3)=-(2.+A2*PLYK(NPLY(1)+2))
      AII(I,J+4)=1.
      F(I)=2.*H(2)**3*(P+DELP)/FASBS
C
C     GOVERNING EQUATIONS FOR THE BOTTOM PLATE
C
      N1=NPLY(1)+7
      N2=NPLY(1)+NPLY(2)+6
      DO 70 I=N1,N2
      J=I-2
      AII(I,J)=1.
      IF(I.EQ.N1) GO TO 71
```

```
      AII(I,J+1)=-4.-A(2)*PLYK(J-5)
      GO TO 72
   71 AII(I,J+1)=-4.-A(2)*PLYK(NPLY(1)+2)
   72 AII(I,J+2)=6.+(2.*A(2)+B(2))*PLYK(J-4)
      IF(I.EQ.N2) GO TO 75
      AII(I,J+3)=-4.-A(2)*PLYK(J-3)
      GO TO 76
   75 AII(I,J+3)=-4.-A(2)*PLYK(J-5)
   76 AII(I,J+4)=1.
      IF(I.EQ.N1) GO TO 73
      IF(I.EQ.N2) GO TO 77
      F(I)=A(2)*BARK(J-5)*BARU(J-5)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
     *+A(2)*BARK(J-3)*BARU(J-3)
      GO TO 74
   73 F(I)=2*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
      GO TO 74
   77 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
   74 CONTINUE
   70 CONTINUE
C
C     SHEAR ON BOTTOM PLATE EQUALS ZERO
C
      NP=NPLY(1)+NPLY(2)
      I=NP+7
      J=NP+4
      AII(I,J)=-1.
      AII(I,J+1)=(2.+A2*PLYK(NP-1))
      AII(I,J+3)=-(2.+A2*PLYK(NP-1))
      AII(I,J+4)=1.
      F(I)=0.
C
C     MOMENT BOUNDARY CONDITION ON BOTTOM PLATE
C
      I=NP+8
      IF(RF(2).GE.1.D10) GO TO 85
      Z=1.
      R=RF(2)
      GO TO 95
   85 Z=0.
      R=1.
   95 AII(I,J)=-R
      AII(I,J+1)=Z*(2.*H(2)*FASSS)+R*(2.+A2*PLYK(NP-1)
     *-H(2)**2*FASSS/FASBS)
      AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)**3*PLYK(NP))
      AII(I,J+3)=Z*2.*H(2)*FASSS+R*(-2.-A2*PLYK(NP-1)
     *+H(2)**2*FASSS/FASBS)
      AII(I,J+4)=R
      F(I)=Z*(2.*H(2)**3*BARK(NP)*BARU(NP))
      RETURN
      END
C
C
      SUBROUTINE SOLVE(U,H,P,DELP,NSDLS,ITT)
C
C
C
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(100,100),B(100),NPLY(2),U(100),F(100)
      DIMENSION SX(100),PLYK(100),H(2)
      DIMENSION BARK(100),BARU(100)
      COMMON/LYP/NPLY
      COMMON/AFM/A,F
      COMMON/PBB/PLYK,BARK,BARU
C
C     SOLUTION OF THE SYSTEM: 9AI(U)=(B)
C
      NP=NPLY(1)+NPLY(2)+8
      DO 444 I=1,NP
  444 B(I)=F(I)
C
C
C
C
C     APPLYING GUASSIAN ELIMINATION TO THE
C     MATRIX OF COEFFICIENTS
C
      DO 2001 I=1,NP
      IR=I
 2042 IF(A(IR,I).NE.0.) GO TO 2041
      IR=IR+1
      IF(IR.GT.NP) GO TO 2001
      GO TO 2042
 2041 NN=IR+1
      DO 2002 L=NN,NP
      IF(DABS(A(L,I)).GT.1.D-30) GO TO 2009
      A(L,I)=0.
      GO TO 2002
 2009 CF=-A(IR,I)/A(L,I)
      DO 2003 J=I,NP
      A(L,J)=A(L,J)*CF+A(IR,J)
      IF(DABS(A(L,J)).LT.1.D-30) A(L,J)=0.0
 2003 CONTINUE
      B(L)=B(L)*CF+B(I)
 2002 CONTINUE
 2001 CONTINUE
C
C     BACK SUBSTITUTION
C
      DO 2011 I=1,NP
      L=NP+1-I
      SUM=0.
      IF(A(L,L).EQ.0.) GO TO 2112
      N=L+1
      IF(N.GT.NP) GO TO 2013
      DO 2013 J=N,NP
      SUM=SUM-A(L,J)*SX(J)
 2013 CONTINUE
      SX(L)=(B(L)+SUM)/A(L,L)
      GO TO 2011
 2112 CONTINUE
      SX(L)=0.
 2011 CONTINUE
C
C     EQUILIBRIUM CHECK
C
      NPTS=NPLY(1)+NPLY(2)+8
```

```
      PT=P+DELP
      N1=NPLY(1)+2
      N2=NPLY(1)+7
      NN=NPLY(1)+NPLY(2)+6
      SUM4=0.
      SUM5=0.
      DO 1444 I=3,N1
      J=I-2
      U(J)=SX(I)
      SUM4=SUM4+SX(I)*PLYK(J)*H(1)
 1444 CONTINUE
      DO 1555 I=N2,NN
      J=I-6
      U(J)=SX(I)
      SUM5=SUM5+SX(I)*PLYK(J)*H(2)
 1555 CONTINUE
      IF(NSDLS.EQ.1) GO TO 810
      PT=PT*2.
      SUM4=SUM4*2.
      SUM5=SUM5*2.
  810 CONTINUE
      NP=NPLY(1)+NPLY(2)
      N=NPLY(1)+NPLY(2)
      II=1
      DO 311 I=1,N
      IF(I.GT.NPLY(1)) II=2
      PL=U(I)*PLYK(I)*H(II)
      IF(I.LE.NPLY(1)) GO TO 311
  311 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FAIL(GAMDL,U,H,P,DELP,BPR,AST,WTH,PFAIL,ANGLE,NODE,
     *IROUT,NOPT4,NULTF,JNT,ITT,NTFL)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NPLY(2),MDAMP(100),H(2),PLYK(100),U(100)
      DIMENSION BARK(100),BARU(100)
      DIMENSION PN(100),MDAMI(100),GAMDL(2),GAMN(100)
      DIMENSION DELNS(5,2),DELBR(5,2),DELSO(5,2)
      DIMENSION UN(100),PFL(5,2),PSTC(5,5,2)
      DIMENSION IPLY(100,2),ANG(5,2),NUMPLY(2)
      DIMENSION PNS(5,2),PBR(5,2),PSO(5,2),PALT(3,2)
      DIMENSION BPSTS(2,10,2,3)
      DIMENSION NPNM(100,2)
      COMMON/COUNT/NPNM
      COMMON/BP1/BPSTS
      COMMON/PSC2/PSTC
      COMMON/FAL1/PNS,PBR,PSO,PALT
      COMMON/FAL3/DELNS,DELBR,DELSO
      COMMON/FAL4/UN,GAMN,MDAMP,MDAMI,PN
      COMMON/FAL5/PFL
      COMMON/PBB/PLYK,BARK,BARU
      COMMON/PRT/NDAM,INPLY,ITYP
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      NP=NPLY(1)+NPLY(2)
C
C
```

```
C     FAIL INCREMENTS THE POINT LOAD TO EACH SUCCESSIVE
C     PLY AND INTERFACE FAILURE UNTIL FINAL JOINT FAILURE
C     TAKES PLACE
C
C     FULL BEARING FAILURE ANALYSIS
C
      IF(BPR.NE.0.0) GO TO 600
      IROUT=1
C
C     LOOP OVER ALL PLIES TO FIND LOAD, LOCATION, AND MODE OF
C     NEXT PLY FAILURE
C
  100 IF(DELP.EQ.0.) GO TO 10
      PFP=1.0D10
      GO TO 15
   10 PFP=1000.
   15 MODEF=0
      DELPF=0.D0
      NN=NPLY(1)+NPLY(2)
      DO 20 I=1,NN
C
C     IF PLY HAS ALREADY LOST STIFFNESS, GO
C     ON TO THE NEXT PLY
C
      IF(MDAMP(I).EQ.10) GO TO 20
C
C     DETERMINE WHICH PLATE THIS PLY IN IN
C
      K=1
      IF(I.GT.NPLY(1)) K=2
C
C     CALCULATE THE LOAD ON PLY FOR CURRENT JOINT LOAD
C
      PL=-H(K)*(PLYK(I)*U(I)+BARK(I)*BARU(I))
C
C     ASSUME FAILURE OCCURS ONLY ON BEARING SIDE
C
C
C
      IF(PL.LT.0..AND.K.EQ.1) GO TO 20
      IF(PL.GT.0..AND.K.EQ.2) GO TO 20
C
C     DETERMINE PLY LOAD NECESSARY TO CAUSE NEXT
C     FAILURE AND ITS MODE
C
      IF(NOPT4.NE.1.AND.NOPT4.NE.3) GO TO 200
      MODE=8
      IN=I-(K-1)*NPLY(1)
      NPY=IPLY(IN,K)
      PF=PFL(NPY,K)
C
C
C     IF PL>PF AT CURRENT JOINT LOAD
C     PREDICT FAILURE
C
      NCC=0
      IF(DELP.NE.0.) GO TO 210
      IF(DABS(PL).LT.DABS(PF)) GO TO 210
      PFP=0.
      INPLY=I
```

```
      MODEF=MODE
      NCC=1
      GO TO 140
  200 NMN=I
      IF(I.GT.NPLY(1)) NMN=I-NPLY(1)
      NX=IPLY(NMN,K)
      IF(PBR(NX,K).LT.PSO(NX,K).OR.PNS(NX,K).LT.PSO(NX,K)) GO TO 700
      MODE=1
      PF=PSO(NX,K)
      IF(MDAMP(I).EQ.1) MODE=5
      IF(MDAMP(I).EQ.1) PF=PALT(1,K)*PF
      GO TO 25
  700 IF(PNS(NX,K).LT.PBR(NX.K)) GO TO 710
      MODE=2
      PF=PBR(NX,K)
      IF(MDAMP(I).EQ.2) MODE=6
      IF(MDAMP(I).EQ.2) PF=PALT(2,K)*PF
      GO TO 25
  710 MODE=3
      PF=PNS(NX,K)
      IF(MDAMP(I).EQ.3) MODE=7
      IF(MDAMP(I).EQ.3) PF=PALT(3,K)*PF
   25 CONTINUE
      NCC=0
      IF(DELP.NE.0.) GO TO 210
      IF(DABS(PL).LT.DABS(PF)) GO TO 210
      PFP=0.
      INPLY=I
      MODEF=MODE
      NCC=1
      GO TO 1212
  210 CONTINUE
C
C     DETERMINE INCREMENTAL JOINT LOAD TO CAUSE
C     PLY FAILURE
C
      IF(ITT.LE.1) GO TO 21
      IF(DABS(DABS(U(I)/UN(I))-1.).LT.1.0D-10) GO TO 20
   21 CONTINUE
      DELPF=(PF-DABS(PN(I)))*1000./(DABS(PL)-DABS(PN(I)))
C
C     A NEGATIVE VALUE OF DELPF INDICATES UNLOADING
C     IN A PLY. THIS NODE IS THEN SKIPPED
C
      IF(DELPF.LT.0.) GO TO 20
C
C     RECORD LOWEST JOINT FAILURE LOAD INCREMENT,
C     PLY IN WHICH IT OCCURS, AND MODE
C
      PFP2=PFP
      IF(DELP.EQ.0) PFP2=1.
      IF(DELPF.GT.PFP2) GO TO 20
      PFP=DELPF
      INPLY=I
      MODEF=MODE
   20 CONTINUE
C
C     LOOP OVER ALL INTERFACES TO FIND LOAD
C     AND LOCATION OF NEXT DELAMINATION
C
```

```
      NN=NPLY(1)+NPLY(2)-2
      DO 50 J=1,NN
C
C     IF INTERFACE HAS ALREADY FAILED, GO TO NEXT
C
      IF(MDAMI(J).EQ.1) GO TO 50
C
C     DETERMINE WHICH PLATE INTERFACE IS IN
C
      K=1
      IF(J.GE.NPLY(1)) K=2
C
C     CALCULATE INTERFACE SHEAR STRAIN FOR CURRENT
C     JOINT LOAD
C
      GAMJ=(U(J+K-1)-U(J+K))/H(K)
C
C     DETERMINE INCREMENTAL JOINT LOAD TO CAUSE
C     INTERFACE FAILURE
C
      IF(ITT.EQ.1) GO TO 47
      IF(DABS(DABS(GAMJ/GAMN(J))-1.).LT.1.0D-10) GO TO 50
   47 CONTINUE
      DELPF=(GAMDL(K)-DABS(GAMN(J)))/(DABS(GAMJ)-DABS(GAMN(J)))*1000.
      IF(DELPF.LT.0.) GO TO 50
C
C
C     RECORD LOWEST JOINT FAILURE LOAD INCREMENT,
C     PLY OF INTERFACE IN WHICH IT OCCURS, AND MODE
C
      PFP2=PFP
      IF(DELP.EQ.0) PFP2=1.
      IF(DELPF.GT.PFP2) GO TO 50
      PFP=DELPF
      INPLY=J
      MODEF=9
   50 CONTINUE
C
C     DETERMINE VALUES AT END OF INCREMENT
C
C     JOINT LOAD AT FAILURE
C
      IF(MODEF.EQ.0) GO TO 325
      P=P+PFP*DELP/1000.
  325 CONTINUE
C
C     NODAL DISPLACEMENTS AND PLY LOADS
C
      NN=NPLY(1)+NPLY(2)
      DO 55 I=1,NN
      UN(I)=UN(I)+(U(I)-UN(I))*PFP/1000.
C
C     UPDATE UN
C
      IF(NCC.EQ.1) UN(I)=U(I)
      K=1
      IF(I.GT.NPLY(1)) K=2
      PN(I)=-H(K)*(PLYK(I)*UN(I)+BARK(I)*BARU(I))
   55 CONTINUE
C
```

```
C     INTERFACE SHEAR STRAINS
C
      NN=NPLY(1)+NPLY(2)-2
      DO 60 J=1,NN
      K=1
      IF(J.GE.NPLY(1)) K=2
      GAMN(J)=(UN(J+K-1)-UN(J+K))/H(K)
   60 CONTINUE
 1212 CONTINUE
C
C     PLY STIFFNESSES, DAMAGE STATES, AND NEXT LOAD
C     INCREMENT
C
      K=1
      IF(INPLY.GT.NPLY(1)) K=2
      NMN=INPLY
      IF(INPLY.GT.NPLY(1)) NMN=INPLY-NPLY(1)
      NX=IPLY(NMN,K)
      IF(MODEF.NE.0) GO TO 70
      DELP=1000.
      NDAM=1
      GO TO 65
   70 IF(MODEF.NE.1) GO TO 80
      IF(INPLY.EQ.1.AND.MDAMI(INPLY).EQ.1) GO TO 75
      IF(INPLY.EQ.NPLY(1).AND.MDAMI(INPLY-1).EQ.1) GO TO 75
      IF(INPLY.EQ.(NPLY(1)+1).AND.MDAMI(INPLY-1).EQ.1) GO TO 75
      IF(INPLY.EQ.(NPLY(1)+NPLY(2)).AND.MDAMI(NPLY(1)+NPLY(2)
     1-2).EQ.1) GO TO 75
      KK=0
      IF(INPLY.GT.NPLY(1)) KK=1
      IF(MDAMI(INPLY-KK-1).EQ.1.AND.MDAMI(INPLY-KK).EQ.1) GO TO 75
      MDAMP(INPLY)=1
      TEMPK=PLYK(INPLY)
      PLYK(INPLY)=DELSO(NX,K)*PLYK(INPLY)
      BARK(INPLY)=(1.-DELSO(NX,K))*TEMPK
      BARU(INPLY)=UN(INPLY)
      DELP=0.
      NDAM=2
      GO TO 65
   75 PLYK(INPLY)=0.0
      BARK(INPLY)=0.
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      DELP=0.
      NDAM=2
      GO TO 65
   80 IF(MODEF.NE.2) GO TO 85
      TEMPK=PLYK(INPLY)
      PLYK(INPLY)=DELNS(NX,K)*TEMPK
      BARK(INPLY)=(1.-DELNS(NX,K))*TEMPK
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=2
      DELP=0.
      NDAM=4
      GO TO 65
   85 IF(MODEF.NE.3) GO TO 90
      TEMPK=PLYK(INPLY)
      PLYK(INPLY)=DELBR(NX,K)*TEMPK
      BARK(INPLY)=(1.-DELBR(NX,K))*TEMPK
      BARU(INPLY)=UN(INPLY)
```

```
      MDAMP(INPLY)=3
      DELP=0.
      NDAM=5
      GO TO 65
   90 IF(MODEF.NE.4) GO TO 140
      PLYK(INPLY)=0.0
      BARK(INPLY)=0.
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      DELP=0.
      NDAM=6
      GO TO 65
  140 IF(MODEF.NE.5) GO TO 110
      PLYK(INPLY)=0.
      BARK(INPLY)=0.
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      DELP=0.
      NDAM=6
      GO TO 65
  110 IF(MODEF.NE.6) GO TO 115
      PLYK(INPLY)=0.0
      BARK(INPLY)=0.0
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      NDAM=7
      DELP=0.
      GO TO 65
  115 IF(MODEF.NE.7) GO TO 120
      PLYK(INPLY)=0.
      BARK(INPLY)=0.
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      DELP=0.
      NDAM=8
      GO TO 65
  120 IF(MODEF.NE.8) GO TO 125
      PLYK(INPLY)=0.
      BARK(INPLY)=0.
      BARU(INPLY)=UN(INPLY)
      MDAMP(INPLY)=10
      NDAM=9
      DELP=0.
      GO TO 65
  125 IF(MODEF.NE.9) GO TO 65
      DELP=1000.
      MDAMI(INPLY)=1
      NDAM=10
      GO TO 65
  600 CONTINUE
C
C     PARTIAL BEARING FAILURE ANALYSIS
C
      IROUT=2
      NPL=NPLY(1)+NPLY(2)
      AJFLNS=1.0D10
      AJFLBR=1.0D10
      AJFLSO=1.0D10
      DO 550 I=1,NPL
      K=1
```

```
      IF(I.GT.NPLY(1)) K=2
      PL=-H(K)*(PLYK(I)*U(I)+BARK(I)*BARU(I))
      IF(PL.LT.0.0.AND.K.EQ.1) GO TO 550
      IF(PL.GT.0.0.AND.K.EQ.2) GO TO 550
      PL=DABS(PL)*AST*WTH*H(K)*NPLY(K)/1000.
      IJ=I
      IF(I.GT.NPLY(1)) IJ=IJ-NPLY(1)
      IP=IPLY(IJ,K)
      NT=1
      IF(BPSTS(K,IP,2,1).LT.0.0) NT=2
      F1=DABS(PSTC(NT,IP,K)/(BPSTS(K,IP,1,1)+PL*BPSTS(K,IP,2,1)))
      F2=DABS(PSTC(3,IP,K)/(BPSTS(K,IP,1,2)+PL*BPSTS(K,IP,2,2)))
      F3=DABS(PSTC(4,IP,K)/(BPSTS(K,IP,1,3)+PL*BPSTS(K,IP,2,3)))
      IF(MDAMP(I).EQ.5) F1=PALT(3,K)*F1
      IF(MDAMP(I).EQ.4) F2=PALT(2,K)*F2
      IF(MDAMP(I).EQ.2) F3=PALT(1,K)*F3
      IF(AJFLNS.GT.F1) NF1=I
      IF(AJFLNS.GT.F1) AJFLNS=F1
      IF(AJFLBR.GT.F2) NF2=I
      IF(AJFLBR.GT.F2) AJFLBR=F2
      IF(AJFLSO.GT.F3) NF3=I
      IF(AJFLSO.GT.F3) AJFLSO=F3
  550 CONTINUE
      IF(AJFLNS.GT.AJFLBR.OR.AJFLNS.GT.AJFLSO) GO TO 560
      INPLY=NF1
  551 NDAM=5
      IF(MDAMP(INPLY).EQ.2) GO TO 571
      IF(MDAMP(INPLY).EQ.4) GO TO 561
      IF(MDAMP(INPLY).EQ.5) NDAM=8
      MDAMP(INPLY)=NDAM
      PFAIL=AJFLNS
      GO TO 64
  560 IF(AJFLBR.GT.AJFLNS.OR.AJFLBR.GT.AJFLSO) GO TO 570
      INPLY=NF2
  561 NDAM=4
      IF(MDAMP(INPLY).EQ.5) GO TO 551
      IF(MDAMP(INPLY).EQ.2) GO TO 571
      IF(MDAMP(INPLY).EQ.4) NDAM=7
      MDAMP(INPLY)=NDAM
      PFAIL=AJFLBR
      GO TO 64
  570 IF(AJFLSO.GT.AJFLNS.OR.AJFLSO.GT.AJFLBR) GO TO 64
      INPLY=NF3
  571 NDAM=2
      IF(MDAMP(INPLY).EQ.5) GO TO 551
      IF(MDAMP(INPLY).EQ.4) GO TO 561
      IF(MDAMP(INPLY).EQ.2) NDAM=6
      MDAMP(INPLY)=NDAM
      PFAIL=AJFLSO
   64 CONTINUE
      K=1
      IF(INPLY.GT.NPLY(1)) K=2
      IPL=INPLY
      IF(IPL.GT.NPLY(1)) IPL=IPL-NPLY(1)
      IPLP=IPLY(IPL,K)
      ANGLE=ANG(IPLP,K)
      NODE=NPNM(IPL,K)
      IF(MDAMP(INPLY).GE.6) GO TO 107
      IF(MDAMP(INPLY).EQ.5) AR=DELNS(IPLP,K)
      IF(MDAMP(INPLY).EQ.4) AR=DELBR(IPLP,K)
```

```
      IF(MDAMP(INPLY).EQ.2) AR=DELSO(IPLP,K)
      TEMPK=PLYK(INPLY)
      PLYK(INPLY)=AR*TEMPK
      BARK(INPLY)=(1.-AR)*TEMPK
      BARU(INPLY)=U(INPLY)
      ITYP=IPLY(IPL,K)
      NTFL=0
      GO TO 103
  107 CONTINUE
      IF(K.EQ.1) NPLY(1)=NPLY(1)-1
      IF(K.EQ.2) NPLY(2)=NPLY(2)-1
      NP=INPLY
      IF(K.EQ.2) NP=INPLY-NPLY(1)
      N=NPLY(K)-NP+2
      ITYP=IPLY(NP,K)
      DO 101 I=1,N
      IPLY(NP+I-1,K)=IPLY(NP+I,K)
      NPNM(NP+I-1,K)=NPNM(NP+I,K)
  101 CONTINUE
      N=NPL-INPLY
      DO 102 I=1,N
      MDAMP(INPLY+I-1)=MDAMP(INPLY+I)
      PLYK(INPLY+I-1)=PLYK(INPLY+I)
      BARK(INPLY+I-1)=BARK(INPLY+I)
  102 BARU(INPLY+I-1)=BARU(INPLY+I)
      NTFL=1
      NULTF=NULTF-1
      IF(NULTF.EQ.0) JNT=0
      IF(NPLY(1).EQ.2.OR.NPLY(2).EQ.2) JNT=0
  103 CONTINUE
      RETURN
   65 CONTINUE
C
C     INCREMENT LOAD IF JOINT HAS NOT FAILED
C
      T1=0.
      T2=0.
      N1=NPLY(1)
      N2=NPLY(2)
      DO 135 I=1,N1
  135 T1=T1+PLYK(I)
      DO 126 I=1,N2
      N3=NPLY(1)+I
  126 T2=T2+PLYK(N3)
      IF(T1.EQ.0.0.OR.T2.EQ.0.0) GO TO 130
      RETURN
  130 JNT=0
      RETURN
      END
C
C
C
      SUBROUTINE PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT,
     *NP,NSDLS,ITT)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(100),PLYK(100)
      DIMENSION NPLY(2),NUMPLY(2),ANG(5,2),IPLY(100,2)
      DIMENSION BARK(100),BARU(100)
```

```
      DIMENSION NPNM(100,2)
      COMMON/COUNT/NPNM
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/PBB/PLYK,DARK,BARU
      COMMON/PRT/NDAM,INPLY,ITYP
C
C     PRINT VALUES AT END OF INCREMENT
C
      IF(ITT.EQ.1) WRITE(6,10)
   10 FORMAT(//,10X,'FAILURE MODE ABBREVIATIONS'                 ',/,/,
     *10X,'ND = NO ADDITIONAL DAMACE AT CURRENT JOINT LOAD',/,
     *10X,'DL = DELAMINATION                                ',/,
     *10X,'SO = SHEAR-OUT                                   ',/,
     *10X,'BR = BEARING                                  ',/,
     *10X,'NS = NET SECTION                              ',/,
     *10X,'SUD= ULTIMATE FAILURE AFTER SO AND DL        ',/,
     *10X,'SU = ULTIMATE FAILURE IN SO                  ',/,
     *10X,'BU = ULTIMATE FAILURE IN BR                  ',/,
     *10X,'NSU= ULTIMATE FAILURE IN NS                 ',/,
     *10X,'ULT= ULTIMATE FAILURE                    ',//,/,
     *4X,'INCREMENT NO',3X,'JOINT LOAD',5X,'NODE',8X,'PLY TYPE',
     *8X,'MODE',/)
      PL=P
      IF(IROUT.EQ.2) PL=PFAIL
      IF(NSDLS.EQ.2) PL=2.*PL
      IF(ITT.EQ.1) PFAILP=0.0D0
      IF(PFAILP.LT.PL) PFAILP=PL
      IF(JNT.EQ.0.AND.PFAILP.EQ.0.0D0) PFAILP=PL
      K=1
      IF(INPLY.GT.NPLY(1)) K=2
      N=IPLY(INPLY,K)
      IF(K.EQ.2) N=IPLY((INPLY-NPLY(1)),K)
      IF(IROUT.EQ.2) N=ITYP
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) ANGLE=ANG(N,K)
      IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) NODE=INPLY
      IF(NDAM.EQ.1) WRITE(6,20) ITT,PL
      IF(NDAM.EQ.2) WRITE(6,30) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.3) WRITE(6,40) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.4) WRITE(6,50) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.5) WRITE(6,60) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.6) WRITE(6,70) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.7) WRITE(6,80) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.8) WRITE(6,90) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.9) WRITE(6,100) ITT,PL,NODE,ANGLE
      IF(NDAM.EQ.10) WRITE(6,110) ITT,PL,NODE,ANGLE
   20 FORMAT(5X,I5,10X,D9.3,34X,'ND')
   30 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' SO')
   40 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' SUD')
   50 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' BR')
   60 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' NS')
   70 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' SU')
   80 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' BU')
   90 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' NSU')
  100 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' ULT')
  110 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,' DEGREE ',5X,' DL')
      IF(JNT.EQ.0) GO TO 220
      GO TO 250
  220 CONTINUE
      WRITE(6,240) PFAILP
  240 FORMAT(//,' THE PREDICTED JOINT FAILURE ',/,
```

```
     *' LOAD IS ',D14.7,'  LBS',//)
  250 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE LINV2F (A,N,IA,AINV,IDGT,WKAREA,IER)
C
      DOUBLE PRECISION   A(IA,N),AINV(IA,N),WKAREA(1),ZERO,ONE
      DATA               ONE/1.0D0/,ZERO/0.0D0/
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER=0
C                                  SET AINV TO THE N X N
C                                  IDENTITY MATRIX
      DO 10 I = 1,N
         DO 5 J = 1,N
            AINV(I,J) = ZERO
    5    CONTINUE
         AINV(I,I) = ONE
   10 CONTINUE
C                                  COMPUTE THE INVERSE OF A
      CALL LEQT2F (A,N,N,IA,AINV,IDGT,WKAREA,IER)
      IF (IER.EQ.0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HLINV2F)
 9005 RETURN
      END
C
C
      SUBROUTINE LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
      DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
      DOUBLE PRECISION   A,B,WKAREA,D1,D2,WA
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER=0
      JER=0
      J = N*N+1
      K = J+N
      MM = K+N
      KK = 0
      MM1 = MM-1
      JJ=1
      DO 5 L=1,N
         DO 5 I=1,N
            WKAREA(JJ)=A(I,L)
            JJ=JJ+1
    5 CONTINUE
C                                  DECOMPOSE A
      CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
     *             WA,IER)
      IF (IER.GT.128) GO TO 25
      IF (IDGT .EQ. 0 .OR. IER .NE. 0) KK = 1
      DO 15 I = 1,M
C                                  PERFORMS THE ELIMINATION PART OF
C                                  AX = B
         CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))
C                                  REFINEMENT OF SOLUTION TO AX = B
```

```
         IF (KK .NE. 0)
     *   CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDGT,WKAREA(J),WKAREA(MM),
     *                WKAREA(K),WKAREA(K),JER)
         DO 10 II=1,N
            B(II,I) = WKAREA(MM1+II)
   10    CONTINUE
         IF (JER.NE.0) GO TO 20
   15 CONTINUE
      GO TO 25
   20 IER = 131
   25 JJ=1
      DO 30 J = 1,N
         DO 30 I = 1,N
            A(I,J)=WKAREA(JJ)
            JJ=JJ+1
   30 CONTINUE
      IF (IER .EQ. 0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HLEQT2F)
 9005 RETURN
      END
C
C
C
      SUBROUTINE LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)
C
      DIMENSION          A(IA,1),LU(IA,1),IPVT(1),EQUIL(1)
      DOUBLE PRECISION   A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH,
     *                   RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.D0,1.D0,4.D0,
     *                   16.D0,.0625D0/
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZATION
      IER = 0
      RN = N
      WREL = ZERO
      D1 = ONE
      D2 = ZERO
      BIGA = ZERO
      DO 10 I=1,N
         BIG = ZERO
         DO 5 J=1,N
            P = A(I,J)
            LU(I,J) = P
            P = DABS(P)
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .GT. BIGA) BIGA = BIG
         IF (BIG .EQ. ZERO) GO TO 110
         EQUIL(I) = ONE/BIG
   10 CONTINUE
      DO 105 J=1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 40
C                                  COMPUTE U(I,J), I=1,...,J-1
         DO 35 I=1,JM1
            SUM = LU(I,J)
            IM1 = I-1
            IF (IDGT .EQ. 0) GO TO 25
C                                  WITH ACCURACY TEST
```

```
            AI = DABS(SUM)
            WI = ZERO
            IF (IM1 .LT. 1) GO TO 20
            DO 15 K=1,IM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   15       CONTINUE
            LU(I,J) = SUM
   20       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 35
C                                  WITHOUT ACCURACY
   25       IF (IM1 .LT. 1) GO TO 35
            DO 30 K=1,IM1
               SUM = SUM-LU(I,K)*LU(K,J)
   30       CONTINUE
            LU(I,J) = SUM
   35    CONTINUE
   40    P = ZERO
C                                  COMPUTE U(J,J) AND L(I,J), I=J+1,...,
         DO 70 I=J,N
            SUM = LU(I,J)
            IF (IDOT .EQ. 0) GO TO 55
C                                  WITH ACCURACY TEST
            AI = DABS(SUM)
            WI = ZERO
            IF (JM1 .LT. 1) GO TO 50
            DO 45 K=1,JM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   45       CONTINUE
            LU(I,J) = SUM
   50       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 65
C                                  WITHOUT ACCURACY TEST
   55       IF (JM1 .LT. 1) GO TO 65
            DO 60 K=1,JM1
               SUM = SUM-LU(I,K)*LU(K,J)
   60       CONTINUE
            LU(I,J) = SUM
   65       Q = EQUIL(I)*DABS(SUM)
            IF (P .GE. Q) GO TO 70
            P = Q
            IMAX = I
   70    CONTINUE
C                                  TEST FOR ALGORITHMIC SINGULARITY
         IF (RN+P .EQ. RN) GO TO 110
         IF (J .EQ. IMAX) GO TO 80
C                                  INTERCHANGE ROWS J AND IMAX
         D1 = -D1
         DO 75 K=1,N
            P = LU(IMAX,K)
            LU(IMAX,K) = LU(J,K)
```

```
            LU(J,K) = P
   75    CONTINUE
         EQUIL(IMAX) = EQUIL(J)
   80    IPVT(J) = IMAX
         D1 = D1*LU(J,J)
   85    IF (DABS(D1) .LE. ONE) GO TO 90
         D1 = D1*SIXTH
         D2 = D2+FOUR
         GO TO 85
   90    IF (DABS(D1) .GE. SIXTH) GO TO 95
         D1 = D1*SIXTN
         D2 = D2-FOUR
         GO TO 90
   95    CONTINUE
         JP1 = J+1
         IF (JP1 .GT. N) GO TO 105
C                                  DIVIDE BY PIVOT ELEMENT U(J,J)
         P = LU(J,J)
         DO 100 I=JP1,N
            LU(I,J) = LU(I,J)/P
  100    CONTINUE
  105 CONTINUE
C                                  PERFORM ACCURACY TEST
      IF (IDGT .EQ. 0) GO TO 9005
      P = 3*N+3
      WA = P*WREL
      IF (WA+10.D0**(-IDGT) .NE. WA) GO TO 9005
      IER = 34
      GO TO 9000
C                                  ALGORITHMIC SINGULARITY
  110 IER = 129
      D1 = ZERO
      D2 = ZERO
 9000 CONTINUE
C                                  PRINT ERROR
      CALL UERTST(IER,6HLUDATF)
 9005 RETURN
      END
C
C
      SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
      DIMENSION          A(IA,1),B(1),APVT(1),X(1)
      DOUBLE PRECISION   A,B,X,SUM,APVT
C                                  FIRST EXECUTABLE STATEMENT
C                                  SOLVE LY = B FOR Y
      DO 5 I=1,N
    5 X(I) = B(I)
      IW = 0
      DO 20 I=1,N
         IP = APVT(I)
         SUM = X(IP)
         X(IP) = X(I)
         IF (IW .EQ. 0) GO TO 15
         IM1 = I-1
         DO 10 J=IW,IM1
            SUM = SUM-A(I,J)*X(J)
   10    CONTINUE
         GO TO 20
   15    IF (SUM .NE. 0.D0) IW = I
```

```
   20 X(I) = SUM
C                                        SOLVE UX = Y FOR X
      DO 30 IB=1,N
         I = N+1-IB
         IP1 = I+1
         SUM = X(I)
         IF (IP1 .GT. N) GO TO 30
         DO 25 J=IP1,N
            SUM = SUM-A(I,J)*X(J)
   25    CONTINUE
   30 X(I) = SUM/A(I,I)
      RETURN
      END
C
C
      SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDOT,APVT,X,RES,DX,IER)
C
      DIMENSION          A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1)
      DIMENSION          APVT(1)
      DIMENSION          ACCXT(2)
      DOUBLE PRECISION   A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT
      DATA               ITMAX/75/,ZERO/0.D0/
C                                  FIRST EXECUTABLE STATEMENT
      IER=0
      XNORM = ZERO
      DO 10 I=1,N
         XNORM = DMAX1(XNORM,DABS(X(I)))
   10 CONTINUE
      IF (XNORM .NE. ZERO) GO TO 20
      IDOT = 50
      GO TO 9005
   20 DO 45 ITER=1,ITMAX
         DO 30 I=1,N
      ACCXT(1) = 0.0D0
      ACCXT(2) = 0.0D0
            CALL VXADD(B(I),ACCXT)
            DO 25 J=1,N
               CALL VXMUL(-A(I,J),X(J),ACCXT)
   25       CONTINUE
            CALL VXSTO(ACCXT,RES(I))
   30    CONTINUE
         CALL LUELMN (UL,IUL,N,RES,APVT,DX)
         DXNORM = ZERO
         XNORM = ZERO
         DO 35 I=1,N
            X(I) = X(I) + DX(I)
            DXNORM = DMAX1(DXNORM,DABS(DX(I)))
            XNORM = DMAX1(XNORM,DABS(X(I)))
   35    CONTINUE
         IF (ITER .NE. 1) GO TO 40
         IDOT = 50
         IF (DXNORM .NE. ZERO) IDOT = -DLOG10(DXNORM/XNORM)
   40    IF (XNORM+DXNORM .EQ. XNORM) GO TO 9005
   45 CONTINUE
C                                  ITERATION DID NOT CONVERGE
      IER = 129
 9000 CONTINUE
      CALL UERTST(IER,6HLUREFN)
 9005 RETURN
      END
```

```
C
C
C
      SUBROUTINE UERTST (IER,NAME)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IER
      INTEGER            NAME(1)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(6),
     *                   NAMSET(6),NAMUPK(6),NIN,NMTB
      DATA               NAMSET/1HU,1HE,1HR,1HS,1HE,1HT/
      DATA               NAMEQ/6*1H /
      DATA               LEVEL/4/,IEQDF/0/,IEQ/1H=/
C                                  UNPACK NAME INTO NAMUPK
C                                  FIRST EXECUTABLE STATEMENT
      CALL USPKD (NAME,6,NAMUPK,NMTB)
C                                  GET OUTPUT UNIT NUMBER
      CALL UGETIO(1,NIN,IOUNIT)
C                                  CHECK IER
      IF (IER.GT.999) GO TO 25
      IF (IER.LT.-32) GO TO 55
      IF (IER.LE.128) GO TO 5
      IF (LEVEL.LT.1) GO TO 30
C                                  PRINT TERMINAL MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
      GO TO 30
    5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
C                                  PRINT WARNING WITH FIX MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
      GO TO 30
   10 IF (IER.LE.32) GO TO 15
C                                  PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
      GO TO 30
   15 CONTINUE
C                                  CHECK FOR UERSET CALL
      DO 20 I=1,6
         IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
   20 CONTINUE
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
   25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
C                                  PRINT NON-DEFINED MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK
   30 IEQDF = 0
      RETURN
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3,
```

```
     1        20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,
     1        20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,
     1        20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
C
C                                  SAVE P FOR P = R CASE
C                                    P IS THE PAGE NAMUPK
C                                    R IS THE ROUTINE NAMUPK
   55 IEQDF = 1
      DO 60 I=1,6
   60 NAMEQ(I) = NAMUPK(I)
   65 RETURN
      END
C
C
C
      SUBROUTINE UGETIO(IOPT,NIN,NOUT)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IOPT,NIN,NOUT
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            NIND,NOUTD
      DATA               NIND/5/,NOUTD/6/
C                                  FIRST EXECUTABLE STATEMENT
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
      NOUT = NOUTD
      GO TO 9005
    5 NIND = NIN
      GO TO 9005
   10 NOUTD = NOUT
 9005 RETURN
      END
C
C
C
      SUBROUTINE VXADD(A,ACC)
C
C                                  SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   A,ACC(2)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   X,Y,Z,ZZ
C                                  FIRST EXECUTABLE STATEMENT
      X = ACC(1)
      Y = A
      IF (DABS(ACC(1)).GE.DABS(A)) GO TO 1
      X = A
      Y = ACC(1)
C                                  COMPUTE Z+ZZ = ACC(1)+A EXACTLY
    1 Z = X+Y
      ZZ = (X-Z)+Y
C                                  COMPUTE ZZ+ACC(2) USING DOUBLE
C                                    PRECISION ARITHMETIC
      ZZ = ZZ+ACC(2)
C                                  COMPUTE ACC(1)+ACC(2) = Z+ZZ EXACTLY
      ACC(1) = Z+ZZ
      ACC(2) = (Z-ACC(1))+ZZ
      RETURN
```

```
      END
C
C
C
      SUBROUTINE VXMUL (A,B,ACC)
C                                  SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   A,B,ACC(2)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   X,HA,TA,HB,TB
      INTEGER            IX(2),I
      LOGICAL*1          LX(8),LI(4)
      EQUIVALENCE        (X,LX(1),IX(1)),(I,LI(1))
      DATA               I/0/
C                                  SPLIT A = HA+TA
C                                        B = HB+TB
C                                  FIRST EXECUTABLE STATEMENT
      X = A
      LI(4) = LX(5)
      IX(2) = 0
      I = (I/16)*16
      LX(5) = LI(4)
      HA=X
      TA=A-HA
      X = B
      LI(4) = LX(5)
      IX(2) = 0
      I = (I/16)*16
      LX(5) = LI(4)
      HB = X
      TB = B-HB
C                                  COMPUTE HA*HB,HA*TB,TA*HB, AND TA*TB
C                                    AND CALL VXADD TO ACCUMULATE THE
C                                    SUM
      X = TA*TB
      CALL VXADD(X,ACC)
      X = HA*TB
      CALL VXADD(X,ACC)
      X = TA*HB
      CALL VXADD(X,ACC)
      X = HA*HB
      CALL VXADD(X,ACC)
      RETURN
      END
C
C
C
      SUBROUTINE VXSTO (ACC,D)
C                                  SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   ACC(2),D
C                                  FIRST EXECUTABLE STATEMENT
      D = ACC(1)+ACC(2)
      RETURN
      END
C
C
C
      SUBROUTINE ZRPOLY (A,NDEG,Z,IER)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NDEG,IER
      DOUBLE PRECISION   A(1),Z(1)
```

```
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,JJ,I,NM1,ICNT,N2,L,NZ,NPI
      REAL               ETA,RMRE,RINFP,REPSP,RADIX,RLO,XX,YY,SINR,
     1                   COSR,RMAX,RMIN,X,SC,XM,FF,DX,DF,BND,XXX,ARE
      REAL               PT(101)
      DOUBLE PRECISION   TEMP(101),P(101),QP(101),RK(101),QK(101),
     1                   SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   T,AA,BB,CC,FACTOR,REPSR1,ZERO,ONE,FN
      LOGICAL            ZEROK
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
C                                 THE FOLLOWING STATEMENTS SET MACHINE
C                                   CONSTANTS USED IN VARIOUS PARTS OF
C                                   THE PROGRAM. THE MEANING OF THE
C                                   FOUR CONSTANTS ARE - REPSR1 THE
C                                   MAXIMUM RELATIVE REPRESENTATION
C                                   ERROR WHICH CAN BE DESCRIBED AS
C                                   THE SMALLEST POSITIVE FLOATING
C                                   POINT NUMBER SUCH THAT 1.+REPSR1 IS
C                                   GREATER THAN 1
C                                 RINFP THE LARGEST FLOATING-POINT
C                                   NUMBER
C                                 REPSP THE SMALLEST POSITIVE
C                                   FLOATING-POINT NUMBER IF THE
C                                   EXPONENT RANGE DIFFERS IN SINGLE
C                                   AND DOUBLE PRECISION THEN REPSP
C                                   AND RINFP SHOULD INDICATE THE
C                                   SMALLER RANGE
C                                 RADIX THE BASE OF THE FLOATING-POINT
C                                   NUMBER SYSTEM USED
      DATA               RINFP/Z7FFFFFFF/
      DATA               REPSP/Z00100000/
      DATA               RADIX/16.0/
      DATA               REPSR1/Z3410000000000000/
      DATA               ZERO/0.0D0/,ONE/1.0D0/
C                                 ZRPOLY USES SINGLE PRECISION
C                                   CALCULATIONS FOR SCALING, BOUNDS
C                                   AND ERROR CALCULATIONS.
C                                 FIRST EXECUTABLE STATEMENT
      IER = 0
      IF (NDEG .GT. 100 .OR. NDEG .LT. 1) GO TO 165
      ETA = REPSR1
      ARE = ETA
      RMRE = ETA
      RLO = REPSP/ETA
C                                 INITIALIZATION OF CONSTANTS FOR
C                                   SHIFT ROTATION
      XX = .7071068
      YY = -XX
      SINR = .9975641
      COSR = -.06975647
      N = NDEG
      NN = N+1
C                                 ALGORITHM FAILS IF THE LEADING
C                                   COEFFICIENT IS ZERO.
      IF (A(1).NE.ZERO) GO TO 5
      IER = 130
      GO TO 9000
```

```
C                                      REMOVE THE ZEROS AT THE ORIGIN IF
C                                        ANY
    5 IF (A(NN).NE.ZERO) GO TO 10
      J = NDEG-N+1
      JJ = J+NDEG
      Z(J) = ZERO
      Z(JJ) = ZERO
      NN = NN-1
      N = N-1
      IF (NN.EQ.1) GO TO 9005
      GO TO 5
C                                      MAKE A COPY OF THE COEFFICIENTS
   10 DO 15 I=1,NN
         P(I) = A(I)
   15 CONTINUE
C                                      START THE ALGORITHM FOR ONE ZERO
   20 IF (N.GT.2) GO TO 30
      IF (N.LT.1) GO TO 9005
C                                      CALCULATE THE FINAL ZERO OR PAIR OF
C                                        ZEROS
      IF (N.EQ.2) GO TO 25
      Z(NDEG) = -P(2)/P(1)
      Z(NDEG+NDEG) = ZERO
      GO TO 145
   25 CALL ZRPQLI (P(1),P(2),P(3),Z(NDEG-1),Z(NDEG+NDEG-1),Z(NDEG),
     1   Z(NDEG+NDEG))
      GO TO 145
C                                      FIND LARGEST AND SMALLEST MODULI OF
C                                        COEFFICIENTS.
   30 RMAX = 0.
      RMIN = RINFP
      DO 35 I=1,NN
         X = ABS(SNGL(P(I)))
         IF (X.GT.RMAX) RMAX = X
         IF (X.NE.0..AND.X.LT.RMIN) RMIN = X
   35 CONTINUE
C                                      SCALE IF THERE ARE LARGE OR VERY
C                                        SMALL COEFFICIENTS COMPUTES A
C                                        SCALE FACTOR TO MULTIPLY THE
C                                        COEFFICIENTS OF THE POLYNOMIAL.
C                                        THE SCALING IS DONE TO AVOID
C                                        OVERFLOW AND TO AVOID UNDETECTED
C                                        UNDERFLOW INTERFERING WITH THE
C                                        CONVERGENCE CRITERION.
C                                      THE FACTOR IS A POWER OF THE BASE
      SC = RLO/RMIN
      IF (SC.GT.1.0) GO TO 40
      IF (RMAX.LT.10.) GO TO 55
      IF (SC.EQ.0.) SC = REPSP*RADIX*RADIX
      GO TO 45
   40 IF (RINFP/SC.LT.RMAX) GO TO 55
   45 L = ALOG(SC)/ALOG(RADIX)+.5
      IF (L .EQ. 0) GO TO 55
      FACTOR = DBLE(RADIX)**L
      DO 50 I=1,NN
   50 P(I) = FACTOR*P(I)
C                                      COMPUTE LOWER BOUND ON MODULI OF
C                                        ZEROS.
   55 DO 60 I=1,NN
   60 PT(I) = ABS(SNGL(P(I)))
```

```
      PT(NN) = -PT(NN)
C                                   COMPUTE UPPER ESTIMATE OF BOUND
      X = EXP((ALOG(-PT(NN))-ALOG(PT(1)))/N)
      IF (PT(N).EQ.0.) GO TO 65
C                                   IF NEWTON STEP AT THE ORIGIN IS
C                                     BETTER, USE IT.
      XM = -PT(NN)/PT(N)
      IF (XM.LT.X) X = XM
C                                   CHOP THE INTERVAL (0,X) UNTIL FF.LE.0
   65 XM = X*.1
      FF = PT(1)
      DO 70 I=2,NN
   70 FF = FF*XM+PT(I)
      IF (FF.LE.0.) GO TO 75
      X = XM
      GO TO 65
   75 DX = X
C                                   DO NEWTON ITERATION UNTIL X
C                                     CONVERGES TO TWO DECIMAL PLACES
   80 IF (ABS(DX/X).LE..005) GO TO 90
      FF = PT(1)
      DF = FF
      DO 85 I=2,N
         FF = FF*X+PT(I)
         DF = DF*X+FF
   85 CONTINUE
      FF = FF*X+PT(NN)
      DX = FF/DF
      X = X-DX
      GO TO 80
   90 BND = X
C                                   COMPUTE THE DERIVATIVE AS THE INTIAL
C                                     K POLYNOMIAL AND DO 5 STEPS WITH
C                                     NO SHIFT
      NM1 = N-1
      FN = ONE/N
      DO 95 I=2,N
   95 RK(I) = (NN-I)*P(I)*FN
      RK(1) = P(1)
      AA = P(NN)
      BB = P(N)
      ZEROK = RK(N).EQ.ZERO
      DO 115 JJ=1,5
         CC = RK(N)
         IF (ZEROK) GO TO 105
C                                   USE SCALED FORM OF RECURRENCE IF
C                                     VALUE OF K AT 0 IS NONZERO
         T = -AA/CC
         DO 100 I=1,NM1
            J = NN-I
            RK(J) = T*RK(J-1)+P(J)
  100    CONTINUE
         RK(1) = P(1)
         ZEROK = DABS(RK(N)).LE.DABS(BB)*ETA*10.
         GO TO 115
C                                   USE UNSCALED FORM OF RECURRENCE
  105    DO 110 I=1,NM1
            J = NN-I
            RK(J) = RK(J-1)
  110    CONTINUE
```

```
          RK(1) = ZERO
          ZEROK = RK(N).EQ.ZERO
  115 CONTINUE
C                                 SAVE K FOR RESTARTS WITH NEW SHIFTS
      DO 120 I=1,N
  120 TEMP(I) = RK(I)
C                                 LOOP TO SELECT THE QUADRATIC
C                                   CORRESPONDING TO EACH NEW SHIFT
      DO 140 ICNT=1,20
C                                 QUADRATIC CORRESPONDS TO A DOUBLE
C                                   SHIFT TO A NON-REAL POINT AND ITS
C                                   COMPLEX CONJUGATE. THE POINT HAS
C                                   MODULUS BND AND AMPLITUDE ROTATED
C                                   BY 94 DEGREES FROM THE PREVIOUS
C                                   SHIFT
         XXX = COSR*XX-SINR*YY
         YY = SINR*XX+COSR*YY
         XX = XXX
         SR = BND*XX
         SI = BND*YY
         U = -SR-SR
         V = BND*BND
C                                 SECOND STAGE CALCULATION, FIXED
C                                   QUADRATIC
         CALL ZRPQLB (20*ICNT,NZ)
         IF (NZ.EQ.0) GO TO 130
C                                 THE SECOND STAGE JUMPS DIRECTLY TO
C                                   ONE OF THE THIRD STAGE ITERATIONS
C                                   AND RETURNS HERE IF SUCCESSFUL.
C                                 DEFLATE THE POLYNOMIAL, STORE THE
C                                   ZERO OR ZEROS AND RETURN TO THE
C                                   MAIN ALGORITHM.
         J = NDEG-N+1
         JJ = J+NDEG
         Z(J) = SZR
         Z(JJ) = SZI
         NN = NN-NZ
         N = NN-1
         DO 125 I=1,NN
  125    P(I) = QP(I)
         IF (NZ.EQ.1) GO TO 20
         Z(J+1) = RLZR
         Z(JJ+1) = RLZI
         GO TO 20
C                                 IF THE ITERATION IS UNSUCCESSFUL
C                                   ANOTHER QUADRATIC IS CHOSEN AFTER
C                                   RESTORING K
  130    DO 135 I=1,N
  135    RK(I) = TEMP(I)
  140 CONTINUE
C                                 RETURN WITH FAILURE IF NO
C                                   CONVERGENCE WITH 20 SHIFTS
      IER = 131
C                                 CONVERT ZEROS (Z) IN COMPLEX FORM
  145 DO 150 I=1,NDEG
         NPI= NDEG+I
         P(I) = Z(NPI)
  150 CONTINUE
      N2 = NDEG+NDEG
      J = NDEG
```

```
      DO 155 I=1,NDEG
         Z(N2-1) = Z(J)
         Z(N2) = P(J)
         N2 = N2-2
         J = J-1
  155 CONTINUE
      IF (IER .EQ. 0) GO TO 9005
C                                SET UNFOUND ROOTS TO MACHINE INFINITY
      N2 = 2*(NDEG-NN)+3
      DO 160 I=1,N
         Z(N2) = RINFP
         Z(N2+1) = RINFP
         N2 = N2+2
  160 CONTINUE
      GO TO 9000
  165 IER = 129
 9000 CONTINUE
      CALL UERTST (IER,6HZRPOLY)
 9005 RETURN
      END
C
C
C
      SUBROUTINE ZRPQLB (L2,NZ)
C                                SPECIFICATIONS FOR ARGUMENTS
      INTEGER            L2,NZ
C                                SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,ITYPE,I,IFLAG
      REAL               ARE,BETAS,BETAV,ETA,OSS,OTS,OTV,OVV,RMRE,SS,
     1                   TS,TSS,TV,TVV,VV
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   SVU,SVV,UI,VI,S,ZERO
      LOGICAL            VPASS,SPASS,VTRY,STRY
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                FIRST EXECUTABLE STATEMENT
      NZ = 0
C                                COMPUTES UP TO L2 FIXED SHIFT
C                                  K-POLYNOMIALS, TESTING FOR
C                                  CONVERGENCE IN THE LINEAR OR
C                                  QUADRATIC CASE. INITIATES ONE OF
C                                  THE VARIABLE SHIFT ITERATIONS AND
C                                  RETURNS WITH THE NUMBER OF ZEROS
C                                  FOUND.
C                                L2 - LIMIT OF FIXED SHIFT STEPS
C                                NZ -NUMBER OF ZEROS FOUND
      BETAV = .25
      BETAS = .25
      OSS = SR
      OVV = V
C                                EVALUATE POLYNOMIAL BY SYNTHETIC
C                                  DIVISION
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      CALL ZRPQLE (ITYPE)
      DO 40 J=1,L2
C                                CALCULATE NEXT K POLYNOMIAL AND
C                                  ESTIMATE V
```

```
      CALL ZRPQLF (ITYPE)
      CALL ZRPQLE (ITYPE)
      CALL ZRPQLG (ITYPE,UI,VI)
      VV = VI
C                                  ESTIMATE S
      SS = 0.
      IF (RK(N).NE.ZERO) SS = -P(NN)/RK(N)
      TV = 1.
      TS = 1.
      IF (J.EQ.1.OR.ITYPE.EQ.3) GO TO 35
C                                  COMPUTE RELATIVE MEASURES OF
C                                    CONVERGENCE OF S AND V SEQUENCES
      IF (VV.NE.0.) TV = ABS((VV-OVV)/VV)
      IF (SS.NE.0.) TS = ABS((SS-OSS)/SS)
C                                  IF DECREASING, MULTIPLY TWO MOST
C                                    RECENT CONVERGENCE MEASURES
      TVV = 1.
      IF (TV.LT.OTV) TVV = TV*OTV
      TSS = 1.
      IF (TS.LT.OTS) TSS = TS*OTS
C                                  COMPARE WITH CONVERGENCE CRITERIA
      VPASS = TVV.LT.BETAV
      SPASS = TSS.LT.BETAS
      IF (.NOT.(SPASS.OR.VPASS)) GO TO 35
C                                  AT LEAST ONE SEQUENCE HAS PASSED THE
C                                    CONVERGENCE TEST. STORE VARIABLES
C                                    BEFORE ITERATING
      SVU = U
      SVV = V
      DO 5 I=1,N
    5 SVK(I) = RK(I)
      S = SS
C                                  CHOOSE ITERATION ACCORDING TO THE
C                                    FASTEST CONVERGING SEQUENCE
      VTRY = .FALSE.
      STRY = .FALSE.
      IF (SPASS.AND.((.NOT.VPASS).OR.TSS.LT.TVV)) GO TO 20
   10 CALL ZRPQLC (UI,VI,NZ)
      IF (NZ.GT.0) RETURN
C                                  QUADRATIC ITERATION HAS FAILED. FLAG
C                                    THAT IT HAS BEEN TRIED AND
C                                    DECREASE THE CONVERGENCE
C                                    CRITERION.
      VTRY = .TRUE.
      BETAV = BETAV*.25
C                                  TRY LINEAR ITERATION IF IT HAS NOT
C                                    BEEN TRIED AND THE S SEQUENCE IS
C                                    CONVERGING
      IF (STRY.OR.(.NOT.SPASS)) GO TO 25
      DO 15 I=1,N
   15 RK(I) = SVK(I)
   20 CALL ZRPQLD (S,NZ,IFLAG)
      IF (NZ.GT.0) RETURN
C                                  LINEAR ITERATION HAS FAILED. FLAG
C                                    THAT IT HAS BEEN TRIED AND
C                                    DECREASE THE CONVERGENCE CRITERION
      STRY = .TRUE.
      BETAS = BETAS*.25
      IF (IFLAG.EQ.0) GO TO 25
C                                  IF LINEAR ITERATION SIGNALS AN
```

```
C                                       ALMOST DOUBLE REAL ZERO ATTEM.
C                                       QUADRATIC INTERATION
            UI = -(S+S)
            VI = S*S
            GO TO 10
C                                     RESTORE VARIABLES
   25       U = SVU
            V = SVV
            DO 30 I=1,N
   30       RK(I) = SVK(I)
C                                     TRY QUADRATIC ITERATION IF IT HAS
C                                       NOT BEEN TRIED AND THE V SEQUENCE
C                                       IS CONVERGING
            IF (VPASS.AND.(.NOT.VTRY)) GO TO 10
C                                     RECOMPUTE QP AND SCALAR VALUES TO
C                                       CONTINUE THE SECOND STAGE
            CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
            CALL ZRPQLE (ITYPE)
   35       OVV = VV
            OSS = SS
            OTV = TV
            OTS = TS
   40 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLC (UU,VV,NZ)
C                                     SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NZ
      DOUBLE PRECISION   UU,VV
C                                     SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,I,ITYPE
      REAL               ARE,EE,ETA,OMP,RELSTP,RMP,RMRE,T,ZM
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   UI,VI,ZERO,PT01,ONE
      LOGICAL            TRIED
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO,PT01,ONE/0.0D0,0.01D0,1.0D0/
C                                     FIRST EXECUTABLE STATEMENT
      NZ = 0
C                                     VARIABLE-SHIFT K-POLYNOMIAL
C                                       ITERATION FOR A QUADRATIC FACTOR
C                                       CONVERGES ONLY IF THE ZEROS ARE
C                                       EQUIMODULAR OR NEARLY SO
C                                     UU,VV - COEFFICIENTS OF STARTING
C                                       QUADRATIC
C                                     NZ - NUMBER OF ZERO FOUND
      TRIED = .FALSE.
      U = UU
      V = VV
      J = 0
C                                     MAIN LOOP
    5 CALL ZRPQLI (ONE,U,V,SZR,SZI,RLZR,RLZI)
C                                     RETURN IF ROOTS OF THE QUADRATIC ARE
C                                       REAL AND NOT CLOSE TO MULTIPLE OR
```

```
C                                      NEARLY EQUAL AND OF OPPOSITE SIGN
      IF ( DABS(DABS(SZR)-DABS(RLZR)).GT.PT01*DABS(RLZR)) RETURN
C                                    EVALUATE POLYNOMIAL BY QUADRATIC
C                                      SYNTHETIC DIVISION
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      RMP = DABS(RA-SZR*RB)+DABS(SZI*RB)
C                                    COMPUTE A RIGOROUS BOUND ON THE
C                                      ROUNDING ERROR IN EVALUTING P
      ZM = SQRT(ABS(SNGL(V)))
      EE = 2.*ABS(SNGL(QP(1)))
      T = -SZR*RB
      DO 10 I=2,N
   10 EE = EE*ZM+ABS(SNGL(QP(I)))
      EE = EE*ZM+ABS(SNGL(RA)+T)
      EE = (5.*RMRE+4.*ARE)*EE-(5.*RMRE+2.*ARE)*(ABS(SNGL(RA)+T)+
     1     ABS(SNGL(RB))*ZM)+2.*ARE*ABS(T)
C                                    ITERATION HAS CONVERGED SUFFICIENTLY
C                                      IF THE POLYNOMIAL VALUE IS LESS
C                                      THAN 20 TIMES THIS BOUND
      IF (RMP.GT.20.*EE) GO TO 15
      NZ = 2
      RETURN
   15 J = J+1
C                                    STOP ITERATION AFTER 20 STEPS
      IF (J.GT.20) RETURN
      IF (J.LT.2) GO TO 25
      IF (RELSTP.GT..01.OR.RMP.LT.OMP.OR.TRIED) GO TO 25
C                                    A CLUSTER APPEARS TO BE STALLING THE
C                                      CONVERGENCE. FIVE FIXED SHIFT
C                                      STEPS ARE TAKEN WITH A U,V CLOSE
C                                      TO THE CLUSTER
      IF (RELSTP.LT.ETA) RELSTP = ETA
      RELSTP = SQRT(RELSTP)
      U = U-U*RELSTP
      V = V+V*RELSTP
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      DO 20 I=1,5
         CALL ZRPQLE (ITYPE)
         CALL ZRPQLF (ITYPE)
   20 CONTINUE
      TRIED = .TRUE.
      J = 0
   25 OMP = RMP
C                                    CALCULATE NEXT K POLYNOMIAL AND NEW
C                                      U AND V
      CALL ZRPQLE (ITYPE)
      CALL ZRPQLF (ITYPE)
      CALL ZRPQLE (ITYPE)
      CALL ZRPQLG (ITYPE,UI,VI)
C                                    IF VI IS ZERO THE ITERATION IS NOT
C                                      CONVERGING
      IF (VI.EQ.ZERO) RETURN
      RELSTP = DABS((VI-V)/VI)
      U = UI
      V = VI
      GO TO 5
      END
C
C
C
```

```
      SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NZ,IFLAG
      DOUBLE PRECISION   SSS
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,I
      REAL               ARE,EE,ETA,OMP,RMP,RMS,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   PV,RKV,T,S,ZERO,PT001
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/,PT001/0.001D0/
C                                  VARIABLE-SHIFT H POLYNOMIAL
C                                    ITERATION FOR A REAL ZERO SSS -
C                                    STARTING ITERATE
C                                  NZ - NUMBER OF ZERO FOUND
C                                  IFLAG - FLAG TO INDICATE A PAIR OF
C                                    ZEROS NEAR REAL AXIS
C                                  FIRST EXECUTABLE STATEMENT
      NZ = 0
      S = SSS
      IFLAG = 0
      J = 0
C                                  MAIN LOOP
    5 PV = P(1)
C                                  EVALUATE P AT S
      QP(1) = PV
      DO 10 I=2,NN
         PV = PV*S+P(I)
         QP(I) = PV
   10 CONTINUE
      RMP = DABS(PV)
C                                  COMPUTE A RIGOROUS BOUND ON THE
C                                    ERROR IN EVALUATING P
      RMS = DABS(S)
      EE = (RMRE/(ARE+RMRE))*ABS(SNGL(QP(1)))
      DO 15 I=2,NN
   15 EE = EE*RMS+ABS(SNGL(QP(I)))
C                                  ITERATION HAS CONVERGED SUFFICIENTLY
C                                    IF THE POLYNOMIAL VALUE IS LESS
C                                    THAN 20 TIMES THIS BOUND
      IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20
      NZ = 1
      SZR = S
      SZI = ZERO
      RETURN
   20 J = J+1
C                                  STOP ITERATION AFTER 10 STEPS
      IF (J.GT.10) RETURN
      IF (J.LT.2) GO TO 25
      IF (DABS(T).GT.PT001*DABS(S-T).OR.RMP.LE.OMP) GO TO 25
C                                  A CLUSTER OF ZEROS NEAR THE REAL
C                                    AXIS HAS BEEN ENCOUNTERED RETURN
C                                    WITH IFLAG SET TO INITIATE A
C                                    QUADRATIC ITERATION
      IFLAG = 1
      SSS = S
      RETURN
```

```
C                                       RETURN IF THE POLYNOMIAL VALUE HAS
C                                         INCREASED SIGNIFICANTLY
   25 OMP = RMP
C                                       COMPUTE T, THE NEXT POLYNOMIAL, AND
C                                         THE NEW ITERATE
      RKV = RK(1)
      QK(1) = RKV
      DO 30 I=2,N
         RKV = RKV*S+RK(I)
         QK(I) = RKV
   30 CONTINUE
      IF (DABS(RKV).LE.DABS(RK(N))*10.*ETA) GO TO 40
C                                       USE THE SCALED FORM OF THE
C                                         RECURRENCE IF THE VALUE OF K AT S
C                                         IS NONZERO
      T = -PV/RKV
      RK(1) = QP(1)
      DO 35 I=2,N
   35 RK(I) = T*QK(I-1)+QP(I)
      GO TO 50
C                                       USE UNSCALED FORM
   40 RK(1) = ZERO
      DO 45 I=2,N
   45 RK(I) = QK(I-1)
   50 RKV = RK(1)
      DO 55 I=2,N
   55 RKV = RKV*S+RK(I)
      T = ZERO
      IF (DABS(RKV).GT.DABS(RK(N))*10.*ETA) T = -PV/RKV
      S = S+T
      GO TO 5
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLE
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLE (ITYPE)
C                                       SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
C                                       SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
C                                       THIS ROUTINE CALCULATES SCALAR
C                                         QUANTITIES USED TO COMPUTE THE
C                                         NEXT K POLYNOMIAL AND NEW
C                                         ESTIMATES OF THE QUADRATIC
C                                         COEFFICIENTS
C                                       ITYPE - INTEGER VARIABLE SET HERE
```

```
C                                         INDICATING HOW THE CALCULATIONS
C                                           ARE NORMALIZED TO AVOID OVERFLOW
C                                       SYNTHETIC DIVISION OF K BY THE
C                                           QUADRATIC 1,U,V
C                                       FIRST EXECUTABLE STATEMENT
      CALL ZRPQLH (N,U,V,RK,QK,C,D)
      IF (DABS(C).GT.DABS(RK(N))*100.*ETA) GO TO 5
      IF (DABS(D).GT.DABS(RK(N-1))*100.*ETA) GO TO 5
      ITYPE = 3
C                                       TYPE=3 INDICATES THE QUADRATIC IS
C                                           ALMOST A FACTOR OF K
      RETURN
    5 IF (DABS(D).LT.DABS(C)) GO TO 10
      ITYPE = 2
C                                       TYPE=2 INDICATES THAT ALL FORMULAS
C                                           ARE DIVIDED BY D
      E = RA/D
      F = C/D
      G = U*RB
      H = V*RB
      A3 = (RA+G)*E+H*(RB/D)
      A1 = RB*F-RA
      A7 = (F+U)*RA+H
      RETURN
   10 ITYPE = 1
C                                       TYPE=1 INDICATES THAT ALL FORMULAS
C                                           ARE DIVIDED BY C
      E = RA/C
      F = D/C
      G = U*E
      H = V*RB
      A3 = RA*E+(H/C+G)*RB
      A1 = RB-RA*(D/C)
      A7 = RA+G*D+H*F
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLF (ITYPE)
C                                       SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
C                                       SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,I
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,TEMP,ZERO
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                       COMPUTES THE NEXT K POLYNOMIALS
C                                           USING SCALARS COMPUTED IN ZRPQLE
C                                       FIRST EXECUTABLE STATEMENT
      IF (ITYPE.EQ.3) GO TO 20
      TEMP = RA
      IF (ITYPE.EQ.1) TEMP = RB
      IF (DABS(A1).GT.DABS(TEMP)*ETA*10.) GO TO 10
C                                       IF A1 IS NEARLY ZERO THEN USE A
C                                           SPECIAL FORM OF THE RECURRENCE
```

```
      RK(1) = ZERO
      RK(2) = -A7*QP(1)
      DO 5 I=3,N
    5 RK(I) = A3*QK(I-2)-A7*QP(I-1)
      RETURN
C                                  USE SCALED FORM OF THE RECURRENCE
   10 A7 = A7/A1
      A3 = A3/A1
      RK(1) = QP(1)
      RK(2) = QP(2)-A7*QP(1)
      DO 15 I=3,N
   15 RK(I) = A3*QK(I-2)-A7*QP(I-1)+QP(I)
      RETURN
C                                  USE UNSCALED FORM OF THE RECURRENCE
C                                    IF TYPE IS 3
   20 RK(1) = ZERO
      RK(2) = ZERO
      DO 25 I=3,N
   25 RK(I) = QK(I-2)
      RETURN
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLG
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLG (ITYPE,UU,VV)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
      DOUBLE PRECISION   UU,VV
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   A4,A5,B1,B2,C1,C2,C3,C4,TEMP,ZERO
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                  COMPUTE NEW ESTIMATES OF THE
C                                    QUADRATIC COEFFICIENTS USING THE
C                                    SCALARS COMPUTED IN ZRPQLE
C                                  USE FORMULAS APPROPRIATE TO SETTING
C                                    OF TYPE.
C                                  FIRST EXECUTABLE STATEMENT
      IF (ITYPE.EQ.3) GO TO 15
      IF (ITYPE.EQ.2) GO TO 5
      A4 = RA+U*RB+H*F
      A5 = C+(U+V*F)*D
      GO TO 10
    5 A4 = (RA+G)*F+H
      A5 = (F+U)*C+V*D
C                                  EVALUATE NEW QUADRATIC COEFFICIENTS.
```

```
C
   10 B1 = -RK(N)/P(NN)
      B2 = -(RK(N-1)+B1*P(N))/P(NN)
      C1 = V*B2*A1
      C2 = B1*A7
      C3 = B1*B1*A3
      C4 = C1-C2-C3
      TEMP = A5+B1*A4-C4
      IF (TEMP.EQ.ZERO) GO TO 15
      UU = U-(U*(C3+C2)+V*(B1*A1+B2*A7))/TEMP
      VV = V*(1+C4/TEMP)
      RETURN
C                                 IF TYPE=3 THE QUADRATIC IS ZEROED
   15 UU = ZERO
      VV = ZERO
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLH (NN,U,V,P,Q,RA,RB)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NN
      DOUBLE PRECISION   P(NN),Q(NN),U,V,RA,RB
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I
      DOUBLE PRECISION   C
C                                 DIVIDES P BY THE QUADRATIC 1,U,V
C                                   PLACING THE QUOTIENT IN Q AND THE
C                                   REMAINDER IN A,B
C                                 FIRST EXECUTABLE STATEMENT
      RB = P(1)
      Q(1) = RB
      RA = P(2)-U*RB
      Q(2) = RA
      DO 5 I=3,NN
         C = P(I)-U*RA-V*RB
         Q(I) = C
         RB = RA
         RA = C
    5 CONTINUE
      RETURN
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLI
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLI (RA,B1,C,SR,SI,RLR,RLI)
C                                 SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   RA,B1,C,SR,SI,RLR,RLI
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   RB,D,E,ZERO,ONE,TWO
      DATA               ZERO,ONE,TWO/0.0D0,1.0D0,2.0D0/
```

```
C                                    CALCULATE THE ZEROS OF THE QUADRATIC
C                                      A*Z**2 + B1*Z + C. THE QUADRATIC
C                                      FORMULA, MODIFIED TO AVOID
C                                      OVERFLOW, IS USED TO FIND THE
C                                      LARGER ZERO IF THE ZEROS ARE REAL
C                                      AND BOTH ZEROS ARE COMPLEX.
C                                    THE SMALLER REAL ZERO IS FOUND
C                                      DIRECTLY FROM THE PRODUCT OF THE
C                                      ZEROS C/A
C                                    FIRST EXECUTABLE STATEMENT
      IF (RA.NE.ZERO) GO TO 10
      SR = ZERO
      IF (B1.NE.ZERO) SR = -C/B1
      RLR = ZERO
    5 SI = ZERO
      RLI = ZERO
      RETURN
   10 IF (C.NE.ZERO) GO TO 15
      SR = ZERO
      RLR = -B1/RA
      GO TO 5
C                                    COMPUTE DISCRIMINANT AVOIDING
C                                      OVERFLOW
   15 RB = B1/TWO
      IF (DABS(RB).LT.DABS(C)) GO TO 20
      E = ONE-(RA/RB)*(C/RB)
      D = DSQRT(DABS(E))*DABS(RB)
      GO TO 25
   20 E = RA
      IF (C.LT.ZERO) E = -RA
      E = RB*(RB/DABS(C))-E
      D = DSQRT(DABS(E))*DSQRT(DABS(C))
   25 IF (E.LT.ZERO) GO TO 30
C                                    REAL ZEROS
      IF (RB.GE.ZERO) D = -D
      RLR = (-RB+D)/RA
      SR = ZERO
      IF (RLR.NE.ZERO) SR = (C/RLR)/RA
      GO TO 5
C                                    COMPLEX CONJUGATE ZEROS
   30 SR = -RB/RA
      RLR = SR
      SI = DABS(D/RA)
      RLI = -SI
      RETURN
      END
C
C
C
      SUBROUTINE LEQ2C  (A,N,IA,B,M,IB,IJOB,WA,WK,IER)
C
      COMPLEX*16         A(IA,1),B(IB,1),WA(N,1),TEMPA,TEMPB,TEMPC
      DOUBLE PRECISION   WK(N),TA(2),TB(2),TC(2)
      DOUBLE PRECISION   AR,AI,BR,BI,CR,CI,DXNORM,XNORM,ZERO
      DOUBLE PRECISION   ACC(2)
      EQUIVALENCE        (TA(1),TEMPA),(TB(1),TEMPB),(TC(1),TEMPC),
     *                   (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI),
     *                   (TC(1),CR),(TC(2),CI)
      DATA               ZERO/0.0D0/
      DATA               ITMAX/50/
```

```
C                                FIRST EXECUTABLE STATEMENT
      IER = 0
      N1 = N+1
      N2 = N+2
      IF (IJOB .EQ. 2) GO TO 15
C                                SAVE MATRIX A
      DO 10 I = 1,N
         DO 5 J = 1,N
            WA(I,J) = A(I,J)
    5    CONTINUE
   10 CONTINUE
C                                FACTOR MATRIX A
      CALL LEQT1C (WA,N,N,B,M,IB,1,WK,IER)
      IF (IER .NE. 0) GO TO 9000
      IF (IJOB .EQ. 1) GO TO 9005
C                                SAVE THE RIGHT HAND SIDES
   15 DO 65 J = 1,M
         DO 20 I = 1,N
            WA(I,N1) = B(I,J)
   20    CONTINUE
C                                OBTAIN A SOLUTION
         CALL LEQT1C(WA,N,N,WA(1,N1),1,N,2,WK,IER)
C                                COMPUTE THE NORM OF THE SOLUTION
         XNORM = ZERO
         DO 25 I = 1,N
            TEMPA = WA(I,N1)
            XNORM = DMAX1(XNORM,DABS(AR),DABS(AI))
   25    CONTINUE
         IF (XNORM .EQ. ZERO) GO TO 65
C                                COMPUTE RESIDUALS
         DO 50 ITER = 1,ITMAX
            DO 40 I = 1,N
               TEMPB = B(I,J)
               ACC(1) = 0.0D0
               ACC(2) = 0.0D0
               CALL VXADD(BR,ACC)
               DO 30 JJ = 1,N
                  TEMPA = A(I,JJ)
                  TEMPB = WA(JJ,N1)
                  CALL VXMUL(-AR,BR,ACC)
                  CALL VXMUL(AI,BI,ACC)
   30          CONTINUE
               CALL VXSTO(ACC,CR)
               TEMPB = B(I,J)
               ACC(1) = 0.0D0
               ACC(2) = 0.0D0
               CALL VXADD(BI,ACC)
               DO 35 JJ = 1,N
                  TEMPA = A(I,JJ)
                  TEMPB = WA(JJ,N1)
                  CALL VXMUL(-AR,BI,ACC)
                  CALL VXMUL(-BR,AI,ACC)
   35          CONTINUE
               CALL VXSTO(ACC,CI)
               WA(I,N2) = TEMPC
   40       CONTINUE
            CALL LEQT1C(WA,N,N,WA(1,N2),1,N,2,WK,IER)
            DXNORM = ZERO
C                                UPDATE THE SOLUTION
            DO 45 I = 1,N
```

```
            WA(I,N1) = WA(I,N1)+WA(I,N2)
            TEMPA = WA(I,N2)
            DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI))
   45     CONTINUE
          IF (XNORM+DXNORM .EQ. XNORM) GO TO 55
   50   CONTINUE
        IER = 130
C                                  STORE THE SOLUTION
   55   DO 60 JK = 1,N
           B(JK,J) = WA(JK,N1)
   60   CONTINUE
        IF (IER .NE. 0) GO TO 9000
   65 CONTINUE
      GO TO 9005
 9000 CONTINUE
      CALL UERTST(IER,6HLEQ2C )
 9005 RETURN
      END
C
C
C
      SUBROUTINE LEQT1C (A,N,IA,B,M,IB,IJOB,WA,IER)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            N,IA,M,IB,IJOB,IER
      COMPLEX*16         A(IA,N),B(IB,M)
      DOUBLE PRECISION   WA(N)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   P,Q,ZERO,ONE,T(2),RN,BIG
      COMPLEX*16         SUM,TEMP
      INTEGER            I,J,JM1,IM1,K,IMAX,JP1,IW,N1
      EQUIVALENCE        (SUM,T(1))
      DATA               ZERO/0.0D0/,ONE/1.D0/
C                                  INITIALIZATION
C                                  FIRST EXECUTABLE STATEMENT
      IER = 0
      IF (IJOB .EQ. 2) GO TO 75
      RN = N
C                                  FIND EQUILIBRATION FACTORS
      DO 10 I=1,N
         BIG = ZERO
         DO 5 J=1,N
            TEMP = A(I,J)
            P = CDABS(TEMP)
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .EQ. ZERO) GO TO 105
         WA(I) = ONE/BIG
   10 CONTINUE
C                                  L-U DECOMPOSITION
      DO 70 J = 1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 25
C                                  COMPUTE U(I,J), I=1,....,J-1
         DO 20 I=1,JM1
            SUM = A(I,J)
            IM1 = I-1
            IF (IM1 .LT. 1) GO TO 20
            DO 15 K=1,IM1
               SUM = SUM-A(I,K)*A(K,J)
   15       CONTINUE
```

```
            A(I,J) = SUM
   20    CONTINUE
   25    P = ZERO
C                                          COMPUTE U(J,J) AND L(I,J), I=J+1,...,
         DO 45 I=J,N
            SUM = A(I,J)
            IF (JM1 .LT. 1) GO TO 40
            DO 35 K=1,JM1
               SUM = SUM-A(I,K)*A(K,J)
   35       CONTINUE
            A(I,J) = SUM
   40       Q = WA(I)*CDABS(SUM)
            IF (P .GE. Q) GO TO 45
            P = Q
            IMAX = I
   45    CONTINUE
C                                          TEST FOR ALGORITHMIC SINGULARITY
         Q = RN+P
         IF (Q .EQ. RN) GO TO 105
         IF (J .EQ. IMAX) GO TO 60
C                                          INTERCHANGE ROWS J AND IMAX
         DO 50 K=1,N
            TEMP = A(IMAX,K)
            A(IMAX,K) = A(J,K)
            A(J,K) = TEMP
   50    CONTINUE
         WA(IMAX) = WA(J)
   60    WA(J) = IMAX
         JP1 = J+1
         IF (JP1 .GT. N) GO TO 70
C                                          DIVIDE BY PIVOT ELEMENT U(J,J)
         TEMP = A(J,J)
         DO 65 I = JP1,N
            A(I,J) = A(I,J)/TEMP
   65    CONTINUE
   70 CONTINUE
   75 IF (IJOB .EQ. 1) GO TO 9005
      DO 103 K = 1,M
C                                          SOLVE UX = Y FOR X
         IW = 0
         DO 90 I = 1,N
            IMAX = WA(I)
            SUM = B(IMAX,K)
            B(IMAX,K) = B(I,K)
            IF (IW .EQ. 0) GO TO 85
            IM1 = I-1
            DO 80 J = IW,IM1
               SUM = SUM-A(I,J)*B(J,K)
   80       CONTINUE
            GO TO 88
   85       IF (T(1) .NE. ZERO .OR. T(2) .NE. ZERO) IW = I
   88       B(I,K) = SUM
   90    CONTINUE
C                                          SOLVE LY = B FOR Y
         N1 = N+1
         DO 100 IW = 1,N
            I = N1-IW
            JP1 = I+1
            SUM = B(I,K)
            IF (JP1 .GT. N) GO TO 98
```

```
          DO 95 J = JP1,N
             SUM = SUM-A(I,J)*B(J,K)
   95     CONTINUE
   98     B(I,K) = SUM/A(I,I)
  100    CONTINUE
  103 CONTINUE
      GO TO 9005
C                                      ALGORITHMIC SINGULARITY
  105 IER = 129
 9000 CONTINUE
C                                      PRINT ERROR
      CALL UERTST(IER,6HLEQT1C)
 9005 RETURN
      END
```

APPENDIX B

SAMCJ Program Listing

```
C*****************************************************************
C*****************************************************************
C**                                                           ***
C**                     PROGRAM SAMCJ                         ***
C**                                                           ***
C**   STRENGTH ANALYSIS OF MULTI-FASTENER COMPOSITE JOINTS    ***
C**                                                           ***
C*****************************************************************
C*****************************************************************
C
C      SAMCJ COMPUTES THE LOAD DISTRIBUTION AMONG FASTENERS IN
C      A MULTI-FASTENED COMPOSITE/ METALLIC JOINT, AND PREDICTS
C      THE JOINT FAILURE LOAD, FAILURE MODE, AND FAILURE LOCATION.
C      THE FASTENER LOAD DISTRIBUTION IS DETERMINED BY A
C      FINITE ELEMENT METHOD WITH THE USE OF SPECIAL FINITE
C      ELEMENTS. THE SUBSEQUENT FAILURE ANALYSIS IS BASED
C      ON AN AVERAGE STRESS FAILURE CRITERION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NPLY(2),WASHD(2),STM(3)
      DIMENSION NEF(2),NLH(2),NOH(2),NPL(2)
      DIMENSION NGEF(2,10,10),NGLH(2,10,10),NGOH(2,10,10)
      DIMENSION NGPL(2,10,10),NUMEF(2,10),NUMLH(2,10)
      DIMENSION NUMOH(2,10),NUMPL(2,10)
      DIMENSION NELORD(2,25,25),NELDIS(50,5,2)
      DIMENSION ELLOAD(50,2),PSMX(50,4),NZERO(50),NBDARY(25)
      DIMENSION XOUT(600),YOUT(600),PLYK(100),BARK(100),BARU(100)
      DIMENSION ELSTFF(50,10,10),ELSTSS(50,50,10)
      DIMENSION OSSX(20),OSSN(20),ANR(200),RHS(200),PBC(200)
      DIMENSION GLSTFF(200,200),ASQM(200,200),ANR2(200)
      DIMENSION RDSTFF(50,2),WGHT(500),ERO(50)
      DIMENSION NELPLS(2,50),NELPT(2,50,50)
      DIMENSION ELTHK(50),NELCON(50,6),NELCNA(50,6)
      DIMENSION GCOORD(150,2),PLYTHK(2,25),NELFAS(25,3)
      DIMENSION NELFSA(25,3)
      DIMENSION ELWDTH(50),NGRID(150),LYPN(50)
      DIMENSION FSCD(50,3),NELTYP(50)
      DIMENSION MTL(3,15),R(2)
      DIMENSION ANGK(5,2),NUMPLY(2),CM(2)
      DIMENSION ANG(5,2),IPLY(100,2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2),H(2)
      DIMENSION STULT(5,2)
      DIMENSION XC(5),YC(5)
      DIMENSION AONT(2),AOBR(2),AOSO(2)
      DIMENSION ELFAIL(50,3)
      COMMON/AOV/AONT,AOBR,AOSO
      COMMON/OSXN/OSSX,OSSN
      COMMON/NPLS/NELPLS,LYPN
      COMMON/XCYC/XC,YC
      COMMON/NCN/NELCON,NELCNA,NF,DIS
      COMMON/STMT/STM,CM
      COMMON/SMX/PSMX
      COMMON/STN/STULT
      COMMON/LAMF/ELFAIL
      COMMON/FSS/FASE,FASV,FASD
```

```
      COMMON/RT/R
      COMMON/MFS/FSCD
      COMMON/NTP/NELTYP
      COMMON/NPT/NOPT2,NOPT6,NOPT7,NOPT8
      COMMON/MOD/E1,E2,G12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/ELP/AX,BX,NOUT,NSTS
      COMMON/FCC/ELWDTH,ELTHK,ELLOAD
      COMMON/NCST/NCASE,NTYPE
      COMMON/DISP/ANR2
      COMMON/PBB/PLYK,BARK,BARU
      COMMON/ELS/ELSTFF,ELSTSS
      COMMON/CMT2/XOUT,YOUT
      COMMON/SER/NT,NB
      DATA Y/'Y'/
      DATA CMC/'C'/
C
C
C     READ IN REQUIRED INPUT DATA
C
C
      WRITE(6,876)
  876 FORMAT(///,10X,' PROGRAM SAMCJ',//,
     *' PROGRAM SAMCJ PREDICTS THE FAILURE LOAD, FAILURE ',/,
     *' LOCATION, AND FAILURE MODE IN MULTIPLY-FASTENED,',/,
     *' SINGLE OR DOUBLE LAP COMPOSITE SHEAR JOINTS.    ',//,
     *' THE ANALYSIS ASSUMES THAT INPUT PARAMETERS ARE ',/,
     *' SPECIFIED IN ENGLISH UNITS - LENGTH IN INCHES,',/,
     *' MODULI AND STRENGTHS IN PSI.              ',//)
      WRITE(6,900)
  900 FORMAT(' ENTER:                    ',/,
     *'        1 FOR SLS (SINGLE LAP SHEAR)',/,
     *'        2 FOR DLS (DOUBLE LAP SHEAR)',/)
      READ(5,*) NSDLS
      WRITE(6,911)
  911 FORMAT(' ENTER:                  ',/,
     *'        1 FOR STATIC TENSION    ',/,
     *'        2 FOR STATIC COMPRESSION',/)
      READ(5,*) LTNCM
  106 FORMAT(A1)
  380 CONTINUE
      DO 300 K=1,2
      IF(K.EQ.1) WRITE(6,912)
      IF(K.EQ.2) WRITE(6,913)
  912 FORMAT(' IS THE TOP PLATE A COMPOSITE OR A METAL?')
  913 FORMAT(' IS THE BOTTOM PLATE A COMPOSITE OR A METAL?')
      WRITE(6,914)
  914 FORMAT(' ENTER C OR M IN THE FIRST FIELD')
      READ(5,106) CM(K)
      WRITE(6,203)
  203 FORMAT(' INPUT MATERIAL DESCRIPTION OF THIS PLATE ',/,
     *'        EX: ASA/3501-6')
      READ(5,204) (MTL(K,I),I=1,15)
  204 FORMAT(15A4)
  300 CONTINUE
      IF(CM(1).NE.CMC.OR.CM(2).NE.CMC) WRITE(6,754)
  754 FORMAT(/,' NOTE: FOR COMPUTATIONAL PURPOSES A ',/,
     *' METALLIC PLATE IS MODELED AS A 30 PLY ',/,
     *' LAMINATE OF 0 DEGREE PLIES WITH ISOTROPIC',/,
     *' MATERIAL PROPERTIES',/)
```

```
      DO 306 K=1,2
      IF(K.EQ.1) WRITE(6,216)
      IF(K.EQ.2) WRITE(6,555)
  216 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF THE TOP PLATE')
  555 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF THE BOTTOM PLATE')
      IF(CM(K).EQ.CMC) GO TO 85
      WRITE(6,95)
   95 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO')
      READ(5,*) E1(K),V12(K)
      E2(K)=E1(K)
      G12(K)=E1(K)/(2.0D0*V12(K))
      V21(K)=V12(K)*E2(K)/E1(K)
      GO TO 306
   85 CONTINUE
      WRITE(6,217)
  217 FORMAT(' INPUT YOUNGS MODULI, E1 AND E2')
      READ(5,*) E1(K),E2(K)
      WRITE(6,218)
  218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO')
      READ(5,*) G12(K),V12(K)
      V21(K)=V12(K)*E2(K)/E1(K)
  306 CONTINUE
  307 CONTINUE
  290 CONTINUE
      DO 303 K=1,2
      IF(CM(K).EQ.CMC) GO TO 45
      NUMPLY(K)=1
      GO TO 303
   45 CONTINUE
      IF(K.EQ.1) WRITE(6,207)
      IF(K.EQ.2) WRITE(6,702)
  207 FORMAT(' INPUT TOTAL NUMBER OF DISTINCT PLY ',/,
     *' ORIENTATIONS IN THE TOP PLATE')
  702 FORMAT(' INPUT TOTAL NUMBER OF DISTINCT PLY ',/,
     *' ORIENTATIONS IN THE BOTTOM PLATE')
      READ(5,*) NUMPLY(K)
  303 CONTINUE
      DO 209 K=1,2
      IF(CM(K).EQ.CMC) GO TO 55
      ANG(1,K)=0.
      GO TO 209
   55 CONTINUE
      N=NUMPLY(K)
      DO 209 L=1,N
      WRITE(6,206) L
  206 FORMAT(' INPUT ORIENTATION OF PLY TYPE NO',I5)
      READ(5,*) ANG(L,K)
  209 CONTINUE
      WRITE(6,1823)
 1823 FORMAT(/,' THICHKNESS VARIATIONS MAY BE APPROXIMATED',/,
     *' BY ASSIGNING DIFFERENT LAYUPS TO ELEMENTS',/,
     *' IN A COMPOSITE PLATE OR BY SPECIFYING DIFFERENT',/,
     *' THICHKNESSES TO ELEMENTS IN A METALLIC PLATE',/)
      IF(NSDLS.EQ.2) WRITE(6,789)
  789 FORMAT(/,' NOTE: FOR THE DOUBLE LAP SHEAR CASE, FOR',/,
     *' THE BOTTOM PLATE, ENTER ONLY HALF FOR THE ',/,
     *' LAYUP FOR A COMPOSIT OR HALF THE THICKNESS ',/,
     *' FOR A METALLIC',/)
      DO 811 I=1,2
      IF(CM(I).EQ.CMC) GO TO 891
```

```
      NELPLS(I,1)=30
      DO 892 III=1,30
  892 NELPT(I,1,III)=1
      GO TO 811
  891 CONTINUE
      IF(I.EQ.1) WRITE(6,812)
      IF(I.EQ.2) WRITE(6,813)
  812 FORMAT(/,' ENTER NUMBER OF DIFFERENT LAYUPS IN THE ',/,
     *' TOP PLATE')
  813 FORMAT(/,' ENTER NUMBER OF DIFFERENT LAYUPS IN THE ',/,
     *' BOTTOM PLATE')
      READ(5,*) NL
      DO 814 J=1,NL
      WRITE(6,815) J
  815 FORMAT(' ENTER NUMBER OF PLIES IN LAYUP NO ',I5)
      READ(5,*) NELPLS(I,J)
      WRITE(6,816)
  816 FORMAT(' ENTER PLY THICKNESS FOR THIS LAYUP')
      READ(5,*) PLYTHK(I,J)
      NN=NELPLS(I,J)
      WRITE(6,818)
  818 FORMAT(' ENTER SEQUENCE OF PLY TYPES FROM TOP TO BOTTOM')
      DO 817 K=1,NN
      READ(5,*) NELPT(I,J,K)
  817 CONTINUE
  814 CONTINUE
  811 CONTINUE
      WRITE(6,855)
  855 FORMAT(/' FASTENER DESCRIPTION:',/)
      WRITE(6,250)
  250 FORMAT(' INPUT MATERIAL DESCRIPTION FOR FASTENER')
      READ(5,251) (MTL(3,I),I=1,15)
  251 FORMAT(15A4)
      WRITE(6,252)
  252 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO FOR',/,
     *' THE FASTENER')
      READ(5,*) FASE,FASV
      WRITE(6,253)
  253 FORMAT(' INPUT THE DIAMETER OF THE FASTENER')
      READ(5,*) FASD
      WRITE(6,888)
  888 FORMAT(/,' FASTENER TYPE                   ',/,
     *'          ENTER:      1 FOR PROTRUDING HEAD ',/,
     *'                      2 FOR COUNTERSUNK HEAD')
      READ(5,*) NFTYP
      R(1)=1.0D10
      R(2)=1.0D10
      IF(NFTYP.EQ.1) GO TO 360
      WRITE(6,889)
  889 FORMAT(/,' ENTER PLATE WHICH CONTAINS THE COUNTERSUNK',/,
     *' HEAD (OPPOSITE PLATE ASSUMES THE NUT HEAD)     ',//,
     *'          ENTER:  1 FOR TOP PLATE       ',/,
     *'                  2 FOR BOTTOM PLATE    ')
      READ(5,*) N
      R(N)=0.0D0
  360 CONTINUE
      WRITE(6,477)
  477 FORMAT(/,' GRID LAYOUT:',/)
C
C     INPUT GRIDS, ELEMENT CONNECTIVITY AND PROPERTIES
```

```
C
C     TOP PLATE
C
      WRITE(6,689)
  689 FORMAT(' ENTER NUMBER OF GRIDS IN TOP PLATE')
      READ(5,*) NGP1
      WRITE(6,371) NGP1
  371 FORMAT(/,' ENTER ',I8,' GRID POINTS                   ',/,
     *'                                                  ',/,
     *' FORMAT: GRID ID, X AND Y COORDINATES ')
      DO 603 I=1,NGP1
      READ(5,*) NGRID(I),GCOORD(I,1),GCOORD(I,2)
  603 CONTINUE
C
C     BOTTOM PLATE
C
      WRITE(6,683)
  683 FORMAT(' ENTER NUMBER OF GRIDS IN BOTTOM PLATE')
      READ(5,*) NGP2
      NOTOT=NGP1+NGP2
      WRITE(6,371) NGP2
      NP1=NGP1+1
      DO 604 I=NP1,NOTOT
      READ(5,*) NGRID(I),GCOORD(I,1),GCOORD(I,2)
  604 CONTINUE
      WRITE(6,883)
  883 FORMAT(/,' ELEMENT DESCRIPTION:',/)
      WRITE(6,399)
  399 FORMAT(/,'                                           ',/,
     *' PLANAR ELEMENTS ARE NUMBERED                  ',/,
     *' CLOCKWISE AS SHOWN:                       ',//,
     *'                          N2      N3                     ',/,
     *'                              N5                         ',/,
     *'                          N1      N4                     ',/,
     *'                                                         ',/,
     *' ELEMENT TYPES ARE DESIGNATED AS FOLLOWS:                ',/,
     *'                                                         ',/,
     *'  4 NODE PLAIN ELEMENT           TYPE NO. 1              ',/,
     *'  5 NODE LOADED HOLE ELEMENT     TYPE NO. 2              ',/,
     *'  4 NODE OPEN HOLE ELEMENT       TYPE NO. 3              ',/,
     *'                                                        ',/,
     *'                                                         ',/,
     *' (NOTE: ENTER N5=0 FOR FOUR NODE ELEMENTS)              ',/)
      WRITE(6,191)
  191 FORMAT(' ENTER NUMBER OF ELEMENTS IN TOP PLATE')
      READ(5,*) NEL1
      DO 474 I=1,NEL1
      WRITE(6,388) I
  388 FORMAT(' FOR ELEMENT NO',I5,/,
     *' ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE')
      READ(5,*) (NELCON(I,J),J=1,6),NELTYP(I)
      DO 591 IL=2,6
      IC=0
      NELCNA(I,1)=NELCON(I,1)
      DO 592 KL=1,NGP1
      IF(NELCON(I,IL).EQ.NGRID(KL)) IC=1
      IF(NELCON(I,IL).EQ.NGRID(KL)) NELCNA(I,IL)=KL
      IF(IC.EQ.1) GO TO 591
  592 CONTINUE
  591 CONTINUE
```

```
      IF(CM(1).EQ.CMC) GO TO 627
      IF(NELTYP(I).NE.1) GO TO 721
      WRITE(6,1721)
 1721 FORMAT(' ENTER ELEMENT THICKNESS')
      READ(5,*) ATH
  721 IF(NELTYP(I).NE.2) GO TO 722
      WRITE(6,723)
  723 FORMAT(' ENTER ELEMENT THICKNESS')
      READ(5,*) ATH
      FSCD(I,1)=GCOORD(NELCNA(I,6),1)
      FSCD(I,2)=GCOORD(NELCNA(I,6),2)
      FSCD(I,3)=FASD/2.0D0
  722 IF(NELTYP(I).NE.3) GO TO 724
      WRITE(6,725)
  725 FORMAT(' ENTER ELEMENT THICKNESS, X AND Y COORDINATES',/,
     *' OF OPEN HOLE AND HOLE RADIUS')
      READ(5,*) ATH,(FSCD(I,J),J=1,3)
  724 ELTHK(I)=ATH/30.0D0
      PLYTHK(1,1)=ATH/30.0D0
      LYPN(I)=1
      GO TO 474
  627 CONTINUE
      IF(NELTYP(I).NE.1) GO TO 726
      WRITE(6,727)
  727 FORMAT(' ENTER ELEMENT LAYUP NO')
      READ(5,*) LYPN(I)
  726 IF(NELTYP(I).NE.2) GO TO 728
      WRITE(6,729)
  729 FORMAT(' ENTER ELEMENT LAYUP NO')
      READ(5,*) LYPN(I)
      FSCD(I,1)=GCOORD(NELCNA(I,6),1)
      FSCD(I,2)=GCOORD(NELCNA(I,6),2)
      FSCD(I,3)=FASD/2.0D0
  728 IF(NELTYP(I).NE.3) GO TO 730
      WRITE(6,731)
  731 FORMAT(' ENTER ELEMENT LAYUP NUMBER, X AND Y ',/,
     *' COORDINATES OF THE OPEN HOLE AND THE HOLE',/,
     *' RADIUS')
      READ(5,*) LYPN(I),(FSCD(I,J),J=1,3)
  730 ELTHK(I)=PLYTHK(1,LYPN(I))
  474 CONTINUE
      WRITE(6,688)
  688 FORMAT(/,' ENTER NUMBER OF ELEMENTS IN BOTTOM PLATE ')
      READ(5,*) NEL2
      NELTOT=NEL1+NEL2
      NP1=NEL1+1
      DO 611 I=NP1,NELTOT
      WRITE(6,800) I
  800 FORMAT(' FOR ELEMENT NO',I5,
     *' ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE')
      READ(5,*) (NELCON(I,J),J=1,6),NELTYP(I)
      DO 593 IL=2,6
      IC=0
      NELCNA(I,1)=NELCON(I,1)
      NIN=NOP1+1
      DO 594 KL=NIN,NGTOT
      IF(NELCON(I,IL).EQ.NGRID(KL)) IC=1
      IF(NELCON(I,IL).EQ.NGRID(KL)) NELCNA(I,IL)=KL
      IF(IC.EQ.1) GO TO 593
  594 CONTINUE
```

```
  593 CONTINUE
      IF(CM(2).EQ.CMC) GO TO 927
      IF(NELTYP(I).NE.1) GO TO 921
      WRITE(6,1921)
 1921 FORMAT(' ENTER ELEMENT THICKNESS')
      READ(5,*) ATH
  921 IF(NELTYP(I).NE.2) GO TO 922
      WRITE(6,1923)
 1923 FORMAT(' ENTER ELEMENT THICKNESS')
      READ(5,*) ATH
      FSCD(I,1)=GCOORD(NELCNA(I,6),1)
      FSCD(I,2)=GCOORD(NELCNA(I,6),2)
      FSCD(I,3)=FASD/2.0D0
      WRITE(6,3443) I,NELCNA(I,6),FSCD(I,1),FSCD(I,2)
 3443 FORMAT(' C I NELCNA FSCD12',I5,2X,5(D9.3,2X))
  922 IF(NELTYP(I).NE.3) GO TO 924
      WRITE(6,925)
  925 FORMAT(' ENTER ELEMENT THICKNESS, X AND Y COORDINATES',/,
     *' OF OPEN HOLE AND HOLE RADIUS')
      READ(5,*) ATH,(FSCD(I,J),J=1,3)
  924 ELTHK(I)=ATH/30.0D0
      PLYTHK(2,1)=ATH/30.0D0
      LYPN(I)=1
      GO TO 611
  927 CONTINUE
      IF(NELTYP(I).NE.1) GO TO 926
      WRITE(6,1927)
 1927 FORMAT(' ENTER ELEMENT LAYUP NO')
      READ(5,*) LYPN(I)
  926 IF(NELTYP(I).NE.2) GO TO 928
      WRITE(6,929)
  929 FORMAT(' ENTER ELEMENT LAYUP NO')
      READ(5,*) LYPN(I)
      FSCD(I,1)=GCOORD(NELCNA(I,6),1)
      FSCD(I,2)=GCOORD(NELCNA(I,6),2)
      FSCD(I,3)=FASD/2.0D0
  928 IF(NELTYP(I).NE.3) GO TO 930
      WRITE(6,931)
  931 FORMAT(' ENTER ELEMENT LAYUP NUMBER, X AND Y ',/,
     *' COORDINATES OF THE OPEN HOLE AND THE HOLE',/,
     *' RADIUS')
      READ(5,*) LYPN(I),(FSCD(I,J),J=1,3)
  930 ELTHK(I)=PLYTHK(2,LYPN(I))
  611 CONTINUE
      WRITE(6,1741)
 1741 FORMAT(/,' FASTENERS ARE MODELED BY EFFECTIVE ',/,
     *' FASTENER ELEMENTS WHICH PROVIDE THE ',/,
     *' ELASTIC LINK BETWEEN THE TOP AND ',/,
     *' BOTTOM PLATES',/)
      WRITE(6,1711)
 1711 FORMAT(' ENTER NUMBER OF FASTENERS IN JOINT ')
      READ(5,*) NUMF
      WRITE(6,716)
  716 FORMAT(/,'                                        ',/,
     *'                                        ',/,
     *' EFFECTIVE FASTENER ELEMENTS ARE        ',/,
     *' NUMBERED AS SHOWN:                       ',/,
     *'                       N1 (TOP PLATE)      ',/,
     *'                                           ',/,
     *'                       N2 (BOTTOM PLATE)   ',/,
```

```
     *'                                                  ',/,
     *' WHERE N1 AND N2 CORRESPOND TO THE CENTRAL',/,
     *' NODES IN LOADED HOLE ELEMENTS             ',//,
     *' FORMAT: ELEMENT ID, N1, N2            ')
      DO 717 I=1,NUMF
      WRITE(6,711) I
  711 FORMAT(' ENTER ELEMENT NO',I5)
      READ(5,*) (NELFAS(I,J),J=1,3)
  717 CONTINUE
C
C     DETERMINE GRID STORAGE LOCATIONS FOR
C     ELEMENT NODES
C
      DO 612 I=1,NEL1
      N=6
      IF(NELTYP(I).NE.2) N=5
      NELCNA(I,1)=NELCON(I,1)
      DO 613 J=2,N
      IC=0
      DO 614 K=1,NOP1
      IF(NELCON(I,J).EQ.NGRID(K)) IC=1
      IF(NELCON(I,J).EQ.NGRID(K)) NELCNA(I,J)=K
      IF(IC.EQ.1) GO TO 613
  614 CONTINUE
  613 CONTINUE
  612 CONTINUE
      NP1=NEL1+1
      DO 395 I=NP1,NELTOT
      N=6
      IF(NELTYP(I).NE.2) N=5
      NELCNA(I,1)=NELCON(I,1)
      DO 616 J=2,N
      IC=0
      NIN=NOP1+1
      DO 617 K=NIN,NGTOT
      IF(NELCON(I,J).EQ.NGRID(K)) IC=1
      IF(NELCON(I,J).EQ.NGRID(K)) NELCNA(I,J)=K
      IF(IC.EQ.1) GO TO 616
  617 CONTINUE
  616 CONTINUE
  395 CONTINUE
      DO 741 I=1,NUMF
      N=2
      NELFSA(I,1)=NELFAS(I,1)
      DO 242 J=1,N
      IC=0
      DO 243 K=1,NGTOT
      IF(NELFAS(I,J+1).EQ.NGRID(K)) IC=1
      IF(NELFAS(I,J+1).EQ.NGRID(K)) NELFSA(I,J+1)=K
      IF(IC.EQ.1) GO TO 242
  243 CONTINUE
  242 CONTINUE
  741 CONTINUE
C
C     COMPUTE ELEMENT WIDTHS
C
      DO 239 I=1,NELTOT
      ELWDTH(I)=DABS(GCOORD(NELCNA(I,3),2)-GCOORD(NELCNA(I,2),2))
  239 CONTINUE
C
```

```
C
C     GROUP ELEMENTS TO AVOID THE DUPLICATE
C     CALCULATION OF IDENTICAL STIFFNESS
C     MATRICIES
C
      WRITE(6,3000)
 3000 FORMAT(/,' TO REDUCE RUN TIMES, ELEMENTS MAY BE ',/,
     *' GROUPED INTO SETS WHICH WILL BE ASSIONED',/,
     *' IDENTICAL STIFFNESS MATRICIES             ',/,
     *' ENTER:      1 TO USE THIS OPTION          ',/,
     *'             2 OTHERWISE    ')
      READ(5,*) NOPT
      IF(NOPT.EQ.1) GO TO 3001
      N1=0
      N2=0
      N3=0
      DO 3002 I=1,NEL1
      IF(NELTYP(I).EQ.1) N1=N1+1
      IF(NELTYP(I).EQ.2) N2=N2+1
 3002 IF(NELTYP(I).EQ.3) N3=N3+1
      NEF(1)=NUMF
      NLH(1)=N2
      NOH(1)=N3
      NPL(1)=N1
      N=NUMF
      DO 3003 I=1,N
 3003 NGEF(1,I,1)=NELFAS(I,1)
      IC=0
      DO 3004 I=1,NEL1
      IF(NELTYP(I).EQ.2) IC=IC+1
      IF(NELTYP(I).EQ.2) NGLH(1,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.2) NUMLH(1,IC)=1
 3004 CONTINUE
      IC=0
      DO 3005 I=1,NEL1
      IF(NELTYP(I).EQ.3) IC=IC+1
      IF(NELTYP(I).EQ.3) NGOH(1,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.3) NUMOH(1,IC)=1
 3005 CONTINUE
      IC=0
      DO 3006 I=1,NEL1
      IF(NELTYP(I).EQ.1) IC=IC+1
      IF(NELTYP(I).EQ.1) NGPL(1,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.1) NUMPL(1,IC)=1
 3006 CONTINUE
      N=NEL1+1
      N1=0
      N2=0
      N3=0
      DO 3007 I=N,NELTOT
      IF(NELTYP(I).EQ.1) N1=N1+1
      IF(NELTYP(I).EQ.2) N2=N2+1
 3007 IF(NELTYP(I).EQ.3) N3=N3+1
      NEF(2)=NUMF
      NLH(2)=N2
      NOH(2)=N3
      NPL(2)=N1
      N=NUMF
      DO 3008 I=1,N
 3008 NGEF(2,I,1)=NELFAS(I,1)
      IC=0
```

```
      N=NEL1+1
      DO 3010 I=N,NELTOT
      IF(NELTYP(I).EQ.2) IC=IC+1
      IF(NELTYP(I).EQ.2) NGLH(2,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.2) NUMLH(2,IC)=1
 3010 CONTINUE
      IC=0
      N=NEL1+1
      DO 3011 I=N,NELTOT
      IF(NELTYP(I).EQ.1) IC=IC+1
      IF(NELTYP(I).EQ.1) NGPL(2,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.1) NUMPL(2,IC)=1
 3011 CONTINUE
      IC=0
      N=NEL1+1
      DO 3012 I=N,NELTOT
      IF(NELTYP(I).EQ.3) IC=IC+1
      IF(NELTYP(I).EQ.3) NGOH(2,IC,1)=NELCON(I,1)
      IF(NELTYP(I).EQ.3) NUMOH(2,IC)=1
 3012 CONTINUE
      GO TO 3013
 3001 CONTINUE
      WRITE(6,3015)
 3015 FORMAT(/,' FOR THE TOP PLATE INPUT NUMBER OF GROUPS',/,
     *' FOR THE EFFECTIVE FASTENER, LOADED HOLE, UNLOADED',/,
     *' HOLE AND PLAIN ELEMENT      ',/,
     *' (INPUT 0 IF ELEMENT TYPE IS NOT USED)')
      READ(5,*) NEF(1),NLH(1),NOH(1),NPL(1)
      WRITE(6,3016)
 3016 FORMAT(' GROUPING OF EFFECTIVE FASTENER ELEMENTS:')
      N=NEF(1)
      DO 3017 I=1,N
      WRITE(6,3018) I
 3018 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8)
      READ(5,*) NUMEF(1,I)
      N1=NUMEF(1,I)
      WRITE(6,3019) N1
 3019 FORMAT(' ENTER ',I8,' ELEMENT IDS')
      READ(5,*) (NOEF(1,I,J),J=1,N1)
 3017 CONTINUE
      WRITE(6,3088)
 3088 FORMAT(/,' GROUPING OF LOADED HOLE ELEMENTS:')
      N=NLH(1)
      DO 3020 I=1,N
      WRITE(6,3021) I
 3021 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER ',I8)
      READ(5,*) NUMLH(1,I)
      N1=NUMLH(1,I)
      WRITE(6,3022) N1
 3022 FORMAT(' INPUT',I8,' ELEMENT IDS')
      READ(5,*) (NGLH(1,I,J),J=1,N1)
 3020 CONTINUE
      IF(NOH(1).EQ.0) GO TO 4071
      WRITE(6,3023)
 3023 FORMAT(' GROUPING OF UNLOADED HOLE ELEMENTS')
      N=NOH(1)
      DO 3024 I=1,N
      WRITE(6,3025) I
 3025 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8)
      READ(5,*) NUMOH(1,I)
```

```
      N1=NUMOH(1,I)
      WRITE(6,3026) N1
 3026 FORMAT(' ENTER',I8,' ELEMENT IDS')
      READ(5,*) (NGOH(1,I,J),J=1,N1)
 3024 CONTINUE
 4071 IF(NPL(1).EQ.0) GO TO 4072
      WRITE(6,3027)
 3027 FORMAT(' GROUPING OF PLAIN ELEMENTS:')
      N=NPL(1)
      DO 3031 I=1,N
      WRITE(6,3032) I
 3032 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8)
      READ(5,*) NUMPL(1,I)
      N1=NUMPL(1,I)
      WRITE(6,3033) N1
 3033 FORMAT(' ENTER',I8,' ELEMENT IDS')
      READ(5,*) (NOPL(1,I,J),J=1,N1)
 3031 CONTINUE
 4072 CONTINUE
      WRITE(6,4015)
 4015 FORMAT(/,' FOR THE BOTTOM PLATE INPUT NUMBER OF GROUPS',/,
     *' FOR THE LOADED HOLE,UNLOADED HOLE, AND PLAIN ',/,
     *' ELEMENTS                                      ',/,
     *' (INPUT 0 IF AN ELEMENT TYPE IS NOT USED)')
      READ(5,*) NLH(2),NOH(2),NPL(2)
      NEF(2)=NEF(1)
      N=NEF(1)
      DO 4017 I=1,N
      NUMEF(2,I)=NUMEF(1,I)
      N1=NUMEF(1,I)
      DO 4019 J=1,N1
 4019 NGEF(2,I,J)=NGEF(1,I,J)
 4017 CONTINUE
      WRITE(6,4088)
 4088 FORMAT(/,' GROUPING OF LOADED HOLE ELEMENTS:')
      N=NLH(2)
      DO 4020 I=1,N
      WRITE(6,4021) I
 4021 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER ',I8)
      READ(5,*) NUMLH(2,I)
      N1=NUMLH(2,I)
      WRITE(6,4022) N1
 4022 FORMAT(' INPUT',I8,' ELEMENT IDS')
      READ(5,*) (NGLH(2,I,J),J=1,N1)
 4020 CONTINUE
      IF(NOH(2).EQ.0) GO TO 4073
      WRITE(6,4023)
 4023 FORMAT(' GROUPING OF UNLOADED HOLE ELEMENTS')
      N=NOH(2)
      DO 4024 I=1,N
      WRITE(6,4025) I
 4025 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8)
      READ(5,*) NUMOH(2,I)
      N1=NUMOH(2,I)
      WRITE(6,4026) N1
 4026 FORMAT(' ENTER',I8,' ELEMENT IDS')
      READ(5,*) (NGOH(2,I,J),J=1,N1)
 4024 CONTINUE
 4073 IF(NPL(2).EQ.0) GO TO 4074
      WRITE(6,4027)
```

```
4027 FORMAT(' GROUPING OF PLAIN ELEMENTS:')
     N=NPL(2)
     DO 4031 I=1,N
     WRITE(6,4032) I
4032 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8)
     READ(5,*) NUMPL(2,I)
     N1=NUMPL(2,I)
     WRITE(6,4033) N1
4033 FORMAT(' ENTER',I8,' ELEMENT IDS')
     READ(5,*) (NGPL(2,I,J),J=1,N1)
4031 CONTINUE
4074 CONTINUE
3013 CONTINUE
     WRITE(6,3737)
3737 FORMAT(/,' INPUT DATA FOR FAILURE ANALYSIS:',/)
     DO 226 K=1,2
     IF(CM(K).NE.CMC) GO TO 2226
     WRITE(6,532) K
 532 FORMAT(' ENTER FIBER ULTIMATE STRAIN VALUES ',/,
    *' IN PLATE NO ',I8,/,
    *'            EPSILON ULT IN COMPRESSION ',/,
    *'            EPSILON ULT IN TENSION ',/,
    *'            GAMMA ULT IN SHEAR ',/)
     READ(5,*) (STULT(I,K),I=1,3)
     GO TO 2227
2226 CONTINUE
     WRITE(6,2229)
2229 FORMAT(' ENTER METALLIC STRENGTHS: ',/,
    *'    TENSILE STRENGTH    ',/,
    *'    COMPRESSIVE STRENGTH ',/,
    *'    SHEAR STRENGTH')
     READ(5,*) STM(1),STM(2),STM(3)
2227 CONTINUE
     WRITE(6,4054)
4054 FORMAT(/,' AN AVERAGE STRESS CRITERIA IS USED TO ',/,
    *' PREDICT FAILURE. AO VALUES ARE REQUIRED AS ',/,
    *' CHARACTERISTIC DISTANCES OVER WHICH STRESSES',/,
    *' ARE TO BE AVERAGED AND COMPARED TO UNNOTCHED',/,
    *' LAMINATE STRENGTHS TO PREDICT FAILURE',/)
     WRITE(6,5432) K
5432 FORMAT(' ENTER AO VALUES FOR STRESS AVERAGING',/,
    *' FOR EACH FAILURE MODE IN PLATE NO',I5,/,
    *'           AONT = NET SECTION        ',/,
    *'           AOBR = BEARING            ',/,
    *'           AOSO = SHEAROUT            ')
     READ(5,*) AONT(K),AOBR(K),AOSO(K)
 226 CONTINUE
C
C     CASE HEADING
C
     WRITE(6,143)
 143 FORMAT(///,10X,'PROGRAM SAMCJ',//)
     IF(NSDLS.EQ.1) WRITE(6,633)
     IF(NSDLS.EQ.2) WRITE(6,634)
 633 FORMAT(2X,'A SINGLE LAP SHEAR PANEL WILL BE ANALYZED',/)
 634 FORMAT(2X,'A DOUBLE LAP SHEAR PANEL WILL BE ANALYZED',/)
     IF(LTNCM.EQ.1) WRITE(6,823)
     IF(LTNCM.EQ.2) WRITE(6,824)
 823 FORMAT(2X,'LOADED IN STATIC TENSION',/)
 824 FORMAT(2X,'LOADED IN STATIC COMPRESSION',/)
```

```
      DO 241 I=1,2
      WRITE(6,600) I
  600 FORMAT(10X,'PLATE NO ',I5,' :',/)
      WRITE(6,601) (MTL(I,J),J=1,15)
  601 FORMAT(2X,15A4,/)
      HT=NELPLS(I,1)*PLYTHK(I,1)
      WRITE(6,691) E1(I),E2(I),G12(I),V12(I),V21(I)
  691 FORMAT(2X,'MATERIAL PROPERTIES',/,/,
     *10X,'E1  =',D9.3,' PSI',/,
     *10X,'E2  =',D9.3,' PSI',/,
     *10X,'G12 =',D9.3,' PSI',/,
     *10X,'NU12=',D9.3,/,
     *10X,'NU21=',D9.3,/)
  241 CONTINUE
      WRITE(6,606)
  606 FORMAT(10X,'FASTENER DESCRIPTION:',/)
      WRITE(6,607) (MTL(3,J),J=1,15)
  607 FORMAT(2X,15 4,/)
      WRITE(6,478) FASD
  478 FORMAT(2X,' DIAMETER = ',D9.3,' INCHES',/)
      WRITE(6,609) FASE,FASV
  609 FORMAT(2X,' MATERIAL PROPERTIES',/,/
     *10X,'E =' D9.3,' PSI',/,
     *10X,'MU=',D9.3,/)
  708 CONTINUE
      WRITE(6,923
  923 FORMAT(//,10X,'FAILURE ANALYSIS',/)
      WRITE(6,558)
  558 FORMAT(2X,' N AVERAGE STRESS CRITERION WILL BE USED',/)
      DO 631 I=1,2
      WRITE(6,632) I
  632 FORMAT(2X,'PLATE NUMBER',I5,/)
      NP=NUMPLY(I)
      IF(CM(I).NE.CM) GO TO 3112
      WRITE(6,713)
  713 FORMAT(/,2X,'FIBER STRAIN ULTIMATES',/)
  776 WRITE(6,677) (STULT(LL,I),LL=1,3)
  677 FORMAT 2X,'EPSILON ULT COMP = ',D9.3,/,
     *2X,'EPSILON ULT TEN  = ',D9.3,/,
     *2X,'GAMMA ULT SHEAR  = ',D9.3,/)
      GO TO 3113
 3112 CONTINUE
      WRITE(6,3114)
 3114 FORMAT(' METALLIC STRENGTHS ',/)
      WRITE(6,3115) STM(1),STM(2),STM(3)
 3115 FORMAT(2X,'TENSILE STRENGTH      = ',D9.3,/,
     *2X,'COMPRESSIVE STRENGTH = ',D9.3,/,
     *2X,'SHEAR STRENGTH       = ',D9.3,/)
 3113 CONTINUE
      WRITE(6,1563)
 1563 FORMAT(//,'  CHARACTERISTIC DISTANCES',/)
      WRITE(6,564) AONT(I),AOBR(I),AOSD(I)
  564 FORMAT(' AONT = ',D9.3,' INCHES',/,
     *' AOBR = ',D9.3,' INCHES',/,
     *' AOSD = ',D9.3,' INCHES',/)
  631 CONTINUE
C
C     THE JOINT LOAD DISTRIBUTION IS CALCULATED USING THE
C     FINITE ELEMENT METHOD WITH SPECIAL PROBLEM-ADAPTED
C     ELEMENTS WHICH EFFECTIVELY REPRESENT THE STIFFNESS
```

```
C      PROPERTIES OF FASTENERS, LOADED HOLES, AND OPEN
C      HOLE REGIONS IN THE JOINT
C
C      INTERNAL APPLIED LOAD SET TO 1 KIP
C
       APP=1000.0
       IF(LTNCM.EQ.2) APP=-APP
       NELTOT=NEL1+NEL2
       NGTOT=NGP1+NGP2
C
C      INITIALIZE ARRAYS
C
       DO 1 I=1,50
       DO 3 J=1,4
       DO 3 K=1,4
     3 ELSTFF(I,J,K)=0.
     1 CONTINUE
       DO 4 I=1,200
       PBC(I)=0.
       RHS(I)=0.
       ANR(I)=0.
       ANR2(I)=0.
       DO 5 J=1,200
       GLSTFF(I,J)=0.
       ASQM(I,J)=0.
     5 CONTINUE
     4 CONTINUE
C
C      CALCULATION OF EFFECTIVE FASTENER ELEMENT
C      STIFFNESS MATRICIES
C
       WRITE(6,8418)
  8418 FORMAT(/,' PAUSE FOR STIFFNESS MATRIX CALCULATIONS',/)
       NLOOP=NEF(1)
       DO 444 I=1,NLOOP
       NEL=NGEF(1,I,1)
       DO 5001 II=1,NUMF
  5001 IF(NEL.EQ.NELFAS(II,1)) IEL=II
C
C      SEARCH FOR LOADED HOLE ELEMENTS CONNECTED TO
C      FASTENER ELEMENT
C
       NTOP=0
       NBOT=0
       DO 643 J=1,NEL1
   643 IF(NELFAS(IEL,2).EQ.NELCON(J,6)) NTOP=J
       NP1=NEL1+1
       DO 446 J=NP1,NELTOT
   446 IF(NELFAS(IEL,3).EQ.NELCON(J,6)) NBOT=J
       NPLY(1)=NELPLS(1,LYPN(NTOP))
       H(1)=ELTHK(NTOP)
       DO 910 JJJ=1,50
   910 IPLY(JJJ,1)=NELPT(1,LYPN(NTOP),JJJ)
       NPLY(2)=NELPLS(2,LYPN(NBOT))
       H(2)=ELTHK(NBOT)
       DO 113 JJJ=1,50
   113 IPLY(JJJ,2)=NELPT(2,LYPN(NBOT),JJJ)
C
C      INITIALIZE PARAMETERS FOR COLLOCATION
C
```

```
      NT=7
      NOUT=52
      NCLL=10
      NB=NOUT+4*NCLL
      AX=FSCD(NTOP,3)
      BX=FSCD(NTOP,3)
      DO 570 L=1,2
      PHI=0.0D0
      IF(L.EQ.2) PHI=90.D0
      DO 530 K=1,2
      NTB=NTOP
      IF(K.EQ.2) NTB=NBOT
C
C     ELEMENT VERTEXES ARE INTERNALLY
C     NUMBERED AS:
C                         3    2
C
C                         4    1
C
      SFX=(GCOORD(NELCNA(NTB,5),1)+GCOORD(NELCNA(NTB,2),1))/2.0D0
      SFY=(GCOORD(NELCNA(NTB,3),2)+GCOORD(NELCNA(NTB,2),2))/2.0D0
      DO 128 JJ=1,4
      XC(JJ)=GCOORD(NELCNA(NTB,6-JJ),1)-FSCD(NTB,1)
      YC(JJ)=GCOORD(NELCNA(NTB,6-JJ),2)-FSCD(NTB,2)
  128 CONTINUE
      XC(5)=XC(1)
      YC(5)=YC(1)
      W=ELWDTH(NTB)
      AST=1000.0
      CALL POLY(W,AST,JK,K,NCLL,LTNCM)
      CALL CIRC(W,AST,JK,K,LTNCM)
      NOPT4=1
      NCASE=1
      NTYPE=NELTYP(IEL)
      CALL FIGEOM(H,PHI,K,NOPT4,NCLL)
      CALL FBOLT(ANGK,H,PHI,K)
  580 CONTINUE
      N=NPLY(1)
      DO 30 II=1,N
      M=IPLY(II,1)
   30 PLYK(II)=ANGK(M,1)
      N=NPLY(2)
      DO 61 II=1,N
      N1=II+NPLY(1)
      N2=IPLY(II,2)
   61 PLYK(N1)=ANGK(N2,2)
C
C     CALCULATION OF FASTENER PROPERTIES
C
      FASG=FASE/(2.*(1.+FASV))
      FASLAM=5.*(1.0+FASV)/(7.+6.*FASV)
      FASR=FASD/2.
      FASA=ACOS(-1.)*FASR**2
      FASI=ACOS(-1.)*FASR**4/4.
      FASSS=FASLAM*FASG*FASA
      FASBS=FASE*FASI
      P=1000.
      CALL CENTD(H,FASSS,FASBS,P)
      CALL SOLVE(H,P,U1,U2)
      IF(L.EQ.2) GO TO 666
```

```
      RDSTFF(IEL,1)=DABS(P/(U1+U2))
      GO TO 570
  666 RDSTFF(IEL,2)=DABS(P/(U1+U2))
  570 CONTINUE
      IF(NUMEF(1,I).EQ.1) GO TO 444
      N=NUMEF(1,I)
      DO 5023 K=2,N
      DO 5024 L=1,NUMF
 5024 IF(NGEF(1,I,K).EQ.NELFAS(L,1)) IEL2=L
      RDSTFF(IEL2,1)=RDSTFF(IEL,1)
      RDSTFF(IEL2,2)=RDSTFF(IEL,2)
 5023 CONTINUE
  444 CONTINUE
 8584 CONTINUE
C
C     CALCULATION OF LOADED HOLE AND UNLOADED HOLE
C     ELEMENT STIFFNESS MATRICIES
C
C
C     INITIALIZE QAUSSIAN QUADRATURE POINTS AND WEIGHTS
C
      NOP=5
      GSSX(1)=-0.9739065285
      GSSX(2)=-0.8650633666
      GSSX(3)=-0.6794095682
      GSSX(4)=-0.4333953941
      GSSX(5)=-0.1488743389
      GSSW(1)=0.0666713443
      GSSW(2)=0.1494513491
      GSSW(3)=0.2190863625
      GSSW(4)=0.2692667193
      GSSW(5)=0.2955242247
      DO 588 I=1,NGP
      GSSX(I+NGP)=-GSSX(NGP-I+1)
      GSSW(I+NGP)=GSSW(NGP-I+1)
  588 CONTINUE
      NAVD=10
      DO 420 KJ=1,2
      ISLH=1
      ISOH=1
      ISPL=1
      NLOOP=NLH(KJ)+NOH(KJ)+NPL(KJ)
      NCLH=0
      NCOH=0
      NCPL=0
      DO 400 L=1,NLOOP
      IF(NCLH.EQ.NLH(KJ)) ISLH=0
      IF(NCLH.EQ.NLH(KJ)) GO TO 6010
      NCLH=NCLH+1
      IEL2=NOLH(KJ,NCLH,1)
      GO TO 6011
 6010 IF(NCOH.EQ.NOH(KJ)) ISOH=0
      IF(NCOH.EQ.NOH(KJ)) GO TO 6020
      NCOH=NCOH+1
      IEL2=NOOH(KJ,NCOH,1)
      GO TO 6011
 6020 IF(NCPL.EQ.NPL(KJ)) ISPL=0
      IF(NCPL.EQ.NPL(KJ)) GO TO 400
      NCPL=NCPL+1
      IEL2=NOPL(KJ,NCPL,1)
```

```
 6011 CONTINUE
      DO 6030 KK=1,NELTOT
 6030 IF(IEL2.EQ.NELCON(KK,1)) IEL=KK
      H(KJ)=ELTHK(IEL)
      NPLY(KJ)=NELPLS(KJ,LYPN(IEL))
      DO 919 JJJ=1,50
      IPLY(JJJ,KJ)=NELPT(KJ,LYPN(IEL),JJJ)
  919 CONTINUE
      NRNK=5
      IF(NELTYP(IEL).EQ.2) NRNK=7
C
C     INTERNAL NUMBERING OF ELEMENT VERTICIES:
C
C                         3      2
C
C                         4      1
C
C
      SFX=(GCOORD(NELCNA(IEL,5),1)+GCOORD(NELCNA(IEL,2),1))/2.0D0
      SFY=(GCOORD(NELCNA(IEL,3),2)+GCOORD(NELCNA(IEL,2),2))/2.0D0
      DO 440 K=1,4
      XC(K)=GCOORD(NELCNA(IEL,6-K),1)-FSCD(IEL,1)
      IF(NELTYP(IEL).EQ.1) XC(K)=GCOORD(NELCNA(IEL,6-K),1)-SFX
      YC(K)=GCOORD(NELCNA(IEL,6-K),2)-FSCD(IEL,2)
      IF(NELTYP(IEL).EQ.1) YC(K)=GCOORD(NELCNA(IEL,6-K),2)-SFY
  440 CONTINUE
      XC(5)=XC(1)
      YC(5)=YC(1)
      AX=FSCD(IEL,3)
      IF(NELTYP(IEL).EQ.1) AX=0.1
      BX=AX
      PI=DARCOS(-1.0D0)
      RAD=PI/180.D0
      NGAUSS=2*NGP
      NGPT=4*NGAUSS**2
      IC=0
      NCPT=2*NGP
C
C     DETERMINE COORDINATES AT WHICH STRESSES AND
C     DISPLACEMENTS ARE TO BE COMPUTED.
C     ELEMENT NATURAL FLEXIBILITY MATRICIES
C     ARE COMPUTED BY INTEGRATING STRESSES
C     FOR EACH LOAD CASE IN THE NATURAL
C     MODE METHOD. THE ELEMENTS ARE DIVIDED
C     INTO FOUR REGIONS AND THE GAUSSIAN POINTS
C     ARE SCALED TO EACH REGION SIZE
C
C
C     REGION 1
C
      DO 15 II=1,NCPT
      DO 15 JJ=1,NCPT
      IC=IC+1
      XOUT(IC)=((-AX-XC(3))/2.)*GSSX(II)+(-AX+XC(3))/2.
      YOUT(IC)=((YC(3)-YC(4))/2.)*GSSX(JJ)+(YC(3)+YC(4))/2.
      WGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(3)-YC(4))*(-AX-XC(3))/4.D0
   15 CONTINUE
C
C     REGION 2
C
```

```
      DO 16 II=1,NCPT
      DO 16 JJ=1,NCPT
      IC=IC+1
      XOUT(IC)=AX*GSSX(II)
      YI=DSQRT(AX**2-XOUT(IC)**2)
      YOUT(IC)=((YC(2)-YI)/2.)*GSSX(JJ)+(YC(2)+YI)/2.
      WGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(2)-YI)*AX/2.0D0
   16 CONTINUE
C
C     REGION 3
C
      DO 17 II=1,NCPT
      DO 17 JJ=1,NCPT
      IC=IC+1
      XOUT(IC)=AX*GSSX(II)
      YI=-DSQRT(AX**2-XOUT(IC)**2)
      YOUT(IC)=((YI-YC(1))/2.)*GSSX(JJ)+(YI+YC(1))/2.
      WGHT(IC)=GSSW(II)*GSSW(JJ)*(YI-YC(1))*AX/2.0D0
   17 CONTINUE
C
C     REGION 4
C
      DO 18 II=1,NCPT
      DO 18 JJ=1,NCPT
      IC=IC+1
      XOUT(IC)=((XC(1)-AX)/2.)*GSSX(II)+(XC(1)+AX)/2.
      YOUT(IC)=((YC(2)-YC(1))/2.)*GSSX(JJ)+(YC(2)+YC(1))/2.
      WGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(2)-YC(1))*(XC(1)-AX)/4.0D0
   18 CONTINUE
      NINT=IC
      N=4*(NCPT**2)
C
C     ADD COORDINATES ALONG WHICH STRESSES WILL
C     BE AVERAGED
C
      ANT=AGNT(KJ)
      ABR=AGBR(KJ)
      ASO=AGSO(KJ)
      SG=1.0
      IF(LTNCM.EQ.2) SG=-1.0
      IF(KJ.EQ.2) SG=-SG
C
C     NET SECTION
C
      ANDO=ANT/FLOAT(NAVD)
      DO 21 II=1,NAVD
      IC=IC+1
      XOUT(IC)=0.0D0
      YOUT(IC)=BX+ANDO/2.+(II-1)*ANDO
   21 CONTINUE
C
C     SHEAROUT
C
      ANSO=ASO/FLOAT(NAVD)
      DO 31 II=1,NAVD
      IC=IC+1
      XOUT(IC)=SG*(BX+ANSO/2.+(II-1)*ANSO)
      YOUT(IC)=BX
   31 CONTINUE
C
```

```
C     BEARING
C
      ANBR=ABR/FLOAT(NAVD)
      DO 41 II=1,NAVD
      IC=IC+1
      XOUT(IC)=SGN*(AX+ANBR/2.+(II-1)*ANBR)
      YOUT(IC)=0.
   41 CONTINUE
C
C     ADD COORDINATES ALONG WHICH ELEMENT LOAD
C     RECOVERY WILL BE COMPUTED
C
      DO 3332 III=1,10
      IC=IC+1
      IF(KJ.EQ.1) XOUT(IC)=XC(3)+0.1*AX
      IF(KJ.EQ.2) XOUT(IC)=XC(1)-0.1*AX
      YOUT(IC)=((YC(2)-YC(1))/2.0D0)*OSSX(III)+(YC(2)+YC(1))/2.0D0
C
C     STRESSES ARE SINGULAR AT THETA = 180 DEG OR Y = 0
C
      IF(DABS(YOUT(IC)).LT.0.01) YOUT(IC)=YOUT(IC-1)
 3332 CONTINUE
 4891 CONTINUE
      NSTS=4*NAVD
      NOUT=4*(NGAUSS**2)
C
C     CALCULATION OF LOADED HOLE, UNLOADED HOLE, AND
C     PLAIN ELEMENT STIFFNESS MATRICIES
C
      THN=DARCOS(-1.0D0)/FLOAT(NOP)
      NN=IEL
      DO 410 J=1,NRNK
      NOPT4=5
      NT=7
      NCLL=10
      NB=52+4*NCLL
      HT=H(KJ)*NPLY(KJ)
      NCASE=J
      NTYPE=NELTYP(IEL)
      CALL MOEO(HT,W,AST,J,NN,KJ,NEL,NCLL)
      CALL MCIR(W,AST,NN,J,NCLL)
      PHI=0.0D0
      CALL AMATRX(H,PHI,KJ)
      CALL FIGEOM(H,PHI,KJ,NOPT4,NCLL)
      CALL INFLN(WGHT,H,NRNK,J,KJ,NN,NOPT)
  410 CONTINUE
 4000 CONTINUE
C
C     COMPUTE ELEMENT FAILURE VALUES BASED
C     ON MAXIMUM FIBER STRAIN ALLOWABLES
C
      HT=H(KJ)*NPLY(KJ)
      IF(NELTYP(IEL).EQ.2) CALL SMAX(HT,KJ,IEL)
      IF(NELTYP(IEL).EQ.3) CALL SMAX(HT,KJ,IEL)
      IF(ISLH.EQ.0) GO TO 6040
      NL=NUMLH(KJ,NCLH)
      IF(NL.EQ.1) GO TO 400
      DO 6041 K=2,NL
      DO 6042 LL=1,NELTOT
 6042 IF(NOLH(KJ,NCLH,K).EQ.NELCON(LL,1)) IEL2=LL
```

```
      DO 6043 ILM=1,10
      DO 6043 ILK=1,10
 6043 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK)
      DO 6044 KK=1,4
 6044 PSMX(IEL2,KK)=PSMX(IEL,KK)
      NNN=4*NAVD
      DO 6045 ILM=1,NNN
      DO 6045 ILK=1,10
 6045 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK)
 6041 CONTINUE
      GO TO 400
 6040 IF(ISOH.EQ.0) GO TO 6046
      NL=NUMOH(KJ,NCOH)
      IF(NL.EQ.1) GO TO 400
      DO 6047 K=2,NL
      DO 6048 LL=1,NELTOT
 6048 IF(NOOH(KJ,NCOH,K).EQ.NELCON(LL,1)) IEL2=LL
      DO 6049 ILM=1,10
      DO 6049 ILK=1,10
 6049 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK)
      DO 6050 KK=1,4
 6050 PSMX(IEL2,KK)=PSMX(IEL,KK)
      NNN=4*NAVD
      DO 6051 ILM=1,NNN
      DO 6051 ILK=1,10
 6051 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK)
 6047 CONTINUE
      GO TO 400
 6046 IF(ISPL.EQ.0) GO TO 400
      NL=NUMPL(KJ,NCPL)
      IF(NL.EQ.1) GO TO 400
      DO 6053 K=2,NL
      DO 6054 LL=1,NELTOT
 6054 IF(NOPL(KJ,NCPL,K).EQ.NELCON(LL,1)) IEL2=LL
      DO 6055 ILM=1,10
      DO 6055 ILK=1,10
 6055 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK)
      DO 6056 KK=1,4
 6056 PSMX(IEL2,KK)=PSMX(IEL,KK)
      NNN=4*NAVD
      DO 6057 ILM=1,NNN
      DO 6057 ILK=1,10
 6057 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK)
 6053 CONTINUE
  400 CONTINUE
  420 CONTINUE
C
C     DETERMINE ELEMENT ARRANGEMENT IN TOP
C     AND BOTTOM PLATES
C
      DO 681 KJ=1,2
      IF(KJ.EQ.2) GO TO 501
      L1=1
      L2=NOP1
      L3=1
      L4=NEL1
      GO TO 502
  501 L1=NOP1+1
      L2=NOTOT
      L3=NEL1+1
```

```
      L4=NELTOT
  502 CONTINUE
      AXMIN=1.D10
      AYMIN=1.D10
      DO 503 I=L1,L2
      IF(AXMIN.GT.GCOORD(I,1)) AXMIN=GCOORD(I,1)
      IF(AYMIN.GT.GCOORD(I,2)) AYMIN-GCOORD(I,2)
      IF(AXMIN.EQ.GCOORD(I,1).AND.AYMIN.EQ.UCOORD(I,2)) NC=I
  503 CONTINUE
      DO 574 I=L3,L4
  574 IF(NELCON(I,2).EQ.NORID(NC)) IEL=I
      NELORD(KJ,1,1)=IEL
      DO 504 I=1,25
      DO 505 J=1,25
      IEL=0
      DO 506 K=L3,L4
  506 IF(NELCON(K,2).EQ.NELCON(NELORD(KJ,J,I),3)) IEL=K
      IF(IEL.EQ.0) GO TO 507
      NELORD(KJ,J+1,I)=IEL
  505 CONTINUE
  507 CONTINUE
      IF(KJ.EQ.1) NROW1=J
      IF(KJ.EQ.2) NROW2=J
      IEL=0
      DO 308 L=L3,L4
  508 IF(NELCON(NELORD(KJ,1,I),5).EQ.NELCON(
     *L,2)) IEL=L
      IF(IEL.EQ.0) GO TO 509
      NELORD(KJ,1,I+1)=IEL
  504 CONTINUE
  509 CONTINUE
      IF(KJ.EQ.1) NCOL1=I
      IF(KJ.EQ.2) NCOL2=I
  681 CONTINUE
C
C     COMPUTE NODAL DEGREES OF FREEDOM
C
      IC=0
      DO 540 KJ=1,2
      IF(KJ.EQ.1) NR=NROW1
      IF(KJ.EQ.1) NC=NCOL1
      IF(KJ.EQ.2) NR=NROW2
      IF(KJ.EQ.2) NC=NCOL2
      NELDIS(NELORD(KJ,1,1),1,1)=IC+1
      NELDIS(NELORD(KJ,1,1),1,2)=IC+2
      NELDIS(NELORD(KJ,1,1),2,1)=IC+3
      NELDIS(NELORD(KJ,1,1),2,2)=IC+4
      IC=IC+4
      IF(NR.EQ.1) GO TO 549
      DO 541 I=2,NR
      NELDIS(NELORD(KJ,I,1),1,1)=NELDIS(NELORD(KJ,I-1
     *,1),2,1)
      NELDIS(NELORD(KJ,I,1),1,2)=NELDIS(NELORD(KJ,I-1
     *,1),2,2)
      NELDIS(NELORD(KJ,I,1),2,1)=IC+1
      NELDIS(NELORD(KJ,I,1),2,2)=IC+2
      IC=IC+2
  541 CONTINUE
  549 CONTINUE
      DO 542 I=1,NC
```

```
      DO 543 J=1,NR
      IF(I.EQ.1) GO TO 544
      NELDIS(NELORD(KJ,J,I),1,1)=NELDIS(NELORD(
     *KJ,J,I-1),4,1)
      NELDIS(NELORD(KJ,J,I),1,2)=NELDIS(NELORD(
     *KJ,J,I-1),4,2)
      NELDIS(NELORD(KJ,J,I),2,1)=NELDIS(NELORD(
     *KJ,J,I-1),3,1)
      NELDIS(NELORD(KJ,J,I),2,2)=NELDIS(NELORD(
     *KJ,J,I-1),3,2)
  544 CONTINUE
      IF(J.EQ.1) GO TO 561
      NELDIS(NELORD(KJ,J,I),4,1)=NELDIS(NELORD(KJ,J-1,I),3,1)
      NELDIS(NELORD(KJ,J,I),4,2)=NELDIS(NELORD(KJ,J-1,I),3,2)
      GO TO 562
  561 CONTINUE
      NELDIS(NELORD(KJ,J,I),4,1)=IC+1
      NELDIS(NELORD(KJ,J,I),4,2)=IC+2
      IC=IC+2
  562 CONTINUE
      IF(NELTYP(NELORD(KJ,J,I)).NE.2) GO TO 545
      NELDIS(NELORD(KJ,J,I),5,1)=IC+1
      NELDIS(NELORD(KJ,J,I),5,2)=IC+2
      IC=IC+2
  545 CONTINUE
      NELDIS(NELORD(KJ,J,I),3,1)=IC+1
      NELDIS(NELORD(KJ,J,I),3,2)=IC+2
      IC=IC+2
  543 CONTINUE
  542 CONTINUE
  540 CONTINUE
C
C     DETERMINE BOUNDARY NODES AND VALUES
C
      NRD=2*(NGP1+NGP2)
      DO 165 I=1,100
  165 PBC(I)=0.
C
C     DISTRIBUTE APPLIED LOAD
C
      ATOT=GCOORD(NELCNA(NELORD(1,NROW1,1),3),2)
     *-GCOORD(NELCNA(NELORD(1,1,1),2),2)
      APL=APP/ATOT
      SG=1.0
      IF(LTNCM.EQ.1) SG=-1.0
      DO 178 I=1,NROW1
      AI=GCOORD(NELCNA(NELORD(1,I,1),3),2)
     *-GCOORD(NELCNA(NELORD(1,I,1),2),2)
      M1=NELDIS(NELORD(1,I,1),1,1)
      M2=NELDIS(NELORD(1,I,1),2,1)
      PBC(M1)=PBC(M1)+SG*(0.5*DABS(APL*AI))
      PBC(M2)=PBC(M2)+SG*(0.5*DABS(APL*AI))
  178 CONTINUE
 1119 CONTINUE
C
C     ASSEMBLE GLOBAL STIFFNESS MATRIX
C
      DO 220 N1=1,NELTOT
      IR=5
      IF(NELTYP(N1).NE.2) IR=4
```

```
C
C     TOP AND BOTTOM PLATE LOADED HOLE AND
C     UNLOADED HOLE ELEMENTS
C
      IC1=0
      DO 425 N2=1,IR
      DO 425 N3=1,2
      M1=NELDIS(N1,N2,N3)
      IC1=IC1+1
      IC2=0
      DO 425 N4=1,IR
      DO 425 N5=1,2
      M2=NELDIS(N1,N4,N5)
      IC2=IC2+1
      GLSTFF(M1,M2)=GLSTFF(M1,M2)+ELSTFF(N1,IC1,IC2)
  425 CONTINUE
  220 CONTINUE
C
C     ADD EFFECTIVE FASTENER ELEMENTS
C
      DO 260 I=1,NUMF
      DO 1561 J=1,NEL1
 1561 IF(NELFAS(I,2).EQ.NELCON(J,6)) N=J
      N1=NELDIS(N,5,1)
      N2=NELDIS(N,5,2)
      NL=NEL1+1
      DO 1562 J=NL,NELTOT
 1562 IF(NELFAS(I,3).EQ.NELCON(J,6)) N=J
      N3=NELDIS(N,5,1)
      N4=NELDIS(N,5,2)
      GLSTFF(N1,N1)=GLSTFF(N1,N1)+RDSTFF(I,1)
      GLSTFF(N1,N3)=GLSTFF(N1,N3)-RDSTFF(I,1)
      GLSTFF(N2,N2)=GLSTFF(N2,N2)+RDSTFF(I,2)
      GLSTFF(N2,N4)=GLSTFF(N2,N4)-RDSTFF(I,2)
      GLSTFF(N3,N3)=GLSTFF(N3,N3)+RDSTFF(I,1)
      GLSTFF(N3,N1)=GLSTFF(N3,N1)-RDSTFF(I,1)
      GLSTFF(N4,N4)=GLSTFF(N4,N4)+RDSTFF(I,2)
      GLSTFF(N4,N2)=GLSTFF(N4,N2)-RDSTFF(I,2)
  260 CONTINUE
      NP=2*(NOP1+NOP2)
C
C     GLOBAL BOUNDARY CONDITIONS
C
      DO 415 I=1,NP
      RHS(I)=PBC(I)
  415 CONTINUE
C
      IC=1
      NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),4,1)
      DO 437 I=1,NROW2
      IC=IC+1
      NZERO(IC)=NELDIS(NELORD(2,I,NCOL2),5,1)
  437 CONTINUE
      IC=IC+1
      NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),4,2)
      NUMZ=IC
C
C     RESTORE REDUCED STIFFNESS MATRIX
C
      IC5=0
```

```
      DO 655 I=1,NP
      DO 665 K=1,NUMZ
      IF(I.EQ.NZERO(K)) GO TO 655
  665 CONTINUE
      ICR=ICR+1
      RHS(ICR)=RHS(I)
      ICC=0
      DO 670 J=1,NP
      DO 680 K=1,NUMZ
      IF(J.EQ.NZERO(K)) GO TO 670
  680 CONTINUE
      ICC=ICC+1
      ASQM(ICR,ICC)=OLSTFF(I,J)
  670 CONTINUE
  655 CONTINUE
      NP=NP-NUMZ
  685 CONTINUE
      DO 695 I=1,NP
      DO 695 J=1,NP
  695 OLSTFF(I,J)=ASQM(I,J)
C
C     APPLYING QUASSIAN ELIMINATION TO THE
C     MATRIX OF COEFFICIENTS
C
      DO 2001 I=1,NP
      IR=I
 2042 IF(DABS(ASQM(IR,I)).GT.1.0D-10) GO TO 2041
      IR=IR+1
      IF(IR.GT.NP) GO TO 2001
      GO TO 2042
 2041 NN=IR+1
      DO 2002 L=NN,NP
      IF(DABS(ASQM(L,I)).GT.1.D-10) GO TO 2009
      ASQM(L,I)=0.
      GO TO 2002
 2009 CF=-ASQM(IR,I)/ASQM(L,I)
      CF1=1.0D0
      IF(DABS(CF).GT.1.0) CF1=1.0D0/CF
      IF(DABS(CF).GT.1.0) CF=1.0D0
      DO 2003 J=I,NP
      ASQM(L,J)=ASQM(L,J)*CF+ASQM(IR,J)*CF1
      IF(DABS(ASQM(L,J)).LT.1.D-10) ASQM(L,J)=0.0
 2003 CONTINUE
      RHS(L)=RHS(L)*CF+RHS(I)*CF1
 2002 CONTINUE
 2001 CONTINUE
C
C
C     BACK SUBSTITUTION
C
      DO 2011 I=1,NP
      L=NP+1-I
      SUM=0.
      IF(ASQM(L,L).EQ.0.) GO TO 2112
      N=L+1
      IF(N.GT.NP) GO TO 2013
      DO 2013 J=N,NP
      SUM=SUM-ASQM(L,J)*ANR(J)
 2013 CONTINUE
      ANR(L)=(RHS(L)+SUM)/ASQM(L,L)
```

```
      GO TO 2011
 2112 CONTINUE
      ANR(L)=0.
 2011 CONTINUE
C
C     CALCULATE NODAL LOADS
C
      IC=0
      DO 44 I=1,NRD
      DO 54 J=1,NUMZ
      IF(I.NE.NZERO(J)) GO TO 54
      ANR2(I)=0.0D0
      GO TO 44
   54 CONTINUE
      IC=IC+1
      ANR2(I)=ANR(IC)
   44 CONTINUE
      WRITE(6,3712)
 3712 FORMAT(/,10X,'ELEMENT FORCES',//)
      DO 500 K=1,NELTOT
      NID=NELCON(K,1)
      WRITE(6,8947) NID
 8947 FORMAT(/,' ELEMENT ID',I8,//,
     *6X,'GRID',9X,'FX',9X,'FY',/)
      IR=5
      KL=K
      IF(KL.GT.NEL) KL=K-NEL
      IF(NELTYP(K).NE.2) IR=4
      DO 510 I=1,IR
      SUMU=0.
      SUMV=0.
      N=2*I-1
      DO 520 J=1,IR
      N1=NELDIS(K,J,1)
      N2=NELDIS(K,J,2)
      SUMU=SUMU+ELSTFF(K,N,(2*J-1))*ANR2(N1)+
     *ELSTFF(K,N,(2*J))*ANR2(N2)
  520 CONTINUE
      N=2*I
      DO 530 J=1,IR
      N1=NELDIS(K,J,1)
      N2=NELDIS(K,J,2)
      SUMV=SUMV+ELSTFF(K,N,(2*J-1))*ANR2(N1)+
     *ELSTFF(K,N,(2*J))*ANR2(N2)
  530 CONTINUE
C
C     STORE ELEMENT LOADS FOR CHECK ON ELEMENT
C     LOAD RECOVERY
C
      IF(K.LE.NEL1.AND.(I.EQ.1.OR.I.EQ.2)) ELLOAD(K,I)=SUMU
      IF(K.GT.NEL1.AND.(I.EQ.3.OR.I.EQ.4)) ELLOAD(K,I-2)=SUMU
      NID=NELCON(K,I+1)
      WRITE(6,3239) NID,SUMU,SUMV
 3239 FORMAT(2X,I8,5X,2(D9.3,2X))
  510 CONTINUE
  500 CONTINUE
C
C     COMPUTE ELEMENT FAILURE LOADS AND DETERMINE
C     CRITICAL ELEMENT TO CALCULATE JOINT FAILURE
C     LOAD
```

```
C
      CALL FCRIT(APP,NEL1,NEL2,NDAM,IN,LTNCM,NAVD)
      FAILV=DABS(ELFAIL(IN,NDAM))
      IF(NSDLS.EQ.2) FAILV=2.*FAILV
      NID=NELCON(IN,1)
      WRITE(6,5555) NID,FAILV
 5555 FORMAT(///,' FAILURE IS PREDICTED TO OCCUR IN ELEMENT',/,
     *' NUMBER',I5,' AT AN APPLIED JOINT LOAD VALUE ',/,
     *' OF ',D14.7,' LBS',/)
      IF(NDAM.EQ.1) WRITE(6,5556)
      IF(NDAM.EQ.2) WRITE(6,5557)
      IF(NDAM.EQ.3) WRITE(6,5558)
 5556 FORMAT(' THE PREDICTED FAILURE MODE IS NET SECTION')
 5557 FORMAT(' THE PREDICTED FAILURE MODE IS SHEAR-OUT ')
 5558 FORMAT(' THE PREDICTED FAILURE MODE IS BEARING')
      STOP
      END
C
C
C
      SUBROUTINE MOED(HT,W,AST,J,IN,KJ,NEL,NCL)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A1A(4),A2A(4)
      DIMENSION XB(200),YB(200),A1(200),A2(200)
      DIMENSION NELTYP(50),THTA(200)
      DIMENSION XC(5),YC(5)
      COMMON/CMT1/XB,YB,A1,A2,THTA
      COMMON/XCYC/XC,YC
      COMMON/NTP/NELTYP
C
C     DETERMINE EXTERIOR COLLOCATION POINTS AND
C     STRESS BOUNDARY CONDITIONS CORRESPONDING
C     TO THE NATURAL LOAD CASES
C
      NCS=5
      IF(NELTYP(IN).NE.2) NCS=3
      JK=0
      DO 15 I=1,4
      A1A(I)=0.
   15 A2A(I)=0.
      A=(YC(2)-YC(1))*HT
      B=(XC(2)-XC(3))*HT
      IF(J.EQ.1) A1A(3)=1.0D0/A
      IF(J.EQ.1.AND.NELTYP(IN).NE.2) A1A(1)=1.0D0/A
      IF(J.EQ.2) A1A(2)=1.0D0/B
      IF(J.EQ.2.AND.NELTYP(IN).NE.2) A1A(4)=1.0D0/B
      IF(J.GT.2.AND.NELTYP(IN).NE.2) GO TO 55
      IF(J.EQ.3) A1A(1)=1.0D0/A
      IF(J.EQ.4) A1A(4)=1.0/B
      IF(J.EQ.1.OR.J.EQ.3) AST=1.0D0/A
      IF(J.EQ.2.OR.J.EQ.4) AST=1.0D0/B
   55 CONTINUE
      W=XC(3)-XC(2)
      IF(J.EQ.1.OR.J.EQ.3.OR.J.EQ.6) W=YC(2)-YC(1)
      DO 10 I=1,4
      X=XC(I)-XC(I+1)
      Y=YC(I+1)-YC(I)
      IF(X.EQ.0.) X=1.D-6
      IF(Y.EQ.0.) Y=1.D-6
```

```
      TH=DATAN2(X,Y)
      TH=TH*180./DARCOS(-0.1D1)
      DX=(XC(I+1)-XC(I))/(NCL+1)
      DY=(YC(I+1)-YC(I))/(NCL+1)
      DO 20 II=1,NCL
      JK=JK+1
      IF(I.EQ.1.OR.I.EQ.3) GO TO 23
      YB(JK)=YC(I)
      XB(JK)=XC(I)+DX*(II+.5)
      IF(II.EQ.1) XB(JK)=XC(I)+(DX/2.)
      GO TO 24
   23 YB(JK)=YC(I)+DY*(II+.5)
      IF(II.EQ.1) YB(JK)=YC(I)+(DY/2.)
      XB(JK)=XC(I)
   24 THTA(JK)=TH
      A1(JK)=A1A(I)
      A2(JK)=A2A(I)
      IF(J.EQ.NCS.AND.(I.EQ.1.OR.I.EQ.3)) A1(JK)=(2*YB(JK)/W)*(3.0D0/A)
      IF(J.EQ.(NCS+1).AND.(I.EQ.2.OR.I.EQ.4)) A1(JK)=(2*XB(JK)/W)*(3.0
     *D0/B)
      IF(J.EQ.(NCS+2).AND.(I.EQ.1.OR.I.EQ.3)) A2(JK)=2.0/DSQRT(A**2+B**2
     *)
      IF(J.EQ.(NCS+2).AND.(I.EQ.2.OR.I.EQ.4)) A2(JK)=-2.D0/DSQRT(A**2+B*
     **2)
20    CONTINUE
10    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE MCIR(W,AST,I,J,NCL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XB(200),YB(200),A1(200),A2(200)
      DIMENSION THTA(200),NELTYP(50)
      COMMON/CNT1/XB,YB,A1,A2,THTA
      COMMON/NTP/NELTYP
      COMMON/ELP/AX,BX
      CON=-1.0D0
      RAD=DARCOS(CON)/180.
      BSTR=DABS((2.*W*AST)/(DARCOS(CON)*AX))
      IF(NELTYP(I).NE.2) BSTR=0.0D0
      IF(NELTYP(I).EQ.2.AND.J.GT.5) BSTR=0.0D0
      IC=4*NCL
C
C     DETERMINE INTERIOR COLLOCATION POINTS AND
C     STRESS BOUNDARY CONDITIONS
C
      NBD=52
      NBI=NBD/4
      DO 10 K=1,4
      CR=(K-1)*DARCOS(CON)/2.
      DO 20 KI=1,NBI
      IC=IC+1
      THINC=(DARCOS(CON)/2.)/FLOAT(NBI)
      A1(IC)=0.
      A2(IC)=0.
      THINC2=THINC/2.
      TH=THINC2+(KI-1)*THINC+CR
      XB(IC)=AX*DCOS(TH)
```

```
      YB(IC)=AX*DSIN(TH)
      THTA(IC)=TH/RAD
      IF(J.GT.4) GO TO 20
      IF(NELTYP(I).NE.2) GO TO 40
      IF(J.EQ.1.AND.(K.EQ.1.OR.K.EQ.4)) A1(IC)=
     *-BSTR*DABS(DCOS(TH))
      IF(J.EQ.2.AND.(K.EQ.3.OR.K.EQ.4)) A1(IC)=
     *-BSTR*DABS(DSIN(TH))
      IF(J.EQ.3.AND.(K.EQ.2.OR.K.EQ.3)) A1(IC)=
     *-BSTR*DABS(DCOS(TH))
      IF(J.EQ.4.AND.(K.EQ.1.OR.K.EQ.2)) A1(IC)=
     *-BSTR*DABS(DSIN(TH))
   40 CONTINUE
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE INFLN(WGHT,H,NRNK,J,KJ,I,NOPT)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ELSTFF(50,10,10),WGHT(500),WK(150)
      DIMENSION ELSTSS(50,50,10),STSV(50),STSA(50,10)
      DIMENSION AN(10,7),UVOUT(20)
      DIMENSION PHI(3,7,400),STEMP(10,10),AO(10,3)
      DIMENSION FINF(10,10),SINF(10,10),AINV(3,3)
      DIMENSION APSX(500),APSY(500),APSXY(500)
      DIMENSION H(2),XC(5),YC(5),NPLY(2)
      DIMENSION IC(10)
      DIMENSION A(10,10),ATEMP2(10,10)
      COMMON/UV/UVOUT
      COMMON/XCYC/XC,YC
      COMMON/ELP/AX,BX,NOUT,NSTS
      COMMON/ELS/ELSTFF,ELSTSS
      COMMON/STS/STSV
      COMMON/INF1/APSX,APSY,APSXY
      COMMON/LYP/NPLY
      COMMON/INV/AINV
C
C     COMPUTE ELEMENT STIFFNESS COEFFICIENTS
C
      IF(J.GT.1) GO TO 200
      DO 7 III=1,10
    7 IC(III)=III
      DO 100 IN1=1,3
      DO 100 IN2=1,7
      DO 100 IN3=1,400
  100 PHI(IN1,IN2,IN3)=0.0D0
      DO 444 N1=1,10
      AO(N1,1)=0.0D0
      AO(N1,2)=0.0D0
      AO(N1,3)=0.0D0
      DO 444 N2=1,7
  444 AN(N1,N2)=0.0D0
      DO 110 IN1=1,10
      DO 110 IN2=1,10
      SINF(IN1,IN2)=0.0D0
      STEMP(IN1,IN2)=0.0D0
```

```
  110 FINF(IN1,IN2)=0.0D0
  200 CONTINUE
C
C     STRESSES AND DISPLACEMENTS ARE STORED
C     FOR EACH LOAD CASE
C
      DO 2107 KLK=1,8
 2107 AN(KLK,J)=UVOUT(KLK)
      IF(NRNK.EQ.5) GO TO 2221
      IF(J.EQ.1) UVOUT(5)=UVOUT(9)
      IF(J.EQ.1) UVOUT(7)=UVOUT(9)
      IF(J.EQ.2) UVOUT(2)=UVOUT(16)
      IF(J.EQ.2) UVOUT(8)=UVOUT(16)
      IF(J.EQ.3) UVOUT(1)=UVOUT(13)
      IF(J.EQ.3) UVOUT(3)=UVOUT(16)
      IF(J.EQ.4) UVOUT(4)=UVOUT(12)
      IF(J.EQ.4) UVOUT(6)=UVOUT(12)
 2221 CONTINUE
      IF(NRNK.EQ.7.AND.J.LT.5) GO TO 371
      AN(9,J)=(UVOUT(9)+UVOUT(13))/2.
      AN(10,J)=(UVOUT(12)+UVOUT(16))/2.
      GO TO 372
  371 AN(9,J)=UVOUT(7+2*J)
      AN(10,J)=UVOUT(8+2*J)
  372 CONTINUE
      DO 15 IS=1,NSTS
   15 STSA(IS,J)=STSV(IS)
      DO 10 IS=1,NGPT
      PHI(1,J,IS)=APSX(IS)
   10 CONTINUE
      DO 20 IS=1,NGPT
      PHI(2,J,IS)=APSY(IS)
   20 CONTINUE
      DO 30 IS=1,NGPT
      PHI(3,J,IS)=APSXY(IS)
   30 CONTINUE
      IF(J.LT.NRNK) RETURN
C
C     INTEGRATION OF STRESSES
C
      DO 1010 III=1,10
 1010 CONTINUE
      NTR=NRNK+3
      DO 45 IK=1,NTR
      DO 45 JK=1,NTR
   45 FINF(IK,JK)=0.
      H1=H(KJ)*NPLY(KJ)
      DO 50 LI=1,NGPT
      DO 60 LJ=1,3
      DO 60 KI=1,NRNK
      SUM=0.
      DO 70 IL=1,3
      SUM=SUM+H1*AINV(LJ,IL)*PHI(IL,KI,LI)
   70 CONTINUE
      STEMP(LJ,KI)=SUM
   60 CONTINUE
      DO 80 LK=1,NRNK
      DO 80 LJ=1,NRNK
      SUM=0.
      DO 90 IL=1,3
```

```
      SUM=SUM+PHI(IL,LK,LI)*STEMP(IL,LJ)
   90 CONTINUE
      FINF(LK,LJ)=FINF(LK,LJ)+SUM*WGHT(LI)
   80 CONTINUE
   50 CONTINUE
      DO 51 III=1,NRNK
      DO 51 JJJ=1,NRNK
   51 STEMP(III,JJJ)=(FINF(III,JJJ)+FINF(JJJ,III))/2.0D0
      DO 52 III=1,NRNK
      DO 52 JJJ=1,NRNK
   52 FINF(III,JJJ)=STEMP(III,JJJ)
      CALL LINV2F(FINF,NRNK,10,SINF,4,WK,IER)
      DO 410 IA=1,NTR
      A0(IA,1)=0.5+0.5*(-1.0)**(IA+1)
      A0(IA,2)=0.5+0.5*(-1.0)**(IA)
  410 A0(IA,3)=0.0D0
      A0(1,3)=DABS(YC(4))
      A0(2,3)=-DABS(XC(4))
      A0(3,3)=-DABS(YC(3))
      A0(4,3)=-DABS(XC(3))
      A0(5,3)=-DABS(YC(2))
      A0(6,3)=DABS(XC(2))
      A0(7,3)=DABS(YC(1))
      A0(8,3)=DABS(XC(1))
      DO 420 KK=1,NTR
      DO 420 LL=1,NRNK
      SUM=0.0D0
      DO 430 JJ=1,NRNK
  430 SUM=SUM+AN(KK,JJ)*SINF(JJ,LL)
  420 STEMP(KK,LL)=SUM
      DO 440 KK=1,NTR
      N1=NTR-2
      DO 440 JJ=N1,NTR
  440 STEMP(KK,JJ)=A0(KK,JJ-NRNK)
      CALL LINV2F(STEMP,NTR,10,FINF,4,WK,IER)
      DO 450 II=1,NRNK
      DO 450 JJ=1,NTR
      SUM=0.0D0
      DO 460 KK=1,NRNK
  460 SUM=SUM+SINF(II,KK)*FINF(KK,JJ)
  450 STEMP(II,JJ)=SUM
      DO 470 II=1,NTR
      DO 470 JJ=1,NTR
      SUM=0.0D0
      DO 480 KK=1,NRNK
  480 SUM=SUM+FINF(KK,II)*STEMP(KK,JJ)
      ELSTFF(I,II,JJ)=SUM
      A(II,JJ)=SUM
  470 CONTINUE
      DO 550 II=1,NRNK
      DO 550 JJ=1,NTR
      SUM=0.0D0
      DO 560 KK=1,NRNK
  560 SUM=SUM+SINF(II,KK)*FINF(KK,JJ)
  550 STEMP(II,JJ)=SUM
      DO 570 II=1,NSTS
      DO 570 JJ=1,NTR
      SUM=0.0D0
      DO 580 KK=1,NRNK
  580 SUM=SUM+[illegible]A(II,KK)*STEMP(KK,JJ)
```

```
  570 ELSTSS(I,II,JJ)=SUM
      RETURN
      END
C
C
C
      SUBROUTINE SMAX(NT,KJ,I)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AINV(3,3),STULT(3,2),AVN(3)
      DIMENSION NV(3)
      DIMENSION PSMX(50,4),STM(3),CM(2)
      DIMENSION NPLY(2),NUMPLY(2),ANG(5,2),IPLY(100,2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2)
      COMMON/MOD/E1,E2,G12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/STMT/STM,CM
      COMMON/INV/AINV
      COMMON/SMX/PSMX
      COMMON/STN/STULT
      DATA CMC/'C'/
      IF(CM(KJ).EQ.CMC) GO TO 222
      PSMX(I,1)=STM(1)
      PSMX(I,2)=STM(2)
      PSMX(I,3)=STM(3)
      PSMX(I,4)=STM(2)
      RETURN
  222 CONTINUE
C
C     COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM
C     FIBER STRAINS FOR EACH FAILURE MODE
C
      DO 100 K=1,3
      DO 10 II=1,3
      NV(II)=0
   10 AVN(II)=0.0D0
      IF(K.EQ.1) NV(1)=1
      IF(K.EQ.2) NV(1)=-1
      IF(K.EQ.3) NV(3)=1
      DO 15 II=1,3
      DO 15 JJ=1,3
      AVN(II)=AVN(II)+AINV(II,JJ)*NV(JJ)
   15 CONTINUE
      NP=NUMPLY(KJ)
      SMX=0.0D0
      RAD=DARCOS(-1.0D0)/180.D0
      DO 25 II=1,NP
      TH=ANG(II,KJ)*RAD
      E11=DCOS(TH)**2*AVN(1)+AVN(2)*DSIN(TH)**2+
     *DCOS(TH)*DSIN(TH)*AVN(3)
      IF(K.NE.1) GO TO 65
      EPRT=E11/STULT(2,KJ)
      GO TO 50
   65 IF(K.NE.2) GO TO 75
      EPRT=E11/STULT(1,KJ)
      GO TO 50
   75 EPRT=E11/STULY(2,KJ)
   50 CONTINUE
      IF(DABS(SMX).LT.DABS(EPRT)) SMX=EPRT
   25 CONTINUE
      IF(DABS(SMX).GT.1.0D-10) GO TO 555
```

```
      PSMX(I,K)=STULT(5,KJ)*G12(KJ)
      GO TO 100
  555 CONTINUE
      PSMX(I,K)=DABS(1.0D0/SMX)
  100 CONTINUE
      PSMX(I,4)=PSMX(1,2)
      RETURN
      END
C
C
C
      SUBROUTINE POLY(W,AST,J,K,NCOL,LTNCM)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XC(5),YC(5),A1(200),A2(200),XB(200)
      DIMENSION YB(200),T(200),A1A(4),A2A(4)
      COMMON/CMT1/XB,YB,A1,A2,T
      COMMON/XCYC/XC,YC
C
C     ARRAY COLLOCATION POINTS AROUND EXTERIOR
C     BOUNDARY AND APPLY STRESS BOUNDARY
C     CONDITION
C
      DO 120 I=1,4
      A1A(I)=0.0
      A2A(I)=0.0
  120 CONTINUE
      IF(LTNCM.EQ.1) A1A(1)=AST
      IF(LTNCM.EQ.2) A1A(1)=-AST
      J=0
      XC(5)=XC(1)
      YC(5)=YC(1)
      DO 10 I=1,4
      X=XC(I)-XC(I+1)
      Y=YC(I+1)-YC(I)
      IF(X.EQ.0.) X=1.D-6
      IF(Y.EQ.0.) Y=1.D-6
      TH=DATAN2(X,Y)
      TH=TH*180./DARCOS(-0.1D1)
      DX=(XC(I+1)-XC(I))/(NCOL+1)
      DY=(YC(I+1)-YC(I))/(NCOL+1)
      DO 20 II=1,NCOL
      J=J+1
      IF(I.EQ.1.OR.I.EQ.3) GO TO 23
      YB(J)=YC(I)
      XB(J)=XC(I)+DX*(II+.5)
      IF(II.EQ.1) XB(J)=XC(I)+(DX/2.)
      GO TO 24
   23 CONTINUE
      YB(J)=YC(I)+DY*(II+.5)
      IF(II.EQ.1) YB(J)=YC(I)+(DY/2.)
      XB(J)=XC(I)
   24 T(J)=TH
      A1(J)=A1A(I)
      A2(J)=A2A(I)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
C
C
C
      SUBROUTINE CIRC(H,AST,JK,K,LTNCM)
C
C     ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY AND
C     APPLY BEARING LOAD IN A COSINUSOIDAL DISTRIBUTION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(600),Y(600),THTA(200),A1(200),A2(200)
      DIMENSION XB(200),YB(200)
      COMMON/FB1/BSTR,XSTR
      COMMON/CMT1/XB,YB,A1,A2,THTA
      COMMON/CMT2/X,Y
      COMMON/ELP/A,B,N
      CON=-1.0
      XSTR=AST
      BSTR=(2.*H*XSTR)/(DARCOS(CON)*B)
      NM4=N-4
      NQ=NM4/4
      DO 20 I=1,N
      JK=JK+1
      TH=((I-1)*2+1)*DARCOS(CON)/N
      X(I)=A*DCOS(TH)
      Y(I)=B*DSIN(TH)
      CS=-X(I)*B*B/(Y(I)*A*A)
      IF(Y(I).GT.0)THTA(JK)=DATAN(CS)-DARCOS(CON)/2.
      IF(Y(I).LT.0)THTA(JK)=DATAN(CS)+DARCOS(CON)/2.
      THTA(JK)=THTA(JK)*180./DARCOS(CON)
      IF(LTNCM.EQ.2) GO TO 25
      IF(I.GT.(NQ+1).AND.I.LT.(N-NQ)) GO TO 204
      GO TO 30
   25 IF(I.LE.(NQ+2).OR.I.GE.(N-NQ-1)) GO TO 204
   30 CONTINUE
      A1(JK)=0.
      A2(JK)=0.
      XB(JK)=X(I)
      YB(JK)=Y(I)
      GO TO 20
  204 IF(Y(I).GT.0.) TETA=DARCOS(-1.)-DATAN(Y(I)/X(I))
      IF(Y(I).LT.0.) TETA=DARCOS(-1.)+DATAN(Y(I)/X(I))
      [illegible](JK)=-1[illegible]*BSTR[illegible]COS(TETA)
      A[illegible](JK)=0.
      XB(JK)=X(I)
      YB(JK)=Y(I)
   20 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FIGEOM(H,PHS,NJ,NOPT4,NCLL)
C
C
C
C
C     [illegible] FINITE GEOMETRY ANALYSIS BY A
C     [illegible] COLLOCATION TECHNIQUE
C
C
C
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),WK(25),AI(3,3),AZ(5),WKK(121),BC(300)
      DIMENSION CH(4),H(2)
      COMPLEX*16 ORHS(100)
      COMPLEX*16 CM(300,90),CMC(300,90),CMCTCM(90,90),RHS(90)
      COMMON/ROOTS/R1,R2
      COMMON/AMT/A
      COMMON/TERMS/P1,Q1,P2,Q2
      COMMON/ELP/AX,BX,NOUT,NSTS
      COMMON/SER/NT,NB
      COMMON/INV/AI
      COMPLEX*16 Z(4),Z1,Z2,Q1,Q2,P1,P2,R1,R2,WA(14883)
C
C     AMATRX CALCULATES THE LAMINATE 'A' MATRIX
C
      CALL AMATRX(H,PHS,KJ)
      N=3
      IDGT=4
      IA=3
C
C     LINV2F INVERTS THE 'A' MATRIX
C
      CALL LINV2F(A,N,IA,AI,IDGT,WK,IER)
      NDEG=4
      AZ(1)=AI(1,1)/AI(2,2)
      AZ(2)=-2.*AI(1,3)/AI(2,2)
      AZ(3)=(2.*AI(1,2)+AI(3,3))/AI(2,2)
      AZ(4)=-2.*AI(2,3)/AI(2,2)
      AZ(5)=1.0D0
C
C     ZRPOLY FINDS THE ROOTS OF THE CHARACTERISTIC EQUATION
C
      CALL ZRPOLY(AZ,NDEG,Z,IER)
C
C     Z(2) AND Z(4) ARE THE COMPLEX CONJUGATES OF Z(1)
C     AND Z(3) RESPECTIVELY
C
      R1=Z(1)
      R2=Z(3)
C
C     THE TWO ROOTS MUST BE CHECKED FOR A UNITARY COMPONENT
C     IN EITHER THE REAL OR IMAGINARY PART; SUCH AN
C     OCCURANCE SIGNIFIES A QUASI-ISOTROPIC LAYUP AND
C     THE VALUE MUST BE PERTURBED SLIGHTLY IN ORDER TO
C     AVOID A SINGULAR MATRIX
C
      CH(1)=R1
      CH(2)=(0.0,-1.0)*R1
      CH(3)=R2
      CH(4)=(0.0,-1.0)*R2
      DO 30 IJK=1,4
      AR=DABS(CH(IJK))
      IF(AR.LE 1.0) GO TO 31
      GO TO 32
   31 IF((1.0-AR).LT.0.02) CH(IJK)=0.98
      GO TO 30
   32 IF((AR-1.0).LT.0.02) CH(IJK)=1.02
   30 CONTINUE
      R1=DCMPLX(CH(1),CH(2))
      R2=DCMPLX(CH(3),CH(4))
```

```
C
C     CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS
C
      P1=AI(1,1)*R1**2+AI(1,2)-AI(1,3)*R1
      P2=AI(1,1)*R2**2+AI(1,2)-AI(1,3)*R2
      Q1=AI(2,2)/R1+AI(1,2)*R1-AI(2,3)
      Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)
C
C     INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS
C
      NT4=4*NT
      NT8=8*NT
      NT8P4=8*NT+4
      NT8P2=8*NT+2
      NT8P1=8*NT+1
      NB2=2*NB
      NWK=NT8P1*(NT8P1+2)
      CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,
     1NT8P1,NB2,NWK,WA,WKK,NOPT4,KJ,NCLL)
      RETURN
      END
C
C
C
      SUBROUTINE AMATRX(H,PHS,K)
C
C     ASSEMBLE THE A MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),ANO(5,2),H(2),NPLY(2),NUMPLY(2)
      DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2)
      DIMENSION IPLY(100,2)
      COMMON/MOD/E1,E2,G12,V12,V21
      COMMON/AMT/A
      COMMON/LYP/NPLY,NUMPLY,ANO,IPLY
      THKNES=NPLY(K)*H(K)
      DENO=1.-E2(K)*V12(K)**2/E1(K)
      Q11=E1(K)/DENO
      Q22=E2(K)/DENO
      Q12=V12(K)*Q22
      Q21=Q12
      Q33=G12(K)
      DO 10 I=1,3
      DO 10 J=1,3
   10 A(I,J)=0.0D0
      NN=NPLY(K)
      T=H(K)
      DO 20 I=1,NN
      LP=IPLY(I,K)
      THTAI=(ANO(LP,K)+PHS)*DARCOS(-1.D0)/180.D0
      C=DCOS(THTAI)
      S=DSIN(THTAI)
      A(1,1)=(Q11*C**4+2.*(Q12+2.*Q33)*C*C*S*S+Q22*S**4)*T+A(1,1)
      A(2,2)=(Q11*S**4+2.*(Q12+2.*Q33)*C*C*S*S+Q22*C**4)*T+A(2,2)
      A(1,2)=((Q11+Q22-4.*Q33)*C*C*S*S+Q12*(C**4+S**4))*T+A(1,2)
      A(2,1)=A(1,2)
      A(3,3)=((Q11+Q22-2.*Q12-2.*Q33)*C*C*S*S+Q33*(C**4+S**4))*T+A(3,3)
      A(1,3)=((Q11-Q12-2.*Q33)*C**3*S+(Q12-Q22+2.*Q33)*S**3*C)*T+A(1,3)
      A(2,3)=((Q11-Q12-2.*Q33)*S**3*C+(Q12-Q22+2.*Q33)*C**3*S)*T+A(2,3)
      A(3,2)=A(2,3)
```

```
      A(3,1)=A(1,3)
   20 CONTINUE
      DO 53 I=1,3
      DO 53 J=1,3
      A(I,J)=A(I,J)/THKNES
   53 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CMAT(BC,CMCTCM,CMC,CM,RHS,ORHS,NT4,NT8,NT8P4,NT8P2,
     1NT8P1,NB2,NWK,WA,WKK,NOPT4,KJ,NCOL)
C
C
C     CMAT OUTPUTS STRESSES, STRAINS, AND DISPLACEMENTS
C     AT SPECIFIED COORDINATES
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASX(400),ASXY(400),UVOUT(20)
      DIMENSION XC(5),YC(5)
      DIMENSION  THTA(200),X(200),Y(200),AMAT(3,3)
      DIMENSION  AIN1(200),AIN2(200),BC(NB2)
      DIMENSION WKK(NT8P1),WORK(700)
      DIMENSION XOUT(600),YOUT(600),STSV(50)
      DIMENSION FUR(400),FTHT(400),FSMR(400)
      DIMENSION RTHT(400),REPX(400),REPY(400),REPXY(400)
      DIMENSION APSX(500),APSY(500),APSXY(500)
      COMPLEX*16 CMCTCM(NT8P1,NT8P1),RHS(NT8P1),PHI1D,PHI2D,XETA1,XETA2
      COMPLEX*16 ACD(25,25),ACD2(25,25),RHS2(25)
      COMPLEX*16 U0,V0
      COMPLEX*16 CM(NB2,NT8P4),CMC(NB2,NT8P1),Z1,Z2,Z11,Z22,R1,R2
      COMPLEX*16 T11,T12,T21,T22,P11,P12,P21,P22
      COMPLEX*16 P1,P2,Q1,Q2,DCMPLX,CO,CSUM,ORHS(NT8P2)
      COMPLEX*16 PHI1DP,PHI2DP,PHI1DN,PHI2DN
      COMPLEX*16 PHI1P,PHI2P,PHI1N,PHI2N,PHI1,PHI2
      COMPLEX*16 PHI3N,PHI3P,PHI3,PHI4N,PHI4P,PHI4
      COMPLEX*16 SV11,SV12,SV21,SV22,RB11,RB21,RB11B,RB21B
      COMPLEX*16 R1B,R2B,P1B,P2B,Q1B,Q2B,WA(NWK)
      COMMON/INF1/APSX,APSY,APSXY
      COMMON/XCYC/XC,YC
      COMMON/NCST/NCASE,NTYPE
      COMMON/XXY1/ASX,ASXY
      COMMON/STS/STSV
      COMMON/UV/UVOUT
      COMMON/ROOTS/R1,R2
      COMMON/TERMS/P1,Q1,P2,Q2
      COMMON/CMT1/X,Y,AIN1,AIN2,THTA
      COMMON/CMT2/XOUT,YOUT
      COMMON/FB2/FUR,FTHT,FSMR
      COMMON/QMT/RTHT,REPX,REPY,REPXY
      COMMON/ELP/AX,BX,NOUT,NSTS
      COMMON/SER/NT,NB
      COMMON/INV/AMAT
      IF(NOPT4.EQ.5.AND.NCASE.GT.1) GO TO 3335
      DO 6666 III=1,NT8P1
      DO 6666 IHH=1,NT8P1
 6666 CMCTCM(III,IHH)=(0.0D0,0.0D0)
      A=AX
```

```
      B=BX
      CO=(0.0,1.0)
      RB11=(Q1-P1*R1)/(A-CO*R1*B)
      RB21=(Q2-P2*R2)/(A-CO*R2*B)
      REALR1=R1
      REALR2=R2
      REALP1=P1
      REALP2=P2
      REALQ1=Q1
      REALQ2=Q2
      RRB11=RB11
      RRB21=RB21
      AIMGR1=CO*R1
      AIMOR2=CO*R2
      AIMGP1=CO*P1
      AIMGP2=CO*P2
      AIMOQ1=CO*Q1
      AIMGQ2=CO*Q2
      ARB11=CO*RB11
      ARB21=CO*RB21
      R1B=DCMPLX(REALR1,AIMGR1)
      R2B=DCMPLX(REALR2,AIMOR2)
      P1B=DCMPLX(REALP1,AIMOP1)
      P2B=DCMPLX(REALP2,AIMOP2)
      Q1B=DCMPLX(REALQ1,AIMGQ1)
      Q2B=DCMPLX(REALQ2,AIMOQ2)
      RB11B=DCMPLX(RRB11,ARB11)
      RB21B=DCMPLX(RRB21,ARB21)
      JJJ=0
      DO 1000 I=1,NB
      J=I*2
      THTAI=THTA(I)*DARCOS(-1.0D0)/180.D0
      C=DCOS(THTAI)
      S=DSIN(THTAI)
      P11=C*P1+S*Q1
      P12=C*P2+S*Q2
      P21=-S*P1+C*Q1
      P22=-S*P2+C*Q2
      T11=(C*C*R1*R1+S*S-2.*C*S*R1)
      T12=(C*C*R2*R2+S*S-2.*C*S*R2)
      T21=(-C*S*R1*R1+C*S-(C*C-S*S)*R1)
      T22=(-C*S*R2*R2+C*S-(C*C-S*S)*R2)
      Z1=X(I)+R1*Y(I)
      Z2=X(I)+R2*Y(I)
      Z11=CDSQRT(Z1*Z1-A*A-R1*R1*B*B)
      Z22=CDSQRT(Z2*Z2-A*A-R2*R2*B*B)
      REAL1=Z11
      AIMG1=-CO*Z11
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMO1).LE.1.D-16)AIMO1=0.0D0
      Z11=DCMPLX(REAL1,AIMO1)
      REAL2=Z22
      AIMO2=-CO*Z22
      IF(DABS(REAL2).LE.1.D-16)REAL2=0.0D0
      IF(DABS(AIMO2).LE.1.D-16)AIMO2=0.0D0
      Z22=DCMPLX(REAL2,AIMO2)
      XETA1=(Z1+Z11)/(A-CO*R1*B)
      IF(CDABS(XETA1).LT.0.999) GO TO 300
      GO TO 310
  300 Z11=-Z11
```

```
      XETA1=(Z1+Z11)/(A-CO*R1*B)
  310 XETA2=(Z2+Z22)/(A-CO*R2*B)
      IF(CDABS(XETA2).LT.0.999) GO TO 320
      GO TO 330
  320 Z22=-Z22
      XETA2=(Z2+Z22)/(A-CO*R2*B)
  330 CONTINUE
      JJJ=JJJ+1
C
C     NORMAL & TANGENTIAL STRESS BOUNDARY CONDITIONS ARE IMPOSED
C
      DO 5 N=1,NT
      NP=N
      CM(J-1,N)=NP*XETA1**NP*T11/Z11
      CM(J-1,2*NT+N)=NP*XCTA2**NP*T12/Z22
      CM(J,N)=NP*XETA1**NP*T21/Z11
      CM(J,2*NT+N)=NP*XETA2**NP*T22/Z22
      NN=-N
      CM(J-1,NT+N)=NN*XETA1**NN*T11/Z11
      CM(J-1,3*NT+N)=NN*XETA2**NN*T12/Z22
      CM(J,NT+N)=NN*XETA1**NN*T21/Z11
      CM(J,3*NT+N)=NN*XETA2**NN*T22/Z22
    5 CONTINUE
      CM(J-1,NT8+1)=T11/Z11
      CM(J-1,NT8+2)=T12/Z22
      CM(J,NT8+1)=T21/Z11
      CM(J,NT8+2)=T22/Z22
 1000 CONTINUE
      DO 195 I=1,NB2
      DO 196 J=1,NT4
      REAL1=CM(I,J)
      AIMG1=-CO*CM(I,J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
      CM(I,J)=DCMPLX(REAL1,AIMG1)
      AIMG2=-AIMG1
      CM(I,NT4+J)=DCMPLX(REAL1,AIMG2)
  196 CONTINUE
  195 CONTINUE
      DO 295 I=1,NB2
      DO 296 J=1,2
      REAL1=CM(I,NT8+J)
      AIMG1=-CO*CM(I,NT8+J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
      CM(I,NT8+J)=DCMPLX(REAL1,AIMG1)
      AIMG2=-AIMG1
      CM(I,NT8+2+J)=DCMPLX(REAL1,AIMG2)
  296 CONTINUE
  295 CONTINUE
      SV11=(P2*Q1B-Q2*P1B)/(Q1*P2-Q2*P1)
      SV12=(P2*Q2B-Q2*P2B)/(Q1*P2-Q2*P1)
      SV21=(Q1*P1B-Q1B*P1)/(Q1*P2-Q2*P1)
      SV22=(Q1*P2B-Q2B*P1)/(Q1*P2-Q2*P1)
      DO 139 I=1,NB2
C
C     IMPOSE RIGID BODY ROTATION CONDITION
C
      CM(I,2*NT+1)=-CM(I,1)*RB21/RB11+CM(I,2*NT+1)
      CM(I,4*NT+1)=-CM(I,1)*RB11B/RB11+CM(I,4*NT+1)
```

```
      CM(I,6*NT+1)=-CM(I,1)*RB21B/RB11+CM(I,6*NT+1)
      CM(I,1)=(0.0,0.0)
C
C     IMPOSE SINGLE-VALUEDNESS CONDITION
C
      CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+1)*SV12+CM(I,NT8+4)
      CM(I,NT8+3)=CM(I,NT8+2)*SV21+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+2)*SV22+CM(I,NT8+4)
      CM(I,NT8+1)=(0.0,0.0)
      CM(I,NT8+2)=(0.0,0.0)
  139 CONTINUE
      DO 141 I=1,NB2
      DO 142 J=2,NT8
  142 CM(I,J-1)=CM(I,J)
      CM(I,NT8)=CM(I,NT8+3)
      CM(I,NT8+1)=CM(I,NT8+4)
  141 CONTINUE
      DO 95 I=1,NB2
      DO 96 J=1,NT8P1
      REAL1=CM(I,J)
      AIMO1=-CO*CM(I,J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIMO1).LE.1.D-16)AIMO1=0.0D0
      CM(I,J)=DCMPLX(REAL1,AIMO1)
      AIMO2=-AIMO1
      CMC(I,J)=DCMPLX(REAL1,AIMO2)
   96 CONTINUE
   95 CONTINUE
      DO 100 I=1,NT8P1
      DO 100 J=1,NT8P1
      CSUM=(0.0,0.0)
      DO 110 K=1,NB2
  110 CSUM=CMC(K,I)*CM(K,J)+CSUM
      CMCTCM(I,J)=CSUM
  100 CONTINUE
 3335 CONTINUE
      DO 120 I=1,NB
      J=I*2
      BC(J-1)=AIN1(I)
  120 BC(J)=AIN2(I)
      DO 130 I=1,NT8P1
      CSUM=(0.0,0.0)
      DO 140 K=1,NB2
  140 CSUM=CMC(K,I)*BC(K)+CSUM
  130 RHS(I)=CSUM
      IJOB=0
      IF(NOPT4.EQ.5.AND.NCASE.GT.1) IJOB=2
      M=1
      CALL LEQ2C(CMCTCM,NT8P1,NT8P1,RHS,M,NT8P1,IJOB,WA,WKK,IER)
      IF(IER.EQ.129) WRITE(6,11)
   11 FORMAT(' TERMINAL ERROR(CMCTCM),IER = 129')
      GRHS(1)=-(RHS(2*NT)*RB21+RHS(4*NT)*RB11B+RHS(6*NT)*RB21B)/RB11
      GRHS(8*NT+1)=RHS(8*NT)*SV11+RHS(8*NT+1)*SV12
      GRHS(8*NT+2)=RHS(8*NT)*SV21+RHS(8*NT+1)*SV22
      DO 151 I=2,NT8
  151 GRHS(I)=RHS(I-1)
C
C     STRESS AND STRAIN CALCULATION
C
```

```
      RAD=DARCOS(-1.0D0)/180.D0
      IC=1
      IC2=1
      SUMU1=0.0D0
      SUMV1=0.0D0
      SUMU2=0.0D0
      SUMV2=0.0D0
      NADD=0
      IF(NOPT4.EQ.1) GO TO 1195
      IF(NTYPE.NE.2.OR.NCASE.GT.4) GO TO 1196
      NADD=4*NCOL
      NIC=1
      DO 197 II=1,NADD
      XOUT(NOUT+NSTS+II)=X(II)
  197 YOUT(NOUT+NSTS+II)=Y(II)
      MN=NOUT+NSTS+NADD
      XOUT(MN+1)=AX
      YOUT(MN+1)=0.0D0
      XOUT(MN+2)=0.0D0
      YOUT(MN+2)=-AX
      XOUT(MN+3)=AX*DCOS(177.D0*RAD)
      YOUT(MN+3)=AX*DSIN(177.D0*RAD)
      XOUT(MN+4)=0.0D0
      YOUT(MN+4)=AX
      NADD=NADD+4
      GO TO 1195
 1196 CONTINUE
      NADD=8
      NIC=2
      MN=NOUT+NSTS
      DO 199 III=1,4
      ICM=5-III
      XOUT(MN+III)=XC(ICM)
      YOUT(MN+III)=YC(ICM)
  199 CONTINUE
      MN=MN+4
      XOUT(MN+1)=AX
      YOUT(MN+1)=0.0D0
      XOUT(MN+2)=0.0D0
      YOUT(MN+2)=-AX
      XOUT(MN+3)=AX*DCOS(177.D0*RAD)
      YOUT(MN+3)=AX*DSIN(177.D0*RAD)
      XOUT(MN+4)=0.0D0
      YOUT(MN+4)=AX
 1195 CONTINUE
      NRCF=NOUT
      IF(NOPT4.EQ.5) NRCF=NOUT+NSTS+NADD
      NINC=NSTS/4
      DO 190 K=1,NRCF
      Z1=XOUT(K)+R1*YOUT(K)
      Z2=XOUT(K)+R2*YOUT(K)
      Z11=CDSQRT(Z1*Z1-A*A -R1*R1*B*B)
      Z22=CDSQRT(Z2*Z2-A*A-R2*R2*B*B)
      XETA1=(Z1+Z11)/(A-CO*R1*B)
      IF(CDABS(XETA1).LT.0.999) GO TO 400
      GO TO 410
  400 Z11=-Z11
      XETA1=(Z1+Z11)/(A-CO*R1*B)
  410 XETA2=(Z2+Z22)/(A-CO*R2*B)
      IF(CDABS(XETA2).LT.0.999) GO TO 420
```

```
      GO TO 430
  420 Z22=-Z22
      XETA2=(Z2+Z22)/(A-CO*R2*B)
  430 CONTINUE
      PHI1DP=(0.0,0.0)
      PHI2DP=(0.0,0.0)
      PHI1DN=(0.0,0.0)
      PHI2DN=(0.0,0.0)
      PHI1P=(0.0,0.0)
      PHI2P=(0.0,0.0)
      PHI1N=(0.0,0.0)
      PHI2N=(0.0,0.0)
      DO 170 N=1,NT
      NP=N
      NN=-N
      PHI1DP=NP*XETA1**NP*GRHS(N)/Z11+PHI1DP
      PHI1DN=NN*XETA1**NN*GRHS(NT+N)/Z11+PHI1DN
      PHI2DP=NP*XETA2**NP*GRHS(2*NT+N)/Z22+PHI2DP
      PHI2DN=NN*XETA2**NN*GRHS(3*NT+N)/Z22+PHI2DN
      PHI1P=XETA1**NP*GRHS(N)+PHI1P
      PHI1N=XETA1**NN*GRHS(NT+N)+PHI1N
      PHI2P=XETA2**NP*GRHS(2*NT+N)+PHI2P
      PHI2N=XETA2**NN*GRHS(3*NT+N)+PHI2N
  170 CONTINUE
      PHI1D=PHI1DP+PHI1DN+GRHS(8*NT+1)/Z11
      PHI2D=PHI2DP+PHI2DN+GRHS(8*NT+2)/Z22
      PHI1=PHI1P+PHI1N+GRHS(8*NT+1)*CDLOG(XETA1)
      PHI2=PHI2P+PHI2N+GRHS(8*NT+2)*CDLOG(XETA2)
      SGMAX=2.*(R1*R1*PHI1D+R2*R2*PHI2D)
      SGMAY=2.*(PHI1D+PHI2D)
      SGMAXY=-2.*(R1*PHI1D+R2*PHI2D)
      EPSX=AMAT(1,1)*SGMAX+AMAT(1,2)*SGMAY+AMAT(1,3)*SGMAXY
      EPSY=AMAT(2,1)*SGMAX+AMAT(2,2)*SGMAY+AMAT(2,3)*SGMAXY
      EPSXY=AMAT(3,1)*SGMAX+AMAT(3,2)*SGMAY+AMAT(3,3)*SGMAXY
      U=2.*(P1*PHI1+P2*PHI2)
      V=2.*(Q1*PHI1+Q2*PHI2)
      PI=DARCOS(-1.D0)
      IF(XOUT(K).GT.0..AND.YOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI
      IF(XOUT(K).LT.0..AND.YOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180.
      IF(XOUT(K).LT.0..AND.YOUT(K).LT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180.
      IF(YOUT(K).LT.0..AND.XOUT(K).GT.0.)
     +TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+360.
      C=DCOS(TETAA*PI/180.)
      S=DSIN(TETAA*PI/180.)
      SGMAR=C**2*SGMAX+S**2*SGMAY+2.*C*S*SGMAXY
      SGMAT=S**2*SGMAX+C**2*SGMAY-2.*C*S*SGMAXY
      SGMART=-C*S*SGMAX+C*S*SGMAY+(C**2-S**2)*SGMAXY
      EPSR=C**2*EPSX+S**2*EPSY+C*S*EPSXY
      EPST=S**2*EPSX+C**2*EPSY-C*S*EPSXY
      EPSRT=2.*(-C*S*EPSX+C*S*EPSY+(C**2-S**2)*(EPSXY/2.))
      UR=U*C+V*S
      IF(NOPT4.EQ.5) GO TO 3338
      RTHT(K)=TETAA
      REPX(K)=EPSX
      REPY(K)=EPSY
      REPXY(K)=EPSXY
      ASX(K)=SGMAX
```

```
      ASXY(K)=SGMAXY
      FUR(K)=UR
      FTHT(K)=TETAA
      FSMR(K)=SGMAR
 3338 CONTINUE
      IF(NOPT4.EQ.5.AND.K.GT.NOUT) GO TO 3339
      APSX(K)=SGMAX
      APSY(K)=SGMAY
      APSXY(K)=SGMAXY
 3339 CONTINUE
      IF(NOPT4.EQ.1) GO TO 190
      IF(NOPT4.EQ.5.AND.K.LE.NOUT.OR.K.GT.(NOUT+NSTS)) GO TO 191
      IF(IC2.LE.NINC) STSV(IC2)=SGMAX
      IF(IC2.GT.NINC.AND.IC2.LE.(2*NINC)) STSV(IC2)=SGMAXY
      IF(IC2.GT.(2*NINC).AND.IC2.LE.(3*NINC)) STSV(IC2)=SGMAX
      IF(IC2.GT.(3*NINC).AND.IC2.LE.(4*NINC)) STSV(IC2)=SGMAX
      IC2=IC2+1
      GO TO 190
  191 CONTINUE
      IF(NIC.EQ.1) GO TO 192
      IF(NOPT4.EQ.5.AND.K.LT.(NRCF-7)) GO TO 190
      UVOUT(IC)=U
      UVOUT(IC+1)=V
      IC=IC+2
      GO TO 190
  192 CONTINUE
      NC=NOUT+NSTS
      IF(K.GT.NC.AND.K.LE.(NC+NCOL)) SUMU1=SUMU1+U
      IF(K.GT.(NC+NCOL).AND.K.LE.(NC+2*NCOL))SUMV1=SUMV1+V
      IF(K.GT.(NC+2*NCOL).AND.K.LE.(NC+3*NCOL)) SUMU2=SUMU2+U
      IF(K.GT.(NC+3*NCOL).AND.K.LE.(NC+4*NCOL)) SUMV2=SUMV2+V
      NNC=NC+4*NCOL
      IF(K.EQ.(NNC+1)) UVOUT(9)=U
      IF(K.EQ.(NNC+1)) UVOUT(10)=V
      IF(K.EQ.(NNC+2)) UVOUT(11)=U
      IF(K.EQ.(NNC+2)) UVOUT(12)=V
      IF(K.EQ.(NNC+3)) UVOUT(13)=U
      IF(K.EQ.(NNC+3)) UVOUT(14)=V
      IF(K.EQ.(NNC+4)) UVOUT(15)=U
      IF(K.EQ.(NNC+4)) UVOUT(16)=V
  190 CONTINUE
C
C     DISPLACEMENTS ARE AVERAGED OVER ELEMENT SIDES FOR
C     CERTAIN LOAD CASES
C
      IF(NIC.NE.1) RETURN
      SUMU1=SUMU1/FLOAT(NCOL)
      SUMV1=SUMV1/FLOAT(NCOL)
      SUMU2=SUMU2/FLOAT(NCOL)
      SUMV2=SUMV2/FLOAT(NCOL)
      UVOUT(1)=SUMU2
      UVOUT(2)=SUMV2
      UVOUT(3)=SUMU2
      UVOUT(4)=SUMV1
      UVOUT(5)=SUMU1
      UVOUT(6)=SUMV1
      UVOUT(7)=SUMU1
      UVOUT(8)=SUMV2
      RETURN
      END
```

```
C
C
C
      SUBROUTINE FBOLT(ANGK,H,PSH,K)
C
C
C     FBOLT CALCULATES THE INDIVIDUAL PLY FOUNDATION
C     MODULI AND THE INDIVIDUAL PLY LOADS
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ATETAA(400),ANG(5,2),ASIGR(400),ASIGRT(400),H(2)
      DIMENSION ASIG1(400),ASIG2(400),ASIG6(400),UR(400),ANGK(5,2)
      DIMENSION FSMR(400),PLXPT(100)
      DIMENSION IPLY(100,2),NPLY(2),NUMPLY(2)
      DIMENSION FKI(100),PLX(100)
      DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2)
      COMMON/STRESS/ASIGR,ASIGRT,ASIG1,ASIG2,ASIG6
      COMMON/ELP/AX,BX,NOUT
      COMMON/FB1/BSTR,XSTR
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/FB2/UR,ATETAA,FSMR
      COMMON/MOD/E11,E22,ESS,PMU12,PMU21
      COMMON/FCT/PLXPT
      RAD=DARCOS(-0.1D1)/180.
      THKTOT=NPLY(K)*H(K)
      NN=NUMPLY(K)
C
C     CALCULATE DELEFF
C
      WORK=0.
      PLOADX=0.
      IF(K.EQ.1) PLD=0.
      DO 210 KK=1,NOUT
      TH1=ATETAA(KK+1)-ATETAA(KK)
      TH2=(ATETAA(KK)+ATETAA(KK+1))/2.
      THETA=TH2*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      R=DSQRT(1./(C**2/AX**2+S**2/BX**2))
      FORCE=((FSMR(KK)+FSMR(KK+1))/2.)*R*TH1*RAD*THKTOT
      WORK=WORK+FORCE*.5*((UR(KK)+UR(KK+1))/2.)
      PLOADX=PLOADX+FORCE*C
  210 CONTINUE
      PLD=PLD+PLOADX
      DELEFF=WORK/PLOADX
C
C
C
C
C     COMPUTE PLY STRESSES FROM LAMINATE STRAINS
C
C     (SIGMA)R,O,RO = (Q)*(EPS)R,O,RO
C
C
      NN=NPLY(K)
      DO 100 J=1,NN
      LP=IPLY(J,K)
      THETA=(ANG(LP,K)+PSH)*RAD
      LI1=1
```

```
      LI2=NOUT
      NCAS=1
      CALL QMATX(RAD,THETA,K,LI1,LI2,NCAS)
C
C
C     INTEGRATE AROUND CIRCULAR BOUNDARY FOR
C     INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION
C     MODULI
C
C
      NNN=LI2-1
      PLOADX=0.
      WK=0.
      DO 70 I=LI1,NNN
      TH1=ATETAA(I+1)-ATETAA(I)
      TH2=(ATETAA(I)+ATETAA(I+1))/2.
      THETA=TH2*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      R=DSQRT(1./(C**2/AX**2+S**2/BX**2))
      FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*R*TH1*RAD*H(K)
      FORCRT=((ASIGRT(I)+ASIGRT(I+1))/2.)*R*TH1*RAD*H(K)
      PLOADX=PLOADX+FORCR*C-FORCRT*S
   70 CONTINUE
      FKI(J)=DABS(PLOADX/(H(K)*DELEFF))
      PLX(J+(K-1)*NPLY(1))=PLOADX
  100 CONTINUE
      NT=NUMPLY(K)
      NN=NPLY(K)
      DO 310 I=1,NT
      DO 310 II=1,NN
      IF(IPLY(II,K).EQ.I) ANGK(I,K)=FKI(II)
      IF(IPLY(II,K).EQ.I) PLXPT(I)=PLX(II+(K-1)*NPLY(1))
  310 CONTINUE
      NP=NUMPLY(K)
      DO 311 I=1,NP
      AA=ANG(I,K)+PSH
  311 CONTINUE
      PLXTOT=0.0D0
      IF(K.EQ.1) BLOAD=0.
      TH=H(K)*NPLY(K)
      BLOAD=(BSTR*DARCOS(-1.0D0)*BX*TH)/2.+BLOAD
      IF(K.EQ.1) GO TO 611
      NN=NPLY(1)+NPLY(2)
      DO 212 I=1,NN
      PLXTOT=PLXTOT+PLX(I)
  212 CONTINUE
  611 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE QMATX(RAD,THETA,K,LI1,LI2,NCAS)
C
C     QMATX PERFORMS BASIC STRESS AND STRAIN
C     TRANSFORMATIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASIGR(400),ASIGRT(400),ASIG1(400),ASIG2(400),ASIG6(400)
```

```
      DIMENSION ATETAA(400),AEPSX(400),AEPSY(400),AEPSXY(400)
      DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400)
      DIMENSION AEPS1(400)
      DIMENSION ASX(400),ASXY(400)
      COMMON/XXY1/ASX,ASXY
      COMMON/MOD/E11,E22,ESS,PMU12,PMU21
      COMMON/STRSS2/AEPS1
      COMMON/STRESS/ASIOR,ASIORT,ASIO1,ASIO2,ASIO6
      COMMON/CMT/ATETAA,AEPSX,AEPSY,AEPSXY
      COMMON/PSC3/SX,SXY
      J=0
      Q11=E11(K)/(1.0-PMU12(K)*PMU21(K))
      Q12=(PMU21(K)*E11(K))/(1.0-PMU12(K)*PMU21(K))
      Q22=E22(K)/(1.0-PMU12(K)*PMU21(K))
      Q66=ESS(K)
      C=DCOS(THETA)
      S=DSIN(THETA)
      BQ11=(Q11*(C**4))+(2.*(Q12+(2.*Q66))*(C**2)*(S**2))+(Q22*(S**4))
      BQ12=((Q11+Q22-(4.*Q66))*(S**2)*(C**2))+(Q12*(S**4+C**4))
      BQ16=((Q11-Q12-(2.*Q66))*(S*(C**3)))+((Q12-Q22+(2.*Q66))*((S**3)*C
     *))
      BQ22=(Q11*(S**4))+(2.*(Q12+(2.*Q66))*(S**2)*(C**2))+(Q22*(C**4))
      BQ26=((Q11-Q12-(2.*Q66))*C*(S**3))+((Q12-Q22+(2.*Q66))*S*(C**3))
      BQ66=((Q11+Q22-(2.*(Q12+Q66)))*(S**2)*(C**2))+(Q66*((C**4)+(S**4)
     *))
      DO 40 I=LI1,LI2
      J=J+1
      IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD
      C=DCOS(THETA)
      S=DSIN(THETA)
      SIOX=BQ11*AEPSX(I)+BQ12*AEPSY(I)+BQ16*AEPSXY(I)
      SIOY=BQ12*AEPSX(I)+BQ22*AEPSY(I)+BQ26*AEPSXY(I)
      SIOXY=BQ16*AEPSX(I)+BQ26*AEPSY(I)+BQ66*AEPSXY(I)
      SX(J)=SIOX
      SXY(J)=SIOXY
      ASIGR(I)=SIOX*C**2+SIOY*S**2+2.*SIOXY*S*C
      ASIGRT(I)=-SIOX*S*C+SIOY*C*S+SIOXY*(C**2-S**2)
      ASIG1(J)=SIOX*C**2+SIOY*S**2+2.*S*C*SIOXY
      ASIG2(J)=SIOX*S**2+SIOY*C**2-2.*S*C*SIOXY
      ASIG6(J)=-C*S*SIGX+SIGY*C*S+(C**2-S**2)*SIGXY
      AEPS1(J)=AEPSX(I)*C**2+AEPSY(I)*S**2+AEPSXY(I)*S*C
   40 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CENTD(H,FASSS,FASBS,P)
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PLYK(100),BARK(100),BARU(100),F(100)
      DIMENSION H(2),RF(2)
      DIMENSION AII(100,100),A(2),B(2)
      DIMENSION NPLY(2)
      COMMON/PBB/PLYK,BARK,BARU
      COMMON/RT/RF
      COMMON/AFM/AII,F
```

```
      COMMON/LYP/NPLY
      NNN=NPLY(1)+NPLY(2)
C
C     SET UP THE CENTRAL DIFFERENCE EQUATIONS
C
      DO 3 I=1,100
      DO 3 J=1,100
    3 AII(I,J)=0.
C
C     NECESSARY CONSTANTS ARE FORMED
C
      DO 7 I=1,2
      A(I)=H(I)**2/FASSS
    7 B(I)=H(I)**4/FASBS
      H12=H(1)/H(2)
      A1=H(1)**2/FASSS
      A2=H(2)**2/FASSS
      NP=NPLY(1)+NPLY(2)
C
C
C     SHEAR AT TOP OF PANEL EQUALS ZERO
C
      AII(1,1)=1.
      AII(1,2)=-(2.+A1*PLYK(2))
      AII(1,4)=2.+A1*PLYK(2)
      AII(1,5)=-1.
      F(1)=0.0
C
C     MOMENT CONDITION AT TOP
C
      IF(RF(1).GE.1.D10) GO TO 50
      Z=1.
      R=RF(1)
      GO TO 60
   50 Z=0.
      R=1.
   60 AII(2,1)=R
      AII(2,2)=(Z*2.*H(1)*FASSS)+R*(-2.-A1*PLYK(2)+(H(1)**2
     **FASSS)/FASBS)
      AII(2,3)=-Z*(4.*H(1)*FASSS+(2*H(1)**2*PLYK(1)*H(1)))
      AII(2,4)=Z*2.*H(1)*FASSS+R*(2.+A1*PLYK(2)-(H(1)**2
     **FASSS)/FASBS)
      AII(2,5)=-R
      F(2)=Z*2.*H(1)**3*BARK(1)*BARU(1)
C
C     GOVERNING EQUATIONS FOR THE TOP PLATE
C
      N2=NPLY(1)
      DO 55 J=1,N2
      I=J+2
      AII(I,J)=1.
      IF(J.EQ.1) GO TO 56
      AII(I,J+1)=-4.-A(1)*PLYK(J-1)
      GO TO 57
   56 AII(I,J+1)=-4.-A(1)*PLYK(2)
   57 AII(I,J+2)=6.+(2.*A(1)+B(1))*PLYK(J)
      IF(J.EQ.N2) GO TO 61
      AII(I,J+3)=-4.-A(1)*PLYK(J+1)
      GO TO 62
   61 AII(I,J+3)=-4.-A(1)*PLYK(NPLY(1)-1)
```

```
62 AII(I,J+4)=1.
   IF(J.EQ.1) GO TO 58
   IF(J.EQ.N2) GO TO 63
   F(I)=A(1)*BARK(J-1)*BARU(J-1)
  *-(2.*A(1)+B(1))*BARK(J)*BARU(J)
  *+A(1)*BARK(J+1)*BARU(J+1)
   GO TO 59
58 F(I)=2.*A(1)*BARK(2)*BARU(2)
  *-(2.*A(1)+B(1))*BARK(1)*BARU(1)
   GO TO 59
63 F(I)=2.*A(1)*BARK(NPLY(1)-1)*BARU(NPLY(1)-1)
  *-(2.*A(1)+B(1))*BARK(J)*BARU(J)
59 CONTINUE
55 CONTINUE
C
C     INTERFACE SHEAR ON TOP PLATE = P
C
   I=NPLY(1)+3
   J=NPLY(1)
   AII(I,J)=1.
   AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1))
   AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)
   AII(I,J+4)=-1.
   F(I)=(-2.*H(1)**3*(P))/FASBS
C
C     SLOPE CONTINUITY
C
   I=NPLY(1)+4
   J=NPLY(1)
   AII(I,J)=1.
   AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS)
   AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS
   AII(I,J+4)=-1.
   AII(I,J+5)=-H12**3
   AII(I,J+6)=H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSS/FASBS)
   AII(I,J+8)=-H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSS/FASBS)
   AII(I,J+9)=H12**3
   F(I)=0.
C
C     MOMENT CONTINUITY
C
   I=NPLY(1)+5
   J=NPLY(1)+1
   AII(I,J)=1.
   AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)))
   AII(I,J+2)=1.
   AII(I,J+5)=-H12**2
   AII(I,J+6)=H12**2*(2.+A2*PLYK(NPLY(1)+1))
   AII(I,J+7)=-H12**2
   F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*
  *BARU(NPLY(1)+1))
C
C     INTERFACE SHEAR ON BOTTOM PLATE
C
   I=NPLY(1)+6
   J=NPLY(1)+5
   AII(I,J)=-1.
   AII(I,J+1)=(2.+A2*PLYK(NPLY(1)+2))
   AII(I,J+3)=-(2.+A2*PLYK(NPLY(1)+2))
   AII(I,J+[illegible])=1.
```

```
      F(I)=2.*H(2)*[illegible]
C
C     GOVERNING EQUATIONS FOR THE BOTTOM PLATE
C
      N1=NPLY(1)+7
      N2=NPLY(1)+NPLY(2)+6
      DO 70 I=N1,N2
      J=I-2
      AII(I,J)=1.
      IF(I.EQ.N1) GO TO 71
      AII(I,J+1)=-4.-A(2)*PLYK(J-5)
      GO TO 72
   71 AII(I,J+1)=-4.-A(2)*PLYK(NPLY(1)+2)
   72 AII(I,J+2)=6.+(2.*A(2)+B(2))*PLYK(J-4)
      IF(I.EQ.N2) GO TO 75
      AII(I,J+3)=-4.-A(2)*PLYK(J-3)
      GO TO 76
   75 AII(I,J+3)=-4.-A(2)*PLYK(J-5)
   76 AII(I,J+4)=1.
      IF(I.EQ.N1) GO TO 73
      IF(I.EQ.N2) GO TO 77
      F(I)=A(2)*BARK(J-5)*BARU(J-5)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
     *+A(2)*BARK(J-3)*BARU(J-5)
      GO TO 74
   73 F(I)=2*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
      GO TO 74
   77 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)
     *-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
   74 CONTINUE
   70 CONTINUE
C
C     SHEAR ON BOTTOM PLATE EQUALS ZERO
C
      NP=NPLY(1)+NPLY(2)
      I=NP+7
      J=NP+4
      AII(I,J)=-1.
      AII(I,J+1)=(2.+A2*PLYK(NP-1))
      AII(I,J+3)=-(2.+A2*PLYK(NP-1))
      AII(I,J+4)=1.
      F(I)=0.
C
C     MOMENT BOUNDARY CONDITION ON BOTTOM PLATE
C
      I=NP+8
      IF(RF(2).GE.1.D10) GO TO 85
      Z=1.
      R=RF(2)
      GO TO 95
   85 Z=0.
      R=1.
   95 AII(I,J)=-R
      AII(I,J+1)=Z*(2.*H(2)*FASSS)+R*(2.+A2*PLYK(NP-1)
     *-H(2)**2*FASSS/FASBS)
      AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)**3*PLYK(NP))
      AII(I,J+3)=Z*2.*H(2)*FASSS+R*(-2.-A2*PLYK(NP-1)
     *+H(2)**2*FASSS/FASBS)
      AII(I,J+4)=R
```

```
      F(I)=Z*(2.*H(2)**3*BARK(NP)*BARU(NP))
      RETURN
      END
C
C
      SUBROUTINE SOLVE(H,P,U1,U2)
C
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(100,100),B(100),NPLY(2),U(100),F(100)
      DIMENSION SX(100),PLYK(100),H(2)
      DIMENSION BARK(100),BARU(100)
      COMMON/LYP/NPLY
      COMMON/AFM/A,F
      COMMON/P33/PLYK,BARK,BARU
C
C     SOLUTION OF THE SYSTEM: A*(U)=(B)
C
      NP=NPLY(1)+NPLY(2)+8
      DO 444 I=1,NP
  444 B(I)=F(I)
C
C     APPLYING GUASSIAN ELIMINATION TO THE
C     MATRIX OF COEFFICIENTS
C
      DO 2001 I=1,NP
      IR=I
 2042 IF(A(IR,I).NE.0.) GO TO 2041
      IR=IR+1
      IF(IR.GT.NP) GO TO 2001
      GO TO 2042
 2041 NN=IR+1
      DO 2002 L=NN,NP
      IF(DABS(A(L,I)).GT.1.D-30) GO TO 2009
      A(L,I)=0.
      GO TO 2002
 2009 CF=-A(IR,I)/A(L,I)
      DO 2003 J=I,NP
      A(L,J)=A(L,J)*CF+A(IR,J)
      IF(DABS(A(L,J)).LT.1.D-30) A(L,J)=0.0
 2003 CONTINUE
      B(L)=B(L)*CF+B(I)
 2002 CONTINUE
 2001 CONTINUE
C
C     BACK SUBSTITUTION
C
      DO 2011 I=1,NP
      L=NP+1-I
      SUM=0.
      IF(A(L,L).EQ.0.) GO TO 2112
      N=L+1
      IF(N.GT.NP) GO TO 2013
      DO 2013 J=N,NP
      SUM=SUM-A(L,J)*SX(J)
 2013 CONTINUE
      SX(L)=(B(L)+SUM)/A(L,L)
      GO TO 2011
```

```
 2112 CONTINUE
      SX(L)=0.
 2011 CONTINUE
      PT=P
      N1=NPLY(1)+2
      N2=NPLY(1)+7
      NN=NPLY(1)+NPLY(2)+6
      DO 1444 I=3,N1
      J=I-2
      U(J)=SX(I)
 1444 CONTINUE
      DO 1555 I=N2,NN
      J=I-6
      U(J)=SX(I)
 1555 CONTINUE
      NP=NPLY(1)+NPLY(2)
C
C     COMPUTE AVERAGE RELATIVE DISPLACEMENTS
C      IN TOP AND BOTTOM PLATES
C
      U1=DABS(U(1)-U(NPLY(1)))/2.
      U2=DABS(U(NPLY(1)+1)-U(NPLY(1)+NPLY(2)))/2.
      RETURN
      END
C
C
C
      SUBROUTINE FCRIT(APP,NEL1,NEL2,NDAM,IN,LTNCM,NAVD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ELSTFF(50,10,10),ELSTSS(50,50,10),U(200)
      DIMENSION OSSX(20),OSSW(20)
      DIMENSION NELDIS(50,5,2)
      DIMENSION PSMX(50,4),AVES(50,3),STRSS(50),DLT(10)
      DIMENSION ELFAIL(50,3),NELTYP(50)
      DIMENSION NELCON(50,6),NELCNA(50,6),NPLY(2)
      DIMENSION ELWDTH(50),ELTHK(50),ELLOAD(50,2)
      DIMENSION NELPLS(2,50),LYPN(50)
      COMMON/ELS/ELSTFF,ELSTSS
      COMMON/OSXW/OSSX,OSSW
      COMMON/NPLS/NELPLS,LYPN
      COMMON/NCN/NELCON,NELCNA,NELDIS
      COMMON/FCC/ELWDTH,ELTHK,ELLOAD
      COMMON/SMX/PSMX
      COMMON/LAMF/ELFAIL
      COMMON/DISP/U
      COMMON/NTP/NELTYP
      COMMON/LYP/NPLY
C
C     DETERMINE ELEMENT FAILURE LOADS IN NET SECTION,
C     SHEAROUT AND BEARING, AND LOCATE THE CRITICAL
C     FASTENER LOCATION. JOINT STRENGTH IS DETERMINED
C     FROM LOWEST ELEMENT FAILURE LOAD
C
      NELTOT=NEL1+NEL2
      NS=NAVD
      NSTS=4*NAVD
      DO 10 I=1,NELTOT
      NRNK=10
      IF(NELTYP(I).EQ.3) NRNK=8
      KJ=I
```

```
      IF(KJ.GT.NEL) KJ=KJ-NEL
      IF(NELTYP(I).EQ.1) GO TO 10
      IC=0
      DO 20 J=1,5
      IC=IC+1
      DLT(IC)=U(NELDIS(I,J,1))
      IC=IC+1
      DLT(IC)=U(NELDIS(I,J,2))
   20 CONTINUE
      DO 30 K=1,NSTS
      SUM=0.0D0
      DO 40 K2=1,NRNK
      SUM=SUM+ELSTSS(I,K,K2)*DLT(K2)
   40 CONTINUE
   30 STRSS(K)=SUM
      SUM1=0.0D0
      SUM2=0.0D0
      SUM3=0.0D0
      SUM4=0.0D0
      DO 50 J=1,NS
      SUM1=SUM1+STRSS(J)
      SUM2=SUM2+STRSS(J+NS)
      SUM3=SUM3+STRSS(J+2*NS)
   50 CONTINUE
      NN=2*NS
      DO 51 II=1,NS
   51 SUM4=SUM4+STRSS(II+3*NS)*OSSW(II)
      AVES(I,1)=SUM1/NS
      AVES(I,2)=SUM2/NS
      AVES(I,3)=SUM3/NS
      IF(I.LE.NEL1) THK=ELTHK(I)*NELPLS(1,LYPN(I))
      IF(I.GT.NEL1) THK=ELTHK(I)*NELPLS(2,LYPN(I))
      ELD=(SUM4/2.D0)*ELWDTH(I)*THK
      PRATIO=DABS((ELLOAD(I,1)+ELLOAD(I,2))/ELD)
C
C     SCALE AVERAGE STRESSES
C
      AVES(I,1)=AVES(I,1)*PRATIO
      AVES(I,2)=AVES(I,2)*PRATIO
      AVES(I,3)=AVES(I,3)*PRATIO
   10 CONTINUE
C
C     COMPUTE JOINT FAILURE LOADS BASED ON
C     ELEMENT LOADS
C
      DO 100 I=1,NELTOT
      IF(NELTYP(I).EQ.1) GO TO 100
      DO 110 J=1,3
      N=J+1
      IF(J.EQ.1.AND.LTNCM.EQ.1) N=1
      IF(J.EQ.1.AND.LTNCM.EQ.2) N=2
      ELFAIL(I,J)=DABS(APP*PSMX(I,N)/AVES(I,J))
  110 CONTINUE
  100 CONTINUE
C
C     SEARCH FOR LOWEST JOINT FAILURE LOAD
C
      INNS=0
      FNS=1.0D10
      INSO=0
```

```
      FSO=1.0D10
      INBR=0
      FBR=1.0D10
      WRITE(6,356)
  356 FORMAT(/,' JOINT LOAD LEVELS CORRESPONDING TO NET ',/,
     *' SECTION (NS), SHEAR-OUT (SO) AND BEARING (BR)',/,
     *' FAILURES AT EVERY LOADED AND UNLOADED HOLE ',/,
     *' ELEMENT ARE PREDICTED AS FOLLOWS:',//,
     *3X,' ELEMENT     NS          SO          BR',/)
      DO 120 I=1,NELTCT
      IF(NELTYP(I).EQ.1) GO TO 120
      IF(FNS.GT.DABS(ELFAIL(I,1))) INNS=I
      IF(FNS.GT.DABS(ELFAIL(I,1))) FNS=DABS(ELFAIL(I,1))
      IF(FSO.GT.DABS(ELFAIL(I,2))) INSO=I
      IF(FSO.GT.DABS(ELFAIL(I,2))) FSO=DABS(ELFAIL(I,2))
      IF(FBR.GT.DABS(ELFAIL(I,3))) INBR=I
      IF(FBR.GT.DABS(ELFAIL(I,3))) FBR=DABS(ELFAIL(I,3))
      WRITE(6,222) NELCON(I,1),ELFAIL(I,1),ELFAIL(I,2),ELFAIL(I,3)
  222 FORMAT(2X,I8,2X,3(D9.3,2X))
  120 CONTINUE
      IF(FNS.GT.FSO.OR.FNS.GT.FBR) GO TO 130
      NDAM=1
      IN=INNS
      GO TO 200
  130 IF(FSO.GT.FNS.OR.FSO.GT.FBR) GO TO 140
      NDAM=2
      IN=INSO
      GO TO 200
  140 IF(FBR.GT.FNS.OR.FBR.GT.FSO) GO TO 200
      NDAM=3
      IN=INBR
  200 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE LINV2F (A,N,IA,AINV,IDGT,WKAREA,IER)
C
      DOUBLE PRECISION   A(IA,N),AINV(IA,N),WKAREA(1),ZERO,ONE
      DATA               ONE/1.0D0/,ZERO/0.0D0/
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER=0
C                                  SET AINV TO THE N X N
C                                  IDENTITY MATRIX
      DO 10 I = 1,N
         DO 5 J = 1,N
            AINV(I,J) = ZERO
    5    CONTINUE
         AINV(I,I) = ONE
   10 CONTINUE
C                                  COMPUTE THE INVERSE OF A
      CALL LEQT2F (A,N,N,IA,AINV,IDGT,WKAREA,IER)
      IF (IER.EQ.0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HLINV2F)
 9005 RETURN
      END
C
```

```
C
      SUBROUTINE LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
      DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
      DOUBLE PRECISION   A,B,WKAREA,D1,D2,WA
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER=0
      JER=0
      J = N*N+1
      K = J+N
      MM = K+N
      KK = 0
      MM1 = MM-1
      JJ=1
      DO 5 L=1,N
         DO 5 I=1,N
            WKAREA(JJ)=A(I,L)
            JJ=JJ+1
    5 CONTINUE
C                                  DECOMPOSE A
      CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
     *             WA,IER)
      IF (IER.GT.128) GO TO 25
      IF (IDGT .EQ. 0 .OR. IER .NE. 0) KK = 1
      DO 15 I = 1,M
C                                  PERFORMS THE ELIMINATION PART OF
C                                  AX = B
         CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))
C                                  REFINEMENT OF SOLUTION TO AX = B
         IF (KK .NE. 0)
     *   CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDGT,WKAREA(J),WKAREA(MM),
     *                WKAREA(K),WKAREA(K),JER)
         DO 10 II=1,N
            B(II,I) = WKAREA(MM1+II)
   10    CONTINUE
         IF (JER.NE.0) GO TO 20
   15 CONTINUE
      GO TO 25
   20 IER = 131
   25 JJ=1
      DO 30 J = 1,M
         DO 30 I = 1,N
            A(I,J)=WKAREA(JJ)
            JJ=JJ+1
   30 CONTINUE
      IF (IER .EQ. 0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HLEQT2F)
 9005 RETURN
      END
C
C
C
      SUBROUTINE LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)
C
      DIMENSION          A(IA,1),LU(IA,1),IPVT(1),EQUIL(1)
      DOUBLE PRECISION   A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH,
     *                   RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.D0,1.D0,4.D0,
```

```
     *                        16.D0,.0625D0/
C                                      FIRST EXECUTABLE STATEMENT
C                                      INITIALIZATION
      IER = 0
      RN = N
      WREL = ZERO
      D1 = ONE
      D2 = ZERO
      BIGA = ZERO
      DO 10 I=1,N
         BIG = ZERO
         DO 5 J=1,N
            P = A(I,J)
            LU(I,J) = P
            P = DABS(P)
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .GT. BIGA) BIGA = BIG
         IF (BIG .EQ. ZERO) GO TO 110
         EQUIL(I) = ONE/BIG
   10 CONTINUE
      DO 105 J=1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 40
C                                      COMPUTE U(I,J), I=1,...,J-1
         DO 35 I=1,JM1
            SUM = LU(I,J)
            IM1 = I-1
            IF (IDOT .EQ. 0) GO TO 25
C                                      WITH ACCURACY TEST
            AI = DABS(SUM)
            WI = ZERO
            IF (IM1 .LT. 1) GO TO 20
            DO 15 K=1,IM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   15       CONTINUE
            LU(I,J) = SUM
   20       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 35
C                                      WITHOUT ACCURACY
   25       IF (IM1 .LT. 1) GO TO 35
            DO 30 K=1,IM1
               SUM = SUM-LU(I,K)*LU(K,J)
   30       CONTINUE
            LU(I,J) = SUM
   35    CONTINUE
   40    P = ZERO
C                                      COMPUTE U(J,J) AND L(I,J), I=J+1,....
         DO 70 I=J,N
            SUM = LU(I,J)
            IF (IDOT .EQ. 0) GO TO 55
C                                      WITH ACCURACY TEST
            AI = DABS(SUM)
            WI = ZERO
            IF (JM1 .LT. 1) GO TO 50
```

```
            DO 45 K=1,JM1
               T = LU(I,K)*LU(K,J)
               SUM = SUM-T
               WI = WI+DABS(T)
   45       CONTINUE
            LU(I,J) = SUM
   50       WI = WI+DABS(SUM)
            IF (AI .EQ. ZERO) AI = BIGA
            TEST = WI/AI
            IF (TEST .GT. WREL) WREL = TEST
            GO TO 65
C                                  WITHOUT ACCURACY TEST
   55       IF (JM1 .LT. 1) GO TO 65
            DO 60 K=1,JM1
               SUM = SUM-LU(I,K)*LU(K,J)
   60       CONTINUE
            LU(I,J) = SUM
   65       Q = EQUIL(I)*DABS(SUM)
            IF (P .GE. Q) GO TO 70
            P = Q
            IMAX = I
   70    CONTINUE
C                                  TEST FOR ALGORITHMIC SINGULARITY
         IF (RN+P .EQ. RN) GO TO 110
         IF (J .EQ. IMAX) GO TO 80
C                                  INTERCHANGE ROWS J AND IMAX
         D1 = -D1
         DO 75 K=1,N
            P = LU(IMAX,K)
            LU(IMAX,K) = LU(J,K)
            LU(J,K) = P
   75    CONTINUE
         EQUIL(IMAX) = EQUIL(J)
   80    IPVT(J) = IMAX
         D1 = D1*LU(J,J)
   85    IF (DABS(D1) .LE. ONE) GO TO 90
         D1 = D1*SIXTH
         D2 = D2+FOUR
         GO TO 85
   90    IF (DABS(D1) .GE. SIXTH) GO TO 95
         D1 = D1*SIXTN
         D2 = D2-FOUR
         GO TO 90
   95    CONTINUE
         JP1 = J+1
         IF (JP1 .GT. N) GO TO 105
C                                  DIVIDE BY PIVOT ELEMENT U(J,J)
         P = LU(J,J)
         DO 100 I=JP1,N
            LU(I,J) = LU(I,J)/P
  100    CONTINUE
  105 CONTINUE
C                                  PERFORM ACCURACY TEST
      IF (IDGT .EQ. 0) GO TO 9005
      P = 3*N+3
      WA = P*WREL
      IF (WA+10.D0**(-IDGT) .NE. WA) GO TO 9005
      IER = 34
      GO TO 9000
C                                  ALGORITHMIC SINGULARITY
```

```
  110 IER = 129
      D1 = ZERO
      D2 = ZERO
 9000 CONTINUE
C                                 PRINT ERROR
      CALL UERTST(IER,6HLUDATF)
 9005 RETURN
      END
C
C
      SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
      DIMENSION           A(IA,1),B(1),APVT(1),X(1)
      DOUBLE PRECISION    A,B,X,SUM,APVT
C                                 FIRST EXECUTABLE STATEMENT
C                                 SOLVE LY = B FOR Y
      DO 5 I=1,N
    5 X(I) = B(I)
      IW = 0
      DO 20 I=1,N
         IP = APVT(I)
         SUM = X(IP)
         X(IP) = X(I)
         IF (IW .EQ. 0) GO TO 15
         IM1 = I-1
         DO 10 J=IW,IM1
            SUM = SUM-A(I,J)*X(J)
   10    CONTINUE
         GO TO 20
   15    IF (SUM .NE. 0.D0) IW = I
   20 X(I) = SUM
C                                 SOLVE UX = Y FOR X
      DO 30 IB=1,N
         I = N+1-IB
         IP1 = I+1
         SUM = X(I)
         IF (IP1 .GT. N) GO TO 30
         DO 25 J=IP1,N
            SUM = SUM-A(I,J)*X(J)
   25   CONTINUE
   30 X(I) = SUM/A(I,I)
      RETURN
      END
C
C
      SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDGT,APVT,X,RES,DX,IER)
C
      DIMENSION           A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1)
      DIMENSION           APVT(1)
      DIMENSION           ACCXT(2)
      DOUBLE PRECISION    A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT
      DATA                ITMAX/75/,ZERO/0.D0/
C                                 FIRST EXECUTABLE STATEMENT
      IER=0
      XNORM = ZERO
      DO 10 I=1,N
         XNORM = DMAX1(XNORM,DABS(X(I)))
   10 CONTINUE
      IF (XNORM .NE. ZERO) GO TO 20
      IDGT = 50
```

```
      GO TO 9005
   20 DO 45 ITER=1,ITMAX
         DO 30 I=1,N
      ACCXT(1) = 0.0D0
      ACCXT(2) = 0.0D0
            CALL VXADD(B(I),ACCXT)
            DO 25 J=1,N
               CALL VXMUL(-A(I,J),X(J),ACCXT)
   25       CONTINUE
            CALL VXSTO(ACCXT,RES(I))
   30    CONTINUE
         CALL LUELMN (UL,IUL,N,RES,APVT,DX)
         DXNORM = ZERO
         XNORM = ZERO
         DO 35 I=1,N
            X(I) = X(I) + DX(I)
            DXNORM = DMAX1(DXNORM,DABS(DX(I)))
            XNORM = DMAX1(XNORM,DABS(X(I)))
   35    CONTINUE
         IF (ITER .NE. 1) GO TO 40
         IDOT = 50
         IF (DXNORM .NE. ZERO) IDOT = -DLOG10(DXNORM/XNORM)
   40    IF (XNORM+DXNORM .EQ. XNORM) GO TO 9005
   45 CONTINUE
C                                 ITERATION DID NOT CONVERGE
      IER = 129
 9000 CONTINUE
      CALL UERTST(IER,6HLUREFN)
 9005 RETURN
      END
C
C
C
      SUBROUTINE UERTST (IER,NAME)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IER
      INTEGER            NAME(1)
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(6),
     *                   NAMSET(6),NAMUPK(6),NIN,NMTB
      DATA               NAMSET/1HU,1HE,1HR,1HS,1HE,1HT/
      DATA               NAMEQ/6*1H /
      DATA               LEVEL/4/,IEQDF/0/,IEQ/1H=/
C                                 UNPACK NAME INTO NAMUPK
C                                 FIRST EXECUTABLE STATEMENT
      CALL USPKD (NAME,6,NAMUPK,NMTB)
C                                 GET OUTPUT UNIT NUMBER
      CALL UGETIO(1,NIN,IOUNIT)
C                                 CHECK IER
      IF (IER.GT.999) GO TO 25
      IF (IER.LT.-32) GO TO 55
      IF (IER.LE.128) GO TO 5
      IF (LEVEL.LT.1) GO TO 30
C                                 PRINT TERMINAL MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
      GO TO 30
    5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
C                                 PRINT WARNING WITH FIX MESSAGE
```

```
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
      GO TO 30
   10 IF (IER.LE.32) GO TO 15
C                                  PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
      GO TO 30
   15 CONTINUE
C                                  CHECK FOR UERSET CALL
      DO 20 I=1,6
         IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
   20 CONTINUE
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
   25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
C                                  PRINT NON-DEFINED MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK
   30 IEQDF = 0
      RETURN
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
C
C                                  SAVE P FOR P = R CASE
C                                    P IS THE PAGE NAMUPK
C                                    R IS THE ROUTINE NAMUPK
   55 IEQDF = 1
      DO 60 I=1,6
   60 NAMEQ(I) = NAMUPK(I)
   65 RETURN
      END
C
C
C
      SUBROUTINE UGETIO(IOPT,NIN,NOUT)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IOPT,NIN,NOUT
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            NIND,NOUTD
      DATA               NIND/5/,NOUTD/6/
C                                  FIRST EXECUTABLE STATEMENT
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
      NOUT = NOUTD
      GO TO 9005
```

```
    5 NIND = NIN
      GO TO 9005
   10 NOUTD = NOUT
 9005 RETURN
      END
C
C
C
      SUBROUTINE VXADD(A,ACC)
C
C                                  SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   A,ACC(2)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   X,Y,Z,ZZ
C                                  FIRST EXECUTABLE STATEMENT
      X = ACC(1)
      Y = A
      IF (DABS(ACC(1)).GE.DABS(A)) GO TO 1
      X = A
      Y = ACC(1)
C                                  COMPUTE Z+ZZ = ACC(1)+A EXACTLY
    1 Z = X+Y
      ZZ = (X-Z)+Y
C                                  COMPUTE ZZ+ACC(2) USING DOUBLE
C                                    PRECISION ARITHMETIC
      ZZ = ZZ+ACC(2)
C                                  COMPUTE ACC(1)+ACC(2) = Z+ZZ EXACTLY
      ACC(1) = Z+ZZ
      ACC(2) = (Z-ACC(1))+ZZ
      RETURN
      END
C
C
C
      SUBROUTINE VXMUL (A,B,ACC)
C                                  SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   A,B,ACC(2)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   X,HA,TA,HB,TB
      INTEGER            IX(2),I
      LOGICAL*1          LX(8),LI(4)
      EQUIVALENCE        (X,LX(1),IX(1)),(I,LI(1))
      DATA               I/0/
C                                  SPLIT A = HA+TA
C                                        B = HB+TB
C                                  FIRST EXECUTABLE STATEMENT
      X = A
      LI(4) = LX(5)
      IX(2) = 0
      I = (I/16)*16
      LX(5) = LI(4)
      HA=X
      TA=A-HA
      X = B
      LI(4) = LX(5)
      IX(2) = 0
      I = (I/16)*16
      LX(5) = LI(4)
      HB = X
      TB = B-HB
```

```
C                                 COMPUTE HA*HB,HA*TB,TA*HB, AND TA*TB
C                                   AND CALL VXADD TO ACCUMULATE THE
C                                   SUM
      X = TA*TB
      CALL VXADD(X,ACC)
      X = HA*TB
      CALL VXADD(X,ACC)
      X = TA*HB
      CALL VXADD(X,ACC)
      X = HA*HB
      CALL VXADD(X,ACC)
      RETURN
      END
C
C
C
      SUBROUTINE VXSTO (ACC,D)
C                                 SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   ACC(2),D
C                                 FIRST EXECUTABLE STATEMENT
      D = ACC(1)+ACC(2)
      RETURN
      END
C
C
C
      SUBROUTINE ZRPOLY (A,NDEG,Z,IER)
C                                 SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NDEG,IER
      DOUBLE PRECISION   A(1),Z(1)
C                                 SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,JJ,I,NM1,ICNT,N2,L,NZ,NPI
      REAL               ETA,RMRE,RINFP,REPSP,RADIX,RLO,XX,YY,SINR,
     1                   COSR,RMAX,RMIN,X,SC,XM,FF,DX,DF,BND,XXX,ARE
      REAL               PT(101)
      DOUBLE PRECISION   TEMP(101),P(101),QP(101),RK(101),QK(101),
     1                   SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   T,AA,BB,CC,FACTOR,REPSR1,ZERO,ONE,FN
      LOGICAL            ZEROK
      COMMON /ZRPGLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
C                                 THE FOLLOWING STATEMENTS SET MACHINE
C                                   CONSTANTS USED IN VARIOUS PARTS OF
C                                   THE PROGRAM. THE MEANING OF THE
C                                   FOUR CONSTANTS ARE - REPSR1 THE
C                                   MAXIMUM RELATIVE REPRESENTATION
C                                   ERROR WHICH CAN BE DESCRIBED AS
C                                   THE SMALLEST POSITIVE FLOATING
C                                   POINT NUMBER SUCH THAT 1.+REPSR1 IS
C                                   GREATER THAN 1
C                                 RINFP THE LARGEST FLOATING-POINT
C                                   NUMBER
C                                 REPSP THE SMALLEST POSITIVE
C                                   FLOATING-POINT NUMBER IF THE
C                                   EXPONENT RANGE DIFFERS IN SINGLE
C                                   AND DOUBLE PRECISION THEN REPSP
C                                   AND RINFP SHOULD INDICATE THE
C                                   SMALLER RANGE
```

```
C                                              RADIX THE DASE OF THE FLOATING-POINT
C                                                NUMBER SYSTEM USED
      DATA                    RINFP/Z7FFFFFFF/
      DATA                    REPSP/Z00100000/
      DATA                    RADIX/16.0/
      DATA                    RE: ' :1/Z341000000000000000/
      DATA                    ZERU/0.0D0/,ONE/1.0D0/
C                                              ZRPOLY USES SINOLE PRECISION
C                                                CALCULATIONS FOR SCALING, BOUNDS
C                                                AND ERROR CALCULATIONS.
C                                              FIRST EXECUTABLE STATEMENT
      IER = 0
      IF (NDEO .OT. 100 .OR. NDEO .LT. 1) GO TO 165
      ETA = REPSR1
      ARE = ETA
      RMRE = ETA
      RLO = REPSP/ETA
C                                              INITIALIZATION OF CONSTANTS FOR
C                                                SHIFT ROTATION
      XX = .7071068
      YY = -XX
      SINR = .9975641
      COSR = -.06975647
      N = NDEO
      NN = N+1
C                                              ALOORITHM FAILS IF THE LEADING
C                                                COEFFICIENT IS ZERO.
      IF (A(1).NE.ZERO) GO TO 5
      IER = 130
      GO TO 9000
C                                              REMOVE THE ZEROS AT THE ORIOIN IF
C                                                ANY
    5 IF (A(NN).NE.ZERO) GO TO 10
      J = NDEO-N+1
      JJ = J+NDEO
      Z(J) = ZERO
      Z(JJ) = ZERO
      NN = NN-1
      N = N-1
      IF (NN.EQ.1) GO TO 9005
      GO TO 5
C                                              MAKE A COPY OF THE COEFFICIENTS
   10 DO 15 I=1,NN
         P(I) = A(I)
   15 CONTINUE
C                                              START THE ALOORITHM FOR ONE ZERO
   20 IF (N.GT.2) GO TO 30
      IF (N.LT.1) GO TO 9005
C                                              CALCULATE THE FINAL ZERO OR PAIR OF
C                                                ZEROS
      IF (N.EQ.2) GO TO 25
      Z(NDEO) = -P(2)/P(1)
      Z(NDEO+NDEO) = ZERO
      GO TO 145
   25 CALL ZRPQLI (P(1),P(2),P(3),Z(NDEO-1),Z(NDEO+NDEO-1),Z(NDEO),
     1   Z(NDEO+NDEO))
      GO TO 145
C                                              FIND LARGEST AND SMALLEST MODULI OF
C                                                COEFFICIENTS.
   30 RMAX = 0.
```

```
      RMIN = RINFP
      DO 35 I=1,NN
         X = ABS(SNGL(P(I)))
         IF (X.GT.RMAX) RMAX = X
         IF (X.NE.0..AND.X.LT.RMIN) RMIN = X
   35 CONTINUE
C                                  SCALE IF THERE ARE LARGE OR VERY
C                                    SMALL COEFFICIENTS COMPUTES A
C                                    SCALE FACTOR TO MULTIPLY THE
C                                    COEFFICIENTS OF THE POLYNOMIAL.
C                                    THE SCALING IS DONE TO AVOID
C                                    OVERFLOW AND TO AVOID UNDETECTED
C                                    UNDERFLOW INTERFERING WITH THE
C                                    CONVERGENCE CRITERION.
C                                  THE FACTOR IS A POWER OF THE BASE
      SC = RLO/RMIN
      IF (SC.GT.1.0) GO TO 40
      IF (RMAX.LT.10.) GO TO 55
      IF (SC.EQ.0.) SC = REPSP*RADIX*RADIX
      GO TO 45
   40 IF (RINFP/SC.LT.RMAX) GO TO 55
   45 L = ALOG(SC)/ALOG(RADIX)+.5
      IF (L .EQ. 0) GO TO 55
      FACTOR = DBLE(RADIX)**L
      DO 50 I=1,NN
   50 P(I) = FACTOR*P(I)
C                                  COMPUTE LOWER BOUND ON MODULI OF
C                                    ZEROS.
   55 DO 60 I=1,NN
   60 PT(I) = ABS(SNGL(P(I)))
      PT(NN) = -PT(NN)
C                                  COMPUTE UPPER ESTIMATE OF BOUND
      X = EXP((ALOG(-PT(NN))-ALOG(PT(1)))/N)
      IF (PT(N).EQ.0.) GO TO 65
C                                  IF NEWTON STEP AT THE ORIGIN IS
C                                    BETTER, USE IT.
      XM = -PT(NN)/PT(N)
      IF (XM.LT.X) X = XM
C                                  CHOP THE INTERVAL (0,X) UNTIL FF.LE.0
   65 XM = X*.1
      FF = PT(1)
      DO 70 I=2,NN
   70 FF = FF*XM+PT(I)
      IF (FF.LE.0.) GO TO 75
      X = XM
      GO TO 65
   75 DX = X
C                                  DO NEWTON ITERATION UNTIL X
C                                    CONVERGES TO TWO DECIMAL PLACES
   80 IF (ABS(DX/X).LE..005) GO TO 90
      FF = PT(1)
      DF = FF
      DO 85 I=2,N
         FF = FF*X+PT(I)
         DF = DF*X+FF
   85 CONTINUE
      FF = FF*X+PT(NN)
      DX = FF/DF
      X = X-DX
      GO TO 80
```

```
   90 BND = X
C                                         COMPUTE THE DERIVATIVE AS THE INTIAL
C                                           K POLYNOMIAL AND DO 5 STEPS WITH
C                                           NO SHIFT
      NM1 = N-1
      FN = ONE/N
      DO 95 I=2,N
   95 RK(I) = (NN-I)*P(I)*FN
      RK(1) = P(1)
      AA = P(NN)
      BB = P(N)
      ZEROK = RK(N).EQ.ZERO
      DO 115 JJ=1,5
         CC = RK(N)
         IF (ZEROK) GO TO 105
C                                         USE SCALED FORM OF RECURRENCE IF
C                                           VALUE OF K AT 0 IS NONZERO
         T = -AA/CC
         DO 100 I=1,NM1
            J = NN-I
            RK(J) = T*RK(J-1)+P(J)
  100    CONTINUE
         RK(1) = P(1)
         ZEROK = DABS(RK(N)).LE.DABS(BB)*ETA*10.
         GO TO 115
C                                         USE UNSCALED FORM OF RECURRENCE
  105    DO 110 I=1,NM1
            J = NN-I
            RK(J) = RK(J-1)
  110    CONTINUE
         RK(1) = ZERO
         ZEROK = RK(N).EQ.ZERO
  115 CONTINUE
C                                         SAVE K FOR RESTARTS WITH NEW SHIFTS
      DO 120 I=1,N
  120 TEMP(I) = RK(I)
C                                         LOOP TO SELECT THE QUADRATIC
C                                           CORRESPONDING TO EACH NEW SHIFT
      DO 140 ICNT=1,20
C                                         QUADRATIC CORRESPONDS TO A DOUBLE
C                                           SHIFT TO A NON-REAL POINT AND ITS
C                                           COMPLEX CONJUGATE. THE POINT HAS
C                                           MODULUS BND AND AMPLITUDE ROTATED
C                                           BY 94 DEGREES FROM THE PREVIOUS
C                                           SHIFT
         XXX = COSR*XX-SINR*YY
         YY = SINR*XX+COSR*YY
         XX = XXX
         SR = BND*XX
         SI = BND*YY
         U = -SR-SR
         V = BND*BND
C                                         SECOND STAGE CALCULATION, FIXED
C                                           QUADRATIC
         CALL ZRPQLB (20*ICNT,NZ)
         IF (NZ.EQ.0) GO TO 130
C                                         THE SECOND STAGE JUMPS DIRECTLY TO
C                                           ONE OF THE THIRD STAGE ITERATIONS
C                                           AND RETURNS HERE IF SUCCESSFUL.
C                                         DEFLATE THE POLYNOMIAL, STORE THE
```

```
C                                            ZERO OR ZEROS AND RETURN TO THE
C                                              MAIN ALGORITHM.
         J = NDEG-N+1
         JJ = J+NDEG
         Z(J) = SZR
         Z(JJ) = SZI
         NN = NN-NZ
         N = NN-1
         DO 125 I=1,NN
  125    P(I) = QP(I)
         IF (NZ.EQ.1) GO TO 20
         Z(J+1) = RLZR
         Z(JJ+1) = RLZI
         GO TO 20
C                                            IF THE ITERATION IS UNSUCCESSFUL
C                                              ANOTHER QUADRATIC IS CHOSEN AFTER
C                                              RESTORING K
  130    DO 135 I=1,N
  135    RK(I) = TEMP(I)
  140 CONTINUE
C                                            RETURN WITH FAILURE IF NO
C                                              CONVERGENCE WITH 20 SHIFTS
         IER = 131
C                                            CONVERT ZEROS (Z) IN COMPLEX FORM
  145 DO 150 I=1,NDEG
         NPI= NDEG+I
         P(I) = Z(NPI)
  150 CONTINUE
      N2 = NDEG+NDEG
      J = NDEG
      DO 155 I=1,NDEG
         Z(N2-1) = Z(J)
         Z(N2) = P(J)
         N2 = N2-2
         J = J-1
  155 CONTINUE
      IF (IER .EQ. 0) GO TO 9005
C                                            SET UNFOUND ROOTS TO MACHINE INFINITY
      N2 = 2*(NDEG-NN)+3
      DO 160 I=1,N
         Z(N2) = RINFP
         Z(N2+1) = RINFP
         N2 = N2+2
  160 CONTINUE
      GO TO 9000
  165 IER = 129
 9000 CONTINUE
      CALL UERTST (IER,6HZRPOLY)
 9005 RETURN
      END
C
C
C
      SUBROUTINE ZRPQLB (L2,NZ)
C                                            SPECIFICATIONS FOR ARGUMENTS
      INTEGER            L2,NZ
C                                            SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,ITYPE,I,IFLAG
      REAL               ARE,BETAS,BETAV,ETA,OSS,OTS,OTV,OVV,RMRE,SS,
     1                   TS,TSS,TV,TVV,VV
```

```
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   SVU,SVV,UI,VI,S,ZERO
      LOGICAL            VPASS,SPASS,VTRY,STRY
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                  FIRST EXECUTABLE STATEMENT
      NZ = 0
C                                  COMPUTES UP TO L2 FIXED SHIFT
C                                    K-POLYNOMIALS, TESTING FOR
C                                    CONVERGENCE IN THE LINEAR OR
C                                    QUADRATIC CASE. INITIATES ONE OF
C                                    THE VARIABLE SHIFT ITERATIONS AND
C                                    RETURNS WITH THE NUMBER OF ZEROS
C                                    FOUND.
C                                  L2 - LIMIT OF FIXED SHIFT STEPS
C                                  NZ -NUMBER OF ZEROS FOUND
      BETAV = .25
      BETAS = .25
      OSS = SR
      OVV = V
C                                  EVALUATE POLYNOMIAL BY SYNTHETIC
C                                    DIVISION
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      CALL ZRPQLE (ITYPE)
      DO 40 J=1,L2
C                                  CALCULATE NEXT K POLYNOMIAL AND
C                                    ESTIMATE V
         CALL ZRPQLF (ITYPE)
         CALL ZRPQLE (ITYPE)
         CALL ZRPQLG (ITYPE,UI,VI)
         VV = VI
C                                  ESTIMATE S
         SS = 0.
         IF (RK(N).NE.ZERO) SS = -P(NN)/RK(N)
         TV = 1.
         TS = 1.
         IF (J.EQ.1.OR.ITYPE.EQ.3) GO TO 35
C                                  COMPUTE RELATIVE MEASURES OF
C                                    CONVERGENCE OF S AND V SEQUENCES
         IF (VV.NE.0.) TV = ABS((VV-OVV)/VV)
         IF (SS.NE.0.) TS = ABS((SS-OSS)/SS)
C                                  IF DECREASING, MULTIPLY TWO MOST
C                                    RECENT CONVERGENCE MEASURES
         TVV = 1.
         IF (TV.LT.OTV) TVV = TV*OTV
         TSS = 1.
         IF (TS.LT.OTS) TSS = TS*OTS
C                                  COMPARE WITH CONVERGENCE CRITERIA
         VPASS = TVV.LT.BETAV
         SPASS = TSS.LT.BETAS
         IF (.NOT.(SPASS.OR.VPASS)) GO TO 35
C                                  AT LEAST ONE SEQUENCE HAS PASSED THE
C                                    CONVERGENCE TEST. STORE VARIABLES
C                                    BEFORE ITERATING
         SVU = U
         SVV = V
         DO 5 I=1,N
```

```
    5    SVK(I) = RK(I)
         S = SS
C                                      CHOOSE ITERATION ACCORDING TO THE
C                                        FASTEST CONVERGING SEQUENCE
         VTRY = .FALSE.
         STRY = .FALSE.
         IF (SPASS.AND.((.NOT.VPASS).OR.TSS.LT.TVV)) GO TO 20
   10    CALL ZRPQLC (UI,VI,NZ)
         IF (NZ.GT.0) RETURN
C                                      QUADRATIC ITERATION HAS FAILED. FLAG
C                                        THAT IT HAS BEEN TRIED AND
C                                        DECREASE THE CONVERGENCE
C                                        CRITERION.
         VTRY = .TRUE.
         BETAV = BETAV*.25
C                                      TRY LINEAR ITERATION IF IT HAS NOT
C                                        BEEN TRIED AND THE S SEQUENCE IS
C                                        CONVERGING
         IF (STRY.OR.(.NOT.SPASS)) GO TO 25
         DO 15 I=1,N
   15    RK(I) = SVK(I)
   20    CALL ZRPQLD (S,NZ,IFLAG)
         IF (NZ.GT.0) RETURN
C                                      LINEAR ITERATION HAS FAILED. FLAG
C                                        THAT IT HAS BEEN TRIED AND
C                                        DECREASE THE CONVERGENCE CRITERION
         STRY = .TRUE.
         BETAS = BETAS*.25
         IF (IFLAG.EQ.0) GO TO 25
C                                      IF LINEAR ITERATION SIGNALS AN
C                                        ALMOST DOUBLE REAL ZERO ATTEMPT
C                                        QUADRATIC INTERATION
         UI = -(S+S)
         VI = S*S
         GO TO 10
C                                      RESTORE VARIABLES
   25    U = SVU
         V = SVV
         DO 30 I=1,N
   30    RK(I) = SVK(I)
C                                      TRY QUADRATIC ITERATION IF IT HAS
C                                        NOT BEEN TRIED AND THE V SEQUENCE
C                                        IS CONVERGING
         IF (VPASS.AND.(.NOT.VTRY)) GO TO 10
C                                      RECOMPUTE QP AND SCALAR VALUES TO
C                                        CONTINUE THE SECOND STAGE
         CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
         CALL ZRPQLE (ITYPE)
   35    OVV = VV
         OSS = SS
         OTV = TV
         OTS = TS
   40 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLC (UU,VV,NZ)
C                                      SPECIFICATIONS FOR ARGUMENTS
```

```
      INTEGER            NZ
      DOUBLE PRECISION   UU,VV
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,I,ITYPE
      REAL               ARE,EE,ETA,OMP,RELSTP,RMP,RMRE,T,ZM
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   UI,VI,ZERO,PT01,ONE
      LOGICAL            TRIED
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO,PT01,ONE/0.0D0,0.01D0,1.0D0/
C                                  FIRST EXECUTABLE STATEMENT
      NZ = 0
C                                  VARIABLE-SHIFT K-POLYNOMIAL
C                                    ITERATION FOR A QUADRATIC FACTOR
C                                    CONVERGES ONLY IF THE ZEROS ARE
C                                    EQUIMODULAR OR NEARLY SO
C                                  UU,VV - COEFFICIENTS OF STARTING
C                                    QUADRATIC
C                                  NZ - NUMBER OF ZERO FOUND
      TRIED = .FALSE.
      U = UU
      V = VV
      J = 0
C                                  MAIN LOOP
    5 CALL ZRPQLI (ONE,U,V,SZR,SZI,RLZR,RLZI)
C                                  RETURN IF ROOTS OF THE QUADRATIC ARE
C                                    REAL AND NOT CLOSE TO MULTIPLE OR
C                                    NEARLY EQUAL AND OF OPPOSITE SIGN
      IF ( DABS(DABS(SZR)-DABS(RLZR)).GT.PT01*DABS(RLZR)) RETURN
C                                  EVALUATE POLYNOMIAL BY QUADRATIC
C                                    SYNTHETIC DIVISION
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      RMP = DABS(RA-SZR*RB)+DABS(SZI*RB)
C                                  COMPUTE A RIGOROUS BOUND ON THE
C                                    ROUNDING ERROR IN EVALUTING P
      ZM = SQRT(ABS(SNGL(V)))
      EE = 2.*ABS(SNGL(QP(1)))
      T = -SZR*RB
      DO 10 I=2,N
   10 EE = EE*ZM+ABS(SNGL(QP(I)))
      EE = EE*ZM+ABS(SNGL(RA)+T)
      EE = (5.*RMRE+4.*ARE)*EE-(5.*RMRE+2.*ARE)*(ABS(SNGL(RA)+T)+
     1     ABS(SNGL(RB))*ZM)+2.*ARE*ABS(T)
C                                  ITERATION HAS CONVERGED SUFFICIENTLY
C                                    IF THE POLYNOMIAL VALUE IS LESS
C                                    THAN 20 TIMES THIS BOUND
      IF (RMP.GT.20.*EE) GO TO 15
      NZ = 2
      RETURN
   15 J = J+1
C                                  STOP ITERATION AFTER 20 STEPS
      IF (J.GT.20) RETURN
      IF (J.LT.2) GO TO 25
      IF (RELSTP.GT..01.OR.RMP.LT.OMP.OR.TRIED) GO TO 25
C                                  A CLUSTER APPEARS TO BE STALLING THE
C                                    CONVERGENCE. FIVE FIXED SHIFT
C                                    STEPS ARE TAKEN WITH A U,V CLOSE
```

```
C                                           TO THE CLUSTER
      IF (RELSTP.LT.ETA) RELSTP = ETA
      RELSTP = SQRT(RELSTP)
      U = U-U*RELSTP
      V = V+V*RELSTP
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
      DO 20 I=1,5
         CALL ZRPQLE (ITYPE)
         CALL ZRPQLF (ITYPE)
   20 CONTINUE
      TRIED = .TRUE.
      J = 0
   25 OMP = RMP
C                                         CALCULATE NEXT K POLYNOMIAL AND NEW
C                                           U AND V
      CALL ZRPQLE (ITYPE)
      CALL ZRPQLF (ITYPE)
      CALL ZRPQLE (ITYPE)
      CALL ZRPQLG (ITYPE,UI,VI)
C                                         IF VI IS ZERO THE ITERATION IS NOT
C                                           CONVERGING
      IF (VI.EQ.ZERO) RETURN
      RELSTP = DABS((VI-V)/VI)
      U = UI
      V = VI
      GO TO 5
      END
C
C
C
      SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)
C                                         SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NZ,IFLAG
      DOUBLE PRECISION   SSS
C                                         SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,J,I
      REAL               ARE,EE,ETA,OMP,RMP,RMS,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   PV,RKV,T,S,ZERO,PT001
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/,PT001/0.001D0/
C                                         VARIABLE-SHIFT H POLYNOMIAL
C                                           ITERATION FOR A REAL ZERO SSS -
C                                           STARTING ITERATE
C                                         NZ - NUMBER OF ZERO FOUND
C                                         IFLAG - FLAG TO INDICATE A PAIR OF
C                                           ZEROS NEAR REAL AXIS
C                                         FIRST EXECUTABLE STATEMENT
      NZ = 0
      S = SSS
      IFLAG = 0
      J = 0
C                                         MAIN LOOP
    5 PV = P(1)
C                                         EVALUATE P AT S
      QP(1) = PV
      DO 10 I=2,NN
```

```
         PV = PV*S+P(I)
         QP(I) = PV
   10 CONTINUE
      RMP = DABS(PV)
C                                      COMPUTE A RIGOROUS BOUND ON THE
C                                        ERROR IN EVALUATING P
      RMS = DABS(S)
      EE = (RMRE/(ARE+RMRE))*ABS(SNGL(QP(1)))
      DO 15 I=2,NN
   15 EE = EE*RMS+ABS(SNGL(QP(I)))
C                                      ITERATION HAS CONVERGED SUFFICIENTLY
C                                        IF THE POLYNOMIAL VALUE IS LESS
C                                        THAN 20 TIMES THIS BOUND
      IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20
      NZ = 1
      SZR = S
      SZI = ZERO
      RETURN
   20 J = J+1
C                                      STOP ITERATION AFTER 10 STEPS
      IF (J.GT.10) RETURN
      IF (J.LT.2) GO TO 25
      IF (DABS(T).GT.PT001*DABS(S-T).OR.RMP.LE.OMP) GO TO 25
C                                      A CLUSTER OF ZEROS NEAR THE REAL
C                                        AXIS HAS BEEN ENCOUNTERED RETURN
C                                        WITH IFLAG SET TO INITIATE A
C                                        QUADRATIC ITERATION
      IFLAG = 1
      SSS = S
      RETURN
C                                      RETURN IF THE POLYNOMIAL VALUE HAS
C                                        INCREASED SIGNIFICANTLY
   25 OMP = RMP
C                                      COMPUTE T, THE NEXT POLYNOMIAL, AND
C                                        THE NEW ITERATE
      RKV = RK(1)
      QK(1) = RKV
      DO 30 I=2,N
         RKV = RKV*S+RK(I)
         QK(I) = RKV
   30 CONTINUE
      IF (DABS(RKV).LE.DABS(RK(N))*10.*ETA) GO TO 40
C                                      USE THE SCALED FORM OF THE
C                                        RECURRENCE IF THE VALUE OF K AT S
C                                        IS NONZERO
      T = -PV/RKV
      RK(1) = QP(1)
      DO 35 I=2,N
   35 RK(I) = T*QK(I-1)+QP(I)
      GO TO 50
C                                      USE UNSCALED FORM
   40 RK(1) = ZERO
      DO 45 I=2,N
   45 RK(I) = QK(I-1)
   50 RKV = RK(1)
      DO 55 I=2,N
   55 RKV = RKV*S+RK(I)
      T = ZERO
      IF (DABS(RKV).GT.DABS(RK(N))*10.*ETA) T = -PV/RKV
      S = S+T
```

```
      GO TO 5
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLE
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLE (ITYPE)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
C                                  THIS ROUTINE CALCULATES SCALAR
C                                    QUANTITIES USED TO COMPUTE THE
C                                    NEXT K POLYNOMIAL AND NEW
C                                    ESTIMATES OF THE QUADRATIC
C                                    COEFFICIENTS
C                                  ITYPE - INTEGER VARIABLE SET HERE
C                                    INDICATING HOW THE CALCULATIONS
C                                    ARE NORMALIZED TO AVOID OVERFLOW
C                                  SYNTHETIC DIVISION OF K BY THE
C                                    QUADRATIC 1,U,V
C                                  FIRST EXECUTABLE STATEMENT
      CALL ZRPQLH (N,U,V,RK,QK,C,D)
      IF (DABS(C).GT.DABS(RK(N))*100.*ETA) GO TO 5
      IF (DABS(D).GT.DABS(RK(N-1))*100.*ETA) GO TO 5
      ITYPE = 3
C                                  TYPE=3 INDICATES THE QUADRATIC IS
C                                    ALMOST A FACTOR OF K
      RETURN
    5 IF (DABS(D).LT.DABS(C)) GO TO 10
      ITYPE = 2
C                                  TYPE=2 INDICATES THAT ALL FORMULAS
C                                    ARE DIVIDED BY D
      E = RA/D
      F = C/D
      G = U*RB
      H = V*RB
      A3 = (RA+G)*E+H*(RB/D)
      A1 = RB*F-RA
      A7 = (F+U)*RA+H
      RETURN
   10 ITYPE = 1
C                                  TYPE=1 INDICATES THAT ALL FORMULAS
C                                    ARE DIVIDED BY C
      E = RA/C
      F = D/C
      G = U*E
```

```
      H = V*RB
      A3 = RA*E+(H/C+G)*RB
      A1 = RB-RA*(D/C)
      A7 = RA+G*D+H*F
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLF (ITYPE)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN,I
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,TEMP,ZERO
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                  COMPUTES THE NEXT K POLYNOMIALS
C                                    USING SCALARS COMPUTED IN ZRPQLE
C                                  FIRST EXECUTABLE STATEMENT
      IF (ITYPE.EQ.3) GO TO 20
      TEMP = RA
      IF (ITYPE.EQ.1) TEMP = RB
      IF (DABS(A1).GT.DABS(TEMP)*ETA*10.) GO TO 10
C                                  IF A1 IS NEARLY ZERO THEN USE A
C                                    SPECIAL FORM OF THE RECURRENCE
      RK(1) = ZERO
      RK(2) = -A7*QP(1)
      DO 5 I=3,N
    5 RK(I) = A3*QK(I-2)-A7*QP(I-1)
      RETURN
C                                  USE SCALED FORM OF THE RECURRENCE
   10 A7 = A7/A1
      A3 = A3/A1
      RK(1) = QP(1)
      RK(2) = QP(2)-A7*QP(1)
      DO 15 I=3,N
   15 RK(I) = A3*QK(I-2)-A7*QP(I-1)+QP(I)
      RETURN
C                                  USE UNSCALED FORM OF THE RECURRENCE
C                                    IF TYPE IS 3
   20 RK(1) = ZERO
      RK(2) = ZERO
      DO 25 I=3,N
   25 RK(I) = QK(I-2)
      RETURN
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLG
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
```

```
C   LATEST REVISION      - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLG (ITYPE,UU,VV)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            ITYPE
      DOUBLE PRECISION   UU,VV
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            N,NN
      REAL               ARE,ETA,RMRE
      DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101)
      DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,
     1                   A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,
     2                   A4,A5,B1,B2,C1,C2,C3,C4,TEMP,ZERO
      COMMON /ZRPQLJ/    P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
     1                   A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
      DATA               ZERO/0.0D0/
C                                  COMPUTE NEW ESTIMATES OF THE
C                                    QUADRATIC COEFFICIENTS USING THE
C                                    SCALARS COMPUTED IN ZRPQLE
C                                  USE FORMULAS APPROPRIATE TO SETTING
C                                    OF TYPE.
C                                  FIRST EXECUTABLE STATEMENT
      IF (ITYPE.EQ.3) GO TO 15
      IF (ITYPE.EQ.2) GO TO 5
      A4 = RA+U*RB+H*F
      A5 = C+(U+V*F)*D
      GO TO 10
    5 A4 = (RA+G)*F+H
      A5 = (F+U)*C+V*D
C                                  EVALUATE NEW QUADRATIC COEFFICIENTS.
C
   10 B1 = -RK(N)/P(NN)
      B2 = -(RK(N-1)+B1*P(N))/P(NN)
      C1 = V*B2*A1
      C2 = B1*A7
      C3 = B1*B1*A3
      C4 = C1-C2-C3
      TEMP = A5+B1*A4-C4
      IF (TEMP.EQ.ZERO) GO TO 15
      UU = U-(U*(C3+C2)+V*(B1*A1+B2*A7))/TEMP
      VV = V*(1+C4/TEMP)
      RETURN
C                                  IF TYPE=3 THE QUADRATIC IS ZEROED
   15 UU = ZERO
      VV = ZERO
      RETURN
      END
C
C
C
      SUBROUTINE ZRPQLH (NN,U,V,P,Q,RA,RB)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NN
      DOUBLE PRECISION   P(NN),Q(NN),U,V,RA,RB
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I
      DOUBLE PRECISION   C
C                                  DIVIDES P BY THE QUADRATIC 1,U,V
C                                    PLACING THE QUOTIENT IN Q AND THE
C                                    REMAINDER IN A,B
```

```
C                                   FIRST EXECUTABLE STATEMENT
      RB = P(1)
      Q(1) = RB
      RA = P(2)-U*RB
      Q(2) = RA
      DO 5 I=3,NN
         C = P(I)-U*RA-V*RB
         Q(I) = C
         RB = RA
         RA = C
    5 CONTINUE
      RETURN
      END
C
C
C
C   IMSL ROUTINE NAME   - ZRPQLI
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
      SUBROUTINE ZRPQLI (RA,B1,C,SR,SI,RLR,RLI)
C                                   SPECIFICATIONS FOR ARGUMENTS
      DOUBLE PRECISION   RA,B1,C,SR,SI,RLR,RLI
C                                   SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   RB,D,E,ZERO,ONE,TWO
      DATA               ZERO,ONE,TWO/0.0D0,1.0D0,2.0D0/
C                                   CALCULATE THE ZEROS OF THE QUADRATIC
C                                     A*Z**2 + B1*Z + C. THE QUADRATIC
C                                     FORMULA, MODIFIED TO AVOID
C                                     OVERFLOW, IS USED TO FIND THE
C                                     LARGER ZERO IF THE ZEROS ARE REAL
C                                     AND BOTH ZEROS ARE COMPLEX.
C                                   THE SMALLER REAL ZERO IS FOUND
C                                     DIRECTLY FROM THE PRODUCT OF THE
C                                     ZEROS C/A
C                                   FIRST EXECUTABLE STATEMENT
      IF (RA.NE.ZERO) GO TO 10
      SR = ZERO
      IF (B1.NE.ZERO) SR = -C/B1
      RLR = ZERO
    5 SI = ZERO
      RLI = ZERO
      RETURN
   10 IF (C.NE.ZERO) GO TO 15
      SR = ZERO
      RLR = -B1/RA
      GO TO 5
C                                   COMPUTE DISCRIMINANT AVOIDING
C                                     OVERFLOW
   15 RB = B1/TWO
      IF (DABS(RB).LT.DABS(C)) GO TO 20
      E = ONE-(RA/RB)*(C/RB)
      D = DSQRT(DABS(E))*DABS(RB)
      GO TO 25
   20 E = RA
      IF (C.LT.ZERO) E = -RA
```

```
      E = RB*(RB/DABS(C))-E
      D = DSQRT(DABS(E))*DSQRT(DABS(C))
   25 IF (E.LT.ZERO) GO TO 30
C                                   REAL ZEROS
      IF (RB.GE.ZERO) D = -D
      RLR = (-RB+D)/RA
      SR = ZERO
      IF (RLR.NE.ZERO) SR = (C/RLR)/RA
      GO TO 5
C                                   COMPLEX CONJUGATE ZEROS
   30 SR = -RB/RA
      RLR = SR
      SI = DABS(D/RA)
      RLI = -SI
      RETURN
      END
C
C
C
      SUBROUTINE LEQ2C  (A,N,IA,B,M,IB,IJOB,WA,WK,IER)
C
      COMPLEX*16         A(IA,1),B(IB,1),WA(N,1),TEMPA,TEMPB,TEMPC
      DOUBLE PRECISION   WK(N),TA(2),TB(2),TC(2)
      DOUBLE PRECISION   AR,AI,BR,BI,CR,CI,DXNORM,XNORM,ZERO
      DOUBLE PRECISION   ACC(2)
      EQUIVALENCE        (TA(1),TEMPA),(TB(1),TEMPB),(TC(1),TEMPC),
     *                   (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI),
     *                   (TC(1),CR),(TC(2),CI)
      DATA               ZERO/0.0D0/
      DATA               ITMAX/50/
C                                   FIRST EXECUTABLE STATEMENT
      IER = 0
      N1 = N+1
      N2 = N+2
      IF (IJOB .EQ. 2) GO TO 15
C                                   SAVE MATRIX A
      DO 10 I = 1,N
         DO 5 J = 1,N
            WA(I,J) = A(I,J)
    5    CONTINUE
   10 CONTINUE
C                                   FACTOR MATRIX A
      CALL LEQT1C (WA,N,N,B,M,IB,1,WK,IER)
      IF (IER .NE. 0) GO TO 9000
      IF (IJOB .EQ. 1) GO TO 9005
C                                   SAVE THE RIGHT HAND SIDES
   15 DO 65 J = 1,M
         DO 20 I = 1,N
            WA(I,N1) = B(I,J)
   20    CONTINUE
C                                   OBTAIN A SOLUTION
         CALL LEQT1C(WA,N,N,WA(1,N1),1,N,2,WK,IER)
C                                   COMPUTE THE NORM OF THE SOLUTION
         XNORM = ZERO
         DO 25 I = 1,N
            TEMPA = WA(I,N1)
            XNORM = DMAX1(XNORM,DABS(AR),DABS(AI))
   25    CONTINUE
         IF (XNORM .EQ. ZERO) GO TO 65
C                                   COMPUTE RESIDUALS
```

```
          DO 50 ITER = 1,ITMAX
             DO 40 I = 1,N
                TEMPB = B(I,J)
                ACC(1) = 0.0D0
                ACC(2) = 0.0D0
                CALL VXADD(BR,ACC)
                DO 30 JJ = 1,N
                   TEMPA = A(I,JJ)
                   TEMPB = WA(JJ,N1)
                   CALL VXMUL(-AR,BR,ACC)
                   CALL VXMUL(AI,BI,ACC)
   30           CONTINUE
                CALL VXSTO(ACC,CR)
                TEMPB = B(I,J)
                ACC(1) = 0.0D0
                ACC(2) = 0.0D0
                CALL VXADD(BI,ACC)
                DO 35 JJ = 1,N
                   TEMPA = A(I,JJ)
                   TEMPB = WA(JJ,N1)
                   CALL VXMUL(-AR,BI,ACC)
                   CALL VXMUL(-BR,AI,ACC)
   35           CONTINUE
                CALL VXSTO(ACC,CI)
                WA(I,N2) = TEMPC
   40        CONTINUE
             CALL LEQT1C(WA,N,N,WA(1,N2),1,N,2,WK,IER)
             DXNORM = ZERO
C                                       UPDATE THE SOLUTION
             DO 45 I = 1,N
                WA(I,N1) = WA(I,N1)+WA(I,N2)
                TEMPA = WA(I,N2)
                DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI))
   45        CONTINUE
             IF (XNORM+DXNORM .EQ. XNORM) GO TO 55
   50     CONTINUE
          IER = 130
C                                       STORE THE SOLUTION
   55     DO 60 JK = 1,N
             B(JK,J) = WA(JK,N1)
   60     CONTINUE
          IF (IER .NE. 0) GO TO 9000
   65 CONTINUE
      GO TO 9005
 9000 CONTINUE
      CALL UERTST(IER,6HLEQ2C )
 9005 RETURN
      END
C
C
C
      SUBROUTINE LEQT1C (A,N,IA,B,M,IB,IJOB,WA,IER)
C                                       SPECIFICATIONS FOR ARGUMENTS
      INTEGER            N,IA,M,IB,IJOB,IER
      COMPLEX*16         A(IA,N),B(IB,M)
      DOUBLE PRECISION   WA(N)
C                                       SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION   P,Q,ZERO,ONE,T(2),RN,BIG
      COMPLEX*16         SUM,TEMP
      INTEGER            I,J,JM1,IM1,K,IMAX,JP1,IW,N1
```

```
      EQUIVALENCE       (SUM,T(1))
      DATA              ZERO/0.0D0/,ONE/1.D0/
C                                  INITIALIZATION
C                                  FIRST EXECUTABLE STATEMENT
      IER = 0
      IF (IJOB .EQ. 2) GO TO 75
      RN = N
C                                  FIND EQUILIBRATION FACTORS
      DO 10 I=1,N
         BIG = ZERO
         DO 5 J=1,N
            TEMP = A(I,J)
            P = CDABS(TEMP)
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .EQ. ZERO) GO TO 105
         WA(I) = ONE/BIG
   10 CONTINUE
C                                  L-U DECOMPOSITION
      DO 70 J = 1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 25
C                                  COMPUTE U(I,J), I=1,...,J-1
         DO 20 I=1,JM1
            SUM = A(I,J)
            IM1 = I-1
            IF (IM1 .LT. 1) GO TO 20
            DO 15 K=1,IM1
               SUM = SUM-A(I,K)*A(K,J)
   15       CONTINUE
            A(I,J) = SUM
   20    CONTINUE
   25    P = ZERO
C                                  COMPUTE U(J,J) AND L(I,J), I=J+1,...,
         DO 45 I=J,N
            SUM = A(I,J)
            IF (JM1 .LT. 1) GO TO 40
            DO 35 K=1,JM1
               SUM = SUM-A(I,K)*A(K,J)
   35       CONTINUE
            A(I,J) = SUM
   40       Q = WA(I)*CDABS(SUM)
            IF (P .GE. Q) GO TO 45
            P = Q
            IMAX = I
   45    CONTINUE
C                                  TEST FOR ALGORITHMIC SINGULARITY
         Q = RN+P
         IF (Q .EQ. RN) GO TO 105
         IF (J .EQ. IMAX) GO TO 60
C                                  INTERCHANGE ROWS J AND IMAX
         DO 50 K=1,N
            TEMP = A(IMAX,K)
            A(IMAX,K) = A(J,K)
            A(J,K) = TEMP
   50    CONTINUE
         WA(IMAX) = WA(J)
   60    WA(J) = IMAX
         JP1 = J+1
         IF (JP1 .GT. N) GO TO 70
```

```
C                                       DIVIDE BY PIVOT ELEMENT U(J,J)
         TEMP = A(J,J)
         DO 65 I = JP1,N
            A(I,J) = A(I,J)/TEMP
   65    CONTINUE
   70 CONTINUE
   75 IF (IJOB .EQ. 1) GO TO 9005
      DO 103 K = 1,M
C                                       SOLVE UX = Y FOR X
         IW = 0
         DO 90 I = 1,N
            IMAX = WA(I)
            SUM = B(IMAX,K)
            B(IMAX,K) = B(I,K)
            IF (IW .EQ. 0) GO TO 85
            IM1 = I-1
            DO 80 J = IW,IM1
               SUM = SUM-A(I,J)*B(J,K)
   80       CONTINUE
            GO TO 88
   85       IF (T(1) .NE. ZERO .OR. T(2) .NE. ZERO) IW = I
   88       B(I,K) = SUM
   90    CONTINUE
C                                       SOLVE LY = B FOR Y
         N1 = N+1
         DO 100 IW = 1,N
            I = N1-IW
            JP1 = I+1
            SUM = B(I,K)
            IF (JP1 .GT. N) GO TO 98
            DO 95 J = JP1,N
               SUM = SUM-A(I,J)*B(J,K)
   95       CONTINUE
   98       B(I,K) = SUM/A(I,I)
  100    CONTINUE
  103 CONTINUE
      GO TO 9005
C                                       ALGORITHMIC SINGULARITY
  105 IER = 129
 9000 CONTINUE
C                                       PRINT ERROR
      CALL UERTST(IER,6HLEQT1C)
 9005 RETURN
      END
```


U.S. Government Printing Office: 198? [illegible]

SUPPLEMENTARY

INFORMATION

DEPARTMENT OF THE AIR FORCE
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES (AFSC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

REPLY TO ATTN OF: IMST (513/255-7466) 1 May 1987

SUBJECT: Correction to AFWAL Technical Reports, AFWAL-TR-86-3034 and 86-3035

TO: ALL ADDRESSES

1. Please delete the second paragraph in the NOTICE page affixed to the inside cover of AFWAL-TR-86-3034, "Strength Analysis of Laminated and Metallic Plates Bolted Together by Many Fasteners" and AFWAL-TR-86-3035, "Design Guide for Bolted Joints in Composite Structures."

2. Please contact the undersigned if you have any questions regarding this letter.

AD-B108123

G. Doben

G. DOBEN
Chief, Scientific & Tech Info Gp
Information Services Branch

cc: AFWAL/FIBRA
(V. Venkayya)

UNITED STATES AIR FORCE
40
SEPTEMBER 18, 1947