

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

AD-B108 123

(2)

AFWAL-TR-86-3035

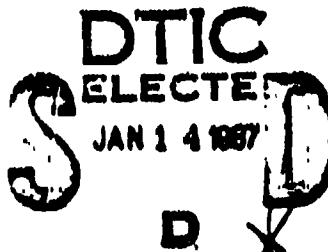
DESIGN GUIDE FOR BOLTED JOINTS IN COMPOSITE STRUCTURES

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

Northrop Corporation, Aircraft Division
One Northrop Avenue
Hawthorne, California 90280

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.

DTIC FILE COPY

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

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

ADB108123L

REPORT DOCUMENTATION PAGE

16. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		17. RESTRICTIVE MARKINGS <i>T/E</i>													
20. SECURITY CLASSIFICATION AUTHORITY		21. 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													
26. DECLASSIFICATION/DOWNGRADING SCHEDULE		28. MONITORING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-86-3035													
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOR 86-211		7a. NAME OF PERFORMING ORGANIZATION Northrop Corporation Aircraft Division													
6a. ADDRESS (City, State and ZIP Code) One Northrop Avenue Hawthorne, CA 90230	8a. OFFICE SYMBOL 111/00940001	7b. NAME OF MONITORING ORGANIZATION Flight Dynamics Laboratory (AFWAL/FIBRA) Air Force Wright Aeronautical Laboratories													
6c. ADDRESS (City, State and ZIP Code)	7b. ADDRESS (City, State and ZIP Code)	7c. ADDRESS (City, State and ZIP Code) Wright-Patterson Air Force Base Dayton, OH 45433-6553													
8c. NAME OF FUNDING/SPONSORING ORGANIZATION	8d. OFFICE SYMBOL 111/00940001	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract F33615-82-C-3217													
10. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NO. <table border="1"> <tr> <th>PROGRAM ELEMENT NO.</th> <th>PROJECT NO.</th> <th>TASK NO.</th> <th>WORK UNIT NO.</th> </tr> <tr> <td>62201F</td> <td>2401</td> <td>02</td> <td>55</td> </tr> </table>			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.	62201F	2401	02	55				
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. DATE OF REPORT (Yr., Mo., Day) March 1986													
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 1/1/81 TO 1/30/85	13c. PAGE COUNT 264													
16. SUPPLEMENTARY NOTATION															
17. COBALT CODES <table border="1"> <tr> <th>FIELD</th> <th>GROUP</th> <th>SUB GRP</th> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>2</td> <td>2</td> <td>2</td> </tr> <tr> <td>3</td> <td>3</td> <td>3</td> </tr> </table>	FIELD	GROUP	SUB GRP	1	1	1	2	2	2	3	3	3	18. SUBJECT TERMS (Consume on reverse if necessary and identify by code numbers) Bolted joints, composite structures, design guidelines, analytical design tools (SASCJ and SAMCJ), user instructions, samples, program listing.		
FIELD	GROUP	SUB GRP													
1	1	1													
2	2	2													
3	3	3													
19. ABSTRACT (Continue on reverse if necessary and identify by code numbers) 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		21. ABSTRACT SECURITY CLASSIFICATION									
UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input checked="" type="checkbox"/> OFIC USERS <input type="checkbox"/>			UNCLASSIFIED												
22a. NAME OF RESPONSIBLE INDIVIDUAL V. B. Venkayya		22b. TELEPHONE NUMBER (513) 235-6992		22c. OFFICE SYMBOL AFWAL/FIBRA											

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. S. 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



Accession For		
NTIS	CRA&I	<input type="checkbox"/>
DTIC	TAU	<input checked="" type="checkbox"/>
Unpublished		
Justification		
By		
Distribution		/
Availability Codes		
Dist	Avail. Ind./or Special	
B-3	51	

TABLE OF CONTENTS

SECTION	PAGE	
1	INTRODUCTION.....	1
	1.1 Scope of the Design Guide.....	2
	1.2 Sample Joint Configurations.....	2
	1.3 Overview of Design Methodology.....	2
	1.4 Analytical Requirements.....	12
	1.5 Test Requirements.....	12
	1.6 Current Restrictions.....	13
2	GENERAL DESIGN GUIDELINES AND JOINT VARIABLES.....	15
	2.1 Joint Location in the Structure.....	15
	2.2 Joint Configurations.....	17
	2.3 Joint Loading.....	17
	2.3.1 Joint Loads at the Structural Level.....	17
	2.3.2 Load Distribution Among Rows of Fasteners.....	20
	2.3.3 Bearing and By-Pass Loads at an Isolated Fastener Location.....	25
	2.4 Failure Modes in Bolted Laminates.....	27
	2.5 Fastener Type, Material and Installation Variables.....	27
	2.5.1 Fastener Type.....	27
	2.5.2 Fastener Material.....	29
	2.5.3 Fastener Size.....	30
	2.5.4 Fastener Fit and Hole Quality.. ..	30
	2.5.5 Fastener Torque-up.....	33
	2.6 Bolted Laminate Properties.....	35
	2.7 Fastener Spacing and Arrangement.....	37
	2.8 Joint Tailoring for Maximum Efficiency.....	40
	2.9 Durability Considerations.....	46

TABLE OF CONTENTS (Concluded)

SECTION	PAGE
3	STRENGTH ANALYSIS OF BOLTED COMPOSITE STRUCTURES..... 49
	3.1 Description of SASCJ Analysis..... 49
	3.1.1 Strength Analysis Procedure for Fully-Loaded Holes..... 51
	3.1.2 Strength Analysis Procedure for Partially-Loaded Holes..... 53
	3.1.3 Inplane Failure Criteria..... 55
	3.1.4 Interlaminar Failure Criterion..... 61
	3.2 SASCJ Input Description..... 61
	3.3 SASCJ Output Description..... 68
	3.4 Description of SAMCJ Analysis..... 69
	3.5 SAMCJ Input Description..... 79
	3.6 SAMCJ Output Description..... 88
4	DESIGN VERIFICATION OF A BOLTED STRUCTURAL ELEMENT... 95
	4.1 Description of the Bolted Structural Element... 95
	4.2 Test Results..... 97
	4.3 Design Verification of the Element Using SAMCJ.. 97
5	CONCLUSIONS..... 105
	REFERENCES..... 107
	Appendix A - SASCJ Program Listing..... 109
	Appendix B - SAMCJ Program Listing..... 183

LIST OF ILLUSTRATIONS

FIGURE	PAGE
1 Sample Bolted Joints in the F/A-18A Aircraft Wing (Reference 1).....	3
2 A Bolted Joint Concept for a Fuselage Structure (Reference 2).....	4
3 Bolted Joint Concepts for Composite Fuselage Structures (Reference 2).....	5
4 Skin-to-Root Fitting Joint in the F-20 Horizontal Stabilizer.....	6
5 Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section (Reference 1).....	7
6 Alternative Bolted Joint Concepts Evaluated for the F/A-18A Vertical Tail Root Section (References 3 to 5).....	9
7 Overview of the Strength Analysis of Bolted Structures.....	11
8 Structural and Analyzed Bolted Joint Configurations..	18
9 Inplane Loads in Typical Wing Skin-to-Substructure Attachments.....	19
10 Sample Joint Configurations that Introduce Significant Out-of-Plane Loads at the Joint Location.....	21
11 Fastener Load Distribution in the Laminated Plate for Two Double Shear Configurations (References 7, 8)....	23
12 Effect of Joint Configuration on Fastener Load Distribution (Reference 9).....	24
13 Interaction Between Bearing and By-Pass Loads at a Fastener Location.....	26
14 Basic Failure Modes in Bolted Laminates and Fastener- Related Failures.....	28
15 Galvanic Compatibility and Corrosion Prevention.....	31

LIST OF ILLUSTRATIONS (Continued)

FIGURE	PAGE
16 Effect of Fastener Size on the Load Distribution (Reference 6).....	32
17 The Envelope of Bolted Laminate Layups for Realizing a Bearing Mode of Failure and the Maximum Bearing Strength (Reference 9).....	36
18 Geometrical Parameters for a Bolted Plate.....	38
19 Effect of Fastener Size on the Tensile Response of Composites-to-Metal Joints in Single Shear.....	39
20 Effect of E/D on the Bearing Strength of Bolted Laminates.....	41
21 Effect of W/D on the Bearing Strength of Bolted Laminates.....	42
22 Effect of S /D on the Strength of Bolted Laminates...	43
23 A Sample Tailored Joint.....	45
24 Effect of Maximum Cyclic Bearing Stress on the Number of R=-1 Fatigue Cycles to Cause Specified Hole Elongations in a Bolted Laminate.....	47
25 Effect of Maximum Cyclic Bearing Stress on Hole Elongation Rate for a Bolted Laminate Under R=-1 Loading.....	48
26 Schematic Representation of a General Single Fastener situation as a Superposition of Unloaded and Fully- Loaded Hole Situations.....	50
27 Flowchart for the Strength Analysis of Laminates with Fully-Loaded Holes.....	52
28 Strength Analysis of Laminates with Partially Loaded Holes Using Average Stress Failure Criteria...	54

LIST OF ILLUSTRATIONS (Continued)

FIGURE	PAGE
29 The Characteristic Distances Used in the Point Stress Failure Criteria.....	56
30 The Characteristic Distances Used in the Average Stress Failure Criteria.....	56
31 Bilinear Elastic Behavior of a Ply.....	58
32 A Schematic Representation of the Overall Load Versus Deflection Response of the Joint.....	59
33 The Characteristic Distance (a_0) Defining the Region Where the Maximum Strain, Hoffman or Hill Criterion is Applied.....	59
34 SASCJ Analysis of a Joint in Single Shear.....	63
35 SASCJ Analysis of a Joint in Double Shear.....	64
36 Sample SASCJ Input.....	65
37 SASCJ Output for the Problem Defined in Figure 36....	70
38 Flow Chart of SAMCJ Operations.....	72
39 Application of Load and Displacement Boundary Conditions in the SAMCJ Code.....	74
40 Finite Element Model of a Sample Tapered Bolted Joint.....	76
41 Element Load Recovery for Various a/D and b/D Ratios.....	78
42 Nine Element Model of Each of the two Bolted Plates in the Sample Joint.....	80
43 SAMCJ Input for the Sample Problem in Figure 42....	81
44 SAMCJ Output for the Problem Defined in Figure 43....	89
45 Comparison of SAMCJ Predictions for the Sample Problem with Test Results from Reference 8.....	93
46 Photograph of an Assembled Test Element.....	96
47 Photograph of the Tab Region of the Failed Element.....	98

LIST OF ILLUSTRATIONS (Concluded)

FIGURE	PAGE	
48	Dimensions of the Critical Skin Tab and the Fuselage Attachment Frame.....	99
49	Load Distribution Among Fasteners, Failure Location and Failure Mode in the Graphite/Epoxy Tabs.....	101
50	Analytically Predicted Element Load Levels to Precipitate Net Section, Bearing and Shear-Out Modes of Tab Failure at Each Fastener Location.....	103

LIST OF TABLES

TABLE	PAGE
1 GALVANIC COMPATIBILITY OF FASTENER MATERIALS WITH COMPOSITES (REFERENCE 12).....	31
2 EFFECTS OF FLAWS AT FASTENER HOLE LOCATIONS (REFERENCE 13).....	34

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 familiarized 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

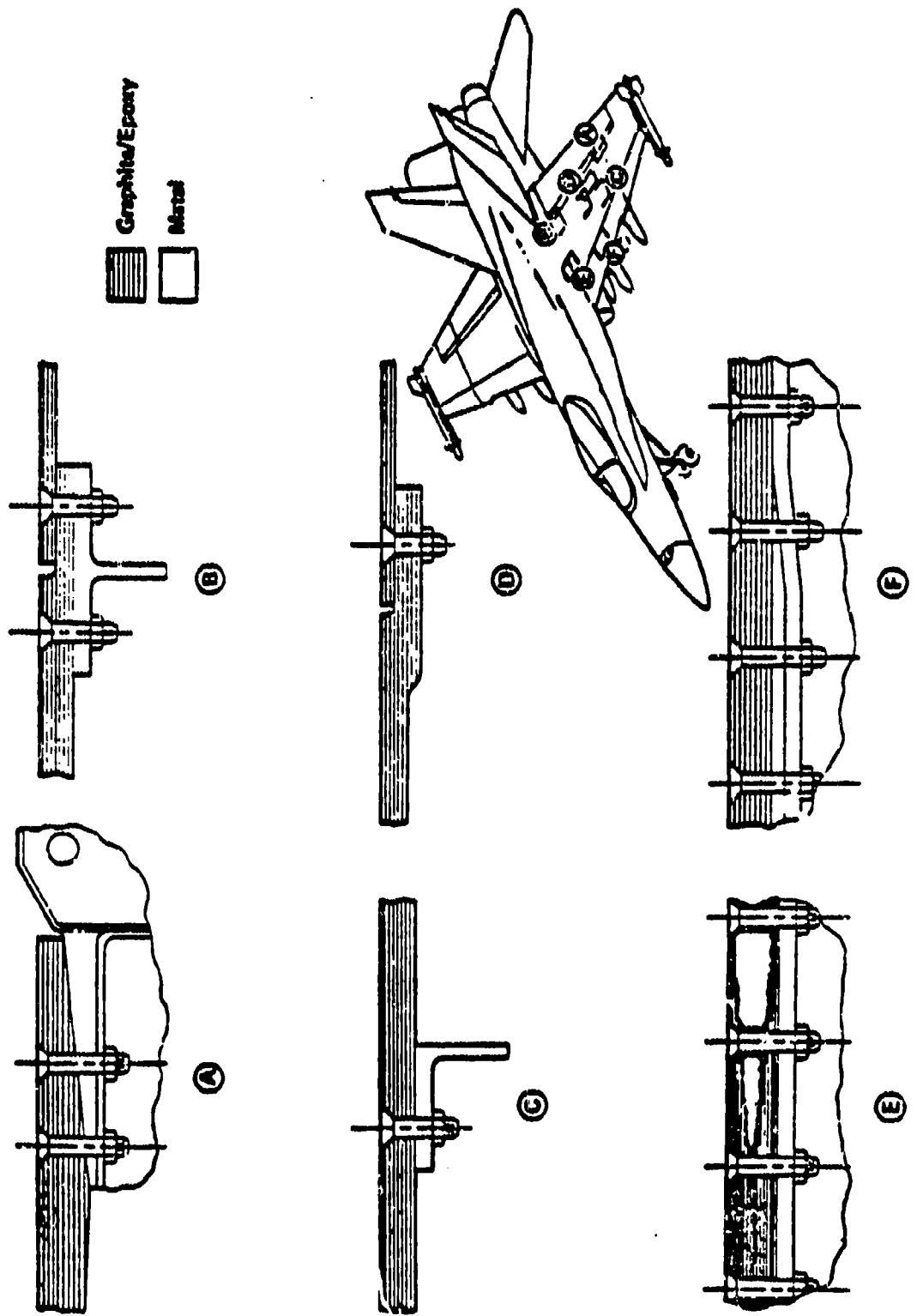
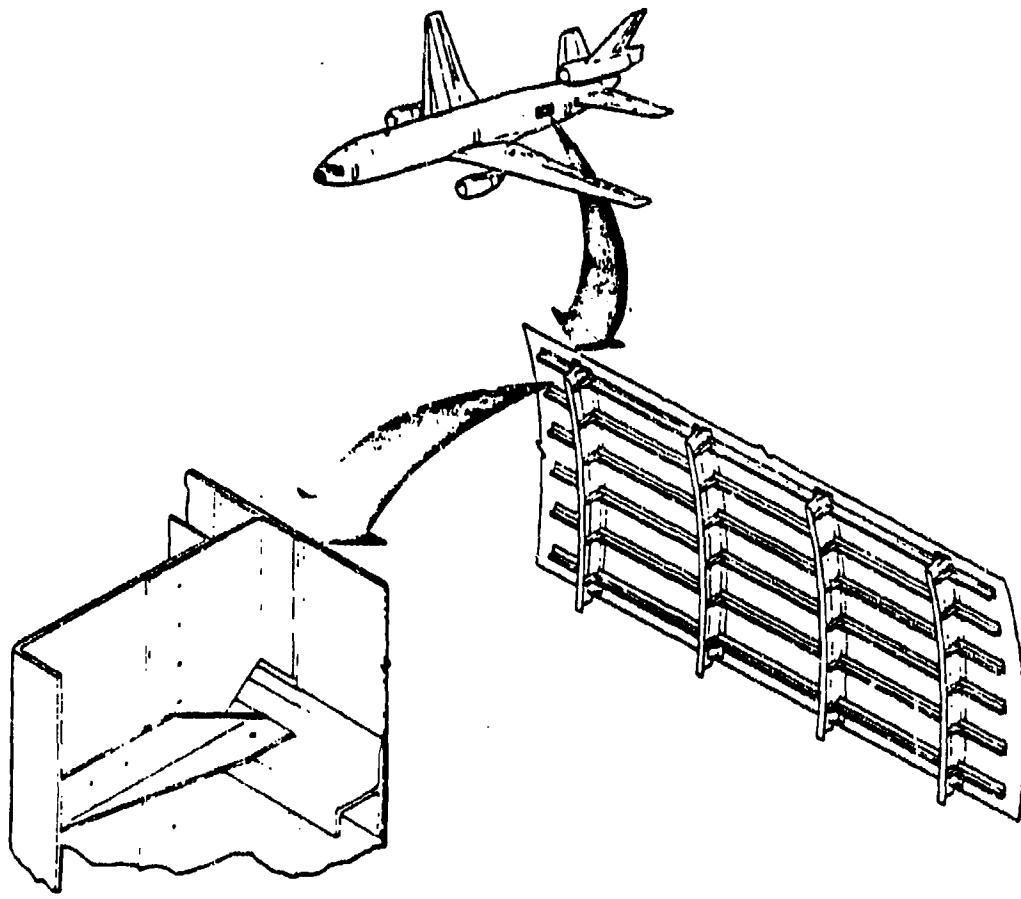


Figure 1. Sample Bolted Joints in the F/A-18A Aircraft Wing (Reference 1).



STIFFENER ATTACHMENT

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

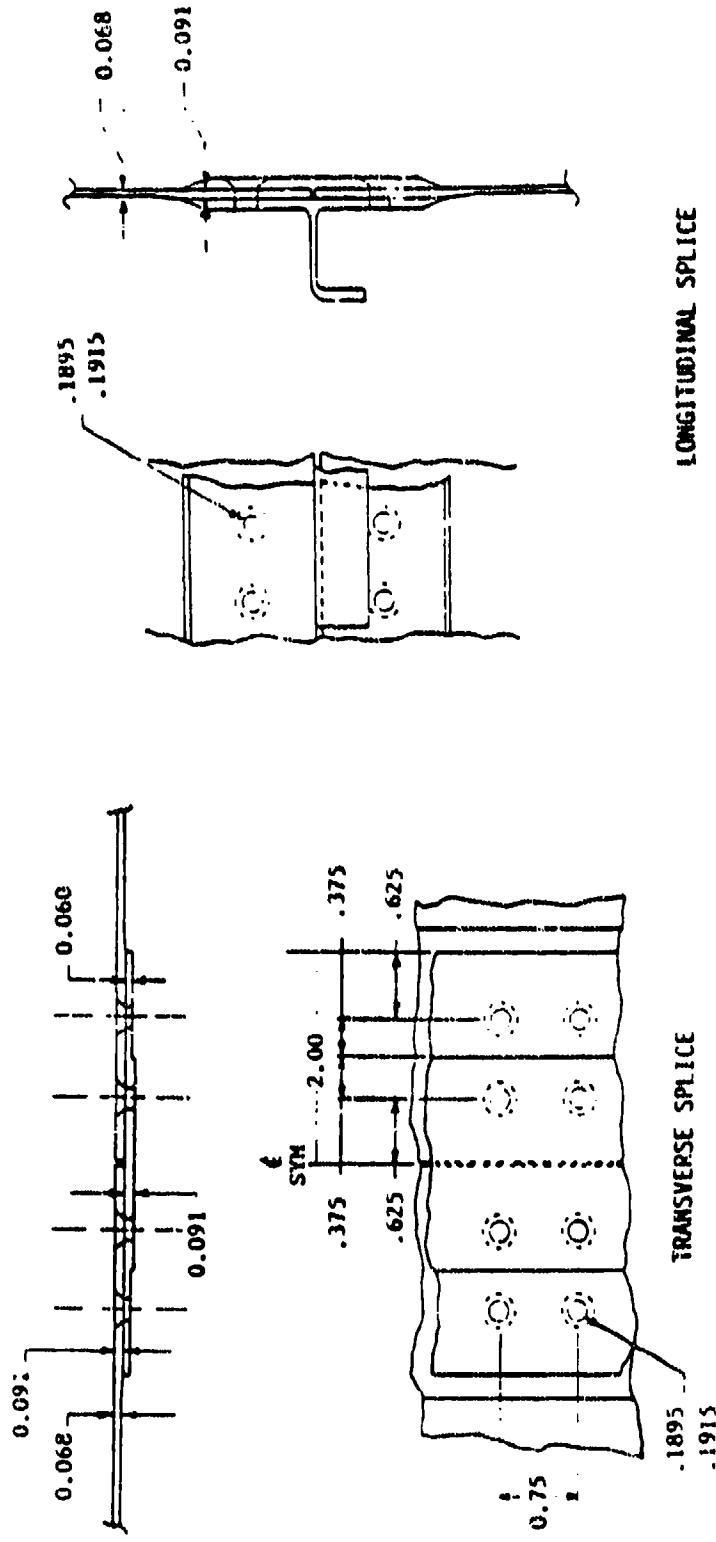


Figure 3. Bolted Joint Concepts for Composite Fuselage Structures (Reference 2).

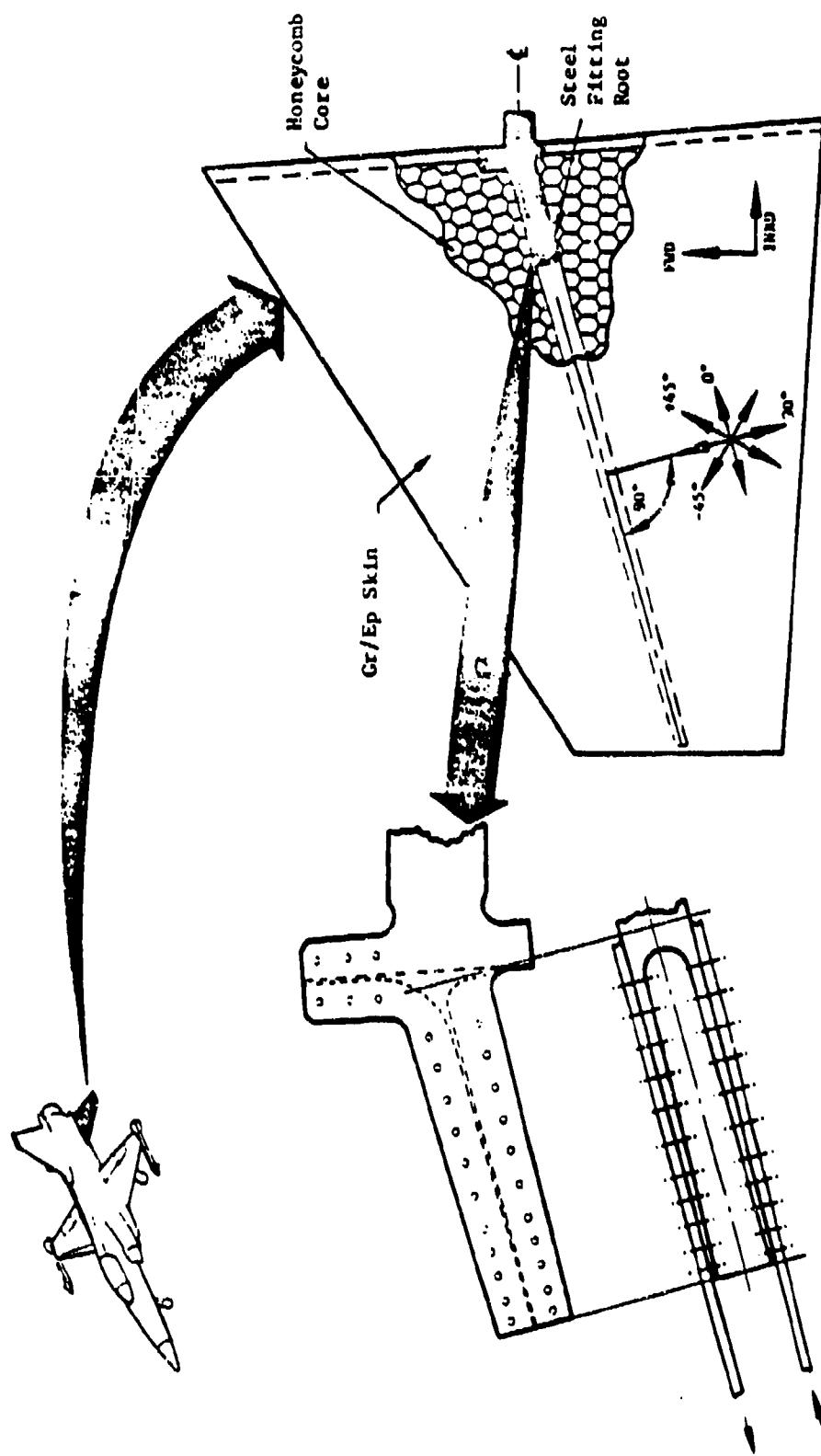


Figure 4. Skin-to-Root Fitting Joint in the F-20 Horizontal Stabilizer.

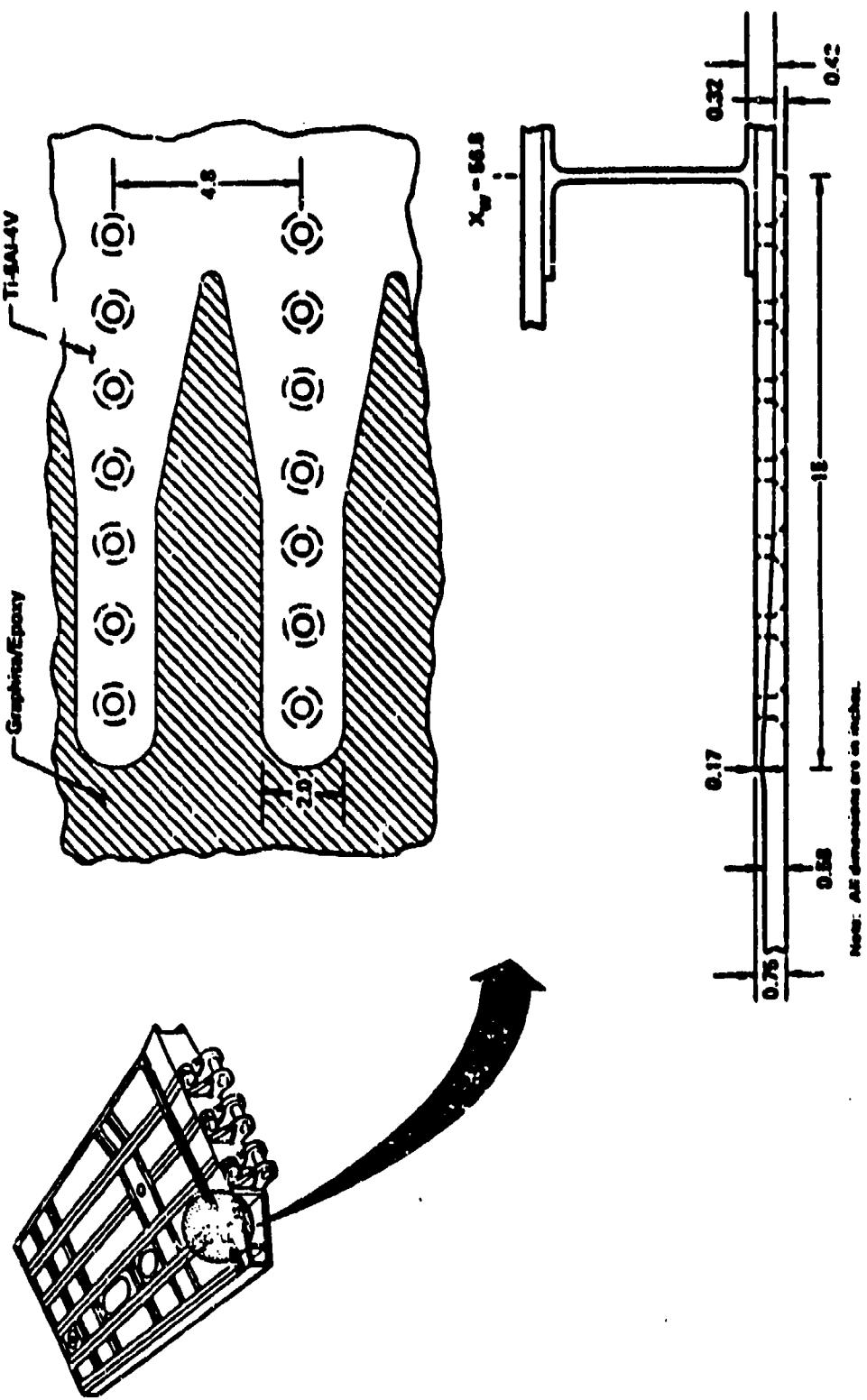


Figure 5 - Alternative Five Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section
(Reference 1).

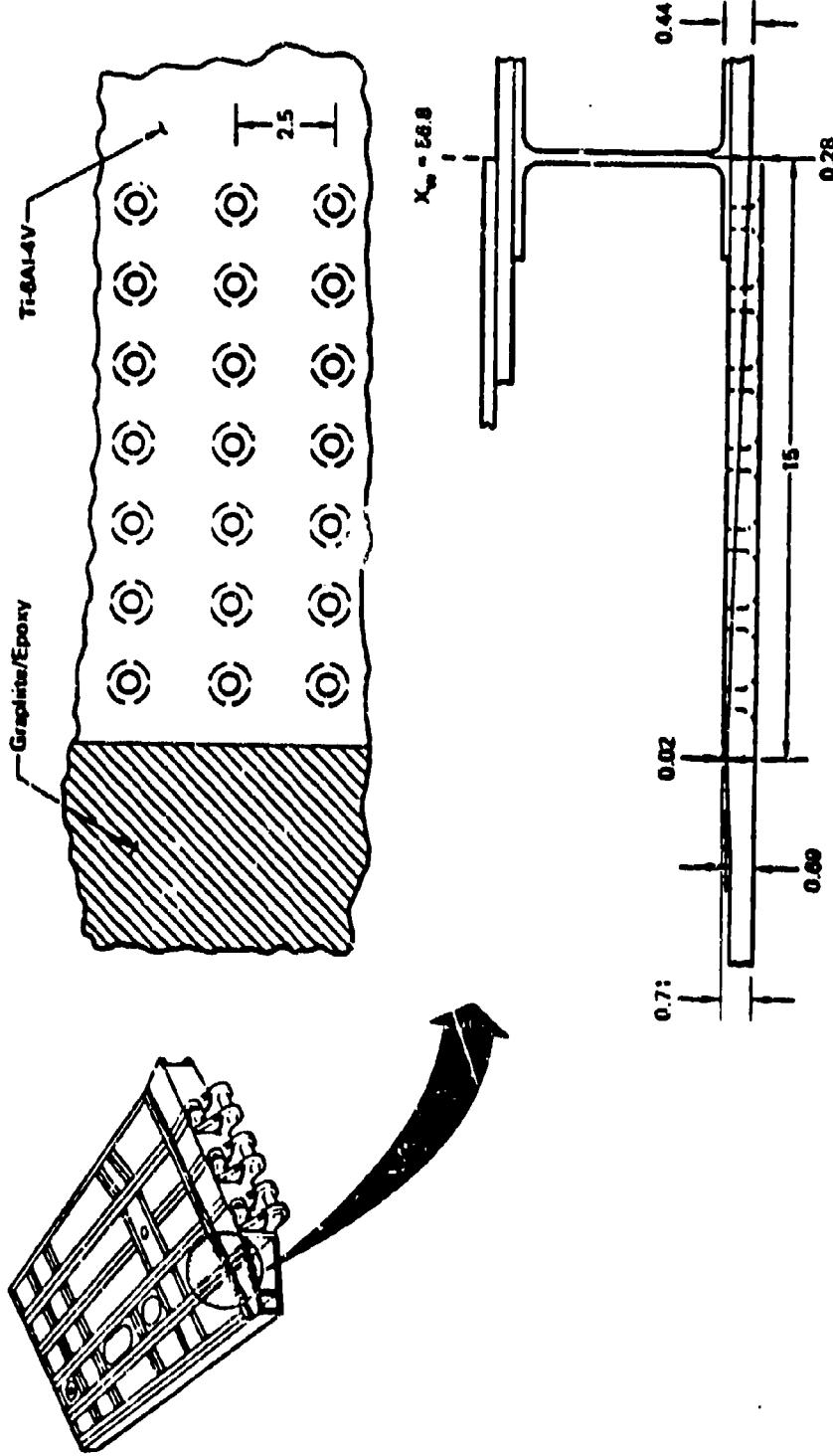


Figure 5 . Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section
(Reference 1; Concluded).

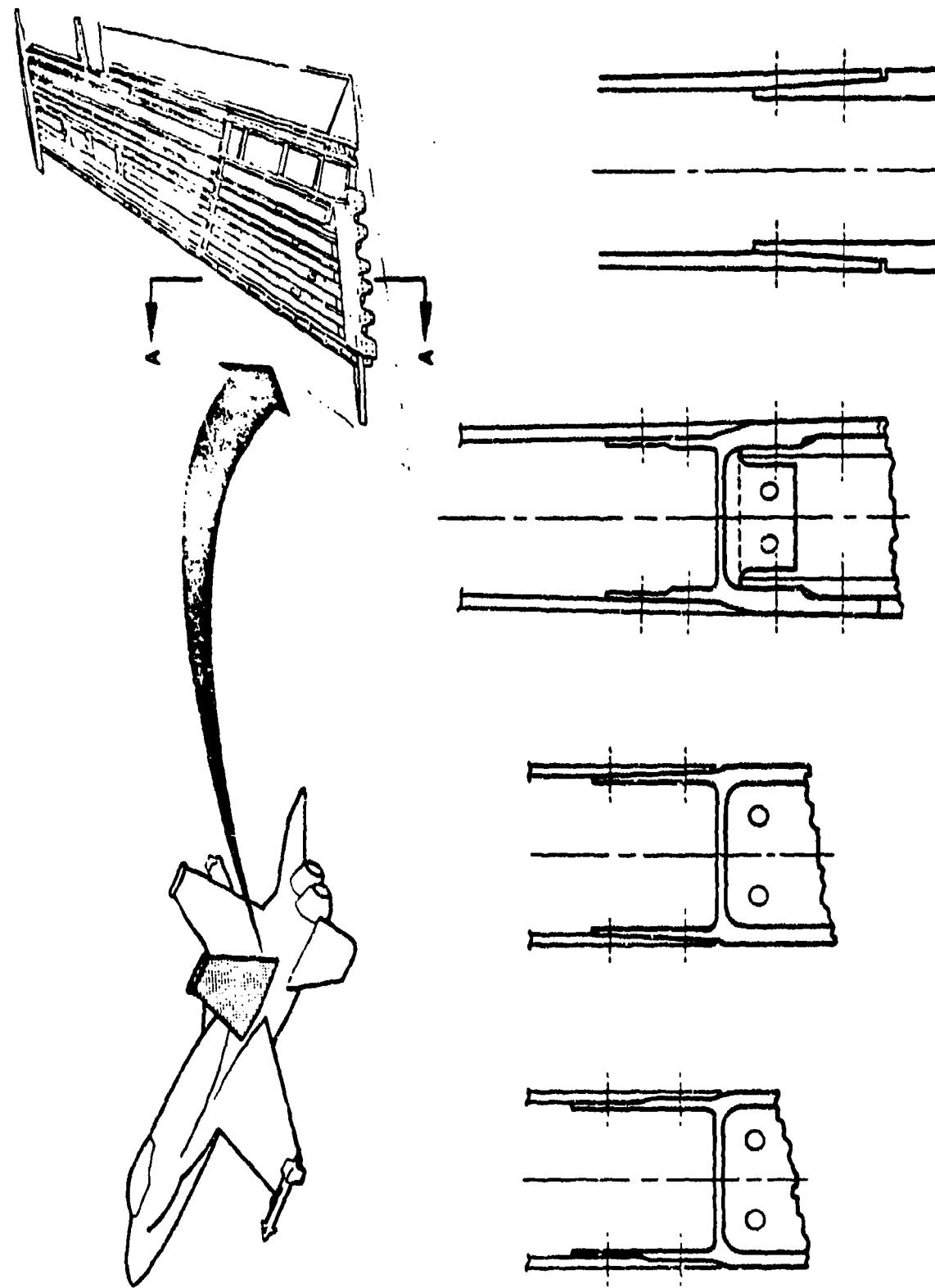


Figure 6. Alternative Bolted Joint Concepts Evaluated for the F/A-18A Vertical Tail Root Section (References 3 to 5).

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.

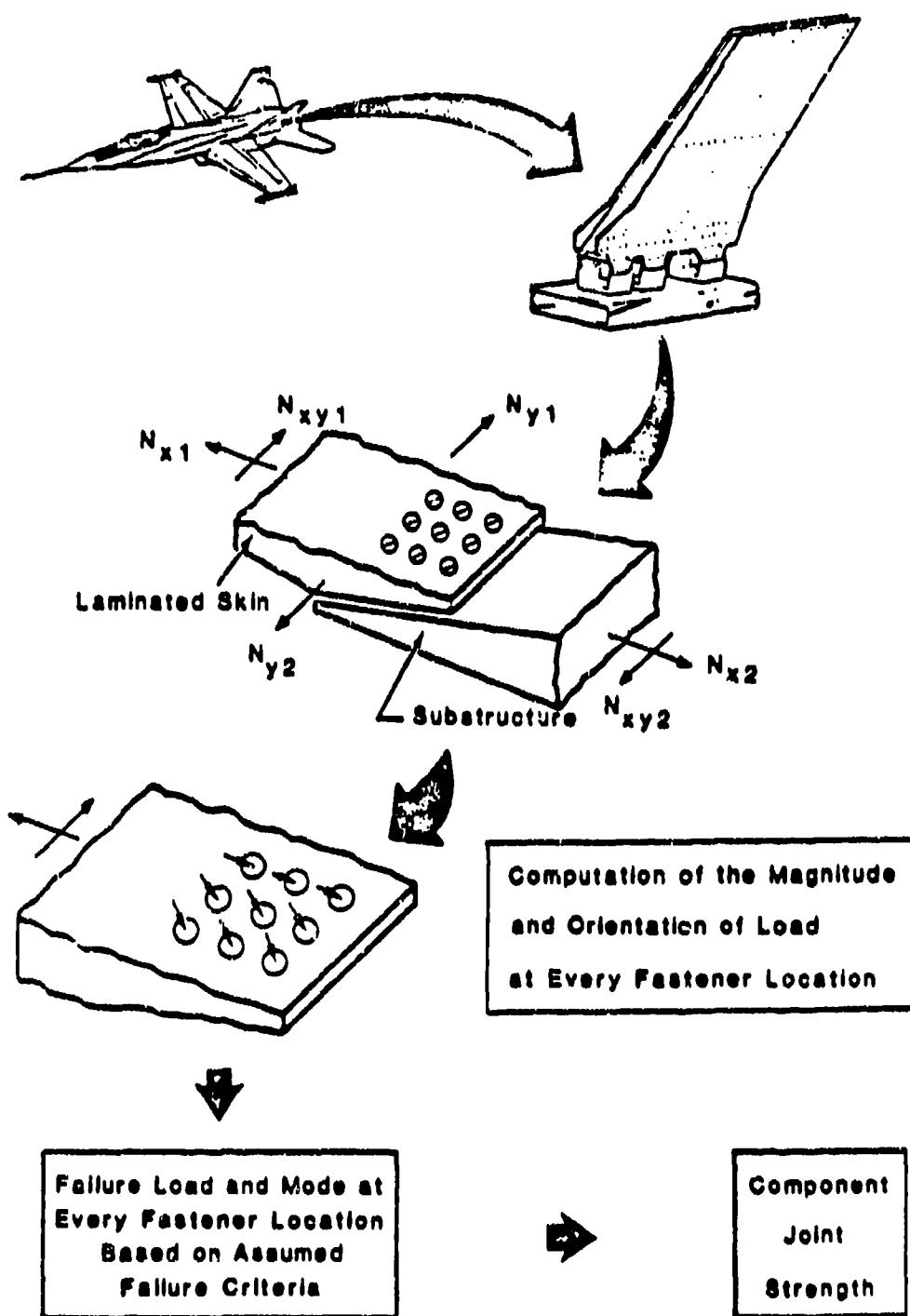


Figure 7. Overview of the Strength Analysis of Bolted Structures.

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 bolted 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 lightning 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

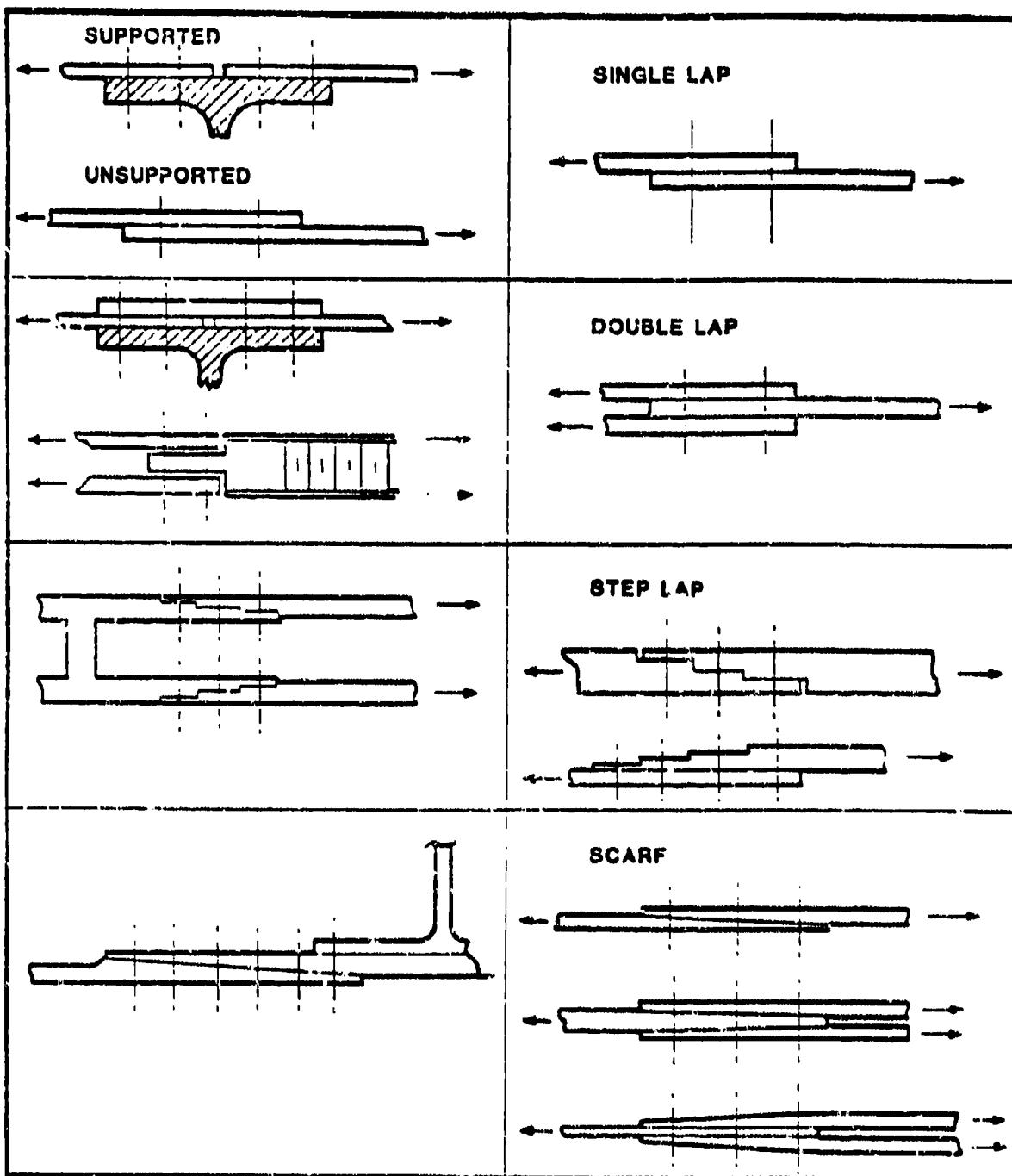
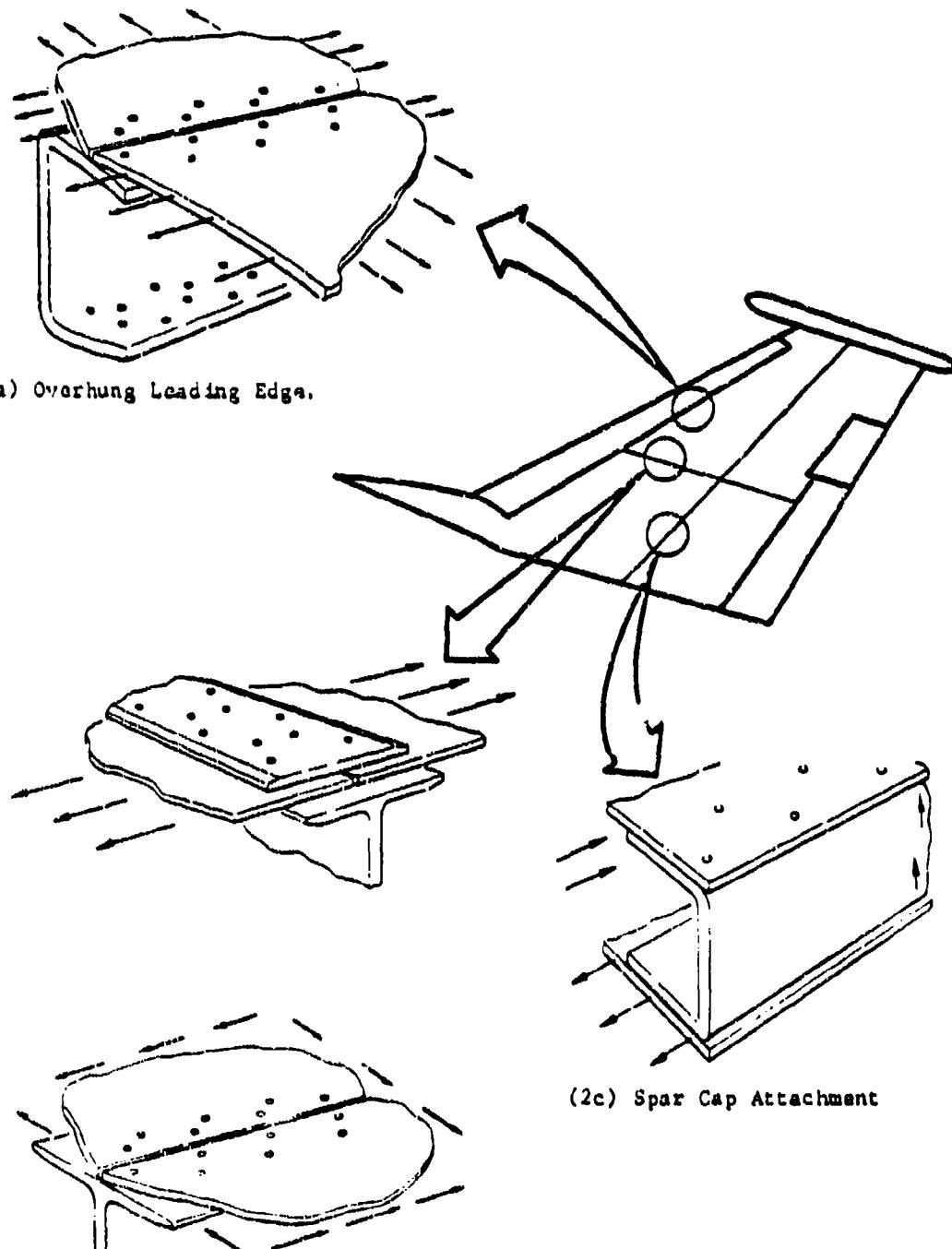
STRUCTURAL CONFIGURATION**ANALYZED CONFIGURATION**

Figure 8. Structural and Analyzed Bolted Joint Configurations.



(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.

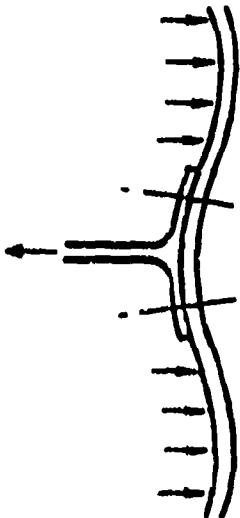
(5)

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

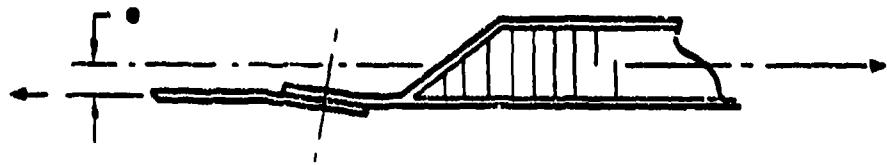
(6)

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.)



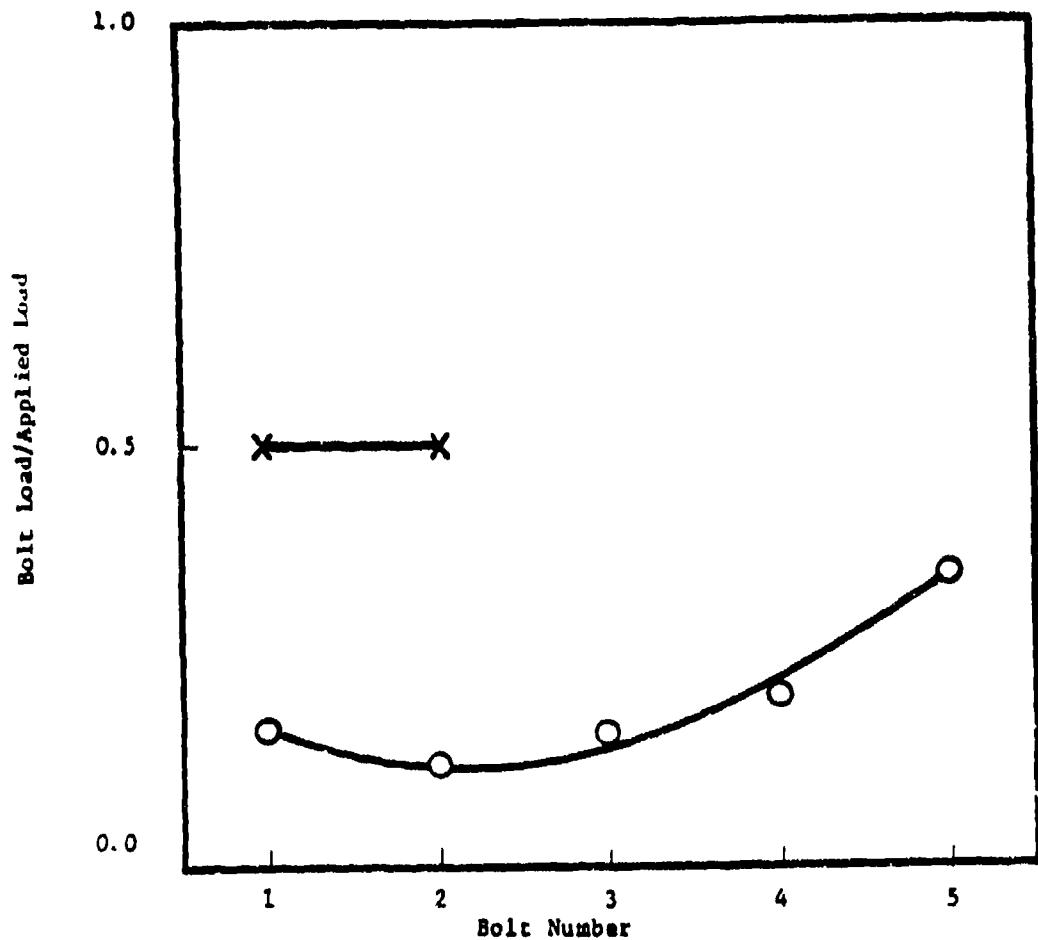
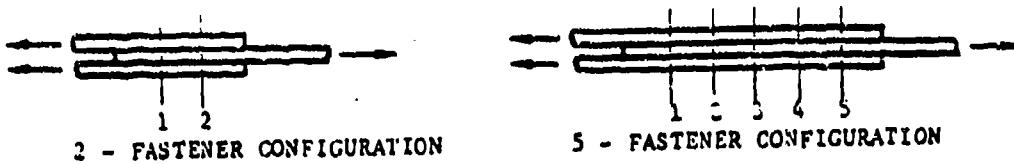
B. OUT-OF-PLANE JOINT LOAD DUE TO LOAD PATH ECCENTRICITY

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).

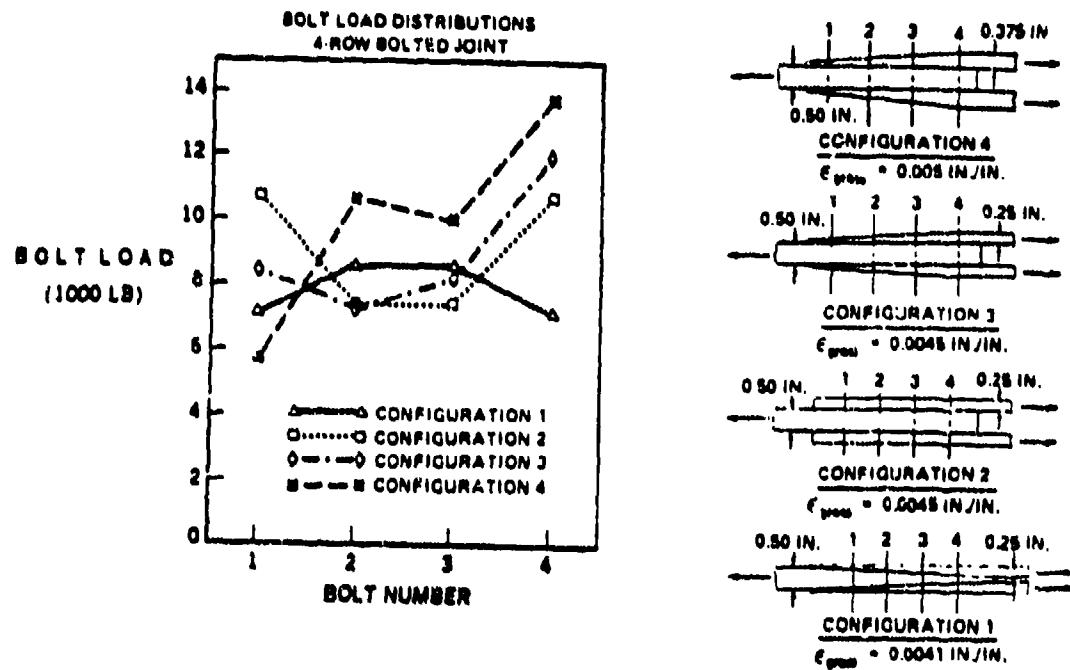


Figure 12. Effect of Joint Configuration on Fastener Load Distribution (Reference 9).

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)

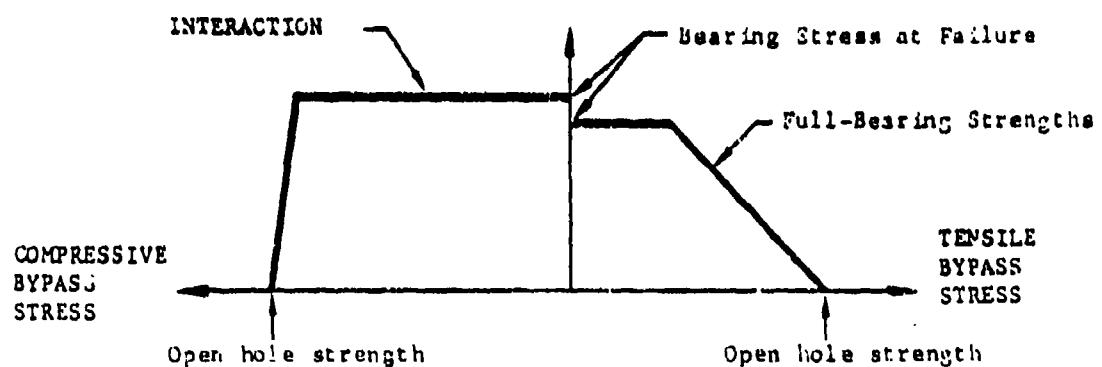
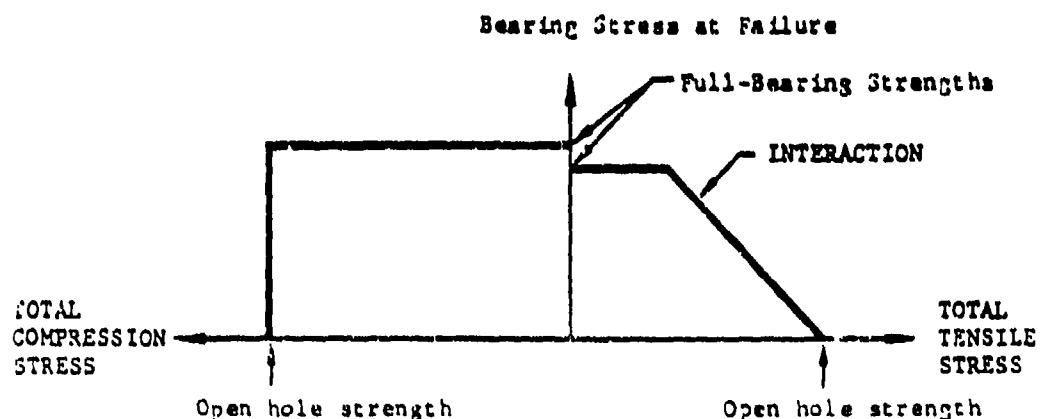
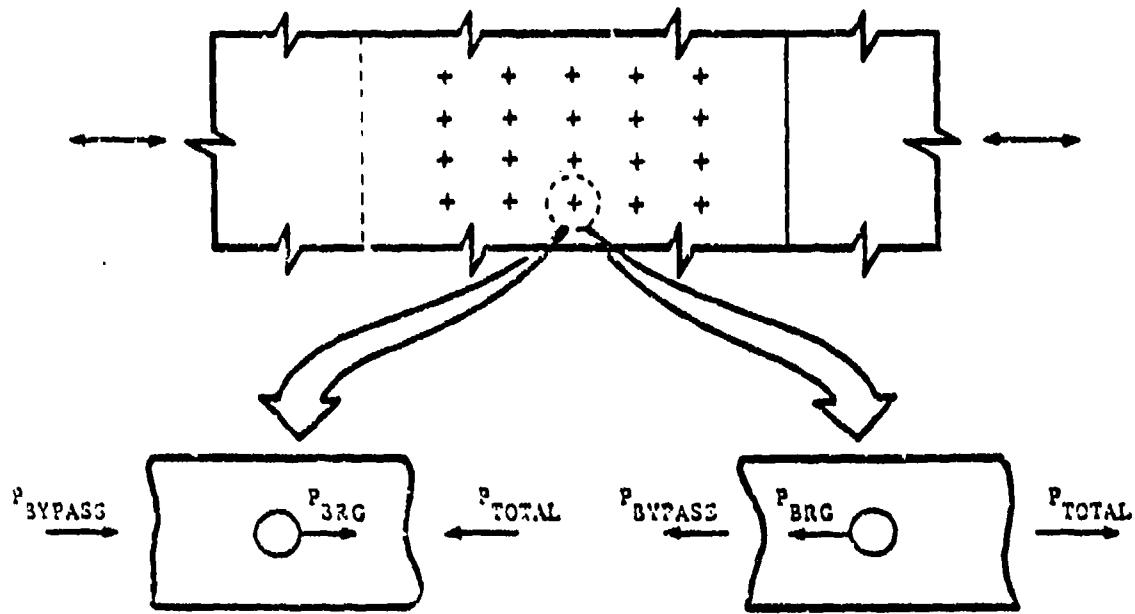


Figure 13. Interaction Between Bearing and by-Pass Loads at a Fastener Location.

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 precluded 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 modes 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).

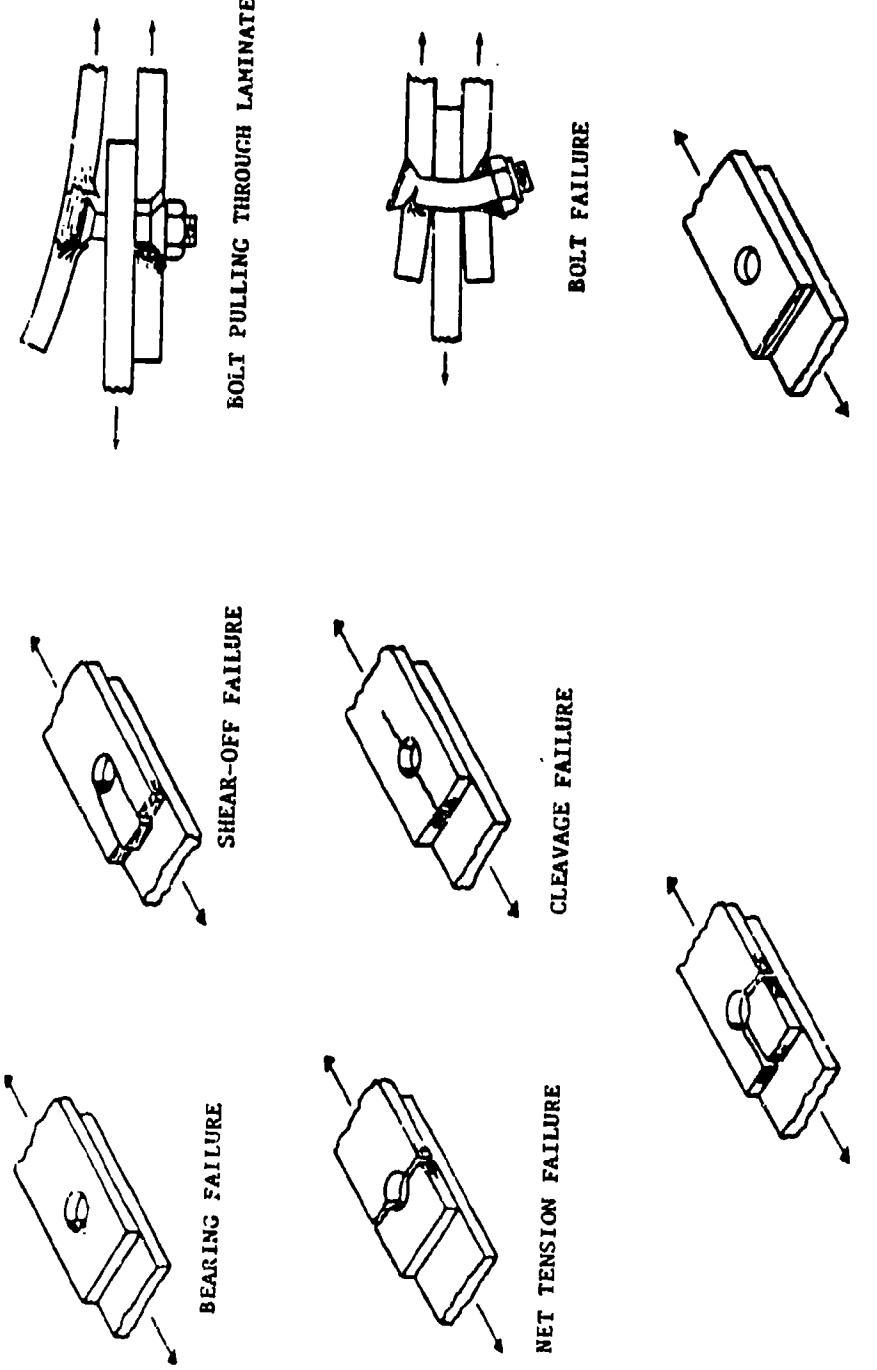


Figure 14. Basic Failure Modes in Bolted Laminates and Fastener-Related Failures.

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.

(1)

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, Cobalt 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 or Magnesium Alloys	Not Compatible

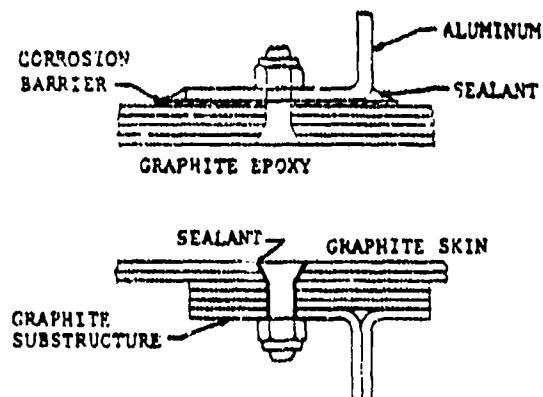


Figure 15. Galvanic Compatibility and Corrosion Prevention.

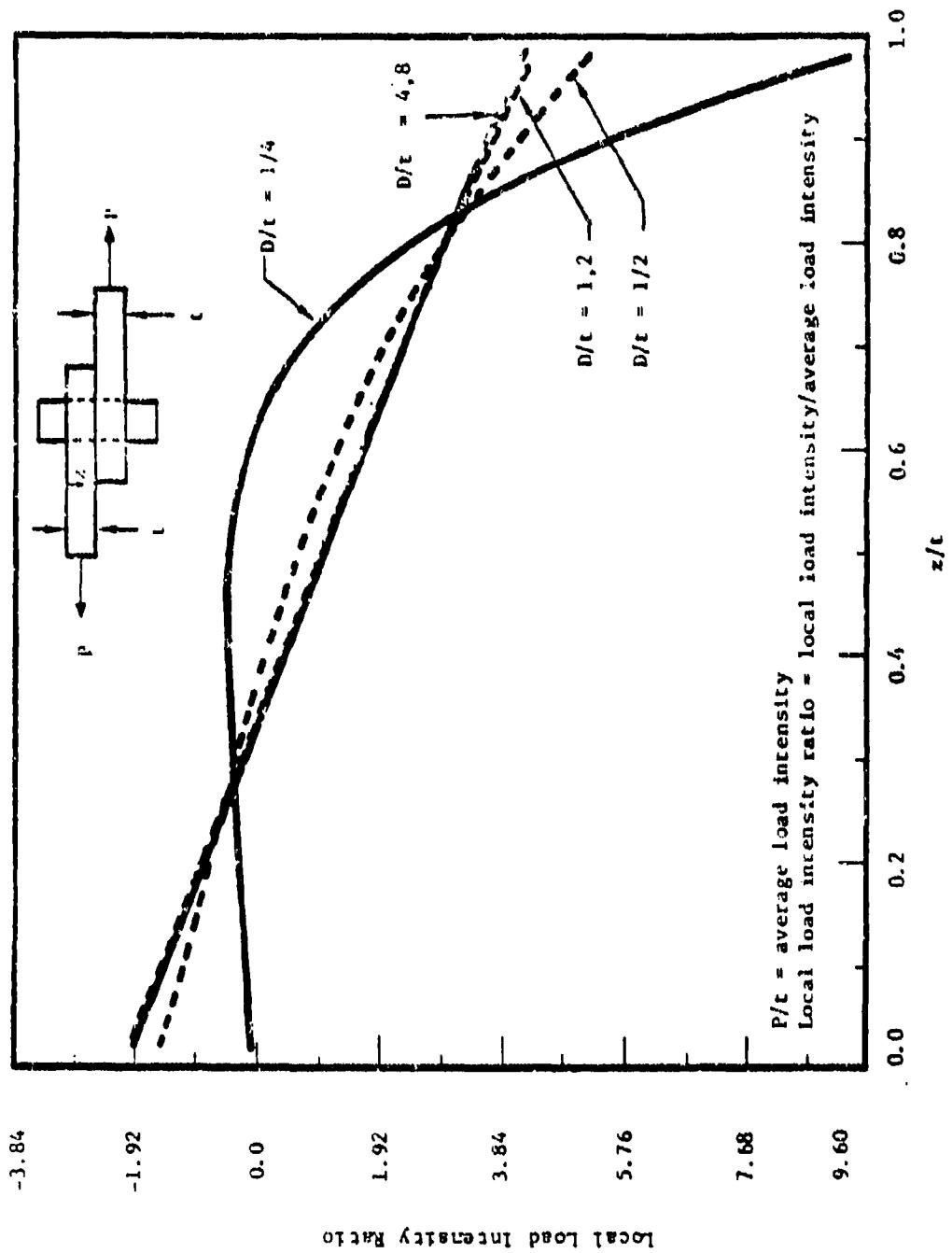


Figure 16. Effect of Fastener Size on the Load Distribution (Reference 6).

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 11).

	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	-32.6	<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 OPTIMAL)				
• 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)				
• 60/40/10 • 0.008	-	-	<2.0	+14.7
• 0.008	-	-	<2.0	+11.2
• 30/60/10 • 0.008	-	-	<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 - 0.862 Moisture by Weight.

were conducted in Reference 13, varying the fastener torque-up value from 0 in-lb to 150 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

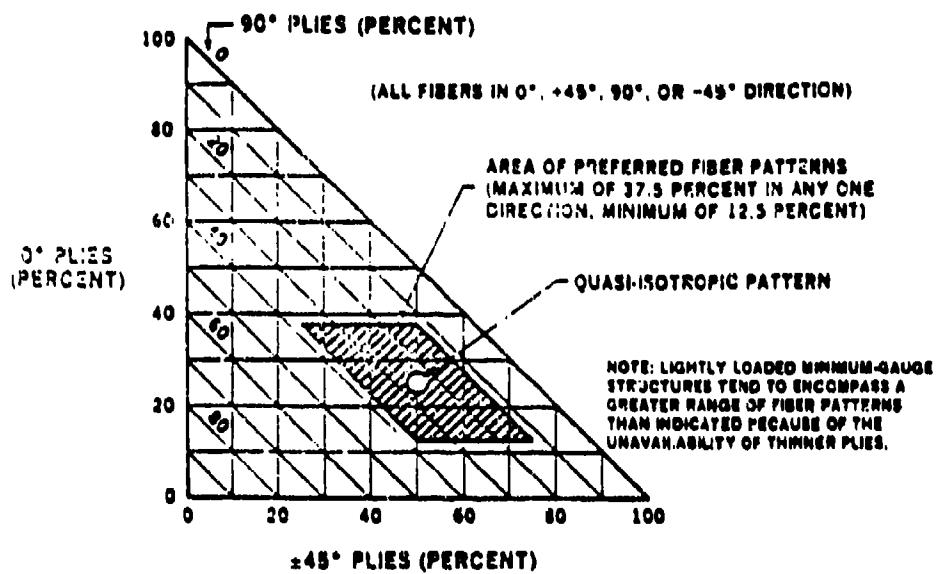


Figure 17. The Envelope of Bolted Laminate Layups for Realizing a Bearing Mode of Failure and the Maximum Bearing Strength (Reference 10).

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 10% 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

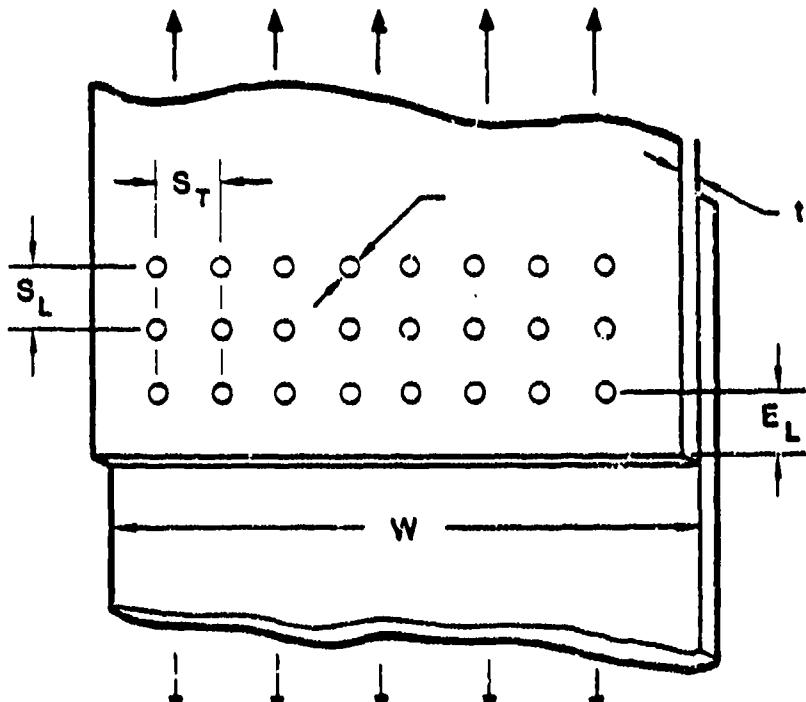
Fastener Spacing and Arrangement

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).

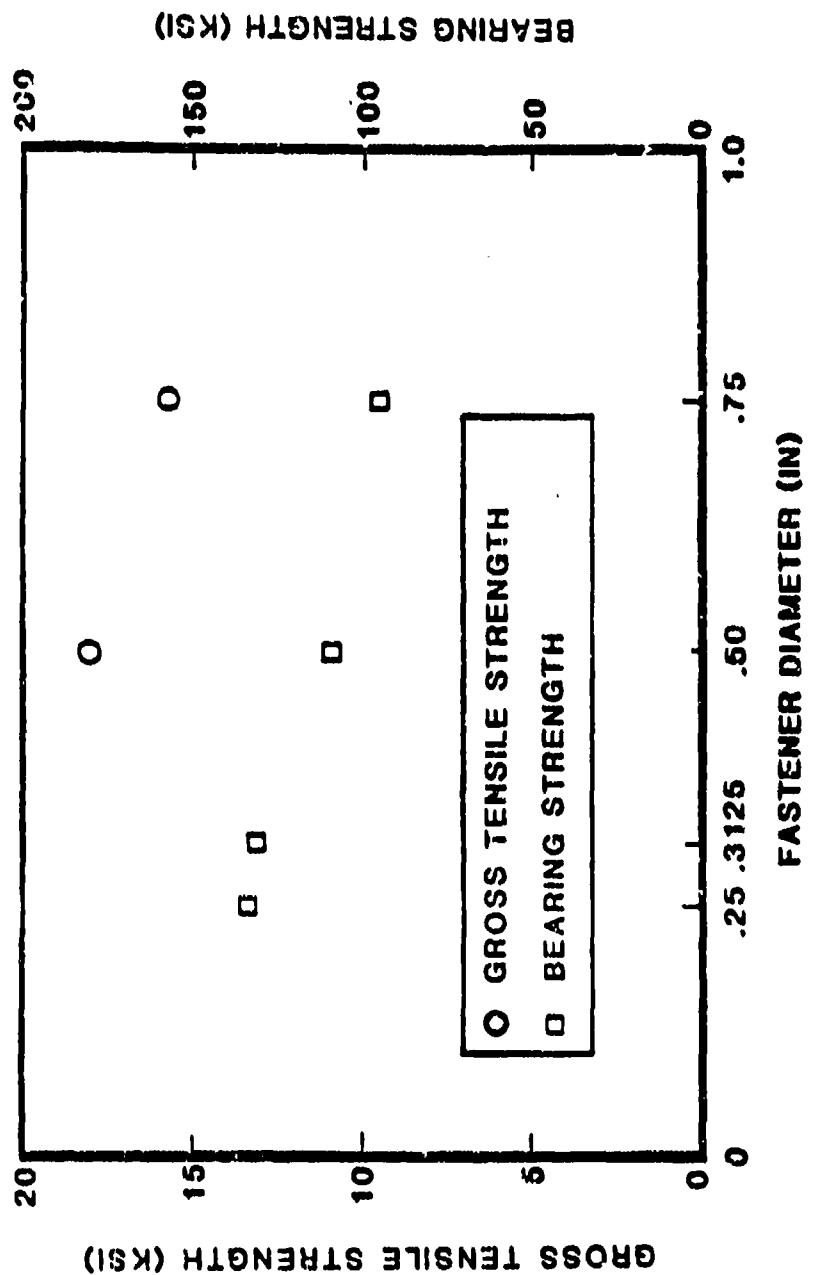
(22)

The bearing stress at failure decreases



PREDOMINANT LOAD DIRECTION

Figure 18. Geometrical Parameters for a Bolted Plate.



NOTE: 20-Ply, 50/40/10, AS1/3601-8 Layup; 0.31 in. Aluminum Plate;
 Torque = 100 in-lbs, E/D = 3, W/D = 8, 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_T/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).

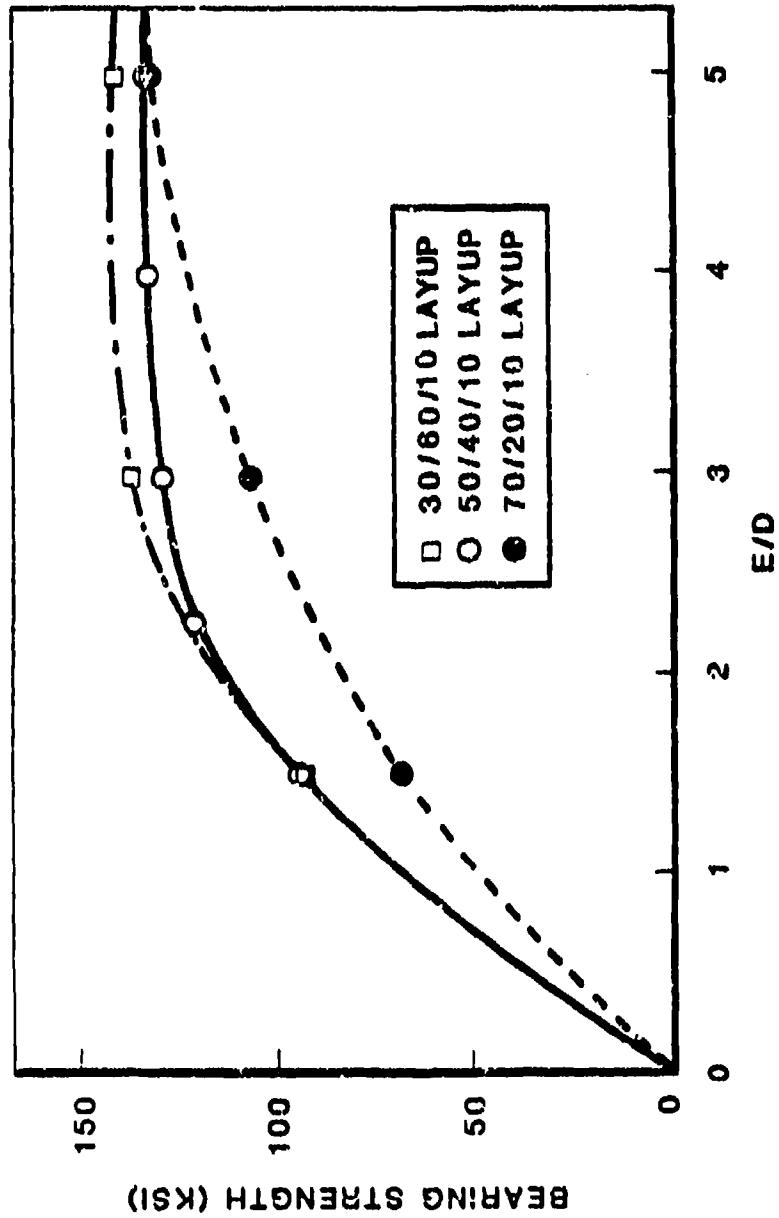
(25)

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)

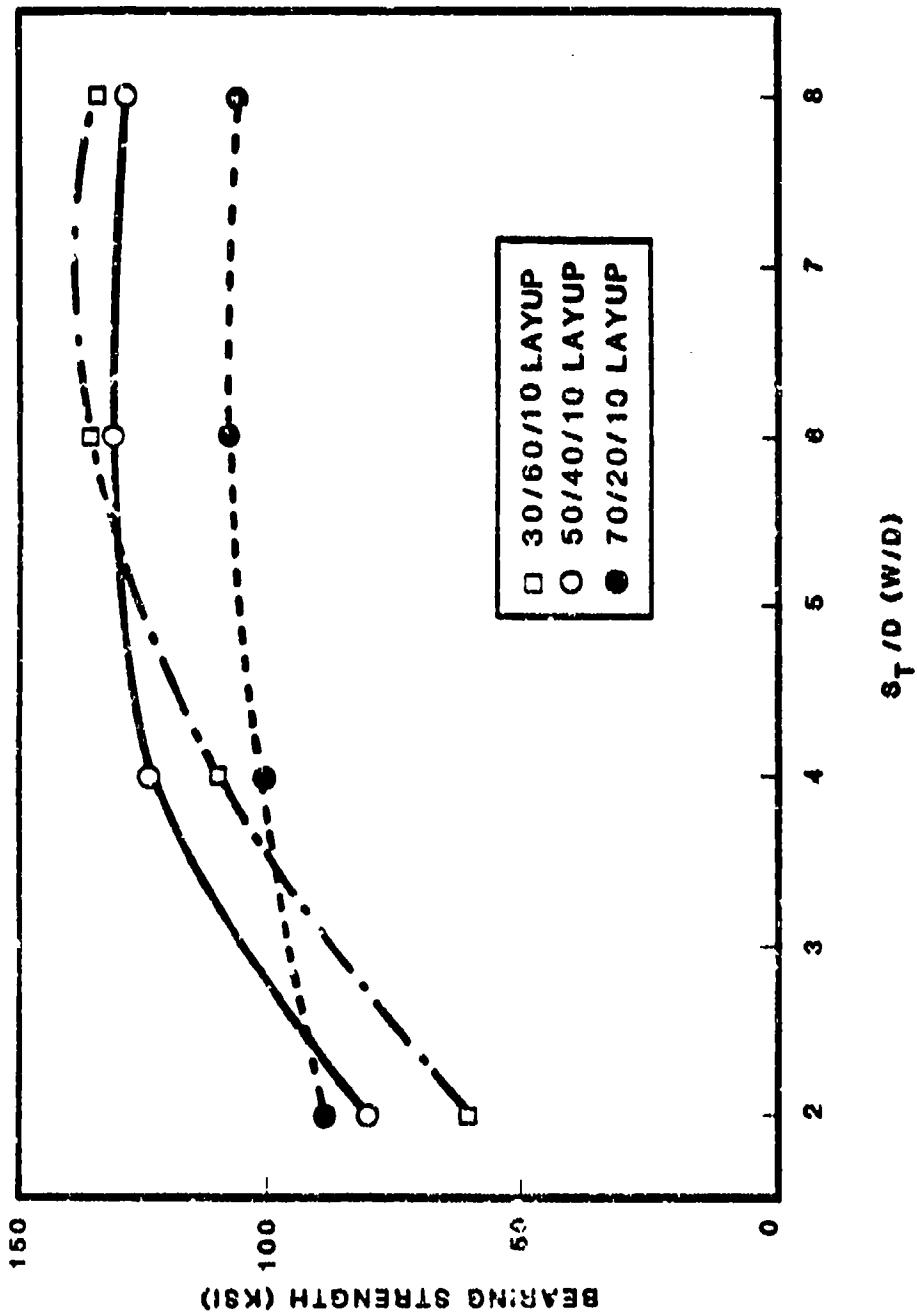
2.8 Joint Tailoring for Maximum Efficiency

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



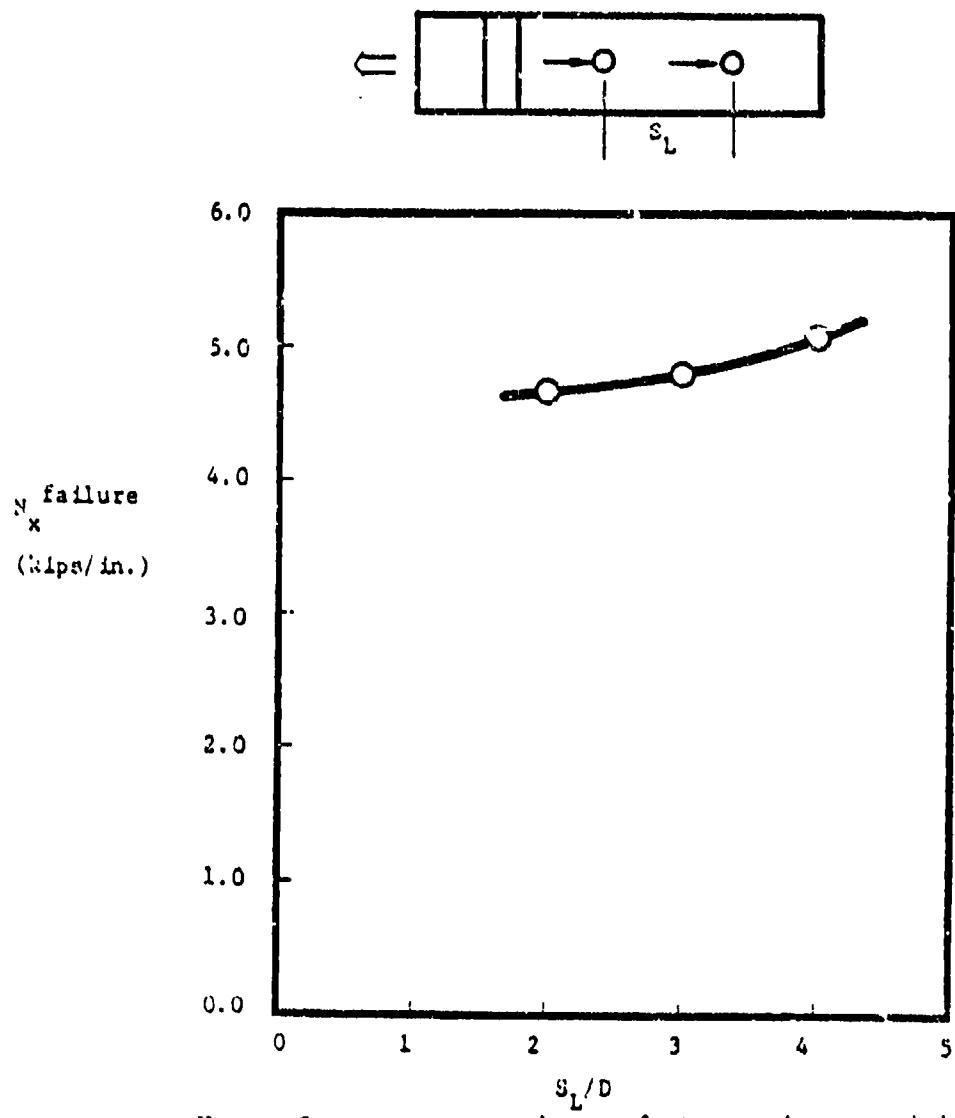
NOTE: Composite-to-metal joints in single shear; 20-ply AS1/3501-6 layups;
0.31 in. aluminum plate; W/D = 6; Torque = 100 in-lbs; Protruding Head,
steel fastener; D = 6/16 in.

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



NOTE: Composite-to-metal joints in single shear; 20-ply, A91/3601-6 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: Composite-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; $S_T/D = W/L = 6$; 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.

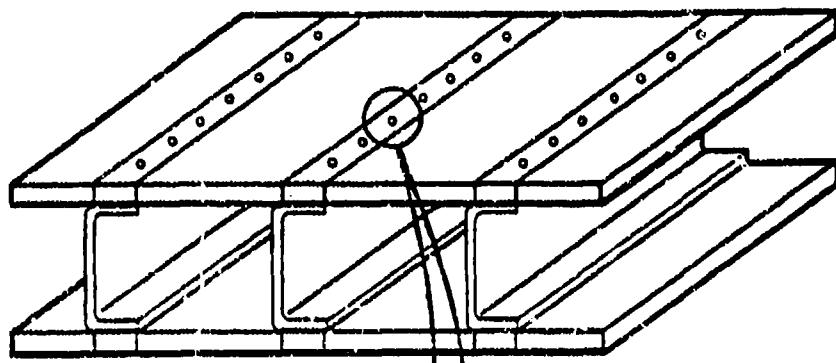
should be controlled to aid producibility and maintainability of the structural joint.

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).

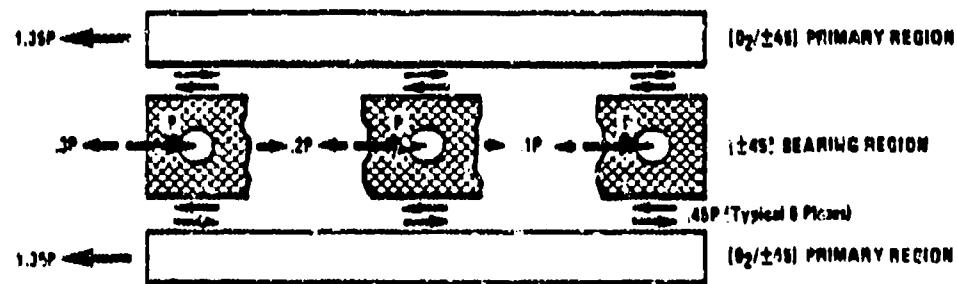
Some experimental concepts have also been demonstrated to be efficient joint tailoring concepts, despite the difficulty they introduce in applying the concept at the production level. An example is shown in Figure 23, where the 0 degree plies in the bolted skin are replaced by + and -45 degree plies 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)



TAILORED BOLTED JOINT



CONVENTIONAL BOLTED JOINT

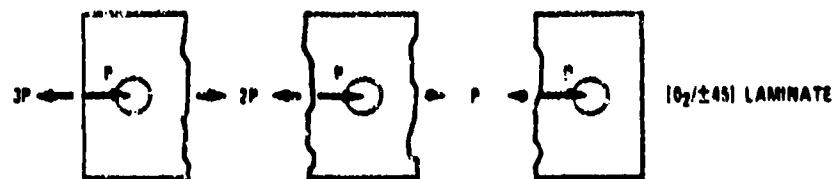


Figure 20. A Sample Tailored Joint.

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.

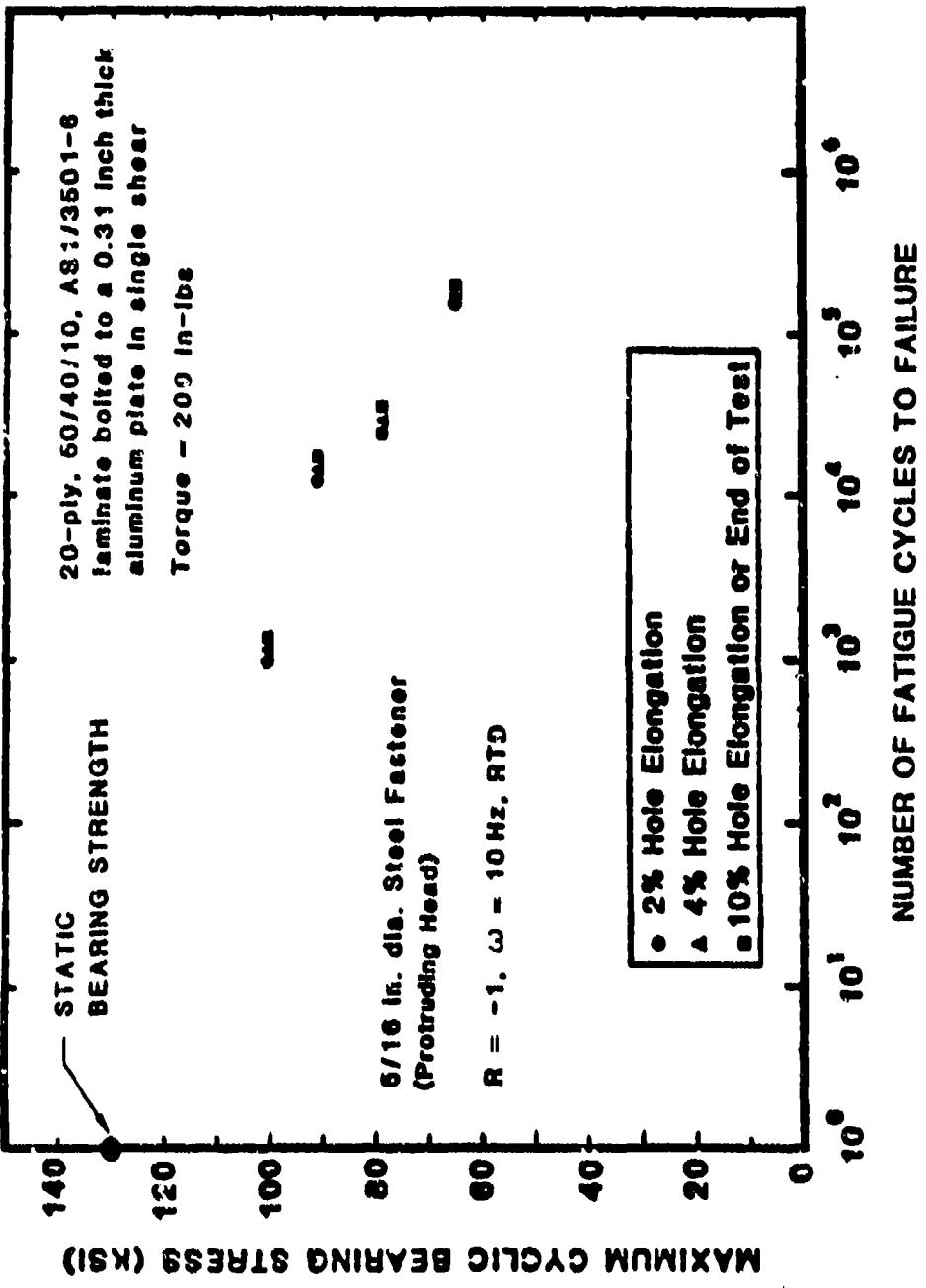


Figure 24. Effect of Maximum Cyclic Bearing Stress on the Number of R-1 Fatigue Cycles to Cause Specified Hole Elongations in a Bolted Laminate.

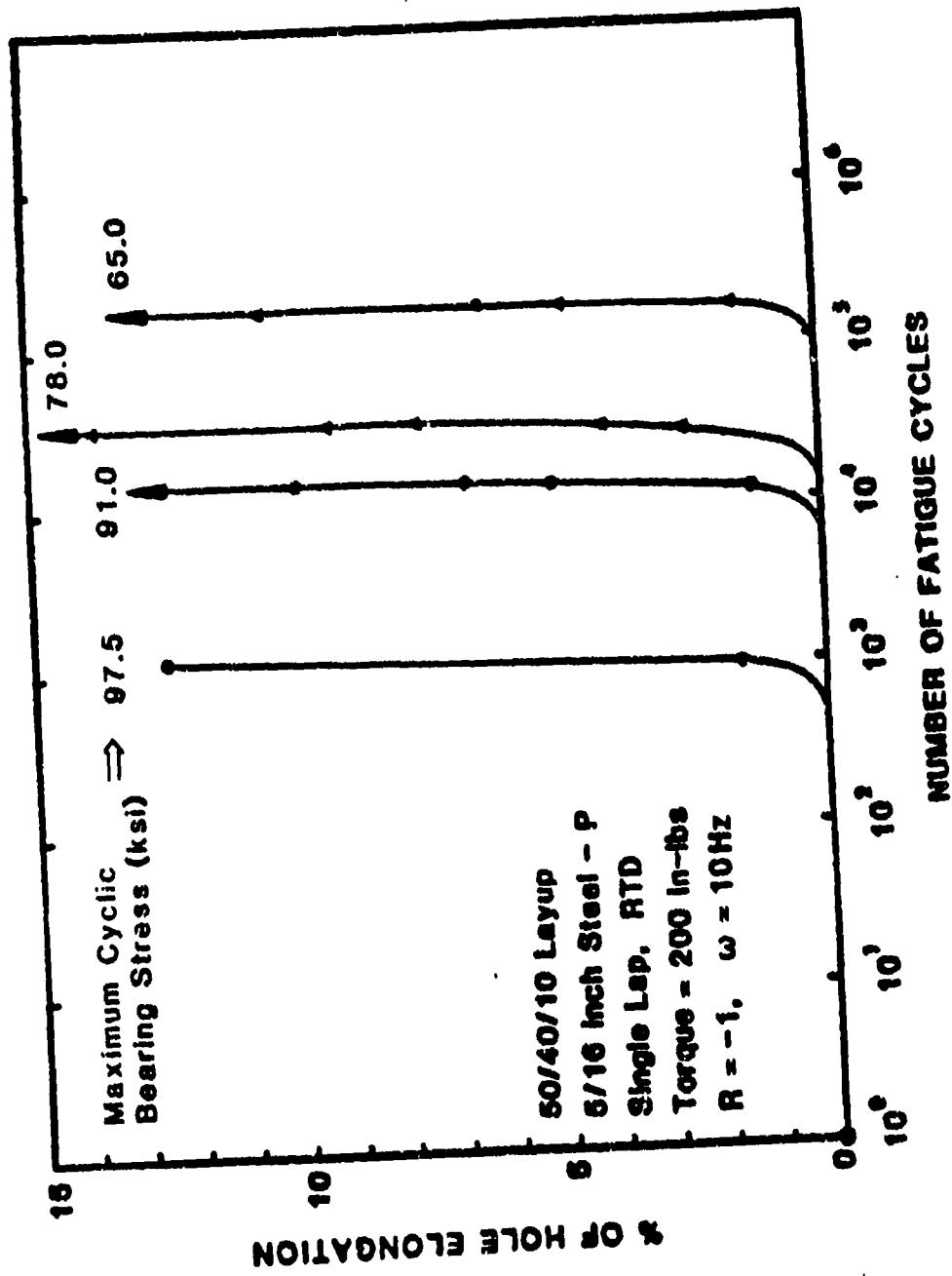


Figure 25. Effect of Maximum Cyclic Bearing Stress on the Hole Elongation Rate for
 a Bolted Laminate under $R=1$ Loading.

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

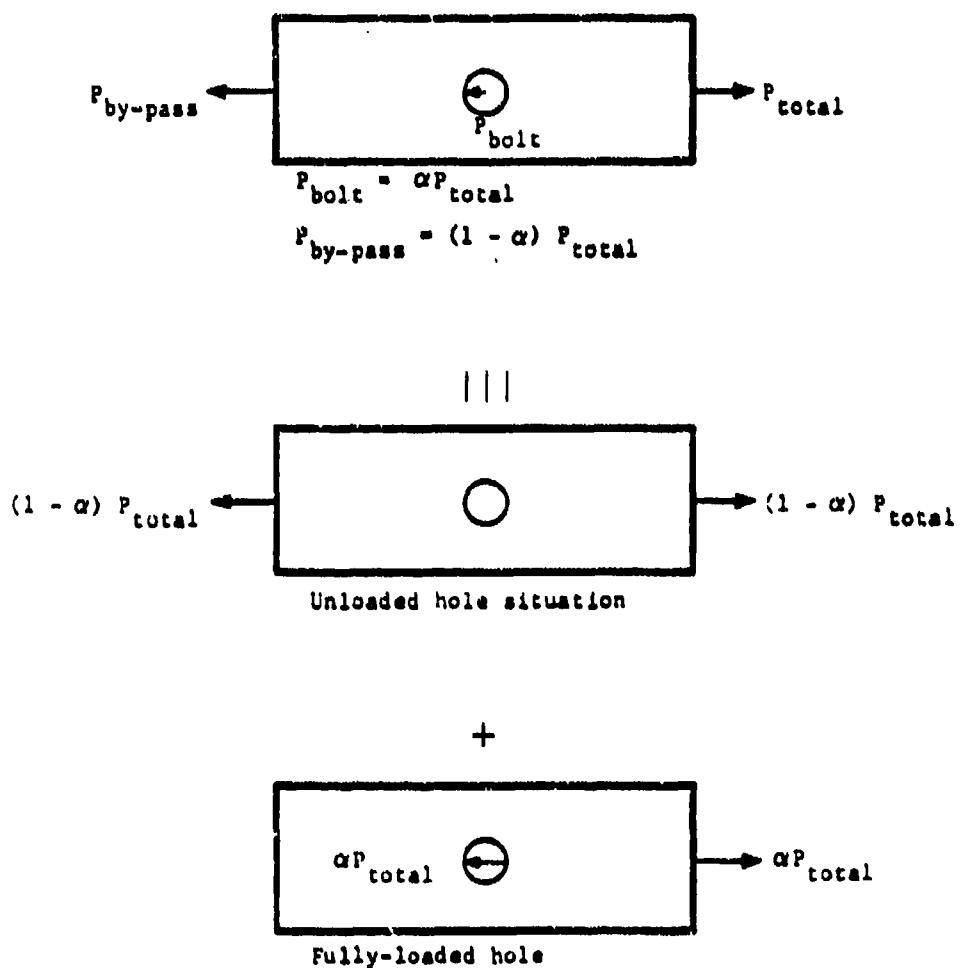


Figure 26. Schematic Representation of a General Single-Fastener Situation as a Superposition of Unloaded and Fully Loaded Hole Situations.

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

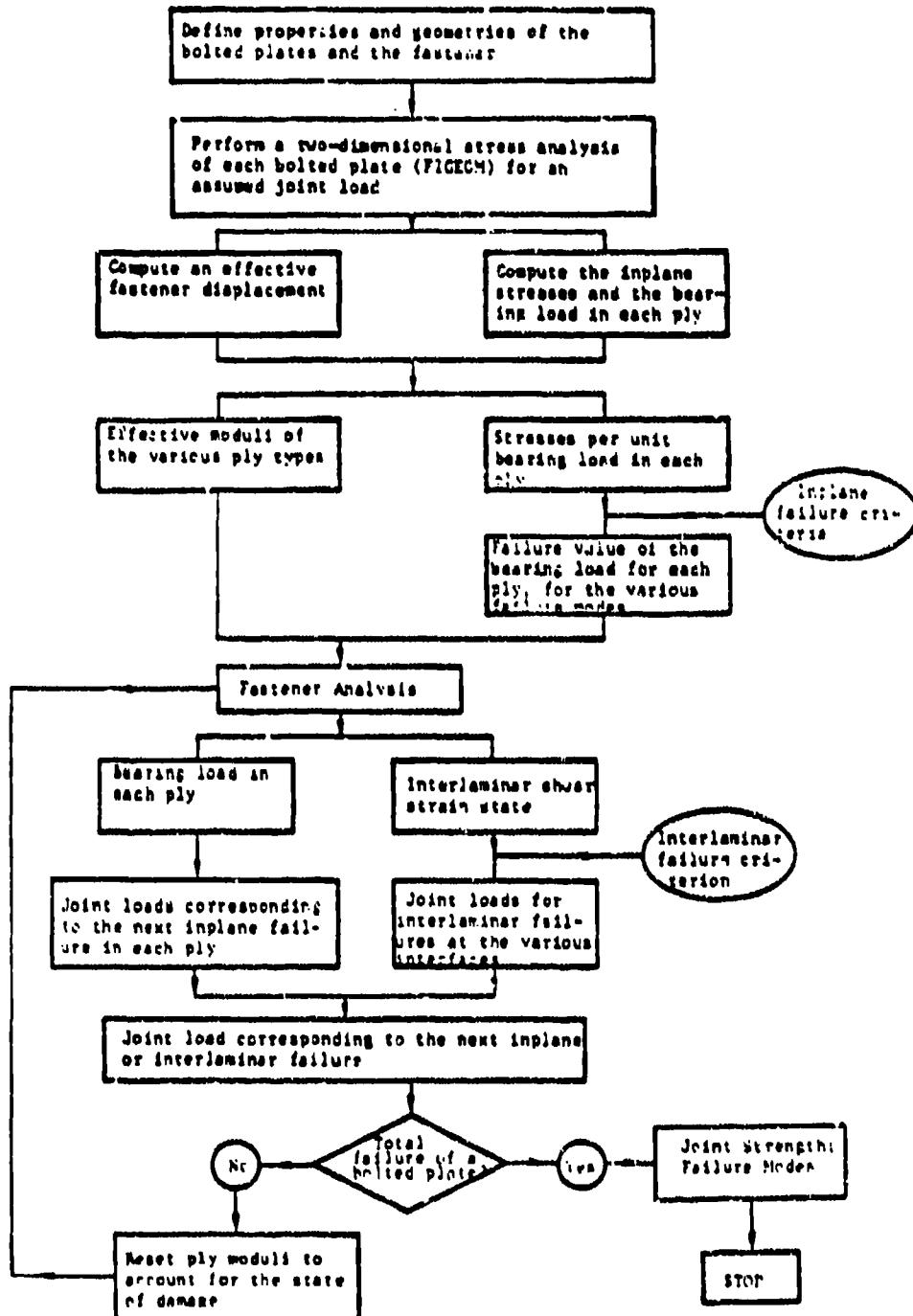


Figure 27. Flowchart for the Strength Analysis of Laminates with Fully Loaded Holes.

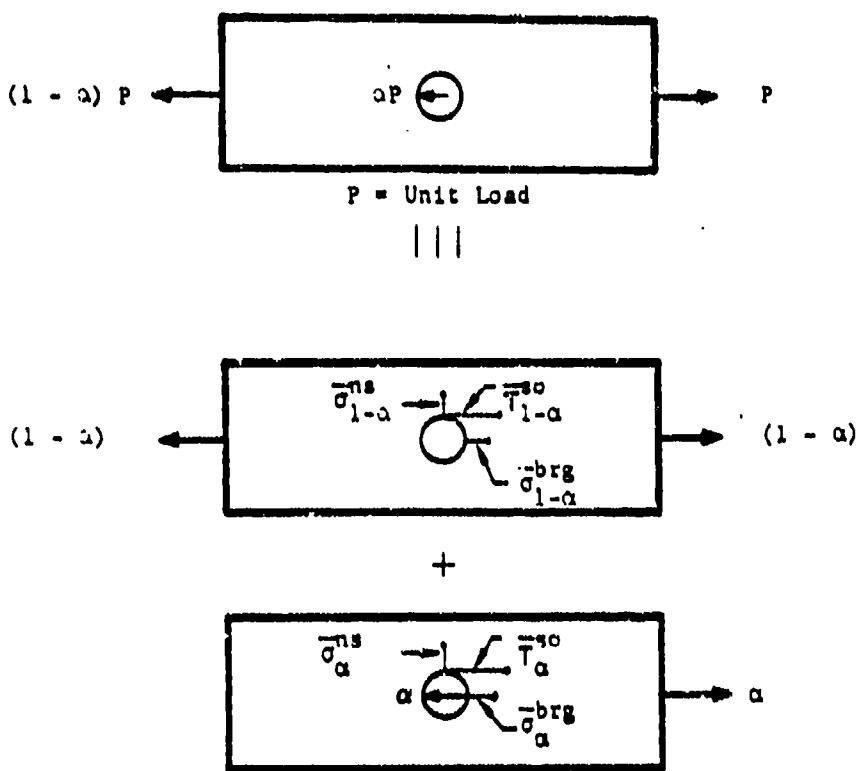
failures, determines the first failure in a bolted plate and the corresponding joint load value.

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 SASCI computer code is restricted to protruding head fasteners, and assumes that fastener failure is precluded. However, when a countersink fastener is specified, SASCI 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,



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

$$\bar{\tau}^{\text{so}} = \int_0^{d_{\text{osso}}} \tau_{xy}(x, D/2) dx = \bar{\tau}_{1-a}^{\text{so}} + \bar{\tau}_a^{\text{so}}$$

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

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 criterion, and (5) Tsai-Hill criterion. The first two criteria predict three modes of failure in each ply--net section, shear-cut 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-cut 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 $\sigma_x (0, D + a_{ons})$ exceeds the unnotched tensile or compressive strength of the ply, as appropriate, a net section ply failure is predicted. When $\sigma_x (D + a_{obrg}, 0)$ exceeds the unnotched compressive strength of the ply, a bearing mode of ply failure is

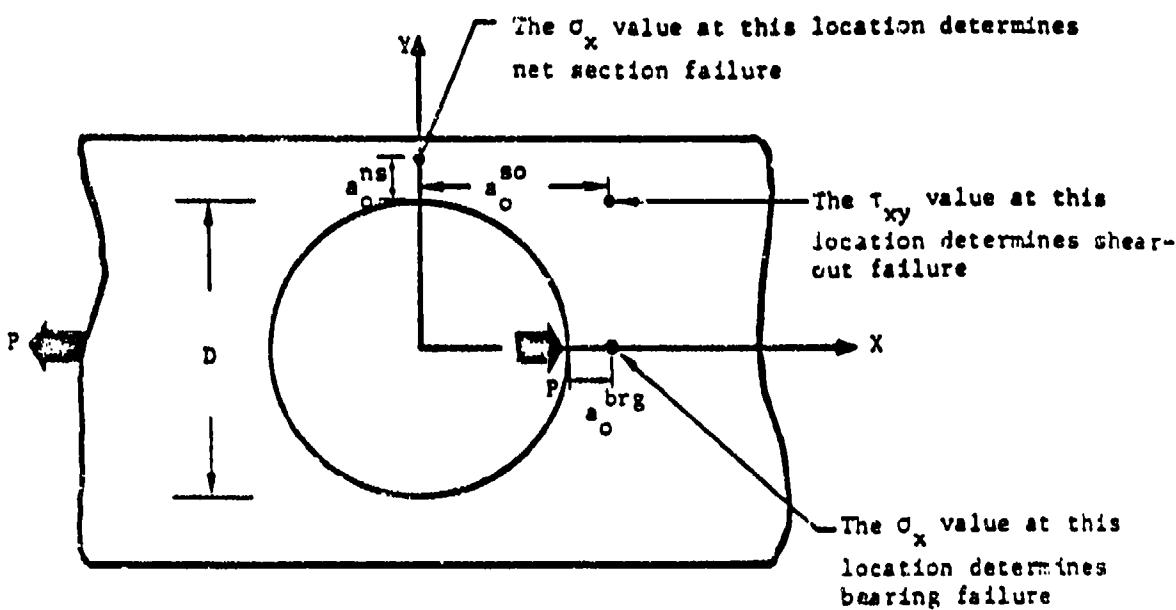


Figure 29. The Characteristic Distances used in the Point Stress Failure Criteria

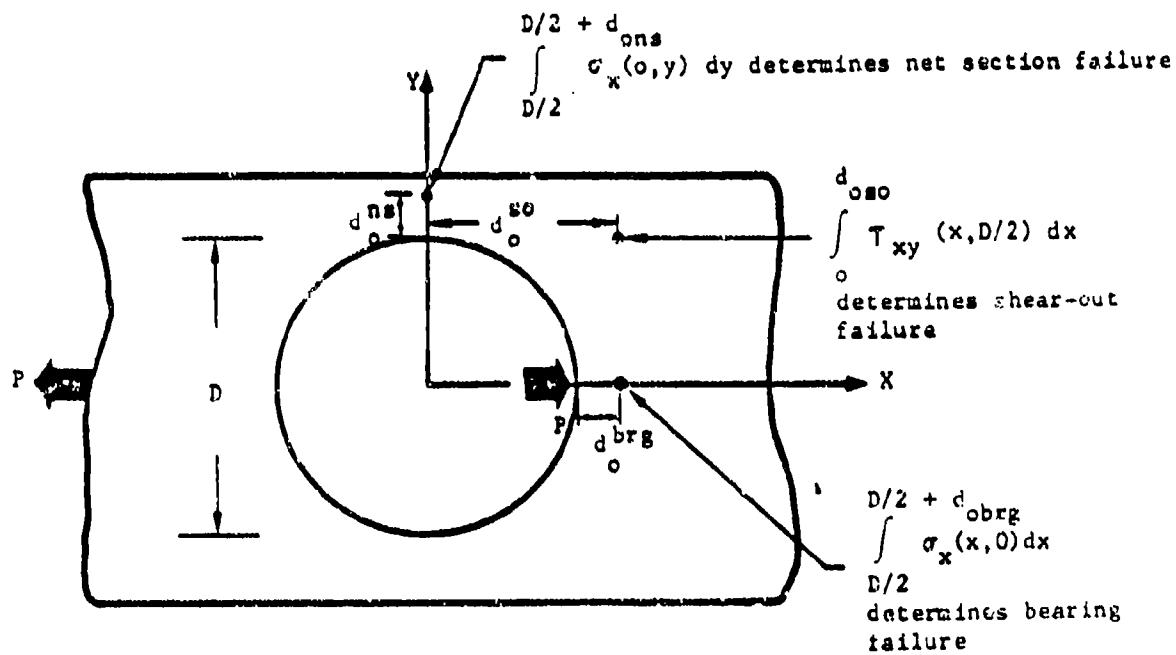


Figure 30. The Characteristic Distances Used in the Average Stress Failure Criteria.

predicted. When τ_{xy} (a_{050} , $D/2$) exceeds the unnotched ply shear strength, a shear-out mode of ply failure is predicted. The average stress failure criterion predicts these failures based on averaged values of the mentioned stress components over selected characteristic distances (d_{050} , d_{050} , and d_{050g}) that are larger in magnitude compared to those used in conjunction with the point stress criterion (see Figure 30).

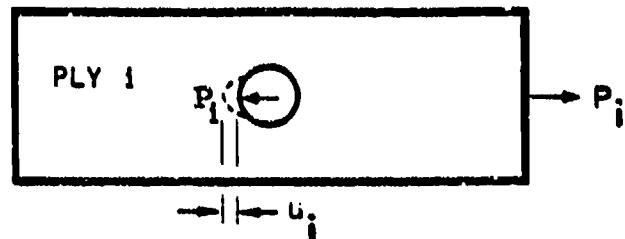
Of the three ply failure modes, only the net section mode causes the ply to become almost ineffective (total failure). The bearing mode of failure causes the ply to suffer a reduction in its effective modulus without losing its load-carrying capacity. The shear-out mode of failure causes a ply to become 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 31. The damaged ply can carry additional load until total ply failure is precipitated. The SASCJ computer code automatically stores the damage state in every ply in the bolted plates, and reassigns values 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 and 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_0) 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

$$K_2 = \alpha K_1$$

$$P_{\text{ULTIMATE}} = \beta P_{\text{INITIAL}}$$



$$q_i = P_i / h_i$$

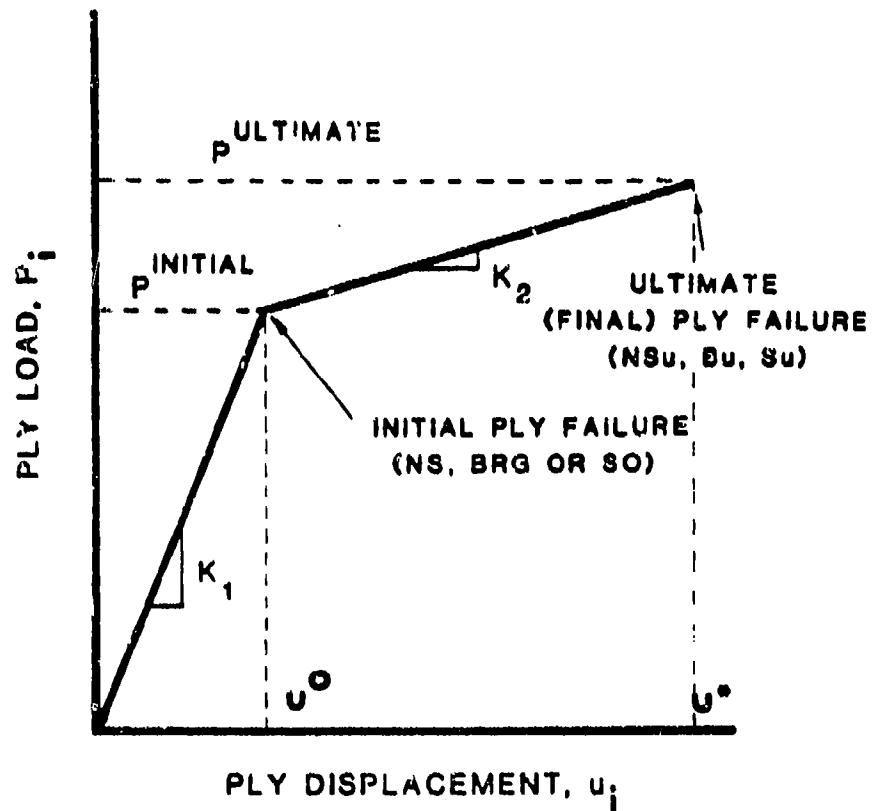


Figure 31. Bilinear Elastic Behavior of a Ply.

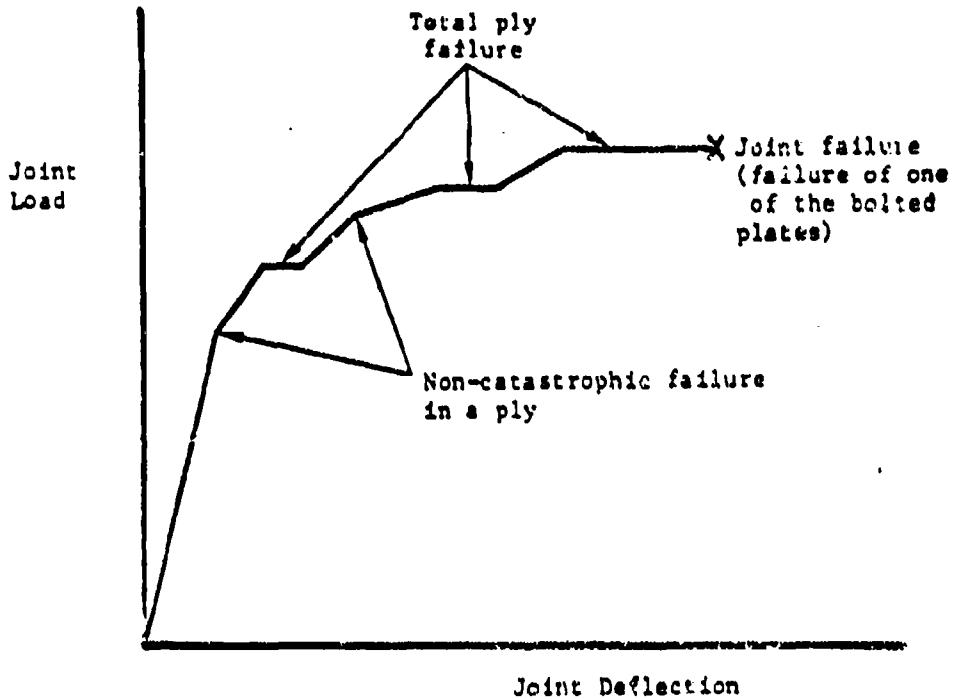


Figure 32. A Schematic Representation of the Overall Load Versus Deflection Response of the Joint.

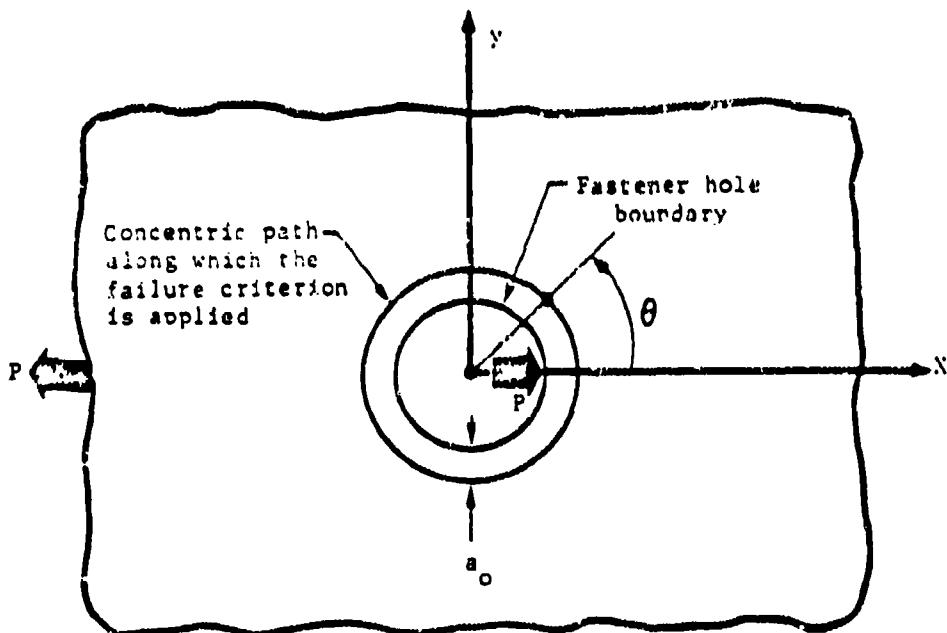


Figure 33. The Characteristic Distance (a_0) Defining the Region Where the Maximum Strain, Hoffman or Hill Criterion is Applied.

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

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

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_0 (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

$$a = [(\sigma_1^2 - \sigma_1 \sigma_2) / X_c X_t + \sigma_2^2 / Y_c Y_t + \sigma_6^2 / S^2] / P_i^2$$

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

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 $\theta=0^\circ$, a bearing mode of failure is suspected. If the failure location is near $\theta=90^\circ$, 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:

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

where h_a 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 quantities 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]_o. SASCJ automatically assumes a metallic plate to be divided into thirty identical layers. The number of layers in a laminate is controlled

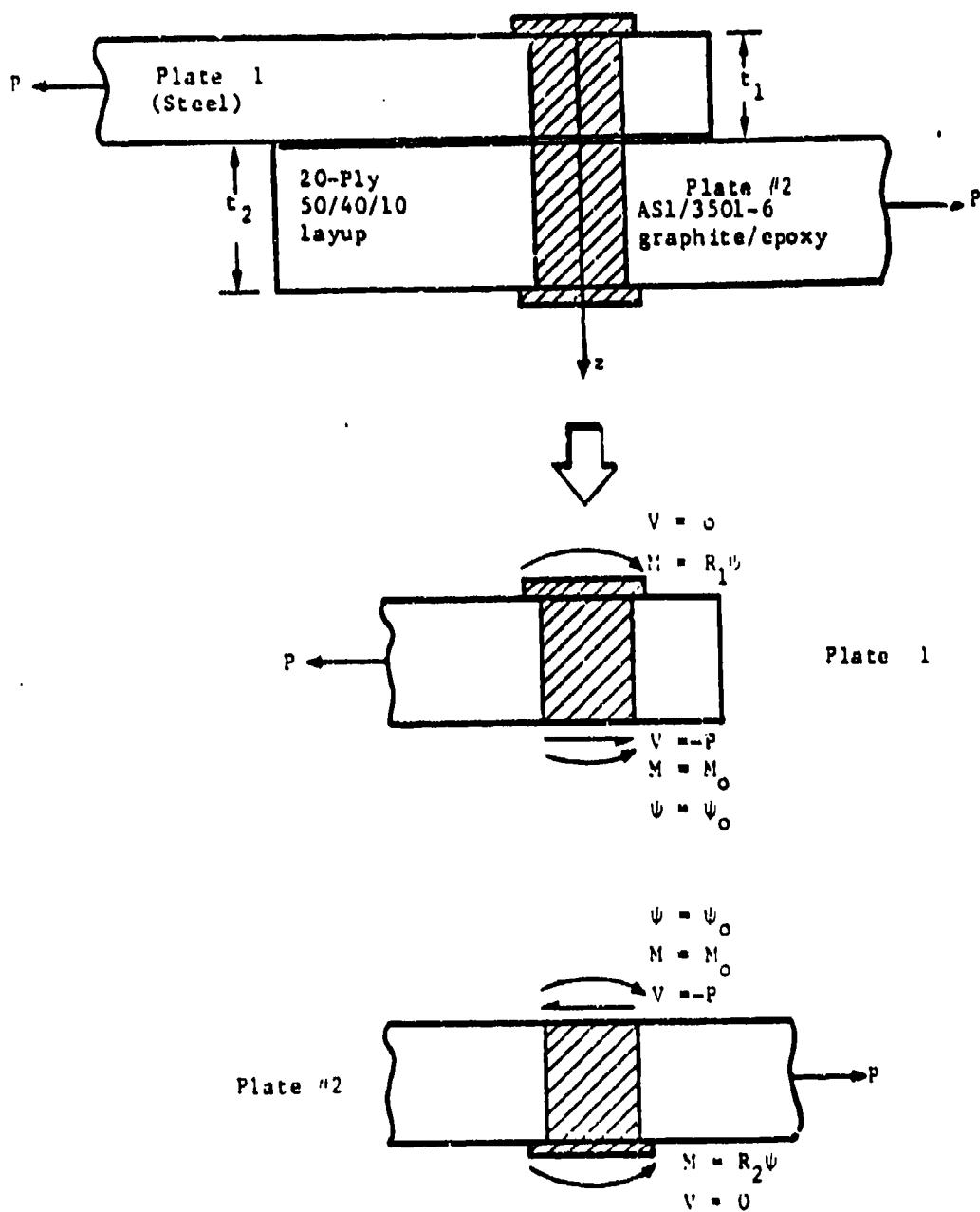


Figure 34. SASCJ Analysis of a Joint in Single Shear.

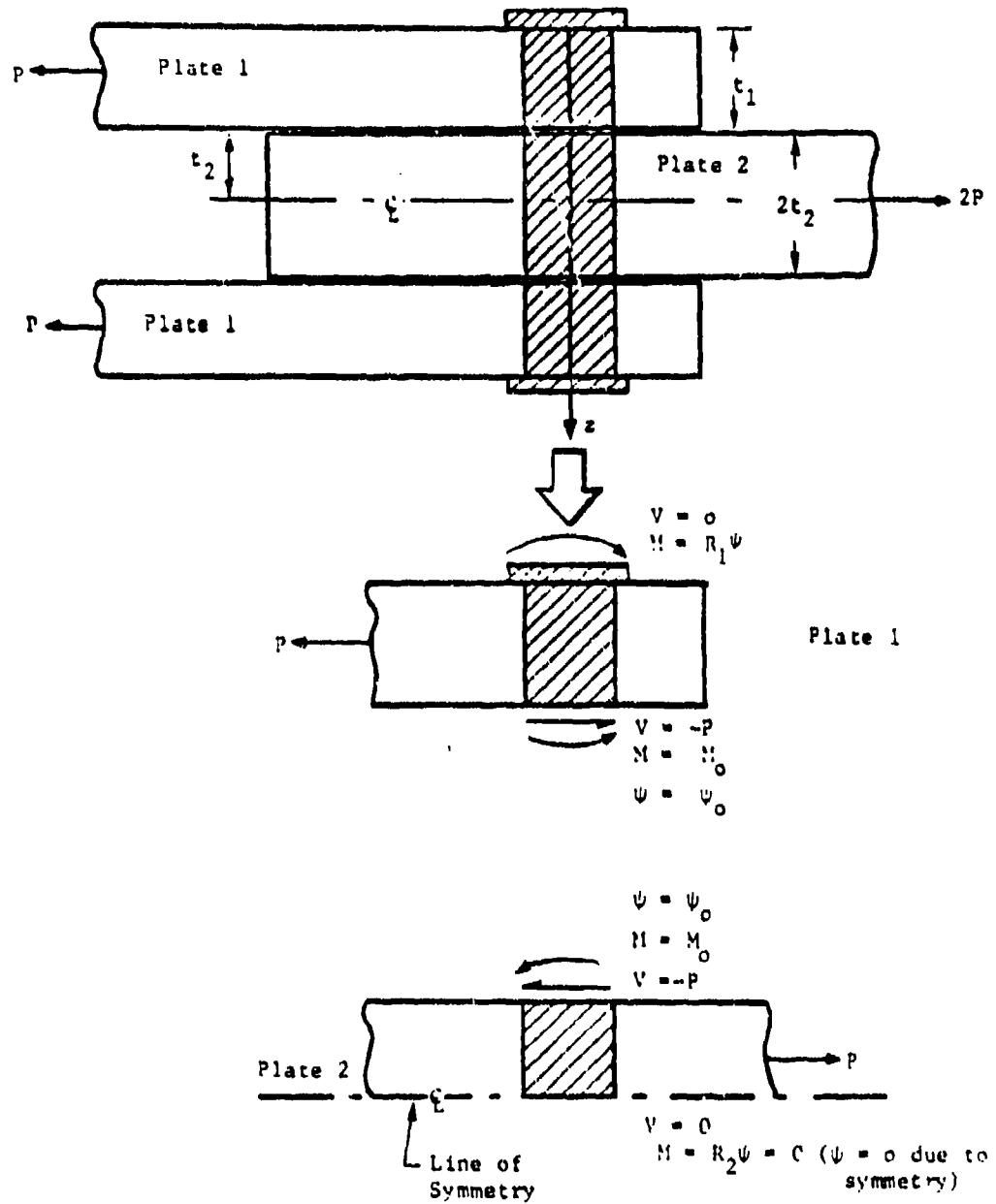


Figure 35. SASCJ Analysis of a Joint in Double Shear.

ପରେବେଳେ କିମ୍ବା କିମ୍ବା କିମ୍ବା

PROGRAM SCSJ PREDICTS FAILURE LOADS OF
STRUCTURALLY AND END-COMPOSITE LAMINATE,
STIFFENED OR DOUBLE LAP SCREW JOINTS.

PROGRAM ASSURES THAT INPUT PARAMETERS ARE
IN ENGLISH UNITS. LENGTHS ARE EXPRESSED IN
INCHES AND STRENGTHS ARE
EXPRESSED IN PSI.

Sammel-SACHEL-Losch.

```

? INPUT TYPE OF PLY FOR PLY NO 11
? INPUT TYPE OF PLY FOR PLY NO 12
? INPUT TYPE OF PLY FOR PLY NO 13
? INPUT TYPE OF PLY FOR PLY NO 14
? INPUT TYPE OF PLY FOR PLY NO 15
? INPUT TYPE OF PLY FOR PLY NO 16
? INPUT TYPE OF PLY FOR PLY NO 17
? INPUT TYPE OF PLY FOR PLY NO 18
? INPUT TYPE OF PLY FOR PLY NO 19
? INPUT TYPE OF PLY FOR PLY NO 20
? INPUT THE ENGINEERING PROPERTIES OF PLATE NO 1
? INPUT VOLUMES RADIUS AND POISSONS RATIO
? 36.646 0.2
? INPUT THE ENGINEERING PROPERTIES OF PLATE NO 2
? INPUT VOLUMES RADIUS, E1 AND E2
? 16.546 1.946
? INPUT THE SHEAR RADIUS AND POISSON'S RATIO FOR
? FASTENER
? 0.546 0.3
? INPUT MATERIAL DESCRIPTION FOR FASTENER
? steel
? INPUT VOLUMES RADIUS AND POISSON'S RATIO FOR
? THE FASTENER
? 0.546 0.3
? INPUT THE DIAMETER OF THE FASTENER
? 0.3165
? FASTENER TYPE
? Enter: 1 for monotonic, 2 for cyclic
? 1
? PLATE GEOMETRIES ARE SPECIFIED BY
? INPUTTING THE COORDINATES OF THE CORNER
? POINTS. NOTE: THE ORIGIN IS AT THE FASTENER
? CENTER; INPUT COORDINATES ACCORDINGLY
? U1      -0.975      0.975
? U2      -0.975      -0.975
? U3      0.975      -0.975
? U4      0.975      0.975
? APPLIED LOAD CONDITIONS:
? FOR PLATE NO 1 (TOP), NORMAL LOADS ARE APPLIED
? BETWEEN U1 AND U2
? FOR PLATE NO 2 (BOTTOM), NORMAL LOADS ARE APPLIED
? BETWEEN U3 AND U4
? FOR PLATE NUMBER 1 :
? ENTER X,Y COORDINATES OF U 1
? 3.0 -0.975
? ENTER X,Y COORDINATES OF U 2
? 3.0 0.975
? ENTER X,Y COORDINATES OF U 3
? -3.0 0.975
? ENTER X,Y COORDINATES OF U 4
? -3.0 -0.975
? FOR PLATE NUMBER 2 :
? ENTER X,Y COORDINATES OF U 1
? 3.0 -0.975
? ENTER X,Y COORDINATES OF U 2
? 3.0 0.975
? ENTER X,Y COORDINATES OF U 3
? -3.0 0.975
? ENTER X,Y COORDINATES OF U 4
? -3.0 -0.975
? SELECT FAILURE CRITERION
? Enter 1 for point stress criterion
? Enter 2 for average stress criterion
? 2
? AVERAGE STRESS CRITERION
? NO IS THE CHARACTERISTIC RESISTANCE LOAD WHICH
? STABILISATION IS AFFECTED AS COMPARED WITH UNPRESSED
? STABILISATION TO SPECIFIED FAILURE
? ENTER AN UNLESS COUNTER-CLOCKWISE TO THE THREE
? PLATE FAILURE MODES IN PLATE NO 1
? MODE 1: MAX STRESS
? MODE 2: BENDING
? MODE 3: SHEAR
? INPUT MODE, MODE, MODE
? 1 2 3

```

Figure 36. Sample SASCI Input
(Continued).

INPUT BETAL FOR NET SECTION IN PLATE
 BE1=2 FOR BLANKET, ULTIMATE
 BE1=3 FOR S-CABIN ULTIMATE
 1.02 1.5 1.12

TO AVOID LENGTHY RUN TIMES DUE TO
 STRESS FIELD RECOMPUTATION SPECIFY THE
 NUMBER OF ULTIMATE PLATE FAILURES AFTER
 WHICH JOINT FAILURE WILL BE PREDICTED
 ENTER NO OF ULTIMATE FAILURES
 ?
 20

ENTER NO VALUES CORRESPONDING TO THE THREE
 PLATE FAILURE MODES IN PLATE NO 2
 AMNT = NET SECTION
 ALDM = BEARING
 AG20 = SHEAR OUT

INPUT AMNT, ALDM, AG20
 ? 0.1 0.025 0.00

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

INPUT FST,FSC,FSO
 ? 250.000 300.000 200.000

FOR PLATE NO 2 ENTER FIRST ULTIMATE
 STRAIN VALUES EPSILON ULT IN COMPRESSION
 EPSILON ULT IN TENSION
 GAMMA ULT IN SHEAR

INPUTS: (NU/20)
 ? 6.014 6.011 6.012

SASCI ASSUMES A BI-LINEAR PLATE BEHAVIOR. THE
 INITIAL MODULUS, E1, IS COMPUTED BY THE CODE.
 THE REDUCED MODULUS, E2, FOR INITIAL FAILURE
 IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED
 BY THE FORMULA E2=E1*E2FACT.

FOR PLATE NUMBER 1 INPUT ALPHA VALUES FOR
 NET SECTION, SHEAROUT AND BEARING FAILURE
 ?
 0.01 0.1 0.1

INPUT SCALE FACTORS FOR P ULTIMATE
 CALCULATION SUCH THAT P=ULTIMATE*FACT(ULTIMATE)
 INPUT BETAL FOR NET SECTION ULTIMATE
 BE12 FOR BEARING ULTIMATE
 BE13 FOR SHEAROUT ULTIMATE
 ?
 1.02 1.5 1.12

SASCI APPROXES A BI-LINEAR PLATE BEHAVIOR. THE
 ENDING MODULUS, E1, IS COMPUTED BY THE CODE.
 THE REDUCED MODULUS, E2, FOR INITIAL FAILURE
 IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED
 BY THE FORMULA E2=AFACT.

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=ULTIMATE*FACT(ULTIMATE)

Figure 36. Sample SASCI 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
 ▲ SINGLE LAP SCAFFOLD JOINT WILL BE ANALYZED
 WITH A PARTIALLY LOADED HOLE
 BURST RATIO = 0.5000-00
 LOADED IN STATIC TENSION

PLATE NO	1
STEEL	
T	= 0.2500-00 INCHES
MATERIAL PROPERTIES	
E1	= 30000-00 PSI
E2	= 30000-00 PSI
G12	= 0.11500-00 PSI
N12	= 0.30000-00
N22	= 0.30000-00
PLATE NO	2
AS1/351-6 1105.00-00/0120/0015	
STEEL	
T	= 0.1200-00 INCHES
MATERIAL PROPERTIES	
E1	= 18500-00 PSI
E2	= 19000-07 PSI
G12	= 0.0500-00 PSI
N12	= 0.20000-00
N22	= 0.30000-01
FASTENER:	
STEEL	
DIAMETER	= 0.3130-00 INCHES
MATERIAL PROPERTIES	
E	= 30000-00 PSI
R	= 0.3000-00
FAILURE ANALYSIS	
AN AVERAGE STRESS CRITERION WILL BE USED	
PLATE NUMBER	1
LAMINATE STRENGTH	

NET SECTION ULTIMATE STEN = -0.2500-00 PSI
 NET SECTION ULTIMATE (COP) = -0.2605-00 PSI
 BEARING ULTIMATE
 SLIP-OUT ULTIMATE
 -0.2620-00 PSI

CHARACTERISTIC DISTANCES

ADOT	= 0.2400-00 INCHES
ACIR	= 0.4400-00 INCHES
ASOZ	= 0.5000-00 INCHES

PLATE NUMBER = 2

LAMINATE STRENGTH

NET SECTION ULTIMATE (STEN) = -0.1220-00 PSI
 NET SECTION ULTIMATE (COP) = -0.1520-00 PSI
 BEARING ULTIMATE
 SLIP-OUT ULTIMATE
 -0.1550-00 PSI

CHARACTERISTIC DISTANCES

ADOT	= 0.1510-00 INCHES
ACIR	= 0.2500-01 INCHES
ASOZ	= 0.3000-01 INCHES

PLATE NUMBER = 1

GEOMETRY OF PLATE NO 1

COORDINATES OF CORNER VERTICES

-3.000.	0.938	3.000.	0.938
-3.000.	-0.938	3.000.	-0.938

FASTENER HOLE DIAMETER = 0.3130-00 INCHES

E/D RATIO = 0.8600-01
 W/RATIO = 0.60-00-01

GEOMETRY OF PLATE NO 2

COORDINATES OF CORNER VERTICES

-3.000.	0.938	3.000.	0.938
-3.000.	-0.938	3.000.	-0.938

FASTENER HOLE DIAMETER = 0.3130-00 INCHES

E/D RATIO = 0.8600-01
 W/RATIO = 0.60-00-01

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

1.000. 0.938 3.000. 0.518
 3.000. 0.938 3.000. -0.518
 FASTENER HOLE DIAMETER = 0.3130+00 INCHES
 E.I. RATIO = 0.0563+01
 U.D RATIO = 0.0400+01

NO : NO ADDITIONAL DAMAGE AT CURRENT JOINT LOAD
 DL : DELAMINATION
 SO : SHEAR-OUT
 DR : Nudging
 NS : NEW SECTION
 SU : ULTIMATE FAILURE AFTER SO AND DR
 SUU : ULTIMATE FAILURE IN SO
 SUU : ULTIMATE FAILURE IN DR
 ULT : ULTIMATE FAILURE

FAILURE MODE ABBREVIATIONS:
 NO : NO ADDITIONAL DAMAGE AT CURRENT JOINT LOAD
 DL : DELAMINATION
 SO : SHEAR-OUT
 DR : Nudging
 NS : NEW SECTION
 SU : ULTIMATE FAILURE AFTER SO AND DR
 SUU : ULTIMATE FAILURE IN SO
 SUU : ULTIMATE FAILURE IN DR
 ULT : ULTIMATE FAILURE

JOINT NO	JOINT LOAD	MODE	PLT TYPE	MODE
1	0.8530+00	NO	NO	DECRE
2	0.8730+00	NO	NO	DECRE
3	0.8710+00	NO	NO	DECRE
4	0.8720+00	NO	NO	DECRE
5	0.8730+00	NO	NO	DECRE
6	0.8730+00	NO	NO	DECRE
7	0.8730+00	NO	NO	DECRE
8	0.8730+00	NO	NO	DECRE
9	0.8740+00	NO	NO	DECRE
10	0.8850+00	NO	NO	DECRE
11	0.8850+00	NO	NO	DECRE
12	0.8850+00	NO	NO	DECRE
13	0.8770+00	NO	NO	DECRE
14	0.8760+00	NO	NO	DECRE
15	0.8660+00	NO	NO	DECRE
16	0.8540+00	NO	NO	DECRE
17	0.8610+00	NO	NO	DECRE
18	0.8890+00	NO	NO	DECRE
19	0.8160+00	NO	NO	DECRE
20	0.8400+00	NO	NO	DECRE
21	0.8210+00	NO	NO	DECRE
22	0.8820+00	NO	NO	DECRE
23	0.8820+00	NO	NO	DECRE
24	0.8840+00	NO	NO	DECRE
25	0.8840+00	NO	NO	DECRE

THE PREDICTED JOINT STRENGTH
IS 15.0 KIPS @ U.D= 1.05

END

20 0.8530+00
 21 0.8730+00
 22 0.8710+00
 23 0.8720+00
 24 0.8730+00
 25 0.8730+00
 26 0.8730+00
 27 0.8730+00
 28 0.8730+00
 29 0.8740+00
 30 0.8850+00
 31 0.8850+00
 32 0.8850+00
 33 0.8850+00
 34 0.8850+00
 35 0.8770+00
 36 0.8760+00
 37 0.8660+00
 38 0.8540+00
 39 0.8610+00
 40 0.8890+00
 41 0.8160+00
 42 0.8400+00
 43 0.8210+00
 44 0.8820+00
 45 0.8820+00
 46 0.8840+00
 47 0.8840+00

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

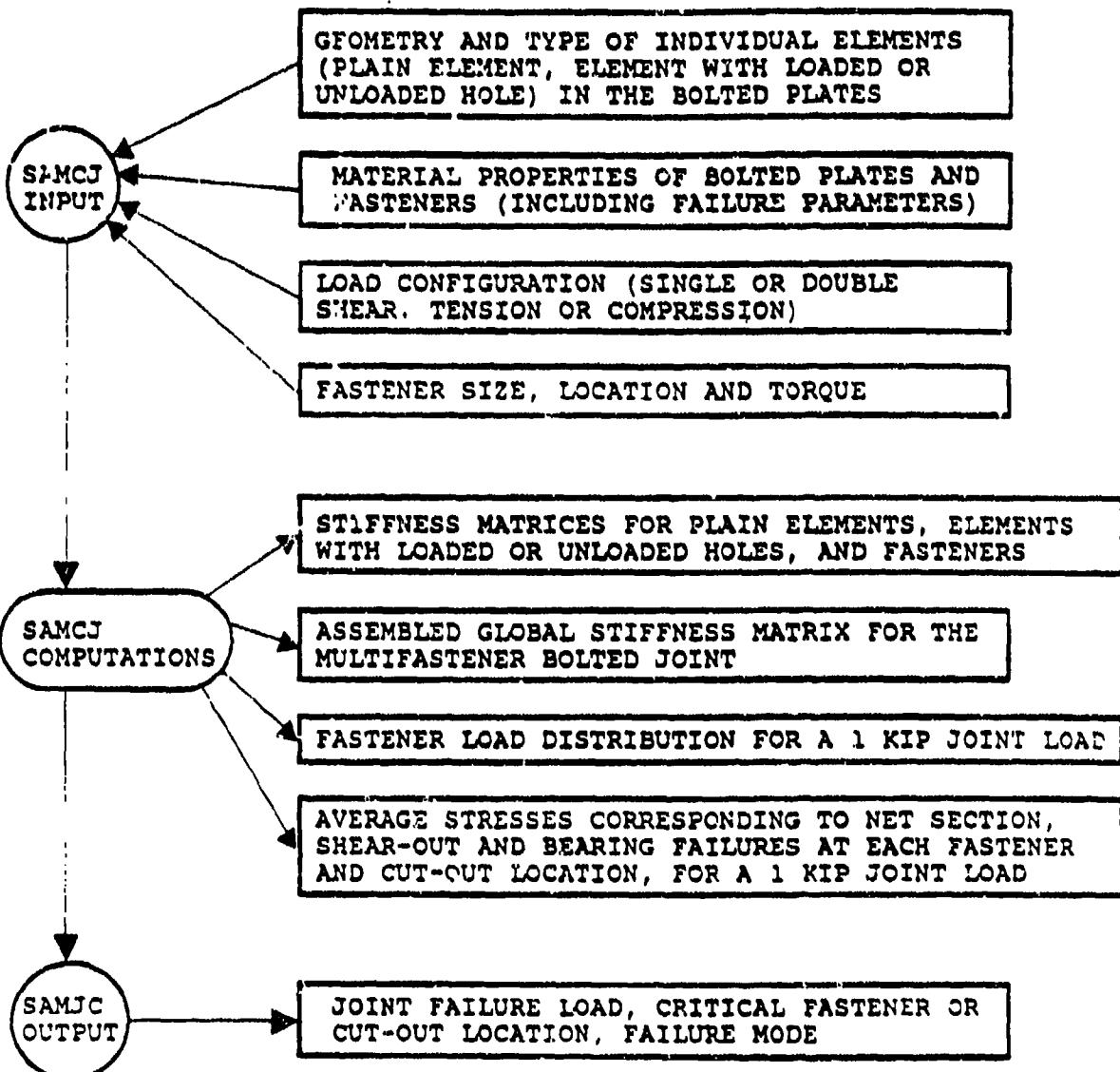


Figure 38. Flow Chart of SAMCJ Operations.

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 30). 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-cut 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

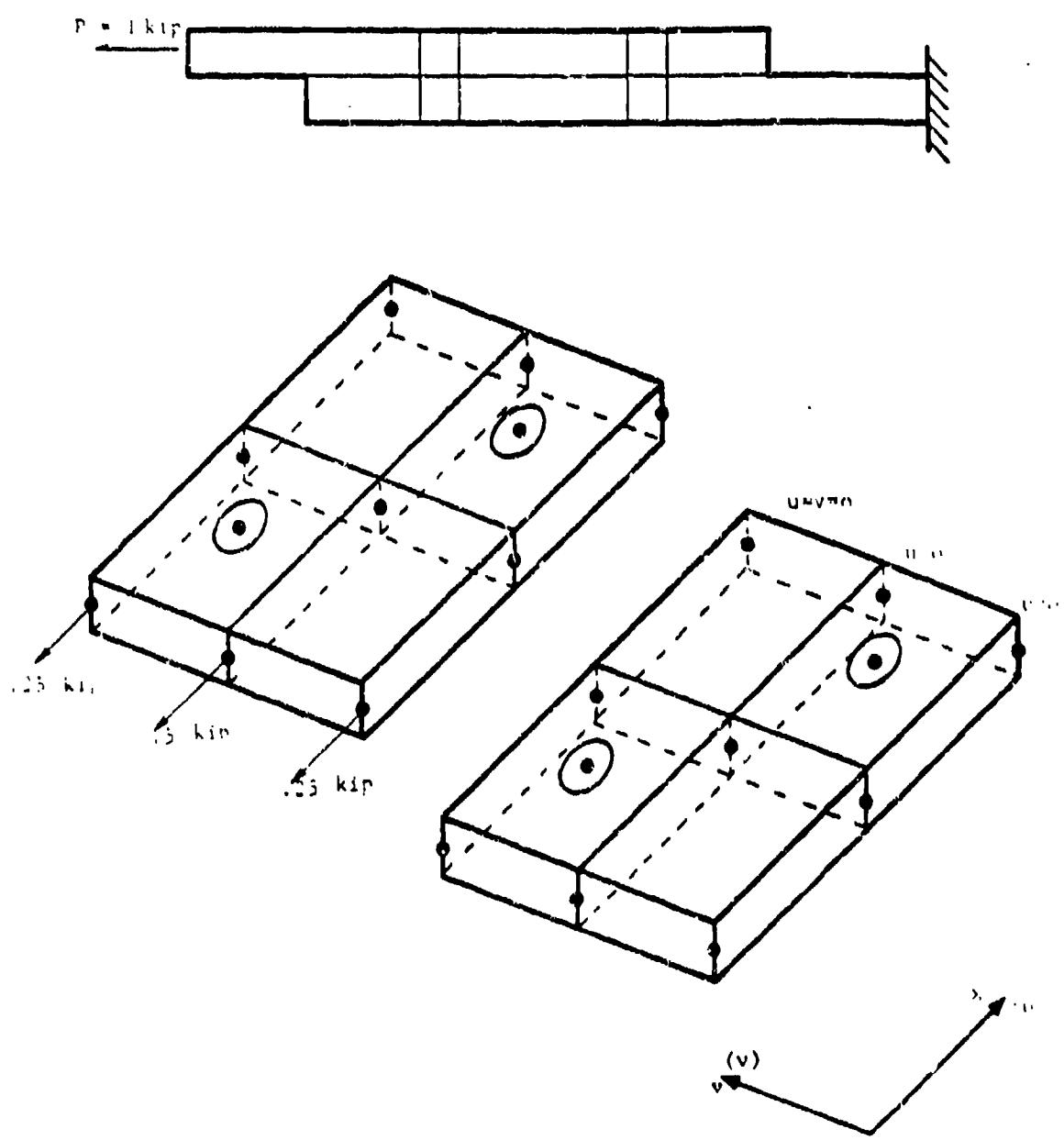


Figure 39. Application of Load and Displacement Boundary Conditions in the SAMCJ Code.

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 matrices (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.

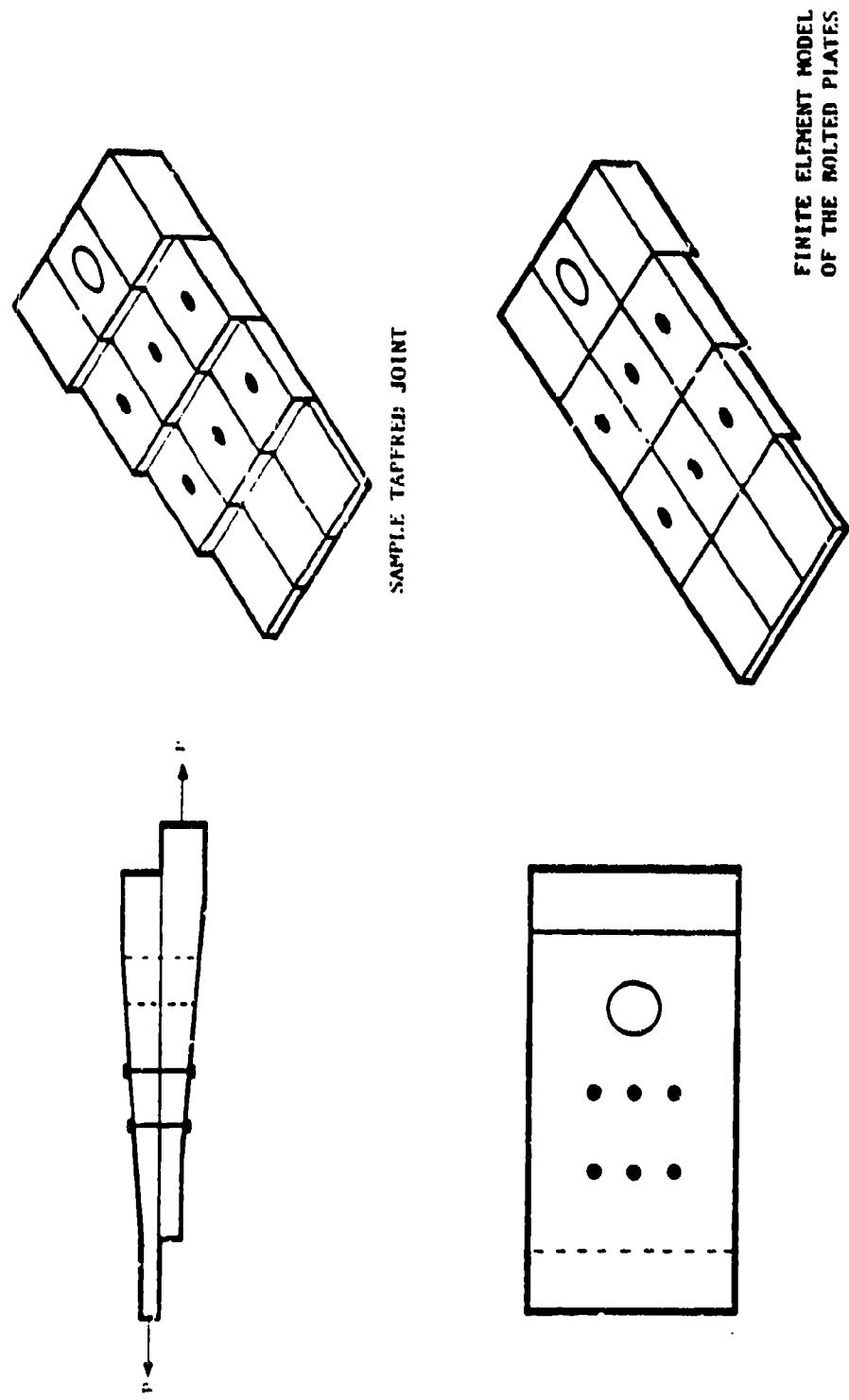
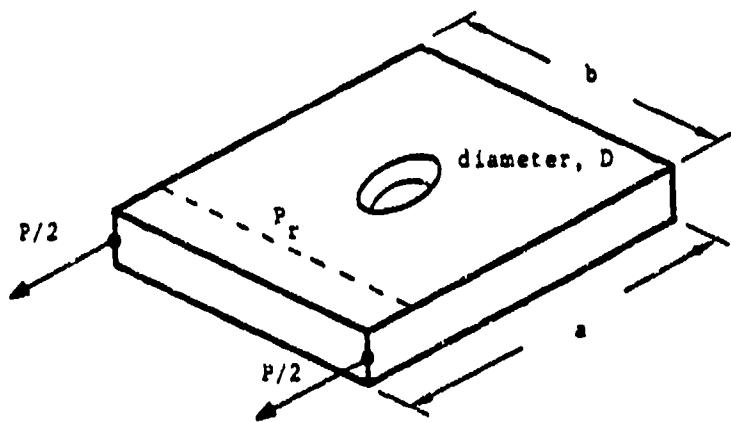


Figure 40. Finite Element Model of a Sample Tapered Bolted Joint.

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



$$t = 0.3125 \text{ inch}$$

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	1.26
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	-0.45
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

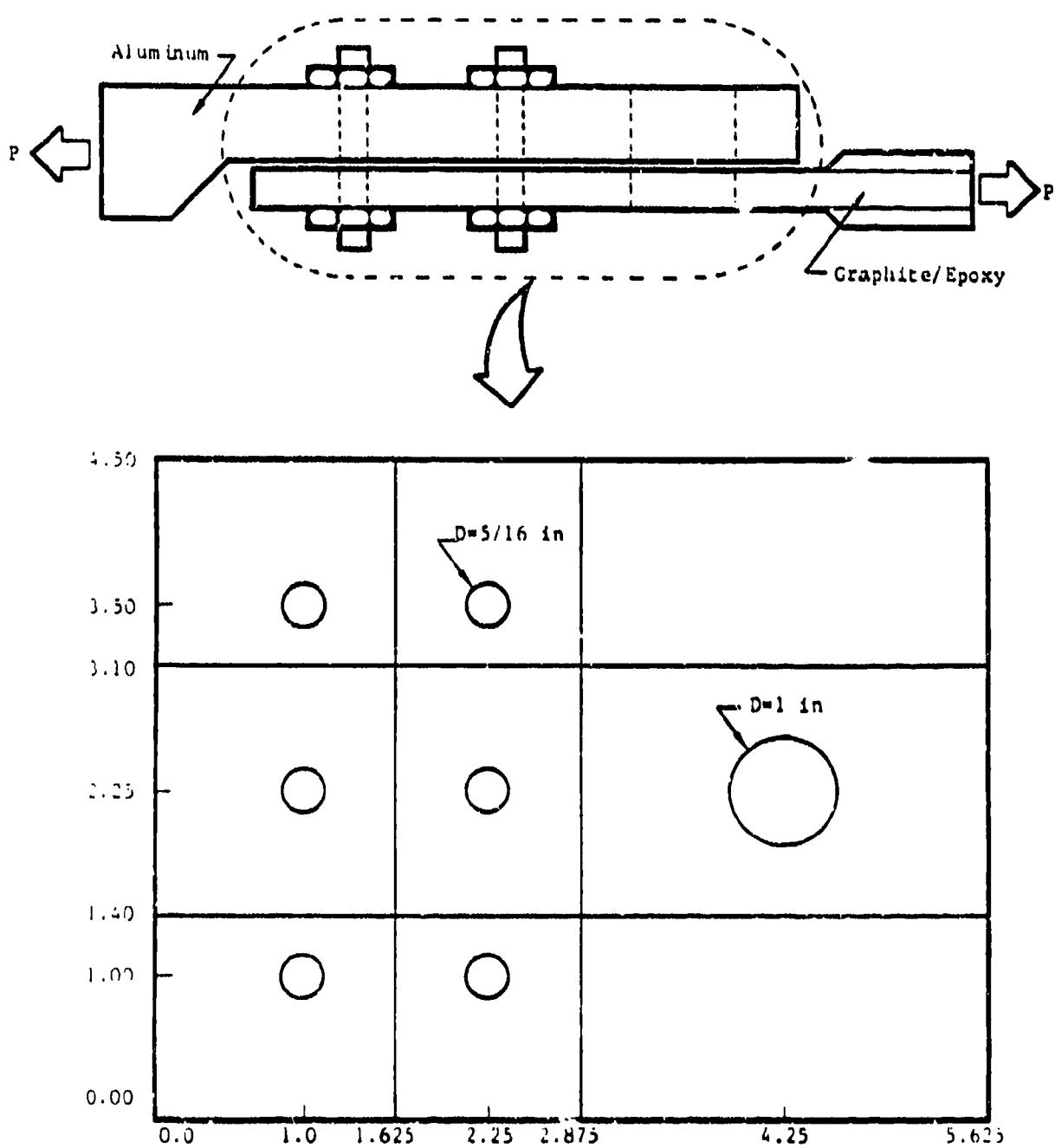


Figure 42. Nine Element Model of Each of the two Bolted Plates in the Sample Joint.

```

1.1. SELECT TEMPERATURE ASSUMED AS TEMPERATURE
1.2. SELECTED TEMPERATURE ASSUMED AS TEMPERATURE

PROGRAM SOURCE [ ]
```

PROGRAM SOURCE PREDICTS THE FAILURE LOAD, FAILURE LOCATION AND FAILURE MODE IN NATIONALLY-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)
    ?
```

```

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
    ?
```

```

2. INPUT MATERIAL DESCRIPTION OF THIS PLATE
    EX: ASA/350-6
    ?
```

OR ELSE

```

3. IS THE BOTTOM PLATE A COMPOSITE OR A METAL?
    ENTER C OR M IN THE FIRST FIELD
    ?
```

```

4. INPUT MATERIAL DESCRIPTION OF THIS PLATE
    EX: ASA/350-6
    ?
```

MATERIAL FOR COMPUTATIONAL PURPOSES A

METAL, FOR COMPUTATIONAL PURPOSES A

METALLIC PLATE IS MODELED AS A 36 PLY

LAMINATE OF 6 DEGREE PLIES WITH ISOTROPIC

MATERIAL PROPERTIES

```

5. INPUT THE ENGINEERING PROPERTIES OF THE TOP PLATE
    INPUT VARIOUS NUMBERS AND POSITIONS ANGLES
    ?
```

```

6. USE S.A. INPUT ENGINEERING PROPERTIES OF THE BOTTOM PLATE
    INPUT VARIOUS ANGLES, E1 AND E2
    ?
```

```

7. USE L.S.A. INPUT THE SHEAR MODULUS AND MAJOR POISSON'S RATIO
    ?
```

```

8. USE G.J. INPUT TOTAL NUMBER OF BISTING PLY
    ?
```

BALANCE LAYUP IN THE BOTTOM PLATE

```

9. INPUT ORIENTATION OF PLY TYPE NO 1
    ?
```

```

10. INPUT ORIENTATION OF PLY TYPE NO 2
    ?
```

```

11. INPUT ORIENTATION OF PLY TYPE NO 3
    ?
```

```

12. INPUT ORIENTATION OF PLY TYPE NO 4
    ?
```

THICKNESS LAYER(S) MAY BE APPROXIMATED

IN ASSIGNING DIFFERENT LAYUPS TO ELEMENTS

IN A COMPOSITE PLATE OR BY SPECIFYING DIFFERENT

THICKNESSES TO ELEMENTS IN A METALLIC PLATE

```

13. ENTER NUMBER OF DIFFERENT LAYERS IN THE
    BOTTOM PLATE
    ?
```

```

14. ENTER NUMBER OF PLIES IN LAYER NO 1
    ?
```

```

15. ENTER PLY THICKNESS FOR THIS LAYER
    ?
```

```

16. ENTER SEQUENCE OF PLY TYPES FROM TOP TO BOTTOM
    ?
```

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

8.3125
 FASTENER TYPE: F
 ENTER: 1
 ?
 22
 CHILD LAYOUTS:
 ENTER NUMBER OF CRIBS IN TOP PLATE
 ?
 ENTER 22 CRIB POINTS
 FORMAT: CRIB ID, X AND Y COORDINATES
 ?
 161 0.0 0.0
 ?
 162 0.0 1.49
 ?
 163 0.0 3.1
 ?
 164 0.0 4.5
 ?
 165 1.0 1.4
 ?
 166 1.0 2.25
 ?
 167 1.0 3.5
 ?
 168 1.425 0.0
 ?
 169 1.425 1.4
 ?
 170 1.425 2.1
 ?
 171 1.425 4.5
 ?
 172 2.25 1.0
 ?
 173 2.25 2.25
 ?
 174 2.25 3.5
 ?
 175 2.275 0.0
 ?
 176 2.275 3.4
 ?
 177 2.275 3.1
 ?
 178 2.275 4.5
 ?
 179 5.025 0.0
 ?
 180 5.025 1.4
 ?
 181 5.025 6.3
 ?
 INPUT MATERIAL DESCRIPTION
 ?
 Input Young's modulus and Poisson's ratio for
 the fastener
 ?
 Input the diameter of the fastener
 ?

Figure 43. SAP(4) Input for the Sample Problem in Figure 42 (Continued).

121 5.625 3.1
 122 5.625 4.5
 ENTER NUMBER OF POINTS IN BOTTOM PLATE
 82
 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 0.8
 ?
 209 1.625 1.4
 ?
 210 1.625 2.1
 ?
 211 1.625 4.5
 ?
 212 2.25 1.0
 ?
 213 2.25 2.25
 ?
 214 2.25 3.5
 ?
 215 2.875 0.8
 ?
 216 2.875 1.4
 ?
 217 2.875 2.1
 ?
 218 2.875 4.5
 ?
 219 5.625 0.0
 ?
 220 5.625 4.5

ELEMENT TYPES ARE DESIGNATED AS FOLLOWS:
 4 NODE PLAIN ELEMENT TYPE NO. 1
 5 NODE LOADED NCLE ELEMENT TYPE NO. 2
 4 NODE OPEN NCLE ELEMENT TYPE NO. 3
 (NOTE: ENTER NO. 3 FOR FOUR NODE ELEMENTS)
 ENTER NUMBER OF ELEMENTS IN TOP PLATE
 ?
 9
 FOR ELEMENT NO. 1 ENTER: ELEMENT 10, N1,N2,N3,N5, ELEMENT TYPE
 101 101 102 103 104 105 2
 ENTER ELEMENT THICKNESS
 ?
 0.5
 FOR ELEMENT NO. 2 ENTER: ELEMENT 10, N1,N2,N3,N5, ELEMENT TYPE
 102 102 103 104 105 106 2
 ENTER ELEMENT THICKNESS
 ?
 0.5
 FOR ELEMENT NO. 3 ENTER: ELEMENT 10, N1,N2,N3,N5, ELEMENT TYPE
 103 103 104 111 110 107 2
 ENTER ELEMENT THICKNESS
 ?
 0.5
 FOR ELEMENT NO. 4 ENTER: ELEMENT 10, N1,N2,N3,N4,N5, ELEMENT TYPE
 104 104 105 116 115 112 2
 ENTER ELEMENT THICKNESS
 ?
 0.5
 FOR ELEMENT NO. 5 ENTER: ELEMENT 10, N1,N2,N3,N4,N5, ELEMENT TYPE
 105 105 110 117 116 113 2
 ENTER ELEMENT THICKNESS
 ?
 0.5
 FOR ELEMENT NO. 6 ENTER: ELEMENT 10, N1,N2,N3,N4,N5, ELEMENT TYPE
 106 106 111 118 117 119 2
 ENTER ELEMENT THICKNESS
 ?
 0.5

ELEMENT DESCRIPTION:
 PLANE ELEMENTS ARE NUMBERED
 CLOCKWISE AS SHOWN

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

```

504 ELEMENT NO ? ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
167 115 116 120 119 @ 1
ENTER ELEMENT THICKNESS
? 0
FOR ELEMENT NO 8 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
108 118 117 121 120 @ 3
ENTER ELEMENT THICKNESS, X AND Y COORDINATES
UP OPEN HOLE AND HOLOC RADIUS
? 0.5 * 25 2.25 @ 5
FOR ELEMENT NO 9 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
109 117 118 122 121 @ 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
201 202 203 204 205 2
ENTER ELEMENT LAYER NO
? 1
FOR ELEMENT NO 11 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
202 203 210 209 206 2
ENTER ELEMENT LAYER 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 LAYER NO
? 2
FOR ELEMENT NO 13 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
204 204 209 216 215 212 2
ENTER ELEMENT LAYER 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 LAYER NO
? 1
FOR ELEMENT NO 15 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
and 8@ 811 812 813 814 2
ENTER ELEMENT LAYER NO
? 1

```

ENTER ELEMENT NO 16 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
 ? 207 215 216 221 219 @ 1
 ENTER ELEMENT LAYER NO
 ? 1
 FOR ELEMENT NO 17 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
 ? 208 216 217 221 220 @ 3
 ENTER ELEMENT LAYER NO
 ENTER X AND Y COORDINATES OF THE OPEN HOLE AND THE HOLE RADIUS
? 1.4 * 25 2.25 @ 5
 FOR ELEMENT NO 18 ENTER: ELEMENT ID, N1, N2, N3, N4, N5, ELEMENT TYPE
? 209 217 218 222 221 @ 1
 ENTER ELEMENT LAYER NO
? 1
 ENTER NUMBER OF FASTENERS IN JOINT
? 6

FASTENERS ARE MODELED BY EFFECTIVE
 FASTENER ELEMENTS WHICH PRODUCE THE
 ELASTIC LINE BETWEEN THE TOP AND
 BOTTOM PLATES
 ENTER NUMBER OF FASTENERS IN JOINT
? 6

EFFECTIVE FASTENER ELEMENTS ARE
 REFERRED AS SIGMA: N1 (TOP PLATE),
 N2 (BOTTOM PLATE)

N1@ N1 AND N2 CORRESPOND TO THE CENTRAL
 NODES IN LOADED HOLE ELEMENTS
 FOR ELEMENT ID, N1, N2
 ENTER ELEMENT NO 1
? 101 105 206
 ENTER ELEMENT NO 2
? 102 105 205
 ENTER ELEMENT NO 3
? 103 107 207
 ENTER ELEMENT NO 4
? 104 108 208
 ENTER ELEMENT NO 5
? 105 103 813
 ENTER ELEMENT NO 6
? 106 104 814

Figure 43. SAMCEF Input for the Sample Problem in Figure 42 (Cont Inued).

TO REDUCE RUN TIME, ELEMENTS MAY BE
 GROUPED INTO SETS WHICH WILL BE ASSIGNED
 IDENTICAL STIFFNESS MATRICES.
 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 :
 GROUPING OF EFFECTIVE FASTENER ELEMENTS:
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

ENTER 6 ELEMENT 105
 101 102 103 104 105 106

GROUPING OF LOADED HOLE ELEMENTS:
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

INPUT 1 ELEMENT 105

101 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 2

INPUT 1 ELEMENT 105
 107

GROUPING OF UNLOADED HOLE ELEMENTS:
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

INPUT 1 ELEMENT 105
 108

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 2

INPUT 1 ELEMENT 105
 109

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 3

INPUT 1 ELEMENT 105
 110

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 4

INPUT 1 ELEMENT 105
 111

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 5

INPUT 1 ELEMENT 105
 112

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 6

INPUT 1 ELEMENT 105
 113

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 7

INPUT 1 ELEMENT 105
 114

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 8

INPUT 1 ELEMENT 105
 115

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 9

INPUT 1 ELEMENT 105
 116

GROUPING OF PLAIN ELEMENTS
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

ENTER 0 ELEMENT 105

108 GROUPING OF LOADED HOLE ELEMENTS
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

?
 1

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

GROUPING OF LOADED HOLE ELEMENTS:
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

INPUT 1 ELEMENT 105
 201

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 2

INPUT 1 ELEMENT 105
 202

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 3

INPUT 1 ELEMENT 105
 203

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 4

INPUT 1 ELEMENT 105
 204

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 5

INPUT 1 ELEMENT 105
 205

ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 6

INPUT 1 ELEMENT 105
 206

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

GROUPING OF UNLOADED PLATE ELEMENTS
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

1 CENTER 1 ELEMENT 109
2 END 2 ELEMENTS

GROUPING OF PLAIN ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

2 CENTER 2 ELEMENT 108
2 END 2 ELEMENTS

INPUT DATA FOR FAILURE ANALYSIS:

ENTER METALLIC STRENGTH:
TENSILE STRENGTH
COMPRESSIVE STRENGTH
SHEAR STRENGTH

250.000 250.000 250.000

AN AVERAGE STRESS CRITERIA IS USED TO
PREDICT FAILURE. NO VALUES ARE REQUIRED AS
CHARACTERISTIC DISTANCES OVER WHICH STRESSES
ARE TO BE CALCULATED AND COMPARED TO UNNOTCHED
LAPWELD STRENGTHS TO PREDICT FAILURE.
ENTER NO VALUES FOR STRESS AVERAGES.
FOR EACH FAILURE MODE IN PLATE NO. 1
ENTER • READING

4050 • SCAFFOLD

6.5 6.5 6.5
ENTER FIBER MASTIFF STRAIN VALUES
IN PLATE NO. 2
1ST LAYER IN COMPRESSION
2ND LAYER IN TENSION
3RD LAYER IN SHEAR

0.0175 0.012 0.012

AN AVERAGE STRESS CRITERIA IS USED TO
PREDICT FAILURE. NO VALUES ARE REQUIRED AS
CHARACTERISTIC DISTANCES OVER WHICH STRESSES
ARE TO BE CALCULATED AND COMPARED TO UNNOTCHED
LAPWELD STRENGTHS TO PREDICT FAILURE.
ENTER NO VALUES FOR STRESS AVERAGES.
FOR EACH FAILURE MODE IN PLATE NO. 2
ENTER • READING

0.1 0.05 0.05

Figure 4J. SANCJ Input for the Sample Problem in Figure 4J (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 arbitrarily 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 SANCJ

A 67/8" LAP SHEAR PANEL WILL BE ANALYZED

LOADED IN STATIC TENSION

PLATE NO 1.

ALUMINUM

MATERIAL PROPERTIES

E1	-9.1699e-06	P11
E2	-9.1699e-07	P21
G12	-0.1671e-06	P31
M112=0	3.699e-09	
M21=0	3.699e-09	

PLATE NO 2.

ALUMINUM

MATERIAL PROPERTIES

E1	-9.1699e-06	P11
E2	-9.1699e-07	P21
G12	-0.1671e-06	P31
M112=0	3.699e-09	
M21=0	3.699e-09	

FASTENER DESCRIPTION

STEEL

DIA/MATERIAL = 0.12500 INCHES

MATERIAL PROPERTIES

E	-9.3038e-06	P11
F2	-9.3038e-07	P21

FAILURE SURFACE

AN AUTOMATIC STRESS CONVENTION WILL BE USED

PLATE NUMBER 1

STRUCTURAL ELEMENTS

THICKNESS	0.062500
CROSSSECTIONAL AREA	0.062500
SECTION LENGTH	0.125000

CHARACTERISTIC DISTANCES

PLATE NUMBER 2

FIBER STRAIN ULTIMATES

STRENGTH	0.1750-01
STRAIN	0.1628-01
STRESS	0.1269-01

CHARACTERISTIC DISTANCES

CHAR	0.1600e-06	INCHES
CHAR	0.2500e-06	INCHES
CHAR	0.2500e-06	INCHES

RESULT FOR STATIC LOAD MATRIX CALCULATIONS

FOR LATER FORCES
(P APPLIED = 10.000 LBS.)

ELEMENT	101	Fx	Fy
2010	-0.1500e-03	0.1738-12	
101	-0.1100e-03	-0.2448-02	
102	0.9950e-03	0.1568-02	
103	0.6730e-03	0.1268-02	
104	0.7570e-03	0.1268-02	
105	0.1533e-03	0.1653e-03	
ELEMENT	102	Fx	Fy
2010	-0.1500e-03	0.1738-03	
102	-0.1100e-03	-0.2448-03	
103	0.6730e-03	0.1268-03	
104	0.7570e-03	0.1268-03	
105	0.1533e-03	0.1653e-03	

CHARACTERISTIC DISTANCES

Copy and print right does not
permit fully legible reproduction

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

Element ID	201	rx	ry	rz
119	-3.100E-01	-1.552E-02		
137	3.165E-03	-1.552E-02		
Element 10	184	fx	fy	
CP10	163	-7.973E-02	-1.560E-02	
149	-1.918E-02	-1.460E-02		
116	0.245E-01	0.183E-02		
115	0.715E-01	0.72E-02		
112	0.158E-03	0.2E-02		
Element 12	185			
CP10	162	-7.760E-02	-5.532E-01	
113	-7.670E-02	0.397E-01		
117	-1.065E-02	0.546E-01		
116	-3.939E-01	-8.150E-01		
113	0.175E-03	0.334E-01		
Element 15	186			
CP10	160	fx	fy	
119	-8.955E-02	0.163E-02		
111	-7.758E-02	0.132E-02		
116	0.765E-01	-7.190E-02		
114	0.157E-03	-6.170E-01		
Element 16	187			
CP10	171	fx	fy	
115	-7.133E+01	-7.732E+01		
116	0.342E+01	0.343E+01		
120	0.371E+01	0.363E+01		
117	0.792E+02	-3.110E+02		
Element 18	188			
CP10	116	0.370E+01	-1.132E+02	
117	0.368E+01	0.113E+02		
121	-3.272E+01	0.222E+02		
120	-3.710E+01	-3.32E+01		
Element 19	189			
CP10	117	fx	fy	
118	0.333E+01	-7.750E+01		
122	-7.705E+01	0.739E+01		
121	0.373E+01	-3.570E+01		
Element 20	201			

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

ELEMENT ID	207		
CNT	FX	FY	FZ
215	-1280.03	0.2250.02	
216	-1620.03	0.3850.02	
217	-2110.03	0.5810.02	
221	0.2120.03	-1590.02	
224	0.1260.03	-1750.02	
213	0.1630.03	-1790.02	

ELEMENT ID	208		
CNT	FX	FY	FZ
216	-2120.03	0.3850.02	
217	-2110.03	0.5810.02	
221	0.2120.03	-1590.02	
224	0.2110.03	-1750.02	

ELEMENT ID	209		
CNT	FX	FY	FZ
217	-1580.03	0.8410.02	
218	-1280.03	0.3850.02	
222	0.1820.03	-1750.02	
221	0.1280.03	0.1630.02	

JOINT LOAD LEVELS CORRESPONDING TO NET SECTION (NS), SEGMENT 1,01, CANTILEVER BEARING (CB) FAILURES AT EVERY LOWER AND UPPER MOULDED JOINT ELEMENT ARE PREDICTED AS FOLLOWS:

ELEMENT	NS	SO	SI	SI
101	0.5750.06	0.8130.06	0.5600.06	
102	0.5450.06	0.7000.06	0.5800.06	
103	0.4170.06	0.6200.06	0.5220.06	
104	0.2620.07	0.1510.07	0.7380.07	
105	0.2110.07	0.1450.07	0.8370.07	
106	0.2030.07	0.1710.07	0.2370.07	
108	0.1480.08	0.6990.08	0.8270.08	
201	0.3460.06	0.1230.06	0.1230.06	
202	0.2200.06	0.1230.06	0.1230.06	
203	0.5120.06	0.1230.06	0.1230.06	
204	0.7640.06	0.6400.06	0.7770.06	
205	0.5220.06	0.4110.06	0.5770.06	
206	0.3860.06	0.5550.06	0.8130.06	
207	0.3730.06	0.1740.06	0.3720.07	

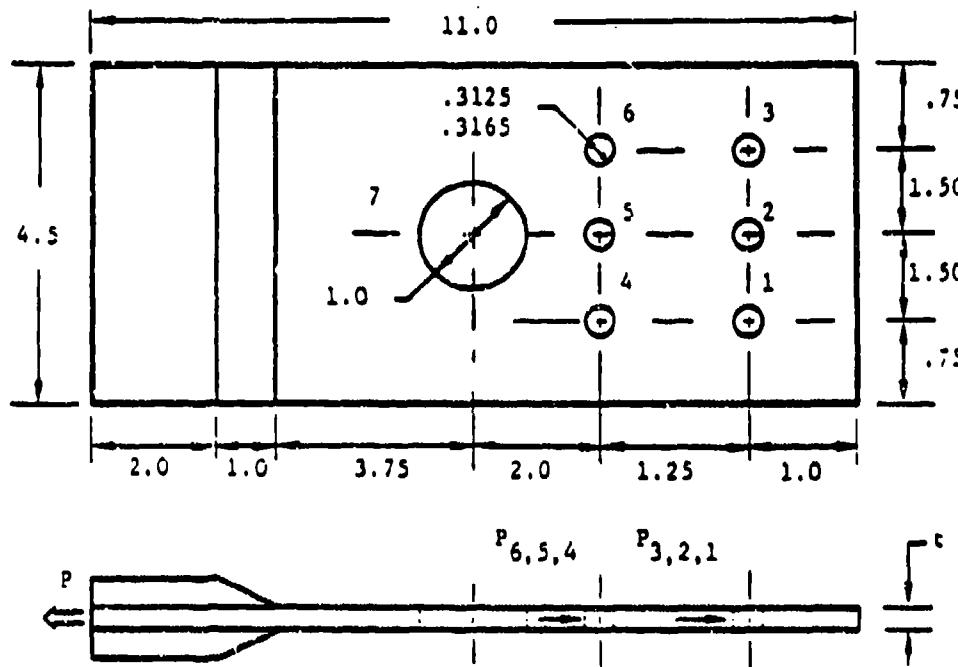
failure is predicted to occur in element number 208 at an applied joint load value of 0.3720.07 LBS

THE PREDICTED failure mode is net section

Figure 44. SANCI 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	0.168	0.165
$P_{failure}$ (kips)	37.3 (52.7)*	42.0
FAILURE LOCATION	7 (6)	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 Test Results

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 occurred in the graphite/epoxy skin tab in a combined mode (see Figure 17). 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 Design Verification of the Element Using SAMCJ

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 $[0_{28}/+45_{12}/90_7]_C$ 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 $[0_{28}/+45_{12}/90_6]_C$ layup, and the bottom region is assumed to be a $[0_{26}/+45_{10}/90_6]_C$ 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

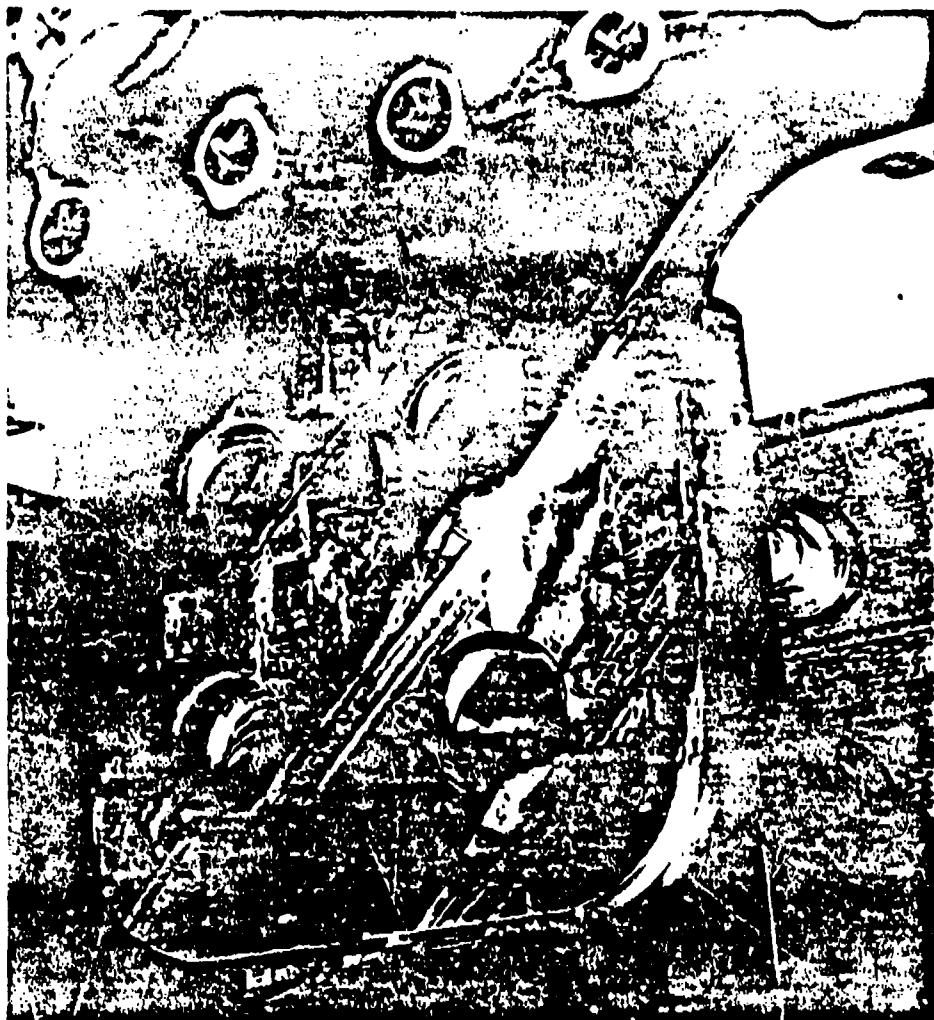


Figure 47. Photograph of the Tab Region of the Failed Element.

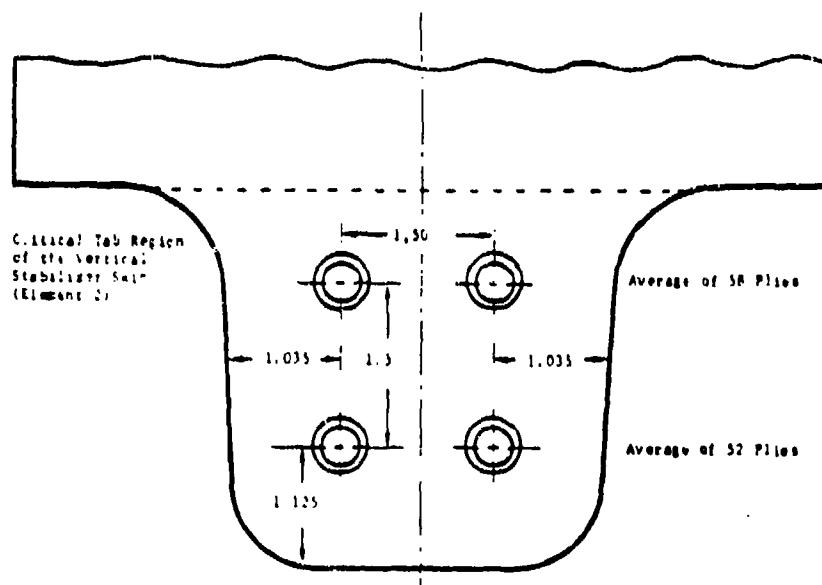
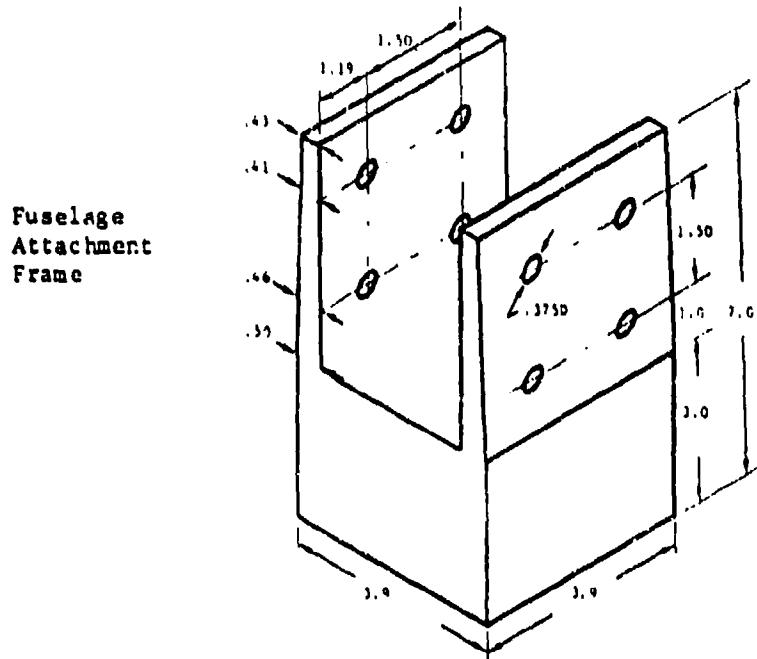


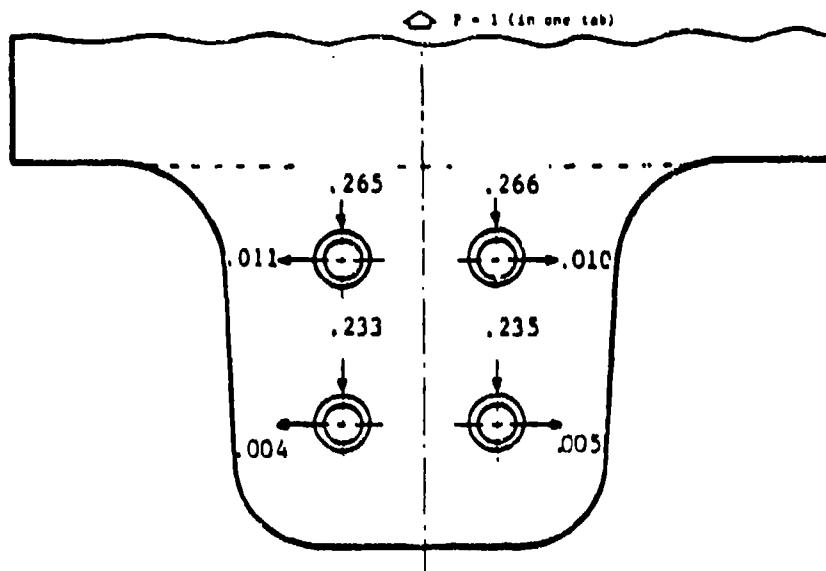
Figure 48. Dimensions of the Critical Skin Tab and the Fuselage Attachment Frame.

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

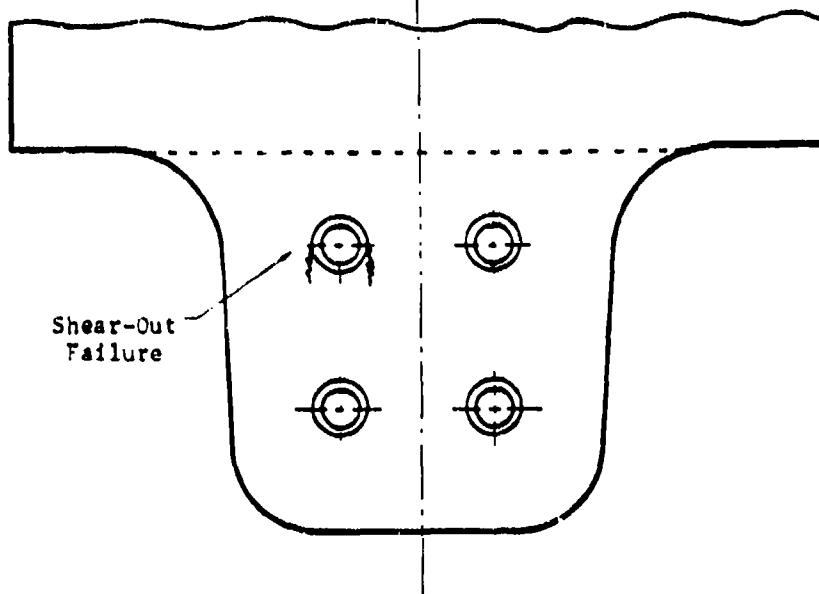
The modeled joint segment is half of the symmetric skin tab-to-fuselage attachment. The total joint failure load is, therefore, twice the predicted load. A single shear load transfer between the AS4/3501-6 graphite/epoxy skin tab and the steel attachment frame is analyzed. The graphite/epoxy tab and 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 ksi, 1.9 ksi and 0.85 ksi 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 (96.0 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 independently verify the 30% margin of safety in the static strength of the test element, due to the approximate analyses used in its preliminary design.

T_{ab} Load = P ; Element Load = $2P$



The numbers below are the values of the applied element load ($2P$), in kips to precipitate the three failure modes at each fastener location.

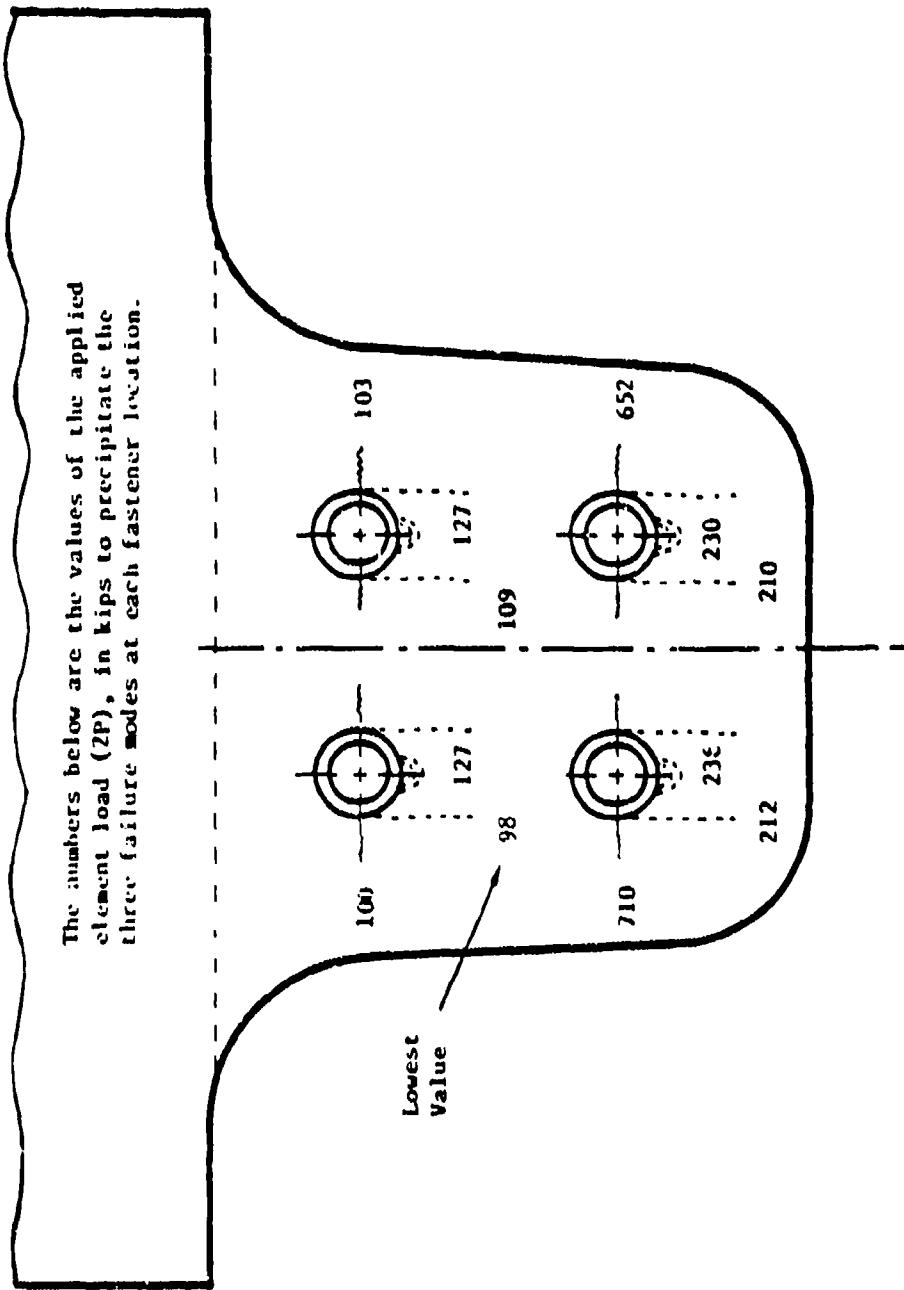


Figure 50. Analytically Predicted Element Load Levels to Precipitate Net Section, Bearing and Shear-Cut Modes of Tab Failure at Each Fastener Location.

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., "Conceptual Design of a High Load Transfer Mechanical Attachment 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. S., "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

```

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00000010
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00000020
CXX          XXXX 00000030
CXX          XXX 00000040
CXX          XXX 00000050
CXX          XXX 00000060
CXX          XXX 00000070
CXX          XXX 00000080
CXX          XXX 00000090
CXX          XXX 00000100
CXX          XXX 00000110
CXX          XXX 00000120
CXX          XXX 00000130
CXX          XXX 00000140
CXX          XXX 00000150
CXX          XXX 00000160
CXX          XXX 00000170
CXX          XXX 00000180
CXX          XXX 00000190
CXX          XXX 00000200
CXX          XXX 00000210
CXX          XXX 00000220
CXX          XXX 00000230
CXX          XXX 00000240
CXX          XXX 00000250
CXX          XXX 00000260
CXX          XXX 00000270
CXX          XXX 00000280
CXX          XXX 00000290
CXX          XXX 00000300
CXX          XXX 00000310
CXX          XXX 00000320
CXX          XXX 00000330
CXX          XXX 00000340
CXX          XXX 00000350
CXX          XXX 00000360
CXX          XXX 00000370
CXX          XXX 00000380
CXX          XXX 00000390
CXX          XXX 00000400
CXX          XXX 00000410
CXX          XXX 00000420
CXX          XXX 00000430
CXX          XXX 00000440
CXX          XXX 00000450
CXX          XXX 00000460
CXX          XXX 00000470
CXX          XXX 00000480
CXX          XXX 00000490
CXX          XXX 00000500
CXX          XXX 00000510
CXX          XXX 00000520
CXX          XXX 00000530
CXX          XXX 00000540
CXX          XXX 00000550

PROGRAM SASCJ
STRENGTH ANALYSIS OF SINGLE-FASTENER COMPOSITE JOINTS

SASCJ PREDICTS LOAD-DEFORMATION CURVES AND FAILURE LOADS OF
MECHANICALLY FASTENED, COMPOSITE LAMINATE, SINGLE LAP OR
SYMMETRICAL DOUBLE LAP SHEAR JOINTS. THE BASIS OF THE FATIGUE
ANALYSIS IS A NONLINEAR FINITE DIFFERENCE SOLUTION OF A BEAM
(FASTENER) ON AN ELASTIC FOUNDATION (COMPOSITE LAMINATE).
SELECTED FAILURE CRITERIA ARE USED TO PREDICT INDIVIDUAL PLY
FAILURES AND MODES (INCLUDING INTERLAMINAR SHEAR). THE LOAD
IS AUTOMATICALLY INCREMENTED TO FINAL FAILURE TO ACCOUNT FOR
THE NONLINEAR JOINT BEHAVIOR.

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(3,2),PNS(3,2),PBR(3,2),PSO(3,2),PALT(3,2)
DIMENSION DELBR(3,2),DELSO(3,2),ANGK(3,2)
DIMENSION ANG(3,2),IPLY(100,2),DONT(2),DOBR(2),DOSO(2)
DIMENSION FNT(3,2),FBR(3,2),FSO(3,2)
DIMENSION BARK(100),BARU(100),HFMC(3,2)
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(3,3,2),RCA(2),RCB(2),NRCOUT(2)
DIMENSION XC(2,3),YC(2,3),SALOH(2)
DIMENSION ADNT(2,2),AOBR(2,2),AOSO(2,2)
COMMON/AOV/AUNT,AOBR,AOSO
COMMON/COUNT/NPNM
COMMON/MOD/E1,E2,O1Z,V12,V21
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
COMMON/RCA/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'/

READ IN REQUIRED INPUT DATA

```

```

      WRITE(6,8,6)
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,N) BPR
      WRITE(6,911)
911 FORMAT(' ENTER: ',/
      *' 1 FOR STATIC TENSION ',/
      *' 2 FOR STATIC COMPRESSION')
      READ(S,N) 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,N) NSDLS
106 FORMAT(A1)
380 CONTINUE
      DO 300 K=1,NLIM
      IF(K.EQ.1) WRITE(6,511)
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: 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(6,721)
721 FORMAT(/, ' NOTE: FOR COMPUTATIONAL PURPOSES A ',/
      *' METALLIC PLATE IS MODELED AS A 30 PLY ',/
      *' LAMINATE OF 0 DEGREE PLYS 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
      00000560
      00000570
      00000580
      00000590
      00000600
      00000610
      00000620
      00000630
      00000640
      00000650
      00000660
      00000670
      00000680
      00000690
      00000700
      00000710
      00000720
      00000730
      00000740
      00000750
      00000760
      00000770
      00000780
      00000790
      00000800
      00000810
      00000820
      00000830
      00000840
      00000850
      00000860
      00000870
      00000880
      00000890
      00000900
      00000910
      00000920
      00000930
      00000940
      00000950
      00000960
      00000970
      00000980
      00000990
      00001000
      00001010
      00001020
      00001030
      00001040
      00001050
      00001060
      00001070
      00001080
      00001090
      00001100
      00001110
      00001120
      00001130
      00001140
      00001150

```

```

      GO TO 301                               00001160
15 CONTINUE                                00001170
      IF(NSDL.S.EQ.2.AND.K.EQ.1) WRITE(6,932) 00001180
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                           00001190
205 FORMAT(' INPUT NUMBER OF PLIES IN PLATE NO',IS,/, 00001200
      ' (N > OR = 2)')
      READ(5,X) NPLY(K)                      00001210
      NLIM2=2*NPLY(K)+1                     00001220
301 CONTINUE                                 00001230
      DO 302 K=1,NLIM                         00001240
      IF(CM(K).EQ.CMC) GO TO 25              00001250
      IF(NSDLS.EQ.2.AND.K.EQ.2) WRITE(6,933) 00001260
933 FORMAT(/, ' FOR THE DOUBLE LAP SHEAR CASE HAVING ',/
      ' A METALLIC PLATE NUMBER TWO, ENTER HALF THE ',/
      ' ACTUAL PLATE THICKNESS ')
      WRITE(6,35) K                           00001270
35 FORMAT(' INPUT THICKNESS OF PLATE NO',IS) 00001280
      READ(5,X) A1                          00001290
      H(K)=A1/NPLY(K)                      00001300
      GO TO 302
25 CONTINUE                                 00001310
      WRITE(6,260) K                           00001320
260 FORMAT(' INPUT PLY THICKNESS IN PLATE NO',IS) 00001330
      READ(5,X) H(K)
302 CONTINUE                                 00001340
      DO 303 K=1,NLIM                         00001350
      IF(CM(K).EQ.CMC) GO TO 45              00001360
      NUMPLY(K)=1                           00001370
      GO TO 303
45 CONTINUE                                 00001380
      WRITE(6,207) <
207 FORMAT(' INPUT NUMBER OF DISTINCT PLY ORIENTATIONS',/
      ' IN PLATE NO',IS)                   00001390
      READ(5,X) NUMPLY(K)
303 CONTINUE                                 00001400
      DO 209 K=1,NLIM                         00001410
      IF(CM(K).EQ.CMC) GO TO 55              00001420
      ANG(1,K)=0.
      GO TO 209
55 CONTINUE                                 00001430
      WRITE(6,487) K                           00001440
487 FORMAT(/, ' FOR PLATE NUMBER',IS,' 1',/)
      N=NUMPLY(K)
      DO 209 L=1,N
      WRITE(6,206) !
206 FORMAT(' INPUT ORIENTATION OF PLY TYPE NO',IS) 00001450
      READ(5,X) ANG(L,K)
209 CONTINUE                                 00001460
      DO 305 K=1,NLIM                         00001470
      IF(CM(K).EQ.CMC) GO TO 65              00001480
      NN=NPLY(K)
      DO 75 IJ=1,NN
75 IPLY(IJ,K)=1                           00001490
      GO TO 305
65 CONTINUE                                 00001500
      WRITE(6,210) K                           00001510

```

```

210 FORMAT(' INPUT TYPE OF PLY IN PLATE NO',I5,' FROM TOP',//,
*' TO BOTTOM',//,
*5X,'PLY TYPE',10X,'ORIENTATION')
N=NPLY(K)
DO 212 L=1,N
WRITE(6,213) L,ANG(L,K)
213 FORMAT(5X,I5,10X,F7.2,' DEGREES')
212 CONTINUE
WRITE(6,211)
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 35
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 YOUNG'S MODULI, E1 AND E2')
READ(5,*) E1(K),E2(K)
WRITE(6,218)
218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISONS 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(S,I),I=1,15)
251 FORMAT(12A4)
WRITE(6,252)
252 FORMAT(' INPUT YOUNG'S MODULUS AND POISONS 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
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
30001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320
00002330
00002340
00002350

```

```

R(2)=1.0L_J
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,X) N
R(N)=0.0D0
360 CONTINUE
C READ IN GEOMETRY AND BOUNDARY DATA
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,X) 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,X) NPT

```

```

IF(NOPT.EQ.1) NOPT4=2          00002960
IF(NOPT.EQ.2) NOPT4=4          00002970
GO TO 601                      00002980
567 CONTINUE                     00002990
      WRITE(6,220)                00003000
220 FORMAT(' SELECT FAILURE CRITERION ',//,
           X' ENTER 1 FOR HOFFMAN/TSAI-HILL CRITERION ',//,
           X' ENTER 2 FOR POINT STRESS CRITERION ',//,
           X' ENTER 3 FOR MAXIMUM STRAIN CRITERION ',//,
           X' ENTER 4 FOR AVERAGE STRESS CRITERION') 00003010
           READ(5,X) NOPT4          00003020
601 CONTINUE                     00003030
      IF(NOPT4.EQ.2.OR.NOPT4.EQ.4) GO TO 221 00003040
      DO 412 K=1,NLIM              00003050
      WRITE(6,222) K              00003060
222 FORMAT(' FOR PLATE NUMBER ',I5,' ENTER RADIUS OF ',//,
           X' CHARACTERISTIC CIRCLE AT WHICH STRESSES ARE',//,
           X' TO BE COMPUTED TO PREDICT FAILURE') 00003070
           READ(5,X) RCA(K)         00003080
           RCB(K)=RCA(K)          00003090
           NRCOUT(K)=50            00003100
           IF(NOPT4.EQ.3) GO TO 591 00003110
           WRITE(6,834)             00003120
834 FORMAT(' ENTER THE FAILURE INDEXES FOR THE ',//,
           X' HOFFMAN/TSAI-HILL CRITERIA 00003130
           X' NOTE: FOR USING TSAI-HILL SET EQUAL THE COMPRESSION ',//,
           X' AND TENSION ULTIMATES IN SIGMA X AND ',//, 00003140
           X' ENTER: SIGMA X ULTIMATE-COMPRESSION ',//, 00003150
           X' SIGMA X ULTIMATE-TENSION ',//, 00003160
           X' SIGMA Y ULTIMATE-COMPRESSION ',//, 00003170
           X' SIGMA Y ULTIMATE-TENSION ',//, 00003180
           X' SIGMA XY ULTIMATE ',//, 00003190
           READ(5,X)(HFMC(I,K),I=1,5) 00003200
           GO TO 412               00003210
591 CONTINUE                     00003220
      WRITE(6,393) K              00003230
393 FORMAT(' ENTER MAXIMUM STRAIN ALLOWABLE FOR',//,
           X' PLATE NUMBER ',I7,' (UNITS: IN/IN)') 00003240
           READ(5,X) SALDN(K)        00003250
412 CONTINUE                     00003260
      IF(NOPT4.EQ.3) GO TO 391 00003270
      IF(NOPT4.EQ.1) GO TO 391 00003280
      GO TO 262                  00003290
221 CONTINUE                     00003300
      IF(NOPT4.EQ.2) WRITE(6,555) 00003310
      IF(NOPT4.EQ.4) WRITE(6,556) 00003320
555 FORMAT(' POINT STRESS CRITERION ',//, 00003330
556 FORMAT(' AVERAGE STRESS CRITERION ',//, 00003340
           X' AO IS THE CHARACTERISTIC DISTANCE OVER WHICH',//, 00003350
           X' STRESSES ARE AVERAGED AND COMPARED WITH UNNOTCHED',//, 00003360
           X' STRENGTHS TO PREDICT FAILURE') 00003370
           DO 226 K=1,NLIM          00003380
           IF(BPR.NE.0.0.AND.BPR.NE.1.0) GO TO 531 00003390
           WRITE(6,225) K            00003400
225 FORMAT(' INPUT AO FOR EACH OF THE THREE PLY FAILURE',//,
           X' MODES OF PLATE NO',I5,/,
           X'           AONT = NET SECTION 00003410
           X'           AOBR = BEARING   00003420
           X'           AOSO = SHEAR OUT 00003430
           N=NUMPLY(K)               00003440
           ',/,'                         00003450
           ',/,'                         00003460
           ',/,'                         00003470
           ',/,'                         00003480
           ',/,'                         00003490
           ',/,'                         00003500
           ',/,'                         00003510
           ',/,'                         00003520
           ',/,'                         00003530
           ',/,'                         00003540
           ',/,'                         00003550

```

```

        WRITE(6,24)
227 FORMAT(' INPUT AONT,AOBR,AND AOSO')
      READ(5,X) AONT(K),AOBR(K),AOSO(K)
      GO TO 226
531 CONTINUE
      WRITE(6,532) K
532 FORMAT(' ENTER AO VALUES CORRESPONDING TO THE THREE',/
     X' PLY FAILURE MODES IN PLATE NO ',I2,',
     X'          AONT = NET SECTION      ',/
     X'          AOBR = BEARING        ',/
     X'          AOSO = SHEAR OUT     ')
      WRITE(6,533)
533 FORMAT(' INPUT AONT, AOBR, AOSO ')
      READ(5,X) 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 ',/
     X' STRESS FIELD RECOMPUTATION SPECIFY THE ',/
     X' NUMBER OF ULTIMATE PLY FAILURES AFTER ',/
     X' WHICH JOINT FAILURE WILL BE PREDICTED ',/
     X' ENTER: NO OF ULTIMATE FAILURES ')
      IF(K.EQ.1) READ(5,X) 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 ',/
     X' REQUIRED TO PREDICT THE THREE FAILURE MODES ',/
     X' FNST=UNNOTCHED STRENGTH IN TENSION ',/
     X' FNSC=UNNOTCHED STRENGTH IN COMPRESSION',/
     X' FSO=UNNOTCHED STRENGTH IN SHEAR-OUT',/
     X' INPUT FNST,FNSC,FSO ')
      READ(5,X) AF1,AF2,AF4
      GO TO 673
672 WRITE(6,674) K
574 FORMAT(' FOR PLATE NO ',I5,' ENTER FIBER ULTIMATE',/
     X' STRAIN VALUES           ')
      EPSILON ULT IN COMPRESSION
      EPSILON ULT IN TENSION
      GAMMA ULT IN SHEAR
      (UNITS: IN/IN')
      READ(5,X) 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,X)=AF2
      PSTC(3,IL,K)=AF3
      PSTC(4,IL,K)=AF4
666 CONTINUE

```

```

229 CONTINUE          00004160
671 CONTINUE          00004170
    IF(NOP14.NE.4) GO TO 261
C
C     NUMBER OF DIVISIONS FOR STRESS AVERAGING      00004180
C     IS SET EQUAL TO 50                               00004190
C
C     NAVD=50                                         00004200
261 CONTINUE          00004210
    IF(BPR.EQ.1.0) GO TO 262                         00004220
    DO 319 K=1,NLIM                                  00004230
    N=NUMPY(K)
    WRITE(6,320) K                                    00004240
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=ALPHAK1.                   ',
         ' FOR PLATE NUMBER ',IS,' INPUT ALPHA VALUES FOR ','',
         ' NET SECTION, SHEAROUT AND BEARING FAILURE   ')
    READ(5,*) AF1,AF2,AF3                            00004250
    DO 321 I=1,N
    DELNS(I,K)=AF1                                  00004260
    DELBR(I,K)=AF2                                  00004270
    DELSO(I,K)=AF3                                  00004280
321 CONTINUE          00004290
    WRITE(6,389)
389 FORMAT(' INPUT SCALE FACTORS FOR P ULTIMATE ','',
         ' CALCULATION SUCH THAT P(ULT)=BETA*P(INITIAL)',/,
         ' INPUT DETAL 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          00004300
391 CONTINUE          00004310
    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 ',IS,'',
         ' (UNITS: IN/IN)                                ')
    READ(5,*) GAMDL(K)
312 CONTINUE          00004320
262 CONTINUE          00004330
C
C     CASE HEADING
C
    WRITE(6,143)          00004340
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 SPEAR 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',//)
  X2X,'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)
613 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(J,J),J=1,15)
607 FORMAT(2X,15A4,/)
  WRITE(6,608) FASD
608 FORMAT(2X,' DIAMETER = ',D9.3,' INCHES',//)
  WRITE(6,609) FASF,FASF
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)
522 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,T 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)

```

```

538 FORMAT(2A,'AN AVERAGE STRESS CRITERION WILL BE USED',//)
      DO 631 I=1,NLIM
      WRITE(6,632) I
632 FORMAT(2X,'PLATE NUMBER',I5,//)
      NP=NUNPLY(I)
      WRITE(6,713)
713 FORMAT(//,2X,'LAMINATE STRENGTH',//)
776 WRITE(6,677) (PSTC(LL,I,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 361
      WRITE(6,644)
644 FORMAT(2X,'CHARACTERISTIC DISTANCES',//)
      WRITE(6,645) AONT(I),DOBR(I),DOSO(I)
645 FORMAT(2X,' AONT = ',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),ADJR(1,I),AOSO(1,I)
554 FORMAT(2X,' AONT = ',D9.3,' INCHES',//,
      *' ADJR = ',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.1) GO TO 23
WRITE(6,871) K
871 FORMAT('5X,' GEOMETRY OF PLATE NO ',IS,I-1',/)
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(6,875) 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=AXX2
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,A3T,NCOL,LTNCH,BPR,IL)
CALL CIRC(H,A3T,JK,K,LTNCH,BPR,IL)
IF(NOPT4.EQ.1,OR.,NOPT4.EQ.3) CALL RCOUT(K)
IF(NOPT4.EQ.2) CALL PSTRUSS(K,LTNCH,BPR,IL)
IF(NOPT4.EQ.4) CALL AVSTRUSS(K,LTNCH,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(ANOK,H,K,NOPT1,LM1)
21 CALL FCRIT(SALON,H,WTH,A3T,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      CALCULATION OF FASTENER STIFFNESSES.

```

```

C
FAS0=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*FASGMFASA
FASBS=FASE*FASI
00006560
00006570
00006580
00006590
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830
00006840
00006850
00006860
00006870
00006880
00006890
00006900
00006910
00006920
00006930
00006940
00006950
00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130
00007140
00007150

C
CCC
INITIALIZATION
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830
00006840
00006850
00006860
00006870
00006880
00006890
00006900
00006910
00006920
00006930
00006940
00006950
00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130
00007140
00007150

C
INCREMENTAL LOADS TO PLY FAILURE, PLY FAILURE
MODES, AND FRACTIONAL STIFFNESS LOSSES ARE
CALCULATED FOR EACH PLY FROM TOP TO BOTTOM
UNTIL FINAL JOINT FAILURE
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830
00006840
00006850
00006860
00006870
00006880
00006890
00006900
00006910
00006920
00006930
00006940
00006950
00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130
00007140
00007150

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(OAMD,L,N,HTH,PFAIL,ANGLE,NODE,
NROUT,NOPT4,NULTF,JNT,ITT,NTFL)
CALL PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT,
NWP,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
CCC
SUBROUTINE STRTH(H,ES1,ES2,ESS,AF1,AF2,AF4,K)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION AINV(3,3),AVNC(3),H(2),NV(3)
DIMENSION NPLY(2),NUMPL(2),ANO(3,2),IPLY(100,2)
DIMENSION WK(25),PSMX(3),SI(2),ES2(2),ESS(2)
DIMENSION E1(2),E2(2),O12(2),V12(2),V21(2)
COMMON/LYP/NPLY,NUMPL,ANO,IPLY
COMMON/MCD/E1,E2,O12,V12,V21
COMMON/AM1/A
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830
00006840
00006850
00006860
00006870
00006880
00006890
00006900
00006910
00006920
00006930
00006940
00006950
00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130
00007140
00007150

```

```

C   COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM      00007160
CCC FIBER STRAINS FOR EACH FAILURE MODE                 00007170
C   CALL AMATRIX(H,K)                                     00007180
H=3                                                       00007190
IDOT=4                                                    00007200
IA=3                                                    00007210
CALL LINV2F(A,H,IA,AINV,IDOT,HK,IER)                  00007220
DO 100 KK=1,3                                         00007230
DO 10 II=1,3                                         00007240
NV(II)=0                                              00007250
10 AVN(II)=0.000                                     00007260
IF(KK.EQ.1) NV(1)=1                                  00007270
IF(KK.EQ.2) NV(1)=-1                                 00007280
IF(KK.EQ.3) NV(3)=1                                 00007290
DO 13 II=1,3                                         00007300
DO 15 JJ=1,3                                         00007310
DO 15 JJ=1,3                                         00007320
AVN(II)=AVN(II)+AINV(II,JJ)*NV(JJ)                  00007330
15 CONTINUE                                           00007340
NP=NIMPLY(K)                                         00007350
SMX=0.000                                             00007360
RAD=DARCOS(-1.000)/180.000                           00007370
DO 25 II=1,NP                                         00007380
TH=ANG(II,K)*RAD                                     00007390
E11=DCOS(TH)*NV2*AVN(1)+AVN(2)*DSIN(TH)*NV2+        00007400
*DCOS(TH)*DSIN(TH)*AVN(3)                           00007410
IF(KK.NE.1) DO TO 65                                 00007420
EPRT=E11/ES2(K)                                       00007430
65 IF(KK.NE.2) DO TO 75                               00007440
DO TO 50                                              00007450
75 EPRT=E11/ES2(K)                                   00007460
50 CONTINUE                                           00007470
IF(DABS(SMX).LT.DABS(EPRT)) SMX=EPRT               00007480
25 CONTINUE                                           00007490
IF(DABS(SMX).GT.1.0D-10) DO TO 555                 00007500
PSMX(KK)=ESS(K)*O12(K)                             00007510
DO TO 100                                             00007520
555 CONTINUE                                           00007530
PSMX(KK)=DABS(1.0D0/SMX)                            00007540
100 CONTINUE                                           00007550
AF1=PSMX(1)                                           00007560
AF2=PSMX(2)                                           00007570
AF4=PSMX(3)                                           00007580
RETURN                                               00007590
END                                                 00007600
C
CCC SUBROUTINE POLY(J,K,XC,YC,H,AST,NCOL,LTHCM,BPR,IL) 00007610
IMPLICIT REALN8(A-H,O-Z)                           00007620
DIMENSION XC(2,5),YC(2,5),A1(400),A2(400),XB(400) 00007630
DIMENSION YB(400),T(400),A1A(4),A2A(4)            00007640
COMMON/CMT1/XB,YB,A1,A2,T                          00007650
CCC ARRAY COLLOCATION POINTS AROUND EXTERIOR BOUNDARY 00007660
CCC AND APPLY STRESS BOUNDARY CONDITIONS             00007670
CCC                                         00007680
                                         00007690
                                         00007700
                                         00007710
                                         00007720
                                         00007730
                                         00007740
                                         00007750

```

```

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(-1.000)
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=(YC(Y,I+1)-YC(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.4) GO TO 23
YB(J)=YC(K,I)
XB(J)=XC(K,I)+DX*(I+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 24
ADT=DAT*II
XB(J)=YC(K,3)*DCOS((II*2.)*ADT)
YB(J)=YC(K,3)*DSIN((II*2.)*ADT)
TH=((PI/2.)*ADT)*180.-PI
GO TO 24
26 CONTINUE
YB(J)=YC(K,1)+DY*(I+5)
IF(II.EQ.1) YB(J)=YC(K,1)+(DY/2.)
XB(J)=XC(K,1)
T(J)=TH
A1(J)=A1A(I)
A2(J)=A2A(I)
24 CONTINUE
CONTINUE
RETURN
END
CCC

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,NPSY,NAST

```

```

COMMON/EL./A,B,N          00008360
C C C C
ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY      00008370
AND APPLY BEARING STRESS IN A COSINUSOIDAL      00008380
DISTRIBUTION      00008390
      00008400
      00008410
      00008420
      00008430
      00008440
      00008450
      00008460
      00008470
      00008480
      00008490
      00008500
      00008510
      00008520
      00008530
      00008540
      00008550
      00008560
      00008570
      00008580
      00008590
      00008600
      00008610
      00008620
      00008630
      00008640
      00008650
      00008660
      00008670
      00008680
      00008690
      00008700
      00008710
      00008720
      00008730
      00008740
      00008750
      00008760
      00008770
      00008780
      00008790
SPECIFY COORDINATES AROUND CHARACTERISTIC CIRCLE      00008800
AN WHICH STRESSES ARE NEEDED FOR THE HOFFMAN/      00008810
TSAI-HILL FAILURE CRITERIA      00008820
      00008830
IMPLICIT REALN8(A-H,O-Z)          00008840
DIMENSION X(400),Y(400),RCA(2),RCB(2),NRC(2)      00008850
COMMON/CMT2/X,Y,NPST,NAST      00008860
COMMON/RC/RCA,RCB,NRC      00008870
COMMON/ELP/AX,BX,NOUT      00008880
RAD=DARCOS(-0.1D1)/180.      00008890
N1=NRC(K)          00008900
DO 40 I=1,N1      00008910
TINCR=360./NRC(K)      00008920
THETA=(I-1)*TINCR*RAD      00008930
C=DCOS(THETA)      00008940
S=DSIN(THETA)      00008950

```

```

R=DSQRT(1./((CX**2/RCA(K)**2)+(SX**2/RCB(K)**2)))
X(I+NOUT)=RXDCOS(THETA)
Y(I+NOUT)=RXDSIN(THETA)
40 CONTINUE
RETURN
END
C
C
C
C
SUBROUTINE PSTRUSS(K,NCS,BPR,IL)
SPECIFY DISCRETE COORDINATES OF POINTS AT WHICH
STRESSES ARE REQUIRED FOR THE POINT STRESS
CRITERION
IMPLICIT REALN8(A-H,O-Z)
DIMENSION X(400),Y(400),DONT(2),DOBR(2)
DIMENSION DOSE(2),NPLY(2),NUMPLY(2),ANO(5,2)
DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSE(2,2)
COMMON/ELP/AX,BX,NOUT
COMMON/CMT2/X,Y,NPST,NAST
COMMON/PSC1/DONI,DOBR,DOSE
COMMON/LYP/NPLY,NUMPLY,ANO,IPLY
COMMON/ADV/AONT,AOBR,AOSE
ANT=DONT(K)
ABR=DOBR(K)
ASE=DOSE(K)
IF(BPR.EQ.0.0.OR.BPR.LT.1.0) GO TO 25
ANT=AONT(IL,K)
ABR=AOBR(IL,K)
ASE=AOSE(IL,K)
55 CONTINUE
L=NOUT+1
SQ=1.0
IF(NCS.EQ.1) SQ=-1.0
X(L)=0.
Y(L)=ANT+BX
X(L+1)=SQ*(AX+ABR)
Y(L+1)=0.
X(L+2)=SQ*(AX+ASE)
Y(L+2)=BX
NPST=3
DO 555 I=1,5
L=NOUT+1+(I-1)
RETURN
END
C
C
C
C
SUBROUTINE AVSYRS(K,NCS,NAVD,BPR,IL)
SPECIFY COORDINATES OF POINTS ALONG WHICH
STRESSES WILL BE AVERAGED FOR THE AVERAGE
STRESS CRITERION
IMPLICIT REALN8(A-H,O-Z)
DIMENSION X(400),Y(400),DONT(2),DOBR(2)
DIMENSION DOSE(2),NPLY(2),NUMPLY(2),ANO(5,2)
DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSE(2,2)
COMMON/ADV/AONT,AOBR,AOSE

```

```

COMMON/EL./AX,BX,NOUT 00009560
COMMON/CMT2/X,Y,NPST,NAST 00009570
COMMON/FSCL/DONT,DOBR,DOSO 00009580
COMMON/LYP/NPLY,HUMPLY,ANQ,IPLY 00009590
ANT=DONT(K) 00009600
ABR=DOBR(K) 00009610
ASO=DOSO(K) 00009620
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 23 00009630
ANT=AONT(IL,K) 00009640
ABR=AOPR(IL,K) 00009650
ASO=AOSO(IL,K) 00009660
25 CONTINUE 00009670
L=NOUT 00009680
SG=1.0 00009690
IF(NCS.EQ.1) SG=-1.0 00009700
ANDO=ANT/FLOAT(NAVD) 00009710
DO 20 I=1,NAVD 00009720
L=L+1 00009730
X(L)=0. 00009740
20 Y(L)=BX+ANDO/2.+((I-1)*ANDO 00009750
ANSO=ASO/FLOAT(NAVD) 00009760
DO 30 I=1,NAVD 00009770
L=L+1 00009780
X(L)=SG*(BX+ANSO/2.+((I-1)*ANSO) 00009790
30 Y(L)=BX 00009800
ANBR=ABR/FLOAT(NAVD) 00009810
DO 40 I=1,NAVD 00009820
L=L+1 00009830
X(L)=SG*(AX+ANBR/2.+((I-1)*ANBR) 00009840
40 Y(L)=0. 00009850
NAST=3*NAVD 00009860
N1=NOUT+1 00009870
N2=N1+NAST 00009880
NN=NCUT+3*NAVD 00009890
RETURN 00009900
END 00009910
00009920
00009930
00009940
00009950
00009960
00009970
00009980
00009990
00010000
00010010
00010020
00010030
00010040
00010050
00010060
00010070
00010080
00010090
00010100
00010110
00010120
00010130
00010140
00010150

SUBROUTINE FIGEOM(H,KJ,NOPT4,ITT)

FIGEOM PERFORMS A FINITE GEOMETRY ANALYSIS
USING THE BOUNDARY COLLOCATION TECHNIQUE

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(3,3),WK(25),AI(3,3),AZ(5),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(1488)

```

C
C
C
C
C
C
C
C

C

```

C      AMA1RX CALCULATES THE LAMINATE 'A' MATRIX          00010160
C      CALL AMATRX(H,KJ)                                00010170
C      I=3                                              03010180
C      IDGT=4                                         06010190
C      A=3                                              00010200
C      LINV2F INVERTS THE 'A' MATRIX                   00010210
C      CALL LINV2F(A,N,IA,AI,1DGT,IK,IER)            00010220
C      NDEG=4                                         00010230
C      AZ(1)=AI(1,1)/AI(2,2)                          00010240
C      AZ(2)=-2.*AI(1,3)/AI(2,2)                      00010250
C      AZ(3)=(2.*AI(1,2)+AI(3,3))/AI(2,2)           00010260
C      AZ(4)=-2.*AI(2,3)/AI(2,2)                      00010270
C      AZ(5)=1.000                                     00010280
C      ZRPOLY FINDS THE ROOTS OF THE CHARACTERISTIC EQUATION 00010290
C      CALL ZRPOLY(AZ,NDEG,Z,IER)                    00010300
C      Z(2) AND Z(4) ARE THE COMPLEX CONJUGATES OF Z(1)    00010310
C      AND Z(3) RESPECTIVELY                         00010320
C      R1=Z(1)                                         00010330
C      R2=Z(3)                                         00010340
C      THE TWO ROOTS MUST BE CHECKED FOR A UNITARY COMPONENT 00010350
C      IN EITHER THE REAL OR IMAGINARY PART; SUCH AN        00010360
C      OCCURANCE SIGNIFIES A QUASI-ISOTROPIC LAYUP AND     00010370
C      THE VALUE MUST BE PERTURBED SLIGHTLY IN ORDER TO     00010380
C      AVOID A SINGULAR MATRIX                         00010390
C
C      CH(1)=R1                                         00010400
C      CH(2)=(0.0,-1.0)*R1                            00010410
C      CH(3)=R2                                         00010420
C      CH(4)=(0.0,-1.0)*R2                            00010430
C      DO 30 IJK=1,4                                    00010440
C      IF(DABS(CH(IJK)).LT.1.0D-10) CH(IJK)=1.0D-10   00010450
C      AR=DABS(CH(IJK))                               00010460
C      IF(AR.LE.1.0) GO TO 11                           00010470
C      GO TO 32                                         00010480
C 31  IF((1.0-AR).LT.0.02) CH(IJK)=0.98             00010490
C 32  GO TO 30                                         00010500
C 33  IF((AR-1.0).LT.0.02) CH(IJK)=1.02             00010510
C 30  CONTINUE                                         00010520
C      R1=DCMPLX(CH(1),CH(2))                        00010530
C      R2=DCMPLX(CH(3),CH(4))                        00010540
C
C      CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS 00010550
C
C      P1=AI(1,1)*R1**2+AI(1,2)-AI(1,3)*R1           00010560
C      P2=AI(1,1)*R2**2+AI(1,2)-AI(1,3)*R2           00010570
C      Q1=AI(2,2)/R1+AI(1,1)*R1-AI(2,3)              00010580
C      Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)              00010590
C
C      INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS 00010600
C
C      NT4=4*NT                                         00010610
C      NT8=8*NT                                         00010620
C
C      00010630
C      00010640
C      00010650
C      00010660
C      00010670
C      00010680
C      00010690
C      00010700
C      00010710
C      00010720
C      00010730
C      00010740
C      00010750

```

```

NT8P4=8*N, +4 00010760
NT8P2=8*NT+2 00010770
NT8P1=8*NT+1 00010780
NB2=2*N8 00010790
NWK=NT8P1*(NT8P1+2) 00010800
CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2, 00010810
INT8P1,NB2,NWK,WA,HKK,AI,NOPT4,KJ,ITT) 00010820
RETURN 00010830
END 00010840
00010850
00010860
00010870
SUBROUTINE AMATRX(H,K) 00010880
ASSEMBLE THE A MATRIX 00010890
00010900
00010910
00010920
IMPLICIT REAL*8(A-H,O-Z) 00010930
DIMENSION A(3,3),ANG(5,2),H(2),NPLY(2),NUMPLY(2) 00010940
DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2) 00010950
DIMENSION IPLY(100,2) 00010960
COMMON/MOD/E1,E2,O12,V12,V21
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY 00010970
COMMON/AMT/A 00010980
THKNES=NPLY(K)*H(K) 00010990
DENO=1.-E2(K)*V12(K)*V22/E1(K) 00011000
Q11=E1(K)/DENO 00011010
Q22=E2(K)/DENO 00011020
Q12=V12(K)*Q22 00011030
Q21=Q12 00011040
Q33=O12(K) 00011050
DO 10 I=1,3 00011060
DO 10 J=1,3 00011070
10 A(I,J)=0. 00011080
NN=NPLY(K) 00011090
DO 20 I=1,NN 00011100
T=H(K) 00011110
LP=IPLY(I,K) 00011120
THTAI=ANG(LP,K)*DARCOS(-1.00)/180.00 00011130
C=DCOS(THTAI) 00011140
S=DSIN(THTAI) 00011150
A(1,1)=(Q11*C*H*4+2.*Q11*S*H*3)*(Q12+C*H*4+S*H*3)+Q22*(S*H*4)*H*T+A(1,1) 00011160
A(2,2)=(Q11*C*H*4+2.*Q12+C*H*3)*(Q12+C*H*4+S*H*3)+Q22*(C*H*4)*H*T+A(2,2) 00011170
A(1,2)=((Q11+Q22-4.*Q33)*C*H*3*S+Q12*(C*H*4+S*H*3))*H*T+A(1,2) 00011180
A(2,1)=A(1,2) 00011190
A(3,3)=((Q11+Q22-2.*Q33)*C*H*3*S+Q33*(C*H*4+S*H*3))*H*T+A(3,3) 00011200
A(1,3)=((Q11-Q12-2.*Q33)*C*H*3*S+(Q12-Q22+2.*Q33)*S*H*3*C)*H*T+A(1,3) 00011210
A(2,3)=((Q11-Q12-2.*Q33)*S*H*3*C+(Q12-Q22+2.*Q33)*C*H*3*S)*H*T+A(2,3) 00011220
A(3,2)=A(2,3) 00011230
A(3,1)=A(1,3) 00011240
20 CONTINUE 00011250
DO 53 I=1,3 00011260
DO 53 J=1,3 00011270
A(I,J)=A(I,J)/THKNES 00011280
53 CONTINUE 00011290
RETURN 00011300
END 00011310
00011320
00011330
00011340
SUBROUTINE CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2, 00011350

```

CCC
CCC

C
C

INT8P1,NB4,,NWK,WK,WA,WKK,AMAT,NOPT4,KJ,ITT)	00011360
C	00011370
C	00011380
C	00011390
C	00011400
C	00011410
C	00011420
C	00011430
C	00011440
C	00011450
C	00011460
C	00011470
C	00011480
C	00011490
C	00011500
C	00011510
C	00011520
C	00011530
C	00011540
C	00011550
C	00011560
C	00011570
C	00011580
C	00011590
C	00011600
C	00011610
C	00011620
C	00011630
C	00011640
C	00011650
C	00011660
C	00011670
C	00011680
C	00011690
A=AX	00011700
B=BX	00011710
CO=(0.0,1.0)	00011720
RB11=(Q1-P1*W1)/(A CO*W1*B)	00011730
RB21=(Q2-P2*W2)/(A CO*W2*B)	00011740
REALR1=R1	00011750
REALR2=R2	00011760
REALP1=P1	00011770
REALP2=P2	00011780
REALQ1=Q1	00011790
REALQ2=Q2	00011800
RRB11=RB11	00011810
RRB21=RB21	00011820
AIMGR1=COMR1	00011830
AIMGR2=COMR2	00011840
AIMGP1=COMP1	00011850
AIMGP2=COMP2	00011860
AIMGQ1=COMQ1	00011870
AIMGQ2=COMQ2	00011880
ARB11=COMRB11	00011890
ARB21=COMRB21	00011900
R1B=DCMPLX(REALR1,AIMGR1)	00011910
R2B=DCMPLX(REALR2,AIMGR2)	00011920
P1B=DCMPLX(REALP1,AIMGP1)	00011930
P2B=DCMPLX(REALP2,AIMGP2)	00011940
Q1B=DCMPLX(REALQ1,AIMGQ1)	00011950
Q2B=DCMPLX(REALQ2,AIMGQ2)	

```

R811B=DCI..,LX(RRB11,ARB11) 00011960
R821B=DCMPLX(RRB21,ARB21) 00011970
JJJ=0 00011980
DO 1000 I=1,NB 00011990
J=I+2 00012000
THTAI=THTA(I)*DARCOS(-1.0D0)/180.0D 00012010
C=DCOS(THTAI) 00012020
S=DSIN(THTAI) 00012030
P11=CXP1+SMQ1 00012040
P12=CXP2+SMQ2 00012050
P21=-SMQ1+CXP1 00012060
P22=-SMQ2+CXP2 00012070
T11=(CXCR1*R1+SMR1-SMC-R1) 00012080
T12=(CXCR2*R2+SMR2-SMC-R2) 00012090
T21=(-CMR1*R1+CMC-(CMC-SMR1)) 00012100
T22=(-CMR2*R2+CMC-(CMC-SMR2)) 00012110
Z1=X(I)+R1*Y(I) 00012120
Z2=X(I)+R2*Y(I) 00012130
Z11=CDOSQRT(Z1*Z1-AXA-R1*R1*B*B) 00012140
Z22=CDOSQRT(Z2*Z2-AXA-R2*R2*B*B) 00012150
REAL1=Z1 00012160
AIM01=-COXZ11 00012170
IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00012180
IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00012190
Z11=DCMPLX(REAL1,AIM01) 00012200
REAL2=Z22 00012210
AIM02=-COXZ22 00012220
IF(DABS(REAL2).LE.1.D-16)REAL2=0.0D0 00012230
IF(DABS(AIM02).LE.1.D-16)AIM02=0.0D0 00012240
Z22=DCMPLX(REAL2,AIM02) 00012250
XETA1=(Z1+Z11)/(A-COMXR1*B) 00012260
IF(CDABS(XETA1).LT.0.999) GO TO 300 00012270
GO TO 310 00012280
300 Z11=-Z11 00012290
XETA1=(Z1+Z11)/(A-COMXR1*B) 00012300
310 XETA2=(Z2+Z22)/(A-COMXR2*B) 00012310
IF(CDABS(XETA2).LT.0.999) GO TO 320 00012320
GO TO 330 00012330
320 Z22=-Z22 00012340
XETA2=(Z2+Z22)/(A-COMXR2*B) 00012350
330 CONTINUE 00012360
JJJ=JJJ+1 00012370
00012380
C C C NORMAL & TANGENTIAL STRESS BOUNDARY CONDITIONS ARE IMPOSED 00012390
DO 5 N=1,NT 00012400
NP=N 00012410
CM(J-1,N)=NP*XETA1*NNP*T11/Z11 00012420
CM(J-1,2*NT+N)=NP*XETA2*NNP*T12/Z22 00012430
CM(J,N)=NP*XETA1*NNP*T21/Z11 00012440
CM(J,2*NT+N)=NP*XETA2*NNP*T22/Z22 00012450
NN=-N 00012460
CM(J-1,NT+N)=NN*XETA1*NNN*T11/Z11 00012470
CM(J-1,3*NT+N)=NN*XETA2*NNN*T12/Z22 00012480
CM(J,NT+N)=NN*XETA1*NNN*T21/Z11 00012490
CM(J,3*NT+N)=NN*XETA2*NNN*T22/Z22 00012500
5 CONTINUE 00012510
CM(J-1,NT8+1)=T11/Z11 00012520
CM(J-1,NT8+2)=T12/Z22 00012530
CM(J,NT8+1)=T21/Z11 00012540
00012550

```

```

1000 CM(J,NT8+4)=T22/222          0001256J
      CONTINUE                      00012570
      DO 195 I=1,NB2                 ...
      DO 196 J=1,NT4                 00012580
      REAL1=CM(I,J)                  00012590
      AIM01=-COVCM(I,J)              00012600
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00012610
      IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00012620
      CM(I,J)=DCMPLX(REAL1,AIM01)       00012630
      AIM02=-AIM01                     00012640
      CM(I,NT4+J)=DCMPLX(REAL1,AIM02)     00012650
196  CONTINUE                      00012660
195  CONTINUE                      00012670
      DO 295 I=1,NB2                 00012680
      DO 296 J=1,2                   ...
      REAL1=CM(I,NT8+J)                00012690
      AIM01=-COVCM(I,NT8+J)           00012700
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00012710
      IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00012720
      CM(I,NT8+J)=DCMPLX(REAL1,AIM01)     00012730
      AIM02=-AIM01                     00012740
      CM(I,NT8+2+J)=DCMPLX(REAL1,AIM02)     00012750
      AIM02=-AIM01                     00012760
296  CONTINUE                      00012770
295  CONTINUE                      00012780
      SV11=(P2*XQ1B-Q2*XPI1)/(Q1*XPI2-Q2*XPI1) 00012800
      SV12=(P2*XQ2B-Q2*XPI2)/(Q1*XPI2-Q2*XPI1) 00012810
      SV21=(Q1*XPI1-Q1*BMP1)/(Q1*XPI2-Q2*XPI1) 00012820
      SV22=(Q1*XPI2-Q2*BMP1)/(Q1*XPI2-Q2*XPI1) 00012830
      DO 139 I=1,NB2                 00012840
C
C   IMPOSE RIGID BODY ROTATION CONDITION          00012850
C
      CM(I,2*NT+1)=-CM(I,1)*RB21/RB11+CM(I,2*NT+1) 00012860
      CM(I,4*NT+1)=-CM(I,1)*RB11B/RB11+CM(I,4*NT+1) 00012870
      CM(I,6*NT+1)=-CM(I,1)*RB21B/RB11+CM(I,6*NT+1) 00012880
      CM(I,1)=(0.0,0.0)                  00012890
C
C   IMPOSE SINGLE-VALUEDNESS CONDITION            00012890
C
      CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)      00012900
      CM(I,NT8+4)=CM(I,NT8+1)*SV12+CM(I,NT8+4)      00012910
      CM(I,NT8+3)=CM(I,NT8+2)*SV21+CM(I,NT8+3)      00012920
      CM(I,NT8+4)=CM(I,NT8+2)*SV22+CM(I,NT8+4)      00012930
      CM(I,NT8+1)=(0.0,0.0)                  00012940
      CM(I,NT8+2)=(0.0,0.0)                  00012950
139  CONTINUE                      00013010
      DO 141 I=1,NB2                 00013020
      DO 142 J=2,NT8                 ...
      142  CM(I,J-1)=CM(I,J)           00013030
      CM(I,NT8)=CM(I,NT8+3)             00013040
      CM(I,NT8+1)=CM(I,NT8+4)           00013050
141  CONTINUE                      00013070
      DO 95 I=1,NB2                 00013080
      DO 96 J=1,NT8P1                 ...
      REAL1=CM(I,J)                  00013090
      AIM01=-COVCM(I,J)              ...
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00013110
      IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00013120
      CM(I,J)=DCMPLX(REAL1,AIM01)       00013130
      AIM02=-AIM01                     00013140
      AIM02=-AIM01                     00013150

```

```

      CMC(I,J)=CMPLX(REAL1,AIMG2)          00013150
96  CONTINUE                               00013170
95  CONTINUE                               00013180
      DO 120 I=1,NB                         00013190
      J=I+2                                 00013200
      BC(J-1)=AIN1(I)                      00013210
120  BC(J)=AIN2(I)                      00013220
      DO 130 I=1,NT8P1                     00013230
      DO 100 J=1,NT8P1                     00013240
      CSUM=(0.0,0.0)                       00013250
      DO 110 K=1,NB2                       00013260
110  CSUM=CMC(K,I)*CMC(K,J)+CSUM        00013270
      CMCTCM(I,J)=CSUM                   00013280
100  CONTINUE                               00013290
      DO 130 I=1,NT8P1                     00013300
      CSUM=(0.0,0.0)                       00013310
      DO 140 K=1,NB2                       00013320
140  CSUM=CMC(K,I)*BC(K)+CSUM          00013330
130  RHS(I)=CSUM                        00013340
      IJOB=0                                00013350
      M=1                                    00013360
      CALL LEQ2C(CMCTCM,NT8P1,NT8P1,RHS,M,NT8P1,IJOB,WA,WKK,IER)
      ORHS(1)=-(RHS(2*N1)+RB21+RHS(4*NT)*RB11B+RHS(6*NT)*RB21B)/RB11 00013370
      ORHS(8*NT+1)=RHS(8*NT)*SV11+RHS(8*NT+1)*SV12 00013380
      ORHS(8*NT+2)=RHS(8*NT)*SV21+RHS(8*NT+1)*SV22 00013390
      DO 151 I=2,NT8
151  ORHS(I)=RHS(I-1)                    00013410
00013420
00013430
00013440
00013450
00013460
00013470
00013480
00013490
00013500
00013510
00013520
00013530
00013540
00013550
00013560
00013570
00013580
00013590
00013600
00013610
00013620
00013630
00013640
00013650
00013660
00013670
00013680
00013690
00013700
00013710
00013720
00013730
00013740
00013750

C   STRESS AND STRAIN CALCULATION
C
      NRCS=NOUT+1
      IF(NOPT4.EQ.1.OR.NOPT4.EQ.3) NRCF=NOUT+NRCOUT(KJ)          00013440
      IF(NOPT4.EQ.2) NRCF=NOUT+NPST                         00013450
      IF(NOPT4.EQ.4) NRCF=NOUT+NAST                         00013460
      DO 190 K=1,NRCF                           00013470
      Z1=XOUT(K)+R1*YOUT(K)                         00013480
      Z2=XOUT(K)+R2*YOUT(K)                         00013490
      Z11=CDSQRT(Z1*X1-A*A -R1*R1*B*B)           00013500
      Z22=CDSQRT(Z2*X2-A*A -R2*R2*B*B)           00013510
      XETA1=(Z1+Z11)/(A-C0*R1*B)                  00013520
      IF(CDABS(XETA1).LT.0.999) GO TO 400          00013530
      GO TO 410
400  Z11=-Z11
      XETA1=(Z1+Z11)/(A-C0*R1*B)                  00013540
410  XETA2=(Z2+Z22)/(A-C0*R2*B)                  00013550
      IF(CDABS(XETA2).LT.0.999) GO TO 420          00013560
      GO TO 430
420  Z22=-Z22
      XETA2=(Z2+Z22)/(A-C0*R2*B)                  00013570
430  CONTINUE
      PHI1DP=(0.0,0.0)                         00013580
      PHI2DP=(0.0,0.0)                         00013590
      PHI1DN=(0.0,0.0)                         00013600
      PHI2DN=(0.0,0.0)                         00013610
      PHI1P=(0.0,0.0)                          00013620
      PHI2P=(0.0,0.0)                          00013630
      PHI1N=(0.0,0.0)                          00013640
      PHI2N=(0.0,0.0)                          00013650
      DO 170 N=1,NT
      NP=N

```

```

NN=-N
PHI1DP=NP*XETA1*N*GRHS(N)/Z11+PHI1DP 00013760
PHI1DII=NN*XETA1**NN*GRHS(NT+N)/Z11+PHI1DII 00013770
PHI2DP=NP*XETA2*N*GRHS(2*NT+N)/Z22+PHI2DP 00013780
PHI2DN=NN*XETA2**NN*GRHS(3*NT+N)/Z22+PHI2DN 00013790
PHI1P=XETA1**NP*GRHS(N)+PHI1P 00013800
PHI1N=XETA1**NN*GRHS(NT+N)+PHI1N 00013810
PHI2P=XETA2**NP*GRHS(2*NT+N)+PHI2P 00013820
PHI2N=XETA2**NN*GRHS(3*NT+N)+PHI2N 00013830
170 CONTINUE 00013840
PHI1D=PHI1DP+PHI1DN*GRHS(8*NN+1)/Z11 00013850
PHI2D=PHI2DP+PHI2DN*GRHS(8*NT+2)/Z22 00013860
PHI1=PHI1P+PHI1N*GRHS(8*NT+1)*CDLOG(XETA1) 00013880
PHI2=PHI2P+PHI2N*GRHS(8*NT+2)*CDLOG(XETA2) 00013890
SGMAX=2.*R1*(R1*PHI1D+R2*R2*PHI2D) 00013900
SGMAY=2.*R1*(PHI1D+PHI2D) 00013910
SGMAXY=2.*R1*(R1*PHI1D+R2*PHI2D) 00013920
EPSX=AMAT(1,1)*SGMAX+AMAT(1,2)*SGMAY+AMAT(1,3)*SGMAXY 00013930
EPSY=AMAT(2,1)*SGMAX+AMAT(2,2)*SGMAY+AMAT(2,3)*SGMAXY 00013940
EPSXY=AMAT(3,1)*SGMAX+AMAT(3,2)*SGMAY+AMAT(3,3)*SGMAXY 00013950
U=2.*R1*(P1*PHI1+P2*PHI2) 00013960
V=2.*R1*(Q1*PHI1+Q2*PHI2) 00013970
PI=DARCDOS(-1,DO) 00013980
IF(XOUT(K).LT.0.,AND,YOUT(K),GT.0.) 00013990
+TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI 00014000
IF(XOUT(K).LT.0.,AND,YOUT(K),GT.0.) 00014010
+TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180. 00014020
IF(XOUT(K).LT.0.,AND,YOUT(K),LT.0.) 00014030
+TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+180. 00014040
IF(YOUT(K).LT.0.,AND,XOUT(K),GT.0.) 00014050
+TETAA=DATAN(YOUT(K)/XOUT(K))*180./PI+360. 00014060
C=DCOS(TETAA*PI/180.) 00014070
S=DSIN(TETAA*PI/180.) 00014080
SGMAR=C**2*SGMAX+S**2*SGMAY+2.*C*S*SGMAXY 00014090
SGMAT=C**2*SGMAX+C**2*SGMAY-2.*C*S*SGMAXY 00014100
SGMART=-C**2*SGMAX+C**2*SGMAY+(C**2-S**2)*SGMAXY 00014110
EPSR=C**2*EPSX+S**2*EPSY+C*S*EPSXY 00014120
EPST=C**2*EPSX+C**2*EPSY-C*S*EPSXY 00014130
EPSRT=2.*(-C*S*EPSX+C*S*EPSY+(C**2-S**2)*(EPSXY/2.)) 00014140
UR=U*C+V*S 00014150
RTHT(K)=TETAA 00014160
REPX(K)=EPSX 00014170
REPY(K)=EPSY 00014180
REPXY(K)=EPSXY 00014190
ASX(K)=SGMAX 00014200
ASY(K)=SGMAXY 00014210
FUR(K)=UR 00014220
FTHT(K)=TETAA 00014230
FSMR(K)=SGMAR 00014240
190 CONTINUE 00014250
RETURN 00014260
END 00014270
00014280
00014290
00014300
00014310
00014320
00014330
00014340
00014350
C
C
C
SUBROUTINE FBOLT(ANOK,H,K,NOPT1,IM1)
C
C
FBOLT CALCULATES THE INDIVIDUAL PLY FOUNDATION
MODULI AND THE INDIVIDUAL PLY LOADS

```

```

C
C
IMPLICIT REAL*8(A-H,O-Z)                                00014360
DIMENSION ATETAA(400),ANG(5,2),ASIGR(400),ASIGRT(400),H(2) 00014370
DIMENSION ASIG1(400),ASIG2(400),ASIG6(400),UR(400),ANGK(5,2) 00014380
DIMENSION FSMR(400),PLXPT(100)                         00014390
DIMENSION IPLY(100,2),NPLY(2),NUMPLY(2)                 00014400
DIMENSION FK(100),PLX(100)                            00014410
DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2)      00014420
DIMENSION RCA(2),RCB(2),NRC(2)                         00014430
COMMON//STRESS//ASIGR,ASIGRT,ASIG1,ASIG2,ASI02,ASI06    00014440
COMMON//ELP//AX,BX,NOUT                                00014450
COMMON//FB1//BSTR,XSTR                                00014460
COMMON//LYP//NPLY,NUMPLY,ANG,IPLY                      00014470
COMMON//FB2//UR,ATETAA,FSMR                           00014480
COMMON//MOD//E11,E22,ESS,PMU12,PMU21                  00014490
COMMON//RC//RCA,RCB,NRC                               00014500
COMMON//FCT//PLXPT                                    00014510
RAD=DARCOS(-0.1D1)/180.
THKTOT=NPLY(K)*H(K)                                 00014520
NN=NUMPLY(K)                                         00014530
C
C
CALCULATE DELEFF                                     00014540
C
WORK=0.
PLOADX=0.
IF(K.EQ.1) PLD=0.
DO 210 KK=1,NOUT                                     00014550
TH1=ATETAA(KK+1)-ATETAA(KK)                         00014560
TH2=(ATETAA(KK)+ATETAA(KK+1))/2.                    00014570
THETA=TH2*RAD                                       00014580
C=DCOS(THETA)                                       00014590
S=DSIN(THETA)                                       00014600
R=DSQRT(1./(C**2/A*X**2+S**2/B*X**2))            00014610
FORCE=((FSMR(KK)+FSMR(KK+1))/2.)*R*TH1*RAD*THKTOT 00014620
WORK=NOPTK*FORCE,5*((UR(KK)+UR(KK+1))/2.)
PLOADX=PLOADX+FORCE*C                               00014630
210 CONTINUE                                         00014640
PLD=PLD+PLOADX                                      00014650
DELEFF=WORK/PLOADX                                  00014660
C
C
COMPUTE PLY STRESSES FROM LAMINATE STRAINS          00014670
C
(SIOMA)R,U,R0 +(Q)*(EPS)R,O,R0                      00014680
C
NN=NPLY(K)
DO 100 J=1,NN                                       00014690
LP=IPLY(J,K)
THETA=ANG(LP,K)*RAD                                00014700
L11=1
L12=NOUT
NCAS=1
CALL QMATX(K,L11,L12,NCAS,NOPT1,RAD,THETA)        00014710
C
C
INTEGRATE AROUND CIRCULAR BOUNDARY FOR              00014720
C

```

```

CCCCC INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION MODULI          00014960
      NNN=L12-1                                                 00014970
      PLOADX=0                                                 00014980
      HK=0                                                 00014990
      DO 70 I=L11,NNN                                         00015000
      TH1=ATETAA(I+1)-ATETAA(I)                                00015010
      TH2=(ATETAA(I)+ATETAA(I+1))/2.                           00015020
      THETA=TH2*RAD                                           00015030
      C=DCOS(THETA)                                           00015040
      S=DSIN(THETA)                                           00015050
      X=(C*X**2/AXX**2+S*X*2/BXX*X2))                         00015060
      FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*XRMT(H1)*RAD*XH(K)    00015070
      FORCRT=(ASIGRT(I)+ASIGRT(I+1))/2.)*XRMT(H1)*RAD*XH(K)     00015080
      PLOADX=PLOADX+FORCR*C-FORCRT*S                         00015090
70 CONTINUE
      FKI(J)=DABS(PLOADX/(H(K)*DELEFF))
      PLOADX+(K-1)*NPLY(1)=PLOADX
100 CONTINUE
      NT=NMPLY(K)
      NN=NPLY(K)
      DO 310 II=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=NMPLY(K)

CCCCC COMPUTE TOTAL BEARING LOAD
      IF(K.EQ.1) GO TO 611
      PLXTOT=0
      TH=H(1)*NPLY(1)+H(2)*NPLY(2)
      BLOAD=(BSTR*DA)*COS(-1.000)*GX*TH)/2.
      NN=NPLY(1)+NPLY(2)
      DO 212 II=1,NN
      PLXTOT=PLXTOT+PLX(II)
      PLXTOT=PLXTOT+PLX(I)
212 CONTINUE
611 CONTINUE
      RETURN
      END

CCCCC SUBROUTINE FCRT(SALOW,H,WTH,AST,K,NOPT1,NOPT4,BPR,NAVD,IL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ASI01(400),ASI02(400),ASI06(400),ASIGR(400)
      DIMENSION UR(400),FSMR(400),ATETAA(400),ASIGRT(400),NUMPLY(2)
      DIMENSION ANG(5,2),IPLY(100,2),HFMC(5,2),PLXPT(100),NPLY(2)
      DIMENSION PNS(5,2),PBR(5,2),P30(5,2),PALT(5,2)
      00015300
      00015310
      00015320
      00015330
      00015340
      00015350
      00015360
      00015370
      00015380
      00015390
      00015400
      00015410
      00015420
      00015430
      00015440
      00015450
      00015460
      00015470
      00015480
      00015490
      00015500
      00015510
      00015520
      00015530
      00015540
      00015550

```

```

DIMENSION H(2),SALOH(2),SX(400),SXY(400),RCA(2),RCB(2),NRC(2)      00015560
DIMENSION AEPS1(400),PFL(5,2),PSTC(3,5,2)                      00015570
DIMENSION BPSTS(2,10,2,3)                                         00015580
COMMON/BP1/BPSTS                                              00015590
COMMON/RC/RCA,RCB,NRC                                         00015600
COMMON/FB1/BSTR,XSTR                                         00015610
COMMON/STRSS2/AEPS1                                         00015620
COMMON/FB2/UR,ATETAA,FSMR                                     00015630
COMMON/FCT/FLXPT                                           00015640
COMMON/ELP/AX,BX,NOUT                                         00015650
COMMON/HFF/HFMC                                           00015660
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY                           00015670
COMMON/FAL3/PFL                                         00015680
COMMON/STRESS/ASIGR,ASIORT,ASI01,ASI02,ASI06               00015690
COMMON/PSC2/PSTC                                         00015710
COMMON/PSC3/SX,SXY                                         00015720
RAD=DARCO3(-0.1D1)/180.                                     00015730
IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,89) K    00015740
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          HOFFMAN/TSAI-HILL CRITERIA
C
C
C          WRITE(6,10)
10 FORMAT(//,' HOFFMAN/TSAI-HILL CRITERION',//)
L11=NOUT+1
L12=L11+NRC(K)
NCAS=2
NN=NUMPLY(K)
DO 402 I=1,NN
THETA=ANG(I,K)*RAD
CALL QMATX(K,L11,L12,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=ASI01(J)/DSB
S2=ASI02(J)/DSB
S6=ASI06(J)/DSB
CALL HOFF(S1,S2,S6,A,B,K)
NNN=NOUT+J
C
C          FOR EACH PLY TYPE FIND THE LOCATION AND MAGNITUDE
C          OF THE HIGHEST HOFFMAN/TSAI-HILL FAILURE INDEX VALUE
C
C
C          P1=(-B+DSQRT(B**2+4*A))/(2.*A)
P1=(-B+DSQRT(B**2+4*A))/(2.*A)                                00016080
P2=(-B-DSQRT(B**2+4*A))/(2.*A)                                00016090
IF(P1.LT.0.D0) PF=P2                                         00016100
IF(P2.LT.0.D0) PF=P1                                         00016110
IF(P1.LT.P2.AND.P1.GT.0.D0) PF=P1                           00016120
IF(P2.LT.P1.AND.P2.GT.0.D0) PF=P2                           00016130
IF(DABS(PF).GT.PFAIL) GO TO 480                            00016140
IF(DABS(PF).GT.PFAIL) GO TO 480                            00016150

```

```

PFAIL=DAB(,PF)
AMAX=A
BMAX=B
LOC=J
480 CONTINUE
404 CONTINUE
A=AMAX
B=BMAX
THTA=ATETA(AHOUT+LOC)

THE CORRESPONDING FAILURE LOAD IS OBTAINED FROM THE INDEX VALUE

XULT=HFMC(1,K)
IF(ASIC1(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(ASIC6(LOC).LT.0.) SULT=-HFMC(3,K)
SR1=ASIC1(LOC)/XULT
SR2=ASIC2(LOC)/YULT
SR6=ASIC6(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 HAS BEEN 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(3FAIL.GT.PFL(I,K)) SFAIL=PFL(I,K)
PLFL=SFAIL*WTHICK(K)*NPY(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

POINT STRESS CRITERION

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
L11=NOUT+1
L12=L11+2
IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) NN=1
DO 100 I=1,NN
  THFTA=ANO(I,K)*RAD
  CALL QMATX(K,L11,L12,NCAS,NOPT1,RAD,THETA)
100 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/(ASTWTHHH(K)*NPLY(K))
BPSTS(K,I,IL,2)=SX(2)*FAC/(ASTWTHHH(K)*NPLY(K))
BPSTS(K,I,IL,3)=SXY(3)*FAC/(ASTWTHHH(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=DSNMPSTC(1,I,K)/DABS(SX(1))
IF(SX(1).LT.0.) PNT=DSNMPSTC(2,I,K)/DABS(SX(1))
PBN=DSNMPSTC(3,I,K)/DABS(SX(2))
PSH=DSNMPSTC(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 ',/
      //, ' SHEAR-OUT 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=NMPLY(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*WTHHH(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*WTHHH(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*WTHHH(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

```

```

40 CONTINUE
C
      IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,55)
55 FORMAT(//,' AVERAGE STRESS CRITERION ',//)
      L1=NOUT+1
      NN=NUMPLY(K)
      NCASJ=2
      IF(BPK.EQ.1.0.AND.NOPT1.EQ.2) NN=1
      DO 105 I=1,NN
      L12=NOUT+3*NAVD
      THETA=ANOC(I,K)*RAD
      CALL QMATX(K,L1,L12,NCAS,NOPT1,RAD,THETA)

      CALCULATE AVERAGE STRESS

      SUM=0.
      N1=1
      N2=NAVD
200  DO 200 IJ=N1,N2
      SUM=SUM+SXY(IJ)
      AS1=SUM/FLOAT(NAVD)
      N1=NAVD+1
      N2=2*NAVD
      SUM=0.
      DO 215 IJ=N1,N2
      SUM=SUM+SXY(IJ)
      AS2=SUM/FLOAT(NAVD)
      SUM=0.
      N1=2*NAVD+1
      N2=3*NAVD
      DO 220 IJ=N1,N2
      SUM=SUM+SXY(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)/PLXPY(I)
      BPST5(K,I,IL,1)=AS1*FAC/(AST*WTHKH(K)*NPPLY(K))
      BPST5(K,I,IL,2)=AS3*FAC/(AST*WTHKH(K)*NPPLY(K))
      BPST5(K,I,IL,3)=AS2*FAC/(AST*WTHKH(K)*NPPLY(K))
      GO TO 105
720  CONTINUE
      IF(BPR.EQ.0.0) DSN=DABS(PLXPY(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,ANOC(I,K),PNT,PBN,PSH
75  FORMAT(//,' FOR PLY TYPE NUMBER ',I5,' WITH ',/
      , ' A PLY ORIENTATION OF ',I1,'.',I3,' 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          00017960
PFAIL1=1.0D10                                00017970
PFAIL2=1.0D10                                00017980
PFAIL3=1.0D10                                00017990
DO 718 I=1,N                                  00018000
IF(PFAIL1.GT.PNS(I,K)) NPY1=I                00018010
IF(PFAIL1.GT.PNS(I,K)) PFAIL1=PNS(I,K)      00018020
IF(PFAIL2.GT.PBR(I,K)) NPY2=I                00018030
IF(PFAIL2.GT.PBR(I,K)) PFAIL2=PBR(I,K)      00018040
IF(PFAIL3.GT.PSO(I,K)) NPY3=I                00018050
IF(PFAIL3.GT.PSO(I,K)) PFAIL3=PSO(I,K)      00018060
718 IF(PFAIL1.GE.PFAIL1.OR.PFAIL1.GE.PFAIL3) GO TO 883 00018070
PFAIL1=PFAIL1*XNTHH(K)*NPLY(K)              00018080
WRITE(6,473) PFAIL1                          00018090
WRITE(6,834) PFAIL1                          00018100
GO TO 881                                     00018110
833 IF(PFAIL2.GE.PFAIL1.OR.PFAIL2.GE.PFAIL3) GO TO 882 00018120
PFAIL2=PFAIL2*XNTHH(K)*NPLY(K)              00018130
WRITE(6,473) PFAIL2                          00018140
WRITE(6,835) PFAIL2                          00018150
GO TO 881                                     00018160
832 IF(PFAIL3.GE.PFAIL1.OR.PFAIL3.GE.PFAIL2) GO TO 881 00018170
PFAIL3=PFAIL3*XNTHH(K)*NPLY(K)              00018180
WRITE(6,476) PFAIL3                          00018190
WRITE(6,886) PFAIL3                          00018200
881 CONTINUE                                    00018210
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                                      00018220
C
CCC MAXIMUM STRAIN CRITERION
90 CONTINUE
PAPP=XSTRXH(K)*NPLY(K)*WTH                 00018230
WRITE(6,772)                                   00018240
772 FORMAT(//,' MAXIMUM STRAIN CRITERION ',//)
L11=NOUT+1                                    00018250
L12=L11+NRC(K)                                00018260
NCAS=2                                         00018270
NN=NUMPY(K)                                    00018280
DO 210 I=1,NN                                  00018290
THETA=ANG(I,K)*RAD                           00018300
CALL QMATX(K,L11,L12,NCAS,NOPT1,RAD,THETA)  00018310
STMAX=-1.0D10                                 00018320
N1=NRC(K)                                    00018330
DO 510 J=1,N1                                  00018340
IF(STMAX.LT.DABS(AEPS1(J))) LOC=J           00018350
510 IF(STMAX.LT.DABS(AEPS1(J))) STMAX=DABS(AEPS1(J)) 00018360
HTA=ATETAA(NOUT+LOC)                         00018370
IF(BPR.EQ.1.0) GO TO 511                      00018380
PFL(I,K)=DABS(PLXPT(I)*SALOW(K)/STMAX)     00018390
WRITE(6,774) I,ANG(I,K),HTA,PFL(I,K)        00018400
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                                     00018410

```

```

511 PFL(I,K)=ABS(PAPP*SALOW(K)/STMAX) 00018560
210 CONTINUE 00018570
IF(BPR.EQ.0.0) GO TO 80 00018580
A=1.0D10 00018590
NN=NUMPLY(K) 00018600
DO 514 I=1,NN 00018610
IF(A.GT.PFL(I,K)) NPY=I 00018620
514 IF(A.GT.PFL(I,K)) A=PFL(I,K) 00018630
WRITE(6,778) A 00018640
778 FORMAT(//,' FOR THE OPEN HOLE LAMINATE, FAILURE IS',//,
*' PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
80 RETURN 00018650
END 00018660
00018670
00018680
00018690
00018700
00018710
00018720
00018730
00018740
00018750
00018760
00018770
00018780
00018790
00018800
00018810
00018820
00018830
00018840
00018850
00018860
00018870
00018880
00018890
00018900
00018910
00018920
00018930
00018940
00018950
00018960
00018970
00018980
00018990
00019000
00019010
00019020
00019030
00019040
00019050
00019060
00019070
00019080
00019090
00019100
00019110
00019120
00019130
00019140
00019150

```

CCC
CCC
CCC
CCC

```

SUBROUTINE QMATX(K,L11,L12,NCAS,NOPT1,RAD,THETA)
QMATX PERFORMS BASIC STRESS AND STRAIN
TRANSFORMATIONS

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION ASIGR(400),ASIGRT(400),ASIG1(400),ASIG2(400),ASIG6(400) 00018780
DIMENSION ATETAA(400),AEPSX(400),AEPSY(400),AEPSXY(400) 00018790
DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400) 00018800
DIMENSION AEPS1(400) 00018810
DIMENSION ASX(400),ASXY(400) 00018820
COMMON/XXY1/ASX,ASXY 00018830
COMMON/MOD/E11,E22,ESS,PMU12,PMU21 00018840
COMMON/STRSS2/AEPS1 00018850
COMMON/STRESS/ASIGR,ASIGRT,ASIG1,ASIG2,ASIG6 00018860
COMMON/QMT/ATETAA,AEPSX,AEPSY,AEPSXY 00018870
COMMON/PSC3/SX,SXY 00018880
J=0 00018890
Q11=E11(K)/(1.0-PMU12(K)*PMU21(K)) 00018900
Q12=(PMU21(K)*E11(K))/(1.0-PMU12(K)*PMU21(K)) 00018910
Q22=E22(K)/(1.0-PMU12(K)*PMU21(K)) 00018920
Q66=ESS(K) 00018930
C=DCOS(THETA) 00018940
S=DSIN(THETA) 00018950
BQ11=(Q11*(C**4)+(2.*Q12+(2.*Q66))*S**2)+(Q22*(S**4)) 00018960
BQ12=((Q11+Q22-(4.*Q66))*S**2)*(C**2)+(Q12*(S**4+C**4)) 00018970
BQ16=((Q11-Q12-(2.*Q66))*(S*(C**3)))+((Q12-Q22+(2.*Q66))*(S**3)*C) 00018980
BQ22=(Q11*(S**4))+((2.*Q12+(2.*Q66))*S**2)*(C**2)+(Q22*(C**4)) 00018990
BQ26=((Q11-Q12-(2.*Q66))*C*(S**3))+((Q12-Q22+(2.*Q66))*S*(C**3)) 00019000
BQ66=((Q11+Q22-(2.*Q12+Q66)))*(S**2)*(C**2)+(Q66*((C**4)+(S**4))) 00019010
*)*
DO 40 I=L11,L12 00019020
J=J+1 00019030
IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD 00019040
C=DCOS(THETA) 00019050
S=DSIN(THETA) 00019060
SIGX=BQ11*AEPSY(I)+BQ12*AEPSY(I)+BQ16*AEPSXY(I) 00019070
SIGY=BQ12*AEPSX(I)+BQ22*AEPSY(I)+BQ2L*AEPSXY(I) 00019080
SIGXY=BQ16*AEPSX(I)+BQ26*AEPSY(I)+BQ66*AEPSXY(I) 00019090
SX(J)=SIGX 00019100
SXY(J)=SIGXY 00019110
IF(NOPT1.EQ.2) SX(J)=ASX(I) 00019120
IF(NOPT1.EQ.2) SXY(J)=ASXY(I) 00019130

```

```

ASIGR(I)=SIGXYNSNC+SIGYNSNM2+2.MSIGXYNSMC 00019160
ASIGT(I)=-SIGXNSMC+SIGYNSMS+SIGXY(CNM2-SNM2) 00019170
MSIG1(J)=SIGXNCNM2+SIGYNSNM2+2.MSNCSIGXY 00019180
ASIO2(J)=SIGXNSNM2+SIGYNCNM2-2.MSNCSIGXY 00019190
ASIG6(J)=-CNSSIGX+SIGYNSMS+(CNM2-SNM2)*SIGXY 00019200
AEPS1(J)=AEPSX(I)*CNM2+AEPSY(I)*SNM2+AEPSXY(I)*MSMC 00019210
00019220
40 CONTINUE 00019230
      RETURN 00019240
      END 00019250
00019260
C C 00019270
      SUBROUTINE HOFF(S1,S2,S6,A,B,K) 00019280
      IMPLICIT REAL*8(A-H,O-Z) 00019290
      DIMENSION HFMC(5,2) 00019300
      COMMON/HFF/HFMC 00019310
00019320
C C COMPUTE THE HOFFMAN/TSA1-HILL FAILURE INDEX 00019330
      A=0.0D0 00019340
      B=0.0D0 00019350
      XC=HFMC(1,K) 00019360
      XT=HFMC(2,K) 00019370
      YC=HFMC(3,K) 00019380
      YT=HFMC(4,K) 00019390
      STC=HFMC(5,K) 00019400
      A=(S1*X2-S1*X2)/(XC*XT)+(S2*X2)/(YC*YT)+(S6/STC)*X2 00019410
      IF(XC.EQ.XT.AND.YC.EQ.YT) GO TO 10 00019420
      B=((XC-XT)/(XC*XT))*S1+((YC-YT)/(YC*YT))*S2 00019430
      GO TO 20 00019440
10 CONTINUE 00019450
      B=0.0D0 00019460
20 CONTINUE 00019470
      RETURN 00019480
      END 00019490
00019500
CCCCC C 00019510
      SUBROUTINE CENTD(RF,H,FASSS,FASBS,P,CELP,ITT) 00019520
      IMPLICIT REAL*8(A-H,O-Z) 00019530
      DIMENSION PLYK(100),BARK(100),BARU(100),F(100) 00019540
      DIMENSION H(2),RF(2) 00019550
      DIMENSION AII(100,100),A(2),B(2) 00019560
      DIMENSION NPLY(2) 00019570
      COMMON/PBB/PLYK,BARK,BARU 00019580
      COMMON/AFM/AII,F 00019590
      COMMON/LYP/NPLY 00019600
00019610
C C SET UP THE CENTRAL DIFFERENCE EQUATIONS 00019620
      DO 3 I=1,100 00019630
      DO 3 J=1,100 00019640
3 AIKI,J)=0. 00019650
00019660
C C NECESSARY CONSTANTS ARE FORMED 00019670
00019680
00019690
00019700
00019710
00019720
00019730
00019740
00019750

```

```

C          DO 7 I=1,2          00019760
          A(I)=H(I)*X2/FASSS 00019770
7         B(I)=H(I)*X4/FASBS 00019780
          H12=H(1)/H(2)       00019790
          A1=H(1)*X2/FASSS  00019800
          A2=H(2)*X2/FASSS  00019810
          NP=NPLY(1)+NPLY(2) 00019820
          00019830
          00019840
          00019850
          00019860
          00019870
          00019880
          00019890
          00019900
          00019910
          00019920
          00019930
          00019940
          00019950
          00019960
          00019970
          00019980
          00019990
          00020000
          00020010
          00020020
          00020030
          00020040
          00020050
          00020060
          00020070
          00020080
          00020090
          00020100
          00020110
          00020120
          00020130
          00020140
          00020150
          00020160
          00020170
          00020180
          00020190
          00020200
          00020210
          00020220
          00020230
          00020240
          00020250
          00020260
          00020270
          00020280
          00020290
          00020300
          00020310
          00020320
          00020330
          00020340
          00020350

C          SHEAR AT TOP OF JOINT EQUALS ZERO
C          AII(1,1)=1          00019850
          AII(1,2)=-(2.+A1*NPLYK(2)) 00019860
          AII(1,4)=2.+A1*NPLYK(2)   00019870
          AII(1,5)=-1.             00019880
          F(1)=0.0                 00019890
          00019900
          00019910
          00019920
          00019930
          00019940
          00019950
          00019960
          00019970
          00019980
          00019990
          00020000
          00020010
          00020020
          00020030
          00020040
          00020050
          00020060
          00020070
          00020080
          00020090
          00020100
          00020110
          00020120
          00020130
          00020140
          00020150
          00020160
          00020170
          00020180
          00020190
          00020200
          00020210
          00020220
          00020230
          00020240
          00020250
          00020260
          00020270
          00020280
          00020290
          00020300
          00020310
          00020320
          00020330
          00020340
          00020350

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*X2.*H(1)*FASSS)+RM*(-2.-A1*NPLYK(2)+H(1)*X2
          *FASSS)/FASBS)           00020030
          AII(2,3)=-Z*X4.*H(1)*FASSS+(2*H(1)*X2*X2*NPLYK(1)*H(1)))
          AII(2,4)=Z*X2.*H(1)*FASSS+RM*(2.+A1*NPLYK(2)-(H(1)*X2
          *FASSS)/FASBS)
          AII(2,5)=-R
          F(2)=Z*X2.*H(1)*X3*BARK(1)*BARU(1)
          00020040
          00020050
          00020060
          00020070
          00020080
          00020090
          00020100
          00020110
          00020120
          00020130
          00020140
          00020150
          00020160
          00020170
          00020180
          00020190
          00020200
          00020210
          00020220
          00020230
          00020240
          00020250
          00020260
          00020270
          00020280
          00020290
          00020300
          00020310
          00020320
          00020330
          00020340
          00020350

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)*NPLYK(J-1)
          GO TO 57
56        AII(I,J+1)=-4.-A(1)*NPLYK(2)
57        AII(I,J+2)=6.+(2.*A(1)+B(1))*NPLYK(J)
          IF(J.EQ.N2) GO TO 61
          AII(I,J+3)=-4.-A(1)*NPLYK(J+1)
          GO TO 62
61        AII(I,J+3)=-4.-A(1)*NPLYK(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
          00020160
          00020170
          00020180
          00020190
          00020200
          00020210
          00020220
          00020230
          00020240
          00020250
          00020260
          00020270
          00020280
          00020290
          00020300
          00020310
          00020320
          00020330
          00020340
          00020350

```

```

63 F(I)=2.*X-.4)*XBARK(NPLY(1)-1)*BARU(NPLY(1)-1)          00020360
  X-(2.*A(1)+B(1))*XBARK(J)*BARU(J)                         00020370
59 CONTINUE
55 CONTINUE
C C
      INTERFACE SHEAR ON TOP PLATE = P+DELP
      I=NPLY(1)+3                                              00020430
      J=NPLY(1)                                                 00020440
      AII(I,J)=1.                                              00020450
      AII(I,J+1)=-(2.+A1*XPLYK(NPLY(1)-1))                  00020460
      AII(I,J+3)=2.+A1*XPLYK(NPLY(1)-1)                      00020470
      AII(I,J+4)=-1.                                           00020480
      F(I)=(-2.*XH(1))*3*(P+DELP)/FASBS                     00020490
C C
      SLOPE CONTINUITY
      I=NPLY(1)+4                                              00020530
      J=NPLY(1)                                                 00020540
      AII(I,J)=1.                                              00020550
      AII(I,J+1)=-(2.+A1*XPLYK(NPLY(1)-1)-H(1)*X*2*XFASSS/FASBS) 00020560
      AII(I,J+3)=2.+A1*XPLYK(NPLY(1)-1)-H(1)*X*2*XFASSS/FASBS 00020570
      AII(I,J+4)=-1.                                           00020580
      AII(I,J+5)=-H12*X*3                                     00020590
      AII(I,J+6)=H12*X*3*(2.+A2*XPLYK(NPLY(1)+2)-H(2)*X*2*XFASSS/FASBS) 00020600
      AII(I,J+8)=-H12*X*3*(2.+A2*XPLYK(NPLY(1)+2)-H(2)*X*2*XFASSS/FASBS) 00020610
      AII(I,J+9)=H12*X*3                                     00020620
      F(I)=0.                                                 00020630
C C
      MOMENT CONTINUITY
      I=NPLY(1)+5                                              00020650
      J=NPLY(1)+1                                              00020660
      AII(I,J)=1.                                              00020670
      AII(I,J+1)=-(2.+A1*XPLYK(NPLY(1)))                   00020680
      AII(I,J+2)=1.                                           00020690
      AII(I,J+5)=-H12*X*2                                     00020700
      AII(I,J+6)=H12*X*2*(2.+A2*XPLYK(NPLY(1)+1))        00020710
      AII(I,J+7)=-H12*X*2                                     00020720
      F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*
      XBARU(NPLY(1)+1))                                       00020730
C C
      INTERFACE SHEAR ON BOTTOM PLATE
      I=NPLY(1)+6                                              00020780
      J=NPLY(1)+5                                              00020790
      AII(I,J)=-1.                                             00020800
      AII(I,J+1)=(2.+A2*XPLYK(NPLY(1)+2))                  00020810
      AII(I,J+3)=-(2.+A2*XPLYK(NPLY(1)+2))                 00020820
      AII(I,J+4)=1.                                           00020830
      F(I)=2.*XH(2)*3*(P+DELP)/FASBS                       00020840
C C
      GOVERNING EQUATIONS FOR THE BOTTOM PLATE
      N1=NPLY(1)+7                                              00020850
      N2=NPLY(1)+NPLY(2)+6                                     00020860
      DO 70 I=N1,N2                                           00020870
      J=I-2                                                 00020880
      AII(I,J)=1.                                              00020890
      IF(I.EQ.N1) GO TO 71                                     00020900
                                                00020910
                                                00020920
                                                00020930
                                                00020940
                                                00020950

```

```

AII(I,J+1,-4,-A(2)*PLYK(J-5)          00020960
GO TO 72                                00020970
71 AII(I,J+1)=-4,-A(2)*PLYK(NPLY(1)+2)  00020980
72 AII(I,J+2)=6,+2.*A(2)+B(2)*PLYK(J-4)  00020990
IF(I.EQ.N2) GO TO 75                    00021000
AII(I,J+3)=-4,-A(2)*PLYK(J-3)          00021010
GO TO 76                                00021020
73 AII(I,J+3)=-4,-A(2)*PLYK(J-5)          00021030
74 AII(I,J+4)=1,                          00021040
IF(I.EQ.N1) GO TO 73                    00021050
IF(I.EQ.N2) GO TO 77                    00021060
F(I)=A(2)*BARK(J-5)*BARU(J-5)          00021070
-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)  00021080
+A(2)*BARK(J-3)*BARU(J-3)            00021090
GO TO 74                                00021100
75 F(I)=2*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2) 00021110
-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)  00021120
GO TO 74                                00021130
76 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)        00021140
-(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)  00021150
74 CONTINUE                               00021160
70 CONTINUE                               00021170
00021180
C   SHEAR ON BOTTOM PLATE EQUALS ZERO       00021190
NP=NPLY(1)+NPLY(2)                      00021200
I=NP+7                                  00021210
J=NP+4                                  00021220
AII(I,J)=-1.                            00021230
AII(I,J+1)=(2.+A2*PLYK(NP-1))         00021240
AII(I,J+3)=-(2.+A2*PLYK(NP-1))        00021250
AII(I,J+4)=1.                          00021260
F(I)=0.                                  00021270
00021280
00021290
C   MOMENT BOUNDARY CONDITION ON BOTTOM PLATE 00021300
I=NP+8                                  00021310
IF(RF(2).OE.1.D10) GO TO 85           00021320
Z=1.                                     00021330
R=RF(2)                                00021340
GO TO 95                                00021350
85 Z=0.                                 00021360
R=1.                                     00021370
95 AII(I,J)=-R                         00021380
AII(I,J+1)=Z*(2.*H(2)*FASSS)+RH(2.+A2*PLYK(NP-1)) 00021390
-H(2)*Z*2*FASSS/FASBS                00021400
AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)*Z*3*PLYK(NP)) 00021410
AII(I,J+3)=Z*2.*H(2)*FASSS+RH(-2.-A2*PLYK(NP-1)) 00021420
-H(2)*Z*2*FASSS/FASBS                00021430
AII(I,J+4)=R                         00021440
F(I)=Z*(2.*H(2)*Z*3*BARK(NP)*BARU(NP)) 00021450
RETURN                                  00021460
END                                     00021470
00021480
00021490
00021500
00021510
00021520
00021530
00021540
00021550
C   SUBROUTINE SOLVE(U,H,P,DELP,NSDLS,ITT)

```

```

IMPLICIT REAL*8(A-H,O-Z)          00021560
DIMENSION A(100,100),B(100),NPLY(2),U(100),F(100) 00021570
DIMENSION SX(100),PLYK(100),H(2)      00021580
DIMENSION BARK(100),BARU(100)        00021590
COMMON/LYP/NPLY                   00021600
COMMON/AFM/A,F                     00021610
COMMON/PBB/PLYK,BARK,BARU         00021620
00021630
C SOLUTION OF THE SYSTEM: PAI(U)=(B) 00021640
C
NP=NPLY(1)+NPLY(2)+8             00021650
DO 444 I=1,NP                  00021660
444 B(I)=F(I)                  00021670
00021680
C
C APPLYING QUASSIAN ELIMINATION TO THE 00021700
C MATRIX OF COEFFICIENTS          00021710
00021720
C
DO 2001 I=1,NP                00021730
IR=I                           00021740
2042 IF(A(IR,I).NE.0.) GO TO 2041 00021750
IR=IR+1
IF(IR.GT.NP) GO TO 2001
GO TO 2042
2041 NN=I:I+1
CD 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.
2003 CONTINUE
2003 J=I,NP
B(L)=B(L)*CF+B(I)
2002 CONTINUE
2001 CONTINUE
C
C BACK SUBSTITUTION
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
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*X2.
SUM4=SUM4*X2.
SUM5=SUM5*X2.
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,HTH,PFAIL,ANGLE,NODE,
NIROUT,NOPT4,NULTF,JNT,ITT,NTFL)
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/PLYX,BARK,BARU
COMMON/PRT/NDAM,INPLY,ITYP
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
NP=NPLY(1)+NPLY(2)
00022160
00022170
00022180
00022190
00022200
00022210
00022220
00022230
00022240
00022250
00022260
00022270
00022280
00022290
00022300
00022310
00022320
00022330
00022340
00022350
00022360
00022370
00022380
00022390
00022400
00022410
00022420
00022430
00022440
00022450
00022460
00022470
00022480
00022490
00022500
00022510
00022520
00022530
00022540
00022550
00022560
00022570
00022580
00022590
00022600
00022610
00022620
00022630
00022640
00022650
00022660
00022670
00022680
00022690
00022700
00022710
00022720
00022730
00022740
00022750

```

```

C FAIL INCREMENTS THE POINT LOAD TO EACH SUCCESSIVE          00022760
CCC PLY AND INTERFACE FAILURE UNTIL FINAL JOINT FAILURE      00022770
CCC TAKES PLACE.                                              00022780
CCC
CCC FULL BEARING FAILURE ANALYSIS                         00022800
CCC
C IF(BPR,NE,0,0) GO TO 600                                00022810
C IROUT=1
CCC
C LOOP OVER ALL PLIES TO FIND LOAD, LOCATION, AND MODE OF 00022820
CCC NEXT PLY FAILURE.                                         00022830
CCC
C 100 IF(DELP,EQ,0,) GO TO 10                            00022840
C PFP=1,DD10
C GO TO 15
C 10 PFP=1000.
C 15 MODEF=0
C DELPF=0,DO
C NN=NPLY(1)+NPLY(2)
C DO 20 I=1,NN
CCC
C IF PLY HAS ALREADY LOST STIFFNESS, GO                  00022950
C ON TO THE NEXT PLY.                                     00022960
CCC
C IF(MDAMP(I),EQ,10) GO TO 20                           00022970
CCC
C DETERMINE WHICH PLATE THIS PLY IS IN.                   00022980
CCC
C K=1
C IF(I,GT,NPLY(1)) K=2
CCC
C CALCULATE THE LOAD ON PLY FOR CURRENT JOINT LOAD       00023000
CCC
C PL=-H(K)*(PLYK(I)*U(I)+BARK(I)*BARU(I))            00023010
CCC
C ASSUME FAILURE OCCURS ONLY ON BEARING SIDE.           00023020
CCC
C
C IF(PL,LT,0,,AND,K,EQ,1) GO TO 20                      00023030
C IF(PL,GT,0,,AND,K,EQ,2) GO TO 20                      00023040
CCC
C DETERMINE PLY LOAD NECESSARY TO CAUSE NEXT             00023050
C FAILURE AND ITS MODE.                                 00023060
CCC
C IF(NOPT4,NE,1,AND,NOPT4,NE,3) GO TO 200              00023070
C MODE=8
C IN=I-(K-1)*NPLY(1)
C NPY=IPLY(IN,K)
C PF=PFL(NPY,K)
CCC
C IF PL>PF AT CURRENT JOINT LOAD                      00023080
C PREDICT FAILURE.                                       00023090
CCC
C NCC=0
C IF(DELP,NE,0,) GO TO 210
C IF(DABS(PL),LT,DABS(PF)) GO TO 210
C PFP=0
C INPLY=I

```

```

      MODEF=MOU_
      NCC=1
      GO TO 140
140  NMN=I
      IF(I.GT.NPLY(1)) NMN=I-NPLY(1)
      NX=IPLY(NMN,K)
      IF(PBR(NX,K).LT.PSD(NX,K).OR.PNS(NX,K).LT.PSD(NX,K)) GO TO 700
      MODE=1
      PF=PSD(NX,K)
      IF(MDAMP(I).EQ.1) MODE=5
      IF(MDAMP(I).EQ.1) PF=PALT(1,K)*PF
      GO TO 25
145  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
150  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
      " DEF=MODE
      NCC=1
      GO TO 1212
210  CONTINUE
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 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 RECORD LOWEST JOINT FAILURE LOAD INCREMENT,
C PLY IN WHICH IT OCCURS, AND MODE
C
      PPP2=PFP
      IF(DELP.EQ.0) PPP2=1.
      IF(DELPF.GT.PFP2) GO TO 20
      PFP=DELPF
      INPLY=I
      MODEF=MODE
20  CONTINUE
C LOOP OVER ALL INTERFACES TO FIND LOAD
C AND LOCATION OF NEXT DELAMINATION
      00023360
      00023370
      00023380
      00023390
      00023400
      00023410
      00023420
      00023430
      00023440
      00023450
      00023460
      00023470
      00023480
      00023490
      00023500
      00023510
      00023520
      00023530
      00023540
      00023550
      00023560
      00023570
      00023580
      00023590
      00023600
      00023610
      00023620
      00023630
      00023640
      00023650
      00023660
      00023670
      00023680
      00023690
      00023700
      00023710
      00023720
      00023730
      00023740
      00023750
      00023760
      00023770
      00023780
      00023790
      00023800
      00023810
      00023820
      00023830
      00023840
      00023850
      00023860
      00023870
      00023880
      00023890
      00023900
      00023910
      00023920
      00023930
      00023940
      00023950

```

```

NN=NPLY(1)+NPLY(2)-2          00023960
DO 50 J=1,NN                   00023970
C
C   IF INTERFACE HAS ALREADY FAILED, GO TO NEXT      00023980
C   IF(MDAMI(J).EQ.1) GO TO 50                         00023990
C
C   DETERMINE WHICH PLATE INTERFACE IS IN             00024000
C
C   K=1                                                 00024010
C   IF(J.GE.NPLY(1)) K=2                               00024020
C
C   CALCULATE INTERFACE SHEAR STRAIN FOR CURRENT     00024030
C   JOINT LOAD                                         00024040
C
C   GAMJ=(U(J+K-1)-U(J+K))/H(K)                      00024050
C
C   DETERMINE INCREMENTAL JOINT LOAD TO CAUSE        00024060
C   INTERFACE FAILURE                                  00024070
C
C   IF(ITT.EQ.1) GO TO 47                             00024080
C   IF(DABS(DABS(GAMJ/GAMH(J))-1.).LT.1.0D-10) GO TO 50 00024090
47 CONTINUE
DELPF=(GAMDL(K)-DABS(GAMH(J)))/(DABS(GAMJ)-DABS(GAMH(J)))X1000. 00024100
IF(DELPF.LT.0.) GO TO 50                           00024110
C
C
C   RECORD LOWEST JOINT FAILURE LOAD INCREMENT,       00024120
C   PLY OF INTERFACE IN WHICH IT OCCURS, AND MODE    00024130
C
C   PPP2=PPP                                         00024140
C   IF(DELPF.EQ.0) PPP2=1.                            00024150
C   IF(DELPF.GT.PPP2) GO TO 50                        00024160
C   PPP=DELPF                                         00024170
C   INPLY=J                                           00024180
C   MODEF=9                                           00024190
50 CONTINUE
C
C   DETERMINE VALUES AT END_OF_INCREMENT              00024200
C
C   JOINT LOAD AT FAILURE                            00024210
C
C   IF(MODEF.EQ.0) GO TO 325.                         00024220
C   P=P+PFPKDELP/1000.                             00024230
325 CONTINUE
C
C   NODAL DISPLACEMENTS AND PLY LOADS               00024240
C
C   NN=NPLY(1)+NPLY(2)                               00024250
C   DO 55 I=1,NN                                     00024260
C   UN(I)=UN(I)+(U(I)-UN(I))XPPP/1000.            00024270
C
C   UPDATE UN                                         00024280
C
C   IF(NCC.EQ.1) UN(I)=U(I)                          00024290
C   K=1                                               00024300
C   IF(I.GT.NPLY(1)) K=2                            00024310
C   PN(I)=-H(K)X(PLYK(I)XUN(I)+BARK(I)XBARUC(I)) 00024320
55 CONTINUE

```

```

C INTERFACE SHEAR STRAINS 00024560
C
NN=NPLY(1)+NPLY(2)-2 00024570
DG 60 J=1,NN 00024580
K=1 00024590
IF(J.GE.NPLY(1)) K=2 00024600
GAMN(J)=(UN(J+K-1)-UN(J+K))/H(K) 00024610
60 CONTINUE 00024620
1212 CONTINUE 00024630
C PLY STIFFNESSES, DAMAGE STATES, AND NEXT LOAD 00024640
C INCREMENT 00024650
C
K=1 00024660
IF(INPLY.GT.NPLY(1)) K=2 00024670
NMN=INPLY 00024680
IF(INPLY.GT.NPLY(1)) NMN=INPLY-NPLY(1) 00024690
NX=IPLY(NMN,K) 00024700
IF(MODEF.NE.0) GO TO 70 00024710
DELP=1000. 00024720
NDAM=1 00024730
GO TO 65 00024740
70 IF(MODEF.NE.1) GO TO 80 00024750
IF(INPLY.EQ.1.AND.MDAMI(INPLY).EQ.1) GO TO 75 00024760
IF(INPLY.EQ.NPLY(1).AND.MDAMI(INPLY-1).EQ.1) GO TO 75 00024770
IF(INPLY.EQ.(NPLY(1)+1).AND.MDAMI(INPLY-1).EQ.1) GO TO 73 00024780
IF(INPLY.EQ.(NPLY(1)+NPLY(2)).AND.MDAMI(NPLY(1)+NPLY(2)
1-2).EQ.1) GO TO 75 00024790
KK=0 00024800
IF(INPLY.GT.NPLY(1)) KK=1 00024810
IF(MDAMI(INPLY-KK-1).EQ.1.AND.MDAMI(INPLY-KK).EQ.1) GO TO 75 00024820
MDAMP(INPLY)=1 00024830
TEMPK=PLYK(INPLY) 00024840
PLYK(INPLY)=DEL50(NX,K)*PLYK(INPLY) 00024850
BARK(INPLY)=(1.-DEL50(NX,K))*TEMPK 00024860
BARU(INPLY)=UN(INPLY) 00024870
DELP=0. 00024880
NDAM=2 00024890
GO TO 65 00024900
75 PLYK(INPLY)=0.0 00024910
BARK(INPLY)=0. 00024920
BARU(INPLY)=UN(INPLY) 00024930
MDAMP(INPLY)=10 00024940
DELP=0. 00024950
NDAM=2 00024960
GO TO 65 00024970
80 IF(MODEF.NE.2) GO TO 85 00024980
TEMPK=PLYK(INPLY) 00024990
PLYK(INPLY)=DELNS(NX,K)*TEMPK 00025000
BARK(INPLY)=(1.-DELNS(NX,K))*TEMPK 00025010
BARU(INPLY)=UN(INPLY) 00025020
MDAMP(INPLY)=2 00025030
DELP=0. 00025040
NDAM=4 00025050
GO TO 65 00025060
85 IF(MODEF.NE.3) GO TO 90 00025070
TEMPK=PLYK(INPLY) 00025080
PLYK(INPLY)=DELR(NX,K)*TEMPK 00025090
BARK(INPLY)=(1.-DELR(NX,K))*TEMPK 00025100
BARU(INPLY)=UN(INPLY) 00025110

```

```

MDAMP(INPLY)=3          00025160
DELP=0.                  00025170
NDAM=5                  00025180
GO TO 65                00025190
90 IF(MODEF.NE.4) GO TO 140 00025200
PLYK(INPLY)=0.0          00025210
BARK(INPLY)=0.            00025220
BARU(INPLY)=UN(INPLY)    00025230
MDAMP(INPLY)=10          00025240
DELP=0.                  00025250
NDAM=6                  00025260
GO TO 65                00025270
140 IF(MODEF.NE.5) GO TO 110 00025280
PLYK(INPLY)=0.           00025290
BARK(INPLY)=0.           00025300
BARU(INPLY)=UN(INPLY)    00025310
MDAMP(INPLY)=10          00025320
DELP=0.                  00025330
NDAM=6                  00025340
GO TO 65                00025350
110 IF(MODEF.NE.6) GO TO 115 00025360
PLYK(INPLY)=0.0          00025370
BARK(INPLY)=0.0          00025380
BARU(INPLY)=UN(INPLY)    00025390
MDAMP(INPLY)=10          00025400
NDAM=7                  00025410
DELP=0.                  00025420
GO TO 65                00025430
115 IF(MODEF.NE.7) GO TO 120 00025440
PLYK(INPLY)=0.           00025450
BARK(INPLY)=0.           00025460
BARU(INPLY)=UN(INPLY)    00025470
MDAMP(INPLY)=10          00025480
DELP=0.                  00025490
NDAM=8                  00025500
GO TO 65                00025510
120 IF(MODEF.NE.8) GO TO 125 00025520
PLYK(INPLY)=0.           00025530
BARK(INPLY)=0.           00025540
BARU(INPLY)=UN(INPLY)    00025550
MDAMP(INPLY)=10          00025560
NDAM=9                  00025570
DELP=0.                  00025580
GO TO 65                00025590
125 IF(MODEF.NE.9) GO TO 65 00025600
DELP=1000.                00025610
MDAMC(INPLY)=1            00025620
NDAM=10                 00025630
GO TO 65                00025640
600 CONTINUE              00025650
00025660
C C PARTIAL BEARING FAILURE ANALYSIS 00025670
00025680
IROUT=2                  00025690
NPL=NPLY(1)+NPLY(2)      00025700
AJFLNS=1.0D10             00025710
AJFLBR=1.0D10             00025720
AJFLSO=1.0D10             00025730
DO 550 I=1,NPL           00025740
K=1                      00025750

```

```

IF(I.GT.N.LY(1)) K=2 00025760
PL=-H(K)*(PLYK(I)*U(I)+BARK(I)*BARU(I)) 00025770
IF(PL.LT.0.0.AND.K.EQ.1) GO TO 550 00025780
IF(PL.GT.0.0.AND.K.EQ.2) GO TO 550 00025790
PL=DABS(PL)*ASTMTHHH(K)*NPLY(K)/1000. 00025800
IJ=I 00025810
IF(I.GT.NPLY(1)).IJ=IJ-NPLY(1). 00025820
IP=IPLY(IJ,K) 00025830
NT=1 00025840
IF(BPSTS(K,IP,2,1).LT.0.0) NT=2 00025850
F1=DABS(PSTC(NT,IP,K)/(BPSTS(K,IP,1,1)+PL*BPSTS(K,IP,2,1))) 00025860
F2=DABS(PSTC(3,IP,K)/(BPSTS(K,IP,1,2)+PL*BPSTS(K,IP,2,2))) 00025870
F3=DABS(PSTC(4,IP,K)/(BPSTS(K,IP,1,3)+PL*BPSTS(K,IP,2,3))) 00025880
IF(MDAMP(I).EQ.3) F1=PALT(3,K)*F1 00025890
IF(MDAMP(I).EQ.4) F2=PALT(2,K)*F2 00025900
IF(MDAMP(I).EQ.2) F3=PALT(1,K)*F3 00025910
IF(AJFLNS.GT.F1) NF1=I 00025920
IF(AJFLNS.GT.F1) AJFLNS=F1 00025930
IF(AJFLBR.GT.F2) NF2=I 00025940
IF(AJFLBR.GT.F2) AJFLBR=F2 00025950
IF(AJFLSO.GT.F3) NF3=I 00025960
IF(AJFLSO.GT.F3) AJFLSO=F3 00025970
550 CONTINUE 00025980
IF(AJFLNS.GT.AJFLBR.OR.AJFLNS.GT.AJFLSO) GO TO 560 00025990
INPLY=NF1 00026000
551 NDAM=5 00026010
IF(MDAMP(INPLY).EQ.2) GO TO 571 00026020
IF(MDAMP(INPLY).EQ.4) GO TO 561 00026030
IF(MDAMP(INPLY).EQ.5) NDAM=8 00026040
MDAMP(INPLY)=NDAM 00026050
PFAIL=AJFLNS 00026060
GO TO 64 00026070
560 IF(AJFLBR.GT.AJFLNS.OR.AJFLBR.GT.AJFLSO) GO TO 570 00026080
INPLY=NF2 00026090
561 NDAM=4 00026100
IF(MDAMP(INPLY).EQ.5) GO TO 551 00026110
IF(MDAMP(INPLY).EQ.2) GO TO 571 00026120
IF(MDAMP(INPLY).EQ.4) NDAM=7 00026130
MDAMP(INPLY)=NDAM 00026140
PFAIL=AJFLBR 00026150
GO TO 64 00026160
570 IF(AJFLSO.GT.AJFLNS.OR.AJFLSO.GT.AJFLBR) GO TO 64 00026170
INPLY=NF3 00026180
571 NDAM=2 00026190
IF(MDAMP(INPLY).EQ.5) GO TO 551 00026200
IF(MDAMP(INPLY).EQ.4) GO TO 561 00026210
IF(MDAMP(INPLY).EQ.2) NDAM=6 00026220
MDAMP(INPLY)=NDAM 00026230
PFAIL=AJFLSO 00026240
64 CONTINUE 00026250
K=1 00026260
IF(INPLY.GT.NPLY(1)) K=2 00026270
IPL=INPLY 00026280
IF(IPL.GT.NPLY(1)) IPL=IPL-NPLY(1) 00026290
IPLP=IPLY(IPL,K) 00026300
ANGLE=ANG(IPLP,K) 00026310
NODE=NPNM(IPL,K) 00026320
IF(MDAMP(INPLY).GE.6) GO TO 107 00026330
IF(NDAMP(INPLY).EQ.5) AR=DELNS(IPLP,K) 00026340
IF(MDAMP(INPLY).EQ.4) AR=DELBR(IPLP,K) 00026350

```

```

IF(MDAMP(.,NPLY).EQ.2) AR=DELSOC(IPLP,K)          00026360
TEMPK=PLYK(INPLY)                                00026370
PLYK(INPLY)=ARXTEMPK                            00026380
BARK(INPLY)=(1.-AR)XTEMPK                         00026390
BARU(INPLY)=U(INPLY)                             00026400
ITYP=IPLY(IPL,K)                                00026410
NTFL=0                                         00026420
GO TO 103                                     00026430
107 CONTINUE
IF(K.EQ.1) NPLY(1)=NPLY(1)-1                   00026440
IF(K.EQ.2) NPLY(2)=NPLY(2)-1                   00026450
NP=INPLY
IF(K.EQ.2) NP=INPLY-NPLY(1)                   00026460
N=NPLY(K)-NP+2                                00026470
IFYP=IPLY(NP,K)                                00026480
DO 101 I=1,N
IPLY(NP+I-1,K)=IPLY(NP+I,K)                  00026490
NPNM(NP+I-1,K)=NPNM(NP+I,X)                  00026500
101 CONTINUE
N=NPL-INPLY                                00026510
DO 102 I=1,N
MDAMP(INPLY+I-1)=MDAMP(INPLY+I)              00026520
PLYK(INPLY+I-1)=PLYK(INPLY+I)                00026530
BARK(INPLY+I-1)=BARK(INPLY+I)                00026540
102 BARU(INPLY+I-1)=ARU(INPLY+I)              00026550
NTFL=1                                         00026560
NULTF=NULTF-1                                00026570
IF(NULTF.EQ.0) JNT=0                           00026580
IF(NPLY(1).EQ.2.OR.NPLY(2).EQ.2) JNT=0       00026590
103 CONTINUE
RETURN                                         00026600
63 CONTINUE
CCC INCREMENT LOAD IF JOINT HAS NOT FAILED      00026610
T1=0.                                         00026620
T2=0.                                         00026630
N1=NPLY(1)                                    00026640
N2=NPLY(2)                                    00026650
DO 135 I=1,N1                                00026660
135 T1=T1+PLYK(I)
DO 126 I=1,N2                                00026670
N3=NPLY(1)+I                                 00026680
126 T2=T2+PLYK(N3)
IF(T1.EQ.0.0.DR.T2.EQ.0.0) GO TO 130      00026690
RETURN                                         00026700
130 JNT=0
RETURN                                         00026710
END                                           00026720
CCC
SUBROUTINE PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT,
NPL,NSDLs,ITY)                               00026730
IMPLICIT REAL*8(A-H,O-Z)                      00026740
DIMENSION U(100),PLYK(100)                    00026750
DIMENSION NPLY(2),NUMPLY(2),ANG(5,2),IPLY(100,2) 00026760
DIMENSION BARK(100),BARU(100)                 00026770

```

```

DIMENSION NPNM(100,2) 00026960
COMMON/COUNT/NPNM 00026970
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY 00026980
COMMON/PBB/PLYK,DARK,BARU 00026990
COMMON/PRT/NDAM,INPLY,ITYP 00027000
C C PRINT VALUES AT END OF INCREMENT 00027010
IF(ITT.EQ.1) WRITE(6,10) 00027020
10 FORMAT(//,10X,'FAILURE MODE ABBREVIATIONS') 00027030
*10X,'ND' = NO ADDITIONAL DAMAGE AT CURRENT JOINT LOAD 00027040
*10X,'DL' = DELAMINATION 00027050
*10X,'SO' = SHEAR-OUT 00027060
*10X,'BR' = BEARING 00027070
*10X,'NS' = NET SECTION 00027080
*10X,'SUD' = ULTIMATE FAILURE AFTER SO AND DL 00027090
*10X,'SU' = ULTIMATE FAILURE IN SO 00027100
*10X,'BU' = ULTIMATE FAILURE IN BR 00027110
*10X,'NSU' = ULTIMATE FAILURE IN NS 00027120
*10X,'ULT' = ULTIMATE FAILURE 00027130
*4X,'INCREMENT NO',3X,'JOINT LOAD',5X,'NODE',8X,'PLY TYPE', 00027140
*8X,'MODE',//,/) 00027150
PL=P 00027160
IF(IROUT.EQ.2) PL=PFAIL 00027170
IF(NSDLS.EQ.2) PL=2.*PL 00027180
IF(ITT.EQ.1) PFAILP=0.000 00027190
IF(PFAILP.LT.PL) PFAILP=PL 00027200
IF(JNT.EQ.0.AND.PFAILP.EQ.0.000) PFAILP=PL 00027210
K=1 00027220
IF(INPLY.GT.NPLY(1)) K=2 00027230
N=IPLY(INPLY,K) 00027240
IF(K.EQ.2) N=IPLY((INPLY-NPLY(1)),K) 00027250
IF(IROUT.EQ.2) N=ITYP 00027260
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) ANGLE=ANG(N,K) 00027270
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) NODE=INPLY 00027280
IF(NDAM.EQ.1) WRITE(6,20) ITT,PL 00027290
IF(NDAM.EQ.2) WRITE(6,30) ITT,PL,NODE,ANGLE 00027300
IF(NDAM.EQ.3) WRITE(6,40) ITT,PL,NODE,ANGLE 00027310
IF(NDAM.EQ.4) WRITE(6,50) ITT,PL,NODE,ANGLE 00027320
IF(NDAM.EQ.5) WRITE(6,60) ITT,PL,NODE,ANGLE 00027330
IF(NDAM.EQ.6) WRITE(6,70) ITT,PL,NODE,ANGLE 00027340
IF(NDAM.EQ.7) WRITE(6,80) ITT,PL,NODE,ANGLE 00027350
IF(NDAM.EQ.8) WRITE(6,90) ITT,PL,NODE,ANGLE 00027360
IF(NDAM.EQ.9) WRITE(6,100) ITT,PL,NODE,ANGLE 00027370
IF(NDAM.EQ.10) WRITE(6,110) ITT,PL,NCDE,ANGLE 00027380
20 FORMAT(5X,I5,10X,D9.3,34X,'ND') 00027390
30 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', SO') 00027400
40 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', SUD') 00027410
50 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', BR') 00027420
60 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', NS') 00027430
70 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', SU') 00027440
80 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', BU') 00027450
90 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', NSU') 00027460
100 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', ULT') 00027470
110 FORMAT(5X,I5,10X,D9.3,3X,I5,5X,F7.3,', DEGREE ',5X,', DL') 00027480
IF(JNT.EQ.0) GO TO 220 00027490
GO TO 250 00027500
220 CONTINUE 00027510
WRITE(6,240) PFAILP 00027520
240 FORMAT(//,' THE PREDICTED JOINT FAILURE',//,/) 00027530
00027540
00027550

```

```

      *' LOAD IS ',D14.7,' LBS',//)
250 CONTINUE
      RETURN
      END
C
C     SUBROUTINE LINV2F (A,N,IA,AINV,IDLGT,WKAREA,IER)          00027560
C
      DOUBLE PRECISION A(IA,N),AINV(IA,N),WKAREA(1),ZERO,ONE        00027570
      DATA             ONE/1.000/,ZERO/0.000/                         00027580
C
C           FIRST EXECUTABLE STATEMENT                            00027590
C           INITIALIZE IER                                         00027600
C
      IER=0
C           SET AINV TO THE N X N                                00027610
C           IDENTITY MATRIX                                     00027620
C
      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                           00027630
      CALL LEQT2F (A,N,N,IA,AINV,IDLGT,WKAREA,IER)            00027640
      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,IDLGT,WKAREA,IER)          00027650
C
      DIMENSION A(IA,1),B(IA,1),WKAREA(1)                      00027660
      DOUBLE PRECISION A,B,WKAREA,D1,D2,WA                     00027670
C
C           FIRST EXECUTABLE STATEMENT                            00027680
C           INITIALIZE IER                                         00027690
C
      IER=0
      JER=0
      J = NMN+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                                         00027700
      CALL LUDATN (WKAREA,N,N,A,IA,IDLGT,D1,D2,WKAREA(J),WKAREAK,
      *                  WA,IER)
      IF (IER.GT.128) GO TO 25
      IF (IDLGT .EQ. 0 .OR. IER .NE. 0) KK = 1
      DO 15 I = 1,M
C           PERFORMS THE ELIMINATION PART OF                 00027710
C           AX = B                                           00027720
      CALL LUELMN (A,IA,N,B(1,1),WKAREAK,1),WKAREA(MM))
C           REFINEMENT OF SOLUTION TO AX = B                 00027730
      15 CONTINUE
      END

```

```

      IF (KK .NE. 0)          00028160
      X CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDGT,WKAREA(J),WKAREA(MM), 00028170
      X WKAREA(K),WKAREA(K),JER). 00028180
      DO 10 II=1,N            00028190
      X B(KK,I) = WKAREA(MM1+II) 00028200
10   CONTINUE               00028210
      IF (JER.NE.0) GO TO 20  00028220
13   CONTINUE               00028230
      GO TO 25               00028240
20   IER = 131              00028250
25   JJ=1                  00028260
      DO 30 J = 1,N          00028270
      DO 30 I = 1,N          00028280
      X A(I,J)=WKAREA(JJ)  00028290
      X JJ=JJ+1              00028300
      30 CONTINUE              00028310
      IF (IER .EQ. 0) GO TO 9005 00028320
9000 CONTINUE               00028330
      CALL UERTST (IER,6HLEQT2F) 00028340
9005 RETURN                00028350
      END                     00028360
C
C
C     SUBROUTINE LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)
C
C     DIMENSION      A(IA,1),LU(IA,1),IPVT(1),EQUIL(1) 00028400
C     DOUBLE PRECISION A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTH,SIXTH, 00028410
      X RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q 00028420
      X DATA ZERO,ONE,FOUR,SIXTH,SIXTH/0.D0,1.D0,4.D0, 00028430
      X 16.D0,.0625D0/ 00028440
C
C           FIRST EXECUTABLE STATEMENT 00028450
C           INITIALIZATION 00028460
      IER = 0                 00028470
      RN = N                 00028480
      WREL = ZERO             00028490
      D1 = ONE                00028500
      D2 = ZERO                00028510
      BIGA = ZERO             00028520
      DO 10 I=1,N             00028530
      X BIG = ZERO             00028540
      DO 5 J=1,N              00028550
      X P = A(I,J)             00028560
      X LU(I,J) = P            00028570
      X P = DABS(P)            00028580
      X IF (P .GT. BIG) BIG = P 00028590
      5  CONTINUE               00028600
      X IF (BIG .GT. BIGA) BIGA = BIG 00028610
      X IF (BIG .EQ. ZERO) GO TO 110 00028620
      X EQUIL(I) = ONE/BIG 00028630
10   CONTINUE               00028640
      105 J=1,N                00028650
      X JM1 = J-1              00028660
      X IF (JM1 .LT. 1) GO TO 40 00028670
C           COMPUTE U(I,J), I=1,...,J-1 00028680
      DO 35 I=1,JM1            00028690
      X SUM = LU(I,J)            00028700
      X IM1 = I-1              00028710
      X IF (IDGT .EQ. 0) GO TO 25 00028720
C           WITH ACCURACY TEST 00028730
      C                         00028740
      C                         00028750

```

```

AI = DABS(SUM)          00028760
WI = ZERO                00028770
IF (IM1 .LT. 1) GO TO 20  00028780
DO 15 K=1,IM1            00028790
   T = LU(I,K)*LU(K,J)
   SUM = SUM-T
   WI = WI+DABS(T)
15  CONTINUE              00028820
   LU(I,J) = SUM          00028830
20  WI = WI+DABS(SUM)    00028840
   IF (AI .EQ. ZERO) AI = BIGA  00028850
   TEST = WI/AI           00028860
   IF (TEST .GT. WREL) WREL = TEST  00028870
   GO TO 35               00028880
C   WITHOUT ACCURACY      00028890
25  IF (IM1 .LT. 1) GO TO 35  00028900
   DO 30 K=1,IM1          00028910
   SUM = SUM-LU(I,K)*LU(K,J)
30  CONTINUE              00028920
   LU(I,J) = SUM          00028930
35  CONTINUE              00028940
40  P = ZERO               00028950
C   COMPUTE U(J,J) AND L(I,J), I=J+1,...,N  00028960
   DO 70 I=J,N             00028970
   SUM = LU(I,J)
   IF (IDOT .EQ. 0) GO TO 55  00028980
C   WITH ACCURACY TEST    00028990
   AI = DABS(SUM)
   WI = ZERO
   IF (JM1 .LT. 1) GO TO 50  00029000
   DO 45 K=1,JM1          00029010
   T = LU(I,K)*LU(K,J)
   SUM = SUM-T
   WI = WI+DABS(T)
45  CONTINUE              00029020
   LU(I,J) = SUM          00029030
50  WI = WI+DABS(SUM)    00029040
   IF (AI .EQ. ZERO) AI = BIGA  00029050
   TEST = WI/AI           00029060
   IF (TEST .GT. WREL) WREL = TEST  00029070
   GO TO 65               00029080
C   WITHOUT ACCURACY TEST  00029090
55  IF (JM1 .LT. 1) GO TO 65  00029100
   DO 60 K=1,JM1          00029110
   SUM = SUM-LU(I,K)*LU(K,J)
60  CONTINUE              00029120
   LU(I,J) = SUM          00029130
65  Q = EQUIL(I)*DABS(SUM)  00029140
   IF (P .GE. Q) GO TO 70  00029150
   P = Q
   IMAX = I
70  CONTINUE              00029160
C   TEST FOR ALGORITHMIC SINGULARITY  00029170
   IF (RN+P .EQ. RN) GO TO 110  00029180
   IF (J .EQ. IMAX) GO TO 80  00029190
C   INTERCHANGE ROWS J AND IMAX  00029200
   D1 = -D1
   DO 75 K=1,N             00029210
   P = LU(IMAX,K)
   LU(IMAX,K) = LU(J,K)  00029220
   LU(J,K) = P             00029230
   P = Q
   IMAX = I
75  CONTINUE              00029240
C   00029250
   IMAX = I
70  CONTINUE              00029260
C   TEST FOR ALGORITHMIC SINGULARITY  00029270
   IF (RN+P .EQ. RN) GO TO 110  00029280
   IF (J .EQ. IMAX) GO TO 80  00029290
C   INTERCHANGE ROWS J AND IMAX  00029300
   D1 = -D1
   DO 75 K=1,N             00029310
   P = LU(IMAX,K)
   LU(IMAX,K) = LU(J,K)  00029320
   LU(J,K) = P             00029330
   P = Q
   IMAX = I
75  CONTINUE              00029340
C   00029350

```

```

    LUL(J,K) = P          00029360
75   CONTINUE           00029370
     EQUIL(IMAX) = EQUIL(J) ... 00029380
80   IPVT(J) = IMAX      00029390
     D1 = D1*LU(J,J)       00029400
85   IF (DABS(D1) .LE. ONE) GO TO 90 00029410
     D1 = D1*SIXTH...     00029420
     D2 = D2+FOUR        00029430
     GO TO 85            00029440
90   IF (DABS(D1) .GE. SIXTH) GO TO 95 00029450
     D1 = D1*SIXTH...     00029460
     D2 = D2-FOUR        00029470
     GO TO 90            00029480
95   CONTINUE           00029490
     JP1 = J+1           00029500
     IF (JP1 .GT. N) GO TO 105 00029510
C          DIVIDE BY PIVOT ELEMENT U(J,J) 00029520
     P = LU(J,J)          00029530
     DO 100 I=JP1,N       00029540
     LU(I,J) = LU(I,J)/P  00029550
100  CONTINUE           00029560
105  CONTINUE           00029570
C          PERFORM ACCURACY TEST 00029580
     IF (IDOT .EQ. 0) GO TO 9005 00029590
     P = 3*N+3             00029600
     WA = P*HREL           00029610
     IF (WA+10.D0*N(-IDOT)...NE. WA) GO TO 9005 00029620
     IER = 34              00029630
     GO TO 9000            00029640
C          ALGORITHMIC SINGULARITY 00029650
110  IER = 129           00029660
     D1 = ZERO             00029670
     D2 = ZERO             00029680
9000  CONTINUE           00029690
C          PRINT_ERROR 00029700
     CALL UERTST(IER,6HLUDATF) 00029710
9005  RETURN             00029720
     END                  00029730
C
C          SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
C          DIMENSION A(IA,1),B(1),APVT(1),X(1) 00029750
C          DOUBLE PRECISION A,B,X,SUM,APVT 00029760
C          FIRST EXECUTABLE STATEMENT 00029770
C          SOLVE LY = B FOR Y 00029780
C
     DO 5 I=1,N           00029790
5 X(I) = B(I)           00029800
     IW = 0                00029810
     DO 20 I=1,N           00029820
       IP = APVT(I)         00029830
       SUM = X(IP)          00029840
       X(IP) = X(I)          00029850
       IF (IW .EQ. 0) GO TO 15 00029860
       IM1 = I-1             00029870
       DO 10 J=IW,IM1        00029880
         SUM = SUM-A(I,J)*X(J) 00029890
10    CONTINUE           00029900
       GO TO 20             00029910
15    IF (SUM .NE. 0.00) IW = 1 00029920
                                         00029930
                                         00029940
                                         00029950

```

```

C 20 X(I) = SUM           00029J63
      DO 30 I=1,N           SOLVE UX = Y FOR X 00029970
        I = N+1-I
        IP1 = I+1
        SUM = X(I)
        IF (IP1 .GT. N) GO TO 30 00029980
        DO 25 J=IP1,N          00030000
          SUM = SUM-A(I,J)*X(J)
25  CONTINUE               00030010
30  X(I) = SUM/A(I,I)      00030020
      RETURN                 00030030
      END                     00030040
                               00030050
                               00030060
                               00030070
                               00030080
                               00030090
                               00030100
C C SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDOT,APVT,X,RES,DX,IER) 00030110
C DIMENSION A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1) 00030120
C DIMENSION APVT(1) 00030130
C DIMENSION ACCXT(2) 00030140
C DOUBLE PRECISION A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT 00030150
C DATA ITMAX/75/,ZERO/0.0D0/ 00030160
C                         FIRST EXECUTABLE STATEMENT 00030170
C
C IER=0 00030180
C XNORM = ZERO 00030190
C DO 10 I=1,N 00030200
C   XNORM = DMAX1(XNORM,DABS(X(I))) 00030210
10  CONTINUE 00030220
C IF (XNORM .NE. ZERO) GO TO 20 00030230
C IDOT = 50 00030240
C GO TO 9005 00030250
C 20 DO 45 ITER=1,ITMAX 00030260
C   DO 30 I=1,N 00030270
C     ACCXT(1) = 0.0D0 00030280
C     ACCXT(2) = 0.0D0 00030290
C     CALL VXADD(B(I),ACCXT) 00030300
C     DO 25 J=1,N 00030310
C       CALL VXMUL(-A(I,J),X(J),ACCXT) 00030320
25  CONTINUE 00030330
C     CALL VXSTO(ACCXT,RES(I)) 00030340
30  CONTINUE 00030350
C     CALL LUELMN (UL,IUL,N,RES,APVT,DX) 00030360
C     DXNORM = ZERO 00030370
C     XNORM = ZERO 00030380
C     DO 35 I=1,N 00030390
C       X(I) = X(I) + DX(I) 00030400
C       DXNORM = DMAX1(DXNORM,DABS(DX(I))) 00030410
C       XNORM = DMAX1(XNORM,DABS(X(I))) 00030420
35  CONTINUE 00030430
C     IF (ITER .NE. 1) GO TO 40 00030440
C     IDOT = 50 00030450
C     IF (DXNORM .NE. ZFRO) IDOT = -DLOG10(DXNORM/XNORM) 00030460
40  IF (XNORM+DXNORM .EQ. XNORM) GO TO 9005 00030470
45 CONTINUE 00030480
C
C IER = 129 00030490
9000 CONTINUE 00030500
C     CALL UERTST(IER,6HLUREFN) 00030510
9005 RETURN 00030520
C     END 00030530
                               00030540
                               00030550

```

```

C          00030560
CCC          00030570
C          00030580
C          SUBROUTINE UERTST (IER,NAME)          SPECIFICATIONS FOR ARGUMENTS 00030590
C          INTEGER           IER          00030600
C          INTEGER           NAME(1)        00030610
C          INTEGER           I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(6), 00030620
C          DATA             NAMSET(6),NAMUPK(6),NIN,NMT$        00030630
C          DATA             NAMSET/1H,U,1HE,1HR,1HS,1HE,1HT/      00030640
C          DATA             NAMEQ/6X1H/        00030650
C          DATA             LEVEL/4/,IEQDF/0/,IEQ/1H=/      00030660
C          DATA             UNPACK NAME INTO NAMUPK        00030670
C          FIRST EXECUTABLE STATEMENT        00030680
C          CALL USPKD (NAME,6,NAMUPK,NMT$)        00030690
C          GET OUTPUT UNIT NUMBER        00030700
C          CALL UGETIO(1,NIN,IOUNIT)        00030710
C          CHECK IER        00030720
C          IF (IER.GT.999) GO TO 25        00030730
C          IF (IER.LT.-32) GO TO 55        00030740
C          IF (IER.LE.128) GO TO 5        00030750
C          IF (LEVEL.LT.1) GO TO 30        00030760
C          PRINT TERMINAL MESSAGE        00030770
C          IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK 00030780
C          IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK        00030790
C          GO TO 30        00030800
5 IF (IER.LE.64) GO TO 10        00030810
IF (LEVEL.LT.2) GO TO 30        00030820
C          PRINT WARNING WITH FIX MESSAGE        00030830
C          IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK 00030840
C          IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK        00030850
C          GO TO 30        00030860
10 IF (IER.LE.32) GO TO 15        00030870
C          PRINT WARNING MESSAGE        00030880
C          IF (LEVEL.LT.3) GO TO 30        00030890
C          IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK 00030900
C          IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK        00030910
C          GO TO 30        00030920
15 CONTINUE          CHECK FOR UERSET CALL        00030930
C          DO 20 I=1,6        00030940
C          IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25        00030950
20 CONTINUE          00030960
C          LEVOLD = LEVEL        00030970
C          LEVEL = IER        00030980
C          IEQ = LEVOLD        00030990
C          IF (LEVEL.LT.0) LEVEL = 4        00031000
C          IF (LEVEL.GT.4) LEVEL = 4        00031010
C          GO TO 30        00031020
25 CONTINUE          00031030
C          IF (LEVEL.LT.4) GO TO 30        00031040
C          PRINT NON-DEFINED MESSAGE        00031050
C          IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK 00031060
C          IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK        00031070
30 IEQDF = 0        00031080
RETURN          00031090
35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3, 00031100
1 20H) FROM IMSL RCUITINE ,6A1,A1,6A1)        00031110
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3, 00031120
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3, 00031130
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3, 00031140
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3, 00031150

```

```

      1      2U,,) FROM IMSL ROUTINE ,6A1,A1,6A1)          00031160
45 FORMAT(18H *** WARNING ERROR,1I1,7H(IER = ,I3,        00031170
      1      20H) FROM IMSL ROUTINE ,6A1,A1,6A1)          00031180
50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I3,        00031190
      1      20H) FROM IMSL ROUTINE ,6A1,A1,6A1)          00031200
      1      20H) FROM IMSL ROUTINE ,6A1,A1,6A1)          00031210
C           SAVE P FOR P = R CASE                      00031220
C           P IS THE PAGE NAMUPK                      00031230
C           R IS THE ROUTINE NAMUPK                     00031240
55 IEQDF = 1                                         00031250
DO 60 I=1,6
60 NAMEQ(I) = NAMUPK(I)
65 RETURN
END

C           SUBROUTINE UGETIO(IOPT,NIN,NOUT)             00031300
C           SPECIFICATIONS FOR ARGUMENTS               00031340
C           INTEGER           IOPT,NIN,NOUT              00031350
C           SPECIFICATIONS FOR LOCAL VARIABLES         00031360
C           INTEGER           NIND,NOUTD                00031370
C           DATA              NIND/5/,NOUTD/6/            00031380
C           FIRST EXECUTABLE STATEMENT                  00031390
C           IF (IOPT.EQ.3) GO TO 10                      00031400
C           IF (IOPT.EQ.2) GO TO 5                       00031410
C           IF (IOPT.NE.1) GO TO 9005                   00031420
C           NIN = NIND                         00031430
C           NOUT = NOUTD                        00031440
C           GO TO 9005                         00031450
5 NIND = NIN                           00031460
GO TO 9005                           00031470
10 NOUTD = NOUT                        00031480
9005 RETURN                           00031490
END

C           SUBROUTINE VXADD(A,ACC)                      00031500
C           SPECIFICATIONS FOR ARGUMENTS               00031540
C           DOUBLE PRECISION A,ACC(2)                 00031560
C           SPECIFICATIONS FOR LOCAL VARIABLES         00031570
C           DOUBLE PRECISION X,Y,Z,ZZ                 00031580
C           FIRST EXECUTABLE STATEMENT                00031590
C           X = ACC(1)                         00031600
C           Y = A                            00031610
C           IF (DABS(ACC(1)).GE.DABS(A)) GO TO 1     00031620
C           X = A                            00031630
C           Y = ACC(1)                         00031640
C           COMPUTE Z+ZZ = ACC(1)+A EXACTLY          00031650
1 Z = X+Y                           00031660
ZZ = (X-Z)+Y                         00031670
C           COMPUTE ZZ+ACC(2) USING DOUBLE           00031680
C           PRECISION ARITHMETIC                  00031690
C           ZZ = ZZ+ACC(2)                         00031700
C           COMPUTE ACC(1)+ACC(2) = Z+ZZ EXACTLY    00031710
C           ACC(1) = Z+ZZ                         00031720
C           ACC(2) = (Z-ACC(1))+ZZ                 00031730
C           RETURN                                00031740
C                                         00031750

```

```

C      END                               00031760
CCC
C      SUBROUTINE VXMUL (A,B,ACC)          SPECIFICATIONS FOR ARGUMENTS 00031770
C      DOUBLE PRECISION A,B,ACC(2)          SPECIFICATIONS FOR LOCAL VARIABLES 00031780
C      DOUBLE PRECISION X,HA,TA,HB,TB      00031790
C      INTEGER IX(2),I                   00031800
C      LOGICAL LX(5),LI(4)               00031810
C      EQUIVALENCE (X,LX(1),IX(1)),(I,LI(1)) 00031820
C      DATA I/O/                         00031830
C      SPLIT A = HA+TA                  00031840
C      B = HB+TB                      00031850
C      FIRST EXECUTABLE STATEMENT        00031860
C      X = A                           00031870
C      LI(4) = LX(5)                   00031880
C      IX(2) = 0                       00031890
C      I = (I/16)*16                  00031900
C      LX(5) = LI(4)                   00031910
C      HA=X                           00031920
C      TA=A-HA                      00031930
C      X = B                           00031940
C      LI(4) = LX(5)                   00031950
C      IX(2) = 0                       00031960
C      I = (I/16)*16                  00031970
C      LX(5) = LI(4)                   00031980
C      'B = X                         00031990
C      TB = B-MB                     00032000
C      COMPUTE HAHB, HANTB, TAHB, AND TAKTB 00032010
C      AND CALL VXADD TO ACCUMULATE THE 00032020
C      SUM                           00032030
C      X = TANTB                     00032040
C      CALL VXADD(X,ACC)              00032050
C      X = HANTB                     00032060
C      CALL VXADD(X,ACC)              00032070
C      X = TAHB                       00032080
C      CALL VXADD(X,ACC)              00032090
C      X = TAKTB                     00032100
C      CALL VXADD(X,ACC)              00032110
C      X = HAHB                       00032120
C      CALL VXADD(X,ACC)              00032130
C      RETURN                         00032140
C      END                           00032150
C
C      SUBROUTINE VXSTO (ACC,D)          SPECIFICATIONS FOR ARGUMENTS 00032160
C      DOUBLE PRECISION ACC(2),D          FIRST EXECUTABLE STATEMENT 00032170
C      D = ACC(1)+ACC(2)                00032180
C      RETURN                         00032190
C      END                           00032200
C
C      SUBROUTINE ZRPOLY (A,NDEQ,Z,IER)  SPECIFICATIONS FOR ARGUMENTS 00032210
C      INTEGER NDEQ,IER                 00032220
C      DOUBLE PRECISION A(1),Z(1)         00032230
C                                         00032240
C                                         00032250
C                                         00032260
C                                         00032270
C                                         00032280
C                                         00032290
C                                         00032300
C                                         00032310
C                                         00032320
C                                         00032330
C                                         00032340
C                                         00032350

```

C	INTEGER	SPECIFICATIONS FOR LOCAL VARIABLES	00032360
	REAL	N,NN,J,JJ,I,NM1,ICHT,N2,L,NZ,NPI	00032370
	REAL	ETA,RMRE,RINFP,REPSP,RADIX,RLO,XX,YY,SINR,	00032380
	DOUBLE PRECISION	COSR,RMAX,RMIN,X,SC,XM,FT,DX,DF,BND,XXX,ARE	00032390
		PT(101)	00032400
	DOUBLE PRECISION	TEMP(101),P(101),QP(101),RK(101),QK(101),	00032410
		SVK(101)	00032420
	DOUBLE PRECISION	SR,SI,U,V,RA,RB,C,D,A1,A2,A3,	00032430
		A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,	00032440
	LOGICAL	T,AA,BU,CC,FACTOR,REPSR1,ZERO,ONE,FN	00032450
	COMMON /ZRPOLJ/	ZEROK	00032460
		P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6	00032470
		A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN	00032480
C		THE FOLLOWING STATEMENTS SET MACHINE CONSTANTS USED IN VARIOUS PARTS OF THE PROGRAM. THE MEANING OF THE FOUR CONSTANTS ARE - REPSR1 THE MAXIMUM RELATIVE REPRESENTATION ERROR WHICH CAN BE DESCRIBED AS THE SMALLEST POSITIVE FLOATING-POINT NUMBER SUCH THAT $1 + \text{REPSR1} = 1$ IS GREATER THAN 1 RINFP THE LARGEST FLOATING-POINT NUMBER REPSP THE SMALLEST POSITIVE FLOATING-POINT NUMBER IF THE EXPONENT RANGE DIFFERS IN SINGLE AND DOUBLE PRECISION THEN REPSP AND RINFP SHOULD INDICATE THE SMALLER RANGE RADIX THE BASE OF THE FLOATING-POINT NUMBER SYSTEM USED	00032490 00032500 00032510 00032520 00032530 00032540 00032550 00032560 00032570 00032580 00032590 00032600 00032610 00032620 00032630 00032640 00032650 00032660 00032670 00032680 00032690 00032700 00032710 00032720 00032730 00032740 00032750 00032760 00032770 00032780 00032790 00032800 00032810 00032820 00032830 00032840 00032850 00032860 00032870 00032880 00032890 00032900 00032910 00032920 00032930 00032940 00032950
C	DATA	RINFP/Z7FFFFFFF/	00032680
C	DATA	REPSP/Z00100000/	00032690
C	DATA	RADIX/16.0/	00032700
C	DATA	REPSR1/Z3417000000000000/	00032710
C	DATA	ZERO/0.0D0/,ONE/1.0D0/	00032720
C		ZRPOLY USES SINGLE PRECISION CALCULATIONS FOR SCALING, BOUNDS AND ERROR CALCULATIONS.	00032730
C		FIRST EXECUTABLE STATEMENT	00032740
C	IER = 0		00032750
C	IF (NDEG .GT. 100 .OR. NDEG .LT. 1) GO TO 165		00032760
C	ETA = REPSR1		00032770
C	ARE = ETA		00032780
C	RMRE = ETA		00032790
C	RLO = REPSP/ETA		00032800
C	XX = .7071068	INITIALIZATION OF CONSTANTS FOR SHIFT ROTATION	00032810
C	YY = -XX		00032820
C	SINR = .9975641		00032830
C	COSR = -.06975647		00032840
C	N = NDEG		00032850
C	NN = N+1		00032860
C	IF (A(1).NE.ZERO) GO TO 3	ALGORITHM FAILS IF THE LEADING COEFFICIENT IS ZERO.	00032870
C	IER = 130		00032880
C	GO TO 9000		00032890
C			00032900
C			00032910
C			00032920
C			00032930
C			00032940
C			00032950

```

C               REMOVE THE ZEROS AT THE ORIGIN & IF      00032960
C               ANY                                         00032970
5 IF (A(NN),NE.ZERO) GO TO 10                         00032980
J = NDEG-N+1                                         00032990
JJ = J+NDEG                                         00033000
Z(J) = ZERO                                         00033010
Z(JJ) * ZERO                                         00033020
NN = NN-1                                           00033030
N = N-1                                             00033040
IF (NN, EQ.1) GO TO 9005                           00033050
GO TO 5                                         00033060
C               MAKE A COPY OF THE COEFFICIENTS        00033070
10 DO 15 I=1,NN                                     00033080
P(I) = A(I)                                         00033090
15 CONTINUE                                         00033100
C               START THE ALGORITHM FOR ONE ZERO       00033110
20 IF (N.GT.2) GO TO 30                           00033120
IF (N.LT.1) GO TO 9005                           00033130
C               CALCULATE THE FINAL ZERO OR PAIR OF    00033140
C               ZEROS                                         00033150
IF (N.EQ.2) GO TO 25                           00033160
Z(NDEG) = -P(2)/P(1)                           00033170
Z(NDEG+NDEG) = ZERO                           00033180
GO TO 145                                         00033190
25 CALL ZRPOLI (P(1),P(2),P(3),Z(NDEG-1),Z(NDEG+NDEG-1),Z(NDEU),
1 Z(NDEG+NDEG))                                00033200
GO TO 145                                         00033210
C               FIND LARGEST AND SMALLEST MODULI OF     00033220
C               COEFFICIENTS.                        00033230
30 RMAX = 0.                                         00033240
RMIN = RINFP                                         00033250
DO 35 I=1,NN                                         00033260
X = ABS(SNGL(P(I)))                               00033270
IF (X.GT.RMAX) RMAX = X                           00033280
IF (X.NE.0.,AND.X.LT.RMIN) RMIN = X             00033290
35 CONTINUE                                         00033300
C               SCALE IF THERE ARE LARGE OR VERY      00033310
C               SMALL COEFFICIENTS COMPUTES A        00033320
C               SCALE FACTOR TO MULTIPLY THE        00033330
C               COEFFICIENTS OF THE POLYNOMIAL.     00033340
C               THE SCALING IS DONE TO AVOID        00033350
C               OVERFLOW AND TO AVOID UNDETECTED   00033360
C               UNDERFLOW INTERFERING WITH THE      00033370
C               CONVERGENCE CRITERION.            00033380
C               THE FACTOR IS A POWER OF THE BASE    00033390
SC = RLO/RMIN                                         00033400
IF (SC.GT.1.0) GO TO 40                           00033410
IF (RMAX.LT.10.) GO TO 55                           00033420
IF (SC.EQ.0.) SC = REPSPXRADIXXRADIX              00033430
GO TO 45                                         00033440
40 IF (RINFP/SC.LT.RMAX) GO TO 55                00033450
45 L = ALOG(SC)/ALOG(RADIX)+.5                  00033460
IF (L.EQ.0) GO TO 55                            00033470
FACTOR = DBLE(RADIX)**L                          00033480
DO 50 I=1,NN                                         00033490
50 P(I) = FACTORXP(I)                           00033500
C               COMPUTE LOWER BOUND ON MODULI OF      00033510
C               ZEROS.                             00033520
55 DO 60 I=1,NN                                         00033530
60 PT(I) = ABS(SNGL(P(I)))                      00033540
                                         00033550

```

```

C PT(NN) = -PT(NN)          COMPUTE UPPER ESTIMATE OF BOUND      00033560
C X = EXP((ALOG(-PT(NN))-ALOG(PT(1)))/N)                         00033570
C IF (PT(N).EQ.0.) GO TO 65                                         00033580
C C IF NEWTON STEP AT THE ORIGIN IS                                00033590
C C BETTER, USE IT.                                                 00033600
C XM = -PT(NN)/PT(N)                                              00033610
C IF (XM.LT.X) X = XM                                             00033620
C C CHOP THE INTERVAL (0,X) UNTIL FF.LE.000033640                  00033630
C 65 XM = XX,1
C   FF = PT(1)
C   DO 70 I=2,NN
C 70 FF = FF*XM+PT(I)
C   IF (FF.LE.0.) GO TO 75
C   X = XM
C   GO TO 65
C 75 DX = X
C C DO NEWTON ITERATION UNTIL X
C C CONVERGES TO TWO DECIMAL PLACES                               00033700
C 80 IF (ABS(DX/X).LE..005) GO TO 90
C   FF = PT(1)
C   DF = FF
C   DO 85 I=2,N
C     FF = FF*XM+PT(I)
C     DF = DF*XM+FF
C 85 CONTINUE
C   FF = FF*XM+PT(NN)
C   DX = FF/DF
C   X = X-DX
C   GO TO 80
C 90 BND = X
C C COMPUTE THE DERIVATIVE AS THE INITIAL
C C K POLYNOMIAL AND DO 5 STEPS WITH
C C NO SHIFT                                                       00033800
C C NM1 = N-1
C C FN = ONE/N
C C DO 95 I=2,N
C 95 RK(I) = (NN-I)*P(I)*FN
C   RK(1) = P(1)
C   AA = P(NN)
C   BB = P(N)
C   ZEROK = RK(N).EQ.ZERO
C   DO 115 JJ=1,5
C     CC = RK(N)
C     IF (ZEROK) GO TO 105
C C USE SCALED FORM OF RECURRENCE IF
C C VALUE OF K AT 0 IS NONZERO                                     00034000
C C T = -AA/CC
C C DO 100 I=1,NM1
C     J = NN-I
C     RK(J) = T*RK(J-1)+P(J)
C 100 CONTINUE
C   RK(1) = P(1)
C   ZEROK = DABS(RK(N)).LE.DABS(BB)*ETAN10.
C   GO TO 115
C C USE UNSCALED FORM OF RECURRENCE                                00034100
C 105 DO 110 I=1,NM1
C     J = NN-I
C     RK(J) = RK(J-1)
C 110 CONTINUE

```

```

RK(1) = ZERO          00034160
ZEROK = RK(N).EQ.ZERO 00034170
115 CONTINUE           00034180
C                      00034190
DO 120 I=1,N           00034200
120 TEMP(I) = RK(I)     00034210
C                      00034220
C                      00034230
DO 140 ICNT=1,20       00034240
C                      00034250
C                      00034260
C                      00034270
C                      00034280
C                      00034290
C                      00034300
C                      00034310
C                      00034320
C                      00034330
C                      00034340
C                      00034350
C                      00034360
C                      00034370
C                      00034380
C                      00034390
C                      00034400
C                      00034410
C                      00034420
C                      00034430
C                      00034440
C                      00034450
C                      00034460
C                      00034470
C                      00034480
C                      00034490
C                      00034500
C                      00034510
C                      00034520
C                      00034530
C                      00034540
C                      00034550
C                      00034560
C                      00034570
C                      00034580
C                      00034590
C                      00034600
C                      00034610
C                      00034620
C                      00034630
C                      00034640
C                      00034650
C                      00034660
C                      00034670
C                      00034680
C                      00034690
C                      00034700
C                      00034710
C                      00034720
C                      00034730
C                      00034740
C                      00034750

SAVE K FOR RESTARTS WITH NEW SHIFTS
LOOP TO SELECT THE QUADRATIC
CORRESPONDING TO EACH NEW SHIFT
QUADRATIC CORRESPONDS TO A DOUBLE
SHIFT TO A NON-REAL POINT AND ITS
COMPLEX CONJUGATE. THE POINT HAS
MODULUS BND AND AMPLITUDE ROTATED
BY 94 DEGREES FROM THE PREVIOUS
SHIFT.
SECOND STAGE CALCULATION, FIXED
QUADRATIC
THE SECOND STAGE JUMPS DIRECTLY TO
ONE OF THE THIRD STAGE ITERATIONS
AND RETURNS HERE IF SUCCESSFUL.
DEFLATE THE POLYNOMIAL, STORE THE
ZERO OR ZEROS AND RETURN TO THE
MAIN ALGORITHM.
IF THE ITERATION IS UNSUCCESSFUL
ANOTHER QUADRATIC IS CHOSEN AFTER
RESTORING K
RETURN WITH FAILURE IF NO
CONVERGENCE WITH 20 SHIFTS
CONVERT ZEROS (Z) IN COMPLEX FORM

```

```

DO 155 I=1,NDEO          00034760
Z(N2-1) = Z(J)           00034770
Z(N2) = P(J)             00034780
N2 = N2-2                00034790
J = J-1                  00034800
155 CONTINUE              00034810
IF (IER .EQ. 0) GO TO 9005 00034820
C                           SET UNFOUND ROOTS TO MACHINE INFINITY 00034830
N2 = 2*(NDEO-NN)+3        00034840
DO 160 I=1,N              00034850
Z(N2) = RINFP             00034860
Z(N2+1) = RINFP            00034870
N2 = N2+2                  00034880
160 CONTINUE              00034890
GO TO 9000                00034900
165 IER = 129              00034910
9000 CONTINUE              00034920
CALL UERTST (IER,6HZRPOLY) 00034930
9005 RETURN                00034940
END                         00034950
C
C
C   SUBROUTINE ZRPQLB (L2,NZ)          SPECIFICATIONS FOR ARGUMENTS 00034960
C   INTEGER      L2,NZ                 SPECIFICATIONS FOR LOCAL VARIABLES 00035000
C   INTEGER      N,NN,J,ITYPE,I,IFLAQ  00035010
C   REAL         AKE,BETAS,BETAV,ETA,OSS,OTS,OTV,OVV,RHRE,SS, 00035020
C   DOUBLE PRECISION    TS,TSS,TV,TVV,VV  00035030
C   DOUBLE PRECISION    PC(101),QPC(101),RK(101),QK(101),SVK(101) 00035040
C   LOGICAL      SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00035050
C   COMMON /ZRPQLJ/    A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00035060
C   DATA         SVU,SVV,UI,VI,S,ZERO  00035070
C               VPASS,SPASS,VTRY,STRY  00035080
C               P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00035090
C               A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00035100
C               ZERO/0.000/  00035120
C   NZ = 0                   FIRST EXECUTABLE STATEMENT 00035130
C
C   COMPUTES UP TO L2 FIXED SHIFT 00035140
C   K-POLYNOMIALS, TESTING FOR 00035150
C   CONVERGENCE IN THE LINEAR OR 00035160
C   QUADRATIC CASE. INITIATES ONE OF 00035170
C   THE VARIABLE SHIFT ITERATIONS AND 00035180
C   RETURNS WITH THE NUMBER OF ZEROS 00035190
C   FOUND. 00035200
C   L2 - LIMIT OF FIXED SHIFT STEPS 00035210
C   NZ - NUMBER OF ZEROS FOUND 00035220
C
C   BETAV = .25              EVALUATE POLYNOMIAL BY SYNTHETIC 00035230
C   BETAS = .25              DIVISION 00035240
C   OSS = SR                 00035250
C   OVV = V                  00035260
C
C   CALL ZRPQLH (NN,U,V,P,QP,RA,RB) 00035270
C   CALL ZRPQLE (ITYPE)        00035280
C   DO 40 J=1,L2              CALCULATE NEXT K POLYNOMIAL AND 00035290
C                               ESTIMATE V 00035300

```

```

CALL ZRPQLF (IITYPE)          00035360
CALL ZRPQLE (IITYPE)          00035370
CALL ZRPQLG (IITYPE,UI,VI)    00035380
VV = VI                      00035390
C                           ESTIMATE S          00035400
SS = 0.                      00035410
IF (RK(N).NE.ZERO) SS = -P(NN)/RK(N) 00035420
TV = 1.                      00035430
TS = 1.                      00035440
IF (J.EQ.1.OR.IITYPE.EQ.3) GO TO 35 00035450
C                           COMPUTE RELATIVE MEASURES OF 00035460
                                CONVERGENCE OF S AND V SEQUENCES 00035470
IF (VV.NE.0.) TV = ABS((VV-OVV)/VV) 00035480
IF (SS.NE.0.) TS = ABS((SS-OSS)/SS) 00035490
C                           .. IF DECREASING, MULTIPLY TWO MOST 00035500
                                RECENT CONVERGENCE MEASURES 00035510
TVV = 1.                      00035520
IF (TV.LT.OTV) TVV = TV*OTV 00035530
TSS = 1.                      00035540
IF (TS.LT.OTS) TSS = TS*OTS 00035550
C                           COMPARE WITH CONVERGENCE CRITERIA 00035560
VPASS = TVV.LT.BETAV 00035570
SPASS = TSS.LT.BETAS 00035580
IF (.NOT.(SPASS.OR.VPASS)) GO TO 35 00035590
C                           AT LEAST ONE SEQUENCE HAS PASSED THE 00035600
                                CONVERGENCE TEST. STORE VARIABLES 00035610
                                BEFORE ITERATING 00035620
SVU = U                      00035630
SVV = V                      00035640
DO 5 I=1,N                  00035650
SVK(I) = RK(I)              00035660
S = SS                      00035670
C                           CHOOSE ITERATION ACCORDING TO THE 00035680
                                FASTEST CONVERGING SEQUENCE 00035690
VTRY = .FALSE.               00035700
STRY = .FALSE.               00035710
IF (SPASS AND ((.NOT.VPASS).OR.TSS.LT.TVV)) GO TO 20 00035720
10   CALL ZRPQLC (UI,VI,NZ)    00035730
IF (NZ.GT.0) RETURN          00035740
C                           QUADRATIC ITERATION HAS FAILED. FLAG 00035750
                                THAT IT HAS BEEN TRIED AND 00035760
                                DECREASE THE CONVERGENCE 00035770
                                CRITERION. 00035780
VTRY = TRUE.                 00035790
BFTAV = BETAV*.25           00035800
C                           TRY LINEAR ITERATION IF IT HAS NOT 00035810
                                BEEN TRIED AND THE S SEQUENCE IS 00035820
                                CONVERGING 00035830
IF (STRY.OR.(.NOT.SPASS)) GO TO 25 00035840
15   DO 15 I=1,N              00035850
RK(I) = SVK(I)              00035860
20   CALL ZRPQLD (S,NZ,IFLAG) 00035870
IF (NZ.GT.0) RETURN          00035880
C                           LINEAR ITERATION HAS FAILED. FLAG 00035890
                                THAT IT HAS BEEN TRIED AND 00035900
                                DECREASE THE CONVERGENCE CRITERION 00035910
STRY = .TRUE.                00035920
BETAS = BETAS*.25           00035930
IF (IFLAG.EQ.0) GO TO 25    00035940
C                           IF LINEAR ITERATION SIGNALS AN 00035950

```

```

C           ALMOST DOUBLE REAL ZERO ATTEM.      00035960
C           QUADRATIC ITERATION                 00035970
C
C           UI = -(S+S)                         00035980
C           VI = S*S                           0C035990
C           GO TO 10                          0U036000
C
C           RESTORE VARIABLES                00036010
C
C           25      U = S*U                         00036020
C           V = SVV                           00036030
C           DO 30 I=1,N                         00036040
C           RK(I) = .K(I)                      00036050
C
C           TRY QUADRATIC ITERATION IF IT HAS    00036060
C           NOT BEEN TRIED AND THE V SEQUENCE     00036070
C           IS CONVERGING                      00036080
C
C           IF (VPASS.AND.(.NOT.VTRY)) GO TO 10    00036090
C
C           RECOMPUTE QP AND SCALAR VALUES TO    00036100
C           CONTINUE THE SECOND STAGE            00036110
C
C           CALL ZRPQLH (NN,U,V,P,QP,RA,RB)        00036120
C           CALL ZRPQLE (ITYPE)                   00036130
C
C           35      OVV = VV                         00036140
C           OSS = SS                           00036150
C           OTV = TV                           00036160
C           OTS = TS                           00036170
C
C           40      CONTINUE                        00036180
C           RETURN
C           END
C
C           SUBROUTINE ZRPQLC (UU,VV,NZ)          SPECIFICATIONS FOR ARGUMENTS 00036220
C
C           INTEGER             NZ                  00036230
C           DOUBLE PRECISION   UU,VV               00036240
C
C           SPECIFICATIONS FOR LOCAL VARIABLES 00036250
C
C           INTEGER             N,NN,J,I,ITYPE      00036260
C           REAL                ARE,EE,ETA,OMP,RELSTP,RMP,RMRH,T,ZM 00036270
C           DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101) 00036280
C           DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00036290
C
C           1                  A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00036300
C           2                  UI,VI,ZERO,PT01,ONE              00036310
C           LOGICAL            TRIED
C           COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00036320
C
C           1                  A7,E,F,G,H,SZR,SZI,PLZR,RLZI,ETA,ARE,RMRE,N,NN 00036330
C           DATA               ZERO,PT01,ONE/0.000,0.01D0,1.CDD/ 00036340
C
C           FIRST EXECUTABLE STATEMENT            00036350
C
C           NZ = 0
C
C           VARIABLE-SHIFT K-POLYNOMIAL          00036410
C           ITERATION FOR A QUADRATIC FACTOR 00036420
C           CONVERGES ONLY IF THE ZEROS ARE    00036430
C           EQUIMODULAR OR NEARLY SO          00036440
C
C           UU,VV - COEFFICIENTS OF STARTING 00036450
C           QUADRATIC
C
C           NZ - NUMBER OF ZERO FOUND        00036460
C
C           TRIED = .FALSE.                    00036470
C           U = UU                           00036480
C           V = VV                           00036490
C           J = 0                            00036500
C
C           MAIN LOOP
C
C           5 CALL ZRPQLI (ONE,U,V,SZR,SZI,RLZR,RLZI) 00036520
C
C           RETURN IF ROOTS OF THE QUADRATIC ARE 00036530
C           REAL AND NOT CLOSE TO MULTIPLE OR 00036540
C                                         00036550

```

```

C      IF ( DABS(DABS(SZR)-DABS(RLZR)).GT.PT01*DABS(RLZR) ) RETURN 00036560
C          EVALUATE POLYNOMIAL BY QUADRATIC 00036570
C          SYNTHETIC DIVISION 00036580
C          CALL ZRPQLH (NN,U,V,P,QP,RA,RB) 00036590
C          RMD = DABS(RA-SZR*RB)+DABS(SZI*RB) 00036600
C          COMPUTE A RIGOROUS BOUND ON THE 00036610
C          ROUNDING ERROR IN EVALUTING P 00036620
C          ZM = SQRT(ABS(SNGL(V))) 00036630
C          EE = 2.*XABS(SNGL(QP(1))) 00036640
C          T = -SZR*RB 00036650
C          DO 10 I=2,N 00036660
C          10 EE = EE*ZM+ABS(SNGL(QP(I))) 00036670
C          EE = EE*ZM+1.05*(SNGL(RA)+T) 00036680
C          EE = (5.*RMRE+4.*ARE)*EE-(5.*RMRE+2.*ARE)*(ABS(SNGL(RA)+T)+ 00036690
C          ABS(SNGL(RD))*ZM)+2.*ARE*A8S(T) 00036700
C          CCC          ITERATION HAS CONVERGED SUFFICIENTLY 00036710
C          IF THE POLYNOMIAL VALUE IS LESS 00036720
C          THAN 20 TIMES THIS BOUND 00036730
C          15 J = J+1 00036740
C          IF (RMP.GT.20.*EE) GO TO 15 00036750
C          NZ = 2 00036760
C          RETURN 00036770
C          15 STOP ITERATION AFTER 20 STEPS 00036780
C          IF (J.GT.20) RETURN 00036790
C          IF (J.LT.2) GO TO 25 00036800
C          IF (RELSTP.OT..01.OR.RMP.LT.OMP.OR.TRIED) GO TO 25 00036810
C          CCC          A CLUSTER APPEARS TO BE STALLING THE 00036820
C          CONVERGENCE. FIVE FIXED SHIFT 00036830
C          STEPS ARE TAKEN WITH A U,V CLOSE 00036840
C          TO THE CLUSTER 00036850
C          20 CONTINUE 00036860
C          TRIFD = .TRUE. 00036870
C          J = 0 00036880
C          25 OMP = RMP 00036890
C          CCC          CALCULATE NEXT K POLYNOMIAL AND NEW 00036900
C          U AND V 00036910
C          CALL ZRPQLE (ITYPE) 00036920
C          CALL ZRPQLF (ITYPE) 00036930
C          CALL ZRPQLE (ITYPE) 00036940
C          CALL ZRPQLG (ITYPE,UI,VI) 00036950
C          CCC          IF VI IS ZERO THE ITERATION IS NOT 00036960
C          CONVERGING 00036970
C          IF (VI.EQ.ZERO) RETURN 00036980
C          RELSTP = DABS((VI-V)/VI) 00036990
C          U = UI 00037000
C          V = VI 00037010
C          GO TO 5 00037020
C          END 00037030
C          CCC          00037040
C          IF (VI.EQ.ZERO) RETURN 00037050
C          RELSTP = DABS((VI-V)/VI) 00037060
C          U = UI 00037070
C          V = VI 00037080
C          GO TO 5 00037090
C          END 00037100
C          CCC          00037110
C          IF (VI.EQ.ZERO) RETURN 00037120
C          RELSTP = DABS((VI-V)/VI) 00037130
C          U = UI 00037140
C          V = VI 00037150

```

```

C SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)          SPECIFICATIONS FOR ARGUMENTS      00037160
C INTEGER NZ,IFLAG                           00037170
C DOUBLE PRECISION SSS                      00037180
C INTEGER NZ,IFLAG                           00037190
C DOUBLE PRECISION SSS                      00037200
C INTEGER N,NN,J,I                          00037210
C REAL ARE,EE,ETA,OMP,RMP,RMS,RMRE        00037220
C DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00037230
C DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3,       00037240
C A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,       00037250
C PV,RKV,T,S,ZERO,PT001                   00037260
C COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00037270
C DATA ZERO/0.0D0/,PT001/0.001D0/           00037280
C                                     VARIABLE-SHIFT H POLYNOMIAL      00037290
C                                     ITERATION FOR A REAL ZERO SSS - 00037300
C                                     STARTING ITERATE             00037310
C                                     NZ - NUMBER OF ZERO FOUND    00037320
C                                     IFLAG - FLAG TO INDICATE A PAIR OF 00037330
C                                     ZEROS NEAR REAL AXIS          00037340
C                                     FIRST EXECUTABLE STATEMENT   00037350
C                                     NZ = 0                         00037360
C                                     S = SSS                         00037370
C                                     IFLAG = 0                      00037380
C                                     J = 0                         00037390
C                                     MAIN LOOP                     00037400
C                                     5 PV = P(1)                    00037410
C                                     EVALUATE P AT S              00037420
C                                     QP(1) = PV                   00037430
C                                     DO 10 I=2,NN                 00037440
C                                     PV = PV*S+P(I)              00037450
C                                     QF(I) = PV                  00037460
C                                     10 CONTINUE                   00037470
C                                     RMP = DABS(PV)               00037480
C                                     COMPUTE A RIGOROUS BOUND ON THE 00037490
C                                     RMS = DABS(S)                00037500
C                                     EE = (RMRE/(ARE+RMRE))*ABS(SNGL(QP(1))) 00037510
C                                     DO 15 I=2,NN                 00037520
C                                     15 EE = EE*RMS+ABS(SNGL(QP(I))) 00037530
C                                     IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20 00037540
C                                     NZ = 1                         00037550
C                                     IF THE POLYNOMIAL VALUE IS LESS 00037560
C                                     THAN 20 TIMES THIS BOUND     00037570
C                                     IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20 00037580
C                                     NZ = 1                         00037590
C                                     IF (J.GT.10) RETURN           00037600
C                                     IF (J.LT.2) GO TO 25          00037610
C                                     IF (DABS(T).GT.PT001*DABS(S-T).OR.RMP.LE.OMP) GO TO 25 00037620
C                                     A CLUSTER OF ZEROS NEAR THE REAL 00037630
C                                     AXIS HAS BEEN ENCOUNTERED RETURN 00037640
C                                     WITH IFLAG SET TO INITIATE A 00037650
C                                     QUADRATIC ITERATION          00037660
C                                     IFLAG = 1                      00037670
C                                     SDS = S                        00037680
C                                     RETURN                         00037690
C                                     STOP ITERATION AFTER 10 STEPS 00037700
C                                     IF (J.LT.2) GO TO 25          00037710
C                                     A CLUSTER OF ZEROS NEAR THE REAL 00037720
C                                     AXIS HAS BEEN ENCOUNTERED RETURN 00037730
C                                     WITH IFLAG SET TO INITIATE A 00037740
C                                     QUADRATIC ITERATION          00037750

```

```

C          RETURN IF THE POLYNOMIAL VALUE HAS      00037760
C          INCREASED SIGNIFICANTLY             00037770
C          COMPUTE T, THE NEXT POLYNOMIAL, AND     00037780
C          THE NEW ITERATE                   00037790
C          RKV = RK(1)                         00037800
C          QK(1) = RKV                         00037810
C          DO 30 I=2,N                         00037820
C              RKV = RKVMS+RK(I)                 00037830
C              QK(I) = RKV                     00037840
C 30 CONTINUE
C          IF (DABS(RKV).LE.DABS(RK(N))*10.*ETA) GO TO 40
C          USE THE SCALED FORM OF THE          00037850
C          RECURRENCE IF THE VALUE OF K AT S     00037860
C          IS NONZERO                         00037870
C          RK(1) = -PV/RKV                      00037880
C          QK(1) = QP(1)                        00037890
C          DO 35 I=2,N                         00037900
C              RK(I) = T*QK(I-1)+QP(I)           00037910
C              GO TO 50                         00037920
C          USE UNSCALED FORM                  00037930
C          RK(1) = ZERO                         00037940
C          DO 45 I=2,N                         00037950
C              RK(I) = QK(I-1)                   00037960
C          RKV = RK(1)                         00037970
C          DO 55 I=2,N                         00037980
C              RKV = RKVMS+RK(I)                 00037990
C              T = ZERO                         00038000
C              IF (DABS(RKV).GT.DABS(RK(N))*10.*ETA) T = -PV/RKV
C              S = S*T                         00038010
C              GO TO 5                         00038020
C          END
C
C          IMSL ROUTINE NAME - ZRPQLE          00038030
C
C-----  

C          COMPUTER      - IBM/DOUBLE          00038100
C          LATEST REVISION - JANUARY 1, 1978    00038110
C          SUBROUTINE ZRPQLE (ITYPE)            00038120
C          INTEOER       ITYPE    SPECIFICATIONS FOR ARGUMENTS 00038130
C          INTEGER        N,NN    SPECIFICATIONS FOR LOCAL VARTABLES 00038140
C          REAL           ARE,ETA,RMRE          00038150
C          DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00038160
C          DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3,          00038170
C          1 COMMON /ZRPQLJ/ A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI          00038180
C          1 P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,          00038190
C          THIS ROUTINE CALCULATES SCALAR          00038200
C          QUANTITIES USED TO COMPUTE THE        00038210
C          NEXT K POLYNOMIAL AND NEW           00038220
C          ESTIMATES OF THE QUADRATIC          00038230
C          COEFFICIENTS                      00038240
C          ITYPE - INTEGER VARIABLE SET HERE   00038250

```

```

C          INDICATING HOW THE CALCULATIONS      00038360
CCC          ARE NORMALIZED TO AVOID OVERFLOW      00038370
C          SYNTHETIC DIVISION OF K BY THE      00038380
C          QUADRATIC 1,U,V      00038390
C          FIRST EXECUTABLE STATEMENT      00038400
C          00038410
C          CALL ZRPQLH (N,U,V,RK,QK,C,D)      00038420
C          IF (DABS(C).GT.DABS(RK(N))X100.XETA) GO TO 5      00038430
C          IF (DABS(D).GT.DABS(RK(N-1))X100.XETA) GO TO 5      00038440
C          ITYPE = 3      00038450
C          TYPE=3 INDICATES THE QUADRATIC IS      00038460
C          ALMOST A FACTOR OF K      00038470
C          RETURN      00038480
C          5 IF (DABS(D).LT.DABS(C)) GO TO 10      00038490
C          ITYPE = 2      00038500
C          TYPE=2 INDICATES THAT ALL FORMULAS      00038510
C          ARE DIVIDED BY D      00038520
C          E = RA/D      00038530
C          F = C/D      00038540
C          G = UXRB      00038550
C          H = VXRB      00038560
C          A3 = (RA+G)XE+HK(RB/D)      00038570
C          A1 = RBXF-RA      00038580
C          A7 = (F+U)*RA+H      00038590
C          RETURN      00038600
C          10 ITYPE = 1      00038610
C          TYPE=1 INDICATES THAT ALL FORMULAS      00038620
C          ARE DIVIDED BY C      00038630
C          E = RA/C      00038640
C          F = D/C      00038650
C          G = UME      00038660
C          H = VXRB      00038670
C          A3 = RAHE+(H/C+G)XRB      00038680
C          A1 = RB-RAK(D/C)      00038690
C          A7 = RA+GKD+HNF      00038700
C          RETURN      00038710
C          END      00038720
C          00038730
C          00038740
C          SUBROUTINE ZRPQLF (ITYPE)      00038750
C          INTEGER ITYPE      00038760
C          INTEGER      00038770
C          REAL N,NN,I      00038780
C          REAL ARE,ETA,RMRE      00038790
C          DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101)      00038800
C          DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3,      00038810
C          1 A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,TEMP,ZERO      00038820
C          COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,      00038830
C          1 A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN      00038840
C          DATA ZERO/0.000/      00038850
C          COMPUTES THE NEXT K POLYNOMIALS      00038860
C          USING SCALARS COMPUTED IN ZRPQLF      00038870
C          FIRST EXECUTABLE STATEMENT      00038880
C          00038890
C          IF (ITYPE.EQ.3) GO TO 20      00038900
C          TEMP = RA      00038910
C          IF (ITYPE.EQ.1) TEMP = RB      00038920
C          IF (DABS(A1).GT.DABS(TEMP)XETAN10.) GO TO 10      00038930
C          IF A1 IS NEARLY ZERO THEN USE A      00038940
C          SPECIAL FORM OF THE RECURRENCE      00038950

```

```

      RK(1) = LXR0          00038960
      RK(2) = -A7XQP(1)    00038970
      DO 5 I=3,N            00038980
      5 RK(I) = A3XQK(I-2)-A7XQP(I-1) 00038990
      RETURN                00039000
C                                     USE SCALED FORM OF THE RECURRENCE 00039010
      10 A7 = A7/A1          00039020
      A3 = A3/A1          00039030
      RK(1) = QP(1)        00039040
      RK(2) = QP(2)-A7XQP(1) 00039050
      DO 15 I=3,N            00039060
      15 RK(I) = A3XQK(I-2)-A7XQP(I-1)+QP(I) 00039070
      RETURN                00039080
C                                     USE UNSCALED FORM OF THE RECURRENCE 00039090
C                                     IF TYPE IS 3                   00039100
      20 RK(1) = ZERO        00039110
      RK(2) = ZERO        00039120
      DO 25 I=3,N            00039130
      25 RK(I) = QK(I-2)    00039140
      RETURN                00039150
      END                   00039160
C                                     00039170
C                                     00039180
C                                     00039190
C                                     IMSL ROUTINE NAME - ZRPQL0 00039200
C                                     00039210
C----- 00039220
C                                     COMPUTER - IBM/DOUBLE 00039230
C                                     00039240
C                                     LATEST REVISION - JANUARY 1, 1978 00039250
C                                     00039260
C                                     SUBROUTINE ZRPQLG (ITYPE,UU,VV) 00039270
C                                     SPECIFICATIONS FOR ARGUMENTS 00039280
C                                     INTEGER ITYPE 00039290
C                                     DOUBLE PRECISION UU,VV 00039300
C                                     00039310
C                                     SPECIFICATIONS FOR LOCAL VARIABLES 00039320
C                                     INTEGER N,NN 00039330
C                                     REAL ARE,ET1,RMRE 00039340
C                                     DOUBLE PRECISION PC(101),QP(101),RK(101),QK(101),SVK(101) 00039350
C                                     DOUBLE PRECISION SP,SI,U,V,RA,RB,C,D,A1,A2,A3, 00039360
C                                     A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00039370
C                                     A4,A5,B1,B2,C1,C2,C3,C4,TEMP,ZERO 00039380
C                                     COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00039390
C                                     DATA A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00039400
C                                     ZERO/0.0D0/ 00039410
C                                     COMPUTE NEW ESTIMATES OF THE 00039420
C                                     QUADRATIC COEFFICIENTS USING THE 00039430
C                                     SCALARS COMPUTED IN ZRPQLG 00039440
C                                     USE FORMULAS APPROPRIATE TO SETTING 00039450
C                                     OF TYPE. 00039460
C                                     FIRST EXECUTABLE STATEMENT 00039470
C                                     00039480
C                                     IF (ITYPE.EQ.3) GO TO 15 00039490
C                                     IF (ITYPE.EQ.2) GO TO 5 00039500
C                                     A4 = RA+U*RB+H*F 00039510
C                                     A5 = C+(U+V*F)*D 00039520
C                                     GO TO 10 00039530
C                                     5 A4 = (RA+G)*F+H 00039540
C                                     A5 = (F+U)*C+V*D 00039550
C                                     EVALUATE NEW QUADRATIC COEFFICIENTS. 00039550

```

```

C
10 B1 = -RK(N)/P(NN) 00039560
B2 = -(RK(N-1)+B1XP(N))/P(NN) 00039570
C1 = VMB2XA1 00039580
C2 = B1XA7 00039590
C3 = B1XB1XA3 00039600
C4 = C1-C2-C3 00039610
TEMP = A5+B1XA4-C4 00039620
IF (TEMP.EQ.ZERO) GO TO 15 00039630
UU = U-(UX(C3+C2)+VX(B1XA1+B2XA7))/TEMP 00039640
VV = VX(1+C4/TEMP) 00039650
RETURN 00039660
00039670
C IF TYPE=3 THE QUADRATIC IS ZEROED 00039680
15 UU = ZERO 00039690
VV = ZERO 00039700
RETURN 00039710
END 00039720
00039730
CCC
SUBROUTINE ZRPOLH (NN,U,V,P,Q,RA,RB) 00039740
C SPECIFICATIONS FOR ARGUMENTS 00039750
INTEGER NN 00039760
DOUBLE PRECISION P(NN),Q(NN),U,V,RA,RB 00039770
C SPECIFICATIONS FOR LOCAL VARIABLES 00039780
INTEGER I 00039790
DOUBLE PRECISION C 00039800
CCC DIVIDES P BY THE QUADRATIC 1,U,V 00039810
PLACING THE QUOTIENT IN Q AND THE 00039820
REMAINDER IN A,B 00039830
FIRST EXECUTABLE STATEMENT 00039840
RB = P(1) 00039850
Q(1) = RB 00039860
RA = P(2)-UXRB 00039870
Q(2) = RA 00039880
DO 5 I=3,NN 00039890
C = P(I)-UXRA-VXRb 00039900
Q(I) = C 00039910
RB = RA 00039920
RA = C 00039930
5 CONTINUE 00039940
RETURN 00039950
END 00039960
00039970
00039980
00039990
00040000
IMSL ROUTINE NAME - ZRPQLI 00040010
C 00040020
C----- 00040030
COMPUTER - IBM/DOUBLE 00040040
C 00040050
LATEST REVISION - JANUARY 1, 1978 00040060
C 00040070
SUBROUTINE ZRPQLI (RA,B1,C,SR,SI,RLR,RLI) 00040080
C SPECIFICATIONS FOR ARGUMENTS 00040090
DOUBLE PRECISION RA,B1,C,SR,SI,RLR,RLI 00040100
C SPECIFICATIONS FOR LOCAL VARIABLES 00040110
DOUBLE PRECISION R6,D,E,ZERO,ONE,TWO 00040120
DATA ZERO,ONE,TWO/0.0D0,1.0D0,2.0D0/ 00040130
00040140
00040150

```

CALCULATE THE ZEROS OF THE QUADRATIC 00040160
 $AZ^2 + BZ + C$. THE QUADRATIC 00040170
 FORMULA, MODIFIED TO AVOID 00040180
 OVERFLOW, IS USED TO FIND THE 00040190
 LARGER ZERO IF THE ZEROS ARE REAL 00040200
 AND BOTH ZEROS ARE COMPLEX. 00040210
 THE SMALLER REAL ZERO IS FOUND 00040220
 DIRECTLY FROM THE PRODUCT OF THE 00040230
 ZEROS C/A 00040240
 FIRST EXECUTABLE STATEMENT 00040250

```

    IF (RA.NE.ZERO) GO TO 10      00040260
    SR = ZERO                    00040270
    IF (B1.NE.ZERO) SR = -C/B1   00040280
    RLR = ZERO                   00040290
  5  SI = ZERO                   00040300
    RLI = ZERO                   00040310
    RETURN                       00040320
  10 IF (C.NE.ZERO) GO TO 15    00040330
    SR = ZERO                   00040340
    RLR = -B1/RA                00040350
    GO TO 5                     00040360
  C COMPUTE DISCRIMINANT AVOIDING 00040370
    OVERFLOW                     00040380
  15 RB = B1/TWO                00040390
    IF (DABS(RB).LT.DABS(C)) GO TO 20 00040400
    E = ONE-(RA/RB)*(C/RB)        00040410
    S = DSQRT(DABS(E))*DABS(RB)  00040420
    GO TO 25                   00040430
  20 E = RA                     00040440
    IF (C.LT.ZERO) E = -RA       00040450
    E = RB*(RB/DABS(C))-E       00040460
    D = DSQRT(DABS(E))*DSQRT(DABS(C)) 00040470
  25 IF (E.LT.ZERO) GO TO 30   00040480
  C REAL ZEROS                 00040490
    IF (RB.OE.ZERO) D = -D     00040500
    RLR = (-RB+D)/RA           00040510
    SR = ZERO                   00040520
    IF (RLR.NE.ZERO) SR = (C/RLR)/RA 00040530
    GO TO 3                     00040540
  C COMPLEX CONJUGATE ZEROS   00040550
  30 SR = -RB/RA               00040560
    RLR = SR                   00040570
    SI = DABS(D/RA)             00040580
    RLI = -SI                  00040590
    RETURN                      00040600
    END                         00040610
  C
  C SUBROUTINE LEQ2C (A,N,IA,B,M,IB,IJOB,WA,WK,IER)
  C
    COMPLEX*16                  A(IA,1),B(IB,1),WA(H,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),
    X                               (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI),
    X                               (TC(1),CR),(TC(2),CI)
    DATA                          ZERO/0.0D0/
    DATA                          ITMAX/50/
  
```

C	IER = 0	FIRST EXECUTABLE STATEMENT	00040760
-	N1 = N+1		00040770
-	N2 = N+2		00040780
C	IF (IJOB .EQ. 2) GO TO 15		00040790
C	DO 10 I = 1,N	SAVE MATRIX A	00040800
-	DO 5 J = 1,N		00040810
-	WA(I,J) = A(I,J)		00040820
5	CONTINUE		00040830
C	10 CONTINUE		00040840
C	FACTOR MATRIX A		00040850
-	CALL LEQTIC (WA,N,N,B,M,IB,1,WK,IER)		00040860
-	IF (IER .NE. 0) GO TO 9000		00040870
-	IF (IJOB .EQ. 1) GO TO 9005		00040880
C	SAVE THE RIGHT HAND SIDES		00040890
15	DO 65 J = 1,M		00040900
-	DO 20 I = 1,N		00040910
-	WA(I,N1) = B(I,J)		00040920
20	CONTINUE		00040930
C	OBTAIN A SOLUTION		00040940
C	CALL LEQTIC(WA,N,N,WA(1,N1),1,N,2,WK,IER)	COMPUTE THE NORM OF THE SOLUTION	00040950
-	XNORM = ZERO		00040960
-	DO 23 I = 1,N		00040970
-	TEMPA = WA(I,N1)		00040980
-	XNORM = DMAX1(XNORM,DABS(AI),DABS(BI))		00040990
23	CONTINUE		00041000
C	IF (XNORM .EQ. ZERO) GO TO 65	COMPUTE RESIDUALS	00041010
C	DO 50 ITER = 1,ITMAX		00041020
-	DO 40 I = 1,N		00041030
-	TEMPB = B(I,J)		00041040
-	ACC(1) = 0.000		00041050
-	ACC(2) = 0.000		00041060
-	CALL VXADD(BR,ACC)		00041070
-	DO 30 JJ = 1,N		00041080
-	TEMPA = A(I,JJ)		00041090
-	TEMPB = WA(JJ,N1)		00041100
-	CALL VXMUL(-AR,PR,ACC)		00041110
-	CALL VXMUL(AI,BI,ACC)		00041120
30	CONTINUE		00041130
-	CALL VXSTO(ACC,CR)		00041140
-	TEMPB = B(I,J)		00041150
-	ACC(1) = 0.000		00041160
-	ACC(2) = 0.000		00041170
-	CALL VXADD(BI,ACC)		00041180
-	DO 35 JJ = 1,N		00041190
-	TEMPA = A(I,JJ)		00041200
-	TEMPB = WA(JJ,N1)		00041210
-	CALL VXMUL(-AR,BI,ACC)		00041220
-	CALL VXMUL(-BR,AI,ACC)		00041230
35	CONTINUE		00041240
-	CALL VXSTO(ACC,CI)		00041250
-	WA(I,N2) = TEMPB		00041260
40	CONTINUE		00041270
-	CALL LEQTIC(WA,N,N,WA(1,N2),1,N,2,WK,IER)		00041280
-	DXNORM = ZERO		00041290
-	DO 45 I = 1,N	UPDATE THE SOLUTION	00041300
-			00041310
-			00041320
-			00041330
-			00041340
-			00041350

```

        ..A(I,N1) = WA(I,N1)+WA(I,N2)          00041360
        TEMPA = WA(I,N2)                      00041370
        DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI)) 00041380
45      CONTINUE                                00041390
        IF (XNORM+DXNORM .EQ. XNORM) GO TO 55    00041400
50      CONTINUE                                00041410
        IER = 130                                00041420
C       STORE THE SOLUTION                      00041430
55      DO 60 JK = 1,N                         00041440
        B(JK,J) = WA(JK,N1)                      00041450
60      CONTINUE                                00041460
        IF (IER .NE. 0) GO TO 9000                00041470
65      CONTINUE                                00041480
        GO TO 9005                                00041490
9000    CONTINUE                                00041500
        CALL VERTST(IER,6HLEQ2C )                 00041510
9005    RETURN                                 00041520
        END                                     00041530
00041540
C
C
C       SUBROUTINE LEQTIC (A,N,IA,B,M,IB,IJOB,WA,IER)
C                                         SPECIFICATIONS FOR ARGUMENTS 00041560
C       INTEGER           N,IA,M,IB,IJOB,IER          00041570
C       COMPLEX*16         A(IA,N),B(IB,M)          00041580
C       DOUBLE PRECISION   WA(N)                   00041590
C                                         SPECIFICATIONS FOR LOCAL VARIABLES 00041600
C       DOUBLE PRECISION   P,Q,ZERO,ONE,T(2),RN,BIG 00041610
C       COMPLEX*16         SUM,TEMP               00041620
C       INTEGER            I,J,JM1,IM1,K,IMAX,JPI,IW,N1 00041630
C       EQUIVALENCE        (SUM,T(1))              00041640
C       DATA               ZERO/0.000/,ONE/1.00/ 00041650
C                                         INITIALIZATION 00041660
C                                         FIRST EXECUTABLE STATEMENT 00041670
C       IER = 0                               00041680
C       IF (IJOB .EQ. 2) GU TO 75             00041690
C       RN = N                               00041700
C                                         FIND EQUILIBRATION FACTORS 00041710
C       DO 10 I=1,N                         00041720
C         BIG = ZERO                      00041730
C         DO 5 J=1,N                     00041740
C           TEMP = A(I,J)                  00041750
C           P = CDABS(TEMP)                00041760
C           IF (P .GT. BIG) BIG = P      00041770
5      CONTINUE                                00041780
        IF (BIG .EQ. ZERO) GO TO 105        00041790
        WA(I) = ONE/BIG                  00041800
10      CONTINUE                                00041810
C                                         L-U DECOMPOSITION 00041820
C       DO 70 J = 1,N                         00041830
C         JM1 = J-1                      00041840
C         IF (JM1 .LT. 1) GO TO 25        00041850
C                                         COMPUTE U(I,J), I=1,...,J-1 00041860
C       DO 20 I=1,JM1                      00041870
C         SUM = A(I,J)                  00041880
C         IM1 = I-1                      00041890
C         IF (IM1 .LT. 1) GO TO 20        00041900
C         DO 15 K=1,IM1                  00041910
C           SUM = SUM-A(I,K)*WA(K,J)    00041920
15      CONTINUE                                00041930
C                                         00041940
C                                         00041950

```

```

      A(I,J) = SUM          00041960
20    CONTINUE          00041970
25    P = ZERO          00041980
C     DO 45 I=J,N          COMPUTE U(J,J) AND L(I,J), I=J+1,...,00041990
      SUM = A(I,J)          00042000
      IF (JM1 .LT. 1) GO TO 40          00042010
      DO 35 K=1,JM1          00042020
      SUM = SUM-A(I,K)*ACK,J)
35    CONTINUE          00042030
      A(I,J) = SUM          00042040
40    Q = WA(I)*CDABS(SUM)          00042050
      IF (P .GE. Q) GO TO 45          00042060
      P = Q          00042070
      IMAX = I          00042080
45    CONTINUE          00042090
C     TEST FOR ALGORITHMIC SINGULARITY          00042100
      Q = RN+P          00042110
      IF (Q .EQ. RN) GO TO 105          00042120
      IF (J .EQ. IMAX) GO TO 60          00042130
C     INTERCHANGE ROWS J AND IMAX          00042140
      DO 50 K=1,N          00042150
      TEMP = A(IMAX,K)          00042160
      A(IMAX,K) = A(J,K)          00042170
      A(J,K) = TEMP          00042180
50    CONTINUE          00042190
      WA(IMAX) = WA(J)          00042200
60    WA(J) = IMAX          00042210
      JP1 = J+1          00042220
      IF (JP1 .GT. N) GO TO 70          00042230
C     DIVIDE BY PIVOT ELEMENT U(J,J)          00042240
      TEMP = A(J,J)          00042250
      DO 65 I = JP1,N          00042260
      A(I,J) = A(I,J)/TEMP          00042270
65    CONTINUE          00042280
70    CONTINUE          00042290
75    IF (IJOB .EQ. 1) GO TO 9005          00042300
      DO 103 K = 1,M          00042310
C     SOLVE UX = Y FOR X          00042320
      IW = 0          00042330
      DO 90 I = 1,N          00042340
      IMAX = WA(I)          00042350
      SUM = B(IMAX,K)          00042360
      B(IMAX,K) = B(I,K)          00042370
      IF (IW .EQ. 0) GO TO 85          00042380
      IM1 = I-1          00042390
      DO 80 J = IW,IM1          00042400
      SUM = SUM-A(I,J)*B(J,K)          00042410
80    CONTINUE          00042420
      GO TO 88          00042430
85    IF (T(1) .NE. ZERO .OR. T(2) .NE. ZERO) IW = I          00042440
88    B(I,K) = SUM          00042450
90    CONTINUE          00042460
C     SOLVE LY = B FOR Y          00042470
      N1 = N+1          00042480
      DO 100 IW = 1,N          00042490
      I = N1-IW          00042500
      JP1 = I+1          00042510
      SUM = B(I,K)          00042520
      IF (JP1 .GT. N) GO TO 98          00042530
                                         00042540
                                         00042550

```

```
DO 75 J = JP1,N          00042560  
    SUM = SUM-A(I,J)*B(J,K) 00042570  
95   CONTINUE             00042580  
98   B(I,K) = SUM/A(I,I)  00042590  
100  CONTINUE             00042600  
103  CONTINUE             00042610  
GO TO 9005               00042620  
C                         ALGORITHMIC SINGULARITY 00042630  
105 IER = 129             00042640  
9000 CONTINUE             00042650  
C                         PRINT ERROR .           00042660  
CALL UERTST(IER,6HLEQT1C) 00042670  
9005 RETURN              00042680  
END                      00042690
```

APPENDIX B
SAMCJ Program Listing

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00000010
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 00000020
CXX XXXX 00000030
CXX XXX 00000040
CXX XXX 00000050
CXX XXX 00000060
CXX XXX 00000070
CXX XXX 00000080
CXX XXX 00000090
CXX XXX 00000100
C SAMCJ COMPUTES THE LOAD DISTRIBUTION AMONG FASTENERS IN 00000110
C A MULTI-FASTENED COMPOSITE/ METALLIC JOINT, AND PREDICTS 00000120
C THE JOINT FAILURE LOAD, FAILURE MODE, AND FAILURE LOCATION. 00000130
C THE FASTENER LOAD DISTRIBUTION IS DETERMINED BY A 00000140
C FINITE ELEMENT METHOD WITH THE USE OF SPECIAL FINITE 00000150
C ELEMENTS. THE SUBSEQUENT FAILURE ANALYSIS IS BASED 00000160
C ON AN AVERAGE STRESS FAILURE CRITERION 00000170
C 00000180
IMPLICIT REAL*8(A-H,O-Z) 00000190
DIMENSION NPLY(2),WASHD(2),STM(3) 00000200
DIMENSION NEF(2),NLH(2),NOH(2),NPL(2) 00000210
DIMENSION NGEF(2,10,10),NGLH(2,10,10),NOOH(2,10,10) 00000220
DIMENSION NGPL(2,10,10),NUMEF(2,10),NUMLH(2,10) 00000230
DIMENSION NMNH(2,10),NUMPL(2,10) 00000240
DIMENSION NELORD(2,25,25),NELDIS(50,5,2) 00000250
DIMENSION ELLOAD(50,2),PSMX(50,4),NZERO(50),NBARY(25) 00000260
DIMENSION XOUT(600),YOUT(600),PLYK(100),BARK(100),BARU(100) 00000270
DIMENSION ELSIFF(50,10,10),ELSTSS(50,50,10) 00000280
DIMENSION OSSX(20),OJSN(20),ANR(200),RHS(200),PBC(200) 00000290
DIMENSION GLSTFF(200,200),ASQM(200,200),ANR2(200) 00000300
DIMENSION RDSTFF(50,2),WOHT(500),ERO(50) 00000310
DIMENSION NELPLS(2,50),NELPT(2,50,50) 00000320
DIMENSION ELTHK(50),NELCON(50,6),HELCNA(50,6) 00000330
DIMENSION GCUORD(150,2),PLYTHK(2,25),HELFAS(25,3) 00000340
DIMENSION NELF,A(25,3) 00000350
DIMENSION ELWDTH(50),NGRID(150),LYPN(50) 00000360
DIMENSION FSCD(50,3),HELTYP(50) 00000370
DIMENSION MTL(3,15),R(2) 00000380
DIMENSION ANGK(5,2),NMPLY(2),CM(2) 00000390
DIMENSION ANG(5,2),IPLY(100,2) 00000400
DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2),H(2) 00000410
DIMENSION STUL(1,4) 00000420
DIMENSION XC(5),YC(5) 00000430
DIMENSION AONT(2),AOBR(2),AOSE(2) 00000440
DIMENSION ELFFAIL(50,3) 00000450
COMMON/AOV/AONT,AOBR,AOSE 00000460
COMMON/GSKW/GJSX,LJSN 00000470
COMMON/NPLS/NELPL,LYPN 00000480
COMMON/XCYC/XC,YC 00000490
COMMON/NCH/NELCON,HELCNA,NF,DIS 00000500
COMMON/STM/STM,I,M 00000510
COMMON/SNX/PSMX 00000520
COMMON/GTH/GTH,STULT 00000530
COMMON/LAMP/ELFAIL 00000540
COMMON/FCT/FASE,FASV,FASD 00000550

```

```

COMMON/RT/R          00000560
COMMON/MFS/FSCD      00000570
COMMON/NTP/NELTYP     00000580
COMMON/NPT/NOPT2,NOPT6,NOPT7,NOPT8   00000590
COMMON/MOD/E1,E2,B12,V12,V21       00000600
COMMON/LYP/NPLY,NMPLY,ANG,IPLY     00000610
COMMON/ELP/AX,BX,NOUT,NSTS        00000620
COMMON/FCC/ELNDTH,ELTHK,ELLOAD    00000630
COMMON/NCST/NCASE,NTYPE          00000640
COMMON/DISP/AHR2                 00000650
COMMON/PBB/PLYK,BARK,BARU        00000660
COMMON/ELS/ELSTFF,ELSTSS         00000670
COMMON/CMT2/XOUT,YOUT           00000680
COMMON/SER/NT,NB                 00000690
DATA Y/'Y'                00000700
DATA CMC/'C'                00000710
                                         00000720
                                         00000730
                                         00000740
                                         00000750
                                         00000760
                                         00000770
                                         00000780
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. ',/
                                         00000790
                                         00000800
                                         00000810
                                         00000820
                                         00000830
                                         00000840
                                         00000850
WRITE(6,876)
900 FORMAT(' ENTER:           ',/
*'      1 FOR SLS (SINGLE LAP SHEAR),/; 00000860
*'      2 FOR DLS (DOUBLE LAP SHEAR),/;) 00000870
READ(5,*1) NSDLS            00000880
WRITE(6,911)                  00000890
                                         00000900
911 FORMAT(' ENTER:           ',/
*'      1 FOR STATIC TENSION ',/; 00000910
*'      2 FOR STATIC COMPRESSION',/) 00000920
                                         00000930
                                         00000940
READ(5,*1) LTNCM            00000950
106 FORMAT(*1)               00000960
380 CONTINUE                  00000970
DO 300 K=1,2                 00000980
IF(K.EQ.1) WRITE(6,912)      00000990
IF(K.EQ.2) WRITE(6,913)      00001000
912 FORMAT(' IS THE TOP PLATE A COMPOSITE OR A METAL?') 00001010
913 FORMAT(' IS THE BOTTOM PLATE A COMPOSITE OR A METAL?') 00001020
WRITE(6,914)                  00001030
914 FORMAT(' ENTER C OR M IN THE FIRST FIELD')
READ(5,106) CM(K)            00001040
WRITE(6,203)                  00001050
203 FORMAT(' INPUT MATERIAL DESCRIPTION OF THIS PLATE ',/
*'      EX: ASA/3501-6')
READ(5,204) (MTL(K,I),I=1,15) 00001060
204 FORMAT(15A4)              00001070
                                         00001080
300 CONTINUE                  00001090
                                         00001100
IF(LM(1).NE.CMC.OR.CM(2).NE.CMC) WRITE(6,754) 00001110
754 FORMAT('/',NOTE: FOR COMPUTATIONAL PURPOSES A ',/
*' METALLIC PLATE IS MODELED AS A 30 PLY ',/
*' LAMINATE OF 0 DEGREE PLYS WITH ISOTROPIC',/
*' MATERIAL PROPERTIES',/) 00001120
                                         00001130
                                         00001140
                                         00001150

```

```

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,X) 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,X) E1(K),E2(K)
WRITE(6,213)
218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO')
READ(5,X) 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 ',/
      X' ORIENTATIONS IN THE TOP PLATE')
702 FORMAT(' INPUT TOTAL NUMBER OF DISTINCT PLY ',/
      X' ORIENTATIONS IN THE BOTTOM PLATE')
READ(5,X) 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,X) ANG(L,K)
209 CONTINUE
WRITE(6,1823)
1823 FORMAT(/,' THICKNESS VARIATIONS MAY BE APPROXIMATED',/
      X' BY ASSIGNING DIFFERENT LAYUPS TO ELEMENTS',/
      X' IN A COMPOSITE PLATE OR BY SPECIFYING DIFFERENT',/
      X' THICKNESSES TO ELEMENTS IN A METALLIC PLATE',/
      X') IF(NSDLs.EQ.2) WRITE(6,789)
789 FORMAT(/,' NOTE: FOR THE DOUBLE LAP SHEAR CASE, FOR',/
      X' THE BOTTOM PLATE, ENTER ONLY HALF FOR THE ',/
      X' LAYUP FOR A COMPOSIT OR HALF THE THICKNESS ',/
      X' FOR A METALLIC',/)
DO 811 I=1,2
IF(CM(I).EQ.CMC) GO TO 891

```

```

NELPLS(I,1)=30          00001760
DO 892 III=1,30         00001770
892 NELPT(I,1,III)=1     00001780
GO TO 811                00001790
891 CONTINUE               00001800
IF(I.EQ.1) WRITE(6,812)    00001810
IF(I.EQ.2) WRITE(6,813)    00001820
812 FORMAT(/, ' ENTER NUMBER OF DIFFERENT LAYUPS IN THE ',/)
X' TOP PLATE')           00001830
813 FORMAT(/, ' ENTER NUMBER OF DIFFERENT LAYUPS IN THE ',/
X' BOTTOM PLATE')        00001840
READ(5,X) NL               00001850
DO 814 J=1,NL              00001860
WRITE(6,815) J             00001870
815 FORMAT(' ENTER NUMBER OF PLIES IN LAYUP NO ',IS)
READ(5,X) NELPLS(I,J)      00001880
WRITE(6,816)               00001890
816 FORMAT(' ENTER PLY THICKNESS FOR THIS LAYUP')
READ(5,X) PLYTHK(I,J)      00001900
NN=NELPLS(I,J)             00001910
WRITE(6,818)               00001920
818 FORMAT(' ENTER SEQUENCE OF PLY TYPES FROM TOP TO BOTTOM')
DO 817 K=1,NN              00001930
READ(5,X) NELPT(I,J,K)      00001940
817 CONTINUE                 00001950
814 CONTINUE                 00001960
811 CONTINUE                 00001970
WRITE(6,855)               00001980
855 FORMAT(/' FASTENER DESCRIPTION:',/)
WRITE(6,250)               00001990
250 FORMAT(' INPUT MATERIAL DESCRIPTION FOR FASTENER')
READ(5,251) (MTL(S,I),I=1,15) 00002000
251 FORMAT(15A4)             00002010
WRITE(6,252)               00002020
252 FORMAT(' INPUT YOUNG'S MODULUS AND POISSON'S RATIO FOR',/
X' THE FASTENER')
READ(5,X) FASE,FASV        00002030
WRITE(6,253)               00002040
253 FORMAT(' INPUT THE DIAMETER OF THE FASTENER')
READ(5,X) FASD             00002050
WRITE(6,888)               00002060
888 FORMAT(/' FASTENER TYPE
X'          1 FOR PROTRUDING HEAD',/
X'          2 FOR COUNTERSUNK HEAD')
READ(5,X) NFTYP            00002070
R(1)=1.0D10                00002080
R(2)=1.0D10                00002090
IF(NFTYP.EQ.1) GO TO 360   00002100
WRITE(6,889)               00002110
889 FORMAT(/' ENTER PLATE WHICH CONTAINS THE COUNTERSUNK',/
X' HEAD (OPPOSITE PLATE ASSUMES THE NUT HEAD)',/
X'          1 FOR TOP PLATE',/
X'          2 FOR BOTTOM PLATE ')
READ(5,X) N                00002120
R(N)=0.0D0                  00002130
360 CONTINUE                 00002140
WRITE(6,477)               00002150
477 FORMAT(/' GRID LAYOUT:',/)
C INPUT GRIDS, ELEMENT CONNECTIVITY AND PROPERTIES

```

```

C      TOP PLATE          00002360
      WRITE(6,689)          00002370
689  FORMAT(' ENTER NUMBER OF GRIDS IN TOP PLATE') 00002380
      READ(5,*) NGP1          00002390
      WRITE(6,371) NGP1          00002400
      371 FORMAT(//,' ENTER ',I8,' GRID POINTS'          00002410
      X'                                ','           00002420
      X' FORMAT! GRID ID, X AND Y COORDINATES ')          00002430
      DO 603 I=1,NGP1          00002440
      READ(5,*) NORID(I),GCOORD(I,1),GCOORD(I,2) 00002450
      603 CONTINUE          00002460
C      BOTTOM PLATE        00002470
      WRITE(6,633)          00002480
683  FORMAT(' ENTER NUMBER OF GRIDS IN BOTTOM PLATE') 00002490
      READ(5,*) NOP2          00002500
      NOTOT=NGP1+NGP2          00002510
      WRITE(6,371) NOP2          00002520
      NP1=NOP1+1          00002530
      DO 604 I=NP1,NOTOT 00002540
      READ(5,*) NORID(I),GCOORD(I,1),GCOORD(I,2) 00002550
      604 CONTINUE          00002560
      WRITE(6,883)          00002570
883  FORMAT(//,' ELEMENT DESCRIPTION:','')          00002580
      WRITE(6,399)          00002590
      399 FORMAT(//,'          00002600
      X' PLANAR ELEMENTS ARE NUMBERED          00002610
      X' CLOCKWISE AS SHOWN'          00002620
      X'          N2   N3          00002630
      X'          N5          00002640
      X'          N1   N4          00002650
      X'          00002660
      X' ELEMENT TYPES ARE DESIGNATED AS FOLLOWS:          00002670
      X'          00002680
      X'          4 NODE PLAIN ELEMENT      TYPE NO. 1 00002690
      X'          9 NODE LOADED HOLE ELEMENT  TYPE NO. 2 00002700
      X'          4 NODE OPEN HOLE ELEMENT   TYPE NO. 3 00002710
      X'          00002720
      X'          (NOTE: ENTER NS=0 FOR FOUR NODE ELEMENTS) 00002730
      WRITE(6,191)          00002740
191  FORMAT(' ENTER NUMBER OF ELEMENTS IN TOP PLATE') 00002750
      READ(5,*) NEL1          00002760
      DO 474 I=1,NEL1          00002770
      WRITE(6,388) I          00002780
      474 FORMAT(' FOR ELEMENT NO',I5,',' 00002790
      X' ENTER! ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE') 00002800
      READ(5,*) (NELCON(I,J),J=1,6),NELTYP(I) 00002810
      DO 591 IL=2,6          00002820
      IC=0          00002830
      NELCHA(I,1)=NELCON(I,1) 00002840
      DO 592 KL=1,NGP1          00002850
      IF(NELCON(I,IL).EQ.NGRID(KL)) IC=1 00002860
      IF(NELCON(I,IL).EQ.NGRID(KL)) NELCHA(I,IL)=KL 00002870
      IF(IC.EQ.1) GO TO 591 00002880
      592 CONTINUE          00002890
      591 CONTINUE          00002900

```

```

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,X) ATH
721 IF(NELTYP(I).NE.2) GO TO 722
WRITE(6,723)
723 FORMAT(' ENTER ELEMENT THICKNESS')
READ(5,X) ATH
FSCD(I,1)=GCOORD(NELCNA(I,6),1)
FSCD(I,2)=GCOORD(NELCNA(I,6),2)
FSCD(I,3)=FASD/2.000
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,X) ATH,(FSCD(I,J),J=1,3)
724 ELTHK(I)=ATH/30.000
PLYTHK(I,1)=ATH/30.000
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,X) LYPN(I)
726 IF(NELTYP(I).NE.2) GO TO 728
WRITE(6,729)
729 FORMAT(' ENTER ELEMENT LAYUP NO')
READ(5,X) LYPN(I)
FSCD(I,1)=GCOORD(NELCNA(I,6),1)
FSCD(I,2)=GCOORD(NELCNA(I,6),2)
FSCD(I,3)=FASD/2.000
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,X) LYPN(I),(FSCD(I,J),J=1,3)
730 ELTHK(I)=PLYTHK(I,LYPN(I))
474 CONTINUE
WRITE(6,688)
688 FORMAT('/',' ENTER NUMBER OF ELEMENTS IN BOTTOM PLATE ')
READ(5,X) NEL2
NELTOT=NEL1+NEL2
NP1=NEL1+1
DO 611 I=NP1,NELTOT
WRITE(6,600) I
800 FORMAT(' FOR ELEMENT NO',I5,
*' ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE')
READ(5,X) (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
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550

```

```

593 CONTINUE          00003560
IF(CM(2).EQ.CMC) GO TO 927 00003570
IF(NELTYP(I).NE.1) GO TO 921 00003580
WRITE(6,1921)               00003590
1921 FORMAT(' ENTER ELEMENT THICKNESS') 00003600
READ(5,X) ATH               00003610
921 IF(NELTYP(I).NE.2) GO TO 922 00003620
WRITE(6,1923)               00003630
1923 FORMAT(' ENTER ELEMENT THICKNESS') 00003640
READ(5,X) ATH               00003650
FSCD(I,1)=GCOORD(NELCNA(I,6),1) 00003660
FSCD(I,2)=GCOORD(NELCNA(I,6),2) 00003670
FSCD(I,3)=FASD/2.000 00003680
WRITE(6,3443) I,NELCNA(I,6),FSCD(I,1),FSCD(I,2) 00003690
3443 FORMAT(' C I NELCNA FSCD12',I5,2X,5(D9.3,2X)) 00003700
922 IF(NELTYP(I).NE.3) GO TO 924 00003710
WRITE(6,925)               00003720
925 FORMAT(' ENTER ELEMENT THICKNESS, X AND Y COORDINATES',//,
*' OF OPEN HOLE AND HOLE RADIUS') 00003730
READ(5,X) ATH,(FSCD(I,J),J=1,3) 00003740
924 ELTHK(I)=ATH/50.000 00003750
PLYTHK(2,1)=ATH/10.000 00003760
LYPH(I)=1 00003770
GO TO 611 00003780
927 CONTINUE          00003800
IF(NELTYP(I).NE.1) GO TO 926 00003810
WRITE(6,1927)               00003820
1927 FORMAT(' ENTER ELEMENT LAYUP NO') 00003830
READ(5,X) LYPN(I)           00003840
926 IF(NELTYP(I).NE.2) GO TO 928 00003850
WRITE(6,929)               00003860
929 FORMAT(' ENTER ELEMENT LAYUP NO') 00003870
READ(5,X) LYPN(I)           00003880
FSCD(I,1)=GCOORD(NELCNA(I,6),1) 00003890
FSCD(I,2)=GCOORD(NELCNA(I,6),2) 00003900
FSCD(I,3)=FASD/2.000 00003910
928 IF(NELTYP(I).NL.3) GO TO 930 00003920
WRITE(6,931)               00003930
931 FORMAT(' ENTER ELEMENT LAYUP NUMBER, X AND Y',//,
*' COORDINATES OF THE OPEN HOLE AND THE HOLE',//,
*' RADIUS') 00003940
READ(5,X) LYPN(I),(FSCD(I,J),J=1,3) 00003950
950 ELTHK(I)=PLYTHK(2,LYPN(I)) 00003960
611 CONTINUE          00003970
WRITE(6,1741)               00003980
1741 FORMAT(' FASTENERS ARE MODELED BY EFFECTIVE',//,
*' FASTENER ELEMENTS WHICH PROVIDE THE',//,
*' ELASTIC LINK BETWEEN THE TOP AND',//,
*' BOTTOM PLATES',//)
WRITE(6,1711)               00003990
1711 FORMAT(' ENTER NUMBER OF FASTENERS IN JOINT ') 00004000
READ(5,X) NUMF               00004010
WRITE(6,16)                 00004020
716 FORMAT('',//,
*' EFFECTIVE FASTENER ELEMENTS ARE
*' NUMBERED AS SHOWN:
*'      N1 (TOP PLATE) 00004030
*'      N2 (BOTTOM PLATE) 00004040
*'                                00004050
*'                                00004060
*'                                00004070
*'                                00004080
*'                                00004090
*'                                00004100
*'                                00004110
*'                                00004120
*'                                00004130
*'                                00004140
*'                                00004150

```

```

      X
      X WHERE N1 AND N2 CORRESPOND TO THE CENTRAL //,
      X NODES IN LOADED HOLE ELEMENTS           //
      X FORMAT: ELEMENT ID, N1, N2             //
      DO 717 I=1,NUMF                         //
      WRITE(6,711) I                           //
711  FORMAT(' ENTER ELEMENT NO',I5)
      READ(5,X) (NELFAS(I,J),J=1,3)
      717 CONTINUE

C DETERMINE GRID STOPAGE LOCATIONS FOR
C ELEMENT NUDES

      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 COMPUTE ELEMENT WIDTHS
C
      DO 239 I=1,NELTOT
      ELWDTH(I)=DABS(XCOORD(NELCNA(I,3),2)-XCOORD(NELCNA(I,2),2))
239  CONTINUE

```

```

C GROUP ELEMENTS TO AVOID THE DUPLICATE          00004760
C CALCULATION OF IDENTICAL STIFFNESS          00004770
C MATRICES                                     00004780
C
C WRITE(6,3000)                                00004790
C
3000 FORMAT(/,' TO REDUCE RUN TIMES, ELEMENTS MAY BE ',/
C   ' GROUPED INTO SETS WHICH WILL BE ASSIGNED',/,/
C   ' IDENTICAL STIFFNESS MATRICES',/,/
C   ' ENTER:    1 TO USE THIS OPTION           ',/,/
C   '           2 OTHERWISE      ')             00004800
C
C READ(S,X) NOPT                               00004810
C IF(NOPT.EQ.1) GO TO 3001                   00004820
C N1=0                                         00004830
C N2=0                                         00004840
C N3=0                                         00004850
C DO 3002 I=1,NEL1                           00004860
C IF(NELTYP(I).EQ.1) N1=N1+1                  00004870
C IF(NELTYP(I).EQ.2) N2=N2+1                  00004880
C IF(NELTYP(I).EQ.3) N3=N3+1                  00004890
C NEF(1)=NUMF                                 00004900
C NLH(1)=N2                                   00004910
C NOH(1)=N3                                   00004920
C NPL(1)=N1                                   00004930
C N=NUMF                                      00004940
C DO 3003 I=1,N                               00004950
C NOFF(1,I,1)=NELFAS(I,1)                     00004960
C IC=0                                         00004970
C DO 3004 I=1,NEL1                           00004980
C IF(NELTYP(I).EQ.2) IC=IC+1                 00004990
C IF(NELTYP(I).EQ.2) NGLH(1,IC,1)=NELCON(I,1) 00005000
C IF(NELTYP(I).EQ.2) NUMLH(1,IC)=1            00005010
C
3004 CONTINUE                                 00005020
C IC=0                                         00005030
C DO 3005 I=1,NEL1                           00005040
C IF(NELTYP(I).EQ.3) IC=IC+1                 00005050
C IF(NELTYP(I).EQ.3) NOOH(1,IC,1)=NELCON(I,1) 00005060
C IF(NELTYP(I).EQ.3) NUMOH(1,IC)=1            00005070
C
3005 CONTINUE                                 00005080
C IC=0                                         00005090
C DO 3006 I=1,NEL1                           00005100
C IF(NELTYP(I).EQ.1) IC=IC+1                 00005110
C IF(NELTYP(I).EQ.1) NGP(1,IC,1)=NELCON(I,1) 00005120
C IF(NELTYP(I).EQ.1) NUMPL(1,IC)=1            00005130
C
3006 CONTINUE                                 00005140
C N=NEL1+1                                  00005150
C N1=0                                         00005160
C N2=0                                         00005170
C N3=0                                         00005180
C DO 3007 I=N,NELTO1                         00005190
C IF(NELTYP(I).EQ.1) N1=N1+1                  00005200
C IF(NELTYP(I).EQ.2) N2=N2+1                  00005210
C IF(NELTYP(I).EQ.3) N3=N3+1                  00005220
C NEF(2)=NUMF                                 00005230
C NLH(2)=N2                                   00005240
C NOH(2)=N3                                   00005250
C NPL(2)=N1                                   00005260
C N=NUMF                                      00005270
C DO 3008 I=1,N                               00005280
C NOFF(2,I,1)=NELFAS(I,1)                     00005290
C IC=0                                         00005300
C
3008 CONTINUE                                 00005310
C IC=0                                         00005320
C
C

```

```

N=NEL1+1          00005360
DO 3010 I=N,NELTOT 00005370
IF(NELTYP(I).EQ.2) IC=IC+1 00005380
IF(NELTYP(I).EQ.2) NGLH(2,IC,1)=NELCON(I,1) 00005390
IF(NELTYP(I).EQ.2) NUMLH(2,IC)=1 00005400
3010 CONTINUE      00005410
IC=0             00005420
N=NEL1+1          00005430
DO 3011 I=N,NELTOT 00005440
IF(NELTYP(I).EQ.1) IC=IC+1 00005450
IF(NELTYP(I).EQ.1) NGPL(2,IC,1)=NELCON(I,1) 00005460
IF(NELTYP(I).EQ.1) NUMPL(2,IC)=1 00005470
3011 CONTINUE      00005480
IC=0             00005490
N=NEL1+1          00005500
DO 3012 I=N,NELTOT 00005510
IF(NELTYP(I).EQ.3) IC=IC+1 00005520
IF(NELTYP(I).EQ.3) NOOH(2,IC,1)=NELCON(I,1) 00005530
IF(NELTYP(I).EQ.3) NUMOH(2,IC)=1 00005540
3012 CONTINUE      00005550
GO TO 3013      00005560
3001 CONTINUE      00005570
WRITE(6,3015)    00005580
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,*1) NEF(1),NLH(1),NOH(1),NPL(1)
WRITE(6,3016)
3016 FORMAT(' GROUPING OF EFFECTIVE FASTENER ELEMENTS')
N=NEF(1)          00005650
DO 3017 I=1,N    00005660
WRITE(6,3018) I  00005670
3018 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8) 00005690
READ(5,*1) NUMEI(1,I)
N1=NUMEI(1,I)    00005700
WRITE(6,3019) N1  00005710
3019 FORMAT(' ENTER ',I8,' ELEMENT IDS')
RFID(5,*1) (NOEF(1,1,J),J=1,N1) 00005720
3017 CONTINUE      00005730
WRITE(6,3083)    00005740
3083 FORMAT(//,' GROUPING IN LOADED HOLE ELEMENTS')
N=NLH(1)          00005750
DO 3020 I=1,N    00005760
WRITE(6,3021) I  00005770
3021 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER ',I8) 00005780
READ(5,*1) NUMLH(1,1)
N1=NUMLH(1,1)    00005790
WRITE(6,3022) N1  00005800
3022 FORMAT(' INPUT',I8,' ELEMENT IDS')
READ(5,*1) (NOLH(1,1,J),J=1,N1) 00005810
3020 CONTINUE      00005820
IF(NOH(1).EQ.0) GO TO 4071 00005830
WRITE(6,3023)    00005840
3023 FORMAT(' GROUPING IN UNLOADED HOLE ELEMENTS')
N=NOH(1)          00005850
DO 3024 I=1,N    00005860
WRITE(6,3025) I  00005870
3025 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8) 00005880
READ(5,*1) NUMOH(1,1)

```

```

N1=NUMOH(1,I)
WRITE(6,3026) N1
3026 FORMAT(' ENTER',I8,' ELEMENT IDS')
READ(5,*)(NOH(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(1,1)
    N1=NUMPL(1,I)
    WRITE(6,3033) N1
3033 FORMAT(' ENTER',I8,' ELEMENT IDS')
    READ(5,*)(NGPL(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
        NOEF(2,I,J)=NOEF(1,I,J)
4019 NOEF(2,I,J)=NOEF(1,I,J)
4017 CONTINUE
    WRITE(6,4088)
4088 FORMAT(/, ' GROUPING OF LOADED HOLE ELEMENTS')
    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,*)(NOH(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,*)(NOH(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:')          00006560
N=NPL(2)                                              00006570
DO 4031 I=1,N                                         00006580
WRITE(6,4032) I                                         00006590
4032 FORMAT(' ENTER NUMBER OF ELEMENTS IN GROUP NUMBER',I8) 00006600
READ(5,*) NUMPL(2,I)                                 00006610
N1=NUMPL(2,I)                                         00006620
WRITE(6,4033) N1                                         00006630
4033 FORMAT(' ENTER',I8,' ELEMENT IDS')             00006640
READ(5,*) (NGLP(2,I,J),J=1,N1)                      00006650
4031 CONTINUE                                         00006660
4074 CONTINUE                                         00006670
3013 CONTINUE                                         00006680
WRITE(6,3737)                                         00006690
3737 FORMAT(/,' INPUT DATA FOR FAILURE ANALYSIS:',/) 00006700
DO 226 K=1,2                                         00006710
IF(CM(K).NE.CMC) GO TO 2226                         00006720
WRITE(6,532) K                                         00006730
532 FORMAT(' ENTER FIBER ULTIMATE STRAIN VALUES ',/
' IN PLATE NO ',I8,/,/
' EPSILON ULT IN COMPRESSION ',/,/
' EPSILON ULT IN TENSION ',/,/
' GAMMA ULT IN SHEAR ',/)                            00006740
READ(5,*) (SULT(I,K),I=1,3)                           00006750
GO TO 2227                                         00006760
2226 CONTINUE                                         00006770
WRITE(6,2229)                                         00006780
2229 FORMAT(' ENTER METALLIC STRENGTHS: ',/
' TENSILE STRENGTH ',/,/
' COMPRESSIVE STRENGTH ',/,/
' SHEAR STRENGTH')                                     00006790
READ(5,*) STM(1),STM(2),STM(3)                      00006800
2227 CONTINUE                                         00006810
WRITE(6,4054)                                         00006820
4054 FORMAT(' AN AVERAGE STRESS CRITERIA IS USED TO ',/
' PREDICT FAILURE. AU VALUES ARE REQUIRED AS ',/
' CHARACTERISTIC DISTANCES OVER WHICH STRESSES ',/
' ARE TO BE AVERAGED AND COMPARED TO UNNOTCHED ',/
' LAMINATE STRENGTHS TO PREDICT FAILURE',/)          00006830
WRITE(6,5432) K                                         00006840
5432 FORMAT(' ENTER AO VALUES FOR STRESS AVERAGING',/
' FOR EACH FAILURE MODE IN PLATE NO',I5,/,/
' AONT = NET SECTION ',/,/
' AOPR = BEARING ',/,/
' AOSO = SHEAROUT ')                                00006850
READ(5,*) AONT(K),AOPR(K),AOSO(K)                  00006860
226 CONTINUE                                         00006870
C CASE HEADING                                         00006880
C
143 FORMAT(//,10X,'PROGRAM SAMCJ',/)                 00006890
IF(NSDLS.EQ.1) WRITE(6,633)                           00006900
IF(NCDLS.EQ.2) WRITE(6,634)                           00006910
633 FORMAT(2X,'A SINGLE LAP SHEAR PANEL WILL BE ANALYZED',/) 00006920
634 FORMAT(2X,'A DOUBLE LAP SHEAR PANEL WILL BE ANALYZED',/) 00006930
IF(LTNCM.EQ.1) WRITE(6,823)                           00006940
IF(LTNCM.EQ.2) WRITE(6,824)                           00006950
823 FORMAT(2X,'LOADED IN STATIC TENSION',/)          00006960
824 FORMAT(2X,'LOADED IN STATIC COMPRESS ON',/)       00006970

```

```

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,1SA4,/)

HT=HEPLLS(I,1,XPLYTHK(I,1))
WRITE(6,691) E1(I),E2(I),G12(I),V12(I),V21(I)
691 FORMAT(2X,'MATERIAL PROPERTIES',/,/
  X10X,'E1 = ',D9.3,' PSI',/,/
  X10X,'E2 = ',D9.3,' PSI',/,/
  X10X,'G12 = ',D9.3,' PSI',/,/
  X10X,'NU12 = ',D9.3,/,
  X10X,'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,/)

478 FORMAT(2X,' DIAMETER = ',D9.3,' INCHES',/)
WRITE(6,609) FASE,FASV
609 FORMAT(2X,' MATERIAL PROPERTIES',/,/
  X10X,'E = ',D9.3,' PSI',/,/
  X10X,'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=1,1
WRITE(6,632) I
632 FORMAT(2X,'PLATE NUMBER',I5,/)
NP=NUMPLY(I)
IF(CM(I,'HE.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,'EP FIBR 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,'HEAR STRENGTH = ',D9.3,/)

3113 CONTINUE
WRITE(6,1543)
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   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          00007760
CCC HOLE REGIONS IN THE JOINT                         00007770
C INTERNAL APPLIED LOAD SET TO 1 KIP                  00007780
CCC APP=1000.0                                         00007790
IF(LTNCM.EQ.2) APP=-APP                            00007800
NELTOT=NELL+NEL2                                     00007810
NGTOT=NGP1+NGP2                                     00007820
00007830
00007840
00007850
00007860
00007870
00007880
00007890
00007900
00007910
00007920
00007930
00007940
00007950
00007960
00007970
00007980
00007990
00008000
00008010
00008020
00008030
00008040
00008050
00008060
00008070
00008080
00008090
00008100
00008110
00008120
00008130
00008140
00008150
00008160
00008170
00008180
00008190
00008200
00008210
00008220
00008230
00008240
00008250
00008260
00008270
00008280
00008290
00008300
00008310
00008320
00008330
00008340
00008350

C INITIALIZE ARRAYS
DO 1 I=1,50                                         00007880
DO 3 J=1,4                                         00007890
DO 3 K=1,4                                         00007900
3 ELSTFF(I,J,K)=0.                                 00007910
1 CONTINUE
DO 4 I=1,200                                       00007920
PBC(I)=0.                                         00007930
HMS(I)=0.                                         00007940
AMR(I)=0.                                         00007950
AHR2(I)=0.                                         00007960
DO 5 J=1,200                                       00007970
GLSTFF(I,J)=0.                                 00007980
ASQM(I,J)=0.                                         00007990
00008000
3 CONTINUE
4 CONTINUE

C CALCULATION OF EFFECTIVE FASTENER ELEMENT          00008010
STIFFNESS MATRICIES                               00008020
C WRITE(6,8418)
8418 FORMAT('PAUSE FOR STIFFNESS MATRIX CALCULATIONS',/)
NLOOP=NEF(1)                                       00008030
DO 444 I=1,NLOOP                                  00008040
NEL=NGEF(1,1,1)                                   00008050
DO 5001 II=1,NUMF                                00008060
5001 IF(NEL.EQ.NELFAS(II,1)) IEI=II             00008070
C SEARCH FOR LOADED HOLE ELEMENTS CONNECTED        00008080
TO FASTENER ELEMENT                               00008090
C NTOP=0                                           00008100
NBOT=0                                           00008110
DO 643 J=1,NEL1                                  00008120
643 IF(NELFAS(IEL,2).EQ.NELCON(J,6)) NTOP=J    00008130
NP1=NEL1+1                                         00008140
DO 446 J=NP1,NELTOT                            00008150
446 IF(NELFAS(IEL,3).EQ.NELCON(J,6)) NBOT=J    00008160
NPLY(1)=NELPLS(1,LYPN(NTOP))                   00008170
H(1)=ELTHK(NTOP)                                 00008180
DO 910 JJJ=1,50                                  00008190
910 IPLY(JJJ,1)=NELPT(1,LYPN(NTOP),JJJ)        00008200
NPLY(2)=NELPLS(2,LYPN(NBOT))                   00008210
H(2)=ELTHK(NBOT)                                 00008220
DO 113 JJJ=1,50                                  00008230
113 IPLY(JJJ,2)=NELPT(2,LYPN(NBOT),JJJ)        00008240
C INITIALIZE PARAMETERS FOR COLLOCATION           00008250
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=0.000
IF(L.EQ.2) PHI=90.00
DO 530 K=1,2
NTB=NTOP
IF(K.EQ.2) NTB=NBOT
C ELEMENT VERTEXES ARE INTERNALLY
C NUMBERED AS:
      3   2
      4   1
SFX=(GCOORD(NELCHA(NTB,5),1)+GCOORD(NELCHA(NTB,2),1))/2.000
SFY=(GCOORD(NELCHA(NTB,3),2)+GCOORD(NELCHA(NTB,2),2))/2.000
DO 128 JJ=1,4
XC(JJ)=GCOORD(NELCHA(NTB,6-JJ),1)-FSCD(NTB,1)
YC(JJ)=GCOORD(NELCHA(NTB,6-JJ),2)-FSCD(NTB,2)
128 CONTINUE
XC(5)=XC(1)
YC(5)=YC(1)
W=ELWDTN(NTB)
AST=1000.0
CALL POLY(W,AST,JK,K,NCLL,LTCM)
CALL CIRC(W,AST,JK,K,LTCM)
NOPT4=1
NCASE=1
HTYPE=HELTYPIEL)
CALL FIGEOM(H,PHI,K,NOPT4,NCLL)
CALL FBOLT(ANGK,H,PHI,K)
580 CONTINUE
H=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 CALCULATION OF FASTENER PROPERTIES
FASG=FASE/(2.0*(1+FASV))
FASLAM=5.0*(1.0+FASV)/(7.+6.*FASV)
FASR=FASD/2.
FASA=ACOS(-1.)*FASR**2
FASI=ACOS(-1.)*FASR**4/4.
FASSG=FASLAM*FASG*FASA
FASBS=FASE*FASI
P=1000.
CALL CENTD(H,FASSG,FASBS,P)
CALL SOLVE(H,P,U1,U2)
IF(L.EQ.2) GO TO 666

```

```

RDSTFFIEL,1)=DABS(P/(U1+U2))          00008960
GO TO 570                                00008970
666 RDSTFFIEL,2)=DABS(P/(U1+U2))          00008980
570 CONTINUE                                00008990
IF(NUMEF(1,I).EQ.1) GO TO 444            00009000
N=NUMEF(1,I)                                00009010
DO 5023 K=2,N                                00009020
DO 5024 L=1,NUMF                            00009030
5024 IF(NGEF(1,I,K).EQ.NELFAS(L,1)) IEL2=L  00009040
      RDSTFFIEL2,1)=RDSTFFIEL,1              00009050
      RDSTFFIEL2,2)=RDSTFFIEL,2              00009060
5023 CONTINUE                                00009070
444 CONTINUE                                00009080
8584 CONTINUE                                00009390
00009100
C   CALCULATION OF LOADED HOLE AND UNLOADED HOLE
C   ELEMENT STIFFNESS MATRICES               00009110
C                                         00009120
C                                         00009130
C   INITIALIZE GAUSSIAN QUADRATURE POINTS AND WEIGHTS 00009140
C                                         00009150
C                                         00009160
C                                         00009170
C                                         00009180
C                                         00009190
C                                         00009200
C                                         00009210
C                                         00009220
C                                         00009230
C                                         00009240
C                                         00009250
C                                         00009260
C                                         00009270
C                                         00009280
C                                         00009290
C                                         00009300
588 CONTINUE                                00009310
NAVD=10                                    00009320
DO 420 KJ=1,2                                00009330
ISLH=1                                     00009340
ISOH=1                                     00009350
ISPL=1                                     00009360
NLOOP=Nlh(KJ)+NOH(KJ)+NPL(KJ)           00009370
NCLH=0                                     00009380
NCOH=0                                     00009390
NCPL=0                                     00009400
DO 400 L=1,NLOOP                            00009410
IF(NCLH.EQ.Nlh(KJ)) ISLH=0                00009420
IF(NCLH.EQ.Nlh(KJ)) GO TO 6010            00009430
NCLH=NCLH+1                                00009440
IEL2=NGlh(KJ,NCLH,1)                      00009450
GO TO 6011                                00009460
6010 IF(NCOH.EQ.NOH(KJ)) ISOH=0            00009470
IF(NCOH.EQ.NOH(KJ)) GO TO 6020            00009480
NCOH=NCOH+1                                00009490
IEL2=NGOH(KJ,NCOH,1)                      00009500
GO TO 6011                                00009510
6020 IF(NCPL.EQ.NPL(KJ)) ISPL=0            00009520
IF(NCPL.EQ.NPL(KJ)) GO TO 400              00009530
NCPL=NCPL+1                                00009540
IEL2=NGPl(KJ,NCPL,1)                      00009550

```

```

6011 CONTINUE 00009560
DO 6030 KK=1,NELTOT 00009570
6030 IFIEL2,EQ.NELCON(KK,1)) IEL=KK 00009580
H(KJ)=ELTHKIEL) 00009590
NPLY(KJ)=NELPLS(KJ,LYPNIEL)) 00009600
DO 919 JJJ=1,50 00009610
IPLY(JJJ,KJ)=NELPT(KJ,LYPNIEL),JJJ) 00009620
919 CONTINUE 00009630
NRNK=5 00009640
IF(NELTYPIEL),EQ.2) NRNK=7 00009650
C 00009660
C INTERNAL NUMBERING OF ELEMENT VERTICES: 00009670
C 00009680
C C 00009690
C C 00009700
C C 00009710
C C 00009720
C C 00009730
SFX=(GCOORD(NELCNAIEL,5),1)+GCOORD(NELCNAIEL,2),1))/2.000 00009740
SFY=(GCOORD(NELCNAIEL,3),2)+GCOORD(NELCNAIEL,2),2))/2.000 00009750
DO 440 K=1,4 00009760
XC(K)=GCOORD(NELCNAIEL,6-K),1)-FSCDIEL,1) 00009770
IF(NELTYPIEL),EQ.1) XC(K)=GCOORD(NELCNAIEL,6-K),1)-SFX 00009780
YC(K)=GCOORD(NELCNAIEL,6-K),2)-FSCDIEL,2) 00009790
IF(NELTYPIEL),EQ.1) YC(K)=GCOORD(NELCNAIEL,6-K),2)-SFY 00009800
440 CONTINUE 00009810
XC(3)=XC(1) 00009820
YC(5)=YC(1) 00009830
AX=FSCDIEL,3) 00009840
IF(NELTYPIEL),EQ.1) AX=0.1 00009850
BX=AX 00009860
PI=DARCCOS(-1.0D0) 00009870
RAD=PI/180.D0 00009880
NOAUSS=2*NGP 00009890
NGPT=4*NOAUSS*2 00009900
IC=0 00009910
NCPT=2*NP 00009920
C 00009930
C DETERMINE COORDINATES AT WHICH STRESSES AND 00009940
C DISPLACEMENTS ARE TO BE COMPUTED. 00009950
C ELEMENT NATURAL FLEXIBILITY MATRICES 00009960
C ARE COMPUTED BY INTEGRATING STRESSES 00009970
C FOR EACH LOAD CASE IN THE NATURAL 00009980
C MODE METHOD. THE ELEMENTS ARE DIVIDED 00009990
C INTO FOUR REGIONS AND THE GAUSSIAN POINTS 00010000
C ARE SCALED TO EACH REGION SIZE 00010010
C C 00010020
C C 00010030
C C 00010040
C C 00010050
C C 00010060
C C 00010070
C C 00010080
C C 00010090
C C 00010100
C C 00010110
C 15 CONTINUE 00010120
C C 00010130
C C 00010140
C C 00010150

```

```

DO 16 II=1,NCPT          00010160
DO 16 JJ=1,NCPT          00010170
IC=IC+1                  00010180
XOUT(IC)=AX*GSSX(II)     00010190
YI=DSQRT(AX**2-XOUT(IC)**2) 00010200
YOUT(IC)=((YC(2)-YI)/2.)*GSSX(JJ)+(YC(2)+YI)/2. 00010210
WGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(2)-YI)*MAX/2.0D0 00010220
16 CONTINUE               00010230
C
CCC REGION 3              00010240
C
DO 17 II=1,NCPT          00010250
DO 17 JJ=1,NCPT          00010260
IC=IC+1                  00010270
XOUT(IC)=AX*GSSX(II)     00010280
YI=-DSQRT(AX**2-XOUT(IC)**2) 00010290
YOUT(IC)=((YI-YC(1))/2.)*GSSX(JJ)+(YI+YC(1))/2. 00010300
WGHT(IC)=GSSW(II)*GSSW(JJ)*(YI-YC(1))*MAX/2.0D0 00010310
00010320
17 CONTINUE               00010330
C
CCC REGION 4              00010340
C
DO 18 II=1,NCPT          00010350
DO 18 JJ=1,NCPT          00010360
IC=IC+1                  00010370
XOUT(IC)=((XC(1)-AX)/2.)*GSSX(II)+(XC(1)+AX)/2. 00010380
YOUT(IC)=((YC(2)-YC(1))/2.)*GSSX(JJ)+(YC(2)+YC(1))/2. 00010390
WGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(2)-YC(1))*(XC(1)-AX)/4.0D0 00010400
00010410
18 CONTINUE               00010420
NINT=IC                  00010430
N=4*(NCPT**2)             00010440
00010450
C
CCC ADD COORDINATES ALONG WHICH STRESSES WILL 00010460
BE AVERAGED               00010470
C
ANT=AONT(KJ)              00010480
ABR=AOPR(KJ)              00010490
ASO=AOSI(KJ)              00010500
SG=1.0                     00010510
IF(LTNCM.EQ.2) SG=-1.0    00010520
IF(KJ.EQ.2) SG=-SG        00010530
00010540
C
CCC NET SECTION            00010550
C
ANDO=ANT/FLOAT(NAVD)      00010560
DO 21 II=1,NAVD           00010570
IC=IC+1                  00010580
XOUT(IC)=0.0D0             00010590
YOUT(IC)=BX+ANDO/2.+(II-1)*ANDO 00010600
00010610
21 CONTINUE               00010620
C
CCC SHEAROUT                00010630
C
ANSO=ASO/FLOAT(NAVD)      00010640
DO 31 II=1,NAVD           00010650
IC=IC+1                  00010660
XOUT(IC)=SG*(BX+ANSO/2.+(II-1)*ANSO) 00010670
YOUT(IC)=BX               00010680
00010690
31 CONTINUE               00010700
C
00010710
00010720
00010730
00010740
00010750

```

```

C BEARING 00010760
C
C ANBR=ABR/FLOAT(NAVD) 00010770
DO 41 II=1,NAVD 00010780
IC=IC+1 00010790
XOUT(IC)=30*(AX+ANBR/2.+(II-1)*ANBR) 00010800
YOUT(IC)=0. 00010810
00010820
00010830
41 CONTINUE 00010840
C ADD COORDINATES ALONG WHICH ELEMENT LOAD 00010850
C RECOVERY WILL BE COMPUTED 00010860
C
C DO 3332 III=1,10 00010870
IC=IC+1 00010880
IF(KJ.EQ.1) XOUT(IC)=XC(3)+0.1*AX 00010890
IF(KJ.EQ.2) XOUT(IC)=XC(1)-0.1*AX 00010900
YOUT(IC)=((YC(2)-YC(1))/2.000)*XOSSX(III)+(YC(2)+YC(1))/2.000 00010910
00010920
00010930
C STRESSES ARE SINGULAR AT THETA = 180 DEG OR Y = 0 00010940
C
C IF(DABS(YOUT(IC)).LT.0.01) YOUT(IC)=YOUT(IC-1) 00010950
3332 CONTINUE 00010960
4891 CONTINUE 00010970
NSTS=4*NAVD 00010980
NOUT=4*(NGAUSS**2) 00010990
00011000
C CALCULATION OF LOADED HOLE, UNLOADED HOLE, AND 00011010
PLAIN ELEMENT STIFFNESS MATRICIES 00011020
C
C THH=DARCOS(-1.0D0)/FLOAT(NOP) 00011030
NHIEL 00011040
DO 410 J=1,NRNK 00011050
NOPT4=5 00011060
NT=7 00011070
NCLL=10 00011080
NB=52+4*NCLL 00011090
HT=H(KJ)*NPLY(KJ) 00011100
NCASE=J 00011110
NTYPE=NELTYP(IEL) 00011120
CALL MOEQ(HT,H,AST,J,NN,KJ,NEL,NCLL) 00011130
CALL MCIR(H,AST,NN,J,NCLL) 00011140
PHI=0.0D0 00011150
CALL AMATRX(H,PHI,KJ) 00011160
CALL FIGEOM(H,PHI,KJ,NOPT4,NCLL) 00011170
CALL INFLNC(WHT,H,NRNK,J,KJ,NN,NOPT) 00011180
410 CONTINUE 00011190
4000 CONTINUE 00011200
00011210
00011220
00011230
C COMPUTE ELEMENT FAILURE VALUES BASED 00011240
ON MAXIMUM FIBER STRAIN ALLOWABLES 00011250
C
C HT=H(KJ)*NPLY(KJ) 00011260
IF(NELTYP(IEL).EQ.2) CALL SMAX(HT,KJ,IEL) 00011270
IF(NELTYP(IEL).EQ.3) CALL SMAX(HT,KJ,IEL) 00011280
IF(ISLH.EQ.0) GO TO 6040 00011290
NL=NUMLH(KJ,NCLL) 00011300
00011310
IF(NL.EQ.1) GO TO 400 00011320
DO 6041 K=2,NL 00011330
DO 6042 LL=1,NELTOT 00011340
6042 IF(NOLH(KJ,NCLH,K).EQ.NELCON(LL,1)) IEL2=LL 00011350

```

```

DO 6045 ILM=1,10          00011360
DO 6043 ILK=1,10          00011370
6043 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK) 00011380
DO 6044 KK=1,4             00011390
6044 PSMX(IEL2,KK)=PSMX(IEL,KK) 00011400
NNN=4XNAVD                00011410
DO 6045 ILM=1,NNN          00011420
DO 6043 ILK=1,10          00011430
6045 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK) 00011440
5091 CONTINUE                00011450
GO TO 400                  00011460
6040 IF(ISOH,EQ.0) GO TO 6046 00011470
NL=NUMOH(KJ,NCOH)          00011480
IF(NL.EQ.1) GO TO 400      00011490
DO 6047 K=2,NL              00011500
DO 6048 LL=1,NELTOT        00011510
6048 IF(NOOH(KJ,NCOH,K).EQ.NELCON(LL,1)) IEL2=LL 00011520
DO 6049 ILM=1,10            00011530
DO 6049 ILK=1,10            00011540
6049 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK) 00011550
DO 6050 KK=1,4             00011560
6050 PSMX(IEL2,KK)=PSMX(IEL,KK) 00011570
NNN=4XNAVD                00011580
DO 6051 ILM=1,NNN          00011590
DO 6051 ILK=1,10          00011600
6051 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK) 00011610
6047 CONTINUE                00011620
GO TO 400                  00011630
6046 IF(ISPL,EQ.0) GO TO 400 00011640
NL=NUMPL(KJ,NCPL)          00011650
IF(NL.EQ.1) GO TO 400      00011660
DO 6053 K=2,NL              00011670
DO 6054 LL=1,NELTOT        00011680
6054 IF(NOP1(KJ,NCPL,K).EQ.NELCON(LL,1)) IEL2=LL 00011690
DO 6055 ILM=1,10            00011700
DO 6055 ILK=1,10            00011710
6055 ELSTFF(IEL2,ILM,ILK)=ELSTFF(IEL,ILM,ILK) 00011720
DO 6056 KK=1,4             00011730
6056 PSMX(IEL2,KK)=PSMX(IEL,KK) 00011740
NNN=4XNAVD                00011750
DO 6057 ILM=1,NNN          00011760
DO 6057 ILK=1,10          00011770
6057 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK) 00011780
6053 CONTINUE                00011790
400 CONTINUE                  00011800
420 CONTINUE                  00011810
00011820
C DETERMINE ELEMENT ARRANGEMENT IN TOP
C AND BOTTOM PLATES          00011830
C                               00011840
C                               00011850
DO 681 KJ=1,2               00011860
IF(KJ.EQ.2) GO TO 501       00011870
L1=1                         00011880
L2=NOP1                      00011890
L3=1                         00011900
L4=NELL                      00011910
GO TO 502                   00011920
501 L1=NOP1+1                 00011930
L2=NGTOT                     00011940
L3=NELL+1)                   00011950

```

```

L4=NELTOT          00011960
502 CONTINUE       00011970
      AXMIN=1.D10   00011980
      AYMIN=1.D10   00011990
      DO 503 I=L1,L2 00012000
      IF(AXMIN.GT.GCOORD(I,1)) AXMIN=GCOORD(I,1) 00012010
      IF(AYMIN.GT.GCOORD(I,2)) AYMIN=CCORD(I,2) 00012020
      IF(AXMIN.EQ.GCOORD(I,1).AND.AYMIN.EQ.GCOORD(I,2)) NC=I 00012030
      00012040
503 CONTINUE       00012050
      DO 574 I=L3,L4 00012060
574 IF(NELCON(I,2).EQ.NORID(NC)) IEL=I 00012070
      NELORD(KJ,1,I)=IEL 00012080
      DO 504 I=1,25 00012090
      DO 505 J=1,25 00012100
      IFL=0 00012110
      DO 506 K=L3,L4 00012120
506 IF(NELCON(K,2).EQ.NELCON(NELORD(KJ,J,I),3)) IEL=K 00012130
      IF(IEL.EQ.0) GO TO 507 00012140
      NELORD(KJ,J+1,I)=IEL 00012150
505 CONTINUE       00012160
507 CONTINUE       00012170
      IF(KJ.EQ.1) NROW1=J 00012180
      IF(KJ.EQ.2) NROW2=J 00012190
      IEL=0 00012200
      DO 308 L=L3,L4 00012210
508 IF(NELCON(NELORD(KJ,I,I),5).EQ.NELCON( 00012220
      M,L)) IEL=L 00012230
      IF(IEL.EQ.0) GO TO 507 00012240
      NELORD(KJ,I,I+1)=IEL 00012250
504 CONTINUE       00012260
509 CONTINUE       00012270
      IF(KJ.EQ.1) NCOL1=I 00012280
      IF(KJ.EQ.2) NCOL2=I 00012290
681 CONTINUE       00012300
C
C COMPUTE NODAL DEGREES OF FREEDOM 00012310
C
      IC=0 00012320
      DO 540 KJ=1,2 00012330
      IF(KJ.EQ.1) NR=NROW1 00012340
      IF(KJ.EQ.1) NC=NCOL1 00012350
      IF(KJ.EQ.2) NR=NROW2 00012360
      IF(KJ.EQ.2) NC=NCOL2 00012370
      NELDIS(NELORD(KJ,1,1),1,1)=IC+1 00012380
      NELDIS(NELORD(KJ,1,1),1,2)=IC+2 00012390
      NELDIS(NELORD(KJ,1,1),2,1)=IC+3 00012400
      NELDIS(NELORD(KJ,1,1),2,2)=IC+4 00012410
      IC=IC+4 00012420
      IF(NR.EQ.1) GO TO 549 00012430
      DO 541 I=2,NR 00012440
      NELDIS(NELORD(KJ,I,1),1,1)=NELDIS(NELORD(KJ,I-1 00012450
      ,1),2,1) 00012460
      NELDIS(NELORD(KJ,I,1),1,2)=NELDIS(NELORD(KJ,I-1 00012470
      ,1),2,2) 00012480
      NELDIS(NELORD(KJ,I,1),2,1)=IC+1 00012490
      NELDIS(NELORD(KJ,I,1),2,2)=IC+2 00012500
      IC=IC+2 00012510
541 CONTINUE       00012520
549 CONTINUE       00012530
      DO 542 I=1,NC 00012540
                                         00012550

```

```

DO 543 J=1,NR          00012560
IF(I.EQ.1) GO TO 544      00012570
NELDIS(NELORD(KJ,J,I),1,1)=NELDIS(NELORD(
  KJ,J,I-1),4,1)      00012580
NELDIS(NELORD(KJ,J,I),1,2)=NELDIS(NELORD(
  KJ,J,I-1),4,2)      00012600
NELDIS(NELORD(KJ,J,I),2,1)=NELDIS(NELORD(
  KJ,J,I-1),3,1)      00012610
NELDIS(NELORD(KJ,J,I),2,2)=NELDIS(NELORD(
  KJ,J,I-1),3,2)      00012620
      00012630
      00012640
      00012650
      00012660
      00012670
544 CONTINUE             00012680
IF(J.EQ.1) GO TO 561      00012690
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)
      00012700
GO TO 562                00012710
561 CONTINUE             00012720
NELDIS(NELORD(KJ,J,I),4,1)=IC+1      00012730
NELDIS(NELORD(KJ,J,I),4,2)=IC+2      00012740
IC=IC+2                  00012750
      00012760
562 CONTINUE             00012770
IF(NELTYP(NELORD(KJ,J,I)),NE.2) GO TO 545      00012780
NELDIS(NELORD(KJ,J,I),5,1)=JC+1      00012790
NELDIS(NELORD(KJ,J,I),5,2)=IC+2
      IC=IC+2
      00012800
545 CONTINUE             00012810
NELDIS(NELORD(KJ,J,I),3,1)=IC+1      00012820
NELDIS(NELORD(KJ,J,I),3,2)=IC+2      00012830
IC=IC+2                  00012840
543 CONTINUE             00012850
542 CONTINUE             00012860
540 CONTINUE             00012870
C
C DETERMINE BOUNDARY NODES AND VALUES
C
NND=2*(NGP1+NGP2)          00012880
DO 165 I=1,100              00012890
  165 PBC(I)=0.              00012900
C
C DISTRIBUTE APPLIED LOAD
C
ATOT=GCOORD(NELCNA(NELORD(1,NROW1,1),3),2)      00012910
X-GCOORD(NELCNA(NELORD(1,1,1),2),2)      00012920
APL=APP/ATOT                  00012930
SG=1.0
IF(LTMCM.EQ.1) SG=-1.0      00012940
      00012950
DO 178 I=1,NROW1            00012960
  AI=GCOORD(NELCNA(NELORD(1,I,1),3),2)      00012970
  X-GCOORD(NELCNA(NELORD(1,I,1),2),2)      00012980
  M1=NELDIS(NELORD(1,I,1),1,1)      00012990
  M2=NELDIS(NELORD(1,I,1),2,1)      00013000
  PBC(M1)=PBC(M1)+SG*(0.5*DABS(APL*AI))
  PBC(M2)=PBC(M2)+SG*(0.5*DABS(APL*AI))
  178 CONTINUE                 00013010
  1119 CONTINUE               00013020
C
C ASSEMBLE GLOBAL STIFFNESS MATRIX
C
DO 220 N1=1,NELTOT          00013030
  IR=5
  IF(NELTYP(N1).NE.2) IR=4      00013040
      00013140
      00013150

```

```

C CCC
      TOP AND BOTTOM PLATE LOADED HOLE AND
      UNLOADED HOLE ELEMENTS
      IC1=0          00013160
      DO 425 N2=1,IR 00013170
      DO 425 N3=1,2 00013180
      M1=NELDIS(N1,N2,N3) 00013190
      IC1=IC1+1      00013200
      IC2=0          00013210
      DO 425 N4=1,IR 00013220
      DO 425 N5=1,2 00013230
      M2=NELDIS(N1,N4,N5) 00013240
      IC2=IC2+1      00013250
      GLSTFF(M1,M2)=GLSTFF(M1,M2)+ELSTFF(N1,IC1,IC2)
      425 CONTINUE    00013260
      229 CONTINUE    00013270
C CCC
      ADD EFFECTIVE FASTENER ELEMENTS
      00013280
      00013290
      00013300
      00013310
      00013320
      00013330
      00013340
      00013350
      00013360
      00013370
      00013380
      00013390
      00013400
      00013410
      00013420
      00013430
      00013440
      00013450
      00013460
      00013470
      00013480
      00013490
      00013500
      00013510
      00013520
      00013530
      00013540
      00013550
      00013560
      00013570
      00013580
      00013590
      00013600
      00013610
      00013620
      00013630
      00013640
      00013650
      00013660
      00013670
      00013680
      00013690
      00013700
      00013710
      00013720
      00013730
      00013740
      00013750
C CCC
      GLOBAL BOUNDARY CONDITIONS
      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),3,1)
      437 CONTINUE
      IC=IC+1
      NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),4,2)
      NUMZ=IC
C CCC
      RESTORE REDUCED STIFFNESS MATRIX
      IC=0

```

```

DO 655 I=1,NP          00013760
DO 665 K=1,NUMZ        00013770
IF(I.EQ.NZERO(K)) GO TO 655    00013780
665 CONTINUE           00013790
ICR=ICR+1              00013800
RHS(ICR)=RHS(I)        00013810
ICC=0                  00013820
DO 670 J=1,NP          00013830
DO 680 K=1,NUMZ        00013840
IF(J.EQ.NZERO(K)) GO TO 670    00013850
670 CONTINUE           00013860
ICC=ICC+1              00013870
ASQM(ICR,ICC)=OLSTFF(I,J)    00013880
680 CONTINUE           00013890
655 CONTINUE           00013900
NP=NP-NUMZ             00013910
685 CONTINUE           00013920
DO 695 I=1,NP          00013930
DO 695 J=1,NP          00013940
695 OLSTFF(I,J)=ASQM(I,J)    00013950
C
C      APPLYING QUASSIAN ELIMINATION TO THE    00013960
C      MATRIX OF COEFFICIENTS                 00013970
C
C      DO 2001 I=1,NP          00013980
IR=I
2042 IF(DABS(ASQM(IR,I)).GT.1.0D-10) GO TO 2041    00013990
IR=IR+1
IF(IR.GT.NP) GO TO 2001    00014000
GO TO 2042
2041 NN=IR+1
DO 2002 L=NN,np          00014010
IF(DABS(ASQM(L,I)).GT.1.D-10) GO TO 2009    00014020
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          00014030
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      BACK SUBSTITUTION
C
DO 2011 I=1,NP          00014230
L=NP+1-I
SUM=0.
IF(ASQM(L,L).EQ.0.) GO TO 2112    00014240
N=L+1
IF(N.GT.NP) GO TO 2013    00014250
DO 2013 J=N,np          00014260
SUM=SUM-ASQM(L,J)*ANR(J)
2013 CONTINUE
ANR(L)=(RHS(L)+SUM)/ASQM(L,L)    00014270

```

```

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,HELTOT
      NID=NELCON(K,1)
      WRITE(6,3947) NID
8947 FORMAT(/,' ELEMENT ID',I8,/,/
      X6X,'GRID',9X,'FX',9X,'FY',/)
      IR=5
      KL=K
      IF(KL.GT.NEL) KL=K-NEL
      IF(NELTYP(K).NE.21) 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*NJ-1))*ANR2(N1)+*
      *ELSTFF(K,N,(2*NJ))*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*NJ-1))*ANR2(N1)+*
      *ELSTFF(K,N,(2*NJ))*ANR2(N2)
530  CONTINUE
C
C      STORE ELEMENT LOADS FOR CHECK ON ELEMENT
      LOAD RECOVERY
      IF(K.LE.NEL1.AND.(I.EQ.1.OR.I.EQ.2)) ELOAD(K,I)=SUMU
      IF(K.GT.NEL1.AND.(I.EQ.3.OR.I.EQ.4)) ELOAD(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
      CRITICAL ELEMENT TO CALCULATE JOINT FAILURE
      LOAD

```

```

C
      CALL FCRIT(APP,NEL1,NEL2,NDAM,IN,LTNCM,NAVD)          00014960
      FAILV=DABS(ELFAIL(IN,NDAM))                          00014970
      IF(NSDLS.EQ.2) FAILV=2.*FAILV                         00014980
      NID=NELCON(IN,.)                                     00014990
      WRITE(6,5555) NID,FAILV                            00015000
      5555 FORMAT(//,' FAILURE IS PREDICTED TO OCCUR IN ELEMENT',//,
      X' NUMBER',I3,' AT AN APPLIED JOINT LOAD VALUE ',/,
      X' OF ',D14.7,' LBS',//)
      IF(NDAM.EQ.1) WRITE(6,5556)                         00015010
      IF(NDAM.EQ.2) WRITE(6,5557)                         00015020
      IF(NDAM.EQ.3) WRITE(6,5558)                         00015030
      5556 FORMAT(' THE PREDICTED FAILURE MODE IS NET SECTION') 00015040
      5557 FORMAT(' THE PREDICTED FAILURE MODE IS SHEAR-OUT ') 00015050
      5558 FORMAT(' THE PREDICTED FAILURE MODE IS BEARING')   00015060
      STOP
      END

CCC
      SUBROUTINE MOED(HT,H,AST,J,IN,KJ,NEL,NCL)           00015070
      C
      IMPLICIT REALN8(A-H,O-Z)                           00015080
      DIMENSION A1A(4),A2A(4)                           00015090
      DIMENSION XB(200),YB(200),A1(200),A2(200)        00015100
      DIMENSION NELTYP(50),HTA(200)                      00015110
      DIMENSION XC(5),YC(5)                           00015120
      COMMON/CMT1/XB,YB,A1,A2,HTA                      00015130
      COMMON/XC/YC                                      00015140
      COMMON/NTP/NELTYP                                00015150
      C
      DETERMINE EXTERIOR COLLOCATION POINTS AND          00015160
      STRESS BOUNDARY CONDITIONS CORRESPONDING          00015170
      TO THE NATURAL LOAD CASES                         00015180
      C
      NCS=3
      IF(NELTYP(IN).NE.2) NCS=3                         00015190
      JK=0
      DO 15 I=1,4
      A1A(I)=0.                                         00015200
      15 A2A(I)=0.
      A=(YC(2)-YC(1))*HT                               00015210
      B=(XC(2)-XC(1))*HT                               00015220
      IF(J.EQ.1) A1A(3)=1.0D0/A                         00015230
      IF(J.EQ.1.AND.NELTYP(IN).NE.2) A1A(1)=1.0D0/A    00015240
      IF(J.EQ.2) A1A(2)=1.0D0/B                         00015250
      IF(J.EQ.2.AND.NELTYP(IN).NE.2) GO TO 55          00015260
      IF(J.GT.2.AND.NELTYP(IN).NE.2) GO TO 55          00015270
      IF(J.EQ.3) A1A(1)=1.0D0/A                         00015280
      IF(J.EQ.4) A1A(4)=1.0D0/B                         00015290
      IF(J.EQ.1.OR.J.EQ.3) AST=1.0D0/A                 00015300
      IF(J.EQ.2.OR.J.EQ.4) AST=1.0D0/B                 00015310
      55 CONTINUE
      H=XC(3)-XC(2)                                     00015320
      IF(J.EQ.1.OR.J.EQ.3.OR.J.EQ.5) H=YC(2)-YC(1)     00015330
      DO 10 I=1,4
      X=XC(I)-XC(I+1)                                 00015340
      Y=YC(I+1)-YC(I)                                 00015350
      IF(X.EQ.0.) X=1.D-6                             00015360
      IF(Y.EQ.0.) Y=1.D-6                             00015370
      10
      00015380
      00015390
      00015400
      00015410
      00015420
      00015430
      00015440
      00015450
      00015460
      00015470
      00015480
      00015490
      00015500
      00015510
      00015520
      00015530
      00015540
      00015550

```

```

TH=DATANZ(X,Y)
TH=TH+180./PIARCOS(-0.1D1)          00015560
DX=(XC(I+1)-XC(I))/(NCL+1)          00015570
DY=(YC(I+1)-YC(I))/(NCL+1)          00015580
DO 20 II=1,NCL                      00015590
JK=JK+1
IF(I.EQ.1.OR.I.EQ.3) GO TO 23      00015600
YB(JK)=YC(I)
XB(JK)=XC(I)+DX*(II+.5)           00015610
IF(II.EQ.1) XB(JK)=XC(I)+(DX/2.)   00015620
GO TO 24                           00015630
23 YB(JK)=YC(I)+DY*(II+.5)         00015640
IF(II.EQ.1) YB(JK)=YC(I)+(DY/2.)   00015650
XB(JK)=XC(I)                      00015660
24 THTA(JK)=TH                     00015700
A1(JK)=A1(A(I))                   00015710
A2(JK)=A2(A(I))                   00015720
IF(J.EQ.NCS.AND.(I.EQ.1.OR.I.EQ.3)) A1(JK)=(2*YB(JK)/W)*(3.0D0/A) 00015730
IF(J.EQ.(NCS+1).AND.(I.EQ.2.OR.I.EQ.4)) A1(JK)=(2*XB(JK)/W)*(3.0 00015740
*DU/3)
IF(J.EQ.(NCS+2).AND.(I.EQ.1.OR.I.EQ.3)) A2(JK)=2.0/DSQRT(AXX2+BXX2) 00015750
*)                               00015770
IF(J.EQ.(NCS+2).AND.(I.EQ.2.OR.I.EQ.4)) A2(JK)=-2.0/DSQRT(AXX2+BXX2) 00015730
*)                               00015790
CONTINUE                         00015800
CONTINUE                         00015810
RETURN                           00015820
END                             00015830
CCC
SUBROUTINE MCIR(W,AST,I,J,NCL)      00015840
IMPLICIT REAL*8(A-H,O-Z)            00015850
DIMENSION XB(200),YB(200),A1(200),A2(200) 00015860
DIMENSION THTA(200),NELTYP(50)       00015870
COMMON/CNT1/XB,YB,A1,A2,THTA        00015880
COMMON/NTP/NELTYP                  00015890
CCM/MO/ELP/AX,BX
CCM=-1.0D0
RAD=DARCUS(CON)/180.
BSTR=DABS((2.*W*AST)/(DARCUS(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
CCCC
DETERMINE INTERIOR COLLOCATION POINTS AND
STRESS BOUNDARY CONDITIONS          00016000
NBD=52
NBI=NBD/4
DO 10 K=1,4                        00016010
CR=(-1)*DARCUS(CON)/2.
DO 20 KI=1,NBI                      00016020
IC=IC+1
THINC=(DARCUS(CON)/2.)/FLOAT(NBI)    00016030
A1(IC)=0.
A2(IC)=0.
THINC2=THINC/2.
TH=.H1NC2+(KI-1)*THINC+CR          00016040
XB(IC)=AX*DCOS(TH)                  00016050
00016060
00016070
00016080
00016090
00016100
00016110
00016120
00016130
00016140
00016150

```

```

YB(IC)=AX*DSIN(TH)          00016160
HTA(IC)=TH/RAD             00016170
IF(J.GT.4) GO TO 20          00016180
IF(NELTYP(1).NE.2) GO TO 40          00016190
IF(J.EQ.1.AND.(K.EQ.1.OR.K.EQ.4)) AI(IC)= 00016200
*-BSTRX*DABS(DCOS(TH))      00016210
IF(J.EQ.2.AND.(K.EQ.3.OR.K.EQ.4)) AI(IC)= 00016220
*-BSTRX*DABS(DSIN(TH))      00016230
IF(J.EQ.3.AND.(K.EQ.2.OR.K.EQ.3)) AI(IC)= 00016240
*-BSTRX*DABS(DCOS(TH))      00016250
IF(J.EQ.4.AND.(K.EQ.1.OR.K.EQ.2)) AI(IC)= 00016260
*-BSTRX*DABS(DSIN(TH))      00016270
60 CONTINUE                  00016280
20 CONTINUE                  00016290
10 CONTINUE                  00016300
RETURN
END

C
C SUBROUTINE INFLN(WGHT,H,NRNK,J,KJ,I,NOPT)
C
IMPLICIT REAL*8(A-H,O-Z)          00016330
DIMENSION ELSTFF(50,10,10),WGHT(500),WK(150) 00016340
DIMENSION ELST: S(50,50,10),STS(50),STS(50,10) 00016350
DIMENSION AH(10,7),UVOUT(20)        00016360
DIMENSION PHI(3,7,400),STEMP(10,10),AO(10,3) 00016380
DIMENSION FINF(10,10),SINF(10,10),AINV(3,3) 00016390
DIMENSION APSX(500),APSY(500),APSXY(500) 00016400
DIMENSION H(2),XC(5),YC(5),NPLY(2) 00016410
DIMENSION IC(10)                  00016420
DIMENSION A(10,10),ATEMP2(10,10) 00016430
COMMON/UV/UVOUT                00016440
COMMON/XCYC/XC,YC              00016450
COMMON/ELP/AX,BX,NOUT,NSTS    00016460
COMMON/ELS/ELSTFF,ELSTSS       00016470
COMMON/STS/STS                 00016480
COMMON/INF1/APSX,APSY,APSXY   00016490
COMMON/I,YP/NPLY               00016500
COMMON/INV/AINV                00016510
COMMON/16520
COMMON/16530
COMMON/16540
COMMON/16550
COMMON/16560
COMMON/16570
COMMON/16580
COMMON/16590
COMMON/16600
COMMON/16610
COMMON/16620
COMMON/16630
COMMON/16640
COMMON/16650
COMMON/16660
COMMON/16670
COMMON/16680
COMMON/16690
COMMON/16700
COMMON/16710
COMMON/16720
COMMON/16730
COMMON/16740
COMMON/16750

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 AO(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)=0.000          00016760
200 CONTINUE                      00016770
C                                     00016780
C                                     STRESSES AND DISPLACEMENTS ARE STORED
C                                     FOR EACH LOAD CASE
C                                     00016800
C                                     00016810
C                                     00016820
C                                     DO 2107 KLK=1,8
2107 AN(KLK,J)=UVOUT(KLK)          00016830
IF(NRINK.EQ.5) GO TO 2221          00016840
IF(J.EQ.1) UVOUT(5)=UVOUT(9)       00016850
IF(J.EQ.1) UVOUT(7)=UVOUT(9)       00016850
IF(J.EQ.2) UVOUT(2)=UVOUT(16)     00016870
IF(J.EQ.2) UVOUT(8)=UVOUT(16)     00016880
IF(J.EQ.3) UVOUT(1)=UVOUT(13)      00016890
IF(J.EQ.3) UVOUT(5)=UVOUT(16)      00016900
IF(J.EQ.4) UVOUT(4)=UVOUT(12)      00016910
IF(J.EQ.4) UVOUT(6)=UVOUT(12)      00016920
00016930
2221 CONTINUE                      00016940
IF(NRINK.EQ.7.AND.J.LT.5) GO TO 371 00016950
AN(9,J)=(UVOUT(9)+UVOUT(13))/2.    00016960
AN(10,J)=(UVOUT(12)+UVOUT(16))/2.  00016970
GO TO 372                         00016980
371 AN(9,J)=UVOUT(7+2*XJ)          00016990
AN(10,J)=UVOUT(8+2*XJ)          00017000
372 CONTINUE                      00017010
DO 15 IS=1,NSTS                  00017020
15 STA(S,J)=STSV(IS)              00017030
DO 10 IS=1,NGPT                  00017040
PHI(1,J,IS)=APSX(IS)              00017050
10 CONTINUE                      00017060
DO 20 JS=1,NGPT                  00017070
PHI(2,J,IS)=APSY(IS)              00017080
20 CONTINUE                      00017090
DO 30 IS=1,NGPT                  00017100
PHI(3,J,IS)=APSXY(IS)             00017110
30 CONTINUE                      00017120
IF(J.LT.NPNK) RETURN              00017130
00017140
INTEGRATION OF STRESSES
DO 1010 III=1,10                 00017150
1010 CONTINUE                      00017160
NTR=NRINK+3                       00017170
DO 45 IK=1,NTR                   00017180
DO 45 JK=1,NTR                   00017190
45 FINF(IK,JK)=0.                  00017200
H1=H1*J*NPPLY(KJ)                00017210
00017220
DO 50 LI=1,NGPT                  00017230
DO 50 LJ=1,3                      00017240
DO 50 KI=1,NRNK                   00017250
SUM=0.                            00017260
00017270
DO 70 IL=1,3                      00017280
SUM=SUM+H1*AINV(LI,IL)*PHI(IL,KI,LI)
70 CONTINUE                      00017290
STE'P(CLJ,KI)=SUM                00017300
00017310
60 CONTINUE                      00017320
DO 80 LK=1,NRNK                   00017330
DO 80 LJ=1,NRNK                   00017340
SUM=0.                            00017350
DO 90 IL=1,3

```

```

      SUM=SUM+PHI(IL,LK,LI)*STEMP(IL,LJ)          00017360
90    CONTINUE
      FINF(LK,LJ)=FINF(LK,LJ)+SUM*WQHT(LI)      00017370
80    CONTINUE
50    CONTINUE
      DO 51 III=1,NR NK
      DO 51 JJJ=1,NR NK
51    STEMP(III,JJJ)=(FINF(III,JJJ)+FINF(JJJ,III))/2.0D0 00017380
      DO 52 III=1,NR NK
      DO 52 JJJ=1,NR NK
52    FINF(III,JJJ)=STEMP(III,JJJ)              00017390
      CALL LINV2F(FINF,N,I,K,10,SINF,4,WK,IER)
      DO 410 IA=1,NTR
      AO(IA,1)=0.5+0.5*(-1.0)**(IA+1)           00017400
      AO(IA,2)=0.5+0.5*(-1.0)**(IA)               00017410
410   AO(IA,3)=0.0D0                            00017420
      AO(1,3)=DABS(YC(4))                         00017430
      AO(2,3)=-DABS(XC(4))                        00017440
      AO(3,3)=-DABS(YC(3))                        00017450
      AO(4,3)=-DABS(XC(3))                        00017460
      AO(5,3)=-DABS(YC(2))                        00017470
      AO(6,3)=DABS(XC(2))                         00017480
      AO(7,3)=DABS(YC(1))                         00017490
      AO(8,3)=DABS(XC(1))                         00017500
      DO 420 KK=1,NTR
      DO 420 LL=1,NR NK
      SUM=0.0D0
      DO 430 JJ=1,NR NK
430   SUM=SUM+AN(KK,JJ)*SINF(JJ,LL)             00017510
420   STEMP(KK,LL)=SUM                           00017520
      DO 440 KK=1,NTR
      N1=NTR-2
      DO 440 JJ=N1,NTR
440   STEMP(KK,JJ)=AC(KK,JJ-NR NK)             00017530
      CALL LINV2F(STEMP,NTR,10,FINF,4,WK,IER)
      DO 450 II=1,NR NK
      DO 450 JJ=1,NTR
      SUM=0.0D0
      DO 460 KK=1,NR NK
460   SUM=SUM+SINF(II,KK)*FINF(KK,JJ)           00017540
450   STEMP(II,JJ)=SUM                           00017550
      DO 470 II=1,NTR
      DO 470 JJ=1,NTR
      SUM=0.0D0
      DO 480 KK=1,NR NK
480   SUM=SUM+FINF(KK,II)*STEMP(KK,JJ)           00017560
      ELSEFF(I,II,JJ)=SUM                         00017570
      AC(II,JJ)=SUM                             00017580
470   CONTINUE
      DO 550 II=1,NR NK
      DO 550 JJ=1,NTR
      SUM=0.0D0
      DO 560 KK=1,NR NK
560   SUM=SUM+SINF(II,KK)*FINF(KK,JJ)           00017590
550   STEMP(II,JJ)=SUM                           00017600
      DO 570 II=1,NSTS
      DO 570 JJ=1,NTR
      SUM=0.0D0
      DO 580 KK=1,NR NK
580   SUM=SUM+CA(II,KK)*STEMP(KK,JJ)           00017610

```

```

570 ELSTSG(I,II,JI)=SUM          00017960
      RETURN                      00017970
      END                         00017980
                                00017990
                                00018000
                                00018010
                                00018020
C C
SUBROUTINE SMAX(HT,KJ,I)          00018030
IMPLICIT REAL*8(A-H,O-Z)          00018040
DIMENSION AINV(3,3),STULT(3,2),AVN(3)
DIMENSION NV(3)                   00018050
DIMENSION PSMX(50,4),STM(3),CM(2)  00018060
DIMENSION NPLY(2),NUMPLY(2),ANG(5,2),IPLY(100,2) 00018070
DIMENSION E1(2),E2(2),Q12(2),V12(2),V21(2)  00018080
COMMON/MOD/E1,E2,Q12,V12,V21    00018090
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY  00018100
COMMON/STMT/STM,CM              00018110
COMMON/INV/AINV                00018120
COMMON/SMX/PSMX                00018130
CONTINUE/STLT/STLT              00018140
DATA CMG/'C'
IF(CH(KJ),E1,CMG) GO TO 222   00018150
PSMX(I,1)=STM(1)               00018160
PSMX(I,2)=STM(2)               00018170
PSMX(I,3)=STM(3)               00018180
PSMX(I,4)=STM(2)               00018190
PSMX(I,5)=STM(2)               00018200
RETURN                         00018210
222 CONTINUE                     00018220
                                00018230
C C
COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM 00018240
FIBER STRAINS FOR EACH FAILURE MODE             00018250
                                00018260
DO 100 K=1,5                      00018270
DO 10 II=1,3                      00018280
NV(I2)=0                          00018290
10 AVN(II)=0.000                 00018300
IF(K.EQ.1) NV(1)=1                00018310
IF(K.EQ.2) NV(1)=-1              00018320
IF(K.EQ.3) NV(3)=1                00018330
DO 15 II=1,3                      00018340
DO 15 JJ=1,7                      00018350
15 NV(JJ)=AVN(II)+AINV(II,JJ)*NV(JJ) 00018360
      CONTINUE
      NP=NUMPLY(KJ)                00018370
      CMX=0.0D0                    00018380
      RAD=DARCOS(-1.0D0)/180.D0   00018390
      DO 25 II=1,NP               00018400
      TH=ANG(II,KJ)*RAD          00018410
      F11=DCOS(TH)*X2*AVN(1)+AVN(2)*DSIN(TH)*XN2+ 00018420
      *DCOS(TH)*DSIN(TH)*AVN(3)  00018430
      IF(K.NE.1) GO TO 65         00018440
      EPRT=E11/STULT(2,KJ)        00018450
      GO TO 50                     00018460
50 IF(K.NE.2) GO TO 75            00018470
      EPRT=E11/STULT(1,KJ)        00018480
      GO TO 50                     00018490
50 EPRT=E11/STULT(2,KJ)          00018500
50 CONTINUE
      IF(DABS(SMX).LT.DARS(EPRT)) SMX=EPRT 00018510
25 CONTINUE
      IF(DABS(SMX).GT.1.0D-10) GO TO 555 00018520
                                00018530
                                00018540
                                00018550

```

```

      PSMX(I,K)=STULT(S,KJ)*G12(KJ)          00018560
      GO TO 100                                00018570
555  CONTINUE                                00018580
      PSMX(I,K)=DAB3(1,ND0/SMX)               00018590
100  CONTINUE                                00018600
      PSMX(I,4)=PSMX(I,2)                     00018610
      RETURN                                   00018620
      END                                     00018630
C
C
C      SUBROUTINE POLY(W,AST,J,K,NCOL,LTNCM)
C
C      IMPLICIT REAL*8(A-H,O-Z)                00018640
C      DIMENSION XC(5),YC(5),A1(200),A2(200),XB(200) 00018650
C      DIMENSION YB(200),T(200),A1A(4),A2A(4)        00018660
C      COMMON/CMT1/XB,YB,A1,A2,T                00018670
C      COMMON/XCYC/XC,YC                         00018680
C
C      ARRAY COLLOCATION POINTS AROUND EXTERIOR    00018690
C      BOUNDARY AND APPLY STRESS BOUNDARY         00018700
C      CONDITION                                  00018710
C
C      DO 120 I=1,4                            00018720
C      A1A(I)=0.0                               00018730
C      A2A(I)=0.0                               00018740
120  CONTINUE                                00018750
      IF(LTNCM.EQ.1) A1A(1)=AST                00018760
      IF(LTNCM.EQ.2) A1A(1)=-AST               00018770
      J=0
      XC(5)=XC(1)                             00018780
      YC(5)=YC(1)                             00018790
      DO 10 I=1,4                            00018800
      X=XC(I)-XC(I+1)                         00018810
      Y=YC(I+1)-YC(I)                         00018820
      IF(X.EQ.0.) X=1.D-6                      00018830
      IF(Y.EQ.0.) Y=1.D-6                      00018840
      TH=DATAN2(X,Y)                         00018850
      TH=TH*180./DARCCOS(-0.1D1)              00018860
      DX=(XC(I+1)-XC(I))/(NCOL+1)            00018870
      DY=(YC(I+1)-YC(I))/(NCOL+1)            00018880
      DO 20 II=1,NCOL                         00018890
      J=J+1
      IF(I.EQ.1.OR.I.EQ.3) GO TO 23          00018900
      YB(J)=YC(I)                           00018910
      XB(J)=XC(I)+DX*(II+.5)                 00018920
      IF(II.EQ.1) XB(J)=XC(I)+(DX/2.)       00018930
      GO TO 24                                00018940
23   CONTINUE                                00018950
      YB(J)=YC(I)+DY*(II+.5)                 00018960
      IF(II.EQ.1) YB(J)=YC(I)+(DY/2.)       00018970
      XB(J)=XC(I)                           00018980
      J=J+1
      T(J)=TH                                 00018990
      A1(J)=A1A(I)                           00019000
      A2(J)=A2A(I)                           00019010
24   T(J)=TH                                 00019020
      GO TO 24                                00019030
      GO TO 24                                00019040
20   CONTINUE                                00019050
10    CONTINUE                                00019060
      RETURN                                   00019070
      END                                     00019080

```

```

SUBROUTINE CIRC(H,AST,JK,K,LTNCM)
ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY AND
APPLY BEARING LOAD IN A COSINUSOIDAL DISTRIBUTION

IMPLICIT REALN8(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,H
CON=-1.0
XSTR=AST
BSTR=(2.*WXXSTR)/(DARCOS(CON)*H)
NMG=N-4
NQ=NH4/4
DO 20 I=1,N
JK=JK+1
TH=((I-1)*H2+1)*DARCOS(CON)/N
X(I)=A*DARCOS(TH)
Y(I)=B*D SIN(TH)
CS=-X(I)*XB/(Y(I)*(A,A))
IF(Y(I).GT.0.0)THTA(JK)=DATAN(CS)-DARCOS(CON)/2.
IF(Y(I).LT.0.0)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)
204 X(I)=((1.-OT_0_1)*TET11*ARCOS(-1.0)-DATAN(Y(I))/X(I))
Y(I)=((1.-OT_0_1)*TET11*ARCOS(-1.0)+DATAN(Y(I))/X(I))
OT_0_1=1.+BSTR*(1.-COS(THTA))
CS(-JK)=0.
XB(JK)=X(I)
YB(JK)=Y(I)
20 CONTINUE
RETURN
END

SUBROUTINE FIGECM(H,PHS,NJ,NOPT4,NCLL)

```

```

IMPLICIT REAL*8(A-H,O-Z)          00019760
DIMENSION A(3,3),WK(25),AX(3,3),AZ(5),WKK(121),BC(300) 00019770
DIMENSION CH(4),H(2)              00019780
COMPLEX*16 CRHS(100)             00019790
COMPLEX*16 CM(300,90),CMC(300,90),CMCTCM(90,90),RH3(90) 00019800
COMMON/ROOTS/R1,R2               00019810
COMMON/AMTA/A                   00019820
COMMON/TERMS/P1,Q1,P2,Q2         00019830
COMMON/ELPA/AX,BX,NOUT,NSTS     00019840
COMMON/SER/NT,NB                 00019850
COMMON/THV/AI                   00019860
COMPLEX*16 Z(4),Z1,Z2,Q1,Q2,P1,P2,R1,R2,WA(14883)      00019870
C
C   CCC AMATRX CALCULATES THE LAMINATE 'A' MATRIX          00019880
C   CALL AMATRX(H,PHS,KJ)          00019890
C   N=3                         00019900
C   IDQT=4                      00019910
C   IA=3                        00019920
C
C   CCC LINV2F INVERTS THE 'A' MATRIX                      00019930
C   CALL LINV2F(A,N,IA,AI,IDQT,WK,IER)                    00019940
C   NDEO=4                      00019950
C   AZ(1)=AI(1,1)/AI(2,2)          00019960
C   AZ(2)=-2.*AI(1,3)/AI(2,2)        00019970
C   AZ(3)=(2.*AI(1,2)+AI(3,3))/AI(2,2)                  00019980
C   AZ(4)=-2.*AI(2,3)/AI(2,2)        00019990
C   AZ(5)=1.0D0                     00020000
C
C   CCC ZRPOLY FINDS THE ROOTS OF THE CHARACTERISTIC EQUATION 00020010
C   CALL ZRPOLY(AZ,NDEO,Z,IER)          00020020
C
C   CCC Z(2) AND Z(4) ARE THE COMPLEX CONJUGATES OF Z(1)      00020030
C   AND Z(3) RESPECTIVELY           00020040
C
C   R1=Z(1)                      00020050
C   R2=Z(3)                      00020060
C
C   CCC THE TWO ROOTS MUST BE CHECKED FOR A UNITARY COMPONENT    00020070
C   IN EITHER THE REAL OR IMAGINARY PART; SUCH AN              00020080
C   OCCURANCE SIGNIFIES A QUASI-ISOTROPIC LAYUP AND          00020090
C   THE VALUE MUST BE PERTURBED SLIGHTLY IN ORDER TO          00020100
C   AVOID A SINGULAR MATRIX           00020110
C
C   CH(1)=R1                      00020120
C   CH(2)=(0.0,-1.0)*R1            00020130
C   CH(3)=R2                      00020140
C   CH(4)=(0.0,-1.0)*R2            00020150
C   DO 30 IJK=1,4                 00020160
C   AR=DABS(CH(IJK))              00020170
C   IF(AR.LE 1.0) GO TO 31         00020180
C   GO TO 32                      00020190
C   31 IF((1.0-AR).LT.0.02) CH(IJK)=0.98          00020200
C   GO TO 30                      00020210
C   32 IF((AR-1.0).LT.0.02) CH(IJK)=1.02          00020220
C   30 CONTINUE                     00020230
C   R1=DCMPLX(CH(1),CH(2))        00020240
C   R2=DCMPLX(CH(3),CH(4))        00020250

```

```

C   CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS      00020360
C   P1=AI(1,1)*R1**2+AI(1,2)-AI(1,3)*R1      00020370
C   P2=AI(1,1)*R2**2+AI(1,2)-AI(1,3)*R2      00020380
C   Q1=AI(2,2)/R1+AI(1,2)*R1-AI(2,3)      00020390
C   Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)      00020400
C   INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS      00020410
C   NT4=4*NT      00020420
C   NT8=8*NT      00020430
C   NT8P4=8*NT+4      00020440
C   NT8P2=8*NT+2      00020450
C   NT8P1=8*NT+1      00020460
C   NB2=2*NB      00020470
C   NKK=NT8P1*(NT8P1+2)      00020480
C   CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,      00020490
C   INT3P1,NB2,NKK,WA,WKK,NOPT4,KJ,NCLL)      00020500
C   RETURN      00020510
C   END      00020520
C   SUBROUTINE AMATRX(H,PMS,K)      00020530
C   ASSEMBLE THE A MATRIX      00020540
C   IMPLICIT REALN8(A-H,O-Z)      00020550
C   DIMENSION A(3,3),ANO(5,2),H(2),NPLY(2),NUMPLY(2)      00020560
C   DIMENSION E1(2),E2(2),Q12(2),V12(2),V21(2)      00020570
C   DIMENSION IPLY(100,2)      00020580
C   COMMON/MOD/E1,E2,Q12,V12,V21      00020590
C   COMMON/AMT/A      00020600
C   COMMON/LYP/NPLY,NUMPLY,ANO,IPLY      00020610
C   THKNESS=NPLY(K)*H(K)      00020620
C   DENO=1.-E2(K)*V12(K)**2/E1(K)      00020630
C   Q11=E1(K)/DENO      00020640
C   Q22=E2(K)/DENO      00020650
C   Q12=V12(K)*Q22      00020660
C   Q21=Q12      00020670
C   Q33=Q12(K)      00020680
C   DO 10 I=1,3      00020690
C   DO 10 J=1,3      00020700
10  A(I,J)=0.0D0      00020710
NN=NPLY(K)
T=H(K)
DO 20 I=1,NN
LP=IPLY(I,K)
THTAI=(ANO(LP,K)+P'M)*DARCOS(-1.D0)/180.D0
C=DCOS(THTAI)
S=DSIN(THTAI)
A(1,1)=(Q11*C**4+2.*((Q12+2.*M*Q33)*C*C*S*S+Q22*S*S*C*C))**T+A(1,1)
A(2,2)=(Q11*S*S*C*C+2.*((Q12+2.*M*Q33)*C*C*S*S+Q22*S*S*C*C))**T+A(2,2)
A(1,2)=((Q11+Q12-4.*M*Q33)*C*C*S*S+Q12*(C*C*4+S*S*4))**T+A(1,2)
A(2,1)=A(1,2)
A(3,3)=((Q11+Q22-2.*M*Q33)*C*C*S*S+Q33*(C*C*4+S*S*4))**T+A(3,3)
A(1,3)=((Q11-Q12-2.*M*Q33)*C*C*S*S+(Q12-Q22+2.*M*Q33)*S*S*C*C)**T+A(1,3)
A(2,3)=((Q11-Q12-2.*M*Q33)*S*S*C+C*(Q12-Q22+2.*M*Q33)*C*C*S*S)**T+A(2,3)
A(3,2)=A(2,3)

```

```

A(3,1)=A(1,3)          00020960
20 CONTINUE             00020970
DO 33 I=1,3             00020980
DO 33 J=1,3             00020990
A(I,J)=A(I,J)/THKNES  00021000
33 CONTINUE             00021010
RETURN                 00021020
END                    00021030
00021040
00021050
00021060
00021070
00021080
00021090
00021100
00021110
00021120
00021130
00021140
00021150
00021160
00021170
00021180
00021190
00021200
00021210
00021220
00021230
00021240
00021250
00021260
00021270
00021280
00021290
00021300
00021310
00021320
00021330
00021340
00021350
00021360
00021370
00021380
00021390
00021400
00021410
00021420
00021430
00021440
00021450
00021460
00021470
00021480
00021490
00021500
00021510
00021520
00021530
00021540
00021550

SUBROUTINE CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,
INT8P1,NB2,NWK,WA,WKK,NOPT4,KJ,NCOL)          00021070
CMAT OUTPUTS STRESSES, STRAINS, AND DISPLACEMENTS  00021100
AT SPECIFIED COORDINATES                         00021110
00021120
00021130
00021140
IMPLICIT REAL*8(A-H,O-Z)                         00021150
DIMENSION ASX(400),ASXY(400),UVOUT(20)           00021160
DIMENSION XC(5),YC(5)                            00021170
DIMENSION THTA(200),X(200),Y(200),AMAT(3,3)      00021180
DIMENSION AIN1(200),AIN2(200),BC(NB2)            00021190
DIMENSION WKK(NT8P1),WORK(700)                   00021200
DIMENSION XOUT(600),YOUT(600),STSV(50)           00021210
DIMENSION FUR(400),FTHT(400),FSMR(400)          00021220
DIMENSION RTHT(400),REPX(400),REPY(400),REPXY(400) 00021230
DIMENSION APSX(500),APSY(500),APSXY(500)         00021240
COMPLEX*16 CMCTCM(NT8P1,NT8P1),RHS(NT8P1),PHI1D,PHI2D,XETA1,XETA2 00021250
COMPLEX*16 ACD(25,25),ACD2(25,25),RH32(25)       00021260
COMPLEX*16 U0,V0                                00021270
COMPLEX*16 CM(NB2,NT8P4),CMC(NB2,NT8P1),Z1,Z2,Z11,Z22,R1,R2 00021280
COMPLEX*16 T11,T12,T21,T22,P11,P12,P21,P22       00021290
COMPLEX*16 P1,P2,Q1,Q2,DCMPLX,C0,CSUM,GRHS(NT8P2) 00021300
COMPLEX*16 PHI1DP,PHI2DP,PHI1DN,PHI2DN           00021310
COMPLEX*16 PHI1P,PHI2P,PHI1H,PHI2N,PHI1,PHI2    00021320
COMPLEX*16 PHI3N,PHI3P,PHI3,PHI4N,PHI4P,PHI4   00021330
COMPLEX*16 SV11,SV12,SV21,SV22,RB11,RB21,RB11B,RB21B 00021340
COMPLEX*16 R1B,R2B,P1B,P2B,Q1B,Q2B,WA(NWK)       00021350
COMMON//INFI//APSX,APSY,APSXY                   00021360
COMMON//XCYC/XC,YC                           00021370
COMMON//NCST//NCASE,NTYPE                      00021380
COMMON//XXY1//ASX,ASXY                        00021390
COMMON//STS//STSV                          00021400
COMMON//UV//UVOUT                         00021410
COMMON//ROOTS//R1,R2                         00021420
COMMON//TERMS//P1,Q1,P2,Q2                   00021430
COMMON//CMT1//X,Y,AIN1,AIN2,THTA            00021440
COMMON//CMT2//XOUT,YOUT                      00021450
COMMON//FB2//FUR,FTHT,FSMR                  00021460
COMMON//QMT//RTHT,REPX,REPY,REPXY           00021470
COMMON//ELP//AX,BX,HOUT,NSTS                00021480
COMMON//SER//NT,NB                           00021490
COMMON//INV//AMAT                          00021500
IF(NOPT4.EQ.5.AND.NCASE.OT.1) GO TO 3335
DO 6666 III=1,NT8P1                         00021510
DO 6666 IHM=1,NT8P1                         00021520
6666 CMCTCM(III,IHM)=(0.000,0.000)
A*AX

```

```

B=BX
CO=(0.0,1.0)
RB11=(Q1-P1×R1)/(A-C0×R1×B)
RB21=(Q2-P2×R2)/(A-C0×R2×B)
REALR1=R1
REALR2=R2
REALP1=P1
REALP2=P2
REALQ1=Q1
REALQ2=Q2
RRB11=RB11
RRB21=RB21
AIMGR1=C0×R1
AIMGR2=C0×R2
AIMGP1=C0×P1
AIMGP2=C0×P2
AIMHQ1=C0×Q1
AIMHQ2=C0×Q2
ARB11=C0×RB11
ARB21=C0×RB21
R1B=DCMPLX(REALR1,AIMGR1)
R2B=DCMPLX(REALR2,AIMGR2)
P1B=DCMPLX(REALP1,AIMGP1)
P2B=DCMPLX(REALP2,AIMGP2)
Q1B=DCMPLX(REALQ1,AIMHQ1)
Q2B=DCMPLX(REALQ2,AIMHQ2)
RB11B=DCMPLX(RRB11,ARB11)
RB21B=DCMPLX(RRB21,ARB21)
JJJ=0
DO 1000 I=1,NB
J=IM2
THTAI=THTA(I)×DARCCOS(-1.0D0)/180.D0
C=DCOS(THTAI)
S=DSIN(THTAI)
P11=CXP1+SXQ1
P12=CXP2+SXQ2
P21=-CXPI+CXQ1
P22=-SXPI+CXQ2
T11=(CXCXR1×R1+SMS-2.×CXHSXR1)
T12=(CYCXR2×R2+SMS-2.×CYSXSR2)
T21=(-CXCXR1×R1+CMS-(CXC-SXS)×R1)
T22=(-CXCXR2×R2+CMS-(CXC-SXS)×R2)
Z1=X(I)+R1×Y(I)
Z2=X(I)+R2×Y(I)
Z11=CDSQRT(Z1×Z1-AHA-R1×R1×BMB)
Z22=CDSQRT(Z2×Z2-AHA-R2×R2×BMB)
REAL1=Z11
AIMG1=-CO×Z11
IF(DABS(REAL1).LE.1.D-14)REAL1=0.0D0
IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.0D0
Z11=DCMPLX(REAL1,AIMG1)
REAL2=Z22
AIMG2=-CO×Z22
IF(DABS(REAL2).LE.1.D-16)REAL2=0.0D0
IF(DABS(AIMG2).LE.1.D-16)AIMG2=0.0D0
Z22=DCMPLX(REAL2,AIMG2)
XETA1=(Z1+Z11)/(A-C0×R1×B)
IF(CDABC(XETA1).LT.0.999) GO TO 300
GO TO 310
300 Z11=-Z11
310

```

```

310 XETA1=(Z1+Z11)/(A-COMR1*B) 00022160
XETA2=(Z2+Z22)/(A-COMR2*B) 00022170
IF(CDABS(XETA2).LT.0.999) GO TO 320 00022180
GO TO 330 00022190
320 Z22=-Z22 00022200
XETA2=(Z2+Z22)/(A-COMR2*B) 00022210
330 CONTINUE 00022220
JJJ=JJJ+1 00022230
00022240
NORMAL & TANGENTIAL STRESS BOUNDARY CONDITIONS ARE IMPOSED 00022250
DO 5 N=1,NT 00022260
NP=N 00022270
CM(J-1,N)=NPNXETA1*NPNT11/Z11 00022280
CM(J-1,2*NT+N)=NPNXETA2*NPNT12/Z22 00022290
CM(J,N)=NPNXETA1*NPNT21/Z11 00022300
CM(J,2*NT+N)=NPNXETA2*NPNT22/Z22 00022310
NN=-N 00022320
CM(J-1,NT+N)=NNNXETA1*NNNNT11/Z11 00022330
CM(J-1,3*NT+N)=NNNXETA2*NNNNT12/Z22 00022340
CM(J,NT+N)=NNNXETA1*NNNNT21/Z11 00022350
CM(J,3*NT+N)=NNNXETA2*NNNNT22/Z22 00022360
5 CONTINUE 00022370
CM(J-1,NT8+1)=T11/Z11 00022380
CM(J-1,NT8+2)=T12/Z22 00022390
CM(J,NT8+1)=T21/Z11 00022400
CM(J,NT8+2)=T22/Z22 00022410
00022420
1000 CONTINUE 00022430
DO 195 I=1,NB2 00022440
DO 196 J=1,NT4 00022450
REAL1=CM(I,J) 00022460
AIM01=-COMXCM(I,J) 00022470
IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00022480
IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00022490
CM(I,J)=DCMPLX(REAL1,AIM01) 00022500
AIM02=-AIM01 00022510
CM(I,NT4+J)=DCMPLX(REAL1,AIM02) 00022520
00022530
196 CONTINUE 00022540
195 CONTINUE 00022550
DO 295 I=1,NB2 00022560
DO 296 J=1,2 00022570
REAL1=CM(I,NT8+J) 00022580
AIM01=-COMXCM(I,NT8+J) 00022590
IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00022600
IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0 00022610
CM(I,NT8+J)=DCMPLX(REAL1,AIM01) 00022620
AIM02=-AIM01 00022630
CM(I,NT8+2+J)=DCMPLX(REAL1,AIM02) 00022640
00022650
296 CONTINUE 00022660
295 CONTINUE 00022670
SV11=(P2*Q1B-Q2*P1B)/(Q1*P2-Q2*P1) 00022680
SV12=(P2*Q2B-Q2*P2B)/(Q1*P2-Q2*P1) 00022690
SV21=(Q1*P1B-Q1*P1)/(Q1*P2-Q2*P1) 00022700
SV22=(Q1*P2B-Q2*P1)/(Q1*P2-Q2*P1) 00022710
DO 139 I=1,NB2 00022720
00022730
IMPOSE RIGID BODY ROTATION CONDITION 00022740
CM(I,2*NT+1)=-CM(I,1)*RB21/RB11+CM(I,2*NT+1) 00022750
CM(I,4*NT+1)=-CM(I,1)*RB11B/RB11+CM(I,4*NT+1)

```

```

C C
      CM(I,6*NT+1)=-CM(I,1)*RB21B/RB11+CM(I,6*NT+1)
      CM(I,1)*(0.0,0.0)                                00022760
      00022770
      00022790
      00022800
      00022810
      00022820
      00022830
      00022840
      00022850
      00022860
      00022870
      00022880
      00022890
      00022900
      00022910
      00022920
      00022930
      00022940
      00022950
      00022960
      00022970
      00022980
      00022990
      00023000
      00023010
      00023020
      00023030
      00023040
      00023050
      00023060
      00023070
      00023080
      00023090
      00023100
      00023110
      00023120
      00023130
      00023140
      00023150
      00023160
      00023170
      00023180
      00023190
      00023200
      00023210
      00023220
      00023230
      00023240
      00023250
      00023260
      00023270
      00023280
      00023290
      00023300
      00023310
      00023320
      00023330
      00023340
      00023350

      IMPOSE SINGLE-VALUEUNESS CONDITION

      CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)
      CI(.,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)
      CI(.,NT8+2)*(0.0,0.0)

139  CONTINUE
      DO 141 I=1,NB2
      DO 142 J=1,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,NT3P1
      REAL1=CM(I,J)
      AIM01=-CC $\times$ CM(I,J)
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0
      IF(DABS(AIM01).LE.1.D-16)AIM01=0.0D0
      CM(I,J)=DCMPLX(REAL1,AIM01)
      AIM02=-AIM01
      CNC(I,J)=DCMPLX(REAL1,AIM02)
95   CONTINUE
96   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
      RHS(I)=CSUM
130   RHS(I)=CSUM
      IJOB=0
      IF(NOPY4.EQ.5.AND.NCASE.GT.1) IJGB=2
      M=1
      CALL LF02C(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

```

```

RAD=DARCOS(-1.0D0)/180.0D0          00023360
IC=1                                  00023370
IC2=1                                00023380
SUMU1=0.0D0                            00023390
SUMV1=0.0D0                            00023400
SUMU2=0.0D0                            00023410
SUMV2=0.0D0                            00023420
NADD=0                                00023430
IF(NOPT4.EQ.1) GO TO 1195              00023440
IF(NTYPE.NE.2.OR.NCASE.GT.4) GO TO 1196 00023450
NADD=4*NCUL                            00023460
NIC=1                                  00023470
DO 197 III=1,NADD                     00023480
XOUT(NOUT+NSTS+III)=X(III)           00023490
YOUT(NOUT+NSTS+III)=Y(III)           00023500
197 MN=NOUT+NSTS+NADD                00023510
XOUT(MN+1)=AX                          00023520
YOUT(MN+1)=0.0D0                      00023530
XOUT(MN+2)=0.0D0                      00023540
YOUT(MN+2)=-AX                        00023550
XOUT(MN+3)=AX*DOS(177.0D0*RAD)       00023560
YOUT(MN+3)=AX*DSIN(177.0D0*RAD)      00023570
XOUT(MN+4)=0.0D0                      00023580
YOUT(MN+4)=AX                          00023590
NADD=NADD+4                           00023600
GO TO 1195                            00023610
1196 C:CONTINUE                         00023620
NADD=8                                00023630
NIC=2                                  00023640
MN=NOUT+NSTS                          00023650
DO 199 III=1,4                         00023660
ICM=5-III                            00023670
XOUT(MN+III)=XC(ICM)                 00023680
YOUT(MN+III)=YC(ICM)                 00023690
199 CONTINUE                           00023700
MN=MN+4                                00023710
XOUT(MN+1)=AX                          00023720
YOUT(MN+1)=0.0D0                      00023730
XOUT(MN+2)=0.0D0                      00023740
YOUT(MN+2)=-AX                        00023750
XOUT(MN+3)=AX*DOS(177.0D0*RAD)       00023760
YOUT(MN+3)=AX*DSIN(177.0D0*RAD)      00023770
XOUT(MN+4)=0.0D0                      00023780
YOUT(MN+4)=AX                          00023790
1195 C:CONTINUE                         00023800
NRCF=NOUT                             00023810
IF(NOPT4.EQ.5) NRCF=NOUT+NSTS+NADD   00023820
NINC=NSTS/4                            00023830
DO 190 K=1,NRCF                      00023840
Z1=XOUT(K)+R1*YOUT(K)                 00023850
Z2=XOUT(K)+R2*YOUT(K)                 00023860
Z11=CD3QRT(Z1*Z1-AMA-R1*R1*B*B)     00023870
Z22=CD5QRT(Z2*Z2-AMA-R2*R2*B*B)     00023880
XETA1=(Z1+Z11)/(A-COMR1*B)           00023890
IF(CDABS(XETA1).LT.0.999) GO TO 400  00023900
GO TO 410                            00023910
400 Z11=-Z11                           00023920
XETA1=(Z1+Z11)/(A-COMR1*B)           00023930
410 XETA2=(Z2+Z22)/(A-COMR2*B)       00023940
IF(CDABS(XETA2).LT.0.999) GO TO 420  00023950

```

```

GO T1 430
420 Z22=-Z22
XET1:=(Z2+Z22)/(A-C0*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=NPNXETA1*NNNPKGRHS(N)/Z11+PHI1DP
PHI1DN=NNNXETA1*NNNPKGRHS(NT+N)/Z11+PHI1DN
PHI2DP=NPNXETA2*NNNPKGRHS(2*NT+N)/Z22+PHI2DP
PHI2DN=NNNXETA2*NNNPKGRHS(3*NT+N)/Z22+PHI2DN
PHI1P-XETA1*NNNPKGRHS(N)+PHI1P
PHI1N-XETA1*NNNPKGRHS(NT+N)+PHI1N
PHI2P-XETA2*NNNPKGRHS(2*NT+N)+PHI2P
PHI2N-XETA2*NNNPKGRHS(3*NT+N)+PHI2N
00023960
00023970
00023980
00023990
00024000
00024010
00024020
00024030
00024040
00024050
00024060
00024070
00024080
00024090
00024100
00024110
00024120
00024130
00024140
00024150
00024160
00024170
00024180
00024190
00024200
00024210
00024220
00024230
00024240
00024250
00024260
00024270
00024280
00024290
00024300
00024310
00024320
00024330
00024340
00024350
00024360
00024370
00024380
00024390
00024400
00024410
00024420
00024430
00024440
00024450
00024460
00024470
00024480
00024490
00024500
00024510
00024520
00024530
00024540
00024550
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*PHI1+R2*R2*PHI2D
SGMAY=2.*R1*PHI1D+R2*R2*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(-1.00)
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
EPSI=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)=SC**AX

```

ASXY(K)=SGMAXY	00024560
FUR(X)=UR	00024570
FTHT(K)=TETAA	00024580
FSMR(K)=SGMAR	00024590
3338 CONTINUE	00024600
IF(NOPT4.EQ.5.AND.K.GT.NOUT) GO TO 3339	00024610
APSX(K)=SGMAX	00024620
APSY(K)=SGMAY	00024630
APSXY(K)=SGMAXY	00024640
3339 CONTINUE	00024650
IF(NOPT4.EQ.1) GO TO 190	00024660
IF(NOPT4.EQ.5.AND.K.LE.NOUT.OR.K.GT.(NOUT+NSTS)) GO TO 191	00024670
IF(IC2.LE.NINC) STSV(IC2)=SGMAX	00024680
IF(IC2.GT.NINC.AND.IC2.LE.(2*NINC)) STSV(IC2)=SGMAXY	00024690
IF(IC2.GT.(2*NINC).AND.IC2.LE.(3*NINC)) STSV(IC2)=SGMAX	00024700
IF(IC2.GT.(3*NINC).AND.IC2.LE.(4*NINC)) STSV(IC2)=SGMAX	00024710
IC2=IC2+1	00024720
GO TO 190	00024730
191 CONTINUE	00024740
IF(NIC.EQ.1) GO TO 192	00024750
IF(NOPT4.EQ.5.AND.X.LT.(NRCF-7)) GO TO 190	00024760
UVOUT(IC)=U	00024770
UVOUT(IC+1)=V	00024780
IC=IC+2	00024790
GO TO 190	00024800
192 CONTINUE	00024810
NC=NOUT+NSTS	00024820
IF(K.GT.NC.AND.K.LE.(NC+NCOL)) SUMU1=SUMU1+U	00024830
IF(K.GT.(NC+NCOL).AND.K.LE.(NC+2*NCOL)) SUMV1=SUMV1+V	00024840
IF(K.GT.(NC+2*NCOL).AND.K.LE.(NC+3*NCOL)) SUMU2=SUMU2+U	00024850
IF(K.GT.(NC+3*NCOL).AND.K.LE.(NC+4*NCOL)) SUMV2=SUMV2+V	00024860
NNC=NC+4*NCOL	00024870
IF(K.EQ.(NNC+1)) UVOUT(9)=U	00024880
IF(K.EQ.(NNC+1)) UVOUT(10)=V	00024890
IF(K.EQ.(NNC+2)) UVOUT(11)=U	00024900
IF(K.EQ.(NNC+2)) UVOUT(12)=V	00024910
IF(K.EQ.(NNC+3)) UVOUT(13)=U	00024920
IF(K.EQ.(NNC+3)) UVOUT(14)=V	00024930
IF(K.EQ.(NNC+4)) UVOUT(15)=U	00024940
IF(K.EQ.(NNC+4)) UVOUT(16)=V	00024950
193 CONTINUE	00024960
DISPLACEMENTS ARE AVERAGED OVER ELEMENT SIDES FOR CERTAIN LOAD CASES	00024970
IF(NIC.NE.1) RETURN	00024980
SUMU1=SUMU1/FLOAT(NCOL)	00024990
SUMV1=SUMV1/FLOAT(NCOL)	00025000
SUMU2=SUMU2/FLOAT(NCOL)	00025010
SUMV2=SUMV2/FLOAT(NCOL)	00025020
UVOUT(1)=SUMU2	00025030
UVOUT(2)=SUMV2	00025040
UVOUT(3)=SUMU2	00025050
UVOUT(4)=SUMV1	00025060
UVOUT(5)=SUMV1	00025070
UVOUT(6)=SUMV1	00025080
UVOUT(7)=SUMU1	00025090
UVOUT(8)=SUMV2	00025100
RETURN	00025110
END	00025120
	00025130
	00025140
	J0025150

```

SUBROUTINE FBOLT(ANGK,H,PSH,K)          00025160
                                         00025170
                                         00025180
                                         00025190
                                         00025200
                                         00025210
                                         00025220
                                         00025230
                                         00025240
                                         00025250
                                         00025260
                                         00025270
                                         00025280
                                         00025290
                                         00025300
                                         00025310
                                         00025320
                                         00025330
                                         00025340
                                         00025350
                                         00025360
                                         00025370
                                         00025380
                                         00025390
                                         00025400
                                         00025410
                                         00025420
                                         00025430
                                         00025440
                                         00025450
                                         00025460
                                         00025470
                                         00025480
                                         00025490
                                         00025500
                                         00025510
                                         00025520
                                         00025530
                                         00025540
                                         00025550
                                         00025560
                                         00025570
                                         00025580
                                         00025590
                                         00025600
                                         00025610
                                         00025620
                                         00025630
                                         00025640
                                         00025650
                                         00025660
                                         00025670
                                         00025680
                                         00025690
                                         00025700
                                         00025710
                                         00025720
                                         00025730
                                         00025740
                                         00025750

FBOLT CALCULATES THE INDIVIDUAL PLY FOUNDATION
MODULI AND THE INDIVIDUAL PLY LOADS

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)

CALCULATE DELEFF

H0PK=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(1.0*(AX**2+BX**2+BXX**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*X
210 CONTINUE
PLD=PLD+PLOADX
DELEFF=WORK/PLOADX

COMPUTE PLY STRESSES FROM LAMINATE STRAINS
(SIGMA)R,0,R0 = (Q)*(EPS)R,0,R0

NN=NPLY(K)
DO 100 J=1,NN
LP=IPLY(J,K)
THETA=(ANG(LP,K)+PSH)*RAD
I1=1

```

```

L12=NOUT          00025760
NCAS=1           00025770
CALL QMATEX(RAD,THETA,K,L11,L12,NCAS) 00025780
00025790
00025800
00025810
00025820
00025830
00025840
00025850
00025860
00025870
00025880
00025890
00025900
00025910
00025920
00025930
00025940
00025950
00025960
00025970
00025980
00025990
00026000
00026010
00026020
00026030
00026040
00026050
00026060
00026070
00026080
00026090
00026100
00026110
00026120
00026130
00026140
00026150
00026160
00026170
00026180
00026190
00026200
00026210
00026220
00026230
00026240
00026250
00026260
00026270
00026280
00026290
00026300
00026310
00026320
00026330
00026340
00026350

CCCCCCC
INTEGRATE AROUND CIRCULAR BOUNDARY FOR      00025900
INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION      00025910
MODULI                                           00025920
00025930
00025940
00025950
00025960
00025970
00025980
00025990
00026000
00026010
00026020
00026030
00026040
00026050
00026060
00026070
00026080
00026090
00026100
00026110
00026120
00026130
00026140
00026150
00026160
00026170
00026180
00026190
00026200
00026210
00026220
00026230
00026240
00026250
00026260
00026270
00026280
00026290
00026300
00026310
00026320
00026330
00026340
00026350

NNN=L12-1          00025900
PLOADX=0.          00025910
WK=0.              00025920
DO 70 I=L11,NNN    00025930
TH1=ATETAA(I+1)-ATETAA(I)          00025940
TH2=(ATETAA(I)+ATETAA(I+1))/2.    00025950
THETA=TH2*RAD      00025960
C=DCOS(THETA)      00025970
S=DSIN(THETA)      00025980
R=DSQRT(1.+(C**2/A**2+S**2/B**2))  00025990
FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*R*TH1*RAD*XH(K) 00026000
FORCRT=((ASIORT(I)+ASIORT(I+1))/2.)*R*TH1*RAD*XH(K) 00026010
PLOADX=PLOADX+FORCR*C-FORCRT*S 00026020
70 CONTINUE        00026030
FKI(J)=DABS(PLOADX/(H(K)*DELEFF)) 00026040
PLX(J+(K-1)*NPLY(1))=PLOADX 00026050
100 CONTINUE        00026060
NT=NMPLY(K)        00026070
NN=NPLY(K)          00026080
DO 310 I=1,NT      00026090
DO 310 II=1,NN      00026100
IF(IPLY(II,K).EQ.I) ANGK(I,K)=FKI(II) 00026110
IF(IPLY(II,K).EQ.I) PI(XPT(I)=PLX(II+(K-1)*NPLY(1)) 00026120
310 CONTINUE        00026130
NP=NMPLY(K)        00026140
DO 311 I=1,NP      00026150
AA=ANG(I,K)+PSH 00026160
311 CONTINUE        00026170
PLXTCT=0.0D0       00026180
IF(K.EQ.1) BLOAD=0. 00026190
TH=H(K)*NPLY(K)   00026200
BLOAD=(BSTR*DARCOS(-1.0D0)*BX*TH)/2.+BLOAD 00026210
IF(K.EQ.1) GO TO 611 00026220
NN=NPLY(1)+NPLY(2) 00026230
DO 212 I=1,NN      00026240
PLXTOT=PLXTOT+PLX(I) 00026250
212 CONTINUE        00026260
611 CONTINUE        00026270
RETURN             00026280
END               00026290
CCCCCCC
SUBROUTINE QMATEX(RAD,THETA,K,L11,L12,NCAS) 00026300
QMATEX PERFORMS BASIC STRESS AND STRAIN      00026310
TRANSFORMATIONS                                00026320
00026330
IMPLICIT REAL*8(A-H,O-Z)                      00026340
DIMENSION ASIGR(400),ASIORT(400),ASI01(400),ASI02(400),ASI06(400) 00026350

```

```

DIMENSION ATETAA(400),AEPSX(400),AEPSY(400),AEPXY(400) 00026360
DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400) 00026370
DIMENSION AEPS1(400) 00026380
DIMENSION ASX(400),ASXY(400) 00026390
COMMON//XXY1/A5X,ASXY 00026400
COMMON//MOD/E11,E22,ESS,PMU12,PMU21 00026410
COMMON//STRSS2/AEPS1 00026420
COMMON//STRESS/ASIGR,ASIGRT,ASIO1,ASIO2,ASI06 00026430
COMMON//CMT/ATETAA,AEPSX,AEPSY,AEPXY 00026440
COMMON//PSC3/SX,SXY 00026450
J=0 00026460
Q11=EI1(K)/(1.0-PMU12(K)*PMU21(K)) 00026470
Q12=(PMU21(K)*EI1(K))/(1.0-PMU12(K)*PMU21(K)) 00026480
Q22=Q22(K)/(1.0-PMU12(K)*PMU21(K)) 00026490
Q66=ESS(K) 00026500
C=DCOS(theta) 00026510
S=DSIN(theta) 00026520
B011=((Q11*(C**4))+(2.*((Q12+(2.*Q66))*((C**2)*(S**2)))+(Q22*(S**4))) 00026530
B012=((Q11+Q22-(4.*Q66))*((S**2)*(C**2)))+(Q12*(S**4+C**4)) 00026540
B013=((Q11-Q12-(2.*Q66))*((S*(C**3))))+((Q12-Q22+(2.*Q66))*((S**3)*(C**1))) 00026550
)
B022=(Q11*(S**4))+(2.*((Q12+(2.*Q66))*((S**2)*(C**2)))+(Q22*(C**4))) 00026570
B026=((Q11-Q12-(2.*Q66))*((S**3)*(C**1)))+((Q12-Q22+(2.*Q66))*((S**2)*(C**3))) 00026580
B066=((Q11+Q22-(2.*((Q12+Q66))))*((S**2)*(C**2)))+(Q66*((C**4)+(S**4))) 00026590
)
DO 40 I=111,L12 00026600
J=J+1 00026610
IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD 00026620
C=DCOS(theta) 00026630
S=DSIN(theta) 00026640
SIGX=BQ11*AEPSX(I)+BQ12*AEPSY(I)+BQ16*AEPXY(I) 00026650
SIGY=BQ12*AEPSX(I)+BQ22*AEPSY(I)+BQ26*AEPXY(I) 00026660
SIGXY=AQ16*AEPSX(I)+BQ26*AEPSY(I)+BQ66*AEPXY(I) 00026670
SX(J)=SIGX 00026680
SXY(J)=SIGXY 00026690
ASIGR(I)=SIGXX*C**2+SIGY*S**2+2.*SIGXY*SNC 00026700
ASIGRT(I)=-SIGXX*SNC+SIGY*C**2+SIGYY*(C**2-S**2) 00026710
ASIG1(J)=SIGXX*C**2+SIGY*S**2+2.*SNC*SIGXY 00026720
ASIG2(J)=SIGXX*S**2+SIGY*C**2-2.*SNC*SIGXY 00026730
ASIG6(J)=-C*S*SIGX+SIGY*CNS+(C**2-S**2)*SIGXY 00026740
AEPS1(J)=AEPSX(I)*C**2+AEPSY(I)*S**2+AEPXY(I)*SNC 00026750
40 CONTINUE 00026760
RETURN 00026770
END 00026780
CCCCC 00026790
C 00026800
SUBROUTINE CENTD(H,FASSS,FASBS,P) 00026810
00026820
00026830
00026840
00026850
00026860
00026870
IMPLICIT REAL*8(A-H,O-Z) 00026880
DIMENSION PLYK(100),BARK(100),BARU(100),F(100) 00026890
DIMENSION H(2),RF(2) 00026900
DIMENSION AII(100,100),A(2),B(2) 00026910
DIMENSION NPLY(2) 00026920
COMMON/PBB/PLYK,BARK,BARU 00026930
COMMON/RT/RF 00026940
COMMON/AFM/AII,F 00026950

```

```

COMMON/LYP/NPLY          00026960
NNN=NPLY(1)+NPLY(2)      00026970
00026980
00026990
00027000
00027010
00027020
00027030
00027040
00027050
00027060
00027070
00027080
00027090
00027100
00027110
00027120
00027130
00027140
00027150
00027160
00027170
00027180
00027190
00027200
00027210
00027220
00027230
00027240
00027250
00027260
00027270
00027280
00027290
00027300
00027310
00027320
00027330
00027340
00027350
00027360
00027370
00027380
00027390
00027400
00027410
00027420
00027430
00027440
00027450
00027460
00027470
00027480
00027490
00027500
00027510
00027520
00027530
00027540
00027550

CCC SET UP THE CENTRAL DIFFERENCE EQUATIONS          00027010
DO 3 I=1,100
DO 3 J=1,100
3 AII(I,J)=0.

CCC NECESSARY CONSTANTS ARE FORMED          00027020
DO 7 I=1,100
A(I)=H(1)*X2/FASSS
7 B(I)=H(I)*X4/FASBS
H12=H(1)/H(2)
A1=H(1)*X2/FASSS
A2=H(2)*X2/FASSS
NP=NPLY(1)+NPLY(2)

CCC SHEAR AT TOP OF PANEL EQUALS ZERO          00027030
AII(1,1)=1.
AII(1,2)=-(2.+A1*XPLYK(2))
AII(1,4)=2.+A1*XPLYK(2)
AII(1,5)=-1.
F(1)=0.0

CCC MOMENT CONDITION AT TOP          00027040
IF(RF(1).OE.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*X2.*H(1)*FASSS)*RM(-2.-A1*XPLYK(2)+(H(1)*X2
*X*FASSS)/FASBS)
AII(2,3)=-Z*(4.*H(1)*FASSS+(2*XH(1)*X2*XPLYK(1)*H(1)))
AII(2,4)=Z*X2.*H(1)*FASSS+RM(2.+A1*XPLYK(2)-(H(1)*X2
*X*FASSS)/FASBS)
AII(2,5)=-R
F(2)=Z*X2.*H(1)*X3*BARK(1)*BARU(1)

CCC GOVERNING EQUATIONS FOR THE TOP PLATE          00027050
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)*XPLYK(J-1)
GO TO 57
56 AII(I,J+1)=-4.-A(1)*XPLYK(2)
57 AII(I,J+2)=6.+(2.*A(1)+B(1))*XPLYK(J)
IF(J.EQ.N2) GO TO 61
AII(I,J+3)=-4.-A(1)*XPLYK(J+1)
GO TO 62
61 AII(I,J+3)=-4.-A(1)*XPLYK(NPLY(1)-1)

```

```

62 AII(I,J+4)=1.          00027560
  IF(J.EQ.1) GO TO 58      00027570
  IF(J.EQ.N2) GO TO 63      00027580
  F(I)=A(1)*BARK(J-1)*BARU(J-1)      00027590
  -(2.*A(1)+B(1))*BARK(J)*BARU(J)      00027600
  +A(1)*BARK(J+1)*BARU(J+1)      00027610
  GO TO 59      00027620
58 F(I)=2.*A(1)*BARK(2)*BARU(2)      00027630
  -(2.*A(1)+B(1))*BARK(1)*BARU(1)      00027640
  GO TO 59      00027650
63 F(I)=2.*A(1)*BARK(NPLY(1)-1)*BARU(NPLY(1)-1)      00027670
  -(2.*A(1)+B(1))*BARK(J)*BARU(J)      00027680
59 CONTINUE      00027690
55 CONTINUE      00027700
C C INTERFACE SHEAR ON TOP PLATE = P      00027710
C C I=NPLY(1)+3      00027720
J=NPLY(1)      00027730
AII(I,J)=1.      00027740
AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1))      00027750
AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)      00027760
AII(I,J+4)=-1.      00027770
F(I)=(-2.*H(1))**3*(P))/FASBS      00027780
C C SLOPE CONTINUITY      00027790
C C I=NPLY(1)+4      00027800
J=NPLY(1)      00027810
AII(I,J)=1.      00027820
AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS)      00027830
AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)-H(1)**2*FASSS/FASBS      00027840
AII(I,J+4)=-1.      00027850
AII(I,J+5)=-H12**3      00027860
AII(I,J+6)=H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSE/FASBS)      00027870
AII(I,J+8)=-H12**3*(2.+A2*PLYK(NPLY(1)+2)-H(2)**2*FASSS/FASBS)      00027880
AII(I,J+9)=H12**3      00027890
F(I)=0.      00027900
C C MOMENT CONTINUITY      00027910
C C I=NPLY(1)+5      00027920
J=NPLY(1)+1      00027930
AII(I,J)=1.      00027940
AII(I,J+1)=-(2.+A1*PLYK(NPLY(1)))      00027950
AII(I,J+2)=1.      00027960
AII(I,J+5)=-H12**2      00027970
AII(I,J+6)=H12**2*(2.+A2*PLYK(NPLY(1)+1))      00027980
AII(I,J+7)=-H12**2      00027990
F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*
  *BARU(NPLY(1)+1))      00028000
C C INTERFACE SHEAR ON BOTTOM PLATE      00028010
C C I=NPLY(1)+6      00028020
J=NPLY(1)+5      00028030
AII(I,J)=-1.      00028040
AII(I,J+1)=(2.+A2*PLYK(NPLY(1)+2))      00028050
AII(I,J+3)=-(2.+A2*PLYK(NPLY(1)+2))      00028060
AII(I,J+4)=1.      00028070

```

```

C F(I)=2.*H(2)*(N3+NP1)  GO TO 330
C GOVERNING EQUATIONS FOR THE BOTTOM PLATE
C N1=NPLY(1)+7
C N2=NPLY(1)+NPLY(2)+6
C DO 70 I=N1,N2
C   I=I-2
C   AII(I,J)=1.
C   IF(I.EQ.1) GO TO 71
C   AII(I,J-1)=-4.-1*(2.*A1*LYK(J-5)
C   J=J+2
C   71 AII(I,J+1)=-4.-A(2)*PLYK(NPLY(1)+2)
C   AII(I,J+2)=6.+A(2)*A(2)+B(2)*XPLYK(J-4)
C   IF(I.EQ.12) GO TO 73
C   AII(I,J-3)=-4.-1*(3)*PLYK(J-3)
C   J=J+2
C   73 AII(I,J+3)=-4.-1*(2)*PLYK(J-5)
C   J=J+4
C   I=I-1
C   IF(I.EQ.11) GO TO 73
C   IF(I.EQ.12) GO TO 77
C   F(I)=A(2)*BARK(J-5)*BARU(J-5)
C   -(2.*A(2)+3*(2))*BARK(J-6)*BARU(J-4)
C   +A(2)*BARK(J-3)*BARU(J-3)
C   GO TO 74
C 73 F(I)=2.*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2)
C   -(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
C   GO TO 74
C 77 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)
C   -(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
C 74 CONTINUE
C 70 CONTINUE
C SHEAR ON BOTTOM PLATE EQUALS ZERO
C NP=NPLY(1)+NPLY(2)
C I=NP+7
C J=NP+4
C AII(I,J)=-1.
C AII(I,J+1)=(2.+A2*XPLYK(NP-1))
C AII(I,J+3)=-(2.+A2*XPLYK(NP-1))
C AII(I,J+4)=1.
C F(I)=0.
C MUMENT BOUNDARY CONDITION ON BOTTOM PLATE
C I=NP+8
C IF(RF(2).GE.1.D10) GO TO 85
C Z=1.
C R=RF(2)
C GO TO 95
C 85 Z=0.
C R=1.
C 95 AII(I,J)=-R
C   AII(I,J+1)=Z*(2.*H(2)*FASSS)+R*(2.+A2*XPLYK(NP-1))
C   -(H(2)*2*FASSS/FASBS)
C   AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)*3*XPLYK(NP))
C   AII(I,J+3)=Z*(2.*H(2)*FASSS+R*(-2.-A2*XPLYK(NP-1))
C   +H(2)*2*FASSS/FASBS)
C   AII(I,J+4)=R

```



```

2112 CONTINUE          00029360
SX(L)=0.              00029370
2011 CONTINUE          00029380
PT=P                 00029390
N1=NPLY(1)+2          00029400
N2=NPLY(1)+7          00029410
NN=NPLY(1)+NPLY(2)+6  00029420
DO 1444 I=3,NN        00029430
J=I-2                00029440
U(J)=SX(I)            00029450
1444 CONTINUE          00029460
DO 1555 I=N2,NN        00029470
J=I-6                00029480
U(J)=SX(I)            00029490
1555 CONTINUE          00029500
NP=NPLY(1)+NPLY(2)    00029510
00029520
C COMPUTE AVERAGE RELATIVE DISPLACEMENTS
C IN TOP AND BOTTOM PLATES          00029530
C U1=DABS(U(1)-U(NPLY(1)))/2.      00029540
C U2=DABS(U(NPLY(1)+1)-U(NPLY(1)+NPLY(2)))/2. 00029550
RETURN               00029560
END                  00029570
00029580
00029590
00029600
00029610
00029620
00029630
00029640
00029650
00029660
00029670
00029680
00029690
00029700
00029710
00029720
00029730
00029740
00029750
00029760
00029770
00029780
00029790
00029800
00029810
00029820
00029830
00029840
00029850
00029860
00029870
00029880
00029890
00029900
00029910
00029920
00029930
00029940
00029950
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          00029960
C
C NELTOT=NEL1+NEL2          00029970
NS=NAVD               00029980
NSTS=4*NAVD             00029990
DO 10 I=1,NELTOT        00029990
NRNK=10                00029990
IF(NELTYP(I).EQ.3) NRNK=8
KJ=I

```

```

IF(KJ.GT.NEL) KJ=NEL          00029960
IF(NELTYP(I).EQ.1) GO TO 10   00029970
IC=0                          00029980
DO 20 J=1,3                   00029990
IC=IC+1                      00030000
DLT(IC)=U(NELDIS(I,J,1))    00030010
IC=IC+1                      00030020
DLT(IC)=U(NELDIS(I,J,2))    00030030
20 CONTINUE                   00030040
DO 30 K=1,NSTS               00030050
SUM=0.0D0                      00030060
DO 40 K2=1,NRK                00030070
SUM=SUM+ELSTSS(I,K,K2)*DLT(K2) 00030080
40 CONTINUE                   00030090
30 STRSS(K)=SUM               00030100
SUM1=0.0D0                      00030110
SUM2=0.0D0                      00030120
SUM3=0.0D0                      00030130
SUM4=0.0D0                      00030140
DO 50 J=1,NS                  00030150
SUM1=SUM1+STXSG(J)             00030160
SUM2=SUM2+STRSS(J+N)           00030170
SUM3=SUM3+STRSS(J+2*NS)        00030180
50 CONTINUE                   00030190
NS=2*NS                         00030200
DO 51 II=1,NS                  00030210
51 SUM4=SUM4+STRSS(II+(NNS)*NS) 00030220
AVES(I,1)=SUM1/NS               00030230
AVES(I,2)=SUM2/NS               00030240
AVES(I,3)=SUM3/NS               00030250
IF(I.LE.NEL1) THK=EI.THK(I)*NELPLS(1,LYPN(I)) 00030260
IF(I.GT.NEL1) THK=ELTHK(I)*NELPLS(2,LYPN(I)) 00030270
ELD=(SUM4/2.0)*ELNDTH(I)*THK 00030280
PRATIO=DABS((ELLOAD(I,1)+ELLOAD(I,2))/ELD) 00030290
C
C   SCALE AVERAGE STRESSES          00030300
C
AVES(I,1)=AVES(I,1)*PRATIO     00030310
AVES(I,2)=AVES(I,2)*PRATIO     00030320
AVES(I,3)=AVES(I,3)*PRATIO     00030330
10 CONTINUE                   00030340
C
C   COMPUTE JOINT FAILURE LOADS BASED ON      00030350
C   ELEMENT LOADS                           00030360
C
DO 100 I=1,NELTOT              00030370
IF(NELTYP(I).EQ.1) GO TO 100   00030380
DO 110 J=1,3                   00030390
N=J+1                          00030400
IF(J.EQ.1.AND.LTNCM.EQ.1) N=1  00030410
IF(J.EQ.1.AND.LTNCM.EQ.2) N=2  00030420
ELFAIL(I,J)=DABS(APP*PSMX(I,N)/AVES(I,J)) 00030430
110 CONTINUE                   00030440
100 CONTINUE                   00030450
C
C   SEARCH FOR LOWEST JOINT FAILURE LOAD      00030460
C
INNS=0                          00030470
FNG=1.0D10                      00030480
ING=0                          00030490
00030500
00030510
00030520
00030530
00030540
00030550

```

```

FSO=1.0D10          00030560
INBR=0              00030570
FBR=1.0D10          00030580
WRITE(6,356)         00030590
356 FORMAT(/, ' JOINT LOAD LEVELS CORRESPONDING TO NET ',/
      X' SECTION (NS), SHEAR-OUT (SO) AND BEARING (BR)',/
      X' FAILURES AT EVERY LOADED AND UNLOADED HOLE ',/
      X' ELEMENT ARE PREDICTED AS FOLLOWS!',/
      X' 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(FSU.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,18,2X,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

CCCC
C SUBROUTINE LINV2F (A,N,IA,AINV,IDOT,WKAREA,IER)
C
C DOUBLE PRECISION A(IA,N),AINV(IA,N),WKAREA(1),ZERO,ONE
C DATA ONE/1.0D0/,ZERO/0.0D0/
C FIRST EXECUTABLE STATEMENT
C INITIALIZE IER
C IER=0
C SET AINV TO THE N X N
C IDENTITY MATRIX
C
C DO 10 I = 1,N
C     DO 5 J = 1,N
C         AINV(I,J) = ZERO
5    CONTINUE
C         AINV(I,I) = ONE
10   CONTINUE
C COMPUTE THE INVERSE OF A
C CALL LEQT2F (A,N,N,IA,AINV,IDOT,WKAREA,IER)
C IF (IER EQ.0) GO TO 9005
9300 CONTINUE
C CALL VERTST (IER,6HLINV2F)
9005 RETURN
END
C

```

```

C SUBROUTINE LEQT2F (A,M,N,IA,B,IDOT,WKAREA,IER)          00031160
C
C DIMENSION      A(IA,1),B(IA,1),WKAREA(1)                00031170
C DOUBLE PRECISION A,B,WKAREA,D1,D2,WA                  00031180
C
C FIRST EXECUTABLE STATEMENT
C INITIALIZE IER                                         00031190
C
C IER=0                                                    00031200
C JER=0                                                    00031210
C J = N+1+N                                               00031220
C K = J+N                                                 00031230
C MM = K+N                                               00031240
C KK = 0                                                 00031250
C MM1 = MM-1                                             00031260
C JJ=1                                                   00031270
C DO 3 L=1,N                                              00031280
C   DO 3 I=1,N                                           00031290
C     WKAREA(JJ)=A(I,L)                                00031300
C     JJ=JJ+1                                             00031310
C
C 5 CONTINUE
C
C CALL LUDATN (WKAREA,N,N,A,IA,1DGT,D1,D2,WKAREA(J),WKAREA(K), 00031320
C               WA,IER)                                 00031330
C
C IF (IER.GT.128) GO TO 25                               00031340
C IF (IDOT.EQ.0 .OR. IER.NE.0) KK = 1                  00031350
C DO 15 I = 1,N                                         00031360
C
C           DECOMPOSE A
C           AX = B                                         00031370
C
C CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))    00031380
C
C           PERFORMS THE ELIMINATION PART OF
C           AX = B                                         00031390
C
C CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),1DGT,WKAREA(J),WKAREA(MM). 00031400
C               WKAREA(K),WKAREA(K),JER)                 00031410
C
C           REFINEMENT OF SOLUTION TO AX = B
C
C IF (KK.NE.0)                                            00031420
C
C CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),1DGT,WKAREA(J),WKAREA(MM). 00031430
C               WKAREA(K),WKAREA(K),JER)                 00031440
C
C           DO 10 II=1,N                                  00031450
C             B(II,I) = WKAREA(MM1+II)                   00031460
C
C 10 CONTINUE
C
C IF (JER.NE.0) GO TO 20                               00031470
C
C 15 CONTINUE
C
C GO TO 25                                              00031480
C
C 20 IER = 131                                           00031490
C
C 25 JJ=1
C
C   DO 30 J = 1,N                                     00031500
C     DO 30 I = 1,N                                     00031510
C       A(I,J)=WKARFA(JJ)                            00031520
C       JJ=JJ+1                                         00031530
C
C 30 CONTINUE
C
C IF (IER.EQ.0) GO TO 9005                           00031540
C
C 9000 CONTINUE
C
C CALL UERTST (IER,6HLEQT2F)
C
C 9005 RETURN
C
C END
C
C
C SUBROUTINE LUDATF (A,LU,N,IA,1DGT,D1,D2,IPVT,EQUIL,WA,IER) 00031550
C
C DIMENSION      A(IA,1),LU(IA,1),IPVT(1),EQUIL(1)        00031560
C DOUBLE PRECISION A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH, 00031570
C
C DATA          RN,WREL,BIGA,BIGP,SUM,AI,WI,T,TEST,Q        00031580
C               ZERO,ONE,FOUR,SIXTN,SIXTH/0.D0,1.D0,4.D0.      00031590
C
C

```

```

N          16.DD..0625D0/
C          FIRST EXECUTABLE STATEMENT
          INITIALIZATION
          IER = 0          00031760
          RN = N          00031770
          WREL = ZERO     00031780
          D1 = ONE         00031800
          D2 = ZERO        00031810
          BIGA = ZERO      00031820
          DO 10 I=1,N      00031830
          BIO = ZERO        00031840
          DO 3 J=1,N        00031850
          P = A(I,J)        00031860
          LU(I,J) = P       00031870
          P = DABS(P)       00031880
          IF (P .GT. BIGA) BIGA = P 00031890
          00031900
          00031910
          3 CONTINUE        00031920
          IF (BIO .GT. BIGA) BIGA = BIO 00031930
          IF (BIO .EQ. ZERO) GO TO 110 00031940
          EQU1(I) = ONE/BIGA 00031950
          00031960
          10 CONTINUE        00031970
          DO 103 J=1,N      00031980
          JMI = J-1         00031990
          IF (JMI .LT. 1) GO TO 40 00032000
          COMPUTE U(I,J), I=1,...,J-1 00032010
          DO 35 I=1,JMI    00032020
          SUM = LU(I,J)      00032030
          IMI = I-1         00032040
          IF (IDOT .EQ. 0) GO TO 25 00032050
          WITH ACCURACY TEST
          AI = DABS(SUM)    00032060
          WI = ZERO          00032070
          IF (IMI .LT. 1) GO TO 20 00032080
          DO 15 K=1,IMI    00032090
          T = LU(I,K)*LU(K,J) 00032100
          SUM = SUM+T        00032110
          WI = WI+DABS(T)    00032120
          15 CONTINUE        00032130
          LU(I,J) = SUM      00032140
          20 WI = WI+DABS(SUM) 00032150
          IF (AI .EQ. ZERO) AI = BIOA 00032160
          TEST = WI/AI       00032170
          IF (TEST .GT. WREL) WREL = TEST 00032180
          GO TO 35          00032190
          C          WITHOUT ACCURACY
          25 IF (IMI .LT. 1) GO TO 35 00032200
          DO 30 K=1,IMI    00032210
          SUM = SUM-LU(I,K)*LU(K,J) 00032220
          30 CONTINUE        00032230
          LU(I,J) = SUM      00032240
          35 CONTINUE        00032250
          40 P = ZERO         00032260
          COMPUTE U(J,J) AND L(I,J), I=J+1,...,N 00032270
          DO 70 I=J,N        00032280
          SUM = LU(I,J)      00032290
          IF (IDOT .EQ. 0) GO TO 55 00032300
          WITH ACCURACY TEST 00032310
          AI = DABS(SUM)    00032320
          WI = ZERO          00032330
          IF (JMI .LT. 1) GO TO 50 00032340
          00032350

```

```

      DC 45 K=1,JM1          00032360
      I = LU(I,K)*LU(K,J)    00032370
      SUM = SUM-T            00032380
      WI = WI+DABS(T)       00032390
45     CONTINUE              00032400
      LU(I,J) = SUM          00032410
50     WI = WI+DABS(SUM)    00032420
      IF (AI .EQ. ZERO) AI = BIGA 00032430
      TEST = WI/AI           00032440
      IF (TEST .GT. WREL) WREL = TEST 00032450
      GO TO 65               00032460
C      WITHOUT ACCURACY TEST 00032470
55     IF (JM .LT. 1) GO TO 65 00032480
      DO 60 K=1,JM1          00032490
      SUM = SUM-LU(I,K)*LU(K,J) 00032500
60     CONTINUE              00032510
      LU(I,J) = SUM          00032520
65     Q = EQUIL(I)*DABS(SUM) 00032530
      IF (P .GE. Q) GO TO 70  00032540
      P = Q                  00032550
      IMAX = I                00032560
70     CONTINUE              00032570
C      TEST FOR ALGORITHMIC SINGULARITY 00032580
      IF (RN+P .EQ. RN) GO TO 110 00032590
      IF (J .EQ. IMAX) GO TO 80  00032600
C      INTERCHANGE ROWS J AND IMAX 00032610
      D1 = -D1                00032620
DO 75 K=1,N          00032630
      P = LU(IMAX,K)          00032640
      LU(IMAX,K) = LU(J,K)    00032650
      LU(J,K) = P              00032660
75     CONTINUE              00032670
      EQUIL(IMAX) = EQUIL(J)  00032680
80     IPVT(J) = IMAX        00032690
      D1 = D1*LU(J,J)          00032700
85     IF (DABS(D1) .LE. ONE) GO TO 90  00032710
      D1 = D1*SIXTH           00032720
      D2 = D2+FOUR            00032730
      GO TO 85               00032740
90     IF (DABS(D1) .GE. SIXTH) GO TO 95  00032750
      D1 = D1*SIXTN           00032760
      D2 = D2-FOUR            00032770
      GO TO 90               00032780
95     CONTINUE              00032790
      JP1 = J+1               00032800
      IF (JP1 .GT. N) GO TO 105 00032810
C      DIVIDE BY PIVOT ELEMENT U(J,J) 00032820
      P = LU(J,J)             00032830
      DO 100 I=JP1,N          00032840
      LU(I,J) = LU(I,J)/P    00032850
100    CONTINUE              00032860
105   CONTINUE              00032870
C      PERFORM ACCURACY TEST 00032880
      IF (IDGT .EQ. 0) GO TO 9005 00032890
      P = 3*N+3               00032900
      WA = P*WREL              00032910
      IF (WA+10.D0**(-IDGT) .NE. WA) GO TO 9005 00032920
      IER = 34                 00032930
      GO TO 4000               00032940
C      ALGORITHMIC SINGULARITY 00032950

```

```

110 IER = 129          00032960
D1 = ZERO             00032970
D2 = ZERO             00032980
9000 CONTINUE
C                      PRINT ERRJR
9005 RETURN
END
C
C SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
DIMENSION A(IA,1),B(1),APVT(1),X(1)          00033060
DOUBLE PRECISION A,B,X,SUM,APVT              00033070
C                      FIRST EXECUTABLE STATEMENT
C                      SOLVE LY = B FOR Y          00033100
DO 5 I=1,N          00033110
5 X(I) = B(I)          00033120
IW = 0               00033130
DO 20 I=1,N          00033140
IP = APVT(I)          00033150
SUM = X(IP)          00033160
X(IP) = X(I)          00033170
IF (IW .EQ. 0) GO TO 15          00033180
IM1 = I-1             00033190
DO 10 J=IW,IM1        00033200
SUM = SUM-A(I,J)*X(J)  00033210
10 CONTINUE           00033220
GO TO 20              00033230
15 IF (SUM .NE. 0.0D0) IW = I          00033240
20 X(I) = SUM          00033250
C                      SOLVE UX = Y FOR X          00033260
DO 30 IB=1,N          00033270
I = N+1-IB            00033280
IP1 = I+1              00033290
SUM = X(I)              00033300
IF (IP1 .GT. N) GO TO 30          00033310
DO 25 J=IP1,N          00033320
SUM = SUM-A(I,J)*X(J)  00033330
25 CONTINUE           00033340
30 X(I) = SUM/A(I,I)    00033350
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) 00033410
DIMENSION APVT(1)          00033420
DIMENSION ACCXT(2)          00033430
DOUBLE PRECISION A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT 00033440
DATA ITMAX/75/,ZERO/0.0D0/ 00033450
C                      FIRST EXECUTABLE STATEMENT
IER=0
XNORM = ZERO          00033460
DO 10 I=1,N          00033470
XNORM = DMAX1(XNORM,DABS(X(I)))
10 CONTINUE           00033480
IF (XNORM .NE. ZERO) GO TO 20          00033490
IDGT = 50              00033500

```

```

20 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,ICUNIT,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=/
                         UNPACK NAME INTO NAMUPK
                         FIRST EXECUTABLE STATEMENT
        CALL USPKD (NAME,6,NAMUPK,NMTB)
        GET OUTPUT UNIT NUMBER
        CALL UGETIO(1,NIN,ICUNIT)
        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(ICUNIT,35) IER,NAMEQ,IEQ,NAMUPK
        IF (IEQDF.EQ.0) WRITE(ICUNIT,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(IONUNIT,40) IER,NAMEQ,IEQ,NAMUPK      00034160
IF (IEQDF.EQ.0) WRITE(IONUNIT,40) IER,NAMUPK                00034170
GO TO 30                                                 00034180
10 IF (IER.LE.32) GO TO 15
C               PRINT WARNING MESSAGE                      00034190
IF (LCVCL.LT.3) GO TO 30                                 00034200
IF (IEQDF.EQ.1) WRITE(IONUNIT,45) IER,NAMEQ,IEQ,NAMUPK      00034210
IF (IEQDF.EQ.0) WRITE(IONUNIT,45) IER,NAMUPK                00034220
GO TO 30                                                 00034230
15 CONTINUE
C               CHECK FOR UERSET CALL                         00034240
DO 20 I=1,6
IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25                     00034250
20 CONTINUE
LEVOLD = LEVEL
LEVEL = IER
IER = LEVOLD
IF (LEVEL.LT.0) LEVEL = 4                                00034260
IF (LEVEL.GT.4) LEVEL = 4                                00034270
GO TO 30                                                 00034280
25 CONTINUE
IF (LEVEL.LT.4) GO TO 30
C               PRINT NON-DEFINED MESSAGE                   00034290
IF (IEQDF.EQ.1) WRITE(IONUNIT,50) IER,NAMEQ,IEQ,NAMUPK      00034300
IF (IEQDF.EQ.0) WRITE(IONUNIT,50) IER,NAMUPK                00034310
30 IEQDF = 0
RETURN
35 FORMAT(19H XXX TERMINAL ERROR,10X,7H(IER = ,I3,
1 20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                  00034320
40 FORMAT(27H XXX WARNING WITH FIX ERROR,2X,7H(IER = ,I3,
1 20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                  00034330
45 FORMAT(18H XXX WARNING ERROR,11X,7H(IER = ,I3,
1 20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                  00034340
50 FORMAT(20H XXX UNDEFINED ERROR,9X,7H(IER = ,I3,
1 20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                  00034350
C               SAVE P FOR P = R CASE
C               P IS THE PAGE NAMUPK                         00034360
C               R IS THE ROUTINE NAMUPK                        00034370
C               00034380
C               00034390
C               00034400
C               00034410
C               00034420
C               00034430
C               00034440
C               00034450
C               00034460
C               00034470
C               00034480
C               00034490
C               00034500
C               00034510
C               00034520
C               00034530
C               00034540
C               00034550
C               00034560
C               00034570
C               00034580
C               00034590
C               00034600
C               00034610
C               00034620
C               00034630
C               00034640
C               00034650
C               00034660
C               00034670
C               00034680
C               00034690
C               00034700
C               00034710
C               00034720
C               00034730
C               00034740
C               00034750

C               SUBROUTINE UGETIO(IONP1,NIN,NOUT)
C               SPECIFICATIONS FOR ARGUMENTS
C               INTEGER          IONP1,NIN,NOUT
C               SPECIFICATIONS FOR LOCAL VARIABLES
C               INTEGER          NIND,NOUTD
C               DATA             NIND$/5/,NOUTD/6/
C               FIRST EXECUTABLE STATEMENT
C               IF (IONP1.EQ.3) GO TO 10
C               IF (IONP1.EQ.2) GO TO 5
C               IF (IONP1.NE.1) GO TO 9005
NIN = NIND
NOUT = NOUTD
GO TO 9005

```

```

      S NIND = NIN          00034760
      GO TO 9005            00034770
10 NOUTD = NOUT          00034780
9005 RETURN              00034790
                           00034800
                           00034810
                           00034820
                           00034830
                           00034840
                           00034850
                           00034860
                           00034870
                           00034880
                           00034890
                           00034900
                           00034910
                           00034920
                           00034930
                           00034940
                           00034950
                           00034960
                           00034970
                           00034980
                           00034990
                           00035000
                           00035010
                           00035020
                           00035030
                           00035040
                           00035050
                           00035060
                           00035070
                           00035080
                           00035090
                           00035100
                           00035110
                           00035120
                           00035130
                           00035140
                           00035150
                           00035160
                           00035170
                           00035180
                           00035190
                           00035200
                           00035210
                           00035220
                           00035230
                           00035240
                           00035250
                           00035260
                           00035270
                           00035280
                           00035290
                           00035300
                           00035310
                           00035320
                           00035330
                           00035340
                           00035350

C
C
      SUBROUTINE VXADD(A,ACC)          SPECIFICATIONS FOR ARGUMENTS
C
      DOUBLE PRECISION A,ACC(2)        SPECIFICATIONS FOR LOCAL VARIABLES
C
      DOUBLE PRECISION X,Y,Z,ZZ        FIRST EXECUTABLE STATEMENT
C
      X = ACC(1)
      Y = A
      IF (DABS(ACC(1)).OE.DABS(A)) GO TO 1
      X = A
      Y = ACC(1)
C
      Z = X+Y                         COMPUTE Z+ZZ = ACC(1)+A EXACTLY
      ZZ = (X-Z)+Y
C
      ZZ = ZZ+ACC(2)                   COMPUTE ZZ+ACC(2) USING DOUBLE
                                      PRECISION ARITHMETIC
C
      ACC(1) = Z+ZZ
      ACC(2) = (Z-ACC(1))+ZZ
      RETURN
      END

C
C
      SUBROUTINE VXMUL (A,B,ACC)          SPECIFICATIONS FOR ARGUMENTS
C
      DOUBLE PRECISION A,B,ACC(2)        SPECIFICATIONS FOR LOCAL VARIABLES
C
      DOUBLE PRECISION X,HA,TA,HB,TB
      INTEGER IX(2),I
      LOGICAL LI(1)
      EQUIVALENCE (X,LX(1)),IX(1),(I,LI(1))
      DATA I/0/
C
      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 HAXTB, HAXTB, TAXHR, AND TAXB
C AND CALL VXADD TO ACCUMULATE THE SUM 00035360
C                                         00035370
C                                         00035380
C                                         00035390
C                                         00035400
C                                         00035410
C                                         00035420
C                                         00035430
C                                         00035440
C                                         00035450
C                                         00035460
C                                         00035470
C                                         00035480
C                                         00035490
C                                         00035500
C                                         00035510
C                                         00035520
C                                         00035530
C                                         00035540
C                                         00035550
C                                         00035560
C                                         00035570
C                                         00035580
C                                         00035590
C                                         00035600
C                                         00035610
C                                         00035620
C                                         00035630
C                                         00035640
C                                         00035650
C                                         00035660
C                                         00035670
C                                         00035680
C                                         00035690
C                                         00035700
C                                         00035710
C                                         00035720
C                                         00035730
C                                         00035740
C                                         00035750
C                                         00035760
C                                         00035770
C                                         00035780
C                                         00035790
C                                         00035800
C                                         00035810
C                                         00035820
C                                         00035830
C                                         00035840
C                                         00035850
C                                         00035860
C                                         00035870
C                                         00035880
C                                         00035890
C                                         00035900
C                                         00035910
C                                         00035920
C                                         00035930
C                                         00035940
C                                         00035950

C SUBROUTINE VXSTO (ACC,D)          SPECIFICATIONS FOR ARGUMENTS 00035520
C DOUBLE PRECISION ACC(2),D          FIRST EXECUTABLE STATEMENT 00035530
C D = ACC(1)+ACC(2)                  00035540
C RETURN                             00035550
C END                                00035560
C                                     00035570
C                                     00035580
C                                     00035590
C                                     00035600
C                                     00035610
C                                     00035620
C                                     00035630
C                                     00035640
C                                     00035650
C                                     00035660
C                                     00035670
C                                     00035680
C                                     00035690
C                                     00035700
C                                     00035710
C                                     00035720
C                                     00035730
C                                     00035740
C                                     00035750
C                                     00035760
C                                     00035770
C                                     00035780
C                                     00035790
C                                     00035800
C                                     00035810
C                                     00035820
C                                     00035830
C                                     00035840
C                                     00035850
C                                     00035860
C                                     00035870
C                                     00035880
C                                     00035890
C                                     00035900
C                                     00035910
C                                     00035920
C                                     00035930
C                                     00035940
C                                     00035950

C SUBROUTINE ZRPOLY (A,NDE0,Z,IER)  SPECIFICATIONS FOR ARGUMENTS 00035620
C INTEGER NDE0,IER                  SPECIFICATIONS FOR LOCAL VARIABLES 00035630
C DOUBLE PRECISION A(1),Z(1)          N,NN,J,JJ,I,NM1,ICNT,N2,L,NZ,NPI 00035640
C                                     ETA,RMRE,RINFP,REPSP,RADIX,RLO,XX,YY,SINR, 00035650
C                                     COSR,RMAX,RMIN,X,SC,XM,FF,DX,DF,BND,XXX,ARE 00035660
C                                     PT(101) 00035670
C                                     TEMP(101),P(101),QP(101),RK(101),QK(101), 00035680
C                                     SVK(101) 00035690
C                                     SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00035700
C                                     A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00035710
C                                     T,AA,BB,CC,FACTOR,REPSR1,ZERO,ONE,FN 00035720
C                                     ZEROK 00035730
C                                     P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00035740
C                                     A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00035750
C                                     THE FOLLOWING STATEMENTS SET MACHINE CONSTANTS USED IN VARIOUS PARTS OF 00035760
C                                     THE PROGRAM. THE MEANING OF THE FOUR CONSTANTS ARE - REPSR1 THE MAXIMUM RELATIVE REPRESENTATION ERROR WHICH CAN BE DESCRIBED AS THE SMALLEST POSITIVE FLOATING POINT NUMBER SUCH THAT 1.+REPSR1 IS GREATER THAN 1. RINFP THE LARGEST FLOATING-POINT NUMBER 00035770
C                                     REPSP THE SMALLEST POSITIVE FLOATING-POINT NUMBER IF THE EXPONENT RANGE DIFFERS IN SINGLE AND DOUBLE PRECISION THEN REPSP AND RINFP SHOULD INDICATE THE SMALLER RANGE 00035780
C                                     00035790
C                                     00035800
C                                     00035810
C                                     00035820
C                                     00035830
C                                     00035840
C                                     00035850
C                                     00035860
C                                     00035870
C                                     00035880
C                                     00035890
C                                     00035900
C                                     00035910
C                                     00035920
C                                     00035930
C                                     00035940
C                                     00035950

```

C	RADIX THE BASE OF THE FLOATING-POINT NUMBER SYSTEM USED	00035960 00035970
C	RINFP/Z7FFFFFF/ REPSP/Z00100000/	00035980 00035990
C	RADIX/16.0/	00036000
C	RE: '1/Z34100000G0000000/	00036010
C	ZERO/0.0D0/, ONE/1.0D0/	00036020
C	ZRPOLY USES SINGLE PRECISION CALCULATIONS FOR SCALING, BOUNDS AND ERROR CALCULATIONS.	00036030 00036040 00036050
C	FIRST EXECUTABLE STATEMENT	00036060
IER = 0		00036070
IF (NDEQ .GT. 100 .OR. NDFG .LT. 1) GO TO 165		00036080
ETA = REPSR1		00036090
ARE = ETA		00036100
RMRE = ETA		00036110
RLO = REPSP/ETA		00036120
C	INITIALIZATION OF CONSTANTS FOR SHIFT ROTATION	00036130 00036140
C	XX = .7071068	00036150
C	YY = -XX	00036160
C	SINR = .9975641	00036170
C	COSR = -.06975647	00036180
C	N = NDEG	00036190
C	NN = N+1	00036200
C	ALGORITHM FAILS IF THE LEADING COEFFICIENT IS ZERO.	00036210 00036220
C	IF (A(1),NE,ZERO) GO TO 5	00036230
C	IER = 130	00036240
C	GO TO 9000	00036250
C	REMOVE THE ZEROS AT THE ORIGIN IF ANY	00036260 00036270
5 IF (A(NN),NE,ZERO) GO TO 10		00036280
C	J = NDEG+N+1	00036290
C	JJ = J+NDEG	00036300
C	Z(J) = ZERO	00036310
C	Z(JJ) = ZERO	00036320
C	NN = NN-1	00036330
C	N = N-1	00036340
C	IF (NN,EQ,1) GO TO 9005	00036350
C	GO TO 5	00036360
C	10 DO 15 I=1,NN	00036370
C	P(I) = A(I)	00036380
C	15 CONTINUE	00036390
C	20 IF (N.GT.2) GO TO 30	00036400
C	IF (N.LT.1) GO TO 9005	00036410 00036420 00036430
C	CALCULATE THE FINAL ZERO OR PAIR OF ZEROS	00036440 00036450
C	IF (N.EQ.2) GO TO 25	00036460
C	Z(NDEG) = -P(2)/P(1)	00036470
C	Z(NDEG+NDEG) = ZERO	00036480
C	GO TO 145	00036490
C	25 CALL ZRPQLI (P(1),P(2),P(3),Z(NDEG-1),Z(NDEG+NDEG-1),Z(NDEG), 1 Z(NDEG+NDEG))	00036500 00036510 00036520
C	GO TO 145	00036530 00036540 00036550
C	FIND LARGEST AND SMALLEST MODULI OF COEFFICIENTS.	00036560 00036570
30 RMAX = 0.		00036580

```

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 = RL0/RMIN
IF (SC.GT.1.0) GO TO 40
IF (RMAX.LT.10.) GO TO 55
IF (SC.EQ.0.) SC = REPSPKRADIXXRADIX
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*X*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)))/L)
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.0.00036940
65 XM = XM+.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*XM+PT(I)
  DF = DF*XM+FF
85 CONTINUE
FF = FF*XM+PT(NN)
DX = FF/DF
X = X-DX
GO TO 60

```

```

90 BND = X          00037160
C
C COMPUTE THE DERIVATIVE AS THE INITIAL 00037170
C K POLYNOMIAL AND DO 5 STEPS WITH 00037180
C NO SHIFT 00037190
00037200
00037210
00037220
00037230
00037240
00037250
00037260
00037270
00037280
00037290
00037300
00037310
00037320
00037330
00037340
00037350
00037360
00037370
00037380
00037390
00037400
00037410
00037420
00037430
00037440
00037450
00037460
00037470
00037480
00037490
00037500
00037510
00037520
00037530
00037540
00037550
00037560
00037570
00037580
00037590
00037600
00037610
00037620
00037630
00037640
00037650
00037660
00037670
00037680
00037690
00037700
00037710
00037720
00037730
00037740
00037750

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 00037310
C VALUE OF K AT 0 IS NONZERO 00037320
T = -AA/LC
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)*ETAN10.
GO TO 115
C USE UNSCALED FORM OF RECURRENCE 00037410
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 X FOR RSTARTS WITH NEW SHIFTS 00037490
DO 120 I=1,N
120 TEMP(I) = RK(I)
C LOOP TO SELECT THE QUADRATIC 00037510
C CORRESPONDING TO EACH NEW SHIFT 00037520
00037530
00037540
00037550
00037560
00037570
00037580
00037590
00037600
00037610
00037620
00037630
00037640
00037650
00037660
00037670
00037680
00037690
00037700
00037710
00037720
00037730
00037740
00037750

XXX = COSR*XX-SINR*YY
YY = SINR*XX+COSR*YY
XX = XXX
SR = BND*XX
SI = BND*YY
U = -SR-SR
V = BND*BND

CALL ZRPOLB (20*ICNT,NZ)
IF (NZ.EQ.0) GO TO 130
C SECOND STAGE CALCULATION, FIXED 00037680
C QUADRATIC 00037690
00037700
00037710
00037720
00037730
00037740
00037750

THE SECOND STAGE JUMPS DIRECTLY TO 00037720
ONE OF THE THIRD STAGE ITERATIONS 00037730
AND RETURNS HERE IF SUCCESSFUL. 00037740
DEFLATE THE POLYNOMIAL, STORE THE 00037750

```

```

C          ZERO OR ZEROS AND RETURN TO THE      00037760
          MAIN ALGORITHM.                  00037770
          00037780
          00037790
          00037800
          00037810
          00037820
          00037830
          00037840
          00037850
          00037860
          00037870
          00037880
          00037890
          00037900
          00037910
          00037920
          00037930
          00037940
          00037950
          00037960
          00037970
          00037980
          00037990
          00038000
          00038010
          00038020
          00038030
          00038040
          00038050
          00038060
          00038070
          00038080
          00038090
          00038100
          00038110
          00038120
          00038130
          00038140
          00038150
          00038160
          00038170
          00038180
          00038190
          00038200
          00038210
          00038220
          00038230
          00038240
          00038250
          00038260
          00038270
          00038280
          00038290
          00038300
          00038310
          00038320
          00038330
          00038340
          00038350

C          IF THE ITERATION IS UNSUCCESSFUL      00037900
          ANOTHER QUADRATIC IS CHOSEN AFTER      00037910
          RESTORING K.                         00037920
          00037930
          00037940
          00037950
          00037960
          00037970
          00037980
          00037990
          00038000
          00038010
          00038020
          00038030
          00038040
          00038050
          00038060
          00038070
          00038080
          00038090
          00038100
          00038110
          00038120
          00038130
          00038140
          00038150
          00038160
          00038170
          00038180
          00038190
          00038200
          00038210
          00038220
          00038230
          00038240
          00038250
          00038260
          00038270
          00038280
          00038290
          00038300
          00038310
          00038320
          00038330
          00038340
          00038350

C          DO 135 I=1,N                         00037900
          RK(I) = TEMP(I)                      00037910
          140 CONTINUE                           00037920
          00037930
          00037940
          00037950
          00037960
          00037970
          00037980
          00037990
          00038000
          00038010
          00038020
          00038030
          00038040
          00038050
          00038060
          00038070
          00038080
          00038090
          00038100
          00038110
          00038120
          00038130
          00038140
          00038150
          00038160
          00038170
          00038180
          00038190
          00038200
          00038210
          00038220
          00038230
          00038240
          00038250
          00038260
          00038270
          00038280
          00038290
          00038300
          00038310
          00038320
          00038330
          00038340
          00038350

C          RETURN WITH FAILURE IF NO      00037900
          CONVERGENCE WITH 20 SHIFTS      00037910
          00037920
          00037930
          00037940
          00037950
          00037960
          00037970
          00037980
          00037990
          00038000
          00038010
          00038020
          00038030
          00038040
          00038050
          00038060
          00038070
          00038080
          00038090
          00038100
          00038110
          00038120
          00038130
          00038140
          00038150
          00038160
          00038170
          00038180
          00038190
          00038200
          00038210
          00038220
          00038230
          00038240
          00038250
          00038260
          00038270
          00038280
          00038290
          00038300
          00038310
          00038320
          00038330
          00038340
          00038350

C          IER = 131                           00037900
          00037910
          00037920
          00037930
          00037940
          00037950
          00037960
          00037970
          00037980
          00037990
          00038000
          00038010
          00038020
          00038030
          00038040
          00038050
          00038060
          00038070
          00038080
          00038090
          00038100
          00038110
          00038120
          00038130
          00038140
          00038150
          00038160
          00038170
          00038180
          00038190
          00038200
          00038210
          00038220
          00038230
          00038240
          00038250
          00038260
          00038270
          00038280
          00038290
          00038300
          00038310
          00038320
          00038330
          00038340
          00038350

C          145 DO 150 I=1,NDEQ               00037900
          NPI = NDEQ+I                         00037910
          P(I) = Z(NPI)                        00037920
          150 CONTINUE                           00037930
          N2 = NDEQ+NDEQ                       00037940
          J = NDEQ                            00037950
          DO 155 I=1,NDEQ                     00037960
          Z(N2-1) = Z(J)                      00037970
          Z(N2) = P(J)                        00037980
          N2 = N2-2                           00037990
          J = J-1                             00038000
          155 CONTINUE                           00038010
          IF (IER .EQ. 0) GO TO 9005          00038020
          C          SET UNFOUND ROOTS TO MACHINE INFINITY 00038030
          N2 = 2*(NDEQ-NN)+3                 00038040
          DO 160 I=1,N                         00038050
          Z(N2) = RINFP                        00038060
          Z(N2+1) = RINFP                      00038070
          N2 = N2+2                           00038080
          160 CONTINUE                           00038090
          GO TO 9000                           00038100
          165 IER = 129                         00038110
          9000 CONTINUE                           00038120
          CALL UERTST (IER,6HZRPOLY)          00038130
          9005 RETURN                           00038140
          END                                00038150
          C          SUBROUTINE ZRPQLB (L2,NZ)        00038260
          C          INTEGER L2,NZ                  SPECIFICATIONS FOR ARGUMENTS 00038270
          C          INTEGER N,NN,J,ITYPE,I,IFLAG  SPECIFICATIONS FOR LOCAL VARIABLES 00038280
          C          REAL ARE,BETAS,BETAV,ETA,OSS,OTS,OTV,OVV,RMRE,SS, 00038290
          C          TS,TSS,TV,TVV,VV              00038300
          C          00038310
          C          00038320
          C          00038330
          C          00038340
          C          00038350

```

```

DOUBLE PRECISION P(101),QP(101),PK(101),QK(101),SVK(101) 00038360
DOUBLE PRECISION SR,SI,U,V,RA,RD,C,D,A1,A2,A3,          00038370
1 A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,                      00038380
2 SVU,SVV,UI,VI,S,ZERO                                     00038390
LOGICAL VPASS,SPASS,VTRY,STRY                            00038400
COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00038410
1 A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN        00038420
DATA ZERO/0.0D0/                                         00038430
C FIRST EXECUTABLE STATEMENT                           00038440
NZ = 0                                                 00038450
C COMPUTES UP TO L2 FIXED SHIFT                      00038460
C K-POLYNOMIALS, TESTING FOR                         00038470
C CONVERGENCE IN THE LINEAR OR                      00038480
C QUADRATIC CASE. INITIATES ONE OF                 00038490
C THE VARIABLE SHIFT ITERATIONS AND                00038500
C RETURNS WITH THE NUMBER OF ZEROS                 00038510
C FOUND.                                            00038520
L2 - LIMIT OF FIXED SHIFT STEPS                   00038530
NZ - NUMBER OF ZEROS FOUND                         00038540
BETAV = .25                                         00038550
BETAS = .25                                         00038560
OSS = SR                                           00038570
OVV = V                                            00038580
C EVALUATE POLYNOMIAL BY SYNTHETIC                  00038590
C DIVISION                                         00038600
CALL ZRPQLH (NN,U,V,P,QP,RA,RB)                   00038610
CALL ZRPQLE (ITYPE)                                00038620
DO 40 J=1,L2                                         00038630
C CALCULATE NEXT K POLYNOMIAL AND                 00038640
C ESTIMATE V                                         00038650
CALL ZRPQLF (ITYPE)                                00038660
CALL ZRPQLE (ITYPE)                                00038670
CALL ZRPQLQ (ITYPE,UI,VI)                           00038680
VV = VI                                           00038690
C ESTIMATE S                                         00038700
SS = 0.                                              00038710
IF (RK(N).NE.ZERO) SS = -P(NN)/RK(N)             00038720
TV = 1.                                              00038730
TS = 1.                                              00038740
IF (J.EQ.1.OR.ITYPE.EQ.3) GO TO 35              00038750
C COMPUTE RELATIVE MEASURES OF                     00038760
C CONVERGENCE OF S AND V SEQUENCES               00038770
IF (VV.NE.0.) TV = ABS((VV-OVV)/VV)               00038780
IF (SS.NE.0.) TS = ABS((SS- USS)/SS)              00038790
C IF DECREASING, MULTIPLY TWO MOST               00038800
C RECENT CONVERGENCE MEASURES                    00038810
TVV = 1.                                             00038820
IF (TV.LT.DTV) TVV = TV*DTV                       00038830
TSS = 1.                                             00038840
IF (TS.LT.DTS) TSS = TSS*DTS                       00038850
C COMPARE WITH CONVERGENCE CRITERIA            00038860
VPASS = TVV.LT.BETAV                             00038870
SPASS = TSS.LT.BETAS                             00038880
IF (.NOT.(SPASS.OR.VPASS)) GO TO 35              00038890
C AT LEAST ONE SEQUENCE HAS PASSED THE          00038900
C CONVERGENCE TEST. STORE VARIABLES             00038910
C BEFORE ITERATING                               00038920
SVU = U                                           00038930
SVV = V                                           00038940
DO 5 I=1,N                                         00038950

```

5	SVK(I) = RK(I)	00038960
C	S = SS	00038970
C	VTRY = .FALSE.	00038980
	STRY = .FALSE.	00038990
10	IF (SPASS.AND.((.NOT.VPASS).OR.TSS.LT.TVV)) GO TO 20	00039000
	CALL ZRPQLC (UI,VI,NZ)	00039010
	IF (NZ.GT.0) RETURN	00039020
C	VTRY = .TRUE.	00039030
C	BETAV = BETAV*.25	00039040
C	TRY LINEAR ITERATION IF IT HAS NOT	00039050
C	BEEN TRIED AND THE S SEQUENCE IS	00039060
C	CONVERGING	00039070
C	IF (STRY.OR.(.NOT.SPASS)) GO TO 25	00039080
15	DO 15 I=1,N	00039090
20	RK(I) = SVK(I)	00039100
	CALL ZRPQLD (S,NZ,IFLAG)	00039110
	IF (NZ.GT.0) RETURN	00039120
C	LINEAR ITERATION HAS FAILED. FLAG	00039130
C	THAT IT HAS BEEN TRIED AND	00039140
C	DECREASE THE CONVERGENCE	00039150
C	CRITERION.	00039160
C	STRY = .TRUE.	00039170
C	BETAS = BETAS*.25	00039180
C	IF (IFLAG.EQ.0) GO TO 25	00039190
C	IF !LINEAR ITERATION SIGNALS AN	00039200
C	ALMOST DOUBLE REAL ZERO ATTEMPT	00039210
C	QUADRATIC ITERATION	00039220
C	UI = -(S+S)	00039230
C	VI = S*3	00039240
C	GO TO 10	00039250
C	RESTORE VARIABLES	00039260
25	U = SVU	00039270
	V = SVV	00039280
	DO 30 I=1,N	00039290
30	RK(I) = SVK(I)	00039300
C	TRY QUADRATIC ITERATION IF IT HAS	00039310
C	NOT BEEN TRIED AND THE V SEQUENCE	00039320
C	IS CONVERGING	00039330
C	IF (VPASS.AND.(.NOT.VTRY)) GO TO 10	00039340
C	RECOMPUTE QP AND SCALAR VALUES TO	00039350
C	CONTINUE THE SECOND STAGE	00039360
C	CALL ZRPQLH (NN,U,V,P,QP,RA,RB)	00039370
C	CALL ZRPQLE (ITYPE)	00039380
35	OVV = VV	00039390
	OSS = SS	00039400
	OTV = TV	00039410
	OTS = TS	00039420
40	CONTINUE	00039430
	RETURN	00039440
	END	00039450
C	SUBROUTINE ZRPQLC (UU,VV,NZ)	00039460
C	SPECIFICATIONS FOR ARGUMENTS	00039470
		00039480
		00039490
		00039500
		00039510
		00039520
		00039530
		00039540
		00039550

```

C      INTEGER          NZ          00039560
C      DOUBLE PRECISION UU,VV       00039570
C      DOUBLE PRECISION N,NN,J,I,ITYPE 00039580
C      REAL             ARE,EE,ETA,OMP,RELSTP,RMP,RMRE,T,ZM 00039590
C      DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00039600
C      DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00039610
C      A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00039620
C      LOGICAL          TRIED      00039630
C      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00039640
C      DATA             A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00039650
C      DATA             ZERO,PT01,ONE/0.000,0.01D0,1.00D0/ 00039660
C      DATA             FIRST EXECUTABLE STATEMENT 00039670
C      NZ = 0           00039680
C      VARIABLE-SHIFT K-POLYNOMIAL 00039690
C      ITERATION FOR A QUADRATIC FACTOR 00039700
C      CONVERGES ONLY IF THE ZEROS ARE 00039710
C      EQUIMODULAR OR NEARLY SO 00039720
C      UU,VV - COEFFICIENTS OF STARTING 00039730
C      QUADRATIC 00039740
C      NZ - NUMBER OF ZERO FOUND 00039750
C      TRIED = .FALSE. 00039760
C      U = UU 00039770
C      V = VV 00039780
C      J = 0 00039790
C      MAIN LOOP 00039800
C      3 CALL ZRPQLI (ONE,U,V,SZR,SZI,RLZR,RLZI) 00039810
C      RETURN IF ROOTS OF THE QUADRATIC ARE 00039820
C      REAL AND NOT CLOSE TO MULTIPLE OR 00039830
C      NEARLY EQUAL AND OF OPPOSITE SIGN 00039840
C      IF ( DABS(DABS(SZR)-DABS(RLZR)) .GT. PT01*DABS(RLZR) ) RETURN 00039850
C      EVALUATE POLYNOMIAL BY QUADRATIC 00039860
C      SYNTHETIC DIVISION 00039870
C      CALL ZRPQLH (NN,U,V,P,QP,KA,RB) 00039880
C      RMP = DABS(RA-SZR*RB)+DABS(SZI*RB) 00039890
C      COMPUTE A RIGOROUS BOUND ON THE 00039900
C      ROUNDING ERROR IN EVALUTING P 00039910
C      ZM = SQRT(ABS(SNGL(V))) 00039920
C      EE = 2.*ABS(SNGL(QP(1))) 00039930
C      T = -SZR*RB 00039940
C      DO 10 I=2,N 00039950
C      10 EE = EE*ZM+ABS(SNGL(QP(I))) 00039960
C      EE = EE*ZM+ABS(SNGL(RA)+T) 00039970
C      EE = (5.*RMRE+4.*ARE)*EE-(5.*4RMRE+2.*ARE)*(ABS(SNGL(RA)+T)+ 00039980
C      ABS(SNGL(RB))/ZM)+2.*ARE*ABS(T) 00039990
C      ITERATION HAS CONVERGED SUFFICIENTLY 00040000
C      IF THE POLYNOMIAL VALUE IS LESS 00040010
C      THAN 20 TIMES THIS BOUND 00040020
C      IF (RMP.GT.20.*EE) GO TO 15 00040030
C      NZ = 2 00040040
C      RETURN 00040050
C      15 J = J+1 00040060
C      STOP ITERATION AFTER 20 STEPS 00040070
C      IF (J.GT.20) RETURN 00040080
C      IF (J.LT.2) GO TO 25 00040090
C      IF (RELSTP.GT..01.OR.RMP.LT.OMP.OR.TRIED) GO TO 25 00040100
C      A CLUSTER APPEARS TO BE STALLING THE 00040110
C      CONVERGENCE. FIVE FIXED SHIFT 00040120
C      STEPS ARE TAKEN WITH A U,V CLOSE 00040130
C      00040140
C      00040150

```

```

C          TO THE CLUSTER           00040160
IF (RELSTP.LT.ETA) RELSTP = ETA
RELSTP = SQRT(RELSTP)           00040170
U = U-UXRELSTP                00040180
V = V+VXRELSTP                00040190
CALL ZRPQLH (NN,U,V,P,QP,RA,RB) 00040200
DO 20 I=1,5                    00040210
  CALL ZRPQLE (ITYPE)          00040220
  CALL ZRPQLF (ITYPE)          00040230
20 CONTINUE                     00040240
  TRIED = .TRUE.
  J = 0                         00040250
25 OMP = RMP                   00040260
C          CALCULATE NEXT K POLYNOMIAL AND NEW    00040270
C          U AND V                 00040280
C          CALL ZRPQLE (ITYPE)          00040290
C          CALL ZRPQLF (ITYPE)          00040300
C          CALL ZRPQLE (ITYPE)          00040310
C          CALL ZRPQLG (ITYPE,UI,VI)    00040320
C          IF VI IS ZERO THE ITERATION IS NOT      00040330
C          CONVERGING                      00040340
C          IF (VI.EQ.ZERO) RETURN          00040350
RELSTP = DBABS((VI-V)/VI)        00040360
U = UI                         00040370
V = VI                         00040380
GO TO 5                         00040390
END                           00040400
CCC
C          SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)          00040410
C          SPECIFICATIONS FOR ARGUMENTS            00040420
C          INTEGER          NZ,IFLAG             00040430
C          DOUBLE PRECISION   SSS                00040440
C          SPECIFICATIONS FOR LOCAL VARIABLES       00040450
C          INTEGER          N,NN,J,X             00040460
C          REAL              ARE,EE,ETA,OMP,RMP,RMS,RMRE 00040470
C          DOUBLE PRECISION  P(101),QP(101),RK(101),QK(101),SVK(101) 00040480
C          DOUBLE PRECISION  SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00040490
C          A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00040500
C          PY,RKV,T,S,ZERO,PT001               00040510
C          P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00040520
C          A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00040530
C          DATA              ZERO/0.0D0/,PT001/0.001D0/ 00040540
C          VARIABLE-SHIFT H POLYNOMIAL           00040550
C          ITERATION FOR A REAL ZERO SSS -      00040560
C          STARTING ITERATE                  00040570
C          NZ - NUMBER OF ZERO FOUND          00040580
C          IFLAG - FLAG TO INDICATE A PAIR OF 00040590
C          ZEROS NEAR REAL AXIS
C          FIRST EXECUTABLE STATEMENT          00040600
CCC
C          NZ = 0                         00040610
C          S = SSS                       00040620
C          IFLAG = 0                     00040630
C          J = 0                         00040640
C          5 PV = P(1)                   00040650
C          QP(1) = PV                  00040660
C          DO 10 I=2,NN                00040670
C          EVALUATE P AT S             00040680
C          10 CONTINUE                   00040690
C          GO TO 5                     00040700
C          MAIN LOOP                  00040710
C          5 EVALUATE P AT S           00040720
C          DO 10 I=2,NN                00040730
C          EVALUATE P AT S             00040740
C          10 CONTINUE                   00040750

```

```

PV = PV*S+P(I)          00040760
QP(I) = PV              00040770
10 CONTINUE               00040780
RMP = DABS(PV)           00040790
C COMPUTE A RIGOROUS BOUND ON THE 00040800
   ERROR IN EVALUATING P      00040810
C RMS = DABS(S)           00040820
EE = (RMRE/(ARE+RMRE))*ABS(SNGL(QP(1))) 00040830
DO 15 I=2,NN             00040840
15 EE = EE*RMS+ABS(SNGL(QP(I))) 00040850
CCC ITERATION HAS CONVERGED SUFFICIENTLY 00040860
   IF THE POLYNOMIAL VALUE IS LESS 00040870
   THAN 20 TIMES THIS BOUND       00040880
C IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20 00040890
NZ = 1                   00040900
SZR = S                  00040910
SZI = ZERO                00040920
RETURN                  00040930
20 J = J+1                00040940
C STOP ITERATION AFTER 10 STEPS 00040950
IF (J.GT.10) RETURN      00040960
IF (J.LT.2) GO TO 25     00040970
C IF (DABS(T).GT.PT001*DABS(S-T).OR.RMP.LE.OMP) GO TO 25 00040980
   A CLUSTER OF ZEROS NEAR THE REAL 00040990
   AXIS HAS BEEN ENCOUNTERED RETURN 00041000
CCC WITH IFLAG SET TO INITIATE A 00041010
   QUADRATIC ITERATION          00041020
C IFLAG = 1                00041030
SSS = S                  00041040
RETURN                  00041050
C RETURN IF THE POLYNOMIAL VALUE HAS 00041060
   INCREASED SIGNIFICANTLY        00041070
C 25 OMP = RMP             00041080
C COMPUTE T, THE NEXT POLYNOMIAL, AND 00041090
   THE NEW ITERATE              00041100
C RKV = RK(1)              00041110
QK(1) = RKV              00041120
DO 30 I=2,N               00041130
   RKV = RKV*S+RK(I)         00041140
   QK(I) = RKV              00041150
30 CONTINUE               00041160
IF (DABS(RKV).LE.DABS(RK(N))*10.*XETA) GO TO 40 00041170
C USE THE SCALED FORM OF THE 00041180
   RECURRENCE IF THE VALUE OF K AT S 00041190
   IS NONZERO                 00041200
C T = -PV/RKV              00041210
RK(1) = QP(1)              00041220
DO 35 I=2,N               00041230
35 RK(I) = T*XQK(I-1)+QP(I) 00041240
GO TO 50                  00041250
C USE UNSCALED FORM        00041260
40 RK(1) = ZERO            00041270
DO 45 I=2,N               00041280
45 RK(I) = QK(I-1)          00041290
50 RKV = RK(1)              00041300
DO 55 I=2,N               00041310
55 RKV = RKV*S+RK(I)        00041320
T = ZERO                  00041330
IF (DABS(RKV).GT.DABS(RK(N))*10.*XETA) T = -PV/RKV 00041340
S = S*T                  00041350

```

```

      GO TO 5                               00041360
      END                                00041370
                                           00041380
                                           00041390
                                           00041400
                                           00041410
                                           00041420
                                           00041430
                                           00041440
                                           00041450
                                           00041460
                                           00041470
                                           00041480
                                           00041490
      IMSL ROUTINE NAME - ZRPQLE           00041490
      COMPUTER      - IBM/DOUBLE          00041490
      LATEST REVISION - JANUARY 1, 1978   00041490
      SUBROUTINE ZRPQLE (ITYPE)           00041490
      INTEGER        ITYPE               SPECIFICATIONS FOR ARGUMENTS 00041500
      INTEGER        N,NN                SPECIFICATIONS FOR LOCAL VARIABLES 00041510
      REAL           ARE,ETA,RMRE
      DOUBLE PRECISION ?(101),QF(101),RK(101),QK(101),SVK(101) 00041540
      DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00041550
      A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00041560
      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00041580
      A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00041590
      THIS ROUTINE CALCULATES SCALAR      00041600
      QUANTITIES USED TO COMPUTE THE    00041610
      NEXT K POLYNOMIAL AND NEW       00041620
      ESTIMATES OF THE QUADRATIC     00041630
      COEFFICIENTS                  00041640
      ITYPE - INTEGER VARIABLE SET HERE 00041650
      INDICATING HOW THE CALCULATIONS 00041660
      ARE NORMALIZED TO AVOID OVERFLOW 00041670
      SYNTHETIC DIVISION OF K BY THE 00041680
      QUADRATIC 1,U,V                 00041690
      FIRST EXECUTABLE STATEMENT      00041700
      CALL ZRPQLH (N,U,V,RK,QK,C,D)    00041710
      IF (DABS(C).GT.DABS(RK(N))X100.XETA) GO TO 5
      IF (DABS(D).GT.DABS(RK(N-1))X100.XETA) GO TO 5
      ITYPE = 3                         00041720
                                         TYPE=3 INDICATES THE QUADRATIC IS 00041730
                                         ALMOST A FACTOR OF K 00041740
      RETURN                            00041750
      5 IF (DABS(D).LT.DABS(C)) GO TO 10 00041760
      ITYPE = 2                         00041770
                                         TYPE=2 INDICATES THAT ALL FORMULAS 00041780
                                         ARE DIVIDED BY D 00041790
      E = RA/D                          00041800
      F = C/D                           00041810
      G = UXRB                          00041820
      H = VXRBD                         00041830
      A3 = (RA+G)XE+HXR(B/D)           00041840
      A1 = RBXF-RA                      00041850
      A7 = (F+U)XRRA+H                  00041860
      RETURN                            00041870
      10 ITYPE = 1                       00041880
                                         TYPE=1 INDICATES THAT ALL FORMULAS 00041890
                                         ARE DIVIDED BY C 00041900
      E = RA/C                          00041910
      F = D/C                           00041920
      G = UXE                           00041930
                                           00041940
                                           00041950

```

```

H = V*RB          00041960
A3 = RAXE+(H/C+G)*RB 00041970
A1 = RB-RAX(D/C) 00041980
A7 = RA+G*D+H*F 00041990
RETURN           00042000
END              00042010
00042020
00042030
00042040
00042050
00042060
00042070
00042080
00042090
00042100
00042110
00042120
00042130
00042140
00042150
00042160
00042170
00042180
00042190
00042200
00042210
00042220
00042230
00042240
00042250
00042260
00042270
00042280
00042290
00042300
00042310
00042320
00042330
00042340
00042350
00042360
00042370
00042380
00042390
00042400
00042410
00042420
00042430
00042440
00042450
00042460
00042470
00042480
00042490
00042500
00042510
00042520
00042530
00042540
00042550

C   SUBROUTINE ZRPQLF (ITYPE)          SPECIFICATIONS FOR ARGUMENTS
C   INTEGER      ITYPE               SPECIFICATIONS FOR LOCAL VARIABLES
C   INTEGER      N,NN,I
C   REAL        ARE,ETA,RMRE
C   DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101)
C   DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3
C   1    A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,TEMP,ZERO
C   1    COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6
C   1    A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
C   DATA        ZERO/0.0D0/          COMPUTES THE NEXT K POLYNOMIALS
C                           USING SCALARS COMPUTED IN ZRPQLE
C                           FIRST EXECUTABLE STATEMENT
C   IF (ITYPE.EQ.3) GO TO 20
C   TEMP = RA
C   IF (ITYPE.EQ.1) TEMP = RB
C   IF (DABS(A1).GT.DABS(TEMP).NE.TAN10.) GO TO 10
C                           IF A1 IS NEARLY ZERO THEN USE A
C                           SPECIAL FORM OF THE RECURRENCE
C   RK(1) = ZERO
C   RK(2) = -A7*QP(1)
C   DO 5 I=3,N
C   5 RK(I) = A3*QK(I-2)-A7*QP(I-1)
C   RETURN
C                           USE SCALED FORM OF THE RECURRENCE
C   10 A7 = A7/A1
C   A3 = A3/A1
C   RK(1) = QP(1)
C   RK(2) = QP(2)-A7*QP(1)
C   DO 15 I=3,N
C   15 RK(I) = A3*QK(I-2)-A7*QP(I-1)+QP(I)
C   RETURN
C                           USE UNSCALED FORM OF THE RECURRENCE
C                           IF TYPE IS 3
C   20 RK(1) = ZERO
C   RK(2) = ZERO
C   DO 25 I=3,N
C   25 RK(I) = QK(I-2)
C   RETURN
C   END
C   -----
C   IMSL ROUTINE NAME - ZRPQLG
C   COMPUTER      - IBM/DOUBLE

```

```

C   LATEST REVISION      - JANUARY 1, 1978          00042560
C   SUBROUTINE ZRPQLG (ITYPE,UU,VV)                  00042570
C                                         SPECIFICATIONS FOR ARGUMENTS 00042580
C   INTEGER           ITYPE                         00042590
C   DOUBLE PRECISION    UU,VV                        00042600
C                                         SPECIFICATIONS FOR LOCAL VARIABLES 00042610
C   INTEGER           N.NN                         00042620
C   REAL              ARE,ETA,RMRE                00042630
C   DOUBLE PRECISION   P(101),QP(101),RK(101),QK(101),SVK(101) 00042640
C   DOUBLE PRECISION   SR,SI,U,V,RA,RB,C,D,A1,A2,A3,        00042650
C   1                 A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,        00042660
C   2                 A4,A5,B1,B2,C1,C2,C3,C4,TEMP,ZERO       00042670
C   COMMON /ZRPQLJ/ P,GP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6 00042680
C   1                 A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00042690
C   DATA              ZERO/0.0D0/                   00042700
C                                         COMPUTE NEW ESTIMATES OF THE 00042710
C                                         QUADRATIC COEFFICIENTS USING THE 00042720
C                                         SCALARS COMPUTED IN ZRPQLE 00042730
C                                         USE FORMULAS APPROPRIATE TO SETTING 00042740
C                                         OF TYPE. 00042750
C                                         FIRST EXECUTABLE STATEMENT 00042760
C
C   IF (ITYPE.EQ.3) GO TO 15                      00042770
C   IF (ITYPE.EQ.2) GO TO 5                      00042780
C   A4 = RA+UMRB+H*F                00042790
C   A5 = C+(U+V*H)*D                00042800
C   GO TO 10                           00042810
C   5 A4 = (RA+G)*F+H                00042820
C   A5 = (F+U)*C+V*D                00042830
C                                         EVALUATE NEW QUADRATIC COEFFICIENTS. 00042840
C
C   10 B1 = -RK(N)/P(NN)                      00042850
C   B2 = -(RK(N-1)+B1*P(N))/P(NN)            00042860
C   C1 = V*B2*A1                         00042870
C   C2 = B1*A7                           00042880
C   C3 = B1*B1*A3                         00042890
C   C4 = C1-C2-C3                         00042900
C   TEMP = A5+B1*A4-C4                     00042910
C   IF (TEMP.EQ.ZERO) GO TO 15             00042920
C   UU = U-(U*(C3+C2)+V*(B1*A1+B2*A7))/TEMP 00042930
C   VV = V*(1+C4/TEMP)                    00042940
C   RETURN                                00042950
C                                         IF TYPE=3 THE QUADRATIC IS ZEROED 00042960
C
C   15 UU = ZERO                          00042970
C   VV = ZERO                          00042980
C   RETURN                                00042990
C   END                                  00043000
C
C   SUBROUTINE ZRPQLH (NN,U,V,P,Q,RA,RB)      00043010
C                                         SPECIFICATIONS FOR ARGUMENTS 00043020
C   INTEGER           NN                         00043030
C   DOUBLE PRECISION   P(NN),Q(NN),U,V,RA,RB 00043040
C                                         SPECIFICATIONS FOR LOCAL VARIABLES 00043050
C   INTEGER           I                          00043060
C   DOUBLE PRECISION   C                          00043070
C                                         DIVIDES P BY THE QUADRATIC I,U,V 00043080
C                                         PLACING THE QUOTIENT IN Q AND THE 00043090
C                                         REMAINDER IN A,B 00043100
C                                         00043110
C                                         00043120
C                                         00043130
C                                         00043140
C                                         00043150

```

FIRST EXECUTABLE STATEMENT	
C	00043160
RB = P(1)	00043170
Q(1) = RB	00043180
RA = P(2)-UXRB	00043190
Q(2) = RA	00043200
DO 5 I=3,NN	00043210
C = P(I)-UXRA-VXRB	00043220
Q(I) = C	00043230
RB = RA	00043240
RA = C	00043250
5 CONTINUE	00043260
RETURN	00043270
END	00043280
CCC	00043290
IMSL ROUTINE NAME - ZRPQLI	00043300
C-----	00043310
C COMPUTER - IBM/DOUBLE	00043320
C LATEST REVISION - JANUARY 1, 1978	00043330
C SUBROUTINE ZRPQLI (RA,B1,C,SR,SI,RLR,RLI)	00043340
C DOUBLE PRECISION RA,B1,C,SR,SI,RLR,RLI	00043350
C DOUBLE PRECISION RB,D,E,ZERO,ONE,TWO	00043360
C DATA ZERO,ONE,TWO/0.000,1.000,2.000/	00043370
CCCCC	00043380
SPECIFICATIONS FOR ARGUMENTS	00043390
SPECIFICATIONS FOR LOCAL VARIABLES	00043400
CALCULATE THE ZEROS OF THE QUADRATIC	00043410
$AZ^2 + BZ + C$. THE QUADRATIC	00043420
FORMULA, MODIFIED TO AVOID	00043430
OVERFLOW, IS USED TO FIND THE	00043440
LARGER ZERO IF THE ZEROS ARE REAL	00043450
AND BOTH ZEROS ARE COMPLEX.	00043460
THE SMALLER REAL ZERO IS FOUND	00043470
DIRECTLY FROM THE PRODUCT OF THE	00043480
ZEROS C/A	00043490
FIRST EXECUTABLE STATEMENT	00043500
00043510	00043520
00043530	00043540
00043550	00043560
00043570	00043580
00043590	00043600
00043610	00043620
00043630	00043640
00043650	00043660
00043670	00043680
00043690	00043700
00043710	00043720
00043730	00043740
00043750	00043760
IF (RA.NE.ZERO) GO TO 10	00043560
SR = ZERO	00043570
IF (B1.NE.ZERO) SR = -C/B1	00043580
RLR = ZERO	00043590
5 SI = ZERO	00043600
RLI = ZERO	00043610
RETURN	00043620
10 IF (C.NE.ZERO) GO TO 15	00043630
SR = ZERO	00043640
RLR = -B1/RA	00043650
GO TO 5	00043660
CC COMPUTE DISCRIMINANT AVOIDING	00043670
OVERFLOW	00043680
15 RB = B1/TWO	00043690
IF (DABS(RB).LT.DABS(C)) GO TO 20	00043700
E = ONE-(RA/RB)*(C/RB)	00043710
D = DSQRT(DABS(E))*DABS(RB)	00043720
GO TO 25	00043730
20 E = RA	00043740
IF (C.LT.ZERO) E = -RA	00043750

```

E = RB*(RB/DABS(C))-E          00043760
D = DSQRT(DABS(E))*DSQRT(DABS(C)) 00043770
25 IF (E.LT.ZERO) GO TO 30      00043780
C                                     REAL ZEROS    00043790
IF (RB.GE.ZERO) D = -D          00043800
RLR = (-RB+D)/RA              00043810
SR = ZERO                      00043820
IF (RLR.NE.ZERO) SR = (C/RLR)/RA 00043830
GO TO 5                         00043840
C                                     COMPLEX CONJUGATE ZEROS 00043850
30 SR = -RB/RA                  00043860
RLR = SR                      00043870
SI = DABS(D/RA)                00043880
RLI = -SI                      00043890
RETURN                         00043900
END                            00043910
CCC                           00043920
CCC                           00043930
CCC                           00043940
SUBROUTINE LEQ2C (A,N,IA,B,M,IB,IJOB,WA,WK,IER) 00043950
C                                     00043960
COMPLEX16   A(IA,1),B(IB,1),WA(N,1),TEMPA,TEMPB,TEMPC 00043970
DOUBLE PRECISION WK(N),TA(2),TB(2),TC(2) 00043980
DOUBLE PRECISION AR,AI,BR,BI,CR,CI,DXNORM,XNORM,ZERO 00043990
DOUBLE PRECISION ACC(2)          00044000
EQUIVALENCE (TA(1),TEMPA),(TB(1),TEMPB),(TC(1),TEMPC), 00044010
               (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI), 00044020
               (TC(1),CR),(TC(2),CI) 00044030
DATA         ZERO/0.0D0/        00044040
DATA         ITMAX/50/          00044050
C                                     FIRST EXECUTABLE STATEMENT 00044060
IER = 0                          00044070
N1 = N+1                         00044080
N2 = N+2                         00044090
IF (IJOB .EQ. 2) GO TO 15       00044100
C                                     SAVE MATRIX A 00044110
DO 10 I = 1,N                    00044120
  DO 5 J = 1,N                   00044130
    WA(I,J) = A(I,J)            00044140
  5  CONTINUE                     00044150
10 CONTINUE                      00044160
C                                     FACTOR MATRIX A 00044170
CALL LEQTIC (WA,N,N,B,M,IB,1,WK,IER) 00044180
IF (IER .NE. 0) GO TO 9000      00044190
IF (IJOB .EQ. 1) GO TO 9005      00044200
C                                     SAVE THE RIGHT HAND SIDES 00044210
15 DO 65 J = 1,M                00044220
  DO 20 I = 1,N                 00044230
    WA(I,N1) = B(I,J)           00044240
  20 CONTINUE                     00044250
C                                     OBTAIN A SOLUTION 00044260
CALL LEQTIC(WA,N,N,WA(1,N1),1,N,2,WK,IER) 00044270
C                                     COMPUTE THE NORM OF THE SOLUTION 00044280
XNORM = ZERO                     00044290
DO 25 I = 1,N                   00044300
  TEMPB = WA(I,N1)              00044310
  XNORM = DMAX1(XNORM,DABS(BI),DABS(AI)) 00044320
  25 CONTINUE                     00044330
C                                     COMPUTE RESIDUALS 00044340
IF (XNORM .EQ. ZERO) GO TO 65 00044350

```

```

DO 50 ITER = 1,ITMAX          00044360
DO 40 I = 1,N                  00044370
    TEMPB = B(I,J)
    ACC(1) = 0.0D0               00044380
    ACC(2) = 0.0D0               00044390
    CALL VXADD(BR,ACC)          00044400
DO 30 JJ = 1,N                  00044410
    TEMPB = A(I,JJ)
    TEMPB = WA(JJ,N1)
    CALL VXMUL(-AR,BR,ACC)      00044420
    CALL VXMUL(AI,BI,ACC)       00044430
30     CONTINUE                  00044440
    CALL VXSTO(ACC,CR)          00044450
    TEMPB = B(I,J)
    ACC(1) = 0.0D0               00044460
    ACC(2) = 0.0D0               00044470
    CALL VXADD(BI,ACC)          00044480
DO 35 JJ = 1,N                  00044490
    TEMPB = A(I,JJ)
    TEMPB = WA(JJ,N1)
    CALL VXMUL(-AR,BI,ACC)      00044500
    CALL VXMUL(-BR,AI,ACC)      00044510
35     CONTINUE                  00044520
    CALL VXSTO(ACC,CI)          00044530
    WA(I,N2) = TEMPC           00044540
40     CONTINUE                  00044550
    CALL LEQTIC(WA,N,N,WA(1,N2),1,N,2,WK,IER) 00044560
    DXNORM = ZERO                00044570
C          UPDATE THE SOLUTION 00044580
    DO 45 I = 1,N                00044590
        WA(I,N1) = WA(I,N1)+WA(I,N2)
        TEMPB = WA(I,N2)
        DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI)) 00044600
45     CONTINUE                  00044610
        IF (XNORM+DXNORM .EQ. XNORM) GO TO 55 00044620
50     CONTINUE                  00044630
        IER = 130                 00044640
C          STORE THE SOLUTION 00044650
55     DO 60 JK = 1,N            00044660
        B(JK,J) = WA(JK,N1)       00044670
60     CONTINUE                  00044680
        IF (IER .NE. 0) GO TO 9000 00044690
65     CONTINUE                  00044700
        GO TO 9005                 00044710
9000    CONTINUE                  00044720
        CALL UERTST(IER,6HLEQ2C ) 00044730
9005    RETURN                   00044740
        END                       00044750
C
C          SUBROUTINE LEQTIC (A,N,IA,B,M,IB,IJOB,WA,IER)
C          SPECIFICATIONS FOR ARGUMENTS 00044760
C          INTEGER N,IA,M,IB,IJOB,IER 00044770
C          COMPLEX*16 A(IA,N),B(IB,M) 00044780
C          DOUBLE PRECISION WA(N)   00044790
C
C          DOUBLE PRECISION P,Q,ZERO,ONE,T(2),RN,BIG 00044800
C          COMPLEX*16 SUM,TEMP           00044810
C          INTEGER I,J,JM1,IM1,K,IMAX,JF1,IW,N1 00044820
C
C          SPECIFICATIONS FOR LOCAL VARIABLES 00044830

```

	EQUIVALENCE	(SUM,T(1))	C 1044960
CC	DATA	ZERO/0.0D0/, ONE/1.D0/	00044970
		INITIALIZATION	00044980
		FIRST EXECUTABLE STATEMENT	00044990
	IER = 0		00045000
	IF (IJOB .EQ. 2) GO TO 75		00045010
C	RN = N		00045020
		FIND EQUILIBRATION FACTORS	00045030
	DO 10 I=1,N		00045040
	BIG = ZERO		00045050
	DO 5 J=1,N		00045060
	TEMP = A(I,J)		00045070
	P = CDABS(TEMP)		00045080
	IF (P .GT. BIG) BIG = P		00045090
5	CONTINUE		00045100
	IF (BIG .EQ. ZERO) GO TO 105		00045110
	WA(I) = ONE/BIG		00045120
10	CONTINUE		00045130
C		L-U DECOMPOSITION	00045140
	DO 70 J = 1,N		00045150
	JMI = J-1		00045160
C	IF (JMI .LT. 1) GO TO 23	COMPUTE U(I,J), I=1,...,J-1	00045170
	DO 20 I=1,JMI		00045180
	SUM = A(I,J)		00045190
	IM1 = I-1		00045200
	IF (IM1 .LT. 1) GO TO 20		00045210
	DO 15 K=1,IM1		00045220
	SUM = SUM-A(I,K)*A(K,J)		00045230
15	CONTINUE		00045240
	A(I,J) = SUM		00045250
20	CONTINUE		00045260
25	P = ZERO		00045270
C		COMPUTE U(J,J) AND L(I,J), I=J+1,...,N	00045280
	DO 45 I=J,N		00045290
	SUM = A(I,J)		00045300
	IF (JMI .LT. 1) GO TO 40		00045310
	DO 35 K=1,JMI		00045320
	SUM = SUM-A(I,K)*A(K,J)		00045330
35	CONTINUE		00045340
	A(I,J) = SUM		00045350
40	Q = WA(I)*CDABS(SUM)		00045360
	IF (P .GE. Q) GO TO 45		00045370
	P = Q		00045380
	IMAX = I		00045390
45	CONTINUE		00045400
C		TEST FOR ALGORITHMIC SINGULARITY	00045410
	Q = RN+P		00045420
	IF (Q .EQ. RN) GO TO 105		00045430
C	IF (J .EQ. IMAX) GO TO 60	INTERCHANGE ROWS J AND IMAX	00045440
	DO 50 K=1,N		00045450
	TEMP = A(IMAX,K)		00045460
	A(IMAX,K) = A(J,K)		00045470
	A(J,K) = TEMP		00045480
50	CONTINUE		00045490
	WA(IMAX) = WA(J)		00045500
60	WA(J) = IMAX		00045510
	JP1 = J+1		00045520
	IF (JP1 .GT. N) GO TO 70		00045530
			00045540
			00045550

C	DIVIDE BY PIVOT ELEMENT A(J,J)	00045560
	DO 65 I = JP1,N	00045570
	A(I,J) = A(I,J)/TEMP	00045580
65	CONTINUE	00045590
70	CONTINUE	00045600
75	IF (IJOB .EQ. 1) GO TO 9005	00045610
	DO 103 K = 1,M	00045620
C	SOLVE UX = Y FOR X	00045630
	IW = 0	00045640
	DO 90 I = 1,N	00045650
	IMAX = WA(I)	00045660
	SUM = B(IMAX,K)	00045670
	B(IMAX,K) = B(I,K)	00045680
	IF (IW .EQ. 0) GO TO 85	00045690
	IM1 = I-1	00045700
	DO 80 J = IW,IM1	00045710
	SUM = SUM-A(I,J)*B(J,K)	00045720
80	CONTINUE	00045730
	GO TO 88	00045740
85	IF (T(1) .NE. ZERO .OR. T(2) .NE. ZERO) IW = I	00045750
88	B(I,K) = SUM	00045760
90	CONTINUE	00045770
C	SOLVE LY = B FOR Y	00045780
	N1 = N+1	00045790
	DO 100 IW = 1,N	00045800
	I = N1-IW	00045810
	JP1 = I+1	00045820
	SUM = B(I,K)	00045830
	IF (JP1 .GT. N) GO TO 98	00045840
	DO 95 J = JP1,N	00045850
	SUM = SUM-A(I,J)*B(J,K)	00045860
95	CONTINUE	00045870
98	B(I,K) = SUM/A(I,I)	00045880
100	CONTINUE	00045890
103	CONTINUE	00045900
	GO TO 9005	00045910
C	ALGORITHMIC SINGULARITY	00045920
103	IER = 129	00045930
9000	CONTINUE	00045940
C	PRINT ERROR	00045950
	CALL UERTST(IER,6HLEQT1C)	00045960
9005	RETURN	00045970
	END	00045980
		00045990

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 AFVAL Technical Reports, AFVAL-TR-86-3034
and 86-3035

o: ALL ADDRESSES

1. Please delete the second paragraph in the NOTICE page affixed to the inside cover of AFVAL-TR-86-3034, "Strength Analysis of Laminated and Metallic Plates Bolted Together by Many Fasteners" and AFVAL-TR-86-3035, "Design Guide for Bolted Joints in Composite Structures."
2. Please contact the undersigned if you have any questions regarding this letter.

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

cc: AFVAL/FIBRA
(V. Venkayya)

AD-B108/123

