

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

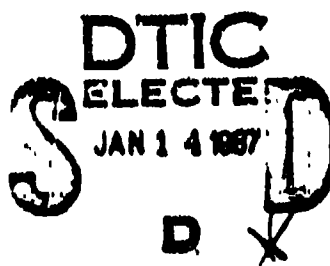
E. S. SAETHER

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

ADB108123L

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE													
1. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		11. RESTRICTIVE MARKINGS T/E											
2. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Distribution Limited to U.S. Government Agencies only; critical technology; 3.86. Other requests must be referred to AFWAL/FIBRA, WPAFB, OH 45433											
2a. DECLASSIFICATION/DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-86-3035											
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOR 86-211		7a. NAME OF MONITORING ORGANIZATION Flight Dynamics Laboratory (AFWAL/FIBRA) Air Force Wright Aeronautical Laboratories											
6a. NAME OF PERFORMING ORGANIZATION Northrop Corporation Aircraft Division		7b. ADDRESS (City, State and ZIP Code) Wright-Patterson Air Force Base Dayton, OH 45433-6553											
6b. ADDRESS (City, State and ZIP Code) One Northrop Avenue Hawthorne, CA 90230		7c. ADDRESS (City, State and ZIP Code)											
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract F33615-82-C-3217											
8b. OFFICE SYMBOL (If applicable)		9. SOURCE OF FUNDING NOS.											
8c. ADDRESS (City, State and ZIP Code)		<table border="1"> <thead> <tr> <th>PROGRAM ELEMENT NO.</th> <th>PROJECT NO.</th> <th>TASK NO.</th> <th>WORK UNIT NO.</th> </tr> </thead> <tbody> <tr> <td>62201F</td> <td>2401</td> <td>02</td> <td>55</td> </tr> </tbody> </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. PERSONAL AUTHOR(S) R. L. Ramkumar, E. S. Saether and D. Chang													
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 1/1/83 TO 1/30/85		13c. DATE OF REPORT (Yr., Mo., Day) March 1986									
13d. PAGE COUNT 264													
16. SUPPLEMENTARY NOTATION													
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)											
<table border="1"> <thead> <tr> <th>FIELD</th> <th>GROUP</th> <th>SUB GROUP</th> </tr> </thead> <tbody> <tr> <td>01</td> <td>03</td> <td></td> </tr> <tr> <td>03</td> <td>05</td> <td></td> </tr> </tbody> </table>		FIELD	GROUP	SUB GROUP	01	03		03	05		Bolted joints, composite structures, design guidelines, analytical design tools (SASCJ and SAMCJ), user instructions, samples, program listing.		
FIELD	GROUP	SUB GROUP											
01	03												
03	05												
19. ABSTRACT (Continue on reverse if necessary and identify by block number) A design guide was developed for bolted composite structural joints. The guide includes general design guidelines for the various joint parameters, an analytical design methodology, a description of the analytical design tools, an illustration of the use of corresponding computer codes (SASCJ and SAMCJ), and a listing of the computer codes. The proposed design procedure is purely analytical, and enables the user to rapidly evaluate many different joint concepts for a selected application. When the bolted structure is fabricated using existing (fully characterized) materials, the design requires no complementary test results. Presented analytical design tools are currently restricted to primarily uniaxially loaded joints and fastener arrangements that are currently used in aircraft structures. Also, the bolted joint is assumed to be strength-critical. However, sample fatigue test results are presented to illustrate a durability check on the joint, assuming a simplified fatigue analysis and assuming that fatigue failure is induced by excessive hole elongation. Despite its current restrictions, this guide is the first government document that provides guidance and analytical tools for the design of bolted composite structures.													
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☒ DTIC USERS ☐		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED											
22a. NAME OF RESPONSIBLE INDIVIDUAL V. B. Venkayya		22b. TELEPHONE NUMBER (Include Area Code) (513) 255-6992		22c. OFFICE SYMBOL AFWAL/FIBRA									

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

PREFACE

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

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

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

Graphics	R. Cordero
Typing	C. Harris



Accession For	
NTIS CRA&I	☐
DTIC TAB	☒
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/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 SASCF 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 SASCF Input Description.....	61
	3.3 SASCF Output Description.....	68
	3.4 Description of SAMCF Analysis.....	69
	3.5 SAMCF Input Description.....	79
	3.6 SAMCF 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 SAMCF..	97
5	CONCLUSIONS.....	105
	REFERENCES.....	107
	Appendix A - SASCF Program Listing.....	109
	Appendix B - SAMCF 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

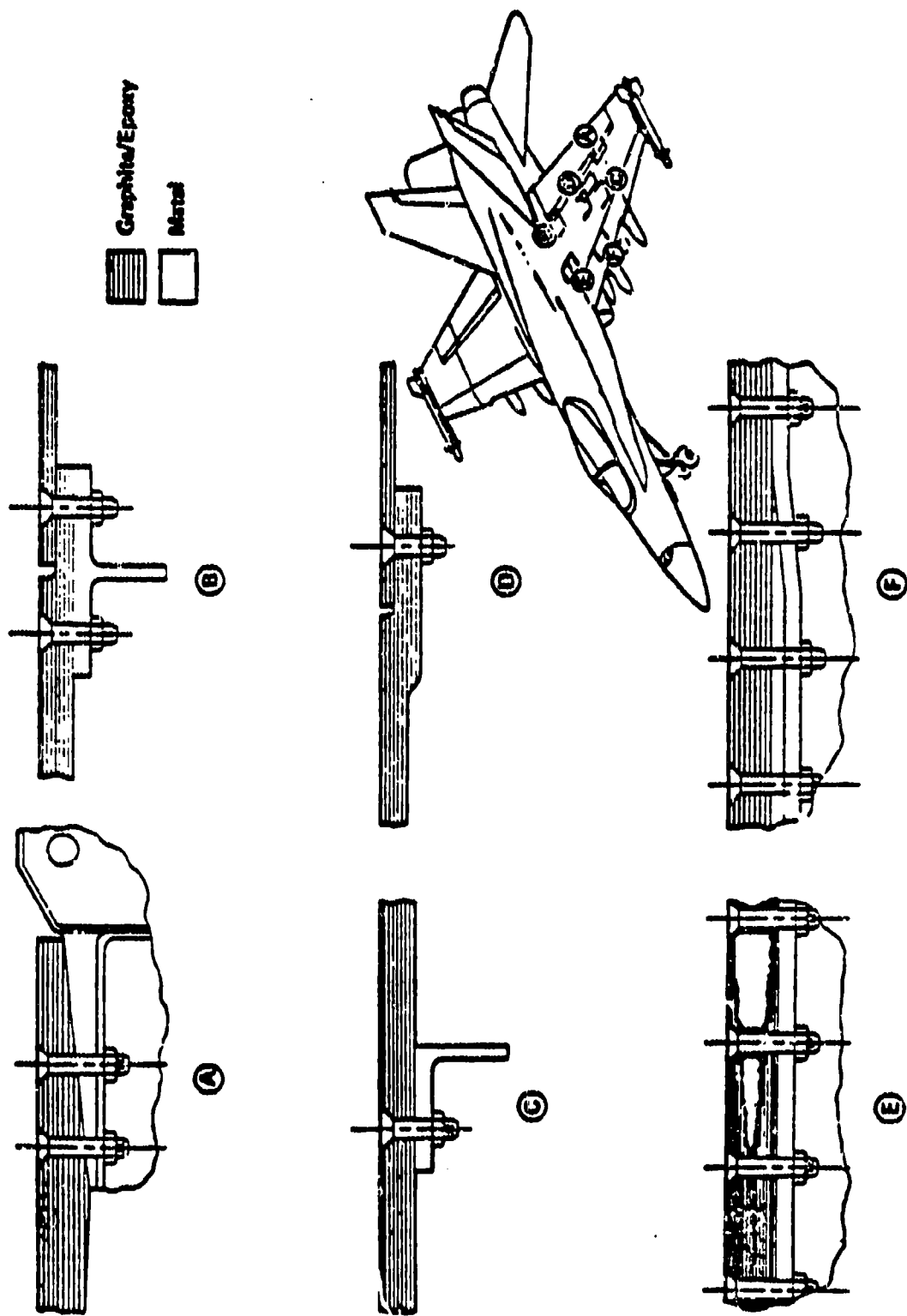
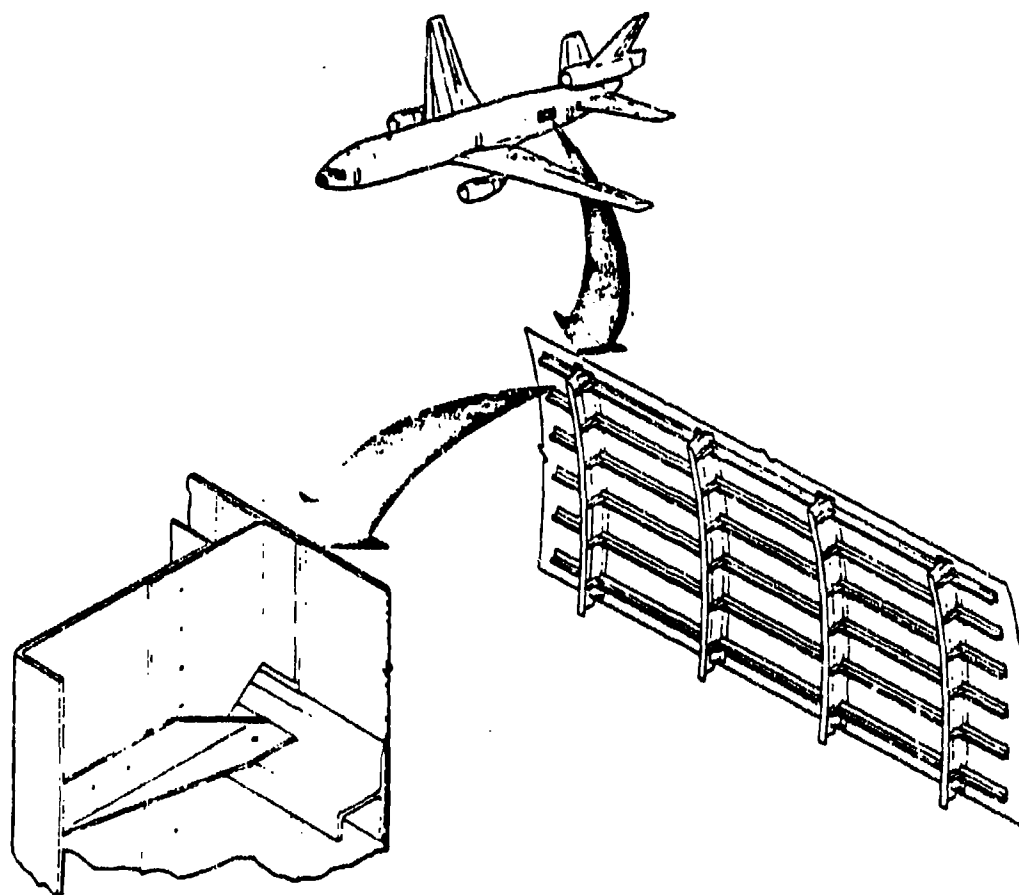


Figure 1. Sample Bolted Joints in the F/A-18A Aircraft Wing (Reference 1).



STIFFENER ATTACHMENT

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

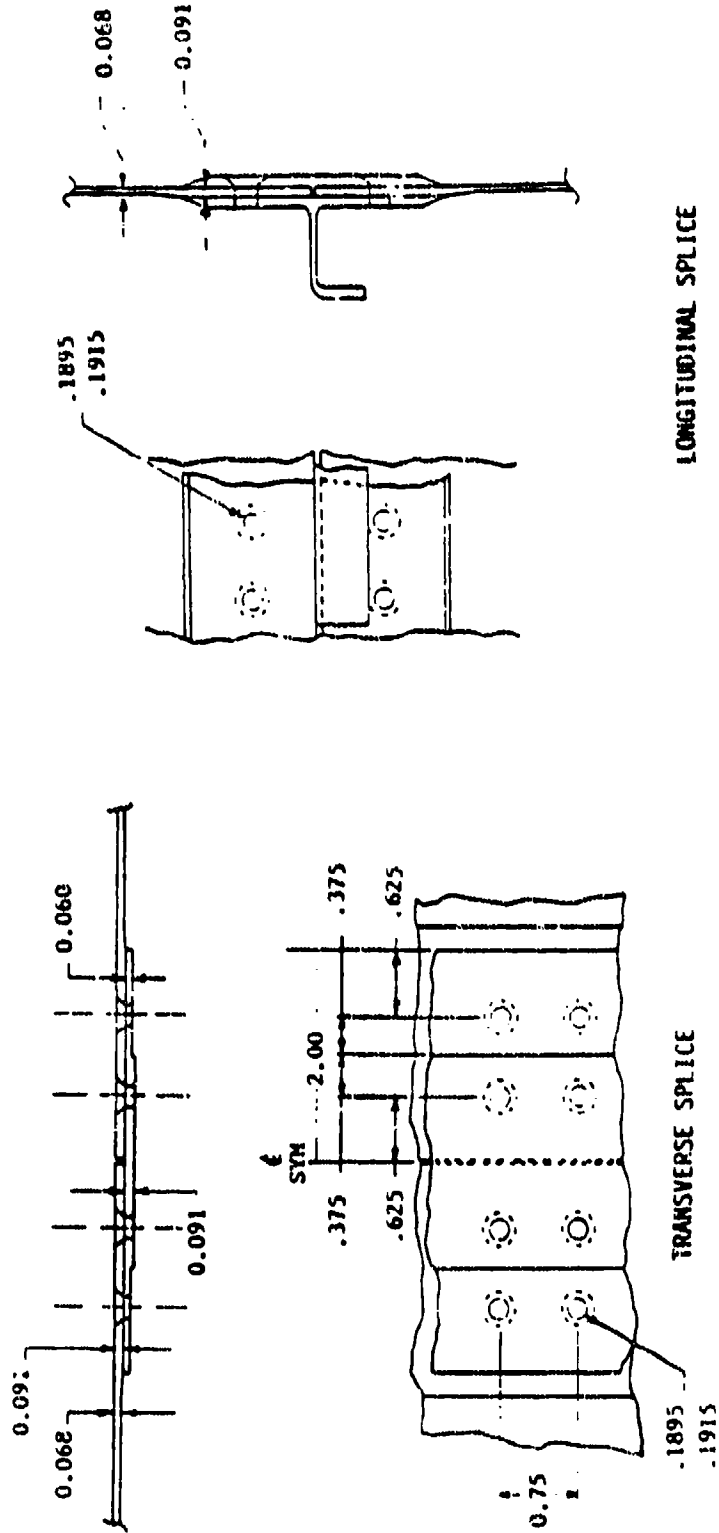


Figure 3. Bolted Joint Concepts for Composite Fuselage Structures (Reference 2).

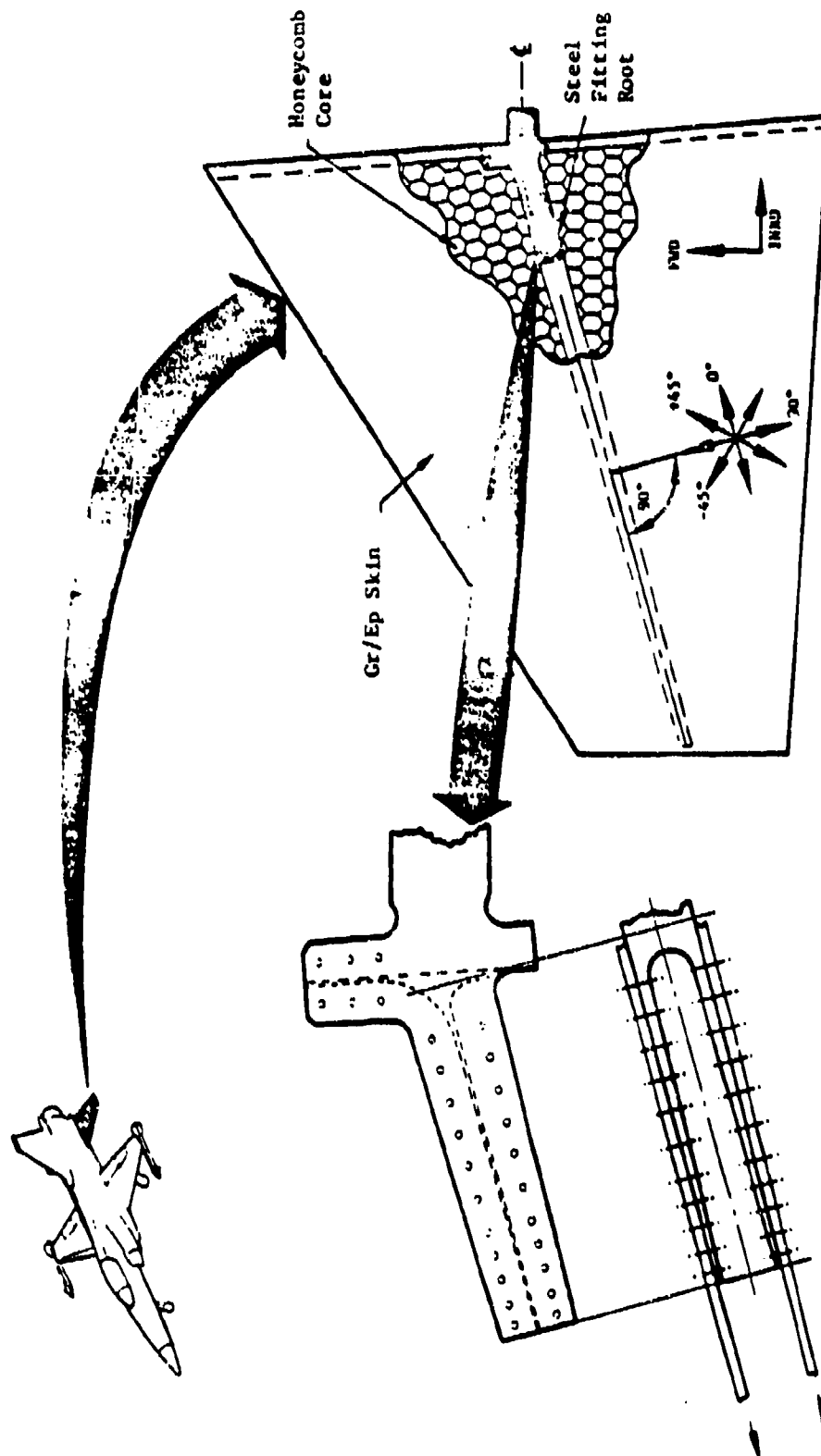


Figure 4. Skin-to-Root Fitting Joint in the F-20 Horizontal Stabilizer.

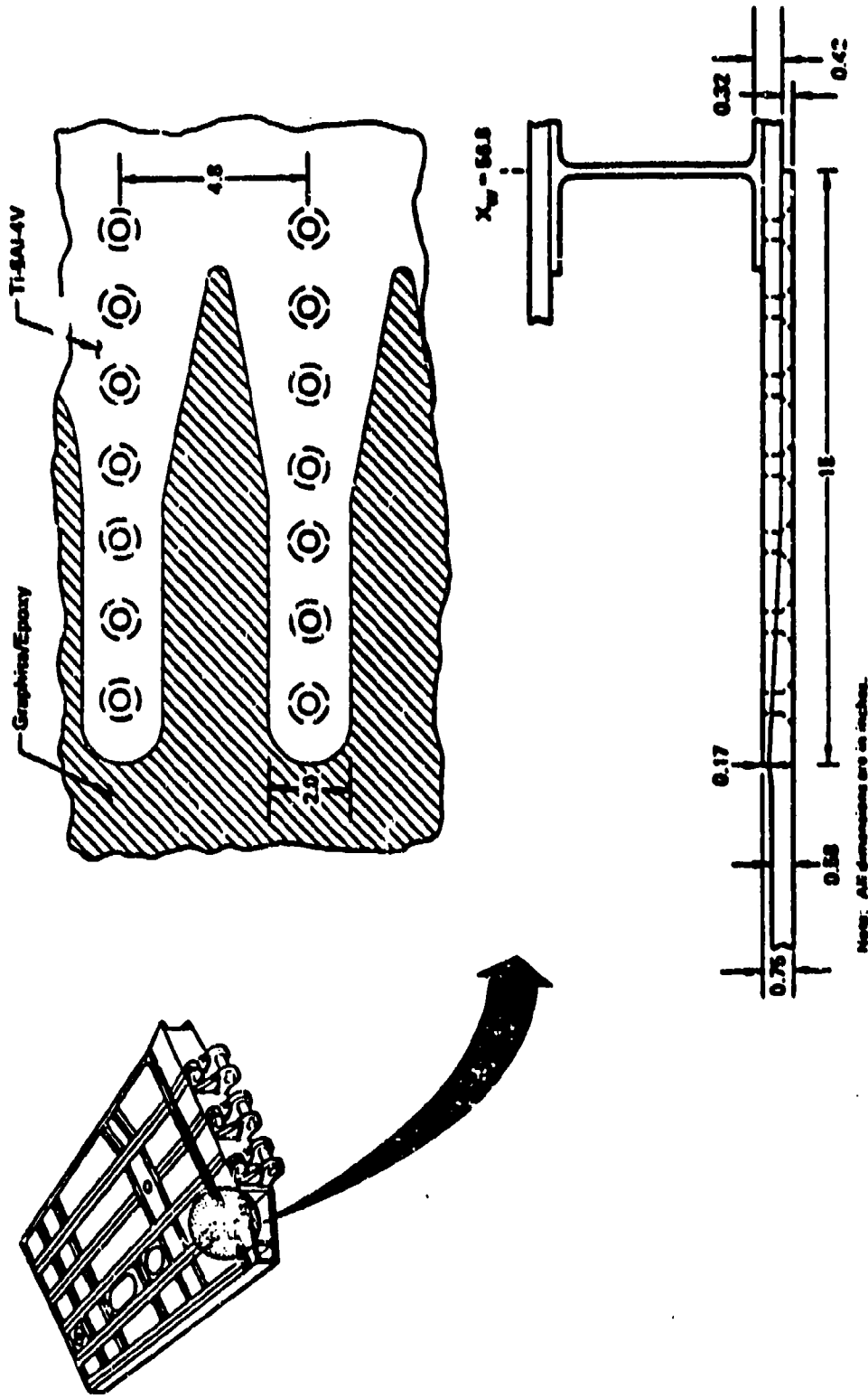


Figure 5 . Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section (Reference 1).

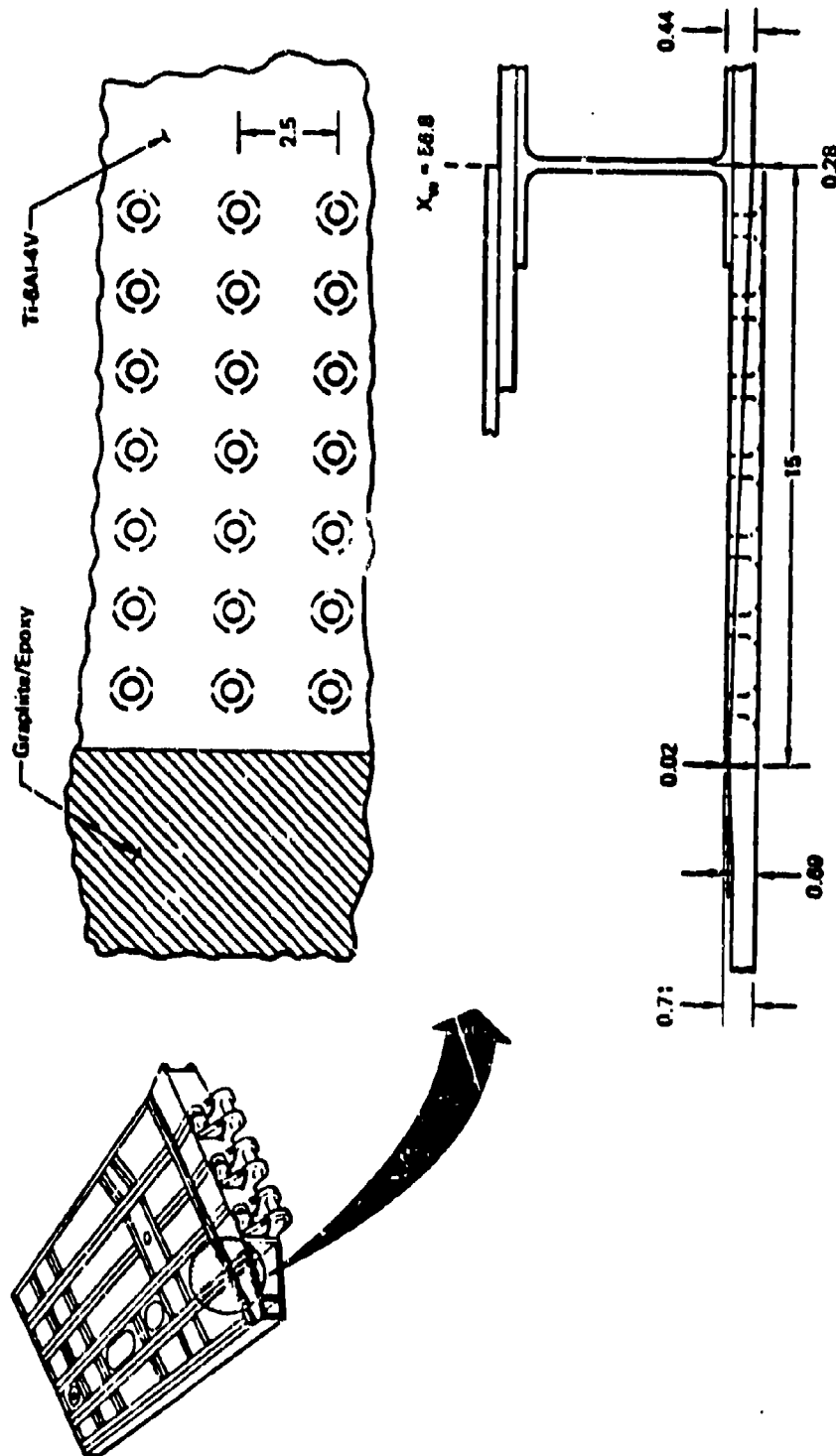


Figure 5. Alternative Bolted Joint Concept Evaluated for the F/A-18A Wing Root Section.
 (Reference 1; Concluded).

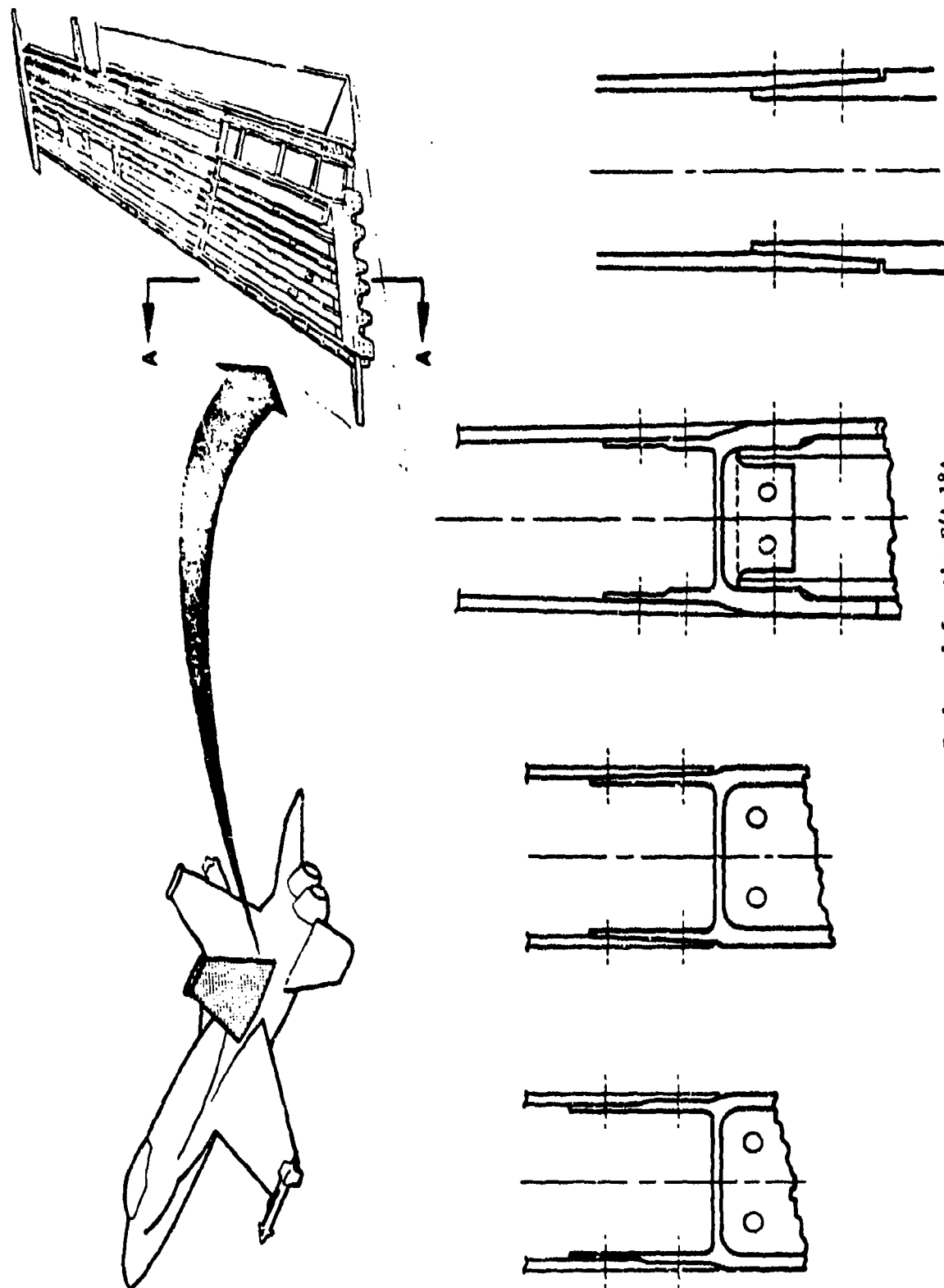


Figure 6. Alternative Bolted Joint Concepts Evaluated for the F/A-18A Vertical Tail Root Section (References 3 to 5).

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 SASCF and the SAMCF computer codes for the strength prediction of single and multiple fastener joints in composites, respectively. The SASCF 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 SAMCF 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.

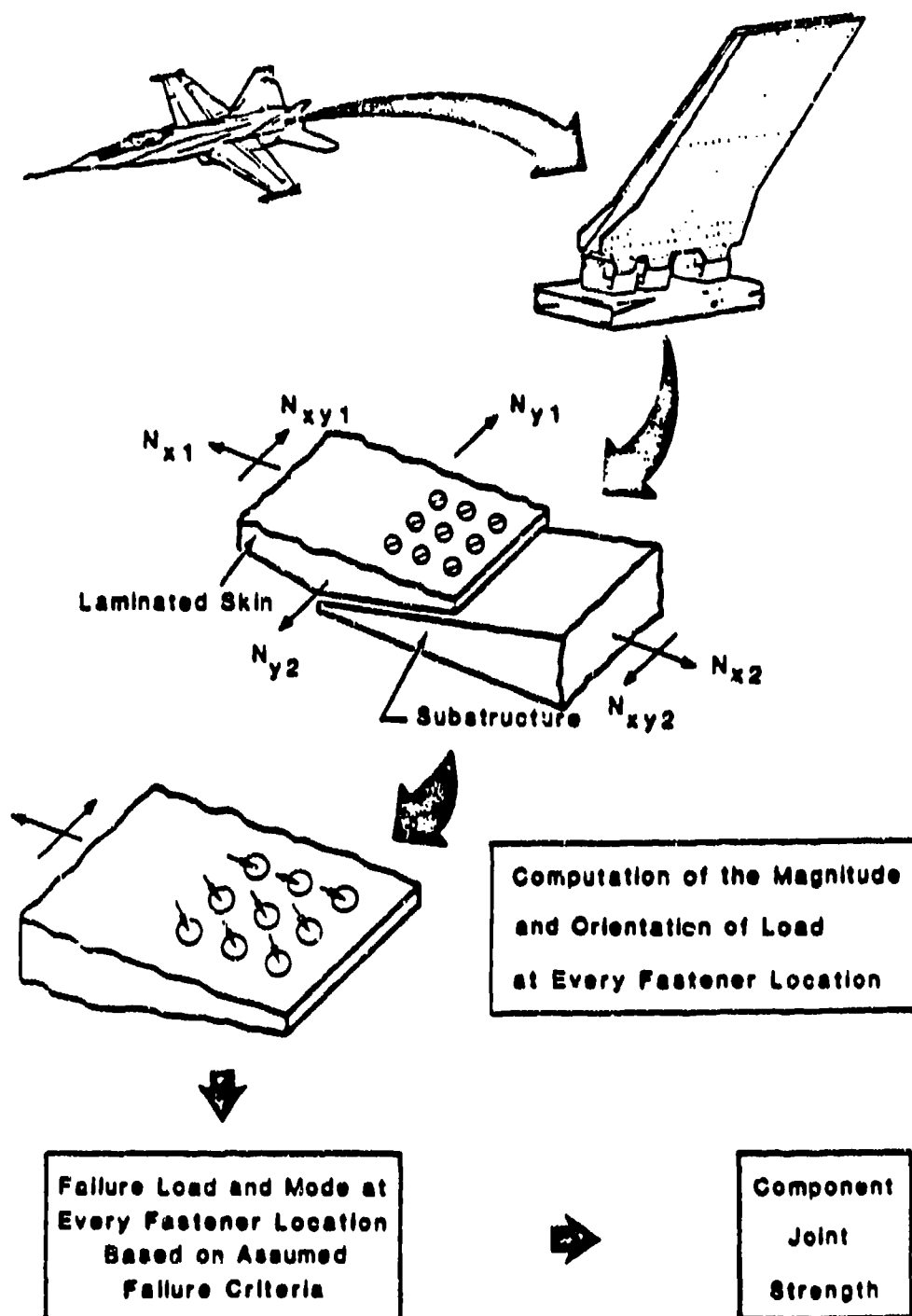


Figure 7. Overview of the Strength Analysis of Bolted Structures.

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 SASCF (Strength Analysis of Single Fastener Composite Joints) or the SAMCF (Strength Analysis of Multiple Fastener Composite Joints) computer code. Plates bolted together by many fasteners are analyzed using SAMCF computer code. Section 3 presents a brief description of these analyses. The reader is referred to References 6 and 7 for further details.

1.5 Test Requirements

A test-independent, purely analytical design tool has been developed to design a bolted joint for composite structures that are fabricated using characterized materials. The engineering properties (Young's moduli in the fiber and transverse directions, major Poisson's ratio and the shear modulus in the fiber coordinate system), the strengths or failure strains (under tension, compression and shear), and the failure parameters for the assumed failure criteria (characteristic distances for net section, shear-out and bearing failure predictions using the average stress

failure criteria, for example) are known for a characterized composite material (lamina). Tests required to obtain the above material properties must be performed on a new (uncharacterized) material system, prior to designing bolted joints for structural parts made from this material. When previously characterized materials are used in the bolted plates, the test requirements are nil for the design of an efficient bolted joint concept.

1.6 Current Restrictions

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

- (1) The developed strength analyses (SASCJ and SAMCJ computer codes) do not account for countersunk fastener effects.
- (2) SASCJ and SAMCJ contain a stress analysis that approximates the fastener/plate contact problem by an assumed radial stress distribution.
- (3) SASCJ AND SAMCJ are restricted to a uniaxial applied loading, in tension or in compression.
- (4) The prediction of the durability of a joint is restricted to the incorporation of the bearing stress at critical fastener locations into experimentally obtained curves for joint life.
- (5) SAMCJ restricts the user to rectangular element geometries and currently used fastener spacing and arrangement.

Despite the above restrictions, the developed analyses and the proposed design procedure mark a significant improvement

over the state-of-the-art with respect to the design and analysis
of bolted joints in composite structures.

SECTION 2

GENERAL DESIGN GUIDELINES AND JOINT VARIABLES

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

- (a) The loads that must be transferred from one part to another.
- (b) The load transfer location in the structure.
- (c) Geometric constraints, if any, at the load transfer location.
- (d) Fastener type, size and arrangement.
- (e) The environmental range the joint will be exposed to.
- (f) The effect of the joint concept on structural efficiency and reliability.

The following sub-sections discuss the primary variables that influence the design of boited 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

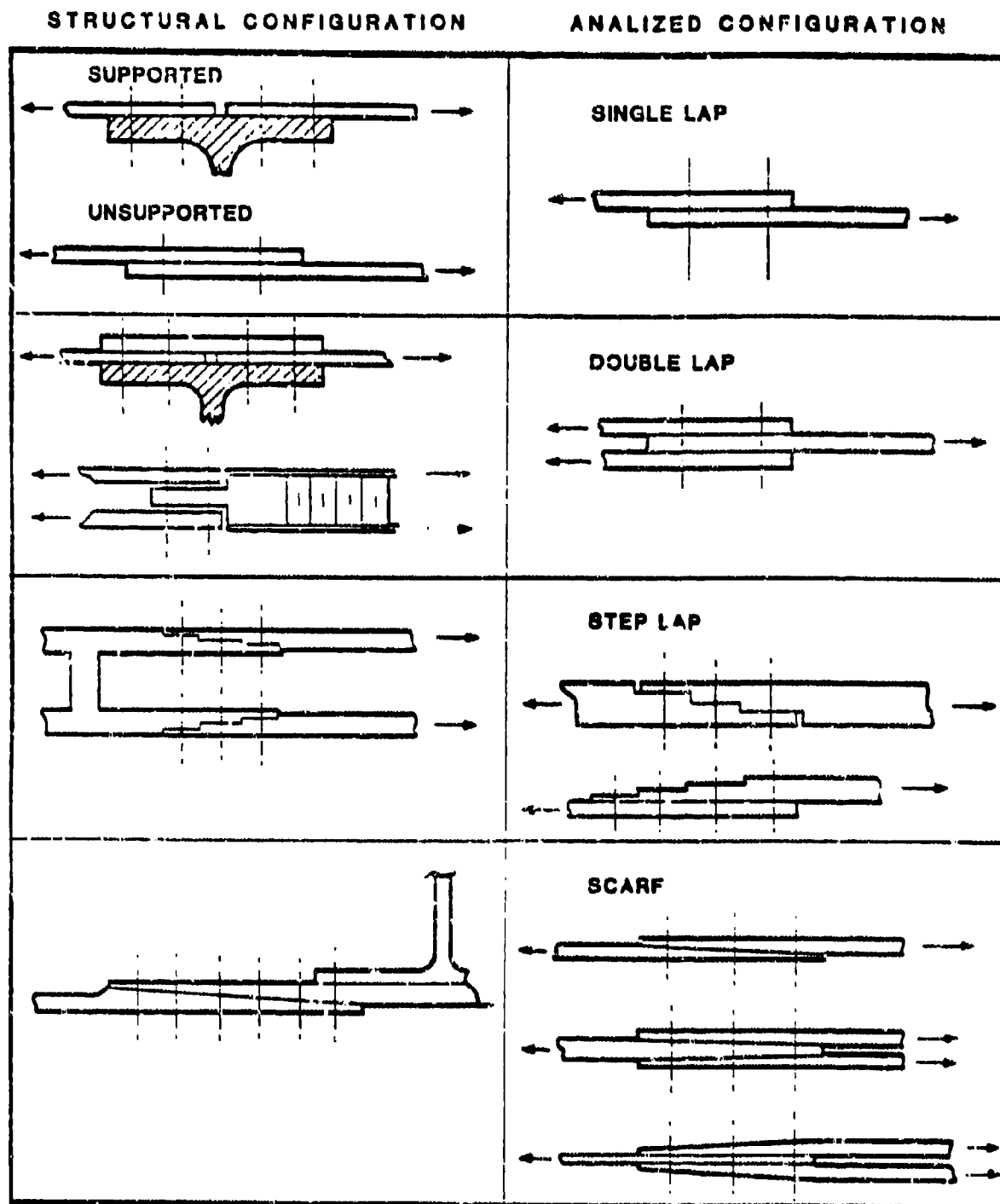


Figure 8. Structural and Analyzed Bolted Joint Configurations.

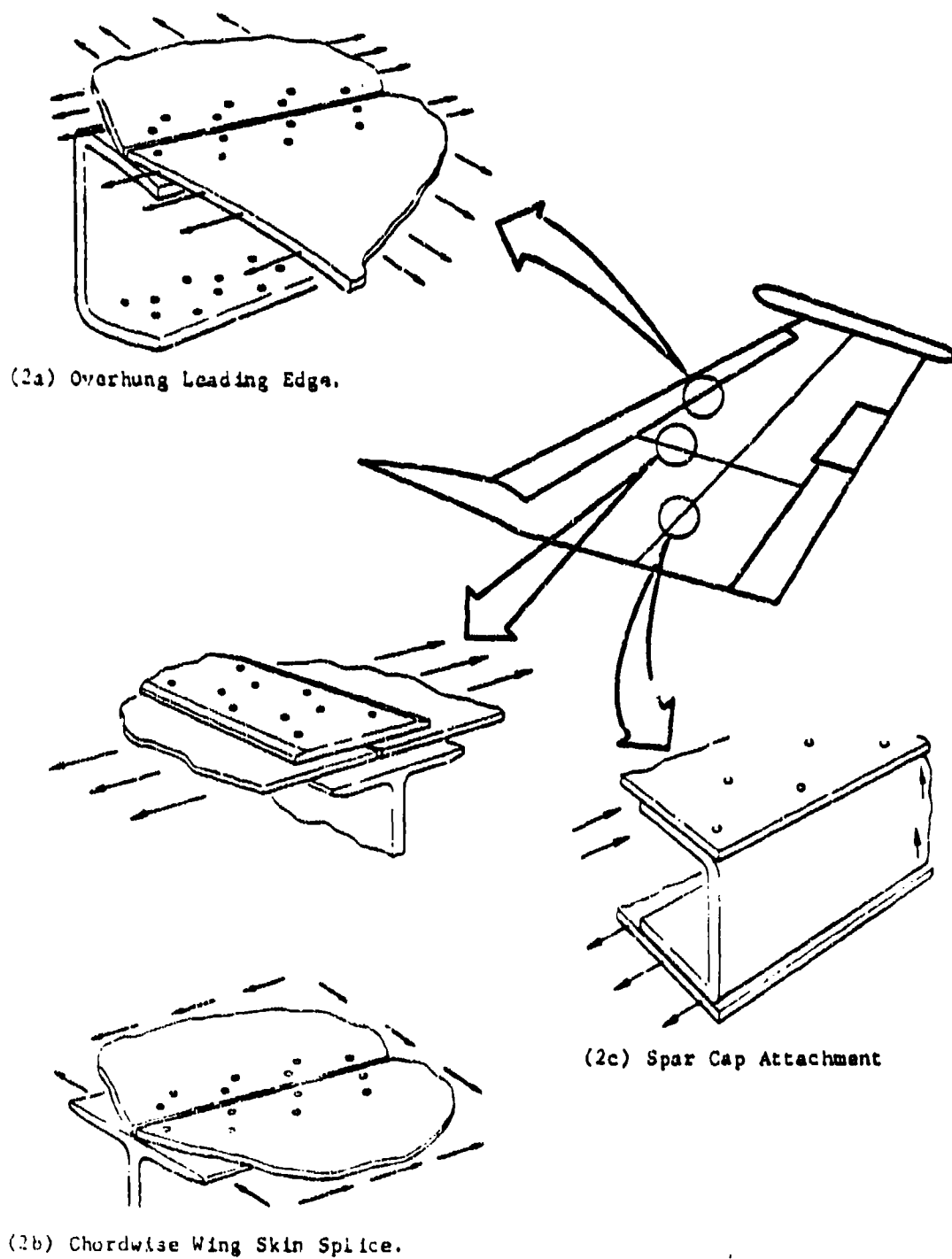


Figure 9. Inplane Loads in Typical Wing Skin-to-Substructure Attachments.

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.

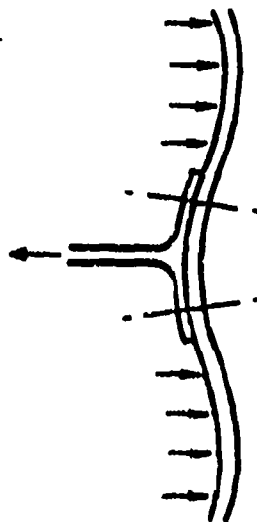
(5)

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

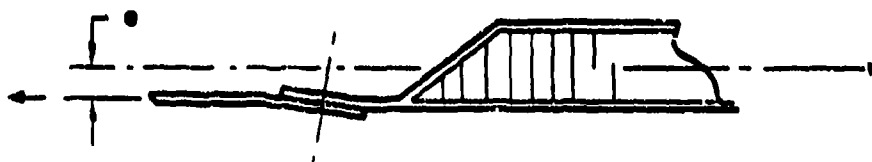
(6)

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.)**



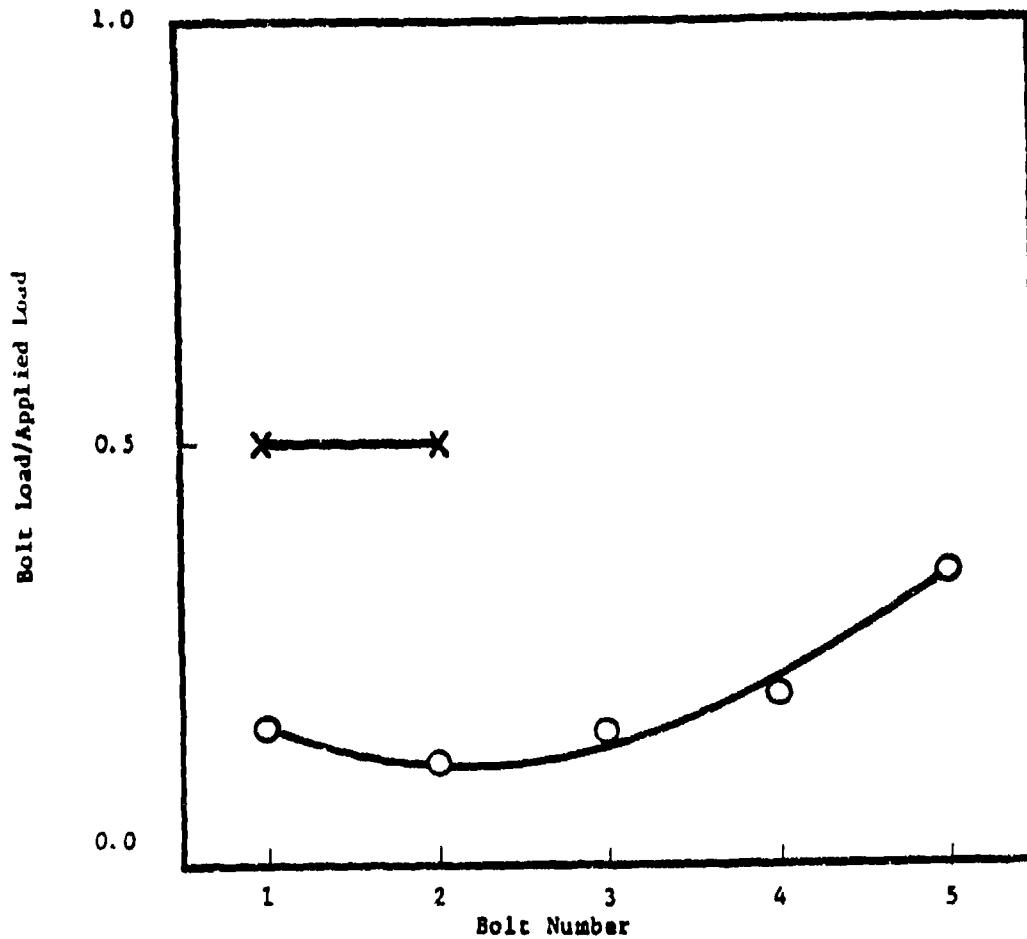
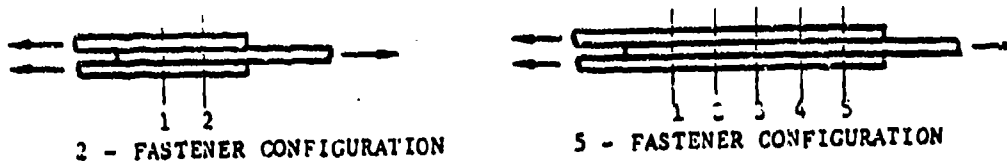
B. OUT-OF-PLANE JOINT LOAD DUE TO LOAD PATH ECCENTRICITY

**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).

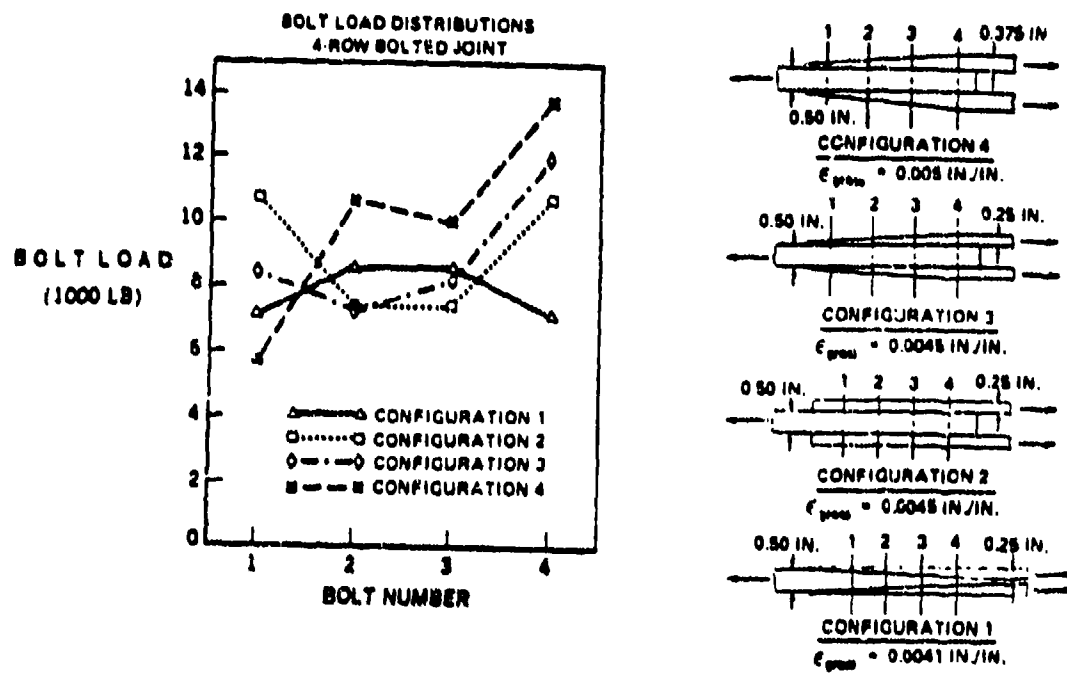


Figure 12. Effect of Joint Configuration on Fastener Load Distribution (Reference 9).

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).

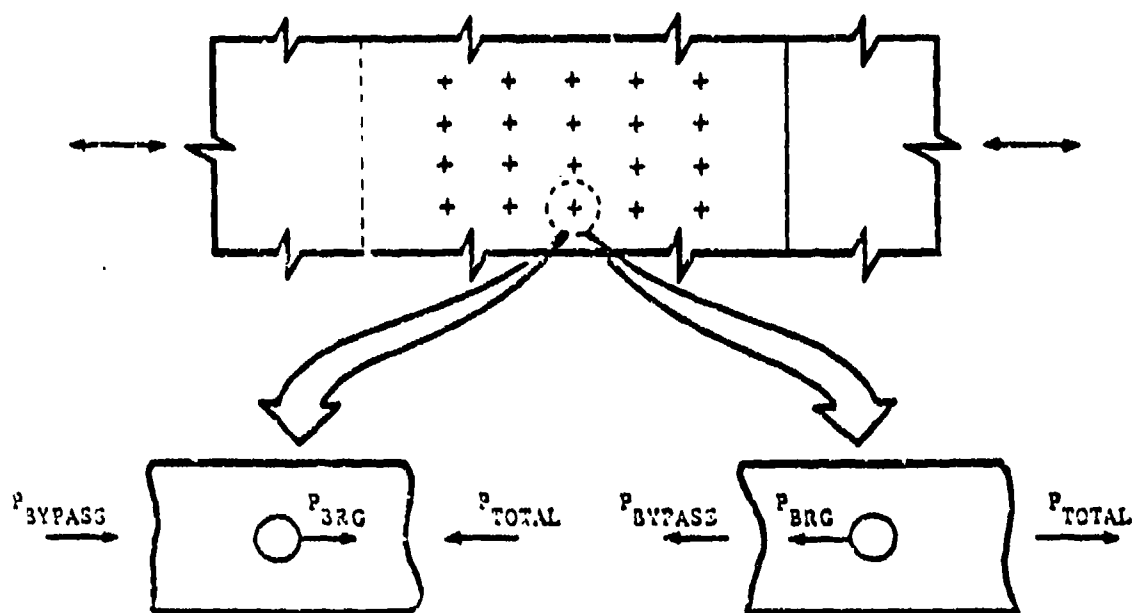
In ductile metals, minimal interaction is assumed between the bearing load and the by-pass load. However, in composites, a significant interaction has been demonstrated between the two loads under tensile loading (see Figure 13). Only a minimal interaction is observed under compression (see Figure 13). 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)



Bearing Stress at Failure

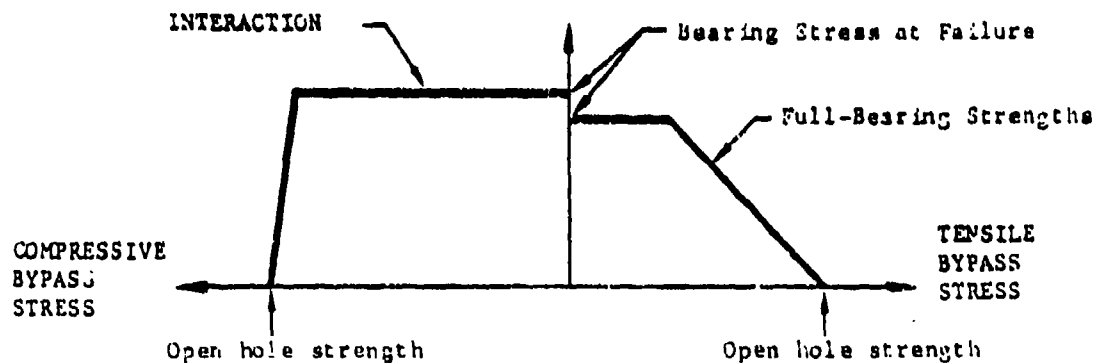
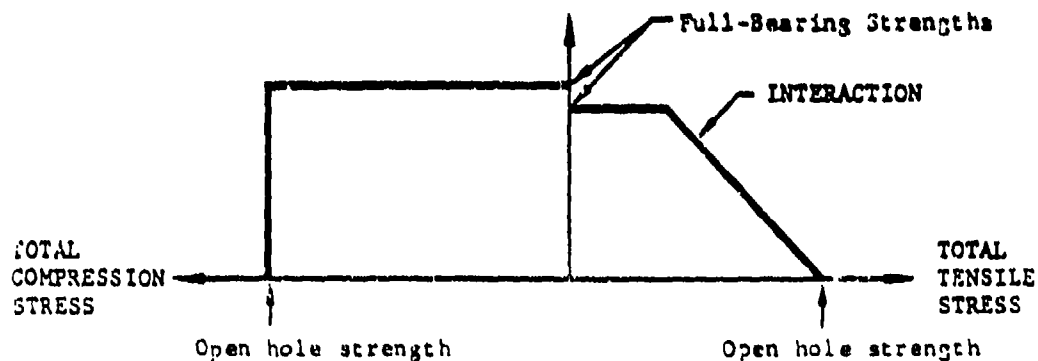


Figure 13. Interaction Between Bearing and by-Pass loads at a Fastener Location.

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.

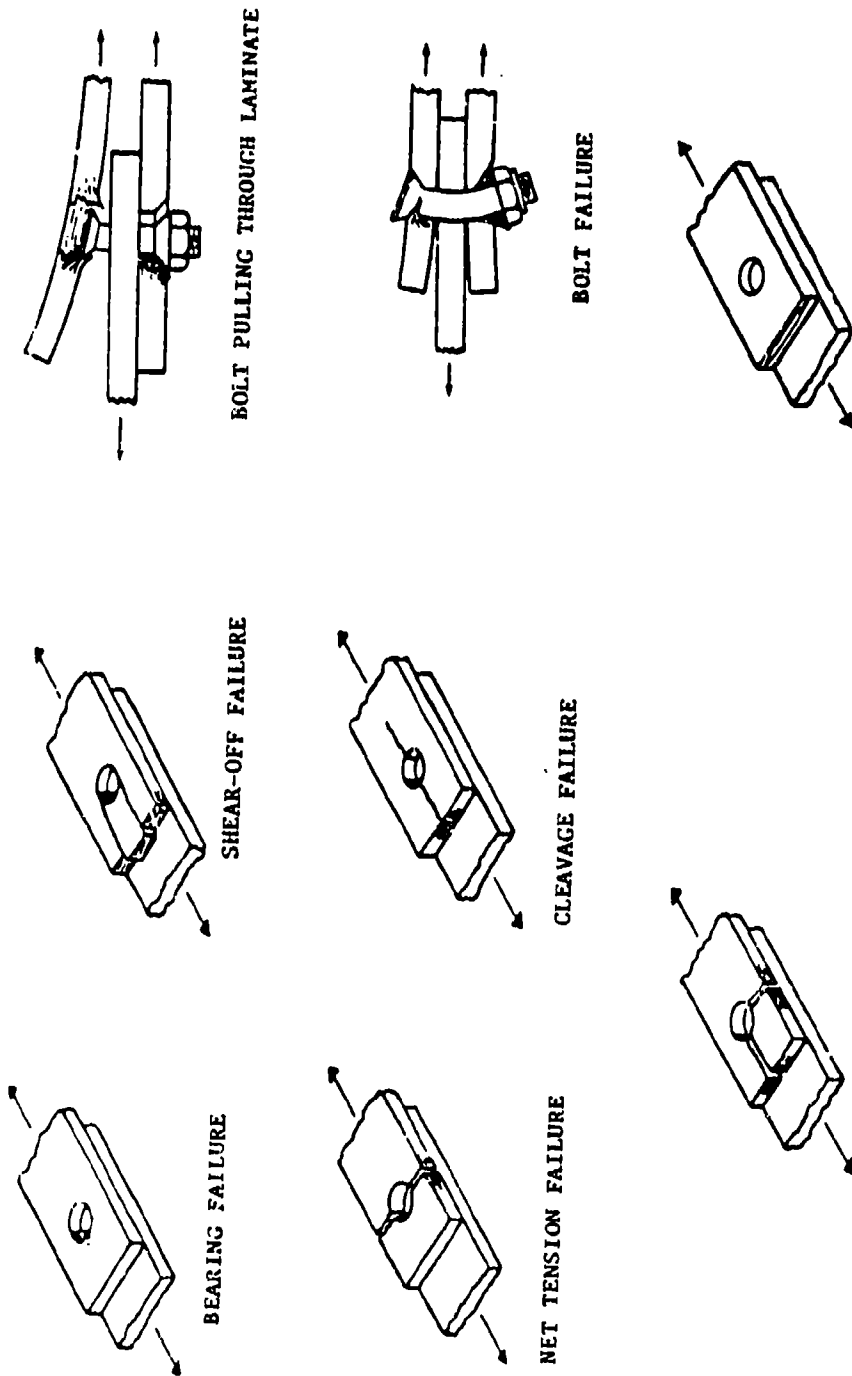
(10)

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).



COMBINATION FAILURE

DELAMINATION FAILURE

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

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

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

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

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

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

Tension head fasteners are preferred over shear head fasteners when the countersunk depth is below approximately 70% of the bolted plate thickness. (11)

2.5.2 Fastener Material

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

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

(12)

2.5.3 Fastener Size

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

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

(13)

2.5.4 Fastener Fit and Hole Quality

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

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

Fastener Material	Compatibility with Graphite/Epoxy Composites
Titanium and its alloys	Very Good
MP-35N, INCO 600 (Nickel, 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

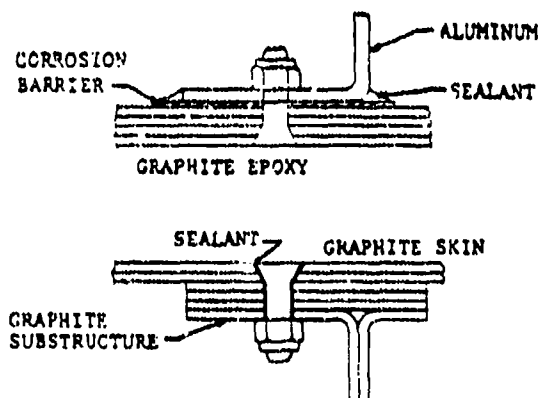


Figure 15. Galvanic Compatibility and Corrosion Prevention.

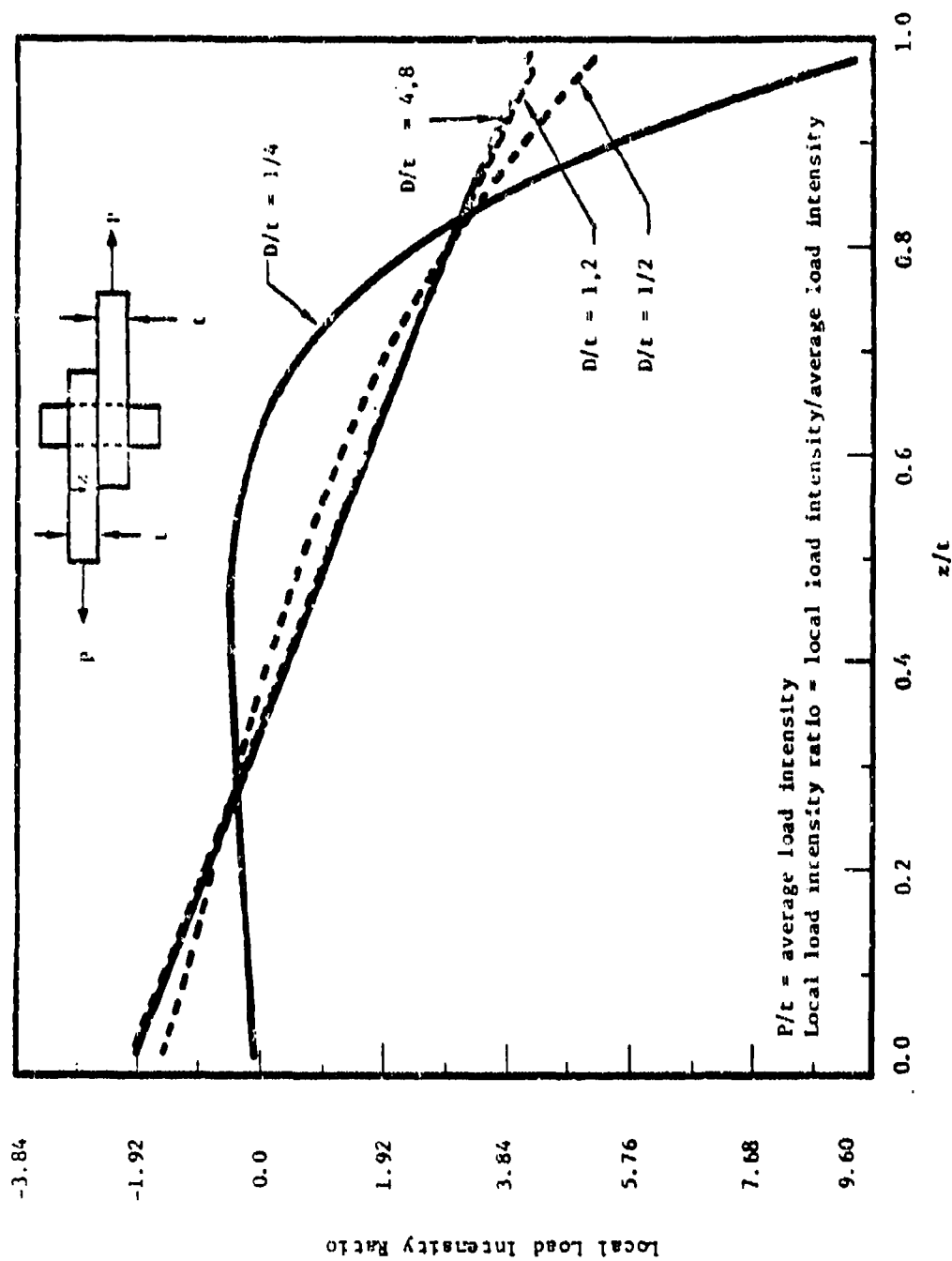


Figure 16. Effect of Fastener Size on the Load Distribution
 (Reference 6).

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 13).

	Percent Change in Strength*			
	COMPRESSION		TENS (OH)	
	RTW	250W	RTD	250W
OUT-OF-ROUND HOLES				
• 50/40/10 LAMINATE	-	-	<2.0	-
• 30/60/10 LAMINATE	-	-	-4.8	-
BROKEN FIBERS EXIT SIDE OF HOLE				
• SEVERE	-3.5	-12.2	-9.5	-
• MODERATE	-6.2	<2.0	-4.9	-
POROSITY AROUND HOLE				
• SEVERE	-12.1	-32.8	<2.0	-
• SEVERE WITH FREEZE-THAW	-13.3	-	-	-
• MODERATE	-5.4	-17.9	-	-
• MODERATE WITH FREEZE THAW	-7.9	-	-	-
IMPROPER FASTENER SEATING DEPTH (50% OF NOMINAL)				
• 80% THICKNESS	-	-	-23.2	-
• 100% THICKNESS	-	-	-56.9	-
TILTED COUNTERSINKS				
• AWAY FROM BEARING SURFACE	-	-20.0	<2.0	-
• TOWARD BEARING SURFACE	-	-22.7	-23.7	-
INTERFERENCE FIT TOLERANCES (INCH)				
• 50/40/10 @ 0.003	-	-	<2.0	+14.7
• @ 0.008	-	-	<2.0	+11.2
• 30/60/10 @ 0.003	-	-	<2.0	+2.4
• @ 0.008	-	-	<2.0	<2.0
FASTENER REMOVAL AND REINSTALLATION				
• 100 CYCLES	-	-7.4	<2.0	-

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

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

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

(18)

2.6 Bolted Laminate Properties

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

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

(19)

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

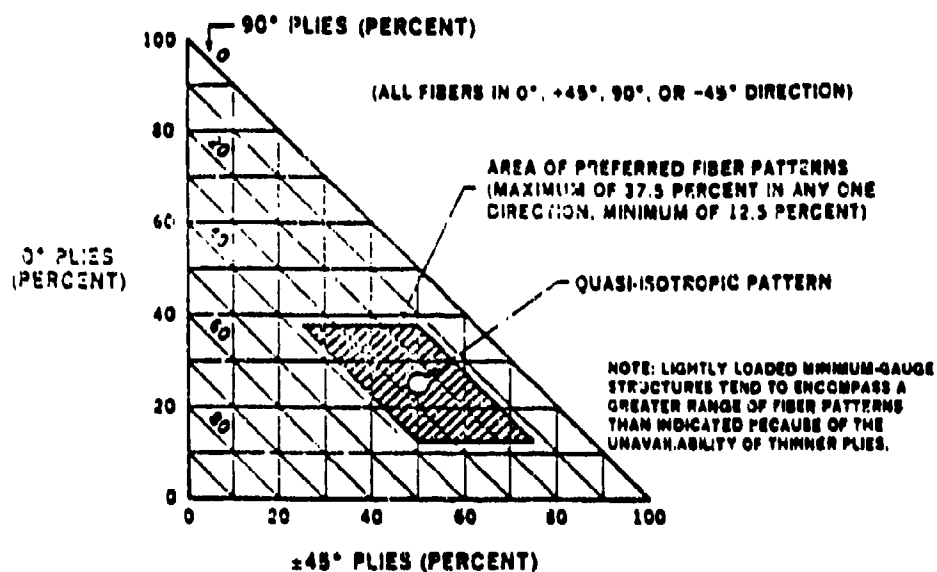


Figure 17. The Envelop of Bolted Laminate Layups for Realizing a Bearing Mode of Failure and the Maximum Bearing Strength (Reference 10).

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

The bearing strength of a laminate is maximum when its layup contains less than 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.

Ply 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)

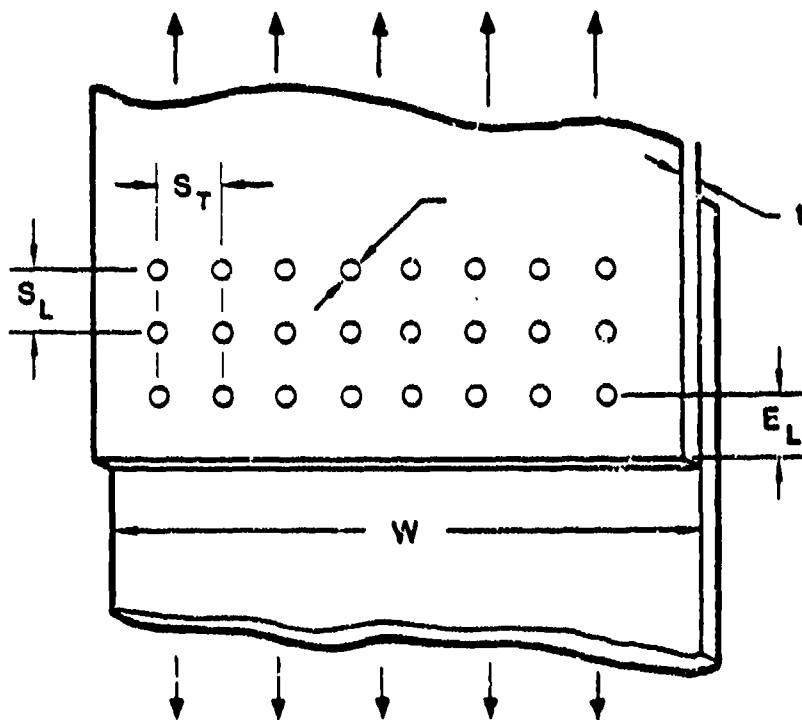
2.7 Fastener Spacing and Arrangement

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).

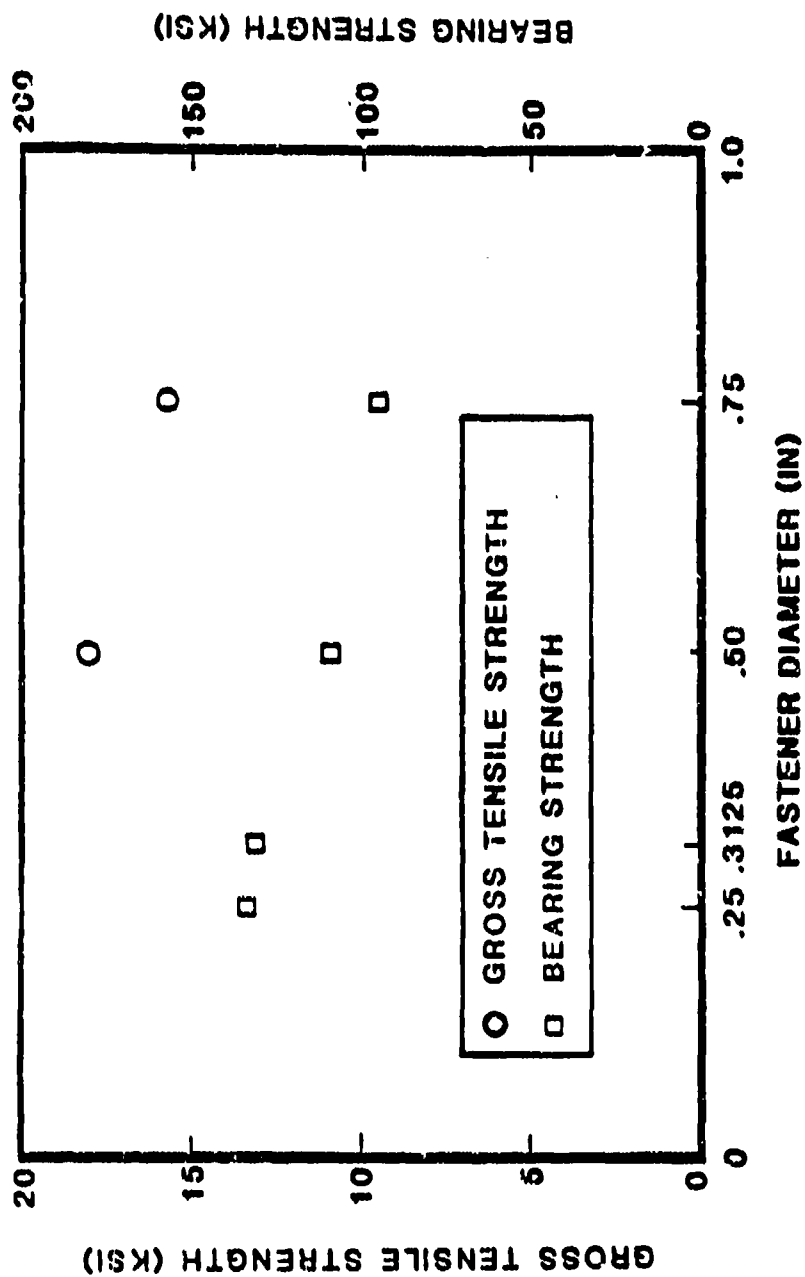
(22)

The bearing stress at failure decreases



PREDOMINANT LOAD DIRECTION

Figure 18. Geometrical Parameters for a Bolted Plate.



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

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

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

(23)

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

(24)

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

(25)

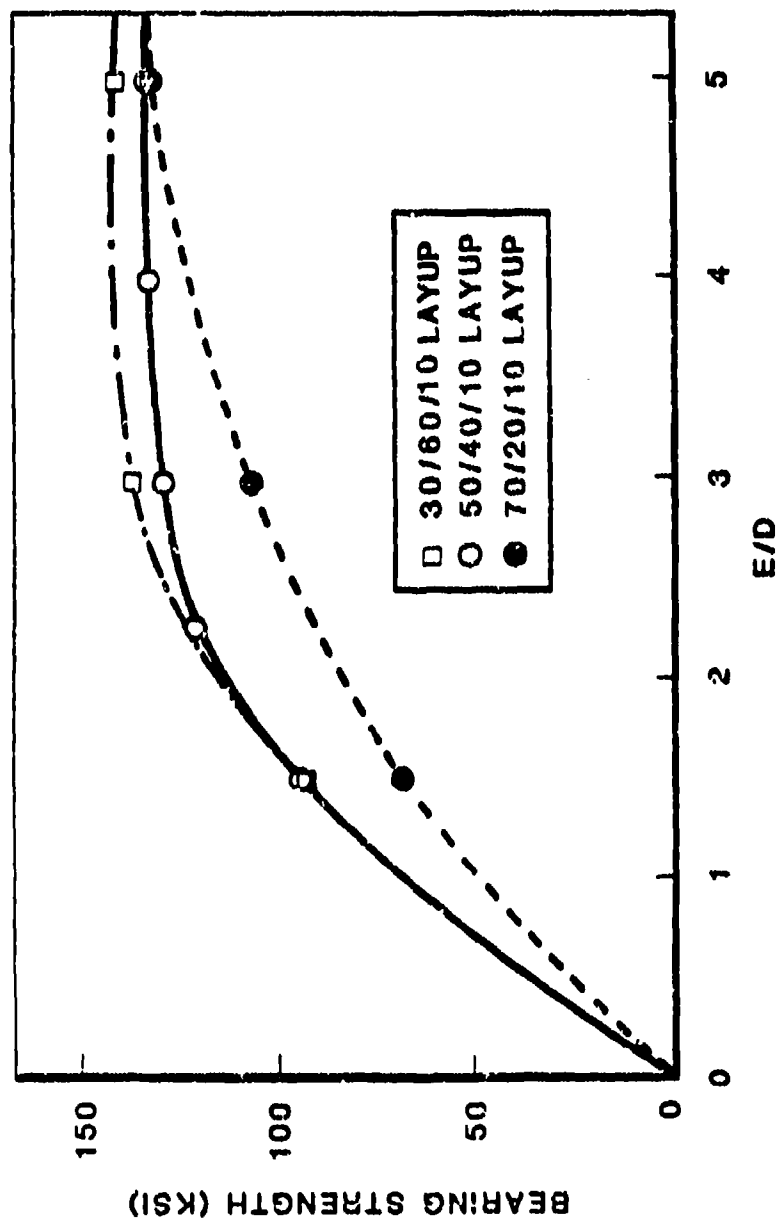
In summary, ensure that $D/t > 1$, $E/D > 3$, W/D (S_T/D) > 4 , $S_L/D > 4$, and the percentage of plies in any orientation is < 40 , to achieve a bearing failure mode and to realize the maximum joint strength.

(26)

2.8

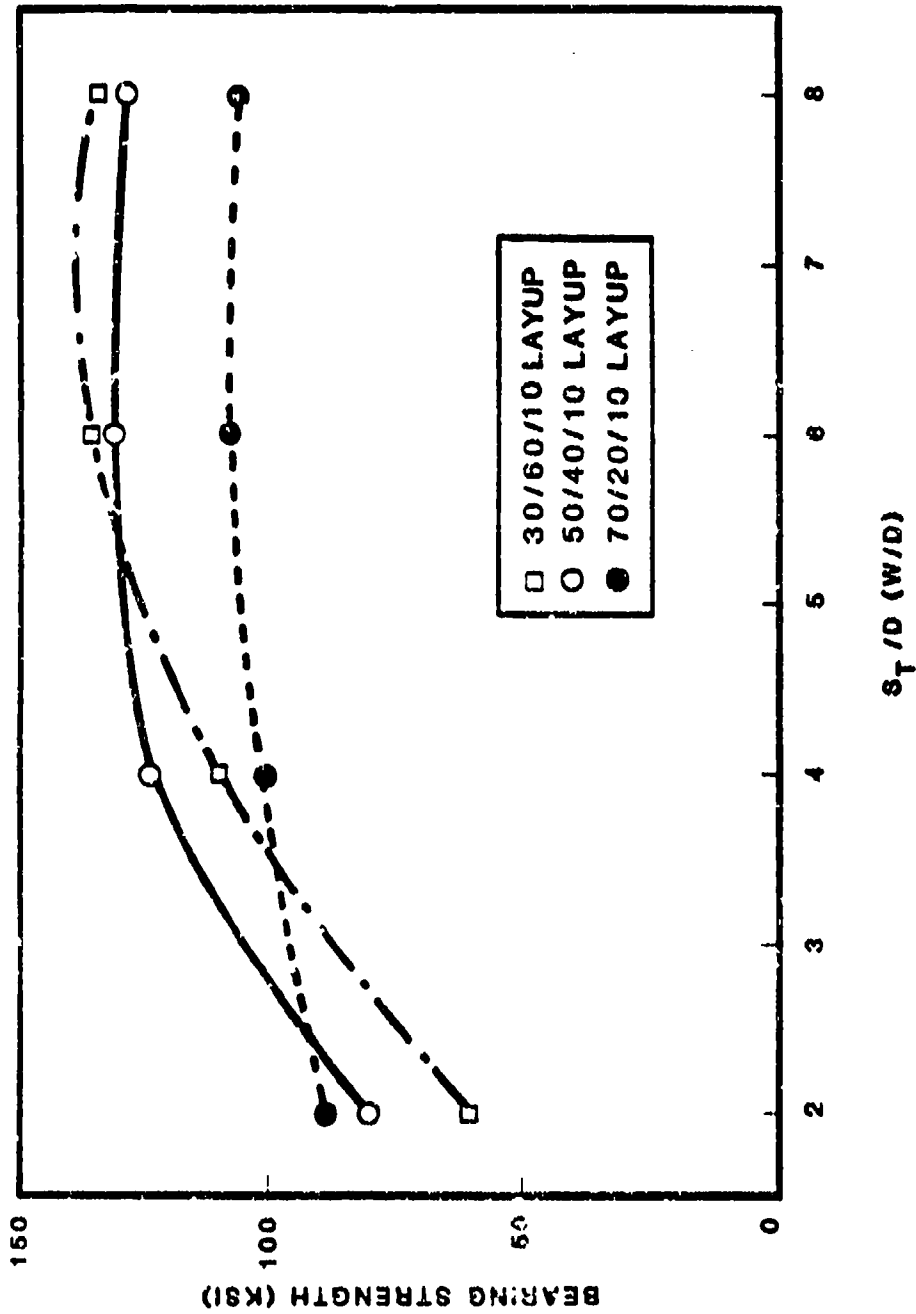
Joint Tailoring for Maximum Efficiency

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



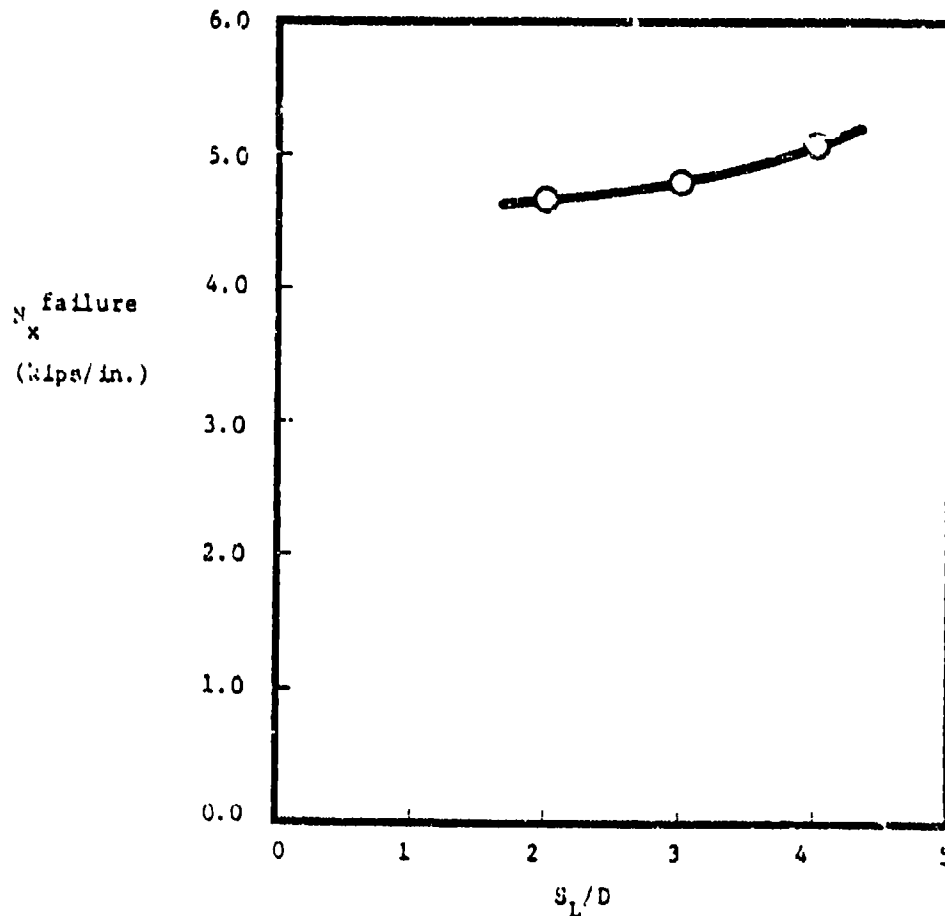
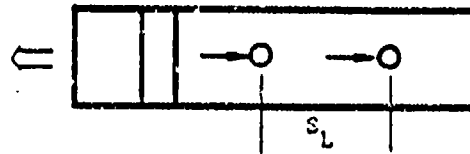
NOTE: Composite-to-metal joints in single shear; 20-ply AS1/3501-6 layups;
 0.31 in. aluminum plate; W/D = 6, Torque = 100 in-lbs; Protruding Head,
 steel fastener; D = 5/16 in.

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



NOTE: Composite-to-metal joints in single shear; 20-ply, AS13601-6 layups;
 0.31 in. aluminum plate; $E/D = 3$; Protruding head steel fastener;
 $D = 5/16$ in.; Torque = 100 in-lb

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; $S_L/D=W/L=6$; 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)

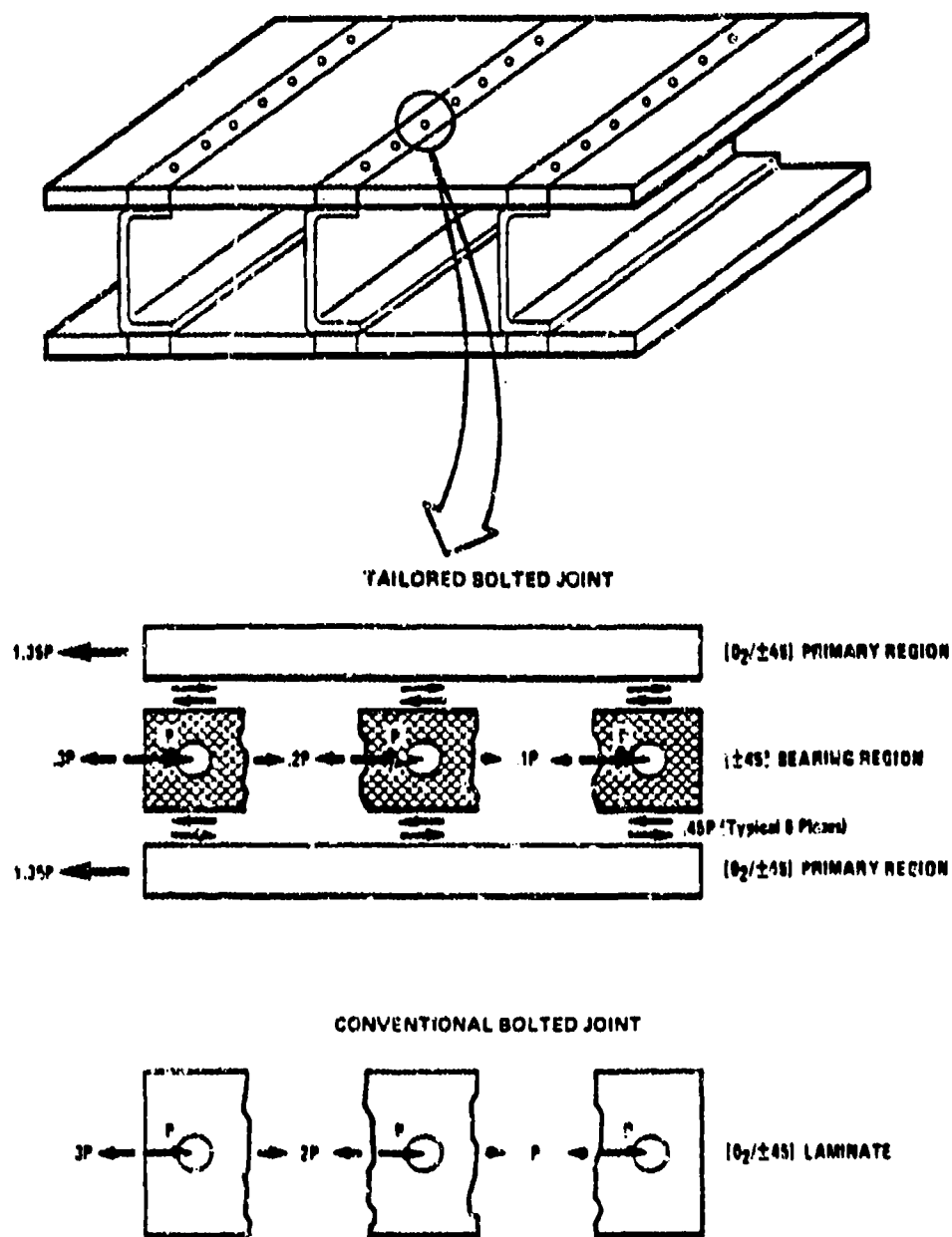


Figure 23. A Sample Tailored Joint.

2.9 Durability Considerations

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

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

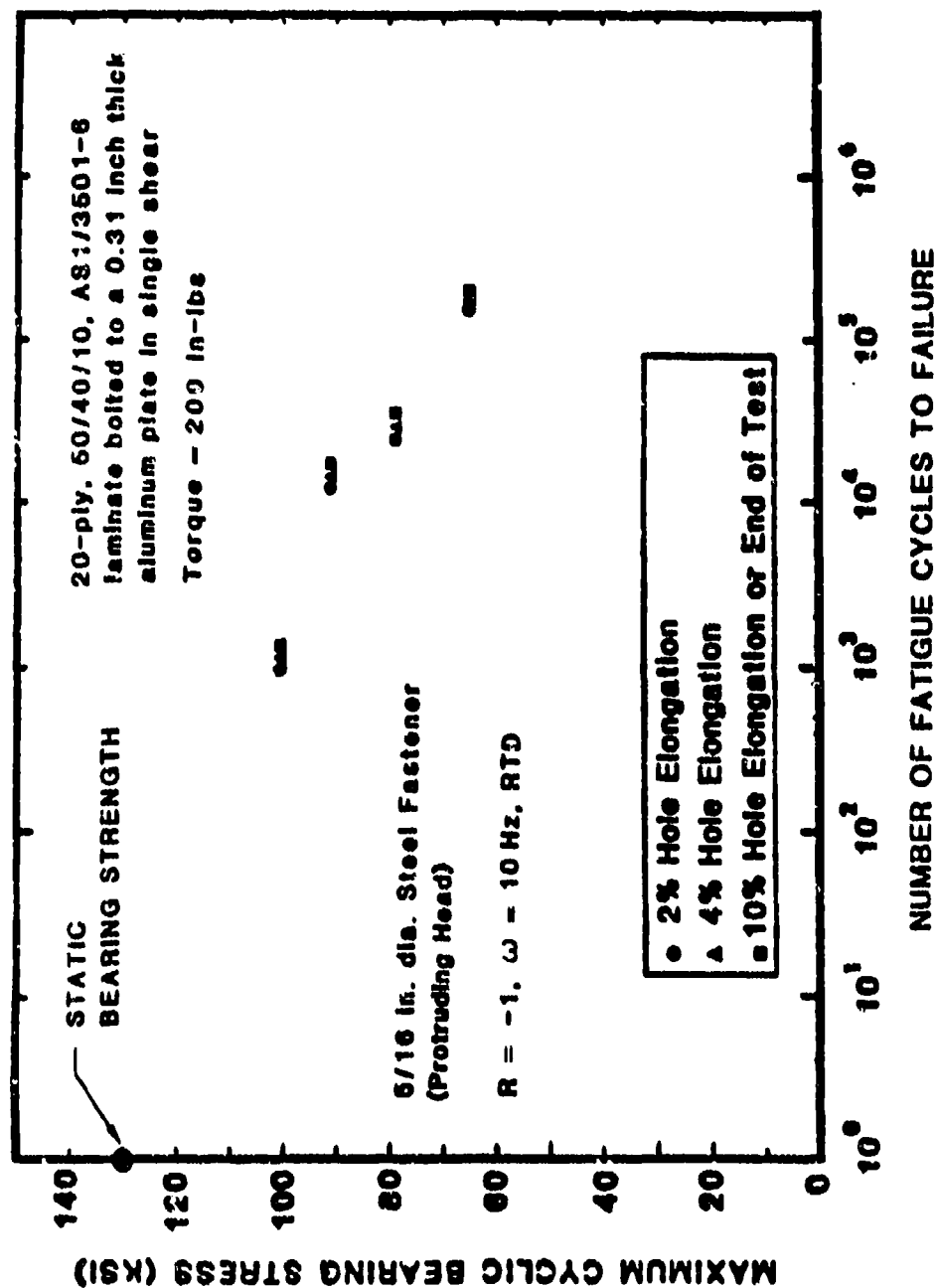


Figure 24. Effect of Maximum Cyclic Bearing Stress on the Number of R-1 Fatigue Cycles to Cause Specified Hole Elongations in a Bolted Laminate.

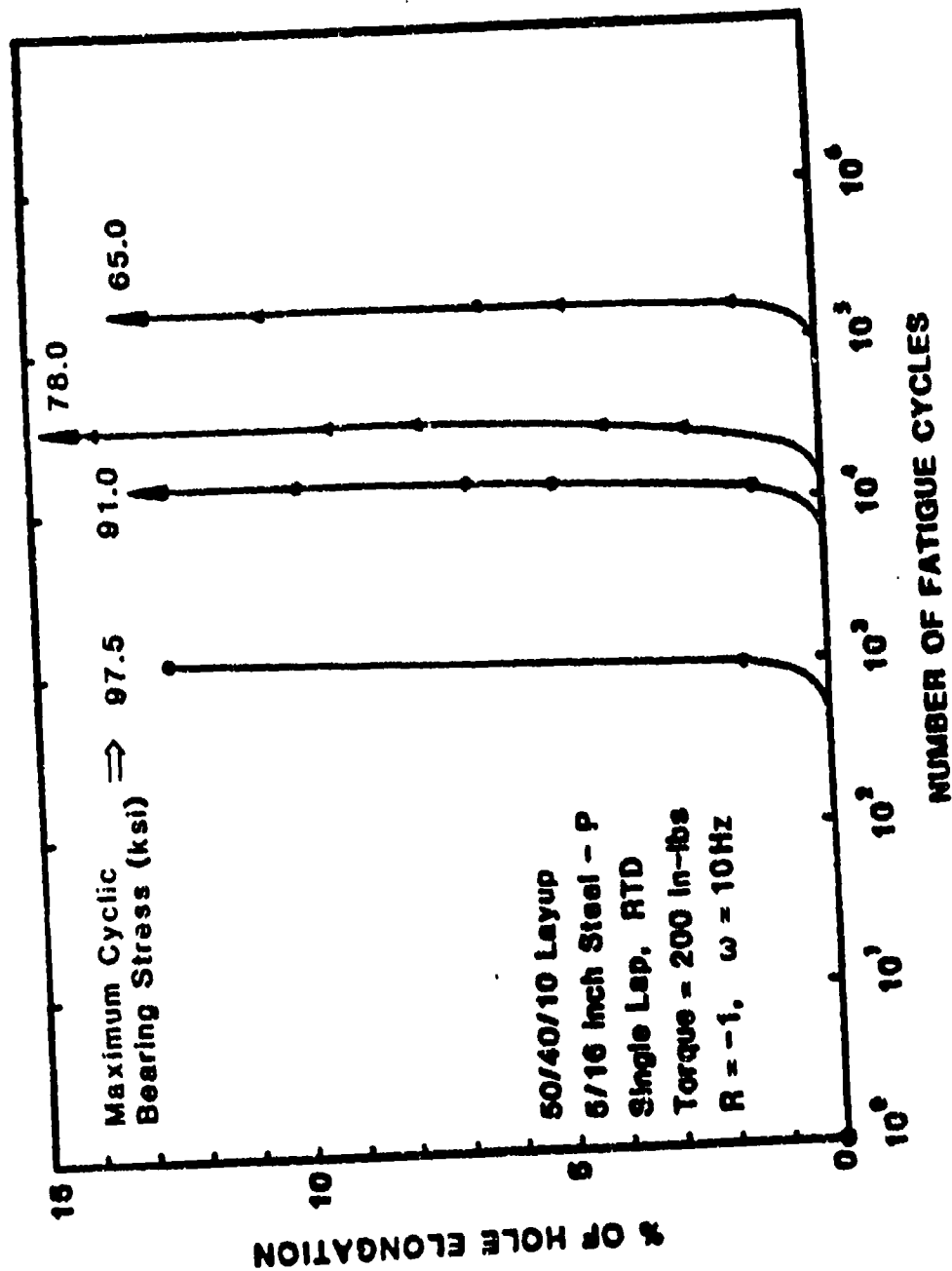


Figure 25. Effect of Maximum Cyclic Bearing Stress on the Hole Elongation Rate for a Bolted Laminate under $R=-1$ Loading.

SECTION 3

STRENGTH ANALYSIS OF BOLTED COMPOSITE STRUCTURES

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

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

3.1 Description of SASCJ Analysis

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

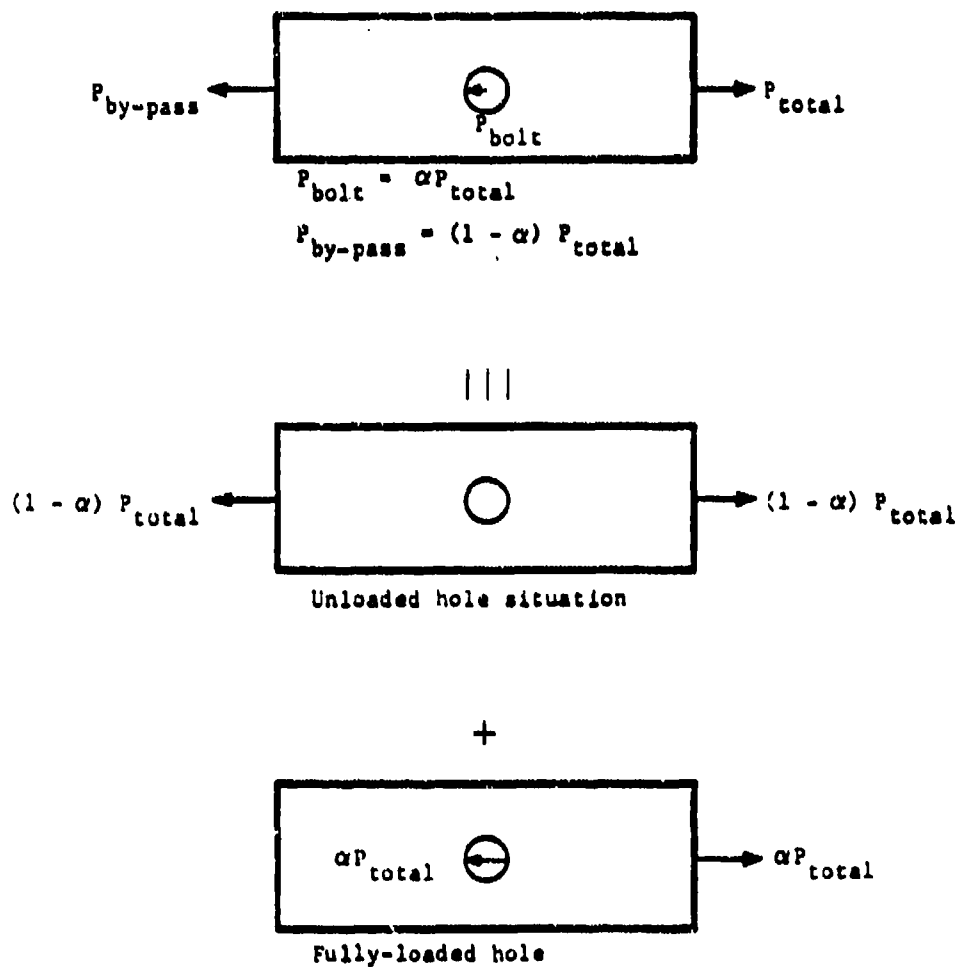


Figure 26. Schematic Representation of a General Single-Fastener Situation as a Superposition of Unloaded and Fully Loaded Hole Situations.

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

3.1.1 Strength Analysis Procedure for Fully-Loaded Holes

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

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

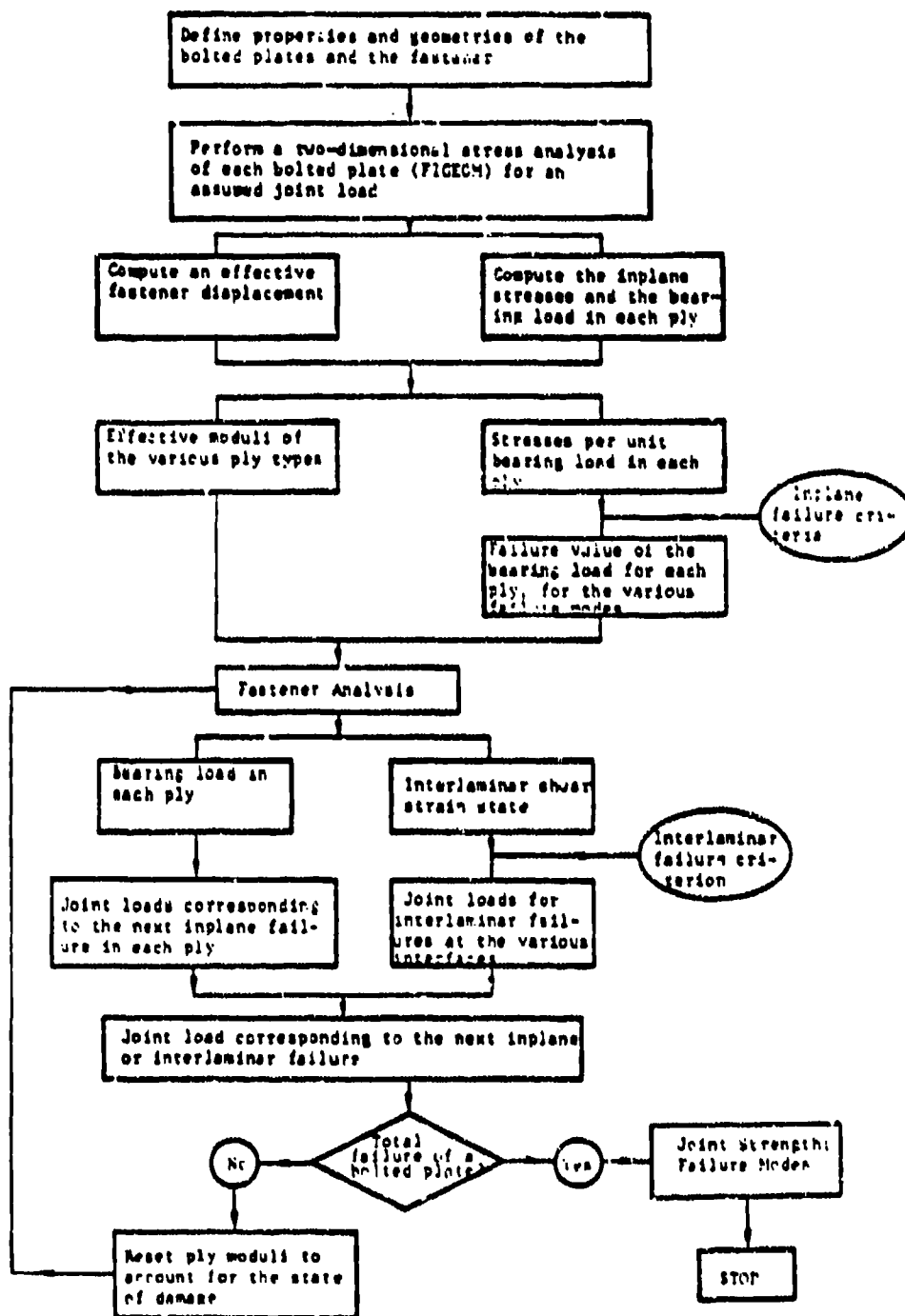


Figure 27. Flowchart for the Strength Analysis of Laminates with Fully Loaded Holes.

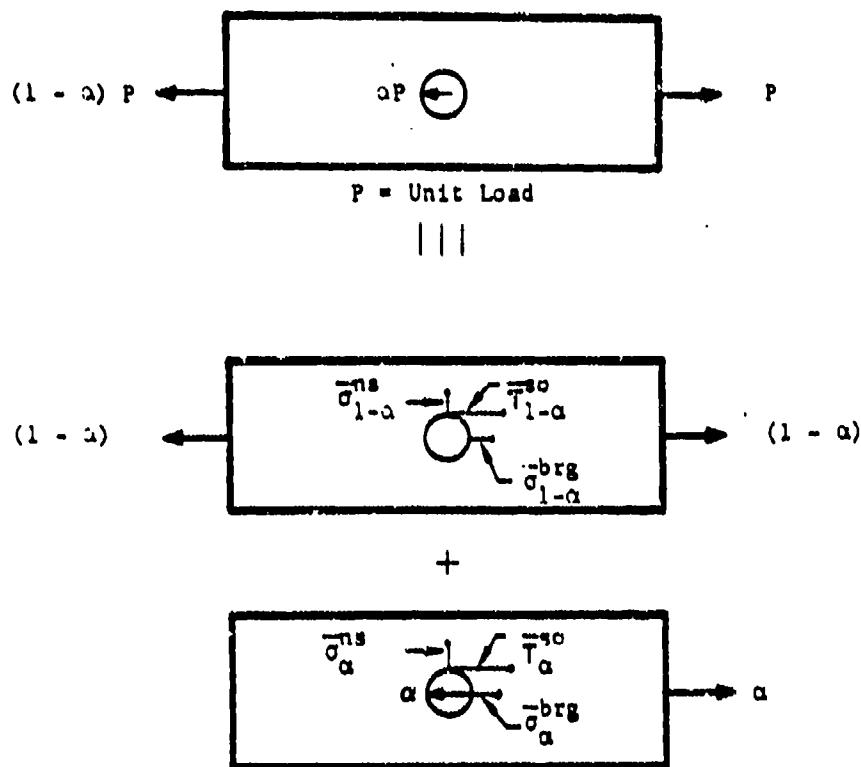
failures, determines the first failure in a bolted plate and the corresponding joint load value.

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

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

3.1.2 Strength Analysis Procedure for Partially-Loaded Holes

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



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

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

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

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-out and bearing. The maximum strain criterion predicts ply failure based on fiber failure. The Hoffman and Tsai-Hill criteria predict ply failure accounting for biaxial stress interaction that is ignored by the first three criteria.

The point stress failure criterion predicts net section, shear-out and bearing failures when the appropriate stress components at selected locations attain unnotched specimen failure values (see Figure 29). a_{ons} , a_{oso} 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

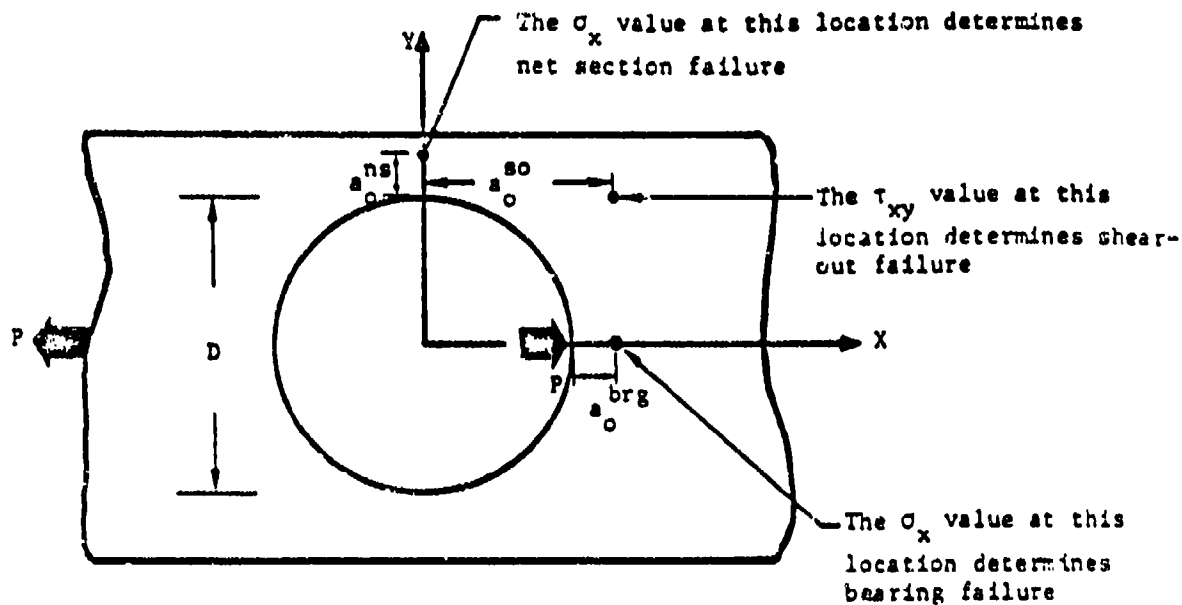


Figure 29. The Characteristic Distances used in the Point Stress Failure Criteria

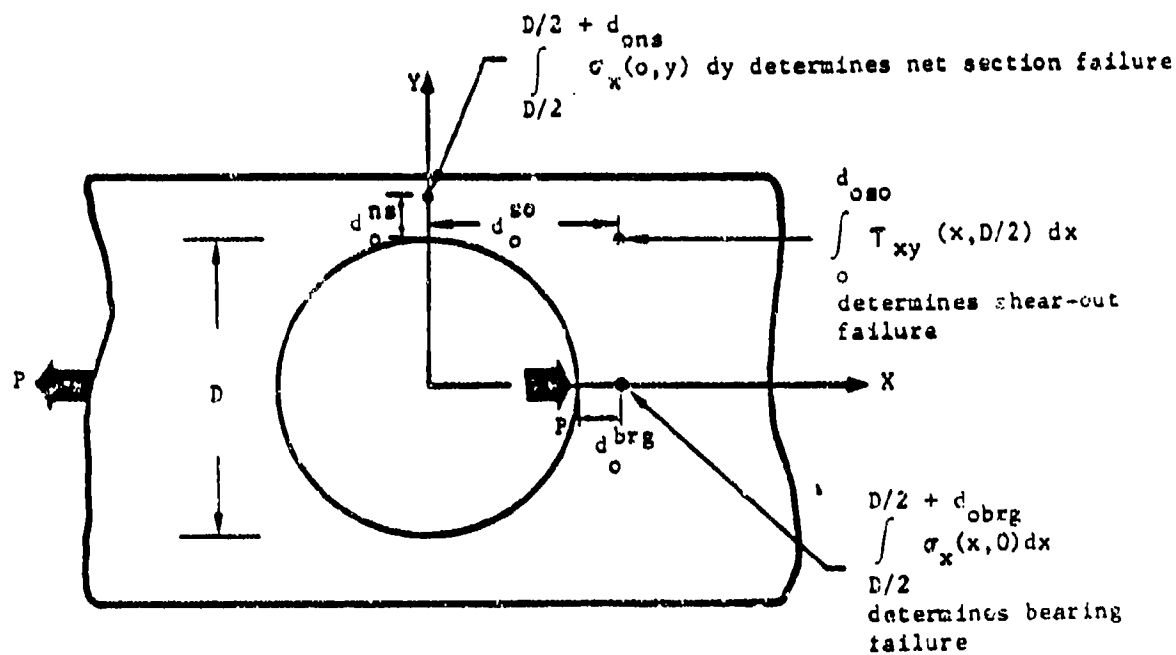


Figure 30. The Characteristic Distances Used in the Average Stress Failure Criteria.

predicted. When τ_{xy} (a_{OSO} , $D/2$) 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_{ONS} , d_{OSO} , and d_{OBRG}) that are larger in magnitude compared to those used in conjunction with the point stress criterion (see Figure 30).

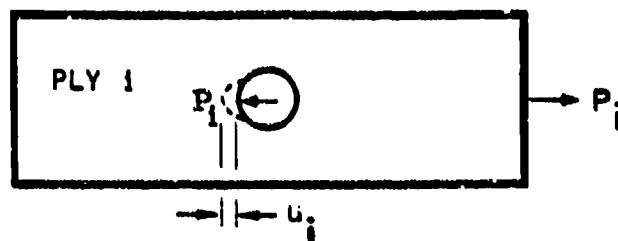
Of the three ply failure modes, only the net section mode causes the ply to become almost ineffective (total failure). The bearing mode of failure causes the ply to suffer a reduction in its effective modulus without losing its load-carrying capacity. The shear-out mode of failure causes a ply to become 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 SASCF 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_o) from the hole boundary (see Figure 33). The location along this path where the selected criterion is satisfied determines the failure location. The maximum strain criterion predicts fiber failure in a ply (total ply failure) when its fiber directional strain exceeds the failure values (ϵ_{11}^{tu} or ϵ_{11}^{cu}).

The Hoffman failure criterion, based on inplane ply

$$K_2 = aK_1$$

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



$$q_i = P_i / h_i$$

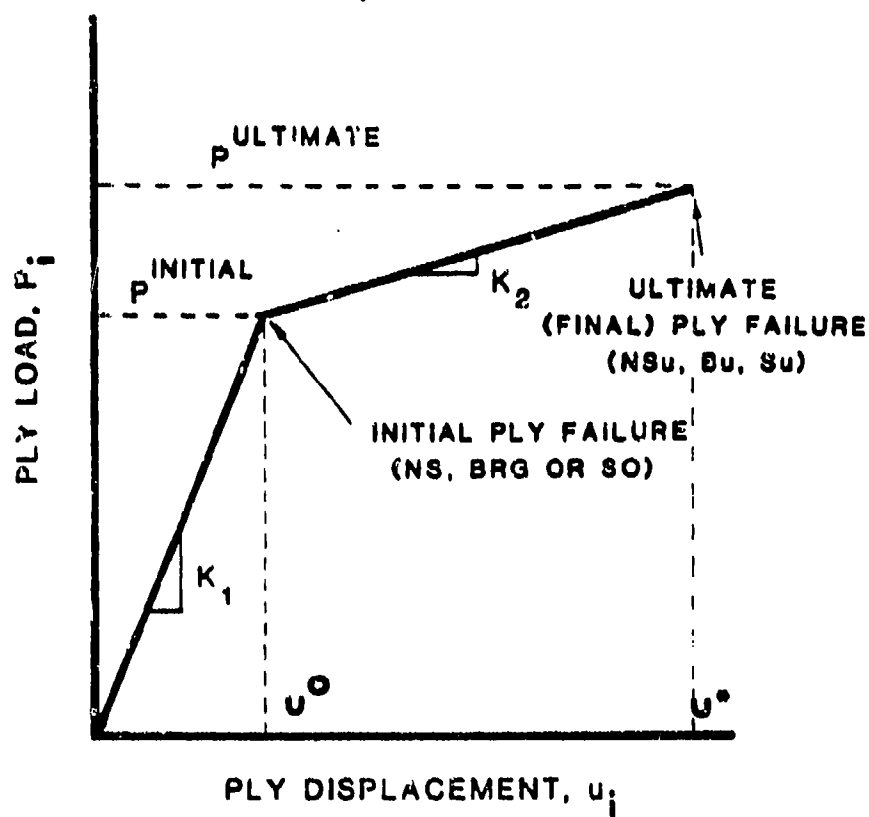


Figure 31. Bilinear Elastic Behavior of a Ply.

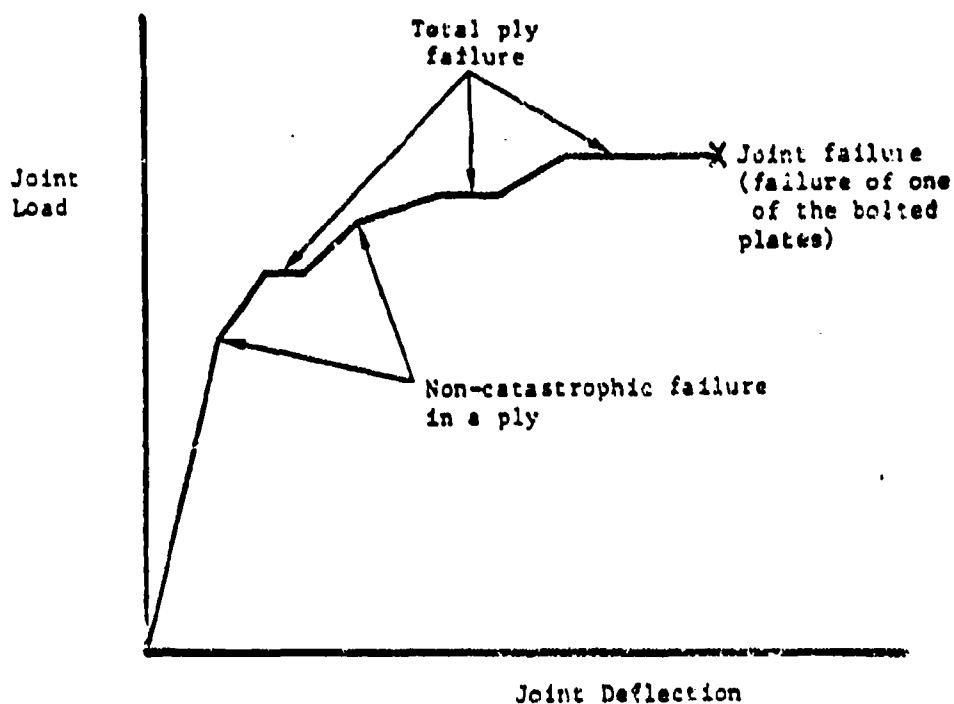


Figure 32. A Schematic Representation of the Overall Load Versus Deflection Response of the Joint.

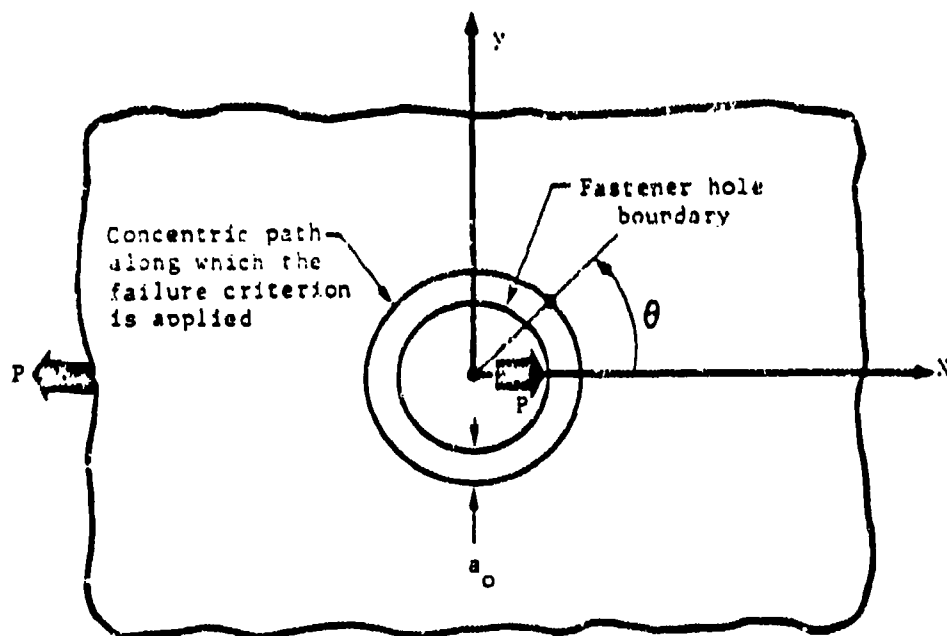


Figure 33. The Characteristic Distance (a_0) Defining the Region Where the Maximum Strain, Hoffman or Hill Criterion is Applied.

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

$$\begin{aligned} & (\sigma_1^2 - \sigma_1\sigma_2)/X_cX_t + \sigma_1(X_c - X_t)/X_cX_t + \sigma_2^2/Y_cY_t + \sigma_2(Y_c - Y_t)/Y_cY_t + \\ & \sigma_6^2/S^2 = H \end{aligned}$$

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 SASCF 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

$$a = [(\sigma_1^2 - \sigma_1\sigma_2)/X_cX_t + \sigma_2^2/Y_cY_t + \sigma_6^2/S^2]/P_1^2$$

$$b = [(X_c - X_t)\sigma_1/X_cX_t + (Y_c - Y_t)\sigma_2/Y_cY_t]/P_1$$

$$c = -1, \text{ and}$$

$$P_1 = \text{ply load at which } \sigma_1, \sigma_2 \text{ and } \sigma_6 \text{ are computed}$$

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 $X_c = X_t$ and $Y_c = Y_t$. This criterion, therefore, does not account for different strengths under tension and compression. The ply failure load (P_f) in this case is computed to be $1/\sqrt{a}$.

3.1.4 Interlaminar Failure Criterion

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

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

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

3.2 SASCJ Input Description

SASCJ assumes a uniaxial tensile or compressive load to

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

A sample SASCH 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 SASCH 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, SASCH 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). SASCH then requests the user to specify the thickness of the metallic plate (0.25). For the laminated plate, SASCH 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]₀. SASCH automatically assumes a metallic plate to be divided into thirty identical layers. The number of layers in a laminate is controlled

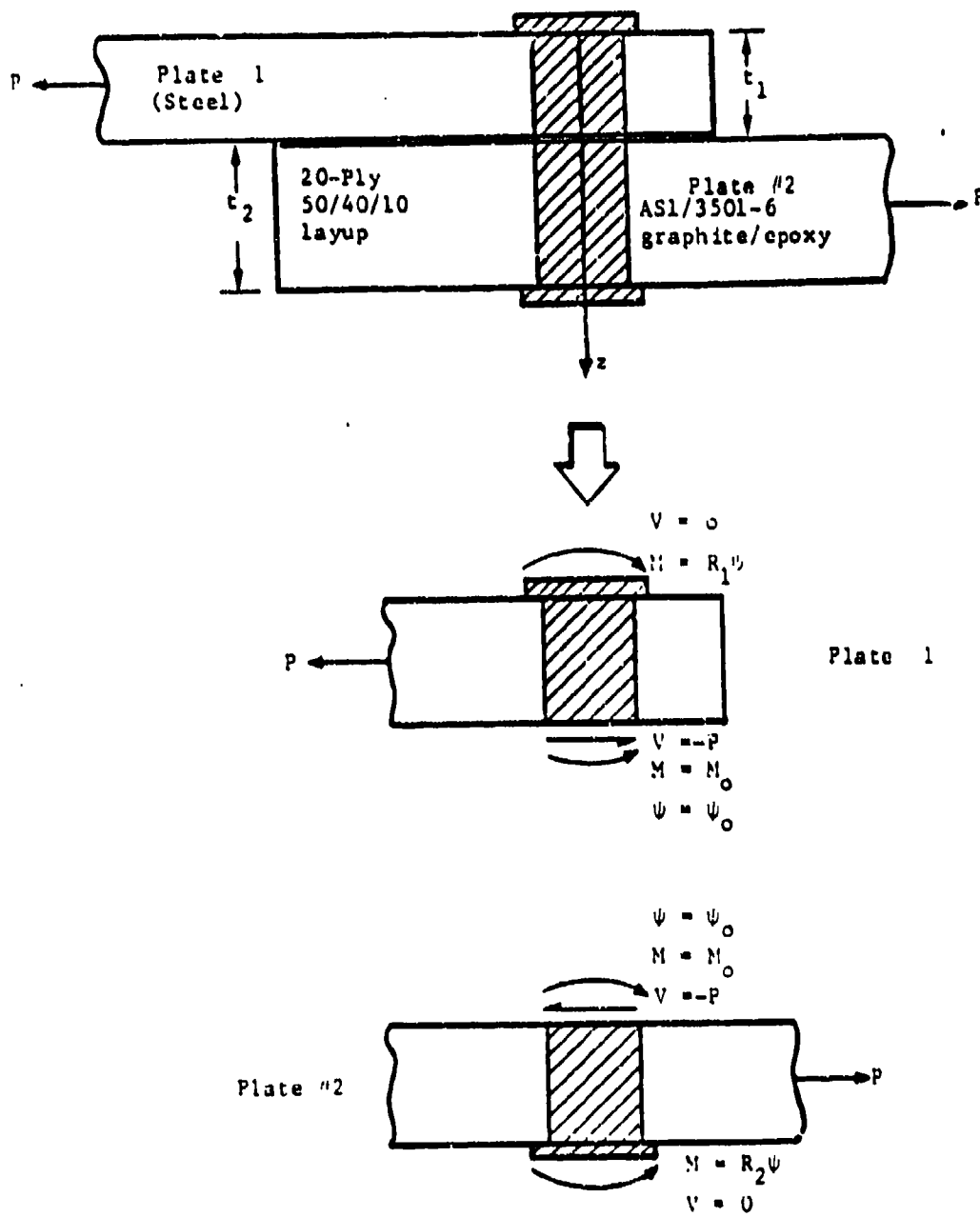


Figure 34. SASCJ Analysis of a Joint in Single Shear.

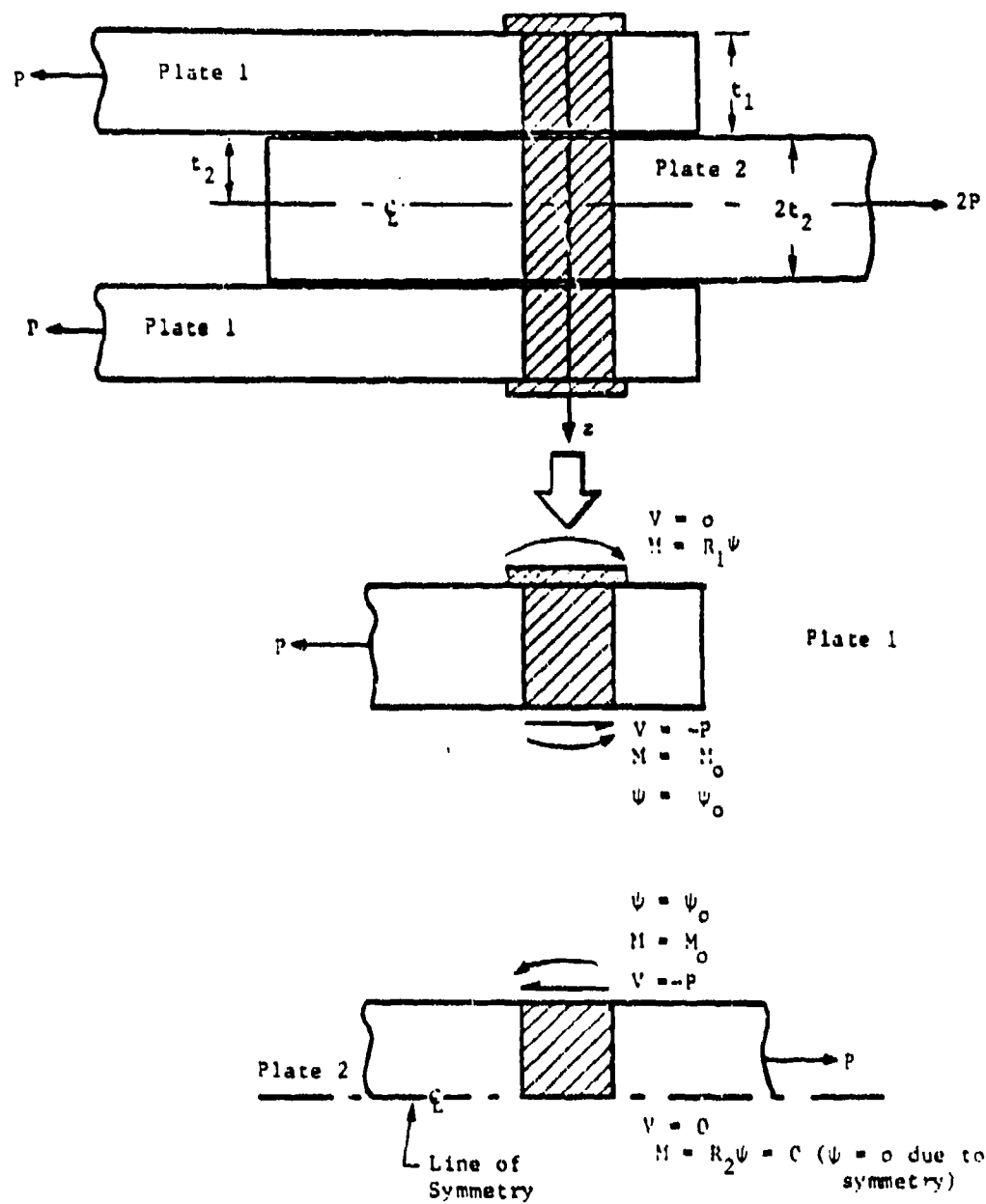


Figure 35. SASCJ Analysis of a Joint in Double Shear.

```

171500061 TEMPERATURE ASSUMED AS MEMBERTEMP
PROGRAM SASCIJ
PROGRAM SASCIJ PREDICTS FAILURE LOADS OF
MECHANICALLY FASTENED COMPOSITE LAP JOINTS.
SINGLE OR DOUBLE LAP SHEAR JOINTS.
PROGRAM ASSURES THAT INPUT PARAMETERS ARE
IN INCHES UNITS - LENGTHS ARE INPUT
IN INCHES AND MODULI AND STRENGTHS ARE
EXPRESSED IN PSI.
ENTER BYPASS RATIO ALPHA:
ALPHA=0 FOR FULL BEARING
ALPHA=1 FOR OPEN HOLE
0 (ALPHA=1 FOR GENERAL BYPASS
7
0.99
ENTER:
1 FOR STATIC TENSION
2 FOR STATIC COMPRESSION
7
1
ENTER:
1 FOR SLS (SINGLE LAP SHEAR)
2 FOR DLS (DOUBLE LAP SHEAR)
7
1
IS THE TOP PLATE A COMPOSITE OR A METAL ?
ENTER C OR M IN THE FIRST FIELD
7
C
INPUT MATERIAL DESCRIPTION OF THIS PLATE
EX: AGA/3501-6
7
AGA/3501-6 (105-0/-45/012/0/30)0
IS THE BOTTOM PLATE A COMPOSITE OR A METAL ?
ENTER C OR M IN THE FIRST FIELD
7
C
INPUT MATERIAL DESCRIPTION OF THIS PLATE
EX: AGA/3501-6
7
AGA/3501-6 (105-0/-45/012/0/30)0
NOTE: FOR COMPUTATIONAL PURPOSES A
METALLIC PLATE IS REVELED AS P.30 PLY
LAMINATE OF 0 DEGREE PLYS WITH ISOTROPIC
MATERIAL PROPERTIES
NOTE: NUMERICAL DESIGNATIONS FOR THE
PLATES ARE:
TOP PLATE - NO 1
BOTTOM PLATE - NO 2
INPUT NUMBER OF PLYS IN PLATE NO 2
7
14 (N = 2)
20
INPUT THICKNESS OF PLATE NO 1
7
0.25
INPUT PLY THICKNESS IN PLATE NO 2
7
0.005
INPUT NUMBER OF DISTINCT PLY ORIENTATIONS
IN PLATE NO 2
7
4
FOR PLATE NUMBER 2 1
INPUT ORIENTATION OF PLY TYPE NO 1
7
0.0
INPUT ORIENTATION OF PLY TYPE NO 2
7
45.0
INPUT ORIENTATION OF PLY TYPE NO 3
7
-45.0
INPUT ORIENTATION OF PLY TYPE NO 4
7
90.0
INPUT TYPE OF PLY IN PLATE NO 2 FROM TOP
TO BOTTOM
PLY TYPE ORIENTATION
1 0.0 DEGREES
2 45.0 DEGREES
3 -45.0 DEGREES
4 90.0 DEGREES
INPUT TYPE OF PLY FOR PLY NO 1
7
1
INPUT TYPE OF PLY FOR PLY NO 2
7
2
INPUT TYPE OF PLY FOR PLY NO 3
7
1
INPUT TYPE OF PLY FOR PLY NO 4
7
2
INPUT TYPE OF PLY FOR PLY NO 5
7
1
INPUT TYPE OF PLY FOR PLY NO 6
7
2
INPUT TYPE OF PLY FOR PLY NO 7
7
1
INPUT TYPE OF PLY FOR PLY NO 8
7
2
INPUT TYPE OF PLY FOR PLY NO 9
7
1
INPUT TYPE OF PLY FOR PLY NO 10
7
2
INPUT TYPE OF PLY FOR PLY NO 11
7
1
INPUT TYPE OF PLY FOR PLY NO 12
7
2
INPUT TYPE OF PLY FOR PLY NO 13
7
1
INPUT TYPE OF PLY FOR PLY NO 14
7
2

```

Figure 36. Sample SASCIJ Input.

```

INPUT TYPE OF PLY FOR PLY NO 11
?
4
INPUT TYPE OF PLY FOR PLY NO 12
?
1
INPUT TYPE OF PLY FOR PLY NO 13
?
1
INPUT TYPE OF PLY FOR PLY NO 14
?
2
INPUT TYPE OF PLY FOR PLY NO 15
?
1
INPUT TYPE OF PLY FOR PLY NO 16
?
2
INPUT TYPE OF PLY FOR PLY NO 17
?
1
INPUT TYPE OF PLY FOR PLY NO 18
?
1
INPUT TYPE OF PLY FOR PLY NO 19
?
1
INPUT TYPE OF PLY FOR PLY NO 20
?
1
INPUT THE ENGINEERING PROPERTIES OF PLATE NO 1
INPUT YOUNG'S MODULUS AND POISSON'S RATIO
30.046 0.2
INPUT THE ENGINEERING PROPERTIES OF PLATE NO 2
INPUT YOUNG'S MODULUS, E1 AND E2
18.546 1.046
INPUT THE SHEAR MODULUS AND MAJOR POISSON'S RATIO
0.8546 0.1
INPUT MATERIAL DESCRIPTION FOR FASTENER
steel
INPUT YOUNG'S MODULUS AND POISSON'S RATIO FOR
THE FASTENER
30.046 0.3
INPUT THE DIAMETER OF THE FASTENER
0.3185
FASTENER TYPE
ENTER 1 FOR PROTRUDING HEAD
?
2 FOR COUNTERSINK HEAD
?
1
PLATE PROPERTIES ARE SPECIFIED BY
INPUTTING THE COORDINATES OF THE CORNER
VERTICES. NOTE: THE ORIGIN IS AT THE FASTENER
CENTER; INPUT COORDINATES ACCORDINGLY
U3 U2
MOLE
U4 U1
CENTROID
U4 U1
APPLIED LOAD CONVENTIONS:
FOR PLATE NO 1 (TOP) NORMAL LOADS ARE APPLIED
BETWEEN U1 AND U4
FOR PLATE NO 2 (BOTTOM) NORMAL LOADS ARE APPLIED
BETWEEN U1 AND U2
FOR PLATE NUMBER 1:
ENTER X,Y COORDINATES OF U 1
3.0 -0.9375
ENTER X,Y COORDINATES OF U 2
2.0 -0.9375
ENTER X,Y COORDINATES OF U 3
-3.0 -0.9375
ENTER X,Y COORDINATES OF U 4
-3.0 -0.9375
FOR PLATE NUMBER 2:
ENTER X,Y COORDINATES OF U 1
3.0 -0.9375
ENTER X,Y COORDINATES OF U 2
3.0 -0.9375
ENTER X,Y COORDINATES OF U 3
-3.0 -0.9375
ENTER X,Y COORDINATES OF U 4
-3.0 -0.9375
SELECT FAILURE CRITERIA:
ENTER 1 FOR POINT STRESS CRITERION
?
ENTER 2 FOR MAXIMUM STRESS CRITERION
?
2
ADVANCE STRESS CRITERION
NO IS THE CHARACTERISTIC DISTANCE OVER WHICH
STRESSES ARE AVERAGED AND COMPARED WITH SPECIFIED
STRESSING TO PREDICT FAILURE
ENTER AN UNLESS OTHERWISE NOTED TO THE THREE
PLY FAILURE MODES IN PLATE NO 1
A001 = NET SECTION
A004 = BEARING
A050 = SHEAR OUT
INPUT A001, A004, A050
0.0 0.4 0.5

```

Figure 36. Sample SASCJ Input
(Cont Inued) .

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

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

24
ENTER NO VALUES CORRESPONDING TO THE THREE
PLY FAILURE MODES IN PLATE NO 2

MODE - NET SECTION
MODE - BEARING
MODE - SHEAR OUT
INPUT MODE, MODE, MODE
?

0.1 0.025 0.00

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

INPUT FST,FSC,FSO
?

250.000 300.000 200.000
FOR PLATE NO 2 ENTER FIBER ULTIMATE
STRAIN VALUES

EPSILON ULT IN COMPRESSION
EPSILON ULT IN TENSION
GAMMA ULT IN SHEAR
(UNITS: IN/IN)
?

0.014 0.011 0.012
SASCJ ASSURES A BILINEAR PLY 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 $E2 = \alpha \cdot E1$.

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 $P1(ULT) = BETA1(INITIAL)$
INPUT BETA1 FOR NET SECTION ULTIMATE
BETA2 FOR BEARING ULTIMATE
BETA3 FOR SHEAROUT ULTIMATE
?

1.02 1.5 1.12
SASCJ ASSURES A BILINEAR PLY 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 $E2 = \alpha \cdot E1$.

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 $P1(ULT) = BETA1(INITIAL)$

Figure 36. Sample SASCJ Input
(Concluded).

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

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

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

3.3 SASCJ Output Description

For the sample problem defined in Figure 36, SASCH 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, SASCH 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 SAMCH Analysis

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

A flow chart of SAMCH operations is presented in Figure 38. As input, SAMCH 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 SAMCH to be subjected to uniaxial tensile or compressive loading, in a single or double shear configuration. Additional input requirements for the SAMCH 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

NET SECTION ULTIMATE (TENS) -0.2500+06 PSI
NET SECTION ULTIMATE (COMP) -0.3000+06 PSI
BEARING ULTIMATE -0.3000+06 PSI
SHEAROUT ULTIMATE -0.2000+06 PSI

CHARACTERISTIC DISTANCES

ADMT - 0.2000+02 INCHES
ROBR - 0.4000+00 INCHES
ROSO - 0.5000+02 INCHES

PLATE NUMBER 2

LAMINATE STRENGTH

NET SECTION ULTIMATE (TENS) -0.1270+06 PSI
NET SECTION ULTIMATE (COMP) -0.1500+06 PSI
BEARING ULTIMATE -0.1500+06 PSI
SHEAROUT ULTIMATE -0.1000+06 PSI

CHARACTERISTIC DISTANCES

ADMT - 0.1000+00 INCHES
ROBR - 0.2000+01 INCHES
ROSO - 0.3000+01 INCHES

GEOMETRY OF PLATE NO 1

COORDINATES OF CORNER VERTICES

-3.000, 0.000 3.000, 0.000

-3.000, -0.000 3.000, -0.000

FASTENER HOLE DIAMETER - 0.3120+00 INCHES

E/D RATIO - 0.0000+01

U/D RATIO - 0.0000+01

GEOMETRY OF PLATE NO 8

COORDINATES OF CORNER VERTICES

PROGRAM SASCI

A SINGLE LAP SHEAR JOINT WILL BE ANALYZED

WITH A PARTIALLY LOADED HOLE

STRESS RATIO - 0.0000+00

LOADED IN STATIC TENSION

PLATE NO 1

STEEL

T - 0.2500+00 INCHES

MATERIAL PROPERTIES

E1 -0.3000+08 PSI

E2 -0.3000+08 PSI

G12 -0.1150+08 PSI

MU12 -0.3000+00

MU21 -0.3000+00

PLATE NO 2

AS1/2501-5 (145/0/-45/012/0/00)S

T - 0.1200+00 INCHES

MATERIAL PROPERTIES

E1 -0.1850+08 PSI

E2 -0.1850+08 PSI

G12 -0.0500+08 PSI

MU12 -0.3000+00

MU21 -0.3000+00

FASTENER

STEEL

DIAMETER - 0.3120+00 INCHES

MATERIAL PROPERTIES

E -0.3000+08 PSI

MU -0.3000+00

FAILURE ANALYSIS

AN AVERAGE STRESS CRITERION WILL BE USED

PLATE NUMBER 1

LAMINATE STRENGTH

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

2025年12月31日

ALL INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED
DATE 08-09-2007 BY 60322 UCBAW

5035

- 3.000, 0.938 3.000, 0.938
 3.000, 0.938 3.000, -0.938
 WAXESTER HOLE DIAMETER = 0.3130-00 INCHES
 E. I. RATIO = 0.9500-01
 U D RATIO = 0.6000-01

!S!D!1!0!1!0!3!6!8!9! 3!0!0! 3!5!7!1!0!1

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

INCREMENT NO	JOINT LOAD	MODE	PLV TYPE	MODE
1	0.850E+04	32	0.0	DE CREE
2	0.875E+04	34	0.0	DE CREE
3	0.871D+04	36	0.0	DE CREE
4	0.871D+04	38	6.0	DE CREE
5	0.872D+04	39	0.0	DE CREE
6	0.872D+04	42	0.0	DE CREE
7	0.873D+04	43	0.0	DE CREE
8	0.873D+04	45	0.0	DE CREE
9	0.873D+04	47	0.0	DE CREE
10	0.874D+04	49	0.0	DE CREE
11	0.896D+04	32	0.0	DE CREE
12	0.817D+04	34	0.0	DE CREE
13	0.747D+04	36	0.0	DE CREE
14	0.676D+04	38	0.0	DE CREE
15	0.540D+04	39	0.0	DE CREE
16	0.534D+04	42	0.0	DE CREE
17	0.461E+04	43	0.0	DE CREE
18	0.383E+04	45	0.0	DE CREE
19	0.316D+04	47	0.0	DE CREE
20	0.244D+04	49	0.0	DE CREE
21	0.021D+04	31	45.000	DE CREE
22	0.031D+04	35	45.000	DE CREE
23	0.022D+04	46	45.000	DE CREE
24	0.025D+04	53	45.000	DE CREE
25	0.044D+04	48	-45.000	DE CREE
26	0.044D+04	37	-45.000	DE CREE
27	0.045D+04	44	-45.000	DE CREE

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

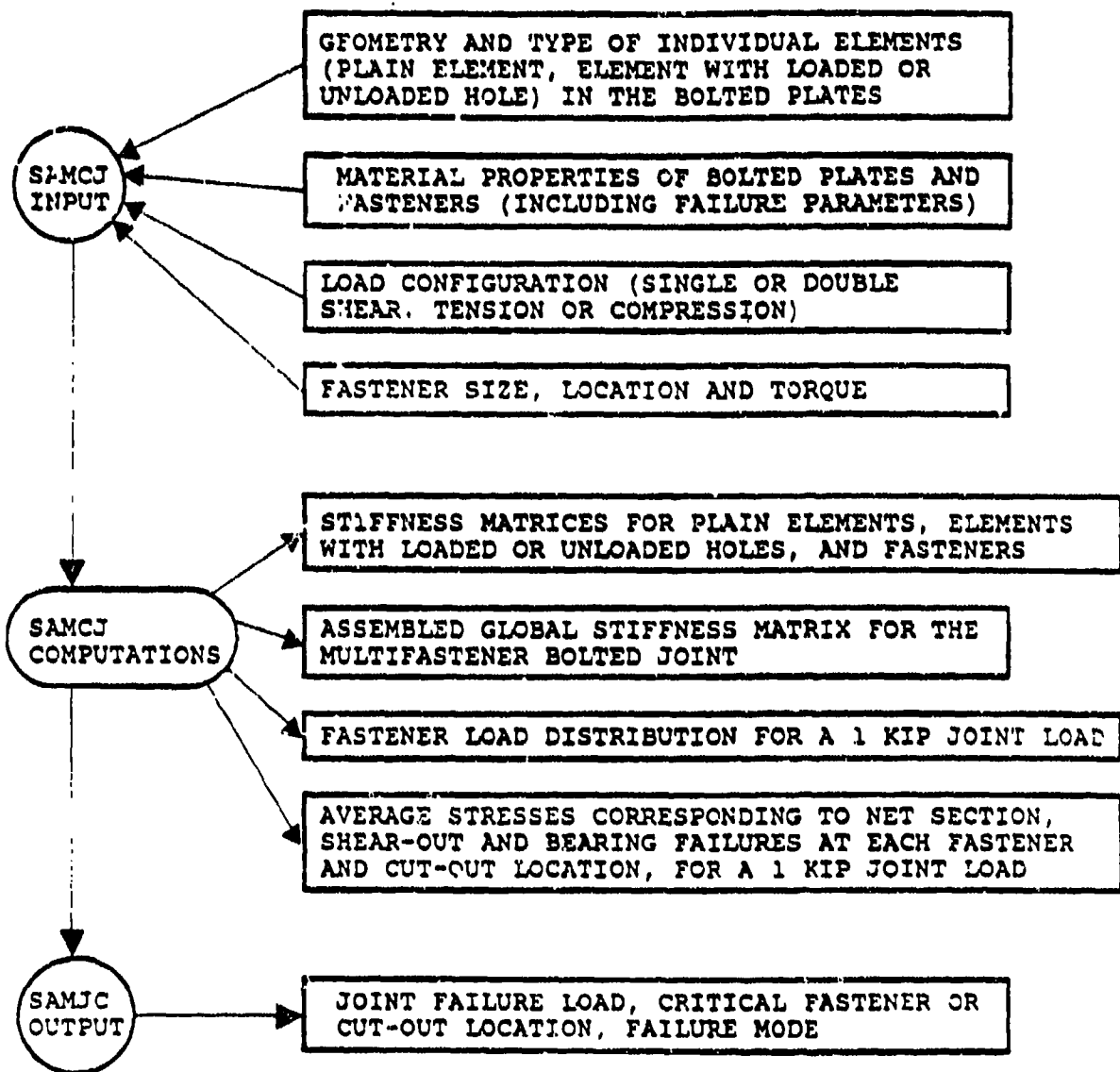


Figure 38. Flow Chart of SAMJC Operations.

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-out and bearing stresses at every fastener and cut-out location, corresponding to a 1 kip joint load.

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

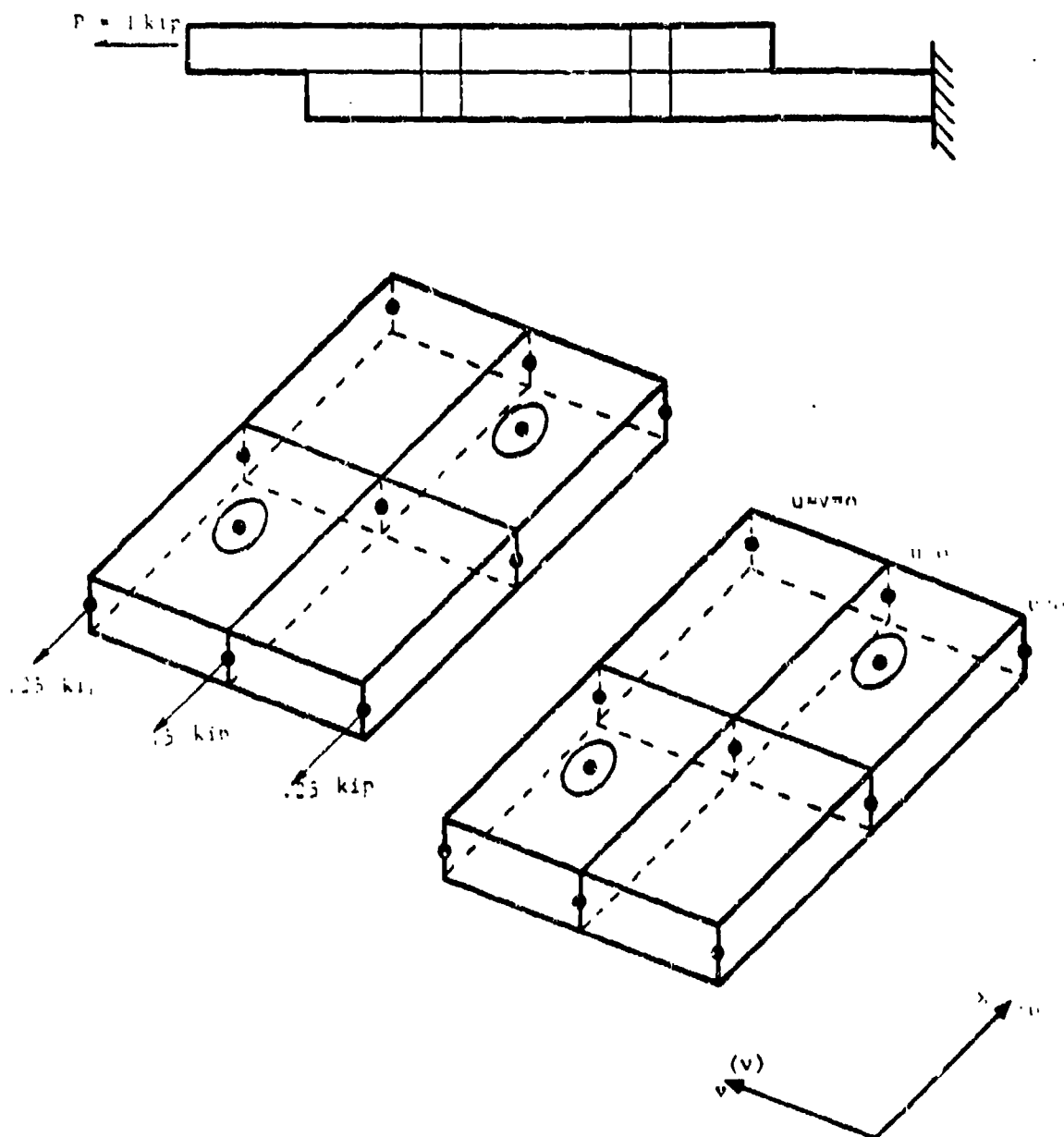


Figure 39. Application of Load and Displacement Boundary Conditions in the SAMCJ Code.

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.

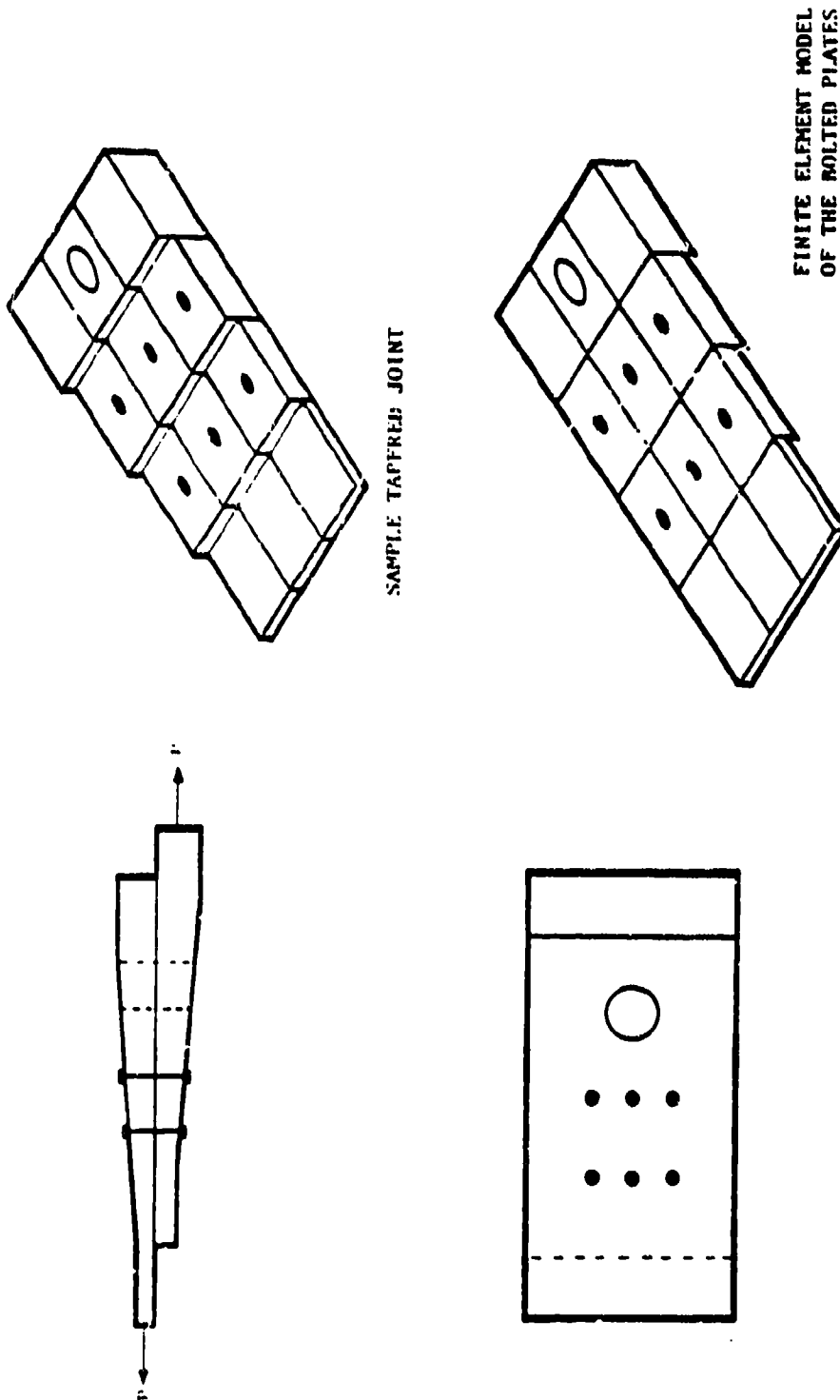
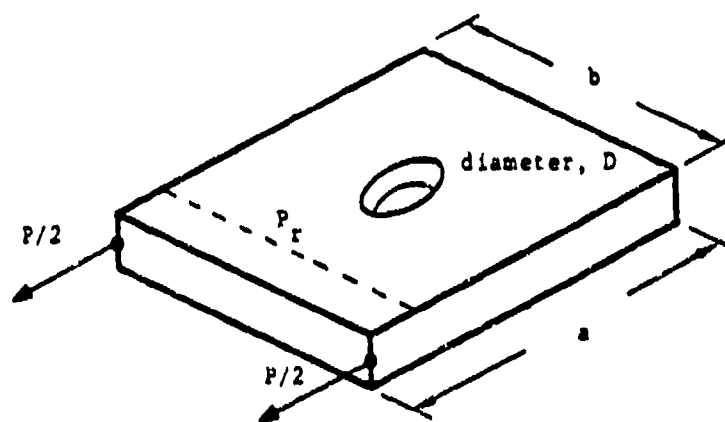


Figure 40. Finite Element Model of a Sample Tapered Bolted Joint.

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	1.74
3.2	3.2	1.76
6.4	3.2	1.37
16.0	3.2	1.16
1.6	6.4	-0.0995
3.2	6.4	0.989
6.4	6.4	1.23
16.0	6.4	1.16
3.2	16.0	-0.46
6.4	16.0	0.029
16.0	16.0	1.23

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

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

3.5 SAMCJ Input Description

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

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

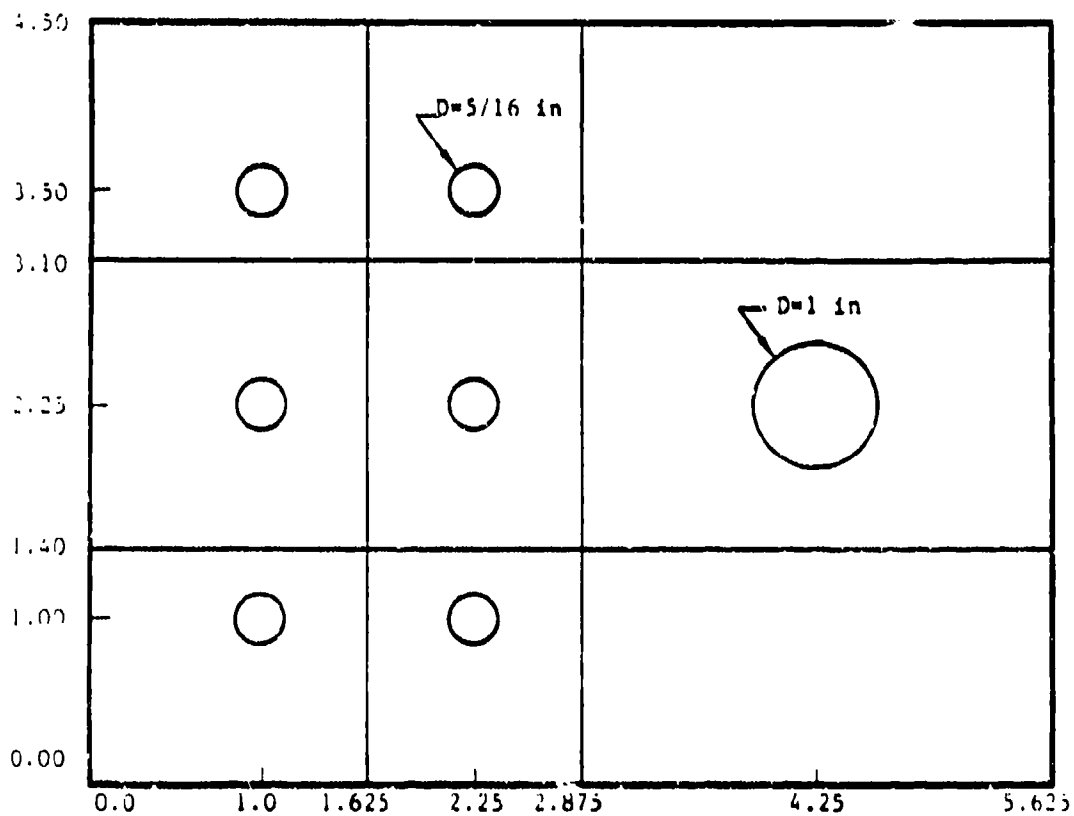
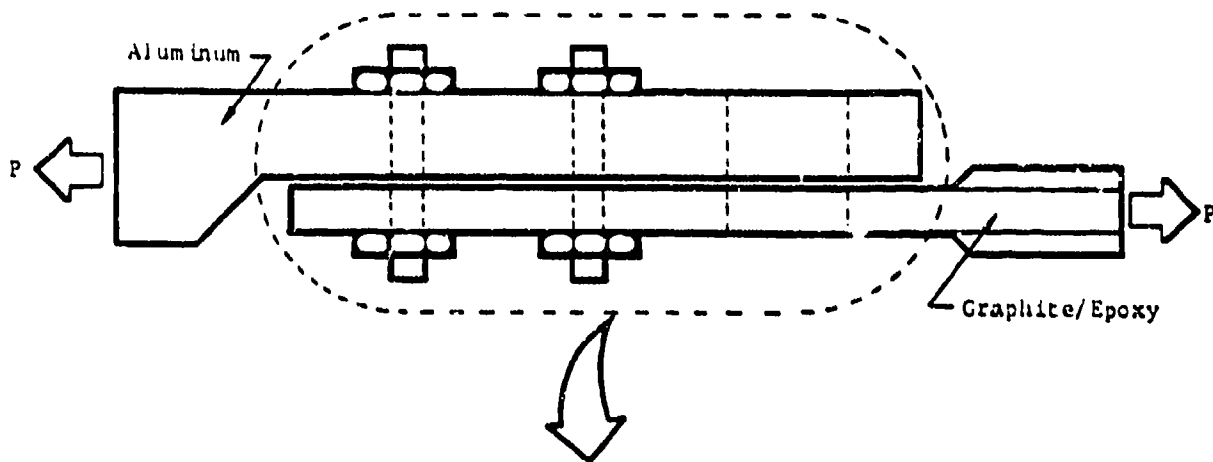


Figure 42. Nine Element Model of Each of the two Bolted Plates in the Sample Joint.

```

CALL NAME
1.50061 TEMPERATURE ASSUMED AS "TEMPERATURE"

PROGRAM NAME

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

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

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

1
ENTER:
1 FOR STATIC TENSION
2 FOR STATIC COMPRESSION

1
1
ENTER:
1 IS THE TOP PLATE A COMPOSITE OR A METAL?
ENTER C OR M IN THE FIRST FIELD
2
1
INPUT MATERIAL DESCRIPTION OF THIS PLATE
EX: ASA/3501-6
1
1
IS THE BOTTOM PLATE A COMPOSITE OR A METAL?
ENTER C OR M IN THE FIRST FIELD
2
1
INPUT MATERIAL DESCRIPTION OF THIS PLATE
EX: ASA/3501-6
EX: ASA/3501-6
EX: ASA/3501-6
1
1
NOTE: FOR COMPUTATIONAL PURPOSES A
METALLIC PLATE IS MODELED AS A 30 PLY
LAMINATE OF 0 DEGREE PLYS WITH ISOTROPIC
MATERIAL PROPERTIES

INPUT THE ENGINEERING PROPERTIES OF THE TOP PLATE
INPUT YOUNG'S MODULUS AND POISSON'S RATIO
1
1
10.0E6 0.3
INPUT THE ENGINEERING PROPERTIES OF THE BOTTOM PLATE
INPUT YOUNG'S MODULUS, E1 AND E2
1
1
10.0E6 1.0E6
INPUT THE SHEAR MODULUS AND MAJOR POISSON'S RATIO
1
1
10.0E6 0.3
INPUT TOTAL NUMBER OF DISTINCT PLY
ORIENTATIONS IN THE BOTTOM PLATE
1
1
INPUT ORIENTATION OF PLY TYPE NO 1
1
0.0

```

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

```

0.3125
FASTENER TYPE
ENTER:
1 1 FINE PROUDING HEAD
2 2 FINE COUNTERSUNK HEAD

GRID LAYOUTS
ENTER NUMBER OF GRIDS IN TOP PLATE
7
22

ENTER 22 GRID POINTS
FORMAT: GRID ID, X AND Y COORDINATES
7
101 0.0 0.0
7
102 0.0 1.40
7
103 0.0 3.1
7
104 0.0 4.5
7
105 1.0 1.0
7
106 1.0 2.25
7
107 1.0 3.5
7
108 1.025 0.0
7
109 1.025 1.4
7
110 1.025 3.1
7
111 1.025 4.5
7
112 2.25 1.0
7
113 2.25 2.25
7
114 2.25 3.5
7
115 2.275 0.0
7
116 2.275 1.4
7
117 2.275 3.1
7
118 2.275 4.5
7
119 5.025 0.0
7
120 5.025 1.4

FASTENER DESCRIPTIONS
INPUT MATERIAL DESCRIPTION FOR FASTENER
Steel
INPUT YOUNG MODULUS AND POISSONS RATIO FOR
THE FASTENER
7
20.00E 0.3
INPUT THE DIAMETER OF THE FASTENER
7

```

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

```

7 121 5.625 3.1
7 122 5.625 4.5
7 ENTER NUMBER OF CPIDS IN BOTTOM PLATE
82
7 ENTER 22 GRID POINTS
7 FORMAT: GRID ID, X AND Y COORDINATES
7 201 0.0 0.0
7 202 0.0 1.4
7 203 0.0 3.1
7 204 0.0 4.5
7 205 1.0 1.0
7 206 1.0 2.25
7 207 1.0 3.5
7 208 1.625 0.0
7 209 1.625 1.4
7 210 1.625 3.1
7 211 1.625 4.5
7 212 2.25 1.0
7 213 2.25 2.25
7 214 2.25 3.5
7 215 2.875 0.0
7 216 2.875 1.4
7 217 2.875 3.1
7 218 2.875 4.5
7 219 5.625 0.0
7 220 5.625 1.4
7 221 5.625 3.1
7 222 5.625 4.5
7 ELEMENT DESCRIPTION:

7 M2 M3
7 M1 M4

ELEMENT TYPES ARE DESIGNATED AS FOLLOWS:
7 4 MODE PLAIN ELEMENT TYPE NO. 1
7 5 MODE LOADED HOLE ELEMENT TYPE NO. 2
7 4 MODE OPEN HOLE ELEMENT TYPE NO. 3

7 (NOTE: ENTER M6-8 FOR FOUR MODE ELEMENTS)
7 ENTER NUMBER OF ELEMENTS IN TOP PLATE
7 9
7 FOR ELEMENT NO 1 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 101 102 100 100 105 2
7 ENTER ELEMENT THICKNESS
7 0.5
7 FOR ELEMENT NO 2 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 102 102 103 110 100 105 2
7 ENTER ELEMENT THICKNESS
7 0.5
7 FOR ELEMENT NO 3 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 103 103 104 111 110 107 2
7 ENTER ELEMENT THICKNESS
7 0.5
7 FOR ELEMENT NO 4 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 104 104 109 115 115 110 2
7 ENTER ELEMENT THICKNESS
7 0.5
7 FOR ELEMENT NO 5 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 105 105 110 117 116 113 8
7 ENTER ELEMENT THICKNESS
7 0.5
7 FOR ELEMENT NO 6 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
7 106 110 111 118 117 114 8
7 ENTER ELEMENT THICKNESS
7 0.5
    
```

PLATE ELEMENTS ARE NUMBERED
 CLOCKWISE AS SHOWN

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

```

FOR ELEMENT NO 7 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
107 115 116 120 119 0 1
ENTER ELEMENT THICKNESS
?
0 8
FOR ELEMENT NO 8 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
108 115 117 121 120 0 3
ENTER ELEMENT THICKNESS, X AND Y COORDINATES
OF OPEN HOLE AND HOLE RADIUS
?
0.5 4.25 2.25 0.5
FOR ELEMENT NO 9 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
109 117 118 122 121 0 1
ENTER ELEMENT THICKNESS
?
0.5
ENTER NUMBER OF ELEMENTS IN BOTTOM PLATE
?
9
FOR ELEMENT NO 10 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
201 201 208 209 208 205 2
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 11 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
202 202 203 210 209 206 2
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 12 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
203 203 204 211 210 207 2
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 13 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
204 204 209 216 215 212 2
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 14 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
205 209 210 217 219 213 2
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 15 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
206 210 211 218 217 214 2
ENTER ELEMENT LAYOUT NO
?
1

FOR ELEMENT NO 16 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
207 215 216 220 219 0 1
ENTER ELEMENT LAYOUT NO
?
1
FOR ELEMENT NO 17 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
208 216 217 221 220 0 3
ENTER ELEMENT LAYOUT NUMBER, X AND Y
COORDINATES OF THE OPEN HOLE AND THE HOLE
RADIUS
?
1 4.25 2.25 0.5
FOR ELEMENT NO 18 ENTER: ELEMENT ID,M1,M2,M3,M4,M5,ELEMENT TYPE
209 217 218 222 221 0 1
ENTER ELEMENT LAYOUT NO
?
1
FASTENERS ARE MODELED BY EFFECTIVE
FASTER ELEMENTS WHICH PROVIDE THE
ELASTIC LINK BETWEEN THE TOP AND
BOTTOM PLATES
ENTER NUMBER OF FASTENERS IN JOINT
?
6

EFFECTIVE FASTENER ELEMENTS ARE
NUMBERED AS SHOWN: M1 (TOP PLATE)
M2 (BOTTOM PLATE)

WHERE M1 AND M2 CORRESPOND TO THE CENTRAL
HOLES IN LOADED HOLE ELEMENTS

FORMAT: ELEMENT ID, M1, M2
ENTER ELEMENT NO 1
?
101 105 205
ENTER ELEMENT NO 2
?
102 105 205
ENTER ELEMENT NO 3
?
103 107 207
ENTER ELEMENT NO 4
?
104 112 218
ENTER ELEMENT NO 5
?
105 113 213
ENTER ELEMENT NO 6
?
106 114 214
  
```

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

TO REDUCE RUN TIMES, ELEMENTS MAY BE
 GROUPED INTO SETS WHICH WILL BE ASSIGNED
 IDENTICAL STIFFNESS MATRICES.
 ENTER: 1 TO USE THIS OPTION
 2 OTHERWISE

1
 2
 3

FOR THE TOP PLATE INPUT NUMBER OF GROUPS
 FOR THE EFFECTIVE FASTENER, LOADED HOLE, UNLOADED
 HOLE AND PLAIN ELEMENT
 (INPUT 0 IF ELEMENT TYPE IS NOT USED)

1 6 1 1
 ENTER NUMBER OF EFFECTIVE FASTENER ELEMENTS
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1

5
 ENTER 6 ELEMENT IDS
 101 102 103 104 105 106

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

1
 INPUT 1 ELEMENT IDS
 101
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 2

1
 INPUT 1 ELEMENT IDS
 102
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 3

1
 INPUT 1 ELEMENT IDS
 103
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 4

1
 INPUT 1 ELEMENT IDS
 104
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 5

1
 INPUT 1 ELEMENT IDS
 105
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 6

1
 INPUT 1 ELEMENT IDS
 106
 GROUPING OF UNLOADED HOLE ELEMENTS
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER :

1
 ENTER 1 ELEMENT IDS
 107 108

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

7
 6 1 1

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

1
 INPUT 1 ELEMENT IDS
 201
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 2

1
 INPUT 1 ELEMENT IDS
 202
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 3

1
 INPUT 1 ELEMENT IDS
 203
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 4

1
 INPUT 2 ELEMENT IDS
 204
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 5

1
 INPUT 1 ELEMENT IDS
 205
 ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 6

1
 INPUT 1 ELEMENT IDS
 206

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

```

GROUPING OF UNLOADED NOTCH ELEMENTS
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1
1
ENTER 1 ELEMENT IDS
200
GROUPING OF PLAIN ELEMENTS:
ENTER NUMBER OF ELEMENTS IN GROUP NUMBER 1
1
ENTER 2 ELEMENT IDS
207 209
INPUT DATA FOR FAILURE ANALYSIS:
ENTER METALLIC STRENGTHS:
TENSILE STRENGTH
COMPRESSIVE STRENGTH
SHEAR STRENGTH
7
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 AVERAGED AND COMPARED TO UNNOTCHED
LAPINATE STRENGTHS TO PREDICT FAILURE
ENTER NO VALUES FOR STRESS AVERAGING
FOR EACH FAILURE MODE IN PLATE NO 1
1
MODE - NET SECTION
MODE - BEARING
MODE - SHEAROUT
7
0.5 0.5 0.5
ENTER FINEST ULTIMATE STRAIN VALUES
IN PLATE NO 2
EPSILON ULT IN COMPRESSION
EPSILON ULT IN TENSION
EPSILON ULT IN SHEAR
7
0.0175 0.012 0.012
ON AVERAGE STRESS CRITERIA IS USED TO
PREDICT FAILURE. NO VALUES ARE REQUIRED AS
CHARACTERISTIC DISTANCES OVER WHICH STRESSES
ARE TO BE AVERAGED AND COMPARED TO UNNOTCHED
LAPINATE STRENGTHS TO PREDICT FAILURE
ENTER NO VALUES FOR STRESS AVERAGING
FOR EACH FAILURE MODE IN PLATE NO 2
MODE - NET SECTION
MODE - BEARING
MODE - SHEAROUT
7
0.1 0.05 0.15
  
```

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

protruding head) are input.

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

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

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

The last four lines of input introduce the failure parameters for the materials in the two plates. For metallic plates, the tensile, compressive and shear strengths (250.0D3 each), and the averaging distances for the net section, bearing and shear-out modes of failure (0.5 each) are input. Since the joints were designed to fail the laminated plates, and SAMCJ was developed primarily for the prediction of the strength of bolted laminates, the failure parameters for the metallic plates were input to be 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 SINGLE LAP SHEAR PANEL WILL BE ANALYZED
LOADED IN STATIC TENSION
  PLATE NO 1
ALUMINUM
MATERIAL PROPERTIES
  E1 -0.1000+00 PSI
  E2 -0.1000+00 PSI
  G12 -0.1079+00 PSI
  MU12 -0.3000+00
  MU21 -0.3000+00
  PLATE NO 2
A51 3041-C
MATERIAL PROPERTIES
  E1 -0.1000+00 PSI
  E2 -0.1000+07 PSI
  G12 -0.1079+00 PSI
  MU12 -0.3000+00
  MU21 -0.3000+00
FASTENER DESCRIPTION
STEEL
DIAMETER - 0.3120+00 INCHES
MATERIAL PROPERTIES
  E -0.3000+00 PSI
  RU -0.3000+00
FAILURE ANALYSIS
AN ALUMINUM SHEAR CRITERION WILL BE USED
PLATE NUMBER 1
METALLIC STRENGTH
  TENSILE STRENGTH - 0.8000+00
  COMPRESSIVE STRENGTH - 0.8000+00
  SHEAR STRENGTH - 0.8000+00
CHARACTERISTIC DISTANCES
  AGENT - 0.5000+00 INCHES
  AGSM - 0.5000+00 INCHES
  AGSC - 0.5000+00 INCHES
  PLATE NUMBER 8
  FIBER STRAIN ULTIMATES
  EPSILON ULT COMP - 0.1750+01
  EPSILON ULT TEN - 0.1820+01
  GAMMA ULT SHEAR - 0.1200+01
CHARACTERISTIC DISTANCES
  AGENT - 0.1000+00 INCHES
  AGSM - 0.2500+00 INCHES
  AGSC - 0.2500+00 INCHES
PAUSE FOR STIFFNESS MATRIX CALCULATIONS
  ELEMENT FORCES
  (P APPLIED - 10.000 LBS)
  ELEMENT ID 101
    GRID
    101
    102
    103
    104
    105
    FX
    FY
  ELEMENT ID 102
    GRID
    106
    107
    108
    109
    110
    111
    112
    113
    114
    115
    116
    117
    118
    119
    120
    121
    122
    123
    124
    125
    126
    127
    128
    129
    130
    131
    132
    133
    134
    135
    136
    137
    138
    139
    140
    141
    142
    143
    144
    145
    146
    147
    148
    149
    150
    151
    152
    153
    154
    155
    156
    157
    158
    159
    160
    161
    162
    163
    164
    165
    166
    167
    168
    169
    170
    171
    172
    173
    174
    175
    176
    177
    178
    179
    180
    181
    182
    183
    184
    185
    186
    187
    188
    189
    190
    191
    192
    193
    194
    195
    196
    197
    198
    199
    200
    201
    202
    203
    204
    205
    206
    207
    208
    209
    210
    211
    212
    213
    214
    215
    216
    217
    218
    219
    220
    221
    222
    223
    224
    225
    226
    227
    228
    229
    230
    231
    232
    233
    234
    235
    236
    237
    238
    239
    240
    241
    242
    243
    244
    245
    246
    247
    248
    249
    250
    251
    252
    253
    254
    255
    256
    257
    258
    259
    260
    261
    262
    263
    264
    265
    266
    267
    268
    269
    270
    271
    272
    273
    274
    275
    276
    277
    278
    279
    280
    281
    282
    283
    284
    285
    286
    287
    288
    289
    290
    291
    292
    293
    294
    295
    296
    297
    298
    299
    300
    301
    302
    303
    304
    305
    306
    307
    308
    309
    310
    311
    312
    313
    314
    315
    316
    317
    318
    319
    320
    321
    322
    323
    324
    325
    326
    327
    328
    329
    330
    331
    332
    333
    334
    335
    336
    337
    338
    339
    340
    341
    342
    343
    344
    345
    346
    347
    348
    349
    350
    351
    352
    353
    354
    355
    356
    357
    358
    359
    360
    361
    362
    363
    364
    365
    366
    367
    368
    369
    370
    371
    372
    373
    374
    375
    376
    377
    378
    379
    380
    381
    382
    383
    384
    385
    386
    387
    388
    389
    390
    391
    392
    393
    394
    395
    396
    397
    398
    399
    400
    401
    402
    403
    404
    405
    406
    407
    408
    409
    410
    411
    412
    413
    414
    415
    416
    417
    418
    419
    420
    421
    422
    423
    424
    425
    426
    427
    428
    429
    430
    431
    432
    433
    434
    435
    436
    437
    438
    439
    440
    441
    442
    443
    444
    445
    446
    447
    448
    449
    450
    451
    452
    453
    454
    455
    456
    457
    458
    459
    460
    461
    462
    463
    464
    465
    466
    467
    468
    469
    470
    471
    472
    473
    474
    475
    476
    477
    478
    479
    480
    481
    482
    483
    484
    485
    486
    487
    488
    489
    490
    491
    492
    493
    494
    495
    496
    497
    498
    499
    500
    501
    502
    503
    504
    505
    506
    507
    508
    509
    510
    511
    512
    513
    514
    515
    516
    517
    518
    519
    520
    521
    522
    523
    524
    525
    526
    527
    528
    529
    530
    531
    532
    533
    534
    535
    536
    537
    538
    539
    540
    541
    542
    543
    544
    545
    546
    547
    548
    549
    550
    551
    552
    553
    554
    555
    556
    557
    558
    559
    560
    561
    562
    563
    564
    565
    566
    567
    568
    569
    570
    571
    572
    573
    574
    575
    576
    577
    578
    579
    580
    581
    582
    583
    584
    585
    586
    587
    588
    589
    590
    591
    592
    593
    594
    595
    596
    597
    598
    599
    600
    601
    602
    603
    604
    605
    606
    607
    608
    609
    610
    611
    612
    613
    614
    615
    616
    617
    618
    619
    620
    621
    622
    623
    624
    625
    626
    627
    628
    629
    630
    631
    632
    633
    634
    635
    636
    637
    638
    639
    640
    641
    642
    643
    644
    645
    646
    647
    648
    649
    650
    651
    652
    653
    654
    655
    656
    657
    658
    659
    660
    661
    662
    663
    664
    665
    666
    667
    668
    669
    670
    671
    672
    673
    674
    675
    676
    677
    678
    679
    680
    681
    682
    683
    684
    685
    686
    687
    688
    689
    690
    691
    692
    693
    694
    695
    696
    697
    698
    699
    700
    701
    702
    703
    704
    705
    706
    707
    708
    709
    710
    711
    712
    713
    714
    715
    716
    717
    718
    719
    720
    721
    722
    723
    724
    725
    726
    727
    728
    729
    730
    731
    732
    733
    734
    735
    736
    737
    738
    739
    740
    741
    742
    743
    744
    745
    746
    747
    748
    749
    750
    751
    752
    753
    754
    755
    756
    757
    758
    759
    760
    761
    762
    763
    764
    765
    766
    767
    768
    769
    770
    771
    772
    773
    774
    775
    776
    777
    778
    779
    780
    781
    782
    783
    784
    785
    786
    787
    788
    789
    790
    791
    792
    793
    794
    795
    796
    797
    798
    799
    800
    801
    802
    803
    804
    805
    806
    807
    808
    809
    810
    811
    812
    813
    814
    815
    816
    817
    818
    819
    820
    821
    822
    823
    824
    825
    826
    827
    828
    829
    830
    831
    832
    833
    834
    835
    836
    837
    838
    839
    840
    841
    842
    843
    844
    845
    846
    847
    848
    849
    850
    851
    852
    853
    854
    855
    856
    857
    858
    859
    860
    861
    862
    863
    864
    865
    866
    867
    868
    869
    870
    871
    872
    873
    874
    875
    876
    877
    878
    879
    880
    881
    882
    883
    884
    885
    886
    887
    888
    889
    890
    891
    892
    893
    894
    895
    896
    897
    898
    899
    900
    901
    902
    903
    904
    905
    906
    907
    908
    909
    910
    911
    912
    913
    914
    915
    916
    917
    918
    919
    920
    921
    922
    923
    924
    925
    926
    927
    928
    929
    930
    931
    932
    933
    934
    935
    936
    937
    938
    939
    940
    941
    942
    943
    944
    945
    946
    947
    948
    949
    950
    951
    952
    953
    954
    955
    956
    957
    958
    959
    960
    961
    962
    963
    964
    965
    966
    967
    968
    969
    970
    971
    972
    973
    974
    975
    976
    977
    978
    979
    980
    981
    982
    983
    984
    985
    986
    987
    988
    989
    990
    991
    992
    993
    994
    995
    996
    997
    998
    999
    1000
  
```

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

Copy of this document does not
 permit fully legible reproduction

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

ELEMENT ID	207			
CRIP	FX	FV		
215	1200+03	0.2560+02		
216	-1630+03	-1080+01		
217	0.1260+03	-1750+02		
219	0.1630+03	-1390+01		
ELEMENT ID	208			
CRIP	FX	FV		
216	-2120+03	0.5850+02		
217	-2110+03	-5910+02		
221	0.2120+03	-1690+02		
220	0.2110+03	0.1750+02		
ELEMENT ID	209			
CRIP	FX	FV		
217	-1590+03	0.8410+01		
218	-1290+03	-2530+02		
222	0.1620+03	-1780+03		
221	0.1260+03	0.1690+02		

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

ELEMENT	NS	SO	BR
101	0.5750+06	0.0830+06	0.4590+06
102	0.5460+06	0.7000+06	0.5010+06
103	0.4170+06	0.6260+06	0.4920+06
104	0.2920+07	0.1510+07	0.7300+06
105	0.3110+07	0.1490+07	0.8370+06
106	0.2030+07	0.1710+07	0.7370+06
108	0.1480+08	0.8900+06	0.8270+06
201	0.3400+06	0.1230+06	0.1200+06
202	0.2950+06	0.9700+06	0.1950+06
203	0.5130+06	0.1200+06	0.1320+06
204	0.7040+05	0.6440+05	0.7700+05
205	0.5290+05	0.4110+05	0.5770+05
206	0.8080+05	0.5550+05	0.8130+05
208	0.3730+05	0.1740+06	0.3020+07

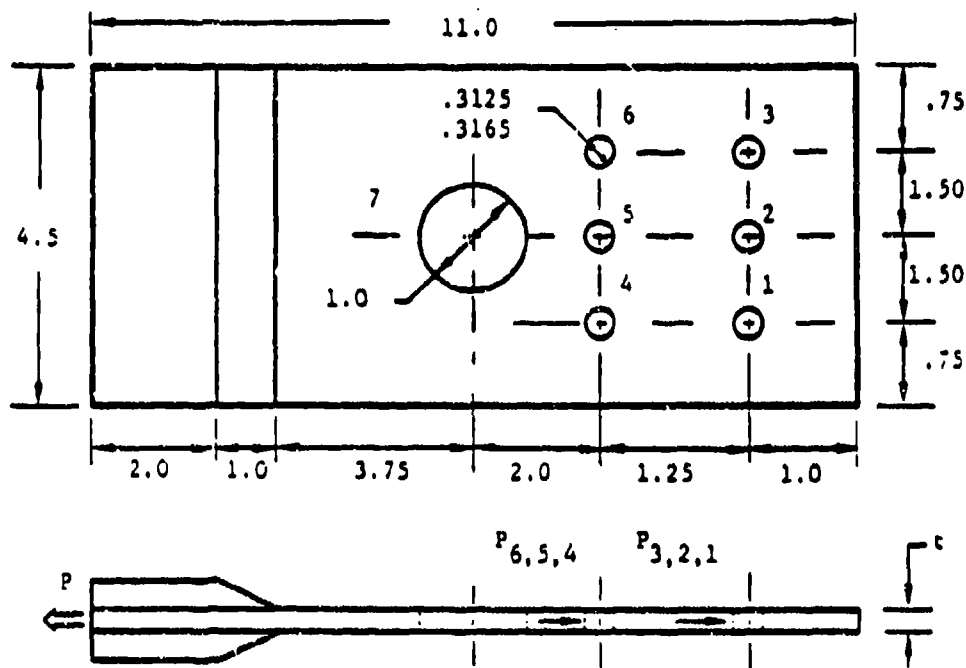
FAILURE IS PREDICTED TO OCCUR IN ELEMENT NUMBER 208 AT AN APPLIED JOINT LOAD VALUE OF 0.3720000+05 LBS

THE PREDICTED FAILURE MODE IS NET SECTION

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

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

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



	SAMCJ PREDICTION	TEST RESULTS (Ref. 2)
P_1/P	0.165	0.162
P_2/P	0.191	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 (5)	7 and 4, 5, 6
FAILURE MODE(S)	NET SECTION (NET SECTION)	NET SECTION

* Next possible failure mode and location at a higher load level
 Figure 45. Comparison of SAMCJ Predictions for the Sample Problem with
 Test Results from Reference 8.

SECTION 4

DESIGN VERIFICATION OF A BOLTED STRUCTURAL ELEMENT

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

4.1 Description of the Bolted Structural Element

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

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

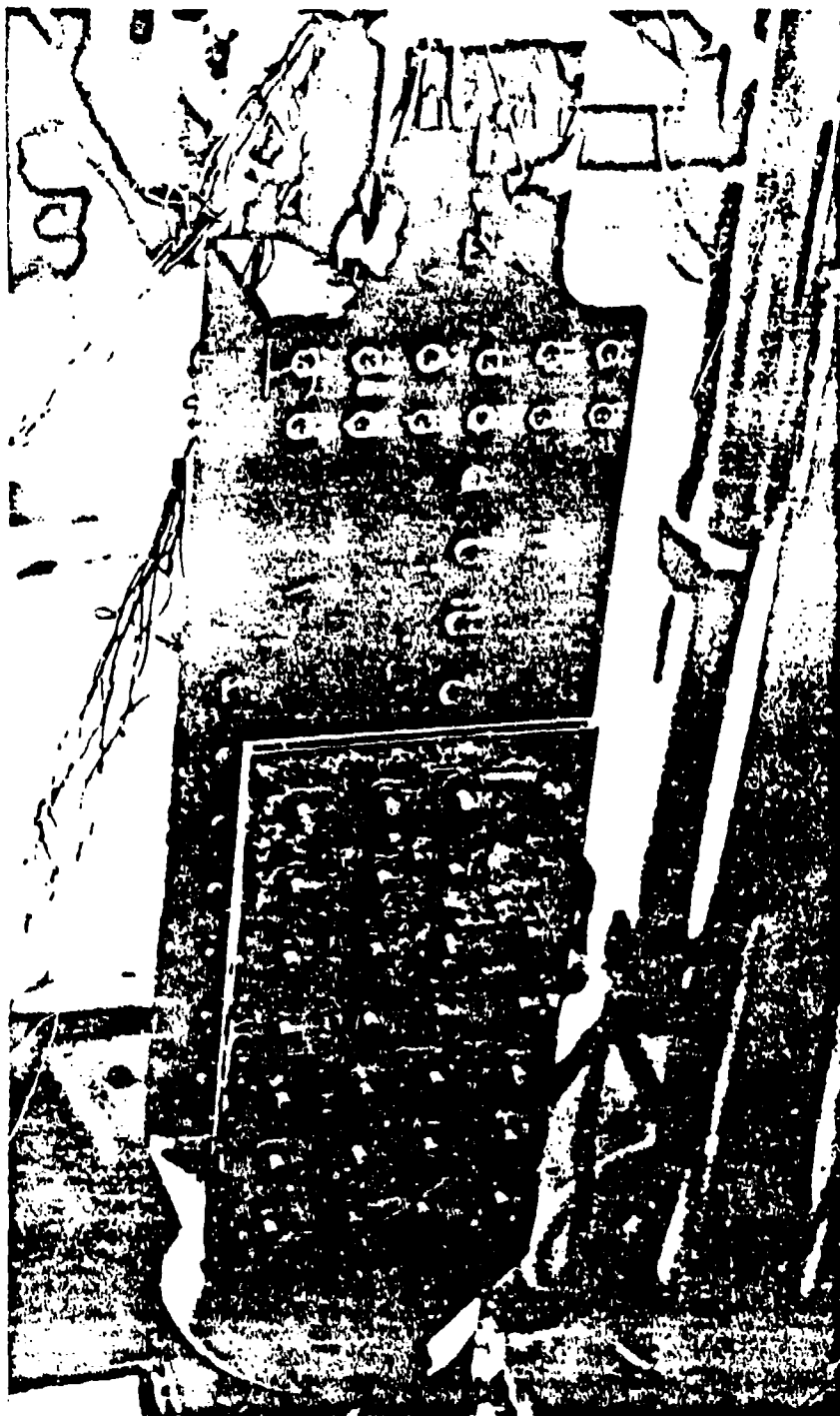


Figure 46. Photograph of an Assembled Test Element.

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

4.2 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 47). The observed failure modes were significantly influenced by the tilting or "digging in" of the countersunk fasteners - a phenomenon that cannot be accounted for by the fastener analysis in the SAMCJ computer code.

4.3 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 $[0_{28}/\pm 45_{12}/90_7]_C$ 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 $[0_{28}/\pm 45_{12}/90_6]_C$ layup, and the bottom region is assumed to be a $[0_{26}/\pm 45_{10}/90_6]_C$ 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

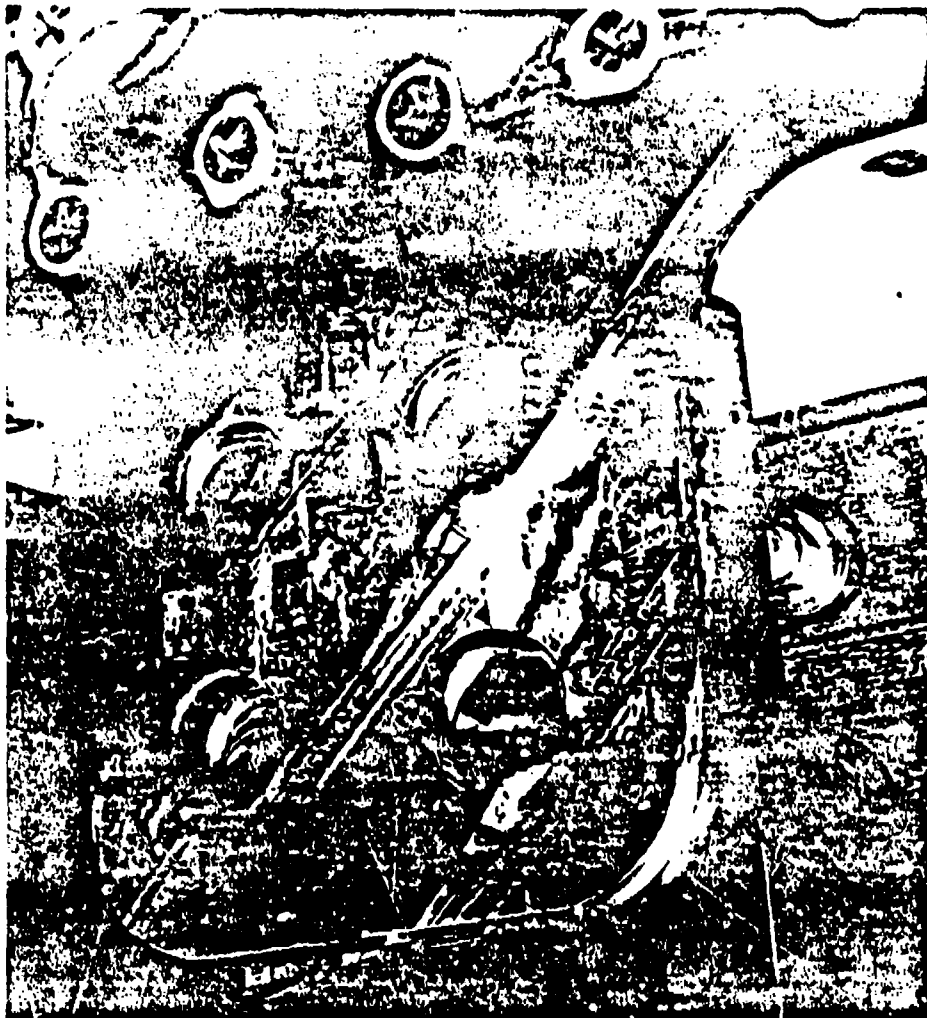


Figure 47. Photograph of the Tab Region of the Failed Element.

Fuselage
Attachment
Frame

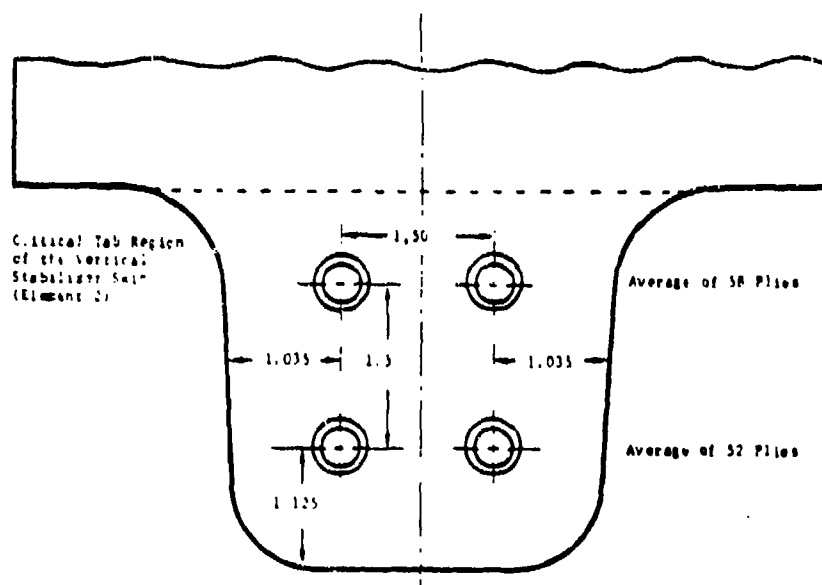
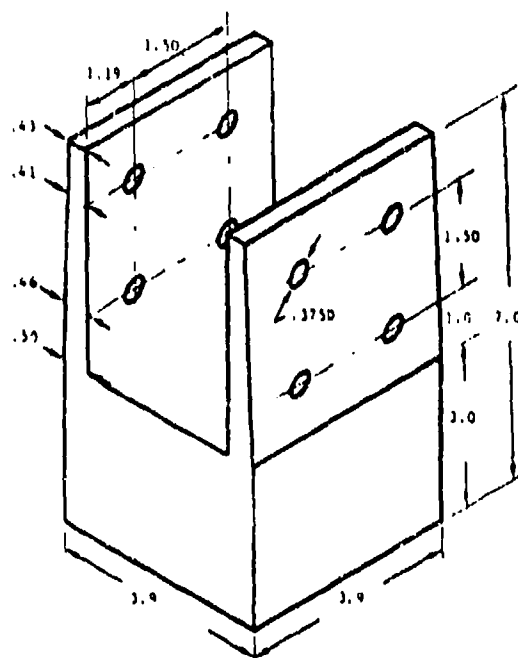


Figure 48. Dimensions of the Critical Skin Tab and the Fuselage Attachment Frame.

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

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

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

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

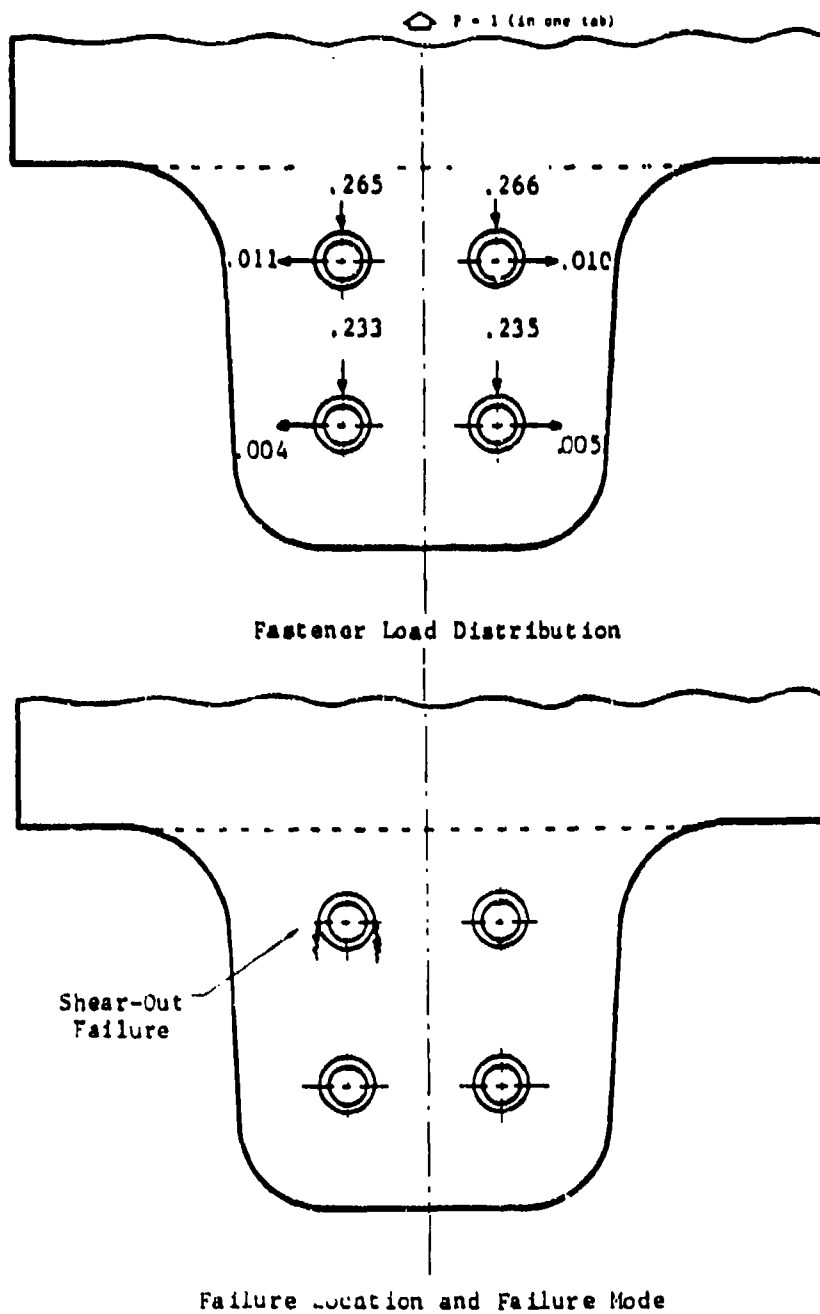


Figure 49. Load Distribution Among Fasteners, Failure Location and Failure Mode in the Graphite/Epoxy Tabs.

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.

Tab 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.

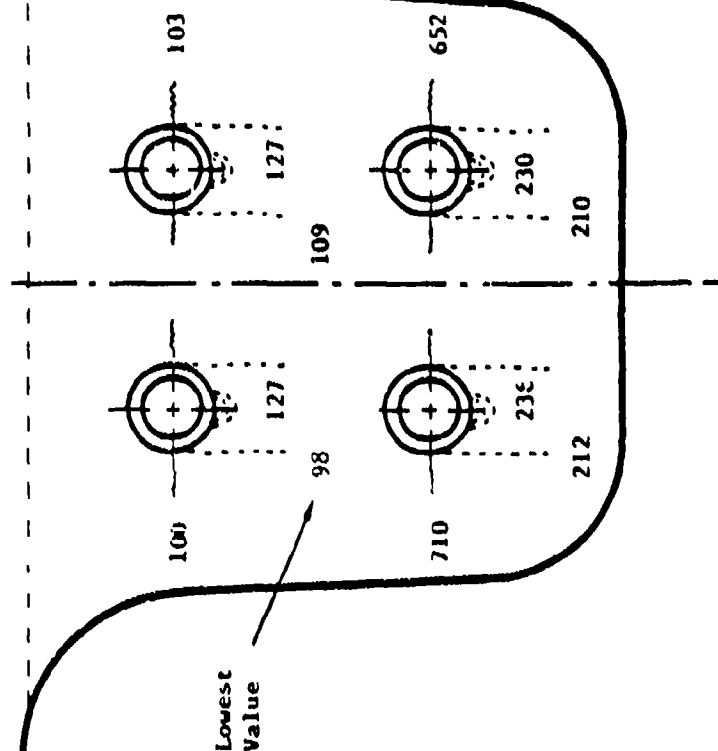


Figure 50. Analytically Predicted Element Load Levels to Precipitate Net Section, Bearing and Shear-Out Modes of Tab Failure at Each Fastener Location.

SECTION 5

CONCLUSIONS

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

REFERENCES

1. Buchanan, D. L. and Garbo, S. P., "Design of Highly Loaded Composite Joints and Attachment for Wing Structures," NADC-81194-60, August 1981.
2. Watts, D. J., et al., "Development of Composites Technology for Joints and Cutouts in Fuselage Structure of Large Transport Aircraft," Semi-Annual Technical Report Number 1, ACEE-34-PR-3419, October 1984.
3. Kong, S. J., "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. ACEF 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 Batah, E. J., "Special Fastener Development for Composite Structure," AFWAL-TR-82-3049, July 1982.
12. Wall, F. T., "Fasteners for Use with Composite Materials," NASC Report Number 78-01, Aerospace Industries Association of America, Inc, January 1978.
13. Garbo, S. P. and Ogonowski, J. M., "Effect of Variances and Manufacturing Tolerances on the Design Strength and Life of Mechanically Fastened Joints," Volume 1, AFWAL-TR-81-3041, April 1981.
14. Ramkumar, R. L. and Tossavainen, E., "Bolted Joints in Composite Structures: Design, Analysis and Verification; Task I Test Results," AFWAL-TR-84-3047, August 1984.
15. Eisenmann, J. R. and Leonhardt, J. L., "Improving Composite Bolted Joint Efficiency by Laminate Tailoring," Joining of Composite Materials, ASTM STP 749, edited by K. T. Kedward, American Society for Testing and Materials, 1981, pp. 117-130.

APPENDIX A

SASCJ Program Listing

၇၇၇

```

WRITE(6,6,6)
876 FORMAT(///,10X,' PROGRAM SASCH',///,
  *' PROGRAM SASCH PREDICTS FAILURE LOADS OF ',///,
  *' MECHANICALLY FASTENED, COMPOSITE LAMINATE, ',///,
  *' SINGLE OR DOUBLE LAP SHEAR JOINTS. ',///,
  *' PROGRAM ASSUMES THAT INPUT PARAMETERS ARE ',///,
  *' IN ENGLISH UNITS - LENGTHS ARE INPUT ',///,
  *' IN INCHES AND MODULI AND STRENGTHS ARE ',///,
  *' EXPRESSED IN PSI ',///,
  *' )
WRITE(6,401)
401 FORMAT(' ENTER BYPASS RATIO ALPHA: ',///,
  *' ALPHA=0 FOR FULL BEARING ',///,
  *' ALPHA=1 FOR OPEN HOLE ',///,
  *' 0<ALPHA<1 FOR GENERAL BYPASS ')
READ(5,*) BPR
WRITE(6,911)
911 FORMAT(' ENTER: ',///,
  *' 1 FOR STATIC TENSION ',///,
  *' 2 FOR STATIC COMPRESSION ',///,
  *' )
READ(5,*) LTNCM
NLIM=1
IF(BPR.EQ.1.0) GO TO 380
NLIM=2
WRITE(6,400)
400 FORMAT(' ENTER: ',///,
  *' 1 FOR SLS (SINGLE LAP SHEAR)',///,
  *' 2 FOR DLS (DOUBLE LAP SHEAR)',///,
  *' )
READ(5,*) NSDLS
106 FORMAT(A1)
380 CONTINUE
DO 300 K=1,NLIM
  IF(K.EQ.1) WRITE(6,511)
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,1),I=1,15)
204 FORMAT(15A4)
300 CONTINUE
  IF(CM(1).NE.CMC.OR.CM(2).NE.CMC) WRITE(6,721)
721 FORMAT(//,' NOTE: FOR COMPUTATIONAL PURPOSES A ',///,
  *' METALLIC PLATE IS MODELED AS A 30 PLY ',///,
  *' LAMINATE OF 0 DEGREE PLIES WITH ISOTROPIC ',///,
  *' MATERIAL PROPERTIES ',///,
  *' )
  IF(BPR.NE.1.0) WRITE(6,494)
  IF(BPR.EQ.1.0) WRITE(6,495)
494 FORMAT(' NOTE: NUMERICAL DESIGNATIONS FOR THE ',///,
  *' PLATES ARE: ',///,
  *' TOP PLATE = NO 1 ',///,
  *' BOTTOM PLATE = NO 2 ',///,
  *' )
495 FORMAT(' NOTE: A SINGLE PLATE WITH AN OPEN ',///,
  *' HOLE IS DESIGNATED AS PLATE NUMBER 1 ',///,
  *' )
  DO 301 K=1,NLIM
  IF(CM(K).EQ.CMC) GO TO 15
  NPLY(K)=30

```

```

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
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500

```

```

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

```

00001160
 00001170
 00001180
 00001190
 00001200
 00001210
 00001220
 00001230
 00001240
 00001250
 00001260
 00001270
 00001280
 00001290
 00001300
 00001310
 00001320
 00001330
 00001340
 00001350
 00001360
 00001370
 00001380
 00001390
 00001400
 00001410
 00001420
 00001430
 00001440
 00001450
 00001460
 00001470
 00001480
 00001490
 00001500
 00001510
 00001520
 00001530
 00001540
 00001550
 00001560
 00001570
 00001580
 00001590
 00001600
 00001610
 00001620
 00001630
 00001640
 00001650
 00001660
 00001670
 00001680
 00001690
 00001700
 00001710
 00001720
 00001730
 00001740
 00001750

```

210 FORMAT(' INPUT TYPE OF PLY IN PLATE NO',I5,' FROM TOP',/,
  *' TO BOTTOM',/,
  *SX,'PLY TYPE',,OX,'ORIENTATION')
  N=NUMPLY(K)
  DO 212 L=1,N
    WRITE(6,713) L,ANG(L,K)
213 FORMAT(SX,I5,10X,F7.2,' DEGREES')
212 CONTINUE
    WRITE(6,711)
711 FORMAT(/)
    N=NPLY(K)
    DO 215 I=1,N
      WRITE(6,214) I
214 FORMAT(' INPUT TYPE OF PLY FOR PLY NO',I5)
      READ(5,*) IPLY(I,K)
215 CONTINUE
305 CONTINUE
      DO 306 K=1,NLIM
        WRITE(6,216) K
216 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF PLATE NO',I5)
        IF(CMK).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 YOUNGS MODULI, E1 AND E2')
        READ(5,*) E1(K),E2(K)
        WRITE(6,218)
218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO')
        READ(5,*) G12(K),V12(K)
        V21(K)=V12(K)*E2(K)/E1(K)
306 CONTINUE
        IF(UPR.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(J,I),I=1,15)
251 FORMAT(15A4)
        WRITE(6,252)
252 FORMAT(' INPUT YOUNG MODULUS AND POISSONS RATIO FOR',/,
  *' THE FASTENER')
        READ(5,*) FASE,FASV
        WRITE(6,253)
253 FORMAT(' INPUT THE DIAMETER OF THE FASTENER')
        READ(5,*) FASD
        WRITE(6,888)
888 FORMAT(/,' FASTENER TYPE',/,
  *' ENTER: 1 FOR PROTRUDING HEAD',/,
  *' 2 FOR COUNTERSUNK HEAD')
        READ(5,*) HFTYP
        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
00001990
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)=1.0E-10
IF(NFTYP.EQ.1) GO TO 360
WRITE(6,889)
889 FORMAT(/,' ENTER PLATE WHICH CONTAINS THE COUNTERSUNK',/,
  *' HEAD (OPPOSITE PLATE ASSUMES THE NUT HEAD) ',/,
  *' ENTER: 1 FOR TOP PLATE ',/,
  *' 2 FOR BOTTOM PLATE ')
READ(5,*) N
R(N)=0.0D0
360 CONTINUE

00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

000
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,*) XC(K,I),YC(K,I)
110 CONTINUE
  IF(K.EQ.2) GO TO 841
  A1=XC(1,1)
  B1=YC(1,1)
  A2=XC(1,2)
  B2=YC(1,2)
  XC(1,1)=XC(1,4)
  YC(1,1)=YC(1,4)
  XC(1,2)=XC(1,3)
  YC(1,2)=YC(1,3)
  XC(1,4)=A1
  YC(1,4)=B1
  XC(1,3)=A2
  YC(1,3)=B2
841 CONTINUE
  WTH=YC(K,2)-YC(K,1)
480 CONTINUE
  IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 567
  WRITE(6,741)
741 FORMAT(' SELECT FAILURE CRITERION: ',/,
  *' ENTER 1 FOR POINT STRESS CRITERION ',/,
  *' ENTER 2 FOR AVERAGE STRESS CRITERION ')
  READ(5,*) NPT

```

```

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

```

```

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

```

```

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

```

229	CONTINUE	00004160
671	CONTINUE	00004170
	IF(NOPT4.NE.4) GO TO 261	00004180
C		00004190
C	NUMBER OF DIVISIONS FOR STRESS AVERAGING	00004200
C	IS SET EQUAL TO 50	00004210
	NAVD=50	00004220
261	CONTINUE	00004230
	IF(BPR.EQ.1.0) GO TO 262	00004240
	DO 319 K=1,NLIM	00004250
	N=NUMPLY(K)	00004260
	WRITE(6,320) K	00004270
320	FORMAT(' SASCJ ASSUMES A BILINEAR PLY BEHAVIOR. THE ',,,	00004280
	*' INITIAL MODULUS, K1, IS COMPUTED BY THE CODE. ',,,	00004290
	*' THE REDUCED MODULUS, K2, FOR INITIAL FAILURE ',,,	00004300
	*' IN NET SECTION, SHEAROUT OR BEARING IS COMPUTED ',,,	00004310
	*' BY THE FORMULA K2=ALPHAK1. ',,,	00004320
	*' FOR PLATE NUMBER ',I5,' INPUT ALPHA VALUES FOR ',,,	00004330
	*' NET SECTION, SHEAROUT AND BEARING FAILURE ')	00004340
	READ(5,*) AF1,AF2,AF3	00004350
	DO 321 I=1,N	00004360
	DELNS(I,K)=AF1	00004370
	DELSR(I,K)=AF2	00004380
	DELSO(I,K)=AF3	00004390
321	CONTINUE	00004400
	WRITE(6,339)	00004410
339	FORMAT(' INPUT SCALE FACTORS FOR P ULTIMATE ',,,	00004420
	*' CALCULATION SUCH THAT P(ULT)=BETA*P(INITIAL) ',,,	00004430
	*' INPUT BETA1 FOR NET SECTION ULTIMATE ',,,	00004440
	*' BETA2 FOR BEARING ULTIMATE ',,,	00004450
	*' BETA3 FOR SHEAROUT ULTIMATE ')	00004460
	READ(5,*) PALT(3,K),PALT(2,K),PALT(1,K)	00004470
319	CONTINUE	00004480
391	CONTINUE	00004490
	IF(BPR.NE.0.0) GO TO 262	00004500
	DO 312 K=1,NLIM	00004510
	GAMD(K)=10.0	00004520
	IF(CMK(K).NE.CMC) GO TO 312	00004530
	WRITE(6,231) K	00004540
231	FORMAT(' INPUT THE APPROXIMATE INTERLAMINAR SHEAR STRAIN ',,,	00004550
	*' ULTIMATE FOR DELAMINATION PREDICTION IN PLATE NO ',I5,,	00004560
	*' (UNITS: IN/IN) ')	00004570
	READ(5,*) GAMD(K)	00004580
312	CONTINUE	00004590
262	CONTINUE	00004600
C		00004610
C	CASE HEADING	00004620
C		00004630
	WRITE(6,143)	00004640
143	FORMAT('PROGRAM SASCJ',,,)	00004650
	IF(NSDLS.EQ.1.AND.BPR.NE.1.0) WRITE(6,633)	00004660
	IF(NSDLS.EQ.2.AND.BPR.NE.1.0) WRITE(6,634)	00004670
633	FORMAT(2X,'A SINGLE LAP SHEAR JOINT WILL BE ANALYZED',,,)	00004680
634	FORMAT(2X,'A DOUBLE LAP SHEAR JOINT WILL BE ANALYZED',,,)	00004690
	IF(BPR.EQ.0.0) WRITE(6,881)	00004700
	IF(BPR.EQ.1.0) WRITE(6,882)	00004710
	IF(BPR.NE.0.0.AND.BPR.NE.1.0) WRITE(6,883) BPR	00004720
881	FORMAT(2X,'WITH A LOADED HOLE',,,)	00004730
882	FORMAT(2X,'WITH AN OPEN HOLE',,,)	00004740
		00004750

```

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

```

0000000000

```

      IF(IL.EQ.1) GO TO 23
      WRITE(6,871) K
871  FORMAT(/,5X,' GEOMETRY OF PLATE NO ',I5,' ',/)
      WRITE(6,872)
872  FORMAT(' COORDINATES OF CORNER VERTEXES ',/)
      IF(K.EQ.1) WRITE(6,873) XC(K,2),YC(K,2),XC(K,3),YC(K,3)
      IF(K.EQ.1) WRITE(6,874) XC(K,1),YC(K,1),XC(K,4),YC(K,4)
      IF(K.EQ.2) WRITE(6,873) XC(K,3),YC(K,3),XC(K,2),YC(K,2)
      IF(K.EQ.2) WRITE(6,874) XC(K,4),YC(K,4),XC(K,1),YC(K,1)
873  FORMAT(2X,F7.3,' ',F7.3,10X,F7.3,' ',F7.3,/)
874  FORMAT(2X,F7.3,' ',F7.3,10X,F7.3,' ',F7.3,/)
      AXD=AXX2.
      WRITE(6,875) AXD
875  FORMAT(' FASTENER HOLE DIAMETER = ',D9.3,' INCHES',/)
      ED=DABS(XC(K,3)-XC(K,1))/AXD
      WD=DABS(YC(K,3)-YC(K,1))/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
C    PROCESS INPUT DATA ON PLATE GEOMETRIES
C
      WTH=YC(K,2)-YC(K,1)
      LMI=L0M
      CALL POLY(JK,K,XC,YC,W,AST,NCOL,LTNCH,BPR,IL)
      CALL CIRC(W,AST,JK,K,LTNCH,BPR,IL)
      IF(NOPT4.EQ.1.OR.NOPT4.EQ.3) CALL RCOUT(K)
      IF(NOPT4.EQ.2) CALL PSTRSS(K,LTNCH,BPR,IL)
      IF(NOPT4.EQ.4) CALL AVSTRS(K,LTNCH,NAVD,BPR,IL)

C
C
C    PERFORM FINITE GEOMETRY ANALYSIS FOR STRESS/DISPLACEMENT
    STATE, COMPUTE FOUNDATION MODULI AND FAILURE VALUES
C
      CALL FIOEM(W,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.L0M.NE.1) CALL FBOLT(ANGK,H,K,NOPT1,LMI)
21  CALL FCRT(SALON,H,WTH,AST,K,NOPT1,NOPT4,BPR,NAVD,IL)
20  CONTINUE
22  CONTINUE
      IF(BPR.EQ.1.0) GO TO 410

C
C
C    PREPARE INPUT FOR SEQUENTIAL PLY FAILURE
    PREDICTION
C
      IF(L0M.GT.1) GO TO 61
      N=NPLY(1)
      DO 30 I=1,N
      M=IPLY(I,1)
30  PLYK(I)=ANGX(M,1)
      N=NPLY(2)
      DO 60 I=1,N
      N1=I+NPLY(1)
      N2=IPLY(I,2)
60  PLYK(N1)=ANGK(N2,2)
61  CONTINUE

C
C
C    CALCULATION OF FASTENER STIFFNESSES.

```

```

C
FASQ=FASE/(2.*(1.+FASV))
FASLAM=5.*(1.0+FASV)/(7.+5.*FASV)
FASR=FASQ/2.
FASA=ACOS(-1.)*FASR**2
FASI=ACOS(-1.)*FASR**4/4.
FASSS=FASLAM*FASQ*FASA
FASBS=FASE*FASI
00006360
00006370
00006380
00006390
00006400
00006410
00006420
00006430
00006440
00006450
00006460
00006470
00006480
00006490
00006500
00006510
00006520
00006530
00006540
00006550
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
INITIALIZATION
IF(LCM.GT.1) GO TO 72
ITT=0
NTFL=0
JNT=1
P=0.
DELP=1000.
DO 5012 I=1,100
  NPNM(I,1)=I
  NPNM(I,2)=I*NPLY(1)
  UN(I)=0.
  QAMN(I)=0.
  MDAHPI(I)=0.
  MDAMI(I)=0.
  PN(I)=0.
  BARK(I)=0.
5012 BARU(I)=0.
72 CONTINUE
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
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

90 CONTINUE
ITT=ITT+1
CALL CENTD(R,H,FASSS,FASBS,P,DELP,ITT)
CALL SOLVE(U,H,P,DELP,NSDLS,ITT)
CALL FAIL(QAMDLS,H,P,DELP,BPR,AST,WTH,PFAIL,ANGLE,NODE,
  *IROUT,NOPT4,NULTF,JNT,ITT,NTFL)
CALL PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT,
  *NPN,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
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
SUBROUTINE STRTH(H,ES1,ES2,ESS,AF1,AF2,AF4,K)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION AINV(3,3),AVN(3),M(2),NV(3)
DIMENSION NPLY(2),NUMPLY(2),ANG(3,2),IPLY(100,2)
DIMENSION WK(25),PSMX(3),ES1(2),ES2(2),ESS(2)
DIMENSION E1(2),E2(2),G12(2),V12(2),V21(2)
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
COMMON/MCD/E1,E2,G12,V12,V21
COMMON/AM1/A
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
C
C
C      COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM
      FIBER STRAINS FOR EACH FAILURE MODE
      CALL AMATRX(H,K)
      N=3
      IDOT=4
      IA=3
      CALL LINVZF(A,N,IA,AINV,IDOT,WK,IER)
      DO 100 KK=1,3
      DO 10 II=1,3
      NV(II)=0
10  AVN(II)=0.000
      IF(KK.EQ.1) NV(1)=1
      IF(KK.EQ.2) NV(1)=-1
      IF(KK.EQ.3) NV(3)=1
      DO 13 II=1,3
      DO 15 JJ=1,3
      AVN(IJ)=AVN(II)+AINV(IJ,JJ)*NV(JJ)
15  CONTINUE
      NP=NUMPLY(K)
      SMX=0.000
      RAD=DARCOS(-1.000)/180.000
      DO 25 II=1,NP
      TH=ANG(II,K)*RAD
      E11=DCOS(TH)*MX2+AVN(1)+AVN(2)*DSIN(TH)*MX2+
      *DCOS(TH)*DSIN(TH)*AVN(3)
      IF(KK.NE.1) GO TO 65
      EPRT=E11/ES2(K)
      GO TO 50
65  IF(KK.NE.2) GO TO 75
      EPRT=E11/ES1(K)
      GO TO 50
75  EPRT=E11/ES2(K)
50  CONTINUE
      IF(DABS(SMX).LT.DABS(EPRT)) SMX=EPRT
25  CONTINUE
      IF(DABS(SMX).GT.1.00-10) GO TO 555
      PSMX(KK)=ESS(K)*M12(K)
      GO TO 100
555  CONTINUE
      PSMX(KK)=DABS(1.000/SMX)
100  CONTINUE
      AF1=PSMX(1)
      AF2=PSMX(2)
      AF4=PSMX(3)
      RETURN
      END

C
C
C
C      SUBROUTINE POLY(J,K,XC,YC,W,AST,NCOL,LTHCM,BPR,IL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XC(2,5),YC(2,5),A1(400),A2(400),XB(400)
      DIMENSION YB(400),T(400),A1A(4),A2A(4)
      COMMON/CMT1/XB,YB,A1,A2,T
      ARRAY COLLOCATION POINTS AROUND EXTERIOR BOUNDARY
      AND APPLY STRESS BOUNDARY CONDITIONS

```

```

00007160
00007170
00007180
00007190
00007200
00007210
00007220
00007230
00007240
00007250
00007260
00007270
00007280
00007290
00007300
00007310
00007320
00007330
00007340
00007350
00007360
00007370
00007380
00007390
00007400
00007410
00007420
00007430
00007440
00007450
00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610
00007620
00007630
00007640
00007650
00007660
00007670
00007680
00007690
00007700
00007710
00007720
00007730
00007740
00007750

```

	DO 120 I=1,4	00007760
	A1A(I)=0.	00007770
	A2A(I)=0.	00007780
120	CONTINUE	00007790
	W=DABS(YC(K,2)-YC(K,1))	00007800
	IF(LTNCM.EQ.1) A1A(1)=1000.0	00007810
	IF(LTNCM.EQ.2) A1A(1)=-1000.0	00007820
	IF(BPR.NE.0.0) A1A(3)=A1A(1)	00007830
	IF(IL.EQ.2) A1A(3)=0.0	00007840
	AST=DABS(A1A(1))	00007850
	J=0	00007860
	XC(K,5)=XC(K,1)	00007870
	YC(K,5)=YC(K,1)	00007880
	PI=DARCOS(-1.000)	00007890
	DAT=PI/NCOL	00007900
	DO 10 I=1,4	00007910
	X=XC(K,I)-XC(K,I+1)	00007920
	Y=YC(K,I+1)-YC(K,I)	00007930
	IF(X.EQ.0.) X=1.D-6	00007940
	IF(Y.EQ.0.) Y=1.D-6	00007950
	TH=DAIAN2(X,Y)	00007960
	TH=TH*180./DARCOS(-0.1D1)	00007970
	DX=(XC(K,I+1)-XC(K,I))/(NCOL+1)	00007980
	DY=(YC(K,I+1)-YC(K,I))/(NCOL+1)	00007990
	DO 20 II=1,NCOL	00008000
	J=J+1	00008010
	IF(I.EQ.1.OR.I.EQ.4) GO TO 23	00008020
	YB(J)=YC(K,I)	00008030
	XB(J)=XC(K,I)+DX*(II-5)	00008040
	IF(II.EQ.1) XB(J)=XC(K,I)+(DX/2.)	00008050
	GO TO 24	00008060
23	CONTINUE	00008070
	IF(XC(K,5).NE.0.0) GO TO 26	00008080
	IF(I.NE.3) GO TO 24	00008090
	ADT=DAT*II	00008100
	XB(J)=YC(K,3)*DCOS((PI/2.)*ADT)	00008110
	YB(J)=YC(K,3)*DSIN((PI/2.)*ADT)	00008120
	TH=((PI/2.)*ADT)*180./PI	00008130
	GO TO 24	00008140
26	CONTINUE	00008150
	YB(J)=YC(K,1)+DY*(II-5)	00008160
	IF(II.EQ.1) YB(J)=YC(K,1)+(DY/2.)	00008170
	XB(J)=XC(K,1)	00008180
24	T(J)=TH	00008190
	A1(J)=A1A(I)	00008200
	A2(J)=A2A(I)	00008210
20	CONTINUE	00008220
10	CONTINUE	00008230
	RETURN	00008240
	END	00008250
C		00008260
C		00008270
C		00008280
	SUBROUTINE CIRC(W,AST,JK,K,LTNCM,BPR,IL)	00008290
	IMPLICIT REAL*8(A-H,O-Z)	00008300
	DIMENSION X(400),Y(400),THTA(400),A1(400),A2(400)	00008310
	DIMENSION XB(400),YB(400)	00008320
	COMMON/FB1/BSTR,XSTR	00008330
	COMMON/CMT1/XB,YB,A1,A2,THTA	00008340
	COMMON/CMF2/X,Y,NPST,HAFT	00008350

```

COMMON/EC/A,B,N                                00008360
ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY  00008370
AND APPLY BEARING STRESS IN A COSINUSOIDAL    00008380
DISTRIBUTION                                   00008390
                                                00008400
CON=-1.0                                         00008410
XSTR=AST                                         00008420
BSTR=(2.*MMXSTR)/(DARCOS(CON)*B)              00008430
IF(BPR.NE.0.0.AND.BPR.NE.1.0.AND.IE.EQ.1) BSTR=0.0 00008440
IF(BPR.EQ.1.0) BSTR=0.                         00008450
NM4=N-4                                          00008460
NQ=NM4/4                                         00008470
DO 20 I=1,N                                     00008480
  JK=JK+1                                        00008490
  TH=((I-1)*2+1)*DARCOS(CON)/N                  00008500
  X(I)=AMDCOS(TH)                               00008510
  Y(I)=AMD SIN(TH)                              00008520
  CS=-X(I)*BMB/(Y(I)*MAA)                       00008530
  IF(Y(I).GT.0) THTA(JK)=DATAN(CS)-DARCOS(CON)/2. 00008540
  IF(Y(I).LT.0) THTA(JK)=DATAN(CS)+DARCOS(CON)/2. 00008550
  THTA(JK)=THTA(JK)*180./DARCOS(CON)            00008560
  IF(LTNCM.EQ.2) GO TO 25                       00008570
  IF(I.GT.(NQ+1).AND.I.LT.(N-NQ)) GO TO 204    00008580
  GO TO 30                                       00008590
25 IF(I.LE.(NQ+2).OR.I.GE.(N-NQ-1)) GO TO 204  00008600
30 CONTINUE                                     00008610
  A1(JK)=0.                                     00008620
  A2(JK)=0.                                     00008630
  XB(JK)=X(I)                                   00008640
  YB(JK)=Y(I)                                   00008650
  GO TO 20                                       00008660
204 IF(Y(I).GT.0.) TETA=ARCCOS(-1.)-DATAN(Y(I)/X(I)) 00008670
    IF(Y(I).LT.0.) TETA=ARCCOS(-1.)+DATAN(Y(I)/X(I)) 00008680
    A1(JK)=-1.0*BSTR*MDABS(DCOS(TETA))           00008690
    A2(JK)=0.                                    00008700
    XB(JK)=X(I)                                  00008710
    YB(JK)=Y(I)                                  00008720
    GO TO 20                                       00008730
20 CONTINUE                                     00008740
  RETURN                                         00008750
END                                              00008760
SUBROUTINE RCDUT(K)                             00008770
SPECIFY COORDINATES AROUND CHARACTERISTIC CIRCLE 00008780
AN WHICH STRESSES ARE NEEDED FOR THE HOFFMAN/  00008790
TSAI-HILL FAILURE CRITERIA                     00008800
IMPLICIT REAL*8(A-H,O-Z)                       00008810
DIMENSION X(400),Y(400),RCA(2),RCB(2),NRC(2)  00008820
COMMON/CMT2/X,Y,NPST,MAST                      00008830
COMMON/RC/RCA,RCB,NRC                         00008840
COMMON/ELP/AX,BX,NOUT                         00008850
RAD=DARCOS(-0.1D1)/180.                       00008860
N1=NRC(K)                                       00008870
DO 40 I=1,N1                                   00008880
  TINC=360./NRC(K)                             00008890
  THETA=(I-1)*TINC*PI/180.                     00008900
  C=DCOS(THETA)                                00008910
  S=DSIN(THETA)                                00008920
  X(I)=X+AX*C                                  00008930
  Y(I)=Y+AY*S                                  00008940
  R(I)=R+AX*C+AY*S                             00008950

```

K=DSQRT(1./((CXX2/RCA(K)*X2)+(SXX2/RCB(K)*X2))
X(I+NGUT)=R*DCOS(THETA)
Y(I+NGUT)=R*DSIN(THETA)
40 CONTINUE
RETURN
END

SUBROUTINE PSTRSS(K,NCS,BPR,IL)

SPECIFY DISCRETE COORDINATES OF POINTS AT WHICH
STRESSES ARE REQUIRED FOR THE POINT STRESS
CRITERION

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(400),Y(400),DONT(2),DOBR(2)
DIMENSION DOSO(2),NPLY(2),NUMPLY(2),ANG(5,2)
DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSO(2,2)
COMMON/ELP/AX,BX,NGUT
COMMON/CMT2/X,Y,NPST,NAST
COMMON/PSC1/DONT,DOBR,DOSO
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
COMMON/AOV/AONT,AOBR,AOSO
AONT=DONT(K)
AOBR=DOBR(K)
AOSO=DOSO(K)
IF(BPR.EQ.0.0.OR.BPR.L2.1.0) GO TO 25
AONT=AONT(IL,K)
AOBR=AOBR(IL,K)
AOSO=AOSO(IL,K)
5 CONTINUE
L=NGUT+1
SQ=1.0
IF(NCS.EQ.1) SQ=-1.0
X(L)=0.
Y(L)=AONT+BX
X(L+1)=SQ*(AX+AOBR)
Y(L+1)=0
X(L+2)=SQ*(AX+AOSO)
Y(L+2)=BX
NPST=3
DO 555 I=1,3
L=NGUT+1+(I-1)
RETURN
END

SUBROUTINE AVSTRS(K,NCS,NAVD,BPR,IL)

SPECIFY COORDINATES OF POINTS ALONG WHICH
STRESSES WILL BE AVERAGED FOR THE AVERAGE
STRESS CRITERION

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(400),Y(400),DONT(2),DOBR(2)
DIMENSION DOSO(2),NPLY(2),NUMPLY(2),ANG(5,2)
DIMENSION IPLY(100,2),AONT(2,2),AOBR(2,2),AOSO(2,2)
COMMON/AOV/AONT,AOBR,AOSO

00008960
00008970
00008980
00008990
00009000
00009010
00009020
00009030
00009040
00009050
00009060
00009070
00009080
00009090
00009100
00009110
00009120
00009130
00009140
00009150
00009160
00009170
00009180
00009190
00009200
00009210
00009220
00009230
00009240
00009250
00009260
00009270
00009280
00009290
00009300
00009310
00009320
00009330
00009340
00009350
00009360
00009370
00009380
00009390
00009400
00009410
00009420
00009430
00009440
00009450
00009460
00009470
00009480
00009490
00009500
00009510
00009520
00009530
00009540
00009550

```

COMMON/EL./AX,BX,NOUT                                00009560
COMMON/CMT2/X,Y,NPST,NAST                            00009570
COMMON/PSCI/DONT,DOBR,DOSO                          00009580
COMMON/LYP/NPLY,NUMPLY,ANG,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=ANT(IL,K)                                         00009640
ABR=ABR(IL,K)                                         00009650
ASO=ASO(IL,K)                                         00009660
25 CONTINUE                                           00009670
L=NOUT                                                00009680
SQ=1.0                                                00009690
IF(NCS.EQ.1) SQ=-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)=SQ*(BX+ANSO/2.+(I-1)*ANSO)                 00009790
    Y(L)=BX                                           00009800
30 ANBR=ABR/FLOAT(NAVD)                              00009810
  DO 40 I=1,NAVD                                       00009820
    L=L+1                                             00009830
    X(L)=SQ*(AX+ANBR/2.+(I-1)*ANBR)                 00009840
    Y(L)=0.                                           00009850
40 NAST=3*NAVD                                       00009860
  N1=NOUT+1                                           00009870
  N2=N1+NAST                                          00009880
  NN=NOUT+3*NAVD                                     00009890
  RETURN                                              00009900
  END                                                  00009910
                                                    00009920
                                                    00009930
SUBROUTINE FIGEOM(H,KJ,NOPT4,ITT)                    00009940
                                                    00009950
                                                    00009960
                                                    00009970
                                                    00009980
FIGEOM PERFORMS A FINITE GEOMETRY ANALYSIS          00009990
USING THE BOUNDARY COLLOCATION TECHNIQUE              00010000
                                                    00010010
                                                    00010020
                                                    00010030
IMPLICIT REAL*8(A-H,O-Z)                             00010040
DIMENSION A(3,3),WK(25),AI(3,3),AZ(3),WKK(121),BC(400) 00010050
DIMENSION CH(4),H(2)                                00010060
COMPLEX*16 GRHS(122)                                00010070
COMPLEX*16 CM(196,124),CMC(196,121),CMCTCM(121,121),RHS(121) 00010080
COMMON/ROOTS/R1,R2                                  00010090
COMMON/TERMS/P1,Q1,P2,Q2                            00010100
COMMON/ELP/AX,BX,NOUT                               00010110
COMMON/SER/NT,NB                                    00010120
COMMON/AMT/A                                          00010130
COMPLEX*16 Z(4),Z1,Z2,Q1,Q2,P1,P2,R1,R2,WA(14884) 00010140
                                                    00010150

```

```

C      AMATRX CALCULATES THE LAMINATE 'A' MATRIX
C      CALL AMATRX(H,KJ)
C      I=3
C      IDGT=4
C      A=3
C      LINVZF INVERTS THE 'A' MATRIX
C      CALL LINVZF(A,N,IA,AI,IDGT,HK,IER)
C      NDEG=4
C      AZ(1)=AI(1,1)/AI(2,2)
C      AZ(2)=-2.*AI(1,3)/AI(2,2)
C      AZ(3)=(2.*AI(1,2)+AI(3,3))/AI(2,2)
C      AZ(4)=-2.*AI(2,3)/AI(2,2)
C      AZ(5)=1.000
C      ZRPOLY FINDS THE ROOTS OF THE CHARACTERISTIC EQUATION
C      CALL ZRPOLY(AZ,NDEG,Z,IER)
C      Z(2) AND Z(4) ARE THE COMPLEX CONJUGATES OF Z(1)
C      AND Z(3) RESPECTIVELY
C      R1=Z(1)
C      R2=Z(3)
C      THE TWO ROOTS MUST BE CHECKED FOR A UNITARY COMPONENT
C      IN EITHER THE REAL OR IMAGINARY PART; SUCH AN
C      OCCURANCE SIGNIFIES A QUASI-ISOTROPIC LAYUP AND
C      THE VALUE MUST BE PERTURBED SLIGHTLY IN ORDER TO
C      AVOID A SINGULAR MATRIX
C      CH(1)=R1
C      CH(2)=(0.0,-1.0)*R1
C      CH(3)=R2
C      CH(4)=(0.0,-1.0)*R2
C      DO 30 IJK=1,4
C      IF(DABS(CH(IJK)).LT.1.0D-10) CH(IJK)=1.0D-10
C      AR=DABS(CH(IJK))
C      IF(AR.LE.1.0) GO TO 11
C      GO TO 32
31 IF((1.0-AR).LT.0.02) CH(IJK)=0.98
C      GO TO 30
32 IF((AR-1.0).LT.0.02) CH(IJK)=1.02
30 CONTINUE
C      R1=DCMPLX(CH(1),CH(2))
C      R2=DCMPLX(CH(3),CH(4))
C      CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS
C      P1=AI(1,1)*R1**2+AI(1,2)*AI(1,3)*R1
C      P2=AI(1,1)*R2**2+AI(1,2)*AI(1,3)*R2
C      Q1=AI(2,2)/R1+AI(1,2)*R1-AI(2,3)
C      Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)
C      INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS
C      NT4=4*NT
C      NT8=8*NT

```

```

00010160
00010170
00010180
00010190
00010200
00010210
00010220
00010230
00010240
00010250
00010260
00010270
00010280
00010290
00010300
00010310
00010320
00010330
00010340
00010350
00010360
00010370
00010380
00010390
00010400
00010410
00010420
00010430
00010440
00010450
00010460
00010470
00010480
00010490
00010500
00010510
00010520
00010530
00010540
00010550
00010560
00010570
00010580
00010590
00010600
00010610
00010620
00010630
00010640
00010650
00010660
00010670
00010680
00010690
00010700
00010710
00010720
00010730
00010740
00010750

```

```

      NT8P4=8*NT+4
      NT8P2=8*NT+2
      NT8P1=8*NT+1
      NB2=2*NB
      NWK=NT8P1*(NT8P1+2)
      CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,
      INT8P1,NB2,NWK,WA,WKK,AI,NOPT4,KJ,ITT)
      RETURN
      END

      SUBROUTINE AMATRX(H,K)
      ASSEMBLE THE A MATRIX

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3,3),ANG(5,2),H(2),NPLY(2),NUMPLY(2)
      DIMENSION E1(2),E2(2),Q12(2),V12(2),V21(2)
      DIMENSION IPLY(100,2)
      COMMON/MOD/E1,E2,Q12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/AMT/A
      THKNES=NPLY(K)*H(K)
      DENO=1.-E2(K)*V12(K)*H2/E1(K)
      Q11=E1(K)/DENO
      Q22=E2(K)/DENO
      Q12=V12(K)*Q22
      Q21=Q12
      Q33=Q12(K)
      DO 10 I=1,3
      DO 10 J=1,3
10  A(I,J)=0.
      NN=NPLY(K)
      DO 20 I=1,NN
      T=H(K)
      LP=IPLY(I,K)
      THTAI=ANG(LP,K)*DARCOS(-1.DO)/180.DO
      C=DCOS(THTAI)
      S=DSIN(THTAI)
      A(1,1)=(Q11*CMH4+2.*(Q12+2.*Q33)*CMCHMS+S+Q22*SMH4)*NT+A(1,1)
      A(2,2)=(Q11*SMH4+2.*(Q12+2.*Q33)*CMCHMS+S+Q22*CMH4)*NT+A(2,2)
      A(1,2)=(Q11+Q22-4.*Q33)*CMCHMS+Q12*(CMH4+SMH4)*NT+A(1,2)
      A(2,1)=A(1,2)
      A(3,3)=(Q11+Q22-2.*Q12-2.*Q33)*CMCHMS+Q33*(CMH4+SMH4)*NT+A(3,3)
      A(1,3)=(Q11-Q12-2.*Q33)*CMH3MS+(Q12-Q22+2.*Q33)*SMH3MC)*NT+A(1,3)
      A(2,3)=(Q11-Q12-2.*Q33)*SMH3MC+(Q12-Q22+2.*Q33)*CMH3MS)*NT+A(2,3)
      A(3,2)=A(2,3)
      A(3,1)=A(1,3)
20  CONTINUE
      DO 53 I=1,3
      DO 53 J=1,3
      A(I,J)=A(I,J)/THKNES
53  CONTINUE
      RETURN
      END

      SUBROUTINE CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,

```

C
C
C
C
C
C

1HT8P1,NB2,NWK,WA,WKK,AMAT,NOPT4,KJ,ITT)

CMAT OUTPUTS STRESSES, STRAINS, AND DISPLACEMENTS
 AT SPECIFIED COORDINATES

```

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION RCA(2),RCB(2),NRCOUT(2)
DIMENSION ASX(400),ASXY(400)
COMMON/XXY1/ASX,ASXY
COMMON/ROOTS/R1,R2
COMMON/TERMS/P1,Q1,P2,Q2
COMMON/CMT1/X,Y,AIN1,AIN2,THTA
COMMON/CMT2/XOUT,YOUT,NPST,NAST
COMMON/FB2/FUR,FTHT,FMR
COMMON/GMT/RTHT,REPX,REPY,REPLY
COMMON/RC/RCA,RCB,NRCOUT
COMMON/ELP/AX,BX,HOUT
COMMON/SER/NT,NB
DIMENSION THTA(400),X(400),Y(400),AMAT(3,3)
DIMENSION AIN1(400),AIN2(400),BC(NB2)
DIMENSION XOUT(400),YOUT(400),WKK(NT8P1)
DIMENSION FUR(400),FTHT(400),FMR(400)
DIMENSION RTHT(400),REPX(400),REPY(400),REPLY(400)
COMPLEX*16 CMCTCM(NT8P1,NT8P1),RHS(NT8P1),PH1D,PHI2D,XETA1,XETA2
COMPLEX*16 CM(NB2,NT8P4),CMC(NB2,NT8P1),Z1,Z2,Z11,Z22,R1,R2
COMPLEX*16 T11,T12,T21,T22,P11,P12,P21,P22
COMPLEX*16 P1,P2,Q1,Q2,DCMPLX,CO,CSUM,GRHS(NT8P2)
COMPLEX*16 PH1DP,PHI2DP,PHI1DN,PHI2DN
COMPLEX*16 PH1P,PHI2P,PHI1N,PHI2N,PHI1,PHI2
COMPLEX*16 SV11,SV12,SV21,SV22,RB11,RB21,RB1B,RB2B
COMPLEX*16 R1B,R2B,P1B,P2B,Q1B,Q2B,WA(NWK)
A=AX
B=BX
CO=(0.0,1.0)
RB11=(Q1-P1MR1)/(A*CMR1NB)
RB21=(Q2-P2MR2)/(A*CMR2NB)
REALR1=R1
REALR2=R2
REALP1=P1
REALP2=P2
REALQ1=Q1
REALQ2=Q2
RRB11=RB11
RRB21=RB21
AIMGR1=CMR1
AIMGR2=CMR2
AIMGP1=CMP1
AIMGP2=CMP2
AIMGQ1=CMQ1
AIMGQ2=CMQ2
ARB11=CO*RB11
ARB21=CO*RB21
R1B=DCMPLX(REALR1,AIMGR1)
R2B=DCMPLX(REALR2,AIMGR2)
P1B=DCMPLX(REALP1,AIMGP1)
P2B=DCMPLX(REALP2,AIMGP2)
Q1B=DCMPLX(REALQ1,AIMGQ1)
Q2B=DCMPLX(REALQ2,AIMGQ2)

```

00011360
 00011370
 00011380
 00011390
 00011400
 00011410
 00011420
 00011430
 00011440
 00011450
 00011460
 00011470
 00011480
 00011490
 00011500
 00011510
 00011520
 00011530
 00011540
 00011550
 00011560
 00011570
 00011580
 00011590
 00011600
 00011610
 00011620
 00011630
 00011640
 00011650
 00011660
 00011670
 00011680
 00011690
 00011700
 00011710
 00011720
 00011730
 00011740
 00011750
 00011760
 00011770
 00011780
 00011790
 00011800
 00011810
 00011820
 00011830
 00011840
 00011850
 00011860
 00011870
 00011880
 00011890
 00011900
 00011910
 00011920
 00011930
 00011940
 00011950

```

RB11B=DCMPLX(RRB11,ARB11) 00011960
RB21B=DCMPLX(RRB21,ARB21) 00011970
JJJ=0 00011980
DO 1000 I=1,NB 00011990
J=IM2 00012000
THTAI=THTA(I)*DARCOS(-1.0D0)/180.D0 00012010
C=DCOS(THTAI) 00012020
S=DSIN(THTAI) 00012030
P11=CXP1+SNQ1 00012040
P12=CXP2+SNQ2 00012050
P21=-SNP1+CXQ1 00012060
P22=-SNP2+CXQ2 00012070
T11=(CXCR1*NR1+SN5-2.*CXSNR1) 00012080
T12=(CXCR2*NR2+SN5-2.*CXSNR2) 00012090
T21=(-CXSNR1*NR1+CX5-(CXCR-SN5)*NR1) 00012100
T22=(-CXSNR2*NR2+CX5-(CXCR-SN5)*NR2) 00012110
Z1=X(I)+R1*Y(I) 00012120
Z2=X(I)+R2*Y(I) 00012130
Z11=CDSQRT(Z1*Z1-AXA-R1*NR1*NB) 00012140
Z22=CDSQRT(Z2*Z2-AXA-R2*NR2*NB) 00012150
REAL1=Z11 00012160
AIMO1=-CONZ11 00012170
IF(DABS(REAL1).LE.1.D-16)REAL1=0.0D0 00012180
IF(DABS(AIMO1).LE.1.D-16)AIMO1=0.0D0 00012190
Z11=DCMPLX(REAL1,AIMO1) 00012200
REAL2=Z22 00012210
AIMO2=-CONZ22 00012220
IF(DABS(REAL2).LE.1.D-16)REAL2=0.0D0 00012230
IF(DABS(AIMO2).LE.1.D-16)AIMO2=0.0D0 00012240
Z22=DCMPLX(REAL2,AIMO2) 00012250
XETA1=(Z1+Z11)/(A-COMR1*NB) 00012260
IF(CDABS(XETA1).LT.0.999) GO TO 300 00012270
GO TO 310 00012280
300 Z11=-Z11 00012290
XETA1=(Z1+Z11)/(A-COMR1*NB) 00012300
310 XETA2=(Z2+Z22)/(A-COMR2*NB) 00012310
IF(CDABS(XETA2).LT.0.999) GO TO 320 00012320
GO TO 330 00012330
320 Z22=-Z22 00012340
XETA2=(Z2+Z22)/(A-COMR2*NB) 00012350
330 CONTINUE 00012360
JJJ=JJJ+1 00012370
C C C 00012380
NORMAL & TANGENTIAL STRESS BOUNDARY CONDITIONS ARE IMPOSED 00012390
DO 5 N=1,NT 00012400
NP=N 00012410
CM(J-1,N)=NP*XETA1*NNPNT11/Z11 00012420
CM(J-1,2*NT+N)=NP*XETA2*NNPNT12/Z22 00012430
CM(J,N)=NP*XETA1*NNPNT21/Z11 00012440
CM(J,2*NT+N)=NP*XETA2*NNPNT22/Z22 00012450
NN=-N 00012460
CM(J-1,NT+N)=NN*XETA1*NNNT11/Z11 00012470
CM(J-1,3*NT+N)=NN*XETA2*NNNT12/Z22 00012480
CM(J,NT+N)=NN*XETA1*NNNT21/Z11 00012490
CM(J,3*NT+N)=NN*XETA2*NNNT22/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
CM(J,NT8+2)=T22/Z22 00012550

```

```

1000 CM(J,NT8+2)=T22/Z22
      CONTINUE
      DO 195 I=1,NB2
      DO 196 J=1,NT4
      REAL1=CM(I,J)
      AIMQ1=-CONJG(CM(I,J))
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.000
      IF(DABS(AIMQ1).LE.1.D-16)AIMQ1=0.000
      CM(I,J)=DCMPLX(REAL1,AIMQ1)
      AIMQ2=-AIMQ1
      CM(I,NT4+J)=DCMPLX(REAL1,AIMQ2)
196 CONTINUE
195 CONTINUE
      DO 295 I=1,NB2
      DO 296 J=1,2
      REAL1=CM(I,NT8+J)
      AIMQ1=-CONJG(CM(I,NT8+J))
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.000
      IF(DABS(AIMQ1).LE.1.D-16)AIMQ1=0.000
      CM(I,NT8+J)=DCMPLX(REAL1,AIMQ1)
      AIMQ2=-AIMQ1
      CM(I,NT8+2+J)=DCMPLX(REAL1,AIMQ2)
296 CONTINUE
295 CONTINUE
      SV11=(P2*Q1B-Q2*P1B)/(Q1*P2-Q2*P1)
      SV12=(P2*Q2B-Q2*P2B)/(Q1*P2-Q2*P1)
      SV21=(Q1*P1B-Q1B*P1)/(Q1*P2-Q2*P1)
      SV22=(Q1*P2B-Q2B*P1)/(Q1*P2-Q2*P1)
      DO 139 I=1,NB2
      IMPOSE RIGID BODY ROTATION CONDITION
      CM(I,2*NT+1)=-CM(I,1)*RB21/RB11+CM(I,2*NT+1)
      CM(I,4*NT+1)=-CM(I,1)*RB11B/RB11+CM(I,4*NT+1)
      CM(I,6*NT+1)=-CM(I,1)*RB21B/RB11+CM(I,6*NT+1)
      CM(I,1)=(0.0,0.0)
      IMPOSE SINGLE-VALUEDNESS CONDITION
      CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+1)*SV12+CM(I,NT8+4)
      CM(I,NT8+3)=CM(I,NT8+2)*SV21+CM(I,NT8+3)
      CM(I,NT8+4)=CM(I,NT8+2)*SV22+CM(I,NT8+4)
      CM(I,NT8+1)=(0.0,0.0)
      CM(I,NT8+2)=(0.0,0.0)
139 CONTINUE
      DO 141 I=1,NB2
      DO 142 J=2,NT8
142 CM(I,J-1)=CM(I,J)
      CM(I,NT8)=CM(I,NT8+3)
      CM(I,NT8+1)=CM(I,NT8+4)
141 CONTINUE
      DO 95 I=1,NB2
      DO 96 J=1,NT8P1
      REAL1=CM(I,J)
      AIMQ1=-CONJG(CM(I,J))
      IF(DABS(REAL1).LE.1.D-16)REAL1=0.000
      IF(DABS(AIMQ1).LE.1.D-16)AIMQ1=0.000
      CM(I,J)=DCMPLX(REAL1,AIMQ1)
      AIMQ2=-AIMQ1

```

0001256J
 00012570
 00012580
 00012590
 00012600
 00012610
 00012620
 00012630
 00012640
 00012650
 00012660
 00012670
 00012680
 00012690
 00012700
 00012710
 00012720
 00012730
 00012740
 00012750
 00012760
 00012770
 00012780
 00012790
 00012800
 00012810
 00012820
 00012830
 00012840
 00012850
 00012860
 00012870
 00012880
 00012890
 00012900
 00012910
 00012920
 00012930
 00012940
 00012950
 00012960
 00012970
 00012980
 00012990
 00013000
 00013010
 00013020
 00013030
 00013040
 00013050
 00013060
 00013070
 00013080
 00013090
 00013100
 00013110
 00013120
 00013130
 00013140
 00013150

```

    CMC(I,J)=DCMPLX(REAL1,AIMG2)
96  CONTINUE
95  CONTINUE
    DO 120 I=1,NB
      J=IM2
      BC(J-1)=AIN1(I)
      BC(J)=AIN2(I)
120  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)*CMC(K,J)+CSUM
      CMCTCM(I,J)=CSUM
100  CONTINUE
      DO 130 I=1,NT8P1
      CSUM=(0.0,0.0)
      DO 140 K=1,NB2
140  CSUM=CMC(K,I)*BC(K)+CSUM
130  RHS(1)=CSUM
      IJOB=0
      M=1
      CALL LEQ2C(CMCTCM,NT8P1,NT8P1,RHS,M,NT8P1,IJOB,WA,WKK,IER)
      ORHS(1)=-(RHS(2*NT)+RB21+RHS(4*NT)*RB11B+RHS(6*NT)*RB21B)/RB11
      ORHS(8*NT+1)=RHS(8*NT)*SV11+RHS(8*NT+1)*MSV12
      ORHS(8*NT+2)=RHS(8*NT)*SV21+RHS(8*NT+1)*MSV22
      DO 151 I=2,NT8
151  ORHS(I)=RHS(I-1)

C
C
C    STRESS AND STRAIN CALCULATION

    NRC3=NOUT+1
    IF(NOPT4.EQ.1.OR.NOPT4.EQ.3) NRCF=NOUT+NRCOUT(KJ)
    IF(NOPT4.EQ.2) NRCF=NOUT+NPST
    IF(NOPT4.EQ.4) NRCF=NOUT+NAST
    DO 190 K=1,NRCF
      Z1=XOUT(K)+R1*YOUT(K)
      Z2=XOUT(K)+R2*YOUT(K)
      Z11=CDSQRT(Z1*Z1-A*A-R1*R1*NMB)
      Z22=CDSQRT(Z2*Z2-A*A-R2*R2*NMB)
      XETA1=(Z1+Z11)/(A-C0*R1*B)
      IF(CDABS(XETA1).LT.0.999) GO TO 400
      GO TO 410
400  Z11=-Z11
      XETA1=(Z1+Z11)/(A-C0*R1*B)
410  XETA2=(Z2+Z22)/(A-C0*R2*B)
      IF(CDABS(XETA2).LT.0.999) GO TO 420
      GO TO 430
420  Z22=-Z22
      XETA2=(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

```

00013150
 00013170
 00013180
 00013190
 00013200
 00013210
 00013220
 00013230
 00013240
 00013250
 00013260
 00013270
 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

ሕዝብ ሕዝብ

135

```

C      INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION          00014960
CCCC      MODULI                                             00014970
C      MNM=LI2-1                                           00014980
C      PLOADX=0.                                           00014990
C      WK=0.                                               00015000
C      DO 70 I=LI1,MNM                                     00015010
C      TH1=ATETAA(I+1)-ATETAA(I)                         00015020
C      TH2=(ATETAA(I)+ATETAA(I+1))/2.                   00015030
C      THETA=TH2*HRAD                                     00015040
C      C=DCOS(THETA)                                       00015050
C      S=DSIN(THETA)                                       00015060
C      R=DSQRT(1./(C**2/AX**2+S**2/BX**2))               00015070
C      FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*HRMTH1*HRAD*H(K) 00015080
C      FORCRT=((ASIGRT(I)+ASIGRT(I+1))/2.)*HRMTH1*HRAD*H(K) 00015090
C      PLOADX=PLOADX+FORCR*C-FORCRT*S                    00015100
70      CONTINUE                                           00015110
C      FK(I)=DABS(PLOADX/H(K)*DELEFF))                  00015120
C      PLX(I)=(K-1)*NPLY(1)+PLOADX                      00015130
100      CONTINUE                                           00015140
C      NT=MNPLY(K)                                         00015150
C      NN=MPLY(K)                                           00015160
C      DO 310 I=1,NT                                       00015170
C      DO 310 II=1,NN                                       00015180
C      IF(IPLY(II,K).EQ.I) ANGK(I,K)=FK(II)             00015190
C      IF(IPLY(II,K).EQ.I) PLXPT(I)=PLX(II+(K-1)*NPLY(1)) 00015200
310      CONTINUE                                           00015210
C      NP=MNPLY(K)                                         00015220
C      NP=MNPLY(K)                                         00015230
C      NP=MNPLY(K)                                         00015240
C      NP=MNPLY(K)                                         00015250
C      NP=MNPLY(K)                                         00015260
C      NP=MNPLY(K)                                         00015270
C      NP=MNPLY(K)                                         00015280
C      NP=MNPLY(K)                                         00015290
C      NP=MNPLY(K)                                         00015300
C      NP=MNPLY(K)                                         00015310
C      NP=MNPLY(K)                                         00015320
C      NP=MNPLY(K)                                         00015330
C      NP=MNPLY(K)                                         00015340
C      NP=MNPLY(K)                                         00015350
C      NP=MNPLY(K)                                         00015360
C      NP=MNPLY(K)                                         00015370
C      NP=MNPLY(K)                                         00015380
C      NP=MNPLY(K)                                         00015390
C      NP=MNPLY(K)                                         00015400
C      NP=MNPLY(K)                                         00015410
C      NP=MNPLY(K)                                         00015420
C      NP=MNPLY(K)                                         00015430
C      NP=MNPLY(K)                                         00015440
C      NP=MNPLY(K)                                         00015450
C      NP=MNPLY(K)                                         00015460
C      NP=MNPLY(K)                                         00015470
C      NP=MNPLY(K)                                         00015480
C      NP=MNPLY(K)                                         00015490
C      NP=MNPLY(K)                                         00015500
C      NP=MNPLY(K)                                         00015510
C      NP=MNPLY(K)                                         00015520
C      NP=MNPLY(K)                                         00015530
C      NP=MNPLY(K)                                         00015540
C      NP=MNPLY(K)                                         00015550

CCCC      COMPUTE TOTAL BEARING LOAD
C      IF(K.EQ.1) GO TO 611
C      PLXTOT=0.
C      TH=H(1)*NPLY(1)+H(2)*NPLY(2)
C      BLOAD=(BSTR*DA*CO5(-1.0D0)*8X*TH)/2.
C      NN=NPLY(1)+NPLY(2)
C      DO 212 I=1,NN
C      PLXTOT=PLXTOT+PLX(I)
212      CONTINUE
611      CONTINUE
C      RETURN
C      END

CCCC      SUBROUTINE FCRT(SALON,H,MTH,AST,K,NOPT1,NOPT4,BPR,NAVD,IL)
CCCC      FAILURE LOAD CALCULATION
CCCC      IMPLICIT REAL*8(A-H,O-Z)
CCCC      DIMENSION ASIG1(400),ASIG2(400),ASIG6(400),ASIGR(400)
CCCC      DIMENSION UR(400),FSMR(400),ATETAA(400),ASIGRT(400),NMPLY(2)
CCCC      DIMENSION ANG(5,2),IPLY(100,2),HFMC(3,2),PLXPT(100),NPLY(2)
CCCC      DIMENSION PNS(5,2),PBR(5,2),P50(5,2),PALT(3,2)

```

```

DIMENSION H(2), SALOW(2), SX(400), SXY(400), RCA(2), RCB(2), NRC(2), 00015560
DIMENSION AEP51(400), PFL(5,2), PSTC(5,5,2) 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/AEP51 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/FAL1/PHS, PBR, PJO, PALT 00015680
COMMON/FAL3/PFL 00015690
COMMON/STRESS/ASIOR, ASIORT, ASIO1, ASIO2, ASIO6 00015700
COMMON/PSC2/PSTC 00015710
COMMON/PSC3/SX, SXY 00015720
RAD=DARCO3(-0.1D1)/180. 00015730
IF(11.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,19) K 00015740
19 FORMAT(//, ' ANALYSIS OF PLATE NO', I3, ' ', //) 00015750
IF(NOPT4.EQ.2) GO TO 20 00015760
IF(NOPT4.EQ.3) GO TO 30 00015770
IF(NOPT4.EQ.4) GO TO 40 00015780
00015790
00015800
00015810
00015820
00015830
00015840
00015850
00015860
00015870
00015880
00015890
00015900
00015910
00015920
00015930
00015940
00015950
00015960
00015970
00015980
00015990
00016000
00016010
00016020
00016030
00016040
00016050
00016060
00016070
00016080
00016090
00016100
00016110
00016120
00016130
00016140
00016150

HOFFMAN/TSAI-HILL CRITERIA

WRITE(6,10)
10 FORMAT(//, ' HOFFMAN/TSAI-HILL CRITERION', //)
LI1=NOUT+1
LI2=LI1+NRC(K)
NCAS=2
NN=NUMPLY(K)
DO 402 I=1, NN
THETA=ANG(I,K)*RAD
CALL QMATX(K, LI1, LI2, NCAS, NOPT1, RAD, THETA)
PFAIL=1.0D10
N1=NRC(K)
IF(BPR.EQ.0.0) DSB=DABS(PLXPT(I))
IF(BPR.EQ.1.0) DSB=DABS(XSTR)
DO 404 J=1, N1
S1=ASIO1(J)/DSB
S2=ASIO2(J)/DSB
S6=ASIO6(J)/DSB
CALL HOFF(S1, S2, S6, A, B, K)
NNN=NOUT+J

FOR EACH PLY TYPE FIND THE LOCATION AND MAGNITUDE
OF THE HIGHEST HOFFMAN/TSAI-HILL FAILURE INDEX VALUE

P1=(-B+DSQRT(B**2+4*A))/(2*MA)
P2=(-B-DSQRT(B**2+4*A))/(2*MA)
IF(P1.LT.0.D0) PF=P2
IF(P2.LT.0.D0) PF=P1
IF(P1.LT.P2.AND.P1.GT.0.D0) PF=P1
IF(P2.LT.P1.AND.P2.GT.0.D0) PF=P2
IF(DABS(PF).GT.PFAIL) GO TO 480

```

```

PFAIL=DABJ(PF)
AMAX=A
BMAX=B
LOC=J
480 CONTINUE
404 CONTINUE
A=AMAX
B=BMAX
THTA=ATETAA(NOUT+LOC)

THE CORRESPONDING FAILURE LOAD IS OBTAINED FROM THE INDEX VALUE

XULT=HFMC(1,K)
IF(ASIO1(LOC).LT.0.) XULT=-HFIC(2,K)
YULT=HFMC(3,K)
IF(ASIC2(LOC).LT.0.) YULT=-HFMC(4,K)
SULT=HFIC(5,K)
IF(ASIO6(LOC).LT.0.) SULT=-HFMC(5,K)
SR1=ASIO1(LOC)/XULT
SR2=ASIO2(LOC)/YULT
SR6=ASIO6(LOC)/SULT
405 IF(BPR.EQ.0.0) WRITE(6,405) I,ANO(I,K),THTA,PFAIL,SR1,SR2,SR6
FORMAT(//,' FOR PLY TYPE NO ',I5,' ( ',D9.3,' DEGREES ),',//,
* THE HIGHEST FAILURE INDEX WAS FOUND AT ',D9.3,' DEGREES',//,
* THE CORRESPONDING FAILURE LOAD = ',D9.3,' LBS',//,
* THE STRESS RATIOS AT THIS LOCATION ARE',//,
* SIG1/XULT = ',D9.3,/,
* SIG2/YULT = ',D9.3,/,
* SIG6/SULT = ',D9.3,/,
PFL(I,K)=PFAIL
402 CONTINUE
IF(BPR.EQ.0.0) GO TO 80
SFAIL=1.0D10
DO 110 I=1,NN
IF(SFAIL.GT.PFL(I,K)) NPY=I
110 IF(SFAIL.GT.PFL(I,K)) SFAIL=PFL(I,K)
PLFL=SFAIL*WTHM(K)*NPLY(K)
WRITE(6,771) PLFL
771 FORMAT(//,' FOR THE OPEN HOLE LAMINATE, FAILURE',//,
* IS PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
GO TO 80
20 CONTINUE

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
LI1=NOUT+1
LI2=LI1+2
IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) NN=I
DO 100 I=1,NN
THETA=ANO(I,K)*RAD
CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 705

```

```

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

```

40 CONTINUE

IF(IL.EQ.1.AND.(BPR.EQ.0.0.OR.BPR.EQ.1.0)) WRITE(6,55)
55 FORMAT(/,' AVERAGE STRESS CRITERION ',//)
LI1=NOUT+1
NN=NUMPLY(K)
NCAS=2
IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) NN=1
DO 105 I=1,NN
LI2=NOUT+3*NAVD
THETA=ANG(I,K)*RAD
CALL QMATX(K,LI1,LI2,NCAS,NOPT1,RAD,THETA)

CALCULATE AVERAGE STRESS

SUM=0.
N1=1
N2=NAVD
DO 200 IJ=N1,N2
200 SUM=SUM+5X(IJ)
AS1=SUM/FLOAT(NAVD)
N1=NAVD+1
N2=2*NAVD
SUM=0.
DO 215 IJ=N1,N2
215 SUM=SUM+5XY(IJ)
AS2=SUM/FLOAT(NAVD)
SUM=0.
N1=2*NAVD+1
N2=3*NAVD
DO 220 IJ=N1,N2
220 SUM=SUM+5X(IJ)
AS3=SUM/FLOAT(NAVD)
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) GO TO 720
IF(IL.EQ.1) FAC=BPR
IF(IL.EQ.2) FAC=(1.-BPR)/PLXPT(1)
BPSTS(K,I,IL,1)=AS1*FAC/(ASTMWTNN(K)*NPLY(K))
BPSTS(K,I,IL,2)=AS3*FAC/(ASTMWTNN(K)*NPLY(K))
BPSTS(K,I,IL,3)=AS2*FAC/(ASTMWTNN(K)*NPLY(K))
GO TO 105
720 CONTINUE
IF(BPR.EQ.0.0) DSN=DABS(PLXPT(1))
IF(BPR.EQ.1.0) DSN=XSTR
PNT=DSN*PSTC(1,I,K)/DABS(AS1)
IF(AS1.LT.0.) PNT=DSN*PSTC(2,I,K)/DABS(AS1)
PBN=DSN*PSTC(3,I,K)/DABS(AS3)
PSH=DSN*PSTC(4,I,K)/DABS(AS2)
IF(BPR.EQ.0.0) WRITE(6,75) I,ANG(I,K),PNT,PBN,PSH
75 FORMAT(/,' FOR PLY TYPE NUMBER ',I5,' WITH ',//
' A PLY ORIENTATION OF ',D9.3,' DEGREES ',//
' NET SECTION FAILURE LOAD = ',D9.3,' LBS ',//
' BEARING FAILURE LOAD = ',D9.3,' LBS ',//
' SHEAROUT FAILURE LOAD = ',D9.3,' LBS ',//)
PNS(I,K)=PNT
PBR(I,K)=PBN
PSO(I,K)=PSH
105 CONTINUE
IF(BPR.EQ.0.0) GO TO 80
N=NUMPLY(K)

00017360
00017370
00017380
00017390
00017400
00017410
00017420
00017430
00017440
00017450
00017460
00017470
00017480
00017490
00017500
00017510
00017520
00017530
00017540
00017550
00017560
00017570
00017580
00017590
00017600
00017610
00017620
00017630
00017640
00017650
00017660
00017670
00017680
00017690
00017700
00017710
00017720
00017730
00017740
00017750
00017760
00017770
00017780
00017790
00017800
00017810
00017820
00017830
00017840
00017850
00017860
00017870
00017880
00017890
00017900
00017910
00017920
00017930
00017940
00017950

```

IF(BPR.EQ.1.0.AND.NOPT1.EQ.2) N=1
PFAIL1=1.0D10
PFAIL2=1.0D10
PFAIL3=1.0D10
DO 718 I=1,N
IF(PFAIL1.GT.PNS(I,K)) NPY1=I
IF(PFAIL1.GT.PNS(I,K)) PFAIL1=PNS(I,K)
IF(PFAIL2.GT.PBR(I,K)) NPY2=I
IF(PFAIL2.GT.PBR(I,K)) PFAIL2=PBR(I,K)
IF(PFAIL3.GT.PSO(I,K)) NPY3=I
718 IF(PFAIL3.GT.PSO(I,K)) PFAIL3=PSO(I,K)
IF(PFAIL1.GE.PFAIL2.OR.PFAIL1.GE.PFAIL3) GO TO 883
PFAIL1=PFAIL1*WTH*H(K)*NPLY(K)
WRITE(6,478) PFAIL1
WRITE(6,834)
GO TO 881
883 IF(PFAIL2.GE.PFAIL1.OR.PFAIL2.GE.PFAIL3) GO TO 882
PFAIL2=PFAIL2*WTH*H(K)*NPLY(K)
WRITE(6,473) PFAIL2
WRITE(6,835)
GO TO 881
882 IF(PFAIL3.GE.PFAIL1.OR.PFAIL3.GE.PFAIL2) GO TO 881
PFAIL3=PFAIL3*WTH*H(K)*NPLY(K)
WRITE(6,478) PFAIL3
WRITE(6,886)
881 CONTINUE
478 FORMAT(//,' FOR THE LAMINATE WITH THE OPEN HOLE, FAILURE',//,
* ' IS PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
884 FORMAT(' PREDICTED FAILURE MODE IS NET SECTION',//)
885 FORMAT(' PREDICTED FAILURE MODE IS BEARING FAILURE',//)
886 FORMAT(' PREDICTED FAILURE MODE IS SHEAR-OUT FAILURE',//)
GO TO 80

C
C
C
90 CONTINUE
PAPP=XSTR*H(K)*NPLY(K)*WTH
WRITE(6,772)
772 FORMAT(//,' MAXIMUM STRAIN CRITERION ',//)
L11=NOUT+1
L12=L11+NRC(K)
NCAS=2
NN=NUMPLY(K)
DO 210 I=1,NN
THETA=ANG(I,K)*RAD
CALL GMATX(K,L11,L12,NCAS,NOPT1,RAD,THETA)
STMAX=-1.0D10
N1=NRC(K)
DO 510 J=1,N1
IF(STMAX.LT.DABS(AEPS1(J))) LOC=J
510 IF(STMAX.LT.DABS(AEPS1(J))) STMAX=DABS(AEPS1(J))
THTA=ATETAA(NOUT+LOC)
IF(BPR.EQ.1.0) GO TO 511
PFL(I,K)=DABS(PLXPT(I)*SALOW(K)/STMAX)
WRITE(6,774) I,ANG(I,K),THTA,PFL(I,K)
774 FORMAT(' FOR PLY TYPE NUMBER ',I5,' WITH ',//,
* ' A PLY ORIENTATION OF ',D9.3,' DEGREES ',//,
* ' FAILURE IS PREDICTED AT ',D9.3,' DEGREES ',//,
* ' AT A PLY LOAD OF ',D9.3,' LBS',//)
GO TO 210

```

00017960
 00017970
 00017980
 00017990
 00018000
 00018010
 00018020
 00018030
 00018040
 00018050
 00018060
 00018070
 00018080
 00018090
 00018100
 00018110
 00018120
 00018130
 00018140
 00018150
 00018160
 00018170
 00018180
 00018190
 00018200
 00018210
 00018220
 00018230
 00018240
 00018250
 00018260
 00018270
 00018280
 00018290
 00018300
 00018310
 00018320
 00018330
 00018340
 00018350
 00018360
 00018370
 00018380
 00018390
 00018400
 00018410
 00018420
 00018430
 00018440
 00018450
 00018460
 00018470
 00018480
 00018490
 00018500
 00018510
 00018520
 00018530
 00018540
 00018550

```

511 PFL(I,K)=ABS(PAPP*SALOW(K)/STMAX)
210 CONTINUE
    IF(BPR.EQ.0.0) GO TO 80
    A=1.0D10
    NN=NUMPLY(K)
    DO 514 I=1,NN
      IF(A.GT.PFL(I,K)) NPY=I
514  IF(A.GT.PFL(I,K)) A=PFL(I,K)
      WRITE(6,778) A
778  FORMAT(//,' FOR THE OPEN HOLE LAMINATE, FAILURE IS',/,
* ' PREDICTED AT A JOINT LOAD OF ',D9.3,' LBS',//)
80  RETURN
    END

SUBROUTINE QMATX(K,LI1,LI2,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)
DIMENSION ATETAA(400),AEP SX(400),AEP SY(400),AEP SXY(400)
DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400)
DIMENSION AEP S1(400)
DIMENSION ASX(400),ASXY(400)
COMMON/XXY1/ASX,ASXY
COMMON/MOD/E11,E22,ESS,PMU12,PMU21
COMMON/STRSS2/AEP S1
COMMON/STRESS/ASIGR,ASIGRT,ASIG1,ASIG2,ASIG6
COMMON/QMT/ATETAA,AEP SX,AEP SY,AEP SXY
COMMON/PSC3/SX,SXY
J=0
Q11=E11(K)/(1.0-PMU12(K)*PMU21(K))
Q12=(PMU21(K)*E11(K))/(1.0-PMU12(K)*PMU21(K))
Q22=E22(K)/(1.0-PMU12(K)*PMU21(K))
Q66=ESS(K)
C=DCOS(THETA)
S=DSIN(THETA)
BQ11=(Q11*(C**4))+(2.*(Q12+(2.*Q66))*(C**2)*(S**2))+(Q22*(S**4))
BQ12=((Q11+Q22-(4.*Q66))*S*(C**2)*(C**2))+(Q12*(S**4+C**4))
BQ16=((Q11-Q12-(2.*Q66))*S*(C**3))+(Q12-Q22+(2.*Q66))*S*(S**3)*C
*)
BQ22=(Q11*(S**4))+(2.*(Q12+(2.*Q66))*(S**2)*(C**2))+(Q22*(C**4))
BQ26=((Q11-Q12-(2.*Q66))*C*(S**3))+(Q12-Q22+(2.*Q66))*S*(C**3))
BQ66=((Q11+Q22-(2.*(Q12+Q66)))*(S**2)*(C**2))+(Q66*(C**4)+(S**4))
*)
DO 40 I=LI1,LI2
  J=J+1
  IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD
  C=DCOS(THETA)
  S=DSIN(THETA)
  SIGX=BQ11*AEP SX(I)+BQ12*AEP SY(I)+BQ16*AEP SXY(I)
  SIGY=BQ12*AEP SX(I)+BQ22*AEP SY(I)+BQ26*AEP SXY(I)
  SIGXY=BQ16*AEP SX(I)+BQ26*AEP SY(I)+BQ66*AEP SXY(I)
  SX(J)=SIGX
  SXY(J)=SIGXY
  IF(NOPT1.EQ.2) SX(J)=ASX(I)
  IF(NOPT1.EQ.2) SXY(J)=ASXY(I)

```

00018560
 00018570
 00018580
 00018590
 00018600
 00018610
 00018620
 00018630
 00018640
 00018650
 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

```

ASIGR(I)=SIGX*CM2+SIGY*SM2+2.*SIGY*SMC      00019160
ASIGRT(I)=-SIGX*SMC+SIGY*CM2+SIGY*(CM2-SM2)  00019170
MSIG1(J)=SIGX*CM2+SIGY*SM2+2.*SMC*SIGY      00019180
ASIG2(J)=SIGX*SM2+SIGY*CM2-2.*SMC*SIGY      00019190
ASIG6(J)=-C*S*SIGX+SIGY*CM2+(CM2-SM2)*SIGY  00019200
AEPS1(J)=AEPSX(I)*CM2+AEPSY(I)*SM2+AEPSXY(I)*SMC 00019210
40 CONTINUE                                00019220
RETURN                                    00019230
END                                      00019240
00019250
00019260
00019270
SUBROUTINE HOFF(S1,S2,S6,A,B,K)              00019280
IMPLICIT REAL*8(A-H,O-Z)                   00019290
DIMENSION HPMC(5,2)                         00019300
COMMON/HFF/HPMC                             00019310
00019320
00019330
00019340
COMPUTE THE HOFFMAN/TSAL-HILL FAILURE INDEX  00019350
00019360
A=0.0D0                                     00019370
B=0.0D0                                     00019380
XC=HPMC(1,K)                               00019390
XT=HPMC(2,K)                               00019400
YC=HPMC(3,K)                               00019410
YT=HPMC(4,K)                               00019420
STC=HPMC(5,K)                              00019430
A=(S1*S2-S1*XT)/(XC*XT)+(S2*S2)/(YC*YT)+(S6/STC)*S2 00019440
IF(XC.EQ.XT.AND.YC.EQ.YT) GO TO 10         00019450
B=((XC-XT)/(XC*XT))*S1+((YC-YT)/(YC*YT))*S2 00019460
GO TO 20                                    00019470
10 CONTINUE                                00019480
B=0.0D0                                     00019490
20 CONTINUE                                00019500
RETURN                                    00019510
END                                      00019520
00019530
00019540
00019550
00019560
00019570
SUBROUTINE CENTD(RF,H,FASSS,FASBS,P,DELP,ITT) 00019580
00019590
00019600
IMPLICIT REAL*8(A-H,O-Z)                   00019610
DIMENSION PLYK(100),BARK(100),BARU(100),F(100) 00019620
DIMENSION H(2),RF(2)                       00019630
DIMENSION AII(100,100),A(2),B(2)           00019640
DIMENSION NPLY(2)                           00019650
COMMON/PBB/PLYK,BARK,BARU                  00019660
COMMON/AFM/AII,F                           00019670
COMMON/LYP/NPLY                             00019680
00019690
00019700
SET UP THE CENTRAL DIFFERENCE EQUATIONS    00019710
00019720
DO 3 I=1,100                               00019730
DO 3 J=1,100                               00019740
3 AII(I,J)=0.                               00019750
00019760
NECESSARY CONSTANTS ARE FORMED

```

```

C      DO 7 I=1,2
      A(I)=H(I)*H2/FASSS
7      B(I)=H(I)*H4/FASBS
      H2=H(1)/H(2)
      A1=H(1)*H2/FASSS
      A2=H(2)*H2/FASSS
      NP=NPLY(1)+NPLY(2)

CCCC      SHEAR AT TOP OF JOINT EQUALS ZERO
      AII(1,1)=1.
      AII(1,2)=-2.+A1*PLYK(2)
      AII(1,4)=2.+A1*PLYK(2)
      AII(1,5)=-1.
      F(1)=0.0

CCCC      MOMENT CONDITION AT TOP
      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*H2.*H(1)*FASSS)+R*(-2.-A1*PLYK(2)+(H(1)*H2
      *FASSS)/FASBS)
      AII(2,3)=-Z*(4.*H(1)*FASSS+(2*H(1)*H2*PLYK(1)*H(1)))
      AII(2,4)=Z*H2.*H(1)*FASSS+R*(2.+A1*PLYK(2)-(H(1)*H2
      *FASSS)/FASBS)
      AII(2,5)=-R
      F(2)=Z*H2.*H(1)*H3*BARK(1)*BARU(1)

CCCC      GOVERNING EQUATIONS FOR THE TOP PLATE
      H2=NPLY(1)
      DO 55 J=1,N2
      I=J+2
      AII(I,J)=1.
      IF(J.EQ.1) GO TO 56
      AII(I,J+1)=-4.-A(1)*PLYK(J-1)
      GO TO 57
56      AII(I,J+1)=-4.-A(1)*PLYK(2)
57      AII(I,J+2)=6.+(2.*A(1)+B(1))*PLYK(J)
      IF(J.EQ.N2) GO TO 61
      AII(I,J+3)=-4.-A(1)*PLYK(J+1)
      GO TO 62
61      AII(I,J+3)=-4.-A(1)*PLYK(NPLY(1)-1)
62      AII(I,J+4)=1.
      IF(J.EQ.1) GO TO 58
      IF(J.EQ.N2) GO TO 63
      F(1)=A(1)*BARK(J-1)*BARU(J-1)
      M=-(2.*A(1)+B(1))*BARK(J)*BARU(J)
      *A(1)*BARK(J+1)*BARU(J+1)
      GO TO 59
58      F(1)=2.*A(1)*BARK(2)*BARU(2)
      M=-(2.*A(1)+B(1))*BARK(1)*BARU(1)
      GO TO 59
  
```

00019750
 00019770
 00019780
 00019790
 00019800
 00019810
 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

```

63 F(I)=2.*A(1)*MBARK(NPLY(1)-1)*MBARU(NPLY(1)-1)
   *- (2.*A(1)+B(1))*BARK(J)*BARU(J)
59 CONTINUE
55 CONTINUE

C
C
C   INTERFACE SHEAR ON TOP PLATE = P+DELP
C
I=NPLY(1)+3
J=NPLY(1)
AII(I,J)=1.
AII(I,J+1)=- (2.+A1*PLYK(NPLY(1)-1))
AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)
AII(I,J+4)=-1.
F(I)=(-2.*H(1)*M3*(P+DELP))/FASBS

C
C
C   SLOPE CONTINUITY
C
I=NPLY(1)+4
J=NPLY(1)
AII(I,J)=1.
AII(I,J+1)=- (2.+A1*PLYK(NPLY(1)-1)-H(1)*M2*FASBS/FASBS)
AII(I,J+3)=2.+A1*PLYK(NPLY(1)-1)-H(1)*M2*FASBS/FASBS
AII(I,J+4)=-1.
AII(I,J+5)=-H12*M3
AII(I,J+6)=H12*M3*(2.+A2*PLYK(NPLY(1)+2)-H(2)*M2*FASBS/FASBS)
AII(I,J+8)=-H12*M3*(2.+A2*PLYK(NPLY(1)+2)-H(2)*M2*FASBS/FASBS)
AII(I,J+9)=H12*M3
F(I)=0.

C
C
C   MOMENT CONTINUITY
C
I=NPLY(1)+5
J=NPLY(1)+1
AII(I,J)=1.
AII(I,J+1)=- (2.+A1*PLYK(NPLY(1)))
AII(I,J+2)=1.
AII(I,J+5)=-H12*M2
AII(I,J+6)=H12*M2*(2.+A2*PLYK(NPLY(1)+1))
AII(I,J+7)=-H12*M2
F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*
*BARU(NPLY(1)+1))

C
C
C   INTERFACE SHEAR ON BOTTOM PLATE
C
I=NPLY(1)+6
J=NPLY(1)+5
AII(I,J)=-1.
AII(I,J+1)=(2.+A2*PLYK(NPLY(1)+2))
AII(I,J+3)=- (2.+A2*PLYK(NPLY(1)+2))
AII(I,J+4)=1.
F(I)=2.*H(2)*M3*(P+DELP)/FASBS

C
C
C   GOVERNING EQUATIONS FOR THE BOTTOM PLATE
C
N1=NPLY(1)+7
N2=NPLY(1)+NPLY(2)+6
DO 70 I=N1,N2
  J=I-2
  AII(I,J)=1.
  IF(I.EQ.N1) GO TO 71

```

00020360
00020370
00020380
00020390
00020400
00020410
00020420
00020430
00020440
00020450
00020460
00020470
00020480
00020490
00020500
00020510
00020520
00020530
00020540
00020550
00020560
00020570
00020580
00020590
00020600
00020610
00020620
00020630
00020640
00020650
00020660
00020670
00020680
00020690
00020700
00020710
00020720
00020730
00020740
00020750
00020760
00020770
00020780
00020790
00020800
00020810
00020820
00020830
00020840
00020850
00020860
00020870
00020880
00020890
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 74                                                         00021020
75 AII(I,J+3)=-4,-A(2)*PLYK(J-5)                                00021030
76 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
M=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)                          00021080
M+A(2)*BARK(J-3)*BARU(J-3)                                     00021090
GO TO 74                                                         00021100
73 F(I)=2.*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2)              00021110
M=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)                          00021120
GO TO 74                                                         00021130
77 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)                            00021140
M=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)                          00021150
74 CONTINUE                                                    00021160
70 CONTINUE                                                    00021170
000 SHEAR ON BOTTOM PLATE EQUALS ZERO                          00021180
NP=NPLY(1)+NPLY(2)                                             00021190
I=NP+7                                                         00021200
J=NP+4                                                         00021210
AII(I,J)=1.                                                    00021220
AII(I,J+1)=(2.+A2*PLYK(NP-1))                                  00021230
AII(I,J+3)=-2.+A2*PLYK(NP-1))                                  00021240
AII(I,J+4)=1.                                                  00021250
F(I)=0.                                                         00021260
000 MOMENT BOUNDARY CONDITION ON BOTTOM PLATE                  00021270
I=NP+8                                                         00021280
IF(RF(2).OE.1.D10) GO TO 85                                    00021290
Z=1.                                                           00021300
R=RF(2)                                                         00021310
GO TO 95                                                         00021320
85 Z=0.                                                         00021330
R=1.                                                           00021340
95 AII(I,J)=-R                                                  00021350
AII(I,J+1)=Z*(2.*H(2)*FASSS)+R*(2.+A2*PLYK(NP-1))            00021360
M=H(2)*M2*FASSS/FASBS)                                         00021370
AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)*M3*PLYK(NP))            00021380
AII(I,J+3)=Z*2.*H(2)*FASSS+R*(-2.-A2*PLYK(NP-1))             00021390
M+H(2)*M2*FASSS/FASBS)                                         00021400
AII(I,J+4)=R                                                   00021410
F(I)=Z*(2.*H(2)*M3*BARK(NP)*BARU(NP))                         00021420
RETURN                                                         00021430
END                                                            00021440
0000 SUBROUTINE SOLVE(U,H,P,DELP,NSDLS,ITT)                   00021450
00021460
00021470
00021480
00021490
00021500
00021510
00021520
00021530
00021540
00021550

```

```

      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
      SOLUTION OF THE SYSTEM: 9AI(U)=(B)                        00021630
      NP=NPLY(1)+NPLY(2)+8                                     00021640
      DO 444 I=1,NP                                             00021650
444    B(I)=F(I)                                                00021660
                                                                00021670
                                                                00021680
                                                                00021690
                                                                00021700
                                                                00021710
                                                                00021720
                                                                00021730
                                                                00021740
                                                                00021750
      APPLYING QUASSIAN ELIMINATION TO THE                     00021760
      MATRIX OF COEFFICIENTS                                   00021770
      DO 2001 I=1,NP                                           00021780
      IR=I                                                       00021790
2042  IF(A(IR,I).NE.0.) GO TO 2041                             00021800
      IR=IR+1                                                    00021810
      IF(IR.GT.NP) GO TO 2001                                   00021820
      GO TO 2042                                                 00021830
2041  NN=IR+1                                                    00021840
      DO 2002 L=NN,NP                                           00021850
      IF(DABS(A(L,I)).GT.1.D-30) GO TO 2009                     00021860
      A(L,I)=0.                                                  00021870
      GO TO 2002                                                 00021880
2009  CF=-A(IR,I)/A(L,I)                                         00021890
      DO 2003 J=I,NP                                           00021900
      A(L,J)=A(L,J)+CF*A(IR,J)                                   00021910
      IF(DABS(A(L,J)).LT.1.D-30) A(L,J)=0.0                   00021920
2003  CONTINUE                                                  00021930
      B(L)=B(L)+CF*B(I)                                         00021940
2002  CONTINUE                                                  00021950
2001  CONTINUE                                                  00021960
                                                                00021970
      BACK SUBSTITUTION                                         00021980
      DO 2011 I=1,NP                                           00021990
      L=NP+1-I                                                  00022000
      SUM=0.                                                     00022010
      IF(A(L,L).EQ.0.) GO TO 2112                               00022020
      N=L+1                                                      00022030
      IF(N.GT.NP) GO TO 2013                                     00022040
      DO 2013 J=N,NP                                           00022050
      SUM=SUM-A(L,J)*SX(J)                                       00022060
2013  CONTINUE                                                  00022070
      SX(L)=(B(L)+SUM)/A(L,L)                                    00022080
      GO TO 2011                                                 00022090
2112  CONTINUE                                                  00022100
      SX(L)=0.                                                  00022110
2011  CONTINUE                                                  00022120
                                                                00022130
      EQUILIBRIUM CHECK                                         00022140
                                                                00022150
      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)*MH(1)
1444 CONTINUE
DO 1555 I=N2,NN
J=I-6
U(J)=SX(I)
SUM5=SUM5+SX(I)*PLYK(J)*MH(2)
1555 CONTINUE
IF(NSDLS.EQ.1) GO TO 810
PT=PT*2.
SUM4=SUM4*2.
SUM5=SUM5*2.
810 CONTINUE
NP=NPLY(1)+NPLY(2)
N=NPLY(1)+NPLY(2)
II=1
DO 311 I=1,N
IF(I.GT.NPLY(1)) II=2
PL=U(I)*PLYK(II)*MH(II)
IF(I.LE.NPLY(1)) GO TO 311
311 CONTINUE
RETURN
END

C
C
SUBROUTINE FAIL(GAMD, U, H, P, DELP, BPR, AST, WTH, PFAIL, ANGLE, NODE,
C
C
C
MIROUT, NOPT4, NULYF, JNT, ITT, NTFL)
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), GAMD(2), GAMN(100)
DIMENSION DELNS(5,2), DELBR(5,2), DELSO(5,2)
DIMENSION UN(100), PFL(5,2), PSTC(5,5,2)
DIMENSION IPLY(100,2), ANG(5,2), NUMPLY(2)
DIMENSION PNS(5,2), PBR(5,2), PSO(5,2), PALT(3,2)
DIMENSION BPSTS(2,10,2,3)
DIMENSION NPNM(100,2)
COMMON/COUNT/NPNM
COMMON/BP1/BPSTS
COMMON/PSC2/PSTC
COMMON/FAL1/PNS, PBR, PSO, PALT
COMMON/FAL3/DELNS, DELBR, DELSO
COMMON/FAL4/UN, GAMN, MDAMP, MDAMI, PN
COMMON/FAL5/PFL
COMMON/PBB/PLYK, BARK, BARU
COMMON/PRT/NDAM, INPLY, ITYP
COMMON/LYP/NPLY, HUMPLY, ANG, IPLY
NP=NPLY(1)+NPLY(2)

```

000000

	MODEF=MODE	00023360
	NCC=1	00023370
	GO TO 140	00023380
200	NMN=1	00023390
	IF(I.GT.NPLY(1)) NMN=I-NPLY(1)	00023400
	NX=IPLY(NMN,K)	00023410
	IF(PBR(NX,K).LT.PSO(NX,K).OR.PNS(NX,K).LT.PSO(NX,K)) GO TO 700	00023420
	MODE=1	00023430
	PF=PSO(NX,K)	00023440
	IF(MDAMP(I).EQ.1) MODE=5	00023450
	IF(MDAMP(I).EQ.1) PF=PALT(1,K)*PF	00023460
	GO TO 25	00023470
700	IF(PNS(NX,K).LT.PBR(NX,K)) GO TO 710	00023480
	MODE=2	00023490
	PF=PBR(NX,K)	00023500
	IF(MDAMP(I).EQ.2) MODE=6	00023510
	IF(MDAMP(I).EQ.2) PF=PALT(2,K)*PF	00023520
	GO TO 25	00023530
710	MODE=3	00023540
	PF=PNS(NX,K)	00023550
	IF(MDAMP(I).EQ.3) MODE=7	00023560
	IF(MDAMP(I).EQ.3) PF=PALT(3,K)*PF	00023570
25	CONTINUE	00023580
	NCC=0	00023590
	IF(DELPH.EQ.0) GO TO 210	00023600
	IF(DABS(PL).LT.DABS(PF)) GO TO 210	00023610
	PFP=0.	00023620
	IIMPLY=1	00023630
	DEF=MODE	00023640
	NCC=1	00023650
	GO TO 1212	00023660
210	CONTINUE	00023670
C		00023680
C	DETERMINE INCREMENTAL JOINT LOAD TO CAUSE	00023690
C	PLY FAILURE	00023700
	IF(ITT.LE.1) GO TO 21	00023710
	IF(DABS(DABS(U(I)/UN(I))-1.).LT.1.0D-10) GO TO 20	00023720
21	CONTINUE	00023730
	DELPH=(PF-DABS(PN(I)))*1000./((DABS(PL)-DABS(PN(I)))	00023740
C		00023750
C	A NEGATIVE VALUE OF DELPH INDICATES UNLOADING	00023760
C	IN A PLY. THIS NODE IS THEN SKIPPED	00023770
	IF(DELPH.LT.0.) GO TO 20	00023780
C		00023790
C	RECORD LOWEST JOINT FAILURE LOAD INCREMENT,	00023800
C	PLY IN WHICH IT OCCURS, AND MODE	00023810
	PFP2=PFP	00023820
	IF(DELPH.EQ.0) PFP2=1.	00023830
	IF(DELPH.GT.PFP2) GO TO 20	00023840
	PFP=DELPH	00023850
	IIMPLY=1	00023860
	MODEF=MODE	00023870
20	CONTINUE	00023880
C		00023890
C	LOOP OVER ALL INTERFACES TO FIND LOAD	00023900
C	AND LOCATION OF NEXT DELAMINATION	00023910
		00023920
		00023930
		00023940
		00023950

	NN=NPLY(1),NPLY(2)-2	00023960
	DO 30 J=1,NN	00023970
G	IF INTERFACE HAS ALREADY FAILED, GO TO NEXT	00023980
G	IF(MDAMI(J).EQ.1) GO TO 50	00023990
G	DETERMINE WHICH PLATE INTERFACE IS IN	00024000
G	K=1	00024010
G	IF(J.GE.NPLY(1)) K=2	00024020
G	CALCULATE INTERFACE SHEAR STRAIN FOR CURRENT	00024030
G	JOINT LOAD	00024040
G	GAMJ=(U(J+K-1)-U(J+K))/H(K)	00024050
G	DETERMINE INCREMENTAL JOINT LOAD TO CAUSE	00024060
G	INTERFACE FAILURE	00024070
G	IF(ITT.EQ.1) GO TO 47	00024080
G	IF(DABS(DABS(GAMJ/GAMN(J))-1.).LT.1.0E-10) GO TO 50	00024090
G	CONTINUE	00024100
G	DELPP=(GAMD(K)-DABS(GAMN(J)))/(DABS(GAMJ)-DABS(GAMN(J)))*1000.	00024110
G	IF(DELPP.LT.0.) GO TO 50	00024120
G	RECORD LOWEST JOINT FAILURE LOAD INCREMENT,	00024130
G	PLY OF INTERFACE IN WHICH IT OCCURS, AND MODE	00024140
G	PPF2=PPF	00024150
G	IF(DELPP.EQ.0) PPF2=1.	00024160
G	IF(DELPP.GT.PPF2) GO TO 50	00024170
G	PPF=DELPP	00024180
G	INPLY=J	00024190
G	MODEF=9	00024200
G	50 CONTINUE	00024210
G	DETERMINE VALUES AT END OF INCREMENT	00024220
G	JOINT LOAD AT FAILURE	00024230
G	IF(MODEF.EQ.0) GO TO 325	00024240
G	P=P+PPF*DELP/1000.	00024250
G	325 CONTINUE	00024260
G	NODAL DISPLACEMENTS AND PLY LOADS	00024270
G	NN=NPLY(1)+NPLY(2)	00024280
G	DO 55 I=1,NN	00024290
G	UN(I)=UN(I)+(U(I)-UN(I))*PPF/1000.	00024300
G	UPDATE UN	00024310
G	IF(NCC.EQ.1) UN(I)=U(I)	00024320
G	K=1	00024330
G	IF(I.GT.NPLY(1)) K=2	00024340
G	PN(I)=-H(K)*PLYK(I)*UN(I)+BARK(I)*BARU(I)	00024350
G	55 CONTINUE	00024360
G		00024370
G		00024380
G		00024390
G		00024400
G		00024410
G		00024420
G		00024430
G		00024440
G		00024450
G		00024460
G		00024470
G		00024480
G		00024490
G		00024500
G		00024510
G		00024520
G		00024530
G		00024540
G		00024550

C	INTERFACE SHEAR STRAINS	00024560
C		00024570
	NH=NPLY(1)+NPLY(2)-2	00024580
	DO 60 J=1,NH	00024590
	K=1	00024600
	IF(J.GE.NPLY(1)) K=2	00024610
	GAMN(J)=(UN(J+K-1)-UN(J+K))/H(K)	00024620
60	CONTINUE	00024630
1212	CONTINUE	00024640
C		00024650
C	PLY STIFFNESSES, DAMAGE STATES, AND NEXT LOAD	00024660
C	INCREMENT	00024670
		00024680
	K=1	00024690
	IF(INPLY.GT.NPLY(1)) K=2	00024700
	NMN=INPLY	00024710
	IF(INPLY.GT.NPLY(1)) NMN=INPLY-NPLY(1)	00024720
	NX=INPLY(NMN,K)	00024730
	IF(MODEF.NE.0) GO TO 70	00024740
	DELP=1000.	00024750
	NDAM=1	00024760
	GO TO 65	00024770
70	IF(MODEF.NE.1) GO TO 80	00024780
	IF(INPLY.EQ.1.AND.MDAMI(INPLY).EQ.1) GO TO 75	00024790
	IF(INPLY.EQ.NPLY(1).AND.MDAMI(INPLY-1).EQ.1) GO TO 75	00024800
	IF(INPLY.EQ.(NPLY(1)+1).AND.MDAMI(INPLY-1).EQ.1) GO TO 75	00024810
	IF(INPLY.EQ.(NPLY(1)+NPLY(2)).AND.MDAMI(NPLY(1)+NPLY(2)-1-2).EQ.1) GO TO 75	00024820
	KK=0	00024830
	IF(INPLY.GT.NPLY(1)) KK=1	00024840
	IF(MDAMI(INPLY-KK-1).EQ.1.AND.MDAMI(INPLY-KK).EQ.1) GO TO 75	00024850
	MDAMP(INPLY)=1	00024860
	TEMPK=PLYK(INPLY)	00024870
	PLYK(INPLY)=DELSO(NX,K)*TEMPK	00024880
	BARK(INPLY)=(1.-DELSO(NX,K))*TEMPK	00024890
	BARU(INPLY)=UN(INPLY)	00024900
	DELP=0.	00024910
	NDAM=2	00024920
	GO TO 65	00024930
75	PLYK(INPLY)=0.0	00024940
	BARK(INPLY)=0.	00024950
	BARU(INPLY)=UN(INPLY)	00024960
	MDAMP(INPLY)=10	00024970
	DELP=0.	00024980
	NDAM=2	00024990
	GO TO 65	00025000
80	IF(MODEF.NE.2) GO TO 85	00025010
	TEMPK=PLYK(INPLY)	00025020
	PLYK(INPLY)=DELS(NX,K)*TEMPK	00025030
	BARK(INPLY)=(1.-DELS(NX,K))*TEMPK	00025040
	BARU(INPLY)=UN(INPLY)	00025050
	MDAMP(INPLY)=2	00025060
	DELP=0.	00025070
	NDAM=4	00025080
	GO TO 65	00025090
85	IF(MODEF.NE.3) GO TO 90	00025100
	TEMPK=PLYK(INPLY)	00025110
	PLYK(INPLY)=DELSR(NX,K)*TEMPK	00025120
	BARK(INPLY)=(1.-DELSR(NX,K))*TEMPK	00025130
	BARU(INPLY)=UN(INPLY)	00025140
		00025150

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
MDAMI(INPLY)=1	00025620
NDAM=10	00025630
GO TO 65	00025640
600 CONTINUE	00025650
PARTIAL BEARING FAILURE ANALYSIS	00025660
IROUT=2	00025670
NPL=NPLY(1)+NPLY(2)	00025680
AJFLNS=1.0D10	00025690
AJFLBR=1.0D10	00025700
AJFLSD=1.0D10	00025710
DO 550 I=1,NPL	00025720
K=1	00025730
	00025740
	00025750

```

IF(I.GT.NPLY(1)) K=2                                00025760
PL=-H(K)*PLY(K)*MU(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)*ASTM*WTH*H(K)*NPLY(K)/1000.            00025800
IJ=I                                                  00025810
IF(I.GT.NPLY(1)) IJ=IJ-NPLY(1)                      00025820
IP=IPLY(IJ,K)                                        00025830
NT=I                                                  00025840
IF(BPSTS(K,IP,2,1).LT.0.0) NT=2                    00025850
F1=DABS(PSTC(NT,IP,K)/(BPSTS(K,IP,1,1)+PLMBPSTS(K,IP,2,1))) 00025860
F2=DABS(PSTC(3,IP,K)/(BPSTS(K,IP,1,2)+PLMBPSTS(K,IP,2,2))) 00025870
F3=DABS(PSTC(4,IP,K)/(BPSTS(K,IP,1,3)+PLMBPSTS(K,IP,2,3))) 00025880
IF(MDAMP(I).EQ.5) 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.OT.F1) NF1=I                              00025920
IF(AJFLNS.OT.F1) AJFLNS=F1                          00025930
IF(AJFLBR.OT.F2) NF2=I                              00025940
IF(AJFLBR.OT.F2) AJFLBR=F2                          00025950
IF(AJFLSO.OT.F3) NF3=I                              00025960
IF(AJFLSO.OT.F3) AJFLSO=F3                          00025970
550 CONTINUE                                         00025980
IF(AJFLNS.OT.AJFLBR.OR.AJFLNS.OT.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.OT.AJFLNS.OR.AJFLBR.OT.AJFLSO) GO TO 571 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.OT.AJFLNS.OR.AJFLSO.OT.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.OT.NPLY(1)) K=2                            00026270
IPL=INPLY                                            00026280
IF(IPL.OT.NPLY(1)) IPL=IPL-NPLY(1)                  00026290
IPLP=IPLY(IPL,K)                                    00026300
ANGLE=ANG(IPLP,K)                                   00026310
NODE=NPHM(IPL,K)                                    00026320
IF(MDAMP(INPLY).EQ.6) GO TO 107                     00026330
IF(MDAMP(INPLY).EQ.5) AR=DELNS(IPLP,K)              00026340
IF(MDAMP(INPLY).EQ.4) AR=DELBK(IPLP,K)              00026350

```

```

IF(MDAMP(INPLY).EQ.2) AR=DELSO(IPLP,K) 00026360
TEMPK=PLYK(INPLY) 00026370
PLYK(INPLY)=AR*TEMPK 00026380
BARK(INPLY)=(1.-AR)*TEMPK 00026390
BARU(INPLY)=U(INPLY) 00026400
ITYP=IPLY(IPL,K) 00026410
NTFL=0 00026420
GO TO 103 00026430
107 CONTINUE 00026440
IF(K.EQ.1) NPLY(1)=NPLY(1)-1 00026450
IF(K.EQ.2) NPLY(2)=NPLY(2)-1 00026460
NP=INPLY 00026470
IF(K.EQ.2) NP=INPLY-NPLY(1) 00026480
N=NPLY(K)-NP+2 00026490
ITYP=IPLY(NP,K) 00026500
DO 101 I=1,N 00026510
IPLY(NP+I-1,K)=IPLY(NP+I,K) 00026520
NPNM(NP+I-1,K)=NPNM(NP+I,K) 00026530
101 CONTINUE 00026540
N=NPL-INPLY 00026550
DO 102 I=1,N 00026560
MDAMP(INPLY+I-1)=MDAMP(INPLY+I) 00026570
PLYK(INPLY+I-1)=PLYK(INPLY+I) 00026580
BARK(INPLY+I-1)=BARK(INPLY+I) 00026590
102 BARU(INPLY+I-1)=BARU(INPLY+I) 00026600
NTFL=1 00026610
NULTF=NULTF-1 00026620
IF(NULTF.EQ.0) JNT=0 00026630
IF(NPLY(1).EQ.2.OR.NPLY(2).EQ.2) JNT=0 00026640
103 CONTINUE 00026650
RETURN 00026660
65 CONTINUE 00026670
C C INCREMENT LOAD IF JOINT HAS NOT FAILED 00026680
C C 00026690
T1=0. 00026700
T2=0. 00026710
N1=NPLY(1) 00026720
N2=NPLY(2) 00026730
DO 135 I=1,N1 00026740
135 T1=T1+PLYK(I) 00026750
DO 126 I=1,N2 00026760
N3=NPLY(1)+I 00026770
126 T2=T2+PLYK(N3) 00026780
IF(T1.EQ.0.0.OR.T2.EQ.0.0) GO TO 130 00026790
RETURN 00026800
130 JNT=0 00026810
RETURN 00026820
END 00026830
C C 00026840
C C 00026850
C C 00026860
SUBROUTINE PRINT(U,P,DELP,PFAIL,ANGLE,BPR,NODE,IROUT,JNT, 00026870
MNP,NSDLS,ITY) 00026880
00026890
00026900
00026910
IMPLICIT REAL*8(A-H,O-Z) 00026920
DIMENSION U(100),PLYK(100) 00026930
DIMENSION NPLY(2),NUNPLY(2),ANG(5,2),IPLY(100,2) 00026940
DIMENSION BARK(100),BARU(100) 00026950

```

```

DIMENSION NPNM(100.2)
COMMON/COUNT/NPNM
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
COMMON/PBB/PLYK,DARK,BARU
COMMON/PRT/NDAM,INPLY,ITYP

PRINT VALUES AT END OF INCREMENT

IF(ITT.EQ.1) WRITE(6,10)
10 FORMAT(//,10X,'FAILURE MODE ABBREVIATIONS'
*10X,'ND' = NO ADDITIONAL DAMAGE AT CURRENT JOINT LOAD'
*10X,'DL' = DELAMINATION
*10X,'SO' = SHEAR-OUT
*10X,'BR' = BEARING
*10X,'NS' = NET SECTION
*10X,'SUD' = ULTIMATE FAILURE AFTER SO AND DL
*10X,'SU' = ULTIMATE FAILURE IN SO
*10X,'BU' = ULTIMATE FAILURE IN BR
*10X,'NSU' = ULTIMATE FAILURE IN NS
*10X,'ULT' = ULTIMATE FAILURE
*4X,'INCREMENT NO',3X,'JOINT LOAD',5X,'NODE',8X,'PLY TYPE',
*8X,'MODE',/)
PL=P
IF(ROUT.EQ.2) PL=PFail
IF(NSDLS.EQ.2) PL=2.*PL
IF(ITT.EQ.1) PFAILP=0.0D0
IF(PFAILP.LT.PL) PFAILP=PL
IF(JNT.EQ.0.AND.PFAILP.EQ.0.0D0) PFAILP=PL
K=1
IF(INPLY.GT.NPLY(1)) K=2
N=IPLY(INPLY,K)
IF(K.EQ.2) N=IPLY((INPLY-NPLY(1)),K)
IF(ROUT.EQ.2) N=ITYP
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) ANGLE=ANG(N,K)
IF(BPR.EQ.0.0.OR.BPR.EQ.1.0) NODE=INPLY
IF(NDAM.EQ.1) WRITE(6,20) ITT,PL
IF(NDAM.EQ.2) WRITE(6,30) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.3) WRITE(6,40) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.4) WRITE(6,50) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.5) WRITE(6,60) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.6) WRITE(6,70) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.7) WRITE(6,80) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.8) WRITE(6,90) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.9) WRITE(6,100) ITT,PL,NODE,ANGLE
IF(NDAM.EQ.10) WRITE(6,110) ITT,PL,NODE,ANGLE
20 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' SO')
30 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' SUD')
40 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' BR')
50 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' NS')
60 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' SU')
70 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' BU')
80 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' NSU')
90 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' ULT')
110 FORMAT(5X,15,10X,D9.3,3X,15,5X,F7.3,' DEGREE ',5X,' DL')
IF(JNT.EQ.0) GO TO 220
GO TO 250
220 CONTINUE
WRITE(6,240) PFAILP
240 FORMAT(//,' THE PREDICTED JOINT FAILURE ',/,

```

00026960
 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

```

*1 LOAD IS 'D14.7' LBS',//)
250 CONTINUE
RETURN
END
C
C
C
SUBROUTINE LINV2F (A,N,IA,AINV,IDGT,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   DO 10 I = 1,N
C     DO 5 J = 1,N
C       AINV(I,J) = ZERO
C     CONTINUE
C   10 CONTINUE
C                               COMPUTE THE INVERSE OF A
C   CALL LEQ2F (A,N,N,IA,AINV,IDGT,WKAREA,IER)
C   IF (IER.EQ.0) GO TO 9005
C   9000 CONTINUE
C   CALL UERTST (IER,6HLINV2F)
C   9005 RETURN
C   END
C
C
C   SUBROUTINE LEQ2F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
C   DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
C   DOUBLE PRECISION  A,B,WKAREA,D1,D2,WA
C                               FIRST EXECUTABLE STATEMENT
C                               INITIALIZE IER
C   IER=0
C   JER=0
C   J = N*N+1
C   K = J+N
C   MM = K+N
C   KK = 0
C   MM1 = MM-1
C   JJ=1
C   DO 5 L=1,N
C     DO 5 I=1,N
C       WKAREA(JJ)=A(I,L)
C       JJ=JJ+1
C     CONTINUE
C   DECOMPOSE A
C   CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
C     * WA,IER)
C   IF (IER.GT.128) GO TO 25
C   IF (IDGT.EQ.0 .OR. IER.NE.0) KK = 1
C   DO 15 I = 1,M
C     PERFORMS THE ELIMINATION PART OF
C     AX = B
C     CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))
C     REFINEMENT OF SOLUTION TO AX = B

```

	IF (KK .NE. 0)	00028160
*	CALL LUREFN (WKAREA,N,N,A,IA,B(1,1),IDOT,WKAREA(J),WKAREA(MM),	00028170
*	WKAREA(K),WKAREA(K),JER).	00028180
	DO 10 II=1,N	00028190
	B(II,I) = WKAREA(MM+II)	00028200
10	CONTINUE	00028210
	IF (JER.NE.0) GO TO 20	00028220
15	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
	A(I,J)=WKAREA(JJ)	00028290
	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
		00028370
		00028380
		00028390
	SUBROUTINE LUDATF (A,LU,N,IA,IDOT,D1,D2,IPVT,EQUIL,WA,IER)	00028400
		00028410
	DIMENSION A(IA,1),LU(IA,1),IPVT(1),EQUIL(1)	00028420
	DOUBLE PRECISION A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTH,SIXTH,	00028430
*	RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q	00028440
	DATA ZERO,ONE,FOUR,SIXTH,SIXTH/0.D0,1.D0,4.D0,	00028450
*	16.D0,.0625D0/	00028460
	FIRST EXECUTABLE STATEMENT	00028470
	INITIALIZATION	00028480
	IER = 0	00028490
	RN = N	00028500
	WREL = ZERO	00028510
	D1 = ONE	00028520
	D2 = ZERO	00028530
	BIGA = ZERO	00028540
	DO 10 I=1,N	00028550
	BIG = ZERO	00028560
	DO 5 J=1,N	00028570
	P = A(I,J)	00028580
	LU(I,J) = P	00028590
	P = DABS(P)	00028600
	IF (P .GT. BIG) BIG = P	00028610
5	CONTINUE	00028620
	IF (BIG .GT. BIGA) BIGA = BIG	00028630
	IF (BIG .EQ. ZERO) GO TO 110	00028640
	EQUIL(I) = ONE/BIG	00028650
10	CONTINUE	00028660
	DO 105 J=1,N	00028670
	JM1 = J-1	00028680
	IF (JM1 .LT. 1) GO TO 40	00028690
	COMPUTE U(I,J), I=1,...,J-1	00028700
	DO 35 I=1,JM1	00028710
	SUM = LU(I,J)	00028720
	IM1 = I-1	00028730
	IF (IDOT .EQ. 0) GO TO 25	00028740
	WITH ACCURACY TEST	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)                         00028800
      SUM = SUM-T                                   00028810
      WI = WI+DABS(T)                              00028820
15    CONTINUE                                     00028830
      LU(I,J) = SUM                                00028840
20    WI = WI+DABS(SUM)                            00028850
      IF (AI .EQ. ZERO) AI = BIGA                  00028860
      TEST = WI/AI                                 00028870
      IF (TEST .GT. WREL) WREL = TEST              00028880
      GO TO 35                                      00028890
C    WITHOUT ACCURACY                             00028900
25    IF (IM1 .LT. 1) GO TO 35                     00028910
      DO 30 K=1,IM1                                00028920
        SUM = SUM-LU(I,K)*LU(K,J)                 00028930
30    CONTINUE                                     00028940
      LU(I,J) = SUM                                00028950
35    CONTINUE                                     00028960
40    P = ZERO                                      00028970
C    COMPUTE U(J,J) AND L(I,J), I=J+1,...,        00028980
    DO 70 I=J,N                                    00028990
      SUM = LU(I,J)                                00029000
      IF (IDOT .EQ. 0) GO TO 55                     00029010
C    WITH ACCURACY TEST                           00029020
      AI = DABS(SUM)                                00029030
      WI = ZERO                                      00029040
      IF (JM1 .LT. 1) GO TO 50                      00029050
      DO 45 K=1,JM1                                 00029060
        T = LU(I,K)*LU(K,J)                         00029070
        SUM = SUM-T                                   00029080
        WI = WI+DABS(T)                              00029090
45    CONTINUE                                     00029100
      LU(I,J) = SUM                                00029110
50    WI = WI+DABS(SUM)                            00029120
      IF (AI .EQ. ZERO) AI = BIGA                  00029130
      TEST = WI/AI                                 00029140
      IF (TEST .GT. WREL) WREL = TEST              00029150
      GO TO 65                                      00029160
C    WITHOUT ACCURACY TEST                         00029170
55    IF (JM1 .LT. 1) GO TO 65                     00029180
      DO 60 K=1,JM1                                00029190
        SUM = SUM-LU(I,K)*LU(K,J)                 00029200
60    CONTINUE                                     00029210
      LU(I,J) = SUM                                00029220
65    Q = EQUIL(I)*DABS(SUM)                       00029230
      IF (P .GE. Q) GO TO 70                       00029240
      P = Q                                          00029250
      IMAX = I                                      00029260
70    CONTINUE                                     00029270
C    TEST FOR ALGORITHMIC SINGULARITY              00029280
      IF (RN+P .EQ. RN) GO TO 110                 00029290
      IF (J .EQ. IMAX) GO TO 80                   00029300
C    INTERCHANGE ROWS J AND IMAX                  00029310
      D1 = -D1                                      00029320
      DO 75 K=1,N                                  00029330
        P = LU(IMAX,K)                             00029340
        LU(IMAX,K) = LU(J,K)                      00029350

```

	LU(J,K) = P	00029360
75	CONTINUE	00029370
	EQUIL(IMAX) = EQUIL(J)	00029380
80	IPVT(J) = IMAX	00029390
	D1 = DIMLU(J,J)	00029400
85	IF (DABS(D1) .LE. ONE) GO TO 90	00029410
	D1 = DIMSIXTH	00029420
	D2 = D2+FOUR	00029430
	GO TO 85	00029440
90	IF (DABS(D1) .GE. SIXTH) GO TO 95	00029450
	D1 = DIMSIXTH	00029460
	D2 = D2-FOUR	00029470
	GO TO 90	00029480
95	CONTINUE	00029490
	JPI = J+1	00029500
	IF (JPI .GT. N) GO TO 105	00029510
C		00029520
	P = LU(J,J)	00029530
	DO 100 I=JPI,N	00029540
	LU(I,J) = LU(I,J)/P	00029550
100	CONTINUE	00029560
105	CONTINUE	00029570
C		00029580
	IF (IDOT .EQ. 0) GO TO 9005	00029590
	P = 3*H+3	00029600
	WA = P*WREL	00029610
	IF (WA+10.D0*H*(-IDOT) .NE. WA) GO TO 9005	00029620
	IER = 34	00029630
	GO TO 9000	00029640
C		00029650
	ALGORITHMIC SINGULARITY	00029660
110	IER = 129	00029670
	D1 = ZERO	00029680
	D2 = ZERO	00029690
9000	CONTINUE	00029700
C		00029710
	PRINT ERROR	00029720
9005	CALL UERTST(IER,6HLDATF)	00029730
	RETURN	00029740
	END	00029750
C		00029760
	SUBROUTINE LUELMN (A,IA,N,B,APVT,X)	00029770
C		00029780
	DIMENSION A(IA,1),B(1),APVT(1),X(1)	00029790
	DOUBLE PRECISION A,B,X,SUM,APVT	00029800
C		00029810
	FIRST EXECUTABLE STATEMENT	00029820
C		00029830
	SOLVE LY = B FOR Y	00029840
	DO 5 I=1,N	00029850
5	X(I) = B(I)	00029860
	IW = 0	00029870
	DO 20 I=1,N	00029880
	IP = APVT(I)	00029890
	SUM = X(IP)	00029900
	X(IP) = X(I)	00029910
	IF (IW .EQ. 0) GO TO 15	00029920
	IM1 = I-1	00029930
	DO 10 J=IW,IM1	00029940
	SUM = SUM-A(I,J)*X(J)	00029950
10	CONTINUE	
	GO TO 20	
15	IF (SUM .NE. 0.D0) IW = I	

<pre> C 20 X(I) = SUM DO 30 I=1,N I = N+1-IB IP1 = I+1 SUM = X(I) IF (IP1 .GT. N) GO TO 30 DO 25 J=IP1,N SUM = SUM-A(I,J)*X(J) 25 CONTINUE 30 X(I) = SUM/A(I,I) RETURN END C C SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDOT,APVT,X,RES,DX,IER) C C DIMENSION A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1) C DIMENSION APVT(1) C DIMENSION ACCXT(2) C DOUBLE PRECISION A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT C DATA ITMAX/75/,ZERO/0.00/ C C IER=0 C XNORM = ZERO C DO 10 I=1,N C XNORM = DMAX1(XNORM,DABS(X(I))) 10 CONTINUE C IF (XNORM .NE. ZERO) GO TO 20 C IDOT = 50 C GO TO 9005 20 DO 45 ITER=1,ITMAX C DO 30 I=1,N C ACCXT(1) = 0.000 C ACCXT(2) = 0.000 C CALL VXADD(B(I),ACCXT) C DO 25 J=1,N C CALL VXMUL(-A(I,J),X(J),ACCXT) 25 CONTINUE C CALL VXSTO(ACCXT,RES(I)) 30 CONTINUE C CALL LUELMN (UL,IUL,N,RES,APVT,DX) C DXNORM = ZERO C XNORM = ZERO C DO 35 I=1,N C X(I) = X(I) + DX(I) C DXNORM = DMAX1(DXNORM,DABS(DX(I))) C XNORM = DMAX1(XNORM,DABS(X(I))) 35 CONTINUE C IF (ITER .NE. 1) GO TO 40 C IDOT = 50 C IF (DXNORM .NE. ZERO) IDOT = -DLOG10(DXNORM/XNORM) 40 IF (XNORM+DXNORM .EQ. XNORM) GO TO 9005 45 CONTINUE C C ITERATION DID NOT CONVERGE C C IER = 129 9000 CONTINUE C CALL UERTST(IER,6HLUREFN) 9005 RETURN END </pre>	<pre> 00029763 00029970 00029980 00029990 00030000 00030010 00030020 00030030 00030040 00030050 00030060 00030070 00030080 00030090 00030100 00030110 00030120 00030130 00030140 00030150 00030160 00030170 00030180 00030190 00030200 00030210 00030220 00030230 00030240 00030250 00030260 00030270 00030280 00030290 00030300 00030310 00030320 00030330 00030340 00030350 00030360 00030370 00030380 00030390 00030400 00030410 00030420 00030430 00030440 00030450 00030460 00030470 00030480 00030490 00030500 00030510 00030520 00030530 00030540 00030550 </pre>
--	--

```

C      00030560
C      00030570
C      00030580
C      00030590
C      00030600
C      00030610
C      00030620
C      00030630
C      00030640
C      00030650
C      00030660
C      00030670
C      00030680
C      00030690
C      00030700
C      00030710
C      00030720
C      00030730
C      00030740
C      00030750
C      00030760
C      00030770
C      00030780
C      00030790
C      00030800
C      00030810
C      00030820
C      00030830
C      00030840
C      00030850
C      00030860
C      00030870
C      00030880
C      00030890
C      00030900
C      00030910
C      00030920
C      00030930
C      00030940
C      00030950
C      00030960
C      00030970
C      00030980
C      00030990
C      00031000
C      00031010
C      00031020
C      00031030
C      00031040
C      00031050
C      00031060
C      00031070
C      00031080
C      00031090
C      00031100
C      00031110
C      00031120
C      00031130
C      00031140
C      00031150

SUBROUTINE UERTST (IER,NAME)
      SPECIFICATIONS FOR ARGUMENTS
      INTEGER      IER
      INTEGER      NAME(1)
      SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER      I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(4),
      NAMSET(6),NAMUPK(6),NIN,NMT9
      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,NMT9)
      GET OUTPUT UNIT NUMBER
      CALL UGETIO(1,NIN,IOUNIT)
      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
      PRINT TERMINAL MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
      GO TO 30
      5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
      PRINT WARNING WITH FIX MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
      GO TO 30
      10 IF (IER.LE.32) GO TO 15
      PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
      GO TO 30
      15 CONTINUE
      CHECK FOR UERSET CALL
      DO 20 I=1,6
      IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
      20 CONTINUE
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
      25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
      PRINT NON-DEFINED MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK
      30 IEQDF = 0
      RETURN
      35 FORMAT(19H *** TERMINAL ERROR,10X,7H( IER = ,I3,
      1      20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
      40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H( IER = ,I3,

```

```

1          20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,IS,
1          20H) FROM IMSL ROUTINE ,6A1,A1,6A1)
50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,IS,
1          20H) FROM IMSL ROUTINE ,6A1,A1,6A1)

C
C
C
C
C          SAVE P FOR P = R CASE
C          P IS THE PAGE NAMUPK
C          R IS THE ROUTINE NAMUPK
55 IEQDF = 1
DO 60 I=1,6
60 NAMEQ(I) = NAMUPK(I)
65 RETURN
END

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

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

```

00031160
 00031170
 00031180
 00031190
 00031200
 00031210
 00031220
 00031230
 00031240
 00031250
 00031260
 00031270
 00031280
 00031290
 00031300
 00031310
 00031320
 00031330
 00031340
 00031350
 00031360
 00031370
 00031380
 00031390
 00031400
 00031410
 00031420
 00031430
 00031440
 00031450
 00031460
 00031470
 00031480
 00031490
 00031500
 00031510
 00031520
 00031530
 00031540
 00031550
 00031560
 00031570
 00031580
 00031590
 00031600
 00031610
 00031620
 00031630
 00031640
 00031650
 00031660
 00031670
 00031680
 00031690
 00031700
 00031710
 00031720
 00031730
 00031740
 00031750

```

C      END
C
C      SUBROUTINE VXMUL (A,B,ACC)
C      DOUBLE PRECISION A,B,ACC(2)
C      DOUBLE PRECISION X,HA,TA,HB,TB
C      INTEGER IX(2),I
C      LOGICAL M1
C      EQUIVALENCE (X,LX(1),IX(1)),(I,LI(1))
C      DATA I/O/
C
C      SPECIFICATIONS FOR ARGUMENTS
C      SPECIFICATIONS FOR LOCAL VARIABLES
C
C      SPLIT A = HA+TA
C      B = HB+TB
C      FIRST EXECUTABLE STATEMENT
C
C      X = A
C      LI(4) = LX(5)
C      IX(2) = 0
C      I = (I/16)*16
C      LX(5) = LI(4)
C      HA=X
C      TA=A-HA
C      X = B
C      LI(4) = LX(5)
C      IX(2) = 0
C      I = (I/16)*16
C      LX(5) = LI(4)
C      HB = X
C      TB = B-HB
C
C      COMPUTE HAHB,HANTB,TANHb, AND TANTB
C      AND CALL VXADD TO ACCUMULATE THE
C      SUM
C
C      X = TANTB
C      CALL VXADD(X,ACC)
C      X = HANTB
C      CALL VXADD(X,ACC)
C      X = TANHb
C      CALL VXADD(X,ACC)
C      X = HAHB
C      CALL VXADD(X,ACC)
C      RETURN
C      END
C
C      SUBROUTINE VXSTO (ACC,D)
C      DOUBLE PRECISION ACC(2),D
C      D = ACC(1)+ACC(2)
C      RETURN
C      END
C
C      SUBROUTINE ZRPOLY (A,NDEO,Z,IER)
C      INTEGER NDEO,IER
C      DOUBLE PRECISION A(1),Z(1)

```

00031760
00031770
00031780
00031790
00031800
00031810
00031820
00031830
00031840
00031850
00031860
00031870
00031880
00031890
00031900
00031910
00031920
00031930
00031940
00031950
00031960
00031970
00031980
00031990
00032000
00032010
00032020
00032030
00032040
00032050
00032060
00032070
00032080
00032090
00032100
00032110
00032120
00032130
00032140
00032150
00032160
00032170
00032180
00032190
00032200
00032210
00032220
00032230
00032240
00032250
00032260
00032270
00032280
00032290
00032300
00032310
00032320
00032330
00032340
00032350

C	INTEGER	N, NN, J, JJ, I, NM1, ICNT, N2, L, N2, NPI	00032360
	REAL	ETA, RMRE, RINFP, REPSR, RADIX, RLO, XX, YY, SINR,	00032370
	1 REAL	COSR, RMAX, RMIN, X, SC, XM, FT, DX, DF, BND, XXX, ARE	00032380
	DOUBLE PRECISION	PT(101)	00032390
	1 DOUBLE PRECISION	TEMP(101), P(101), QP(101), RK(101), GK(101),	00032400
		SVK(101)	00032410
	1 DOUBLE PRECISION	SR, SI, U, V, RA, RB, C, D, A1, A2, A3,	00032420
		A6, A7, E, F, G, H, SZR, SZI, RLZR, RLZI,	00032430
	2	T, AA, BU, CC, FACTOR, REPSR1, ZERO, ONE, FN	00032440
	LOGICAL	ZEROK	00032450
	COMMON /ZRPQLJ/	P, QP, RK, QK, SVK, SR, SI, U, V, RA, RB, C, D, A1, A2, A3, A6,	00032460
	1	A7, E, F, G, H, SZR, SZI, RLZR, RLZI, ETA, ARE, RMRE, N, NN	00032470
		THE FOLLOWING STATEMENTS SET MACHINE	00032480
		CONSTANTS USED IN VARIOUS PARTS OF	00032490
		THE PROGRAM. THE MEANING OF THE	00032500
		FOUR CONSTANTS ARE - REPSR1 THE	00032510
		MAXIMUM RELATIVE REPRESENTATION	00032520
		ERROR WHICH CAN BE DESCRIBED AS	00032530
		THE SMALLEST POSITIVE FLOATING	00032540
		POINT NUMBER SUCH THAT 1.+REPSR1 IS	00032550
		GREATER THAN 1	00032560
		RINFP THE LARGEST FLOATING-POINT	00032570
		NUMBER	00032580
		REPSR THE SMALLEST POSITIVE	00032590
		FLOATING-POINT NUMBER IF THE	00032600
		EXPONENT RANGE DIFFERS IN SINGLE	00032610
		AND DOUBLE PRECISION THEN REPSR	00032620
		AND RINFP SHOULD INDICATE THE	00032630
		SMALLER RANGE	00032640
		RADIX THE BASE OF THE FLOATING-POINT	00032650
		NUMBER SYSTEM USED	00032660
	DATA	RINFP/27FFFFFFF/	00032670
	DATA	REPSR/200100000/	00032680
	DATA	RADIX/16.0/	00032690
	DATA	REPSR1/23410000000000000/	00032700
	DATA	ZERO/0.000/ ONE/1.000/	00032710
		ZRPOLY USES SINGLE PRECISION	00032720
		CALCULATIONS FOR SCALING, BOUNDS	00032730
		AND ERROR CALCULATIONS.	00032740
		FIRST EXECUTABLE STATEMENT	00032750
	IER = 0		00032760
	IF (NDEG .GT. 100 .OR. NDEG .LT. 1) GO TO 165		00032770
	ETA = REPSR1		00032780
	ARE = ETA		00032790
	RMRE = ETA		00032800
	RLO = REPSR/ETA		00032810
		INITIALIZATION OF CONSTANTS FOR	00032820
		SHIFT ROTATION	00032830
	XX = .7071068		00032840
	YY = -XX		00032850
	SINR = .9975641		00032860
	COSR = -.06975647		00032870
	N = NDEG		00032880
	NN = N+1		00032890
		ALGORITHM FAILS IF THE LEADING	00032900
		COEFFICIENT IS ZERO.	00032910
	IF (A(1).NE.ZERO) GO TO 5		00032920
	IER = 130		00032930
	GO TO 9000		00032940
			00032950

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

C	PT(NN) = -PT(NN)	COMPUTE UPPER ESTIMATE OF BOUND	00033560
	X = EXP((ALOG(-PT(NN))-ALOG(PT(1)))/N)		00033570
	IF (PT(N).EQ.0.) GO TO 65		00033580
C		IF NEWTON STEP AT THE ORIGIN IS	00033590
C		BETTER, USE IT.	00033600
	XN = -PT(NN)/PT(N)		00033610
	IF (XM.LT.X) X = XM		00033620
C		CHOP THE INTERVAL (0,X) UNTIL FF.LE.	00033630
65	XM = XM.1		00033640
	FF = PT(1)		00033650
	DO 70 I=2,NN		00033660
70	FF = FF*XM+PT(I)		00033670
	IF (FF.LE.0.) GO TO 75		00033680
	X = XM		00033690
	GO TO 65		00033700
75	DX = X		00033710
C		DO NEWTON ITERATION UNTIL X	00033720
C		CONVERGES TO TWO DECIMAL PLACES	00033730
80	IF (ABS(DX/X).LE..005) GO TO 90		00033740
	FF = PT(1)		00033750
	DF = FF		00033760
	DO 85 I=2,N		00033770
	FF = FF*X+PT(I)		00033780
	DF = DF*X+FF		00033790
85	CONTINUE		00033800
	FF = FF*X+PT(NN)		00033810
	DX = FF/DF		00033820
	X = X-DX		00033830
	GO TO 80		00033840
90	BND = X		00033850
C		COMPUTE THE DERIVATIVE AS THE INITIAL	00033860
C		K POLYNOMIAL AND DO 5 STEPS WITH	00033870
C		NO SHIFT	00033880
	NM1 = N-1		00033890
	FN = ONE/N		00033900
	DO 95 I=2,N		00033910
95	RK(I) = (NN-I)*P(I)*FN		00033920
	RK(1) = P(1)		00033930
	AA = P(NN)		00033940
	BB = P(N)		00033950
	ZEROK = RK(N).EQ.ZERO		00033960
	DO 115 JJ=1,5		00033970
	CC = RK(N)		00033980
	IF (ZEROK) GO TO 105		00033990
C		USE SCALED FORM OF RECURRENCE IF	00034000
C		VALUE OF K AT 0 IS NONZERO	00034010
	T = -AA/CC		00034020
	DO 100 I=1,NM1		00034030
	J = NN-I		00034040
	RK(J) = T*RK(J-1)+P(J)		00034050
100	CONTINUE		00034060
	RK(1) = P(1)		00034070
	ZEROK = DABS(RK(N)).LE.DABS(BB)*ETAN10.		00034080
	GO TO 115		00034090
C		USE UNSCALED FORM OF RECURRENCE	00034100
105	DO 110 I=1,NM1		00034110
	J = NN-I		00034120
	RK(J) = RK(J-1)		00034130
110	CONTINUE		00034140
			00034150

```

      RK(1) = ZERO
      ZEROK = RK(N).EQ.ZERO
115 CONTINUE
      DO 120 I=1,N
120 TEMP(I) = RK(I)
      DO 140 ICNT=1,20
      XXX = COSRXXX-SINRYYY
      YY = SINRXXX+COSRYYY
      XX = XXX
      SR = BNDXXX
      SI = BNDYYY
      U = -SR-SR
      V = BNDMBND
      CALL ZRPQLB (20*ICNT,NZ)
      IF (NZ.EQ.0) GO TO 130
      J = NDEG-N+1
      JJ = J+NDEG
      Z(J) = SZR
      Z(JJ) = SZI
      NN = NN-NZ
      N = NN-1
      DO 125 I=1,NN
125 P(I) = QP(I)
      IF (NZ.EQ.1) GO TO 20
      Z(J+1) = RLZR
      Z(JJ+1) = RLZI
      GO TO 20
      DO 135 I=1,N
135 RX(I) = TEMP(I)
140 CONTINUE
      IER = 131
      DO 145 I=1,NDEG
      NPI = NDEG+I
      P(I) = Z(NPI)
150 CONTINUE
      NZ = NDEG+NDEG
      J = NDEG
  
```

00034160
 00034170
 00034180
 00034190
 00034200
 00034210
 00034220
 00034230
 00034240
 00034250
 00034260
 00034270
 00034280
 00034290
 00034300
 00034310
 00034320
 00034330
 00034340
 00034350
 00034360
 00034370
 00034380
 00034390
 00034400
 00034410
 00034420
 00034430
 00034440
 00034450
 00034460
 00034470
 00034480
 00034490
 00034500
 00034510
 00034520
 00034530
 00034540
 00034550
 00034560
 00034570
 00034580
 00034590
 00034600
 00034610
 00034620
 00034630
 00034640
 00034650
 00034660
 00034670
 00034680
 00034690
 00034700
 00034710
 00034720
 00034730
 00034740
 00034750

```

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,6H ZRPOLY)                     00034930
9005 RETURN                                      00034940
END                                              00034950
C                                              00034960
C                                              00034970
C                                              00034980
SUBROUTINE ZRPQLB (L2,NZ)                        00034990
C SPECIFICATIONS FOR ARGUMENTS                00035000
  INTEGER L2,NZ                                00035010
C SPECIFICATIONS FOR LOCAL VARIABLES          00035020
  INTEGER N,NN,J,ITYPE,I,IFLAG                 00035030
  REAL ARE,BETAS,BETAV,ETA,OSS,OTS,OTV,OVV,2HRE,SS, 00035040
  TS,TSS,TV,TVV,VV                            00035050
  1 DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00035060
  DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00035070
  1 A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,          00035080
  2 SVU,SVV,UI,VI,S,ZERO                      00035090
  LOGICAL VPASS,SPASS,VTRY,STRY               00035100
  COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00035110
  1 A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,2HRE,N,NN 00035120
  DATA ZERO/0.000/                          00035130
C FIRST EXECUTABLE STATEMENT                  00035140
  NZ = 0                                       00035150
C COMPUTES UP TO L2 FIXED SHIFT              00035160
  K-POLYNOMIALS, TESTING FOR                 00035170
  CONVERGENCE IN THE LINEAR OR               00035180
  QUADRATIC CASE. INITIATES ONE OF           00035190
  THE VARIABLE SHIFT ITERATIONS AND          00035200
  RETURNS WITH THE NUMBER OF ZEROS           00035210
  FOUND.                                      00035220
  L2 - LIMIT OF FIXED SHIFT STEPS            00035230
  NZ -NUMBER OF ZEROS FOUND                  00035240
C                                              00035250
C                                              00035260
C                                              00035270
C                                              00035280
C                                              00035290
C                                              00035300
C                                              00035310
C                                              00035320
C                                              00035330
C                                              00035340
C                                              00035350
  BETAV = .25
  BETAS = .25
  OSS = SR
  OVV = V
C EVALUATE POLYNOMIAL BY SYNTHETIC
  DIVISION
  CALL ZRPQLH (NN,U,V,P,QP,RA,RB)
  CALL ZRPQL (ITYPE)
  DO 40 J=1,L2
C CALCULATE NEXT K POLYNOMIAL AND
  ESTIMATE V

```

	CALL ZRPQLF (ITYPE)	00035360
	CALL ZRPQLE (ITYPE)	00035370
	CALL ZRPQLG (ITYPE,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.ITYPE.EQ.3) GO TO 35	00035450
C	COMPUTE RELATIVE MEASURES OF	00035460
C	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
C	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
C	CONVERGENCE TEST. STORE VARIABLES	00035610
C	BEFORE ITERATING	00035620
	SVU = U	00035630
	SVV = V	00035640
	DO 5 I=1,N	00035650
5	SVK(I) = RK(I)	00035660
	S = SS	00035670
C	CHOOSE ITERATION ACCORDING TO THE	00035680
C	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
C	THAT IT HAS BEEN TRIED AND	00035760
C	DECREASE THE CONVERGENCE	00035770
C	CRITERION.	00035780
	VTRY = TRUE.	00035790
	BETAU = BETAU*.25	00035800
C	TRY LINEAR ITERATION IF IT HAS NOT	00035810
C	BEEN TRIED AND THE S SEQUENCE IS	00035820
C	CONVERGING	00035830
	IF (STRY.OR.(.NOT.SPASS)) GO TO 25	00035840
	DO 15 I=1,N	00035850
15	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
C	THAT IT HAS BEEN TRIED AND	00035900
C	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 INTERATION           00035970
      UI = -(S+S) 00035980
      VI = SWS 00035990
      GO TO 10 00036000
C          RESTORE VARIABLES 00036010
25      U = SWU 00036020
      V = SVV 00036030
      DO 30 I=1,N 00036040
30      RK(I) = K(I) 00036050
C          TRY QUADRATIC ITERATION IF IT HAS 00036060
C          NOT BEEN TRIED AND THE V SEQUENCE 00036070
C          IS CONVERGING 00036080
C          IF (VPASS.AND.(.NOT.VTRY)) GO TO 10 00036090
C          RECOMPUTE QP AND SCALAR VALUES TO 00036100
C          CONTINUE THE SECOND STAGE 00036110
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB) 00036120
      CALL ZRPQLE (ITYPE) 00036130
35      OVV = VV 00036140
      OSS = SS 00036150
      OTV = TV 00036160
      OTS = TS 00036170
40      CONTINUE 00036180
      RETURN 00036190
      END 00036200
C          00036210
C          00036220
C          00036230
C          SUBROUTINE ZRPQLC (UU,VV,NZ) 00036240
C          SPECIFICATIONS FOR ARGUMENTS 00036250
      INTEGER NZ 00036260
      DOUBLE PRECISION UU,VV 00036270
C          SPECIFICATIONS FOR LOCAL VARIABLES 00036280
      INTEGER N,NN,J,I,ITYPE 00036290
      REAL ARE,EE,ETA,OMP,RELSTP,RMP,RMRH,T,ZM 00036300
      DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00036310
      DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00036320
      A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI, 00036330
      UI,VI,ZERO,PT01,ONE 00036340
      LOGICAL TRIED 00036350
      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00036360
      A7,E,F,G,H,SZR,SZI,PLZR,RLZI,ETA,ARE,RMRE,N,NN 00036370
      DATA ZERO,PT01,ONE/0.000,0.0100,1.000/ 00036380
C          FIRST EXECUTABLE STATEMENT 00036390
      NZ = 0 00036400
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          UU,VV - COEFFICIENTS OF STARTING 00036450
C          QUADRATIC 00036460
C          NZ - NUMBER OF ZERO FOUND 00036470
      TRIED = .FALSE. 00036480
      U = UU 00036490
      V = VV 00036500
      J = 0 00036510
C          MAIN LOOP 00036520
5      CALL ZRPQLE (ONE,U,V,SZR,SZI,RLZR,RLZI) 00036530
      RETURN IF ROOTS OF THE QUADRATIC ARE 00036540
      REAL AND NOT CLOSE TO MULTIPLE OR 00036550
  
```

```

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

```

```

C      SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)                                00037160
C      INTEGER NZ,IFLAG SPECIFICATIONS FOR ARGUMENTS                    00037170
C      DOUBLE PRECISION SSS                                           00037180
C      INTEGER N,NN,J,I SPECIFICATIONS FOR LOCAL VARIABLES             00037190
C      REAL ARE,EE,ETA,OMP,OMP,RMS,RMRE                                00037200
C      DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101)        00037210
C      DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3,                 00037220
C      A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,                               00037230
C      PV,RKV,T,S,ZERO,PT001                                          00037240
C      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00037250
C      A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN                00037260
C      DATA ZERO/0.000/,PT001/0.00100/                               00037270
C      ----- VARIABLE-SHIFT H POLYNOMIAL                             00037280
C      ----- ITERATION FOR A REAL ZERO SSS -                         00037290
C      ----- STARTING ITERATE                                         00037300
C      ----- NZ - NUMBER OF ZERO FOUND                                00037310
C      ----- IFLAG - FLAG TO INDICATE A PAIR OF                      00037320
C      ----- ZEROS NEAR REAL AXIS                                     00037330
C      ----- FIRST EXECUTABLE STATEMENT                               00037340
C      NZ = 0                                                            00037350
C      S = SSS                                                            00037360
C      IFLAG = 0                                                         00037370
C      J = 0                                                             00037380
C      ----- MAIN LOOP                                               00037390
C      5 PV = P(1) EVALUATE P AT S                                       00037400
C      QP(1) = PV                                                         00037410
C      DO 10 I=2,NN                                                       00037420
C      PV = PV*S+P(I)                                                     00037430
C      QP(I) = PV                                                         00037440
C      10 CONTINUE                                                         00037450
C      RMP = DABS(PV)                                                      00037460
C      ----- COMPUTE A RIGOROUS BOUND ON THE                         00037470
C      ----- ERROR IN EVALUATING P                                    00037480
C      RMS = DABS(S)                                                       00037490
C      EE = (RMRE/(ARE+RMRE))*ABS(SNGL(QP(1)))                             00037500
C      DO 15 I=2,NN                                                         00037510
C      15 EE = EE*RMS+ABS(SNGL(QP(I)))                                     00037520
C      ----- ITERATION HAS CONVERGED SUFFICIENTLY                    00037530
C      ----- IF THE POLYNOMIAL VALUE IS LESS                         00037540
C      ----- THAN 20 TIMES THIS BOUND                                00037550
C      IF (RMP.GT.20.*((ARE+RMRE)*EE-RMRE*RMP)) GO TO 20                 00037560
C      NZ = 1                                                                00037570
C      SZR = S                                                              00037580
C      SZI = ZERO                                                            00037590
C      RETURN                                                                00037600
C      20 J = J+1                                                           00037610
C      ----- STOP ITERATION AFTER 10 STEPS                            00037620
C      IF (J.GT.10) RETURN                                                  00037630
C      IF (J.LT.2) GO TO 25                                                 00037640
C      IF (DABS(T).GT.PT001*DABS(S-T).OR.RMP.LE.OMP) GO TO 25            00037650
C      ----- A CLUSTER OF ZEROS NEAR THE REAL                      00037660
C      ----- AXIS HAS BEEN ENCOUNTERED RETURN                      00037670
C      ----- WITH IFLAG SET TO INITIATE A                          00037680
C      ----- QUADRATIC ITERATION                                     00037690
C      IFLAG = 1                                                            00037700
C      SSS = S                                                              00037710
C      RETURN                                                                00037720
C      -----                                                         00037730
C      -----                                                         00037740
C      -----                                                         00037750

```

C		RETURN IF THE POLYNOMIAL VALUE HAS	00037760
C		INCREASED SIGNIFICANTLY	00037770
C	25 OMP = RMP	COMPUTE T, THE NEXT POLYNOMIAL, AND	00037780
C		THE NEW ITERATE	00037790
	RKV = RK(1)		00037800
	QK(1) = RKV		00037810
	DO 30 I=2,N		00037820
	RKV = RKVMS+RK(I)		00037830
	QK(I) = RKV		00037840
	50 CONTINUE		00037850
	IF (DABS(RKV).LE.DABS(RK(N))*10.*ETA) GO TO 40		00037860
C		USE THE SCALED FORM OF THE	00037870
C		RECURRENCE IF THE VALUE OF K AT S	00037880
C		IS NONZERO	00037890
	T = -PV/RKV		00037900
	RK(1) = QP(1)		00037910
	DO 35 I=2,N		00037920
	35 RK(I) = T*QK(I-1)+QP(I)		00037930
	GO TO 50		00037940
C		USE UNSCALED FORM	00037950
	40 RK(1) = ZERO		00037960
	DO 45 I=2,N		00037970
	45 RK(I) = QK(I-1)		00037980
	50 RKV = RK(1)		00037990
	DO 55 I=2,N		00038000
	55 RKV = RKVMS+RK(I)		00038010
	T = ZERO		00038020
	IF (DABS(RKV).GT.DABS(RK(N))*10.*ETA) T = -PV/RKV		00038030
	S = S+T		00038040
	GO TO 5		00038050
	END		00038060
C			00038070
C			00038080
C			00038090
C			00038100
C	IMSL ROUTINE NAME - ZRPQLE		00038110
C			00038120
C			00038130
C	COMPUTER - IBM/DOUBLE		00038140
C			00038150
C	LATEST REVISION - JANUARY 1, 1978		00038160
C			00038170
C	SUBROUTINE ZRPQLE (ITYPE)		00038180
C		SPECIFICATIONS FOR ARGUMENTS	00038190
C	INTEGER ITYPE		00038200
C		SPECIFICATIONS FOR LOCAL VARIABLES	00038210
	INTEGER N,NN		00038220
	REAL ARE,ETA,RMRE		00038230
	DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101)		00038240
	DOUBLE PRECISION SR,S1,U,V,RA,RB,C,D,A1,A2,A3,		00038250
	1 A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI		00038260
	1 COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,S1,U,V,RA,RB,C,D,A1,A2,A3,A6,		00038270
	1 A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN		00038280
C		THIS ROUTINE CALCULATES SCALAR	00038290
C		QUANTITIES USED TO COMPUTE THE	00038300
C		NEXT K POLYNOMIAL AND NEW	00038310
C		ESTIMATES OF THE QUADRATIC	00038320
C		COEFFICIENTS	00038330
C		ITYPE - INTEGER VARIABLE SET HERE	00038340
C			00038350

```

C      INDICATING HOW THE CALCULATIONS ARE NORMALIZED TO AVOID OVERFLOW
C      SYNTHETIC DIVISION OF K BY THE QUADRATIC 1,U,V
C      FIRST EXECUTABLE STATEMENT
C      CALL ZRPQLH (N,U,V,RK,QK,C,D)
C      IF (DABS(C).GT.DABS(RK(N))*100.*ETA) GO TO 5
C      IF (DABS(D).GT.DABS(RK(N-1))*100.*ETA) GO TO 5
C      ITYPE = 3
C      TYPE=3 INDICATES THE QUADRATIC IS ALMOST A FACTOR OF K
C      RETURN
C      5 IF (DABS(D).LT.DABS(C)) GO TO 10
C      ITYPE = 2
C      TYPE=2 INDICATES THAT ALL FORMULAS ARE DIVIDED BY D
C      E = RA/D
C      F = C/D
C      G = U*RB
C      H = V*RB
C      A3 = (RA+G)*E+H*(RB/D)
C      A1 = RB*F-RA
C      A7 = (F+U)*RA+H
C      RETURN
C      10 ITYPE = 1
C      TYPE=1 INDICATES THAT ALL FORMULAS ARE DIVIDED BY C
C      E = RA/C
C      F = D/C
C      G = U*E
C      H = V*RB
C      A3 = RA*E+(H/C+G)*RB
C      A1 = RB-RA*(D/C)
C      A7 = RA+G*D+H*F
C      RETURN
C      END
C      SUBROUTINE ZRPQLF (ITYPE)
C      INTEGER ITYPE
C      SPECIFICATIONS FOR ARGUMENTS
C      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      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/0.000/
C      COMPUTES THE NEXT K POLYNOMIALS 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)*ETA*10.) GO TO 10
C      IF A1 IS NEARLY ZERO THEN USE A SPECIAL FORM OF THE RECURRENCE

```

00038360
 00038370
 00038380
 00038390
 00038400
 00038410
 00038420
 00038430
 00038440
 00038450
 00038460
 00038470
 00038480
 00038490
 00038500
 00038510
 00038520
 00038530
 00038540
 00038550
 00038560
 00038570
 00038580
 00038590
 00038600
 00038610
 00038620
 00038630
 00038640
 00038650
 00038660
 00038670
 00038680
 00038690
 00038700
 00038710
 00038720
 00038730
 00038740
 00038750
 00038760
 00038770
 00038780
 00038790
 00038800
 00038810
 00038820
 00038830
 00038840
 00038850
 00038860
 00038870
 00038880
 00038890
 00038900
 00038910
 00038920
 00038930
 00038940
 00038950

```

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

```

C	10	B1 = -RK(N)/P(NN)	00039560
		B2 = -(RK(N-1)+B1*P(N))/P(NN)	00039570
		C1 = VMB2*A1	00039580
		C2 = B1*A7	00039590
		C3 = B1*B1*A3	00039600
		C4 = C1-C2-C3	00039610
		TEMP = A5+B1*A4-C4	00039620
		IF (TEMP.EQ.ZERO) GO TO 15	00039630
		UU = U-(U*(C3+C2)+V*(B1*A1+B2*A7))/TEMP	00039640
		VV = V*(1+C4/TEMP)	00039650
		RETURN	00039660
C		IF TYPE=3 THE QUADRATIC IS ZEROED	00039670
	15	UU = ZERO	00039680
		VV = ZERO	00039690
		RETURN	00039700
		END	00039710
C			00039720
C			00039730
C			00039740
C		SUBROUTINE ZRPOLH (NN,U,V,P,Q,RA,RB)	00039750
		SPECIFICATIONS FOR ARGUMENTS	00039760
C		INTEGER NN	00039770
		DOUBLE PRECISION P(NN),Q(NN),U,V,RA,RB	00039780
C		SPECIFICATIONS FOR LOCAL VARIABLES	00039790
		INTEGER I	00039800
		DOUBLE PRECISION C	00039810
C			00039820
C		DIVIDES P BY THE QUADRATIC 1,U,V	00039830
C		PLACING THE QUOTIENT IN Q AND THE	00039840
C		REMAINDER IN A,B	00039850
C		FIRST EXECUTABLE STATEMENT	00039860
		RB = P(1)	00039870
		Q(1) = RB	00039880
		RA = P(2)-U*RB	00039890
		Q(2) = RA	00039900
		DO 5 I=3,NN	00039910
		C = P(I)-U*RA-V*RB	00039920
		Q(I) = C	00039930
		RB = RA	00039940
		RA = C	00039950
	5	CONTINUE	00039960
		RETURN	00039970
		END	00039980
C			00039990
C			00040000
C			00040010
C		IMSL ROUTINE NAME - ZRPQLI	00040020
C			00040030
C			00040040
C			00040050
C		COMPUTER - IBM/DOUBLE	00040060
C			00040070
C		LATEST REVISION - JANUARY 1, 1978	00040080
C			00040090
C		SUBROUTINE ZRPQLI (RA,B1,C,SR,SI,RLR,RLI)	00040100
		SPECIFICATIONS FOR ARGUMENTS	00040110
		DOUBLE PRECISION RA,B1,C,SR,SI,RLR,RLI	00040120
C		SPECIFICATIONS FOR LOCAL VARIABLES	00040130
		DOUBLE PRECISION RE,D,E,ZERO,ONE,TWO	00040140
		DATA ZERO,ONE,TWO/0.000,1.000,2.000/	00040150

U

C	IER = 0	FIRST EXECUTABLE STATEMENT	00040760
	N1 = N+1		00040770
	N2 = N+2		00040780
	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
10	CONTINUE		00040840
C	CALL LEQ1C(WA,N,N,B,M,1B,1,WK,IER)	FACTOR MATRIX A	00040850
	IF (IER .NE. 0) GO TO 9000		00040860
	IF (IJOB .EQ. 1) GO TO 9005		00040870
C	DO 65 J = 1,M	SAVE THE RIGHT HAND SIDES	00040880
	DO 20 I = 1,N		00040890
	WA(I,N1) = B(I,J)		00040900
20	CONTINUE		00040910
C	CALL LEQ1C(WA,N,N,WA(1,N1),1,N,2,WK,IER)	OBTAIN A SOLUTION	00040920
C	XNORM = ZERO	COMPUTE THE NORM OF THE SOLUTION	00040930
	DO 25 I = 1,N		00040940
	TEMPA = WA(I,N1)		00040950
	XNORM = DMAX1(XNORM,DABS(AR),DABS(AI))		00040960
25	CONTINUE		00040970
	IF (XNORM .EQ. ZERO) GO TO 65		00040980
C	DO 50 ITER = 1,ITMAX	COMPUTE RESIDUALS	00040990
	DO 40 I = 1,N		00041000
	TEMPB = B(I,J)		00041010
	ACC(1) = 0.000		00041020
	ACC(2) = 0.000		00041030
	CALL VXADD(BR,ACC)		00041040
	DO 30 JJ = 1,N		00041050
	TEMPA = A(I,JJ)		00041060
	TEMPB = WA(JJ,N1)		00041070
	CALL VXMUL(-AR,PR,ACC)		00041080
	CALL VXMUL(AI,BI,ACC)		00041090
30	CONTINUE		00041100
	CALL VXSTO(ACC,CR)		00041110
	TEMPB = B(I,J)		00041120
	ACC(1) = 0.000		00041130
	ACC(2) = 0.000		00041140
	CALL VXADD(BI,ACC)		00041150
	DO 35 JJ = 1,N		00041160
	TEMPA = A(I,JJ)		00041170
	TEMPB = WA(JJ,N1)		00041180
	CALL VXMUL(-AR,BI,ACC)		00041190
	CALL VXMUL(-BR,AI,ACC)		00041200
35	CONTINUE		00041210
	CALL VXSTO(ACC,C1)		00041220
	WA(I,N2) = TEMPC		00041230
40	CONTINUE		00041240
	CALL LEQ1C(WA,N,N,WA(1,N2),1,N,2,WK,IER)		00041250
	DXNORM = ZERO		00041260
	DO 45 I = 1,N	UPDATE THE SOLUTION	00041270
			00041280
			00041290
			00041300
			00041310
			00041320
			00041330
			00041340
			00041350

```

      WA(I,N1) = WA(I,N1)+WA(I,N2)
      TEMPA = WA(I,N2)
      DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI))
45  CONTINUE
      IF (XNORM+DXNORM .EQ. XNORM) GO TO 55
50  CONTINUE
      IER = 130
C   STORE THE SOLUTION
55  DO 60 JK = 1,N
      B(JK,J) = WA(JK,N1)
60  CONTINUE
      IF (IER .NE. 0) GO TO 9000
65  CONTINUE
      GO TO 9005
9000 CONTINUE
      CALL UERTST(IER,AHLEQ2C)
9005 RETURN
      END
C
C
C   SUBROUTINE LEQ1C (A,N,IA,B,M,IB,IJOB,WA,IER)
C   SPECIFICATIONS FOR ARGUMENTS
      INTEGER N,IA,M,IB,IJOB,IER
      COMPLEX*16 A(IA,N),B(IB,M)
      DOUBLE PRECISION WA(N)
C   SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION P,Q,ZERO,ONE,T(2),RN,BIG
      COMPLEX*16 SUM,TEMP
      INTEGER I,J,JM1,IM1,K,IMAX,JP1,IN,N1
      EQUIVALENCE (SUM,T(1))
      DATA ZERO/0.0D0/,ONE/1.0D0/
C   INITIALIZATION
C   FIRST EXECUTABLE STATEMENT
      IER = 0
      IF (IJOB .EQ. 2) GO TO 75
      RN = N
C   FIND EQUILIBRATION FACTORS
      DO 10 I=1,N
      BIG = ZERO
      DO 5 J=1,N
      TEMP = A(I,J)
      P = CDABS(TEMP)
      IF (P .GT. BIG) BIG = P
5  CONTINUE
      IF (BIG .EQ. ZERO) GO TO 105
      WA(I) = ONE/BIG
10 CONTINUE
C   L-U DECOMPOSITION
      DO 70 J = 1,N
      JM1 = J-1
      IF (JM1 .LT. 1) GO TO 25
C   COMPUTE U(I,J), I=1,...,J-1
      DO 20 I=1,JM1
      SUM = A(I,J)
      IM1 = I-1
      IF (IM1 .LT. 1) GO TO 20
      DO 15 K=1,IM1
      SUM = SUM-A(I,K)*A(K,J)
15  CONTINUE

```

00041360
 00041370
 00041380
 00041390
 00041400
 00041410
 00041420
 00041430
 00041440
 00041450
 00041460
 00041470
 00041480
 00041490
 00041500
 00041510
 00041520
 00041530
 00041540
 00041550
 00041560
 00041570
 00041580
 00041590
 00041600
 00041610
 00041620
 00041630
 00041640
 00041650
 00041660
 00041670
 00041680
 00041690
 00041700
 00041710
 00041720
 00041730
 00041740
 00041750
 00041760
 00041770
 00041780
 00041790
 00041800
 00041810
 00041820
 00041830
 00041840
 00041850
 00041860
 00041870
 00041880
 00041890
 00041900
 00041910
 00041920
 00041930
 00041940
 00041950

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

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

ALGORITHMIC SINGULARITY

PRINT ERROR

00042560
00042570
00042580
00042590
00042600
00042610
00042620
00042630
00042640
00042650
00042660
00042670
00042680
00042690

APPENDIX B
SAMCJ Program Listing

התאחדות העובדים

C
C
C
C
C

```

COMMON/RT/R
COMMON/MFS/F5CD
COMMON/NTP/NELTYP
COMMON/NPT/NOPT2,NOPT6,NOPT7,NOPT8
COMMON/MD/E1,E2,G12,V12,V21
COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
COMMON/ELP/AX,8X,NOUT,NSTS
COMMON/FCC/ELWDTH,ELTHK,ELLOAD
COMMON/NCST/NCASE,NTYPE
COMMON/DISP/ANR2
COMMON/PBB/PLYK,BARK,BARU
COMMON/ELS/ELSTFF,ELSTSS
COMMON/CMT2/XOUT,YOUT
COMMON/SER/NT,NB
DATA Y,Y,Y
DATA CMC,'C'

      READ IN REQUIRED INPUT DATA

      WRITE(6,876)
876  FORMAT(///,10X,' PROGRAM SAMCJ',///,
  * ' PROGRAM SAMCJ PREDICTS THE FAILURE LOAD, FAILURE ',/,
  * ' LOCATION, AND FAILURE MODE IN MULTIPLY-FASTENED, ',/,
  * ' SINGLE OR DOUBLE LAP COMPOSITE SHEAR JOINTS. ',/,
  * ' THE ANALYSIS ASSUMES THAT INPUT PARAMETERS ARE ',/,
  * ' SPECIFIED IN ENGLISH UNITS - LENGTH IN INCHES, ',/,
  * ' MODULI AND STRENGTHS IN PSI. ',/)
      WRITE(6,900)
900  FORMAT(' ENTER: ',/,
  * ' 1 FOR SLS (SINGLE LAP SHEAR)',/,
  * ' 2 FOR DLS (DOUBLE LAP SHEAR)',/)
      READ(5,M) NSDLS
      WRITE(6,911)
911  FORMAT(' ENTER: ',/,
  * ' 1 FOR STATIC TENSION ',/,
  * ' 2 FOR STATIC COMPRESSION',/)
      READ(5,M) LTNCM
106  FORMAT(A1)
380  CONTINUE
      DO 300 K=1,2
      IF(K.EQ.1) WRITE(6,912)
      IF(K.EQ.2) WRITE(6,913)
912  FORMAT(' IS THE TOP PLATE A COMPOSITE OR A METAL?')
913  FORMAT(' IS THE BOTTOM PLATE A COMPOSITE OR A METAL?')
      WRITE(6,914)
914  FORMAT(' ENTER C OR M IN THE FIRST FIELD')
      READ(5,106) CM(K)
      WRITE(6,203)
203  FORMAT(' INPUT MATERIAL DESCRIPTION OF THIS PLATE ',/,
  * ' EX: ASA/3501-6')
      READ(5,204) (MTL(K,I),I=1,15)
204  FORMAT(15A4)
300  CONTINUE
      IF(CM(1).NE.CMC.OR.CM(2).NE.CMC) WRITE(6,754)
754  FORMAT(//,' NOTE: FOR COMPUTATIONAL PURPOSES A ',/,
  * ' METALLIC PLATE IS MODELED AS A 30 PLY ',/,
  * ' LAMINATE OF 0 DEGREE PLYS WITH ISOTROPIC ',/,
  * ' MATERJ' ' PROPERTIES',/)

```

```

DO 306 K=1,2
IF(K.EQ.1) WRITE(6,216)
IF(K.EQ.2) WRITE(6,555)
216 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF THE TOP PLATE')
555 FORMAT(' INPUT THE ENGINEERING PROPERTIES OF THE BOTTOM PLATE')
IF(CM(K).EQ.CMC) GO TO 85
WRITE(6,95)
95 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO')
READ(5,M) 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,M) E1(K),E2(K)
WRITE(6,213)
218 FORMAT(' INPUT THE SHEAR MODULUS AND MAJOR POISSONS RATIO')
READ(5,M) G12(K),V12(K)
V21(K)=V12(K)*E2(K)/E1(K)
306 CONTINUE
307 CONTINUE
290 CONTINUE
DO 303 K=1,2
IF(CM(K).EQ.CMC) GO TO 45
NUMPLY(K)=1
GO TO 303
45 CONTINUE
IF(K.EQ.1) WRITE(6,207)
IF(K.EQ.2) WRITE(6,702)
207 FORMAT(' INPUT TOTAL NUMBER OF DISTINCT PLY ',/,
* ' ORIENTATIONS IN THE TOP PLATE')
702 FORMAT(' INPUT TOTAL NUMBER OF DISTINCT PLY ',/,
* ' ORIENTATIONS IN THE BOTTOM PLATE')
READ(5,M) 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,M) ANG(L,K)
209 CONTINUE
WRITE(6,1823)
1823 FORMAT(/, ' THICKNESS VARIATIONS MAY BE APPROXIMATED',/,
* ' BY ASSIGNING DIFFERENT LAYUPS TO ELEMENTS',/,
* ' IN A COMPOSITE PLATE OR BY SPECIFYING DIFFERENT',/,
* ' THICKNESSES TO ELEMENTS IN A METALLIC PLATE',/)
IF(NSDLS.EQ.2) WRITE(6,789)
789 FORMAT(/, ' NOTE: FOR THE DOUBLE LAP SHEAR CASE, FOR',/,
* ' THE BOTTOM PLATE, ENTER ONLY HALF FOR THE ',/,
* ' LAYUP FOR A COMPOSIT OR HALF THE THICKNESS ',/,
* ' FOR A METALLIC',/)
DO 811 I=1,2
IF(CM(I).EQ.CMC) GO TO 891
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750

```

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 ',/,	00001830
* ' TOP PLATE')	00001840
813 FORMAT(/, ' ENTER NUMBER OF DIFFERENT LAYUPS IN THE ',/,	00001850
* ' BOTTOM PLATE')	00001860
READ(5,M) NL	00001870
DO 814 J=1,NL	00001880
WRITE(6,815) J	00001890
815 FORMAT(' ENTER NUMBER OF PLIES IN LAYUP NO ',I5)	00001900
READ(5,M) NELPLS(I,J)	00001910
WRITE(6,816)	00001920
816 FORMAT(' ENTER PLY THICKNESS FOR THIS LAYUP')	00001930
READ(5,M) PLYTHK(I,J)	00001940
NN=NELPLS(I,J)	00001950
WRITE(6,818)	00001960
818 FORMAT(' ENTER SEQUENCE OF PLY TYPES FROM TOP TO BOTTOM')	00001970
DO 817 K=1,NN	00001980
READ(5,M) NELPT(I,J,K)	00001990
817 CONTINUE	00002000
814 CONTINUE	00002010
811 CONTINUE	00002020
WRITE(6,855)	00002030
855 FORMAT(/, ' FASTENER DESCRIPTION:',/,)	00002040
WRITE(6,250)	00002050
250 FORMAT(' INPUT MATERIAL DESCRIPTION FOR FASTENER')	00002060
READ(5,251) (MTL(3,I),I=1,15)	00002070
251 FORMAT(15A4)	00002080
WRITE(6,252)	00002090
252 FORMAT(' INPUT YOUNGS MODULUS AND POISSONS RATIO FOR',/,	00002100
* ' THE FASTENER')	00002110
READ(5,M) FASE,FASV	00002120
WRITE(6,253)	00002130
253 FORMAT(' INPUT THE DIAMETER OF THE FASTENER')	00002140
READ(5,M) FASD	00002150
WRITE(6,888)	00002160
888 FORMAT(/, ' FASTENER TYPE',/,	00002170
* ' ENTER: 1 FOR PROTRUDING HEAD',/,	00002180
* ' 2 FOR COUNTERSUNK HEAD')	00002190
READ(5,M) NFTYP	00002200
R(1)=1.0D10	00002210
R(2)=1.0D10	00002220
IF(NFTYP.EQ.1) GO TO 360	00002230
WRITE(6,889)	00002240
889 FORMAT(/, ' ENTER PLATE WHICH CONTAINS THE COUNTERSUNK',/,	00002250
* ' HEAD (OPPOSITE PLATE ASSUMES THE NUT HEAD)',/,	00002260
* ' ENTER: 1 FOR TOP PLATE',/,	00002270
* ' 2 FOR BOTTOM PLATE')	00002280
READ(5,M) N	00002290
R(N)=0.0D0	00002300
360 CONTINUE	00002310
WRITE(6,477)	00002320
477 FORMAT(/, ' GRID LAYOUT:',/,)	00002330
C INPUT GRIDS, ELEMENT CONNECTIVITY AND PROPERTIES	00002340
	00002350

00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
0000248
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002775
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

IF(CM(1).EQ.CMC) GO TO 627	00002960
IF(NELTYP(I).NE.1) GO TO 721	00002970
WRITE(6,1721)	00002980
1721 FORMAT(' ENTER ELEMENT THICKNESS')	00002990
READ(5,X) ATH	00003000
721 IF(NELTYP(I).NE.2) GO TO 722	00003010
WRITE(6,723)	00003020
723 FORMAT(' ENTER ELEMENT THICKNESS')	00003030
READ(5,X) ATH	00003040
FSCD(I,1)=GCOORD(NELCHA(I,6),1)	00003050
FSCD(I,2)=GCOORD(NELCHA(I,6),2)	00003060
FSCD(I,3)=FASD/2.000	00003070
722 IF(NELTYP(I).NE.3) GO TO 724	00003080
WRITE(6,725)	00003090
725 FORMAT(' ENTER ELEMENT THICKNESS, X AND Y COORDINATES',/,	00003100
*' OF OPEN HOLE AND HOLE RADIUS')	00003110
READ(5,X) ATH,(FSCD(I,J),J=1,3)	00003120
724 ELTHK(I)=ATH/30.000	00003130
PLYTHK(I,1)=ATH/30.000	00003140
LYPN(I)=1	00003150
GO TO 474	00003160
627 CONTINUE	00003170
IF(NELTYP(I).NE.1) GO TO 726	00003180
WRITE(6,727)	00003190
727 FORMAT(' ENTER ELEMENT LAYUP NO')	00003200
READ(5,X) LYPN(I)	00003210
726 IF(NELTYP(I).NE.2) GO TO 728	00003220
WRITE(6,729)	00003230
729 FORMAT(' ENTER ELEMENT LAYUP NO')	00003240
READ(5,X) LYPN(I)	00003250
FSCD(I,1)=GCOORD(NELCHA(I,6),1)	00003260
FSCD(I,2)=GCOORD(NELCHA(I,6),2)	00003270
FSCD(I,3)=FASD/2.000	00003280
728 IF(NELTYP(I).NE.3) GO TO 730	00003290
WRITE(6,731)	00003300
731 FORMAT(' ENTER ELEMENT LAYUP NUMBER, X AND Y ',/,	00003310
*' COORDINATES OF THE OPEN HOLE AND THE HOLE',/,	00003320
*' RADIUS')	00003330
READ(5,X) LYPN(I),(FSCD(I,J),J=1,3)	00003340
730 ELTHK(I)=PLYTHK(1,LYPN(I))	00003350
474 CONTINUE	00003360
WRITE(6,688)	00003370
688 FORMAT(/,' ENTER NUMBER OF ELEMENTS IN BOTTOM PLATE '	00003380
READ(5,X) NEL2	00003390
NELTOT=NEL1+NEL2	00003400
NP1=NEL1+1	00003410
DO 611 I=NP1,NELTOT	00003420
WRITE(6,800) I	00003430
800 FORMAT(' FOR ELEMENT NO',I5,	00003440
*' ENTER: ELEMENT ID,N1,N2,N3,N4,N5,ELEMENT TYPE')	00003450
READ(5,X) (NELCON(I,J),J=1,6),NELTYP(I)	00003460
DO 593 IL=2,6	00003470
IC=0	00003480
NELCHA(I,1)=NELCON(I,1)	00003490
NIN=NP1+1	00003500
DO 594 KL=NIN,NGTOT	00003510
IF(NELCON(I,IL).EQ.NGRID(KL)) IC=1	00003520
IF(NELCON(I,IL).EQ.NGRID(KL)) NELCHA(I,IL)=KL	00003530
IF(IC.EQ.1) GO TO 593	00003540
594 CONTINUE	00003550

```

593 CONTINUE
  IF(CM(2).EQ.CMC) GO TO 927
  IF(NELTYP(I).NE.1) GO TO 921
  WRITE(6,1921)
1921 FORMAT(' ENTER ELEMENT THICKNESS')
  READ(5,M) ATH
  921 IF(NELTYP(I).NE.2) GO TO 922
  WRITE(6,1923)
1923 FORMAT(' ENTER ELEMENT THICKNESS')
  READ(5,M) ATH
  FSCD(I,1)=GCOORD(NELCNA(I,6),1)
  FSCD(I,2)=GCOORD(NELCNA(I,6),2)
  FSCD(I,3)=FASD/2.0D0
  WRITE(6,3443) I,NELCNA(I,6),FSCD(I,1),FSCD(I,2)
3443 FORMAT(' C I NELCNA FSCD12',I5,2X,5(D9.3,2X))
  922 IF(NELTYP(I).NE.3) GO TO 924
  WRITE(6,925)
  925 FORMAT(' ENTER ELEMENT THICKNESS, X AND Y COORDINATES',/,
    * OF OPEN HOLE AND HOLE RADIUS')
  READ(5,M) ATH,(FSCD(I,J),J=1,3)
  924 ELTHK(I)=ATH/30.0D0
  PLYTHK(2,1)=ATH/30.0D0
  LYPN(I)=1
  GO TO 611
  927 CONTINUE
  IF(NELTYP(I).NE.1) GO TO 926
  WRITE(6,1927)
1927 FORMAT(' ENTER ELEMENT LAYUP NO')
  READ(5,M) LYPN(I)
  926 IF(NELTYP(I).NE.2) GO TO 928
  WRITE(6,929)
  929 FORMAT(' ENTER ELEMENT LAYUP NO')
  READ(5,M) LYPN(I)
  FSCD(I,1)=GCOORD(NELCNA(I,6),1)
  FSCD(I,2)=GCOORD(NELCNA(I,6),2)
  FSCD(I,3)=FASD/2.0D0
  928 IF(NELTYP(I).NE.3) GO TO 930
  WRITE(6,931)
  931 FORMAT(' ENTER ELEMENT LAYUP NUMBER, X AND Y ',/,
    * COORDINATES OF THE OPEN HOLE AND THE HOLE',/,
    * RADIUS')
  READ(5,M) LYPN(I),(FSCD(I,J),J=1,3)
  930 ELTHK(I)=PLYTHK(2,LYPN(I))
  611 CONTINUE
  WRITE(6,1741)
1741 FORMAT(/,' FASTENERS ARE MODELED BY EFFECTIVE ',/,
    * FASTENER ELEMENTS WHICH PROVIDE THE ',/,
    * ELASTIC LINK BETWEEN THE TOP AND ',/,
    * BOTTOM PLATES',/)
  WRITE(6,1711)
1711 FORMAT(' ENTER NUMBER OF FASTENERS IN JOINT ')
  READ(5,M) NUMF
  WRITE(6,,16)
  716 FORMAT(/,
    *
    * EFFECTIVE FASTENER ELEMENTS ARE
    * NUMBERED AS SHOWN:
    *
    *      N1 (TOP PLATE)
    *
    *      N2 (BOTTOM PLATE)
    *

```

00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004150

```

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

  DETERMINE GRID STORAGE LOCATIONS FOR
  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
      CONTINUE
    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=NOPI+1
        DO 617 K=NIN,NOPTOT
          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
        CONTINUE
      CONTINUE
    CONTINUE
    395 CONTINUE
    DO 741 J=1,NUMF
      N=2
      NELFSA(I,1)=NELFAS(I,1)
      DO 242 J=1,N
        IC=0
        DO 243 K=1,NOTOT
          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
        CONTINUE
      CONTINUE
    CONTINUE
    741 CONTINUE

  COMPUTE ELEMENT WIDTHS

  DO 239 I=1,NELTOT
    ELWIDTH(I)=DABS(GCOORD(NELCNA(I,3),2)-GCOORD(NELCNA(I,2),2))
  239 CONTINUE

```

```

00004160
00004170
00004180
00004190
00004200
00004210
00004220
00004230
00004240
00004250
00004260
00004270
00004280
00004290
00004300
00004310
00004320
00004330
00004340
00004350
00004360
00004370
00004380
00004390
00004400
00004410
00004420
00004430
00004440
00004450
00004460
00004470
00004480
00004490
00004500
00004510
00004520
00004530
00004540
00004550
00004560
00004570
00004580
00004590
00004600
00004610
00004620
00004630
00004640
00004650
00004660
00004670
00004680
00004690
00004700
00004710
00004720
00004730
00004740
00004750

```

```

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

```

```

00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005080
00005090
00005100
00005110
00005120
00005130
00005140
00005150
00005160
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005310
00005320
00005330
00005340
00005350

```

```

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

```

```

00005360
00005370
00005380
00005390
00005400
00005410
00005420
00005430
00005440
00005450
00005460
00005470
00005480
00005490
00005500
00005510
00005520
00005530
00005540
00005550
00005560
00005570
00005580
00005590
00005600
00005610
00005620
00005630
00005640
00005650
00005660
00005670
00005680
00005690
00005700
00005710
00005720
00005730
00005740
00005750
00005760
00005770
00005780
00005790
00005800
00005810
00005820
00005830
00005840
00005850
00005860
00005870
00005880
00005890
00005900
00005910
00005920
00005930
00005940
00005950

```

```

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

```

```

00005960
00005970
00005980
00005990
00006000
00006010
00006020
00006030
00006040
00006050
00006060
00006070
00006080
00006090
00006100
00006110
00006120
00006130
00006140
00006150
00006160
00006170
00006180
00006190
00006200
00006210
00006220
00006230
00006240
00006250
00006260
00006270
00006280
00006290
00006300
00006310
00006320
00006330
00006340
00006350
00006360
00006370
00006380
00006390
00006400
00006410
00006420
00006430
00006440
00006450
00006460
00006470
00006480
00006490
00006500
00006510
00006520
00006530
00006540
00006550

```

```

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

```

```

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

```

```

DO 241 I=1,2
WRITE(6,600) I
600 FORMAT(10X,'PLATE NO ',I5,' ',/)
WRITE(6,601) (MTL(I,J),J=1,15)
601 FORMAT(2X,15A4,/)
HT=NELPLS(I,1)*PLYTHK(I,1)
WRITE(6,691) E1(I),E2(I),G12(I),V12(I),V21(I)
691 FORMAT(2X,'MATERIAL PROPERTIES',/,/,
  *10X,'E1' =',D9.3,' PSI',/,
  *10X,'E2' =',D9.3,' PSI',/,
  *10X,'G12' =',D9.3,' PSI',/,
  *10X,'NU12' =',D9.3,/,
  *10X,'NU21' =',D9.3,/)
241 CONTINUE
WRITE(6,606)
606 FORMAT(10X,'FASTENER DESCRIPTION',/,/)
WRITE(6,607) (MTL(3,J),J=1,15)
607 FORMAT(2X,15A4,/)
WRITE(6,647) FASD
647 FORMAT(2X,' DIAMETER = ',D9.3,' INCHES',/)
WRITE(6,609) FASE,FASV
609 FORMAT(2X,' MATERIAL PROPERTIES',/,/,
  *10X,'E' =',D9.3,' PSI',/,
  *10X,'MU' =',D9.3,/)
708 CONTINUE
WRITE(6,923)
923 FORMAT(10X,'FAILURE ANALYSIS',/,/)
WRITE(6,558)
558 FORMAT(2X,' AN AVERAGE STRESS CRITERION WILL BE USED',/,)
DO 631 I=1,2
WRITE(6,632) I
632 FORMAT(2X,' PLATE NUMBER',I5,/)
NP=NUMPLY(I)
IF(CM(1,1,1,1)) GO TO 3112
WRITE(6,713)
713 FORMAT(2X,' FIBER STRAIN ULTIMATES',/,)
776 WRITE(6,677) (STULT(LL,I),LL=1,3)
677 FORMAT(2X,' EPSILON ULT COMP = ',D9.3,/,
  *2X,' EPSILON ULT TEN = ',D9.3,/,
  *2X,' GAMMA ULT SHEAR = ',D9.3,/)
GO TO 3113
3112 CONTINUE
WRITE(6,3114)
3114 FORMAT(' METALLIC STRENGTHS ',/,)
WRITE(6,3115) STM(1),STM(2),STM(3)
3115 FORMAT(2X,' TENSILE STRENGTH = ',D9.3,/,
  *2X,' COMPRESSIVE STRENGTH = ',D9.3,/,
  *2X,' SHEAR STRENGTH = ',D9.3,/)
3113 CONTINUE
WRITE(6,1563)
1563 FORMAT(10X,' CHARACTERISTIC DISTANCES',/,)
WRITE(6,564) AONT(I),AOBR(I),AOSO(I)
564 FORMAT(' AONT = ',D9.3,' INCHES',/,
  *' AOBR = ',D9.3,' INCHES',/,
  *' AOSO = ',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

```

```

00007160
00007170
00007180
00007190
00007200
00007210
00007220
00007230
00007240
00007250
00007260
00007270
00007280
00007290
00007300
00007310
00007320
00007330
00007340
00007350
00007360
00007370
00007380
00007390
00007400
00007410
00007420
00007430
00007440
00007450
00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610
00007620
00007630
00007640
00007650
00007660
00007670
00007680
00007690
00007700
00007710
00007720
00007730
00007740
00007750

```

```

C      PROPERTIES OF FASTENERS, LOADED HOLES, AND OPEN
C      HOLE REGIONS IN THE JOINT
C
C      INTERNAL APPLIED LOAD SET TO 1 KIP
C
C      APP=1000.0
C      IF(LTNCM.EQ.2) APP=-APP
C      NELTOT=NEL1+NEL2
C      NGTOT=NGP1+NGP2
C
C      INITIALIZE ARRAYS
C
C      DO 1 I=1,50
C      DO 3 J=1,4
C      DO 3 K=1,4
C      3 ELSTFF(I,J,K)=0.
C      1 CONTINUE
C      DO 4 I=1,200
C      PBC(I)=0.
C      HHS(I)=0.
C      AHR(I)=0.
C      AHR2(I)=0.
C      DO 5 J=1,200
C      GLSTFF(I,J)=0.
C      ASQM(I,J)=0.
C      5 CONTINUE
C      4 CONTINUE
C
C      CALCULATION OF EFFECTIVE FASTENER ELEMENT
C      STIFFNESS MATRICIES
C
C      WRITE(6,8418)
C      8418 FORMAT(/,' PAUSE FOR STIFFNESS MATRIX CALCULATIONS',/)
C      NLOOP=NEF(1)
C      DO 444 I=1,NLOOP
C      NEL=NGEF(1,1)
C      DO 5001 II=1,NUMF
C      5001 IF(NEL.EQ.NELFAS(II,1)) IEL=II
C
C      SEARCH FOR LOADED HOLE ELEMENTS CONNECTED TO
C      FASTENER ELEMENT
C
C      NTOP=0
C      NBOT=0
C      DO 643 J=1,NEL1
C      643 IF(NELFAS(IEL,2).EQ.NELCON(J,6)) NTOP=J
C      NP1=NEL1+1
C      DO 446 J=NP1,NELTOT
C      446 IF(NELFAS(IEL,3).EQ.NELCON(J,6)) NBOT=J
C      NPLY(1)=NELPLS(1,LYPN(NTOP))
C      H(1)=ELTHK(NTOP)
C      DO 910 JJJ=1,50
C      910 IPLY(JJJ,1)=NELPT(1,LYPN(NTOP),JJJ)
C      NPLY(2)=NELPLS(2,LYPN(NBOT))
C      H(2)=ELTHK(NBOT)
C      DO 113 JJJ=1,50
C      113 IPLY(JJJ,2)=NELPT(2,LYPN(NBOT),JJJ)
C
C      INITIALIZE PARAMETERS FOR COLLOCATION

```

```

00007760
00007770
00007780
00007790
00007800
00007810
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

```

```

      NT=7
      NOUT=57
      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
      ELEMENT VERTEXES ARE INTERNALLY
      NUMBERED AS:
          3 2
          4 1
      SFX=(GCOORD(NELCNA(NTB,5),1)+GCOORD(NELCNA(NTB,2),1))/2.000
      SFY=(GCOORD(NELCNA(NTB,3),2)+GCOORD(NELCNA(NTB,2),2))/2.000
      DO 128 JJ=1,4
      XC(JJ)=GCOORD(NELCNA(NTB,6-JJ),1)-FSCD(NTB,1)
      YC(JJ)=GCOORD(NELCNA(NTB,6-JJ),2)-FSCD(NTB,2)
128  CONTINUE
      XC(5)=XC(1)
      YC(5)=YC(1)
      W=ELWDTH(NTB)
      AST=1000.0
      CALL POLY(W,AST,JK,K,NCLL,LTNCM)
      CALL CIRC(W,AST,JK,K,LTNCM)
      NOPT4=1
      NCASE=1
      NTYPE=NELTYP(1EL)
      CALL FIGEOM(H,PHI,K,NOPT4,NCLL)
      CALL FBOLT(ANGK,H,PHI,K)
580  CONTINUE
      N=NPLY(1)
      DO 30 II=1,N
      N=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)
      CALCULATION OF FASTENER PROPERTIES
      FASG=FASE/(2.*(1+FASV))
      FASLAM=5.*(1.0+FASV)/(7.+6.*FASV)
      FASR=FASD/2.
      FASA=ACOS(-1.)*FASR**2
      FASI=ACOS(-1.)*FASR**4/4.
      FASSS=FASLAM**FASG**FASA
      FASBS=FASE**FASI
      P=1000.
      CALL CENTD(H,FASSS,FASBS,P)
      CALL SOLVE(H,P,U1,U2)
      IF(L.EQ.2) GO TO 666

```

```

00008360
00008370
00008380
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
00008800
00008810
00008820
00008830
00008840
00008850
00008860
00008870
00008880
00008890
00008900
00008910
00008920
00008930
00008940
00008950

```

CCCCCCCC

```

6011 CONTINUE
DO 6030 KK=1,NELTOT
6030 IF(IEL2.EQ.NELCON(KK,1)) IEL=KK
      H(KJ)=ELTHK(IEL)
      NPLY(KJ)=NELPLS(KJ,LYPN(IEL))
      DO 919 JJJ=1,50
      IPLY(JJJ,KJ)=NELPT(KJ,LYPN(IEL),JJJ)
919 CONTINUE
      NRNK=5
      IF(NELTYP(IEL).EQ.2) NRNK=7

      INTERNAL NUMBERING OF ELEMENT VERTICIES:

              3      2
              4      1

      SFX=(GCOORD(NELCNA(IEL,5),1)+GCOORD(NELCNA(IEL,2),1))/2.000
      SFY=(GCOORD(NELCNA(IEL,3),2)+GCOORD(NELCNA(IEL,2),2))/2.000
      DO 440 K=1,4
      XC(K)=GCOORD(NELCNA(IEL,6-K),1)-FSCD(IEL,1)
      IF(NELTYP(IEL).EQ.1) XC(K)=GCOORD(NELCNA(IEL,6-K),1)-SFX
      YC(K)=GCOORD(NELCNA(IEL,6-K),2)-FSCD(IEL,2)
      IF(NELTYP(IEL).EQ.1) YC(K)=GCOORD(NELCNA(IEL,6-K),2)-SFY
440 CONTINUE
      XC(5)=XC(1)
      YC(5)=YC(1)
      AX=FSCD(IEL,3)
      IF(NELTYP(IEL).EQ.1) AX=0.1
      BX=AX
      PI=DARCOS(-1.000)
      RAD=PI/180.00
      NOAUSS=2*NGP
      NGPT=4*NOAUSS*2
      IC=0
      NCPT=2*NGP

      DETERMINE COORDINATES AT WHICH STRESSES AND
      DISPLACEMENTS ARE TO BE COMPUTED.
      ELEMENT NATURAL FLEXIBILITY MATRICIES
      ARE COMPUTED BY INTEGRATING STRESSES
      FOR EACH LOAD CASE IN THE NATURAL
      MODE METHOD. THE ELEMENTS ARE DIVIDED
      INTO FOUR REGIONS AND THE GAUSSIAN POINTS
      ARE SCALED TO EACH REGION SIZE

      REGION 1

      DO 15 II=1,NCPT
      DO 15 JJ=1,NCPT
      IC=IC+1
      XOUT(IC)=((-AX-XC(1))/2.)*GSSX(II)+(-AX+XC(3))/2.
      YOUT(IC)=(YC(3)-YC(1))/2.)*GSSX(JJ)+(YC(3)+YC(4))/2.
      NGHT(IC)=GSSW(II)*GSSW(JJ)*(YC(3)-YC(4))*(-AX-XC(3))/4.00
15 CONTINUE

      REGION 2

```

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

C	BEARING	00010760
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
41	CONTINUE	00010820
C	ADD COORDINATES ALONG WHICH ELEMENT LOAD	00010830
C	RECOVERY WILL BE COMPUTED	00010840
C	DO 3332 III=1,10	00010850
	IC=IC+1	00010860
	IF(KJ.EQ.1) XOUT(IC)=XC(3)+0.1*AX	00010870
	IF(KJ.EQ.2) XOUT(IC)=XC(1)-0.1*AX	00010880
	YOUT(IC)=((YC(2)-YC(1))/2.000)*OSSX(III)+(YC(2)+YC(1))/2.000	00010890
C	STRESSES ARE SINGULAR AT THETA = 180 DEG OR Y = 0	00010900
C	IF(DABS(YOUT(IC)).LT.0.01) YOUT(IC)=YOUT(IC-1)	00010910
3332	CONTINUE	00010920
4891	CONTINUE	00010930
	NSTS=4*NAVD	00010940
	NOUT=4*(NOAUSS**2)	00010950
C	CALCULATION OF LOADED HOLE, UNLOADED HOLE, AND	00010960
C	PLAIN ELEMENT STIFFNESS MATRICIES	00010970
C	THN=DARCOS(-1.000)/FLOAT(NOP)	00010980
	NN=IEL	00010990
	DO 410 J=1,NNRK	00011000
	NOPT4=5	00011010
	NT=7	00011020
	NCLL=10	00011030
	NB=52+4*NCLL	00011040
	HT=H(KJ)*NPLY(KJ)	00011050
	HCASE=J	00011060
	NTYPE=NELTYP(IEL)	00011070
	CALL MOED(HT,W,AST,J,NN,KJ,NEL,NCLL)	00011080
	CALL MCIR(W,AST,NN,J,NCLL)	00011090
	PHI=0.000	00011100
	CALL AMATRX(H,PHI,KJ)	00011110
	CALL FIGEOM(H,PHI,KJ,NOPT4,NCLL)	00011120
	CALL INFLN(WOHT,H,NNRK,J,KJ,NN,NOPT)	00011130
410	CONTINUE	00011140
4000	CONTINUE	00011150
C	COMPUTE ELEMENT FAILURE VALUES BASED	00011160
C	ON MAXIMUM FIBER STRAIN ALLOWABLES	00011170
C	HT=H(KJ)*NPLY(KJ)	00011180
	IF(NELTYP(IEL).EQ.2) CALL SMAX(HT,KJ,IEL)	00011190
	IF(NELTYP(IEL).EQ.3) CALL SMAX(HT,KJ,IEL)	00011200
	IF(ISLH.EQ.0) GO TO 6040	00011210
	NL=NUMLH(KJ,NCLH)	00011220
	IF(NL.EQ.1) GO TO 400	00011230
	DO 6041 K=2,NL	00011240
	DO 6042 LL=1,NELTOT	00011250
6042	IF(NOLH(KJ,NCLH,K).EQ.NELCON(LL,1)) IEL2=LL	00011260
		00011270
		00011280
		00011290
		00011300
		00011310
		00011320
		00011330
		00011340
		00011350

DO 6043 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=4*NAVD	00011410
DO 6045 ILM=1,NNN	00011420
DO 6045 ILK=1,10	00011430
6045 ELSTSS(IEL2,ILM,ILK)=ELSTSS(IEL,ILM,ILK)	00011440
6041 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(NCOH(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=4*NAVD	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(NCPL(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=4*NAVD	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
C DETERMINE ELEMENT ARRANGEMENT IN TOP	00011820
C AND BOTTOM PLATES	00011830
C	00011840
DO 681 KJ=1,2	00011850
IF(KJ.EQ.2) GO TO 501	00011860
L1=1	00011870
L2=NGP1	00011880
L3=1	00011890
L4=NEL1	00011900
GO TO 502	00011910
501 L1=NGP1+1	00011920
L2=NOTOT	00011930
L3=NEL1+1	00011940
	00011950

```

    L4=NELTOT
502 CONTINUE
    AXMIN=1.D10
    AYMIN=1.D10
    DO 503 I=L1,L2
      IF(AXMIN.GT.GCOORD(I,1)) AXMIN=GCOORD(I,1)
      IF(AYMIN.GT.GCOORD(I,2)) AYMIN=GCOORD(I,2)
      IF(AXMIN.EQ.GCOORD(I,1).AND.AYMIN.EQ.GCOORD(I,2)) NC=I
503 CONTINUE
    DO 574 I=L3,L4
      IF(NELCON(I,2).EQ.NORID(NC)) IEL=I
      NELGRD(KJ,1,1)=IEL
      DO 504 I=1,25
      DO 505 J=1,25
        IFL=0
        DO 506 K=L3,L4
          IF(NELCON(K,2).EQ.NELCON(NELORD(KJ,J,I),3)) IEL=K
          IF(IEL.EQ.0) GO TO 507
          NELORD(KJ,J+1,I)=IEL
505 CONTINUE
507 CONTINUE
          IF(KJ.EQ.1) NROW1=J
          IF(KJ.EQ.2) NROW2=J
          IEL=0
          DO 508 L=L3,L4
            IF(NELCON(NELORD(KJ,I+1,I),5).EQ.NELCON(
              L,2)) IEL=L
            IF(IEL.EQ.0) GO TO 507
            NELORD(KJ,I+1,I)=IEL
504 CONTINUE
509 CONTINUE
            IF(KJ.EQ.1) NCOL1=I
            IF(KJ.EQ.2) NCOL2=I
681 CONTINUE
    CCG
    COMPUTE NODAL DEGREES OF FREEDOM
    IC=0
    DO 540 KJ=1,2
      IF(KJ.EQ.1) NR=NROW1
      IF(KJ.EQ.1) NC=NCOL1
      IF(KJ.EQ.2) NR=NROW2
      IF(KJ.EQ.2) NC=NCOL2
      NELD1S(NELORD(KJ,1,1),1,1)=IC+1
      NELD1S(NELORD(KJ,1,1),1,2)=IC+2
      NELD1S(NELORD(KJ,1,1),2,1)=IC+3
      NELD1S(NELORD(KJ,1,1),2,2)=IC+4
      IC=IC+4
      IF(NR.EQ.1) GO TO 549
      DO 541 I=2,NR
        NELD1S(NELORD(KJ,I,1),1,1)=NELDIS(NELORD(KJ,I-1
          ,1),2,1)
        NELD1S(NELORD(KJ,I,1),1,2)=NELDIS(NELORD(KJ,I-1
          ,1),2,2)
        NELD1S(NELORD(KJ,I,1),2,1)=IC+1
        NELD1S(NELORD(KJ,I,1),2,2)=IC+2
      IC=IC+2
541 CONTINUE
549 CONTINUE
      DO 542 I=1,NC

```

```

00011960
00011970
00011980
00011990
00012000
00012010
00012020
00012030
00012040
00012050
00012060
00012070
00012080
00012090
00012100
00012110
00012120
00012130
00012140
00012150
00012160
00012170
00012180
00012190
00012200
00012210
00012220
00012230
00012240
00012250
00012260
00012270
00012280
00012290
00012300
00012310
00012320
00012330
00012340
00012350
00012360
00012370
00012380
00012390
00012400
00012410
00012420
00012430
00012440
00012450
00012460
00012470
00012480
00012490
00012500
00012510
00012520
00012530
00012540
00012550

```

```

DO 543 J=1,NR
  IF(I.EQ.1) GO TO 544
  NELDIS(NELORD(KJ,J,I),1,1)=NELDIS(NELORD(
  *KJ,J,I-1),4,1)
  NELDIS(NELORD(KJ,J,I),1,2)=NELDIS(NELORD(
  *KJ,J,I-1),4,2)
  NELDIS(NELORD(KJ,J,I),2,1)=NELDIS(NELORD(
  *KJ,J,I-1),3,1)
  NELDIS(NELORD(KJ,J,I),2,2)=NELDIS(NELORD(
  *KJ,J,I-1),3,2)
544 CONTINUE
  IF(J.EQ.1) GO TO 561
  NELDIS(NELORD(KJ,J,I),4,1)=NELDIS(NELORD(KJ,J-1,I),3,1)
  NELDIS(NELORD(KJ,J,I),4,2)=NELDIS(NELORD(KJ,J-1,I),3,2)
  GO TO 562
561 CONTINUE
  NELDIS(NELORD(KJ,J,I),4,1)=IC+1
  NELDIS(NELORD(KJ,J,I),4,2)=IC+2
  IC=IC+2
562 CONTINUE
  IF(HELTY(NELORD(KJ,J,I)).NE.2) GO TO 543
  NELDIS(NELORD(KJ,J,I),5,1)=IC+1
  NELDIS(NELORD(KJ,J,I),5,2)=IC+2
  IC=IC+2
543 CONTINUE
  NELDIS(NELORD(KJ,J,I),3,1)=IC+1
  NELDIS(NELORD(KJ,J,I),3,2)=IC+2
  IC=IC+2
543 CONTINUE
542 CONTINUE
540 CONTINUE
C
C      DETERMINE BOUNDARY NODES AND VALUES
C
  NRD=2*(NGP1+NGP2)
  DO 165 I=1,100
165 PBC(I)=0.
C
C      DISTRIBUTE APPLIED LOAD
C
  ATOT=GCOORD(NELCNA(NELORD(1,NROW1,1),3),2)
  *GCCORD(NELCNA(NELORD(1,1,1),2),2)
  APL=APP/ATOT
  SG=1.0
  IF(LTHCM.EQ.1) SG=-1.0
  DO 178 I=1,NROW1
  A1=GCOORD(NELCNA(NELORD(1,I,1),3),2)
  *GCCORD(NELCNA(NELORD(1,I,1),2),2)
  M1=NELDIS(NELORD(1,I,1),1,1)
  M2=NELDIS(NELORD(1,I,1),2,1)
  PBC(M1)=PBC(M1)+SG*(0.5*DABS(APL*M1))
  PBC(M2)=PBC(M2)+SG*(0.5*DABS(APL*M2))
178 CONTINUE
1119 CONTINUE
C
C      ASSEMBLE GLOBAL STIFFNESS MATRIX
C
  DO 220 N1=1,NELTOT
  IR=5
  IF(HELTY(N1).NE.2) IR=4

```

```

00012560
00012570
00012580
00012590
00012600
00012610
00012620
00012630
00012640
00012650
00012660
00012670
00012680
00012690
00012700
00012710
00012720
00012730
00012740
00012750
00012760
00012770
00012780
00012790
00012800
00012810
00012820
00012830
00012840
00012850
00012860
00012870
00012880
00012890
00012900
00012910
00012920
00012930
00012940
00012950
00012960
00012970
00012980
00012990
00013000
00013010
00013020
00013030
00013040
00013050
00013060
00013070
00013080
00013090
00013100
00013110
00013120
00013130
00013140
00013150

```

C		00013160
C	TOP AND BOTTOM PLATE LOADED HOLE AND	00013170
C	UNLOADED HOLE ELEMENTS	00013180
	IC1=0	00013190
	DO 425 N2=1,IR	00013200
	DO 425 N3=1,2	00013210
	M1=NELDIS(N1,N2,N3)	00013220
	IC1=IC1+1	00013230
	IC2=0	00013240
	DO 425 N4=1,IR	00013250
	DO 425 N5=1,2	00013260
	M2=NELDIS(N1,N4,N5)	00013270
	IC2=IC2+1	00013280
	OLSTFF(M1,M2)=OLSTFF(M1,M2)+ELSTFF(N1,IC1,IC2)	00013290
425	CONTINUE	00013300
220	CONTINUE	00013310
C		00013320
C	ADD EFFECTIVE FASTENER ELEMENTS	00013330
	DO 260 I=1,NUMF	00013340
	DO 1541 J=1,NEL1	00013350
1541	IF(NELFAS(I,2).EQ.NELCON(J,6)) N=J	00013360
	N1=NELDIS(N,5,1)	00013370
	N2=NELDIS(N,5,2)	00013380
	NL=NEL1+1	00013390
	DO 1562 J=NL,NELTOT	00013400
1562	IF(NELFAS(I,3).EQ.NELCON(J,6)) N=J	00013410
	N3=NELDIS(N,5,1)	00013420
	N4=NELDIS(N,5,2)	00013430
	OLSTFF(N1,N1)=OLSTFF(N1,N1)+RDSTFF(I,1)	00013440
	OLSTFF(N1,N3)=OLSTFF(N1,N3)+RDSTFF(I,1)	00013450
	OLSTFF(N2,N2)=OLSTFF(N2,N2)+RDSTFF(I,2)	00013460
	OLSTFF(N2,N4)=OLSTFF(N2,N4)+RDSTFF(I,2)	00013470
	OLSTFF(N3,N3)=OLSTFF(N3,N3)+RDSTFF(I,1)	00013480
	OLSTFF(N3,N1)=OLSTFF(N3,N1)+RDSTFF(I,1)	00013490
	OLSTFF(N4,N4)=OLSTFF(N4,N4)+RDSTFF(I,2)	00013500
	OLSTFF(N4,N2)=OLSTFF(N4,N2)+RDSTFF(I,2)	00013510
260	CONTINUE	00013520
	NP=2*(NCP1+NCP2)	00013530
C		00013540
C	GLOBAL BOUNDARY CONDITIONS	00013550
	DO 415 I=1,NP	00013560
	RHS(I)=PBC(I)	00013570
415	CONTINUE	00013580
C		00013590
	IC=1	00013600
	NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),4,1)	00013610
	DO 437 I=1,NROW2	00013620
	IC=IC+1	00013630
	NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),3,1)	00013640
437	CONTINUE	00013650
	IC=IC+1	00013660
	NZERO(IC)=NELDIS(NELORD(2,1,NCOL2),4,2)	00013670
	NUMZ=IC	00013680
C		00013690
C	RESTORE REDUCED STIFFNESS MATRIX	00013700
	ICF=0	00013710
		00013720
		00013730
		00013740
		00013750

[illegible]

0000

	GO TO 2011	00014360
3112	CONTINUE	00014370
	ANR(L)=0.	00014380
2011	CONTINUE	00014390
C		00014400
C	CALCULATE NODAL LOADS	00014410
C		00014420
	IC=0	00014430
	DO 44 I=1,NRD	00014440
	DO 54 J=1,NUMZ	00014450
	IF(I.NE.NZERO(J)) GO TO 54	00014460
	ANR2(I)=0.0D0	00014470
	GO TO 44	00014480
54	CONTINUE	00014490
	IC=IC+1	00014500
	ANR2(I)=ANR(IC)	00014510
44	CONTINUE	00014520
	WRITE(6,3712)	00014530
3712	FORMAT(/,10X,'ELEMENT FORCES',//)	00014540
	DO 500 K=1,HELTOF	00014550
	NID=HELCON(K,1)	00014560
	WRITE(6,3947) NID	00014570
8947	FORMAT(/,' ELEMENT ID',IS,/,	00014580
	*6X,'GRID',9X,'FX',9X,'FY',/)	00014590
	IR=5	00014600
	KL=K	00014610
	IF(KL.GT.NE1) AL=K-NEL	00014620
	IF(NE1YP(K).NE.2) IR=4	00014630
	DO 510 I=1,IR	00014640
	SUMU=0.	00014650
	SUMV=0.	00014660
	N=2*I-1	00014670
	DO 520 J=1,IR	00014680
	N1=HELDIS(K,J,1)	00014690
	N2=HELDIS(K,J,2)	00014700
	SUMU=SUMU+ELSTFF(K,N,(2*J-1))*ANR2(N1)+	00014710
	*ELSTFF(K,N,(2*J))*ANR2(N2)	00014720
520	CONTINUE	00014730
	N=2*I	00014740
	DO 530 J=1,IR	00014750
	N1=HELDIS(K,J,1)	00014760
	N2=HELDIS(K,J,2)	00014770
	SUMV=SUMV+ELSTFF(K,N,(2*J-1))*ANR2(N1)+	00014780
	*ELSTFF(K,N,(2*J))*ANR2(N2)	00014790
530	CONTINUE	00014800
C		00014810
C	STORE ELEMENT LOADS FOR CHECK ON ELEMENT	00014820
C	LOAD RECOVERY	00014830
C		00014840
	IF(K.LE.NEL1.AND.(I.EQ.1.OR.I.EQ.2)) ELLOAD(K,I)=SUMU	00014850
	IF(K.GT.NEL1.AND.(I.EQ.3.OR.I.EQ.4)) ELLOAD(K,I-2)=SUMU	00014860
	NID=HELCON(K,I+1)	00014870
	WRITE(6,3239) NID,SUMU,SUMV	00014880
3239	FORMAT(2X,I8,5X,2(D9.3,2X))	00014890
510	CONTINUE	00014900
500	CONTINUE	00014910
C		00014920
C	COMPUTE ELEMENT FAILURE LOADS AND DETERMINE	00014930
C	CRITICAL ELEMENT TO CALCULATE JOINT FAILURE	00014940
C	LOAD	00014950

```

C      CALL FCRTT(APP,NEL1,NEL2,NDAM,IN,LTNCM,NAVD)
      FAILV=DABS(ELFAIL(IN,NDAM))
      IF(NSDLS.EQ.2) FAILV=2.*FAILV
      NID=NELCON(IN,1)
      WRITE(6,5555) NID,FAILV
5555  FORMAT(///,' FAILURE IS PREDICTED TO OCCUR IN ELEMENT',/,
      *' NUMBER',I5,' AT AN APPLIED JOINT LOAD VALUE ',/,
      *' OF ',D14.7,' LBS',/)
      IF(NDAM.EQ.1) WRITE(6,5556)
      IF(NDAM.EQ.2) WRITE(6,5557)
      IF(NDAM.EQ.3) WRITE(6,5558)
5556  FORMAT(' THE PREDICTED FAILURE MODE IS NET SECTION')
5557  FORMAT(' THE PREDICTED FAILURE MODE IS SHEAR-OUT ')
5558  FORMAT(' THE PREDICTED FAILURE MODE IS BEARING')
      STOP
      END

C
C
C      SUBROUTINE MGEO(HT,W,AST,J,IN,KJ,NEL,NCL)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A1A(4),A2A(4)
      DIMENSION XB(200),YB(200),A1(200),A2(200)
      DIMENSION NELTYP(50),THTA(200)
      DIMENSION XC(5),YC(5)
      COMMON/CMT1/XB,YB,A1,A2,THTA
      COMMON/XC/YC,XC,YC
      COMMON/NTP/NELTYP

      DETERMINE EXTERIOR COLLOCATION POINTS AND
      STRESS BOUNDARY CONDITIONS CORRESPONDING
      TO THE NATURAL LOAD CASES

      NCS=5
      IF(NELTYP(IN).NE.2) NCS=3
      JK=0
      DO 15 I=1,4
      A1A(I)=0.
15  A2A(I)=0.
      A=(YC(2)-YC(1))*HT
      B=(XC(2)-XC(3))*HT
      IF(J.EQ.1) A1A(3)=1.0D0/A
      IF(J.EQ.1.AND.NELTYP(IN).NE.2) A1A(1)=1.0D0/A
      IF(J.EQ.2) A1A(2)=1.0D0/B
      IF(J.EQ.2.AND.NELTYP(IN).NE.2) A1A(4)=1.0D0/B
      IF(J.EQ.2.AND.NELTYP(IN).NE.2) GO TO 55
      IF(J.EQ.3) A1A(1)=1.0D0/A
      IF(J.EQ.4) A1A(4)=1.0/B
      IF(J.EQ.1.OR.J.EQ.3) AST=1.0D0/A
      IF(J.EQ.2.OR.J.EQ.4) AST=1.0D0/B
55  CONTINUE
      W=XC(3)-XC(2)
      IF(J.EQ.1.OR.J.EQ.3.OR.J.EQ.6) W=YC(2)-YC(1)
      DO 10 I=1,4
      X=XC(I)-XC(I+1)
      Y=YC(I+1)-YC(I)
      IF(X.EQ.0.) X=1.D-6
      IF(Y.EQ.0.) Y=1.D-6

```

00014960
 00014970
 00014980
 00014990
 00015000
 00015010
 00015020
 00015030
 00015040
 00015050
 00015060
 00015070
 00015080
 00015090
 00015100
 00015110
 00015120
 00015130
 00015140
 00015150
 00015160
 00015170
 00015180
 00015190
 00015200
 00015210
 00015220
 00015230
 00015240
 00015250
 00015260
 00015270
 00015280
 00015290
 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

```

TH=DATAN2(X,Y)
TH=TH*180./DARCOS(-0.1D1)
DX=(XC(I+1)-XC(I))/(NCL+1)
DY=(YC(I+1)-YC(I))/(NCL+1)
DO 20 II=1,NCL
JK=JK+1
IF(I.EQ.1.OR.I.EQ.3) GO TO 23
YB(JK)=YC(I)
XB(JK)=XC(I)+DX*(II+.5)
IF(II.EQ.1) XB(JK)=XC(I)+(DX/2.)
GO TO 24
23 YB(JK)=YC(I)+DY*(II+.5)
IF(II.EQ.1) YB(JK)=YC(I)+(DY/2.)
XB(JK)=XC(I)
24 THTA(JK)=TH
A1(JK)=A1A(I)
A2(JK)=A2A(I)
IF(J.EQ.NCS.AND.(I.EQ.1.OR.I.EQ.3)) A1(JK)=(2*YB(JK)/W)*(3.0D0/A)
IF(J.EQ.(NCS+1).AND.(I.EQ.2.OR.I.EQ.4)) A1(JK)=(2*XB(JK)/W)*(3.0
*DO/3)
IF(J.EQ.(NCS+2).AND.(I.EQ.1.OR.I.EQ.3)) A2(JK)=2.0/DSQRT(A**2+B**2
*)
IF(J.EQ.(NCS+2).AND.(I.EQ.2.OR.I.EQ.4)) A2(JK)=-2.0/DSQRT(A**2+B**2
**2)
20 CONTINUE
10 CONTINUE
RETURN
END

SUBROUTINE MCIR(W,AST,I,J,NCL)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XB(200),YB(200),A1(200),A2(200)
DIMENSION THTA(200),NELTYP(50)
COMMON/CNT1/XB,YB,A1,A2,THTA
COMMON/NTP/NELTYP
COMMON/ELP/AX,DX
CCN=-1.0D0
RAD=DARCOS(CON)/180.
BSTR=DABS((2.*W*AST)/(DARCOS(CON)*AX))
IF(NELTYP(I).NE.2) BSTR=0.0D0
IF(NELTYP(I).EQ.2.AND.J.GT.5) BSTR=0.0D0
IC=4*NCL

DETERMINE INTERIOR COLLOCATION POINTS AND
STRESS BOUNDARY CONDITIONS

NBD=52
NBI=NBD/4
DO 1) K=1,4
CR=(K-1)*DARCOS(CON)/2.
DO 20 KI=1,NBI
IC=IC+1
THINC=(DARCOS(CON)/2.)/FLOAT(NBI)
A1(IC)=0.
A2(IC)=0.
THINC2=THINC/2.
THINC2+(KI-1)*THINC+CR
XB(IC)=AX*DCOS(TH)

```

```

YB(IC)=AX*DSIN(TH)
THTA(IC)=TH/RAD
IF(J.GT.4) GO TO 20
IF(NELTYP(1).NE.2) GO TO 40
IF(J.EQ.1.AND.(K.EQ.1.OR.K.EQ.4)) A1(IC)=
  X-BSTR*DABS(DCOS(TH))
IF(J.EQ.2.AND.(K.EQ.3.OR.K.EQ.4)) A1(IC)=
  X-BSTR*DABS(DSIN(TH))
IF(J.EQ.3.AND.(K.EQ.2.OR.K.EQ.3)) A1(IC)=
  X-BSTR*DABS(DCOS(TH))
IF(J.EQ.4.AND.(K.EQ.1.OR.K.EQ.2)) A1(IC)=
  X-BSTR*DABS(DSIN(TH))
40 CONTINUE
20 CONTINUE
10 CONTINUE
  RETURN
  END

```

C
C
C

SUBROUTINE INFLN(WGHT,H,HRNK,J,KJ,I,NOPT)

```

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION ELSTFF(50,10,10),WGHT(500),WK(150)
DIMENSION ELSTS(50,50,10),STSV(50),STSA(50,10)
DIMENSION AN(10,7),UVOUT(20)
DIMENSION PHI(3,7,400),STEMP(10,10),AO(10,3)
DIMENSION FINF(10,10),SINF(10,10),AINV(3,3)
DIMENSION APSX(500),APSY(500),APSYX(500)
DIMENSION H(2),XC(5),YC(5),NPLY(2)
DIMENSION IC(10)
DIMENSION A(10,10),ATEMP2(10,10)
COMMON/UV/UVOUT
COMMON/XCYC/XC,YC
COMMON/ELP/AX,BX,NOUT,NSTS
COMMON/ELS/ELSTFF,ELSTSS
COMMON/STS/STSV
COMMON/INF1/APSX,APSY,APSYX
COMMON/IYP/NPLY
COMMON/INV/AINV

```

C
C
C

COMPUTE ELEMENT STIFFNESS COEFFICIENTS

```

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)=0.000
DO 444 N1=1,10
AO(N1,1)=0.000
AO(N1,2)=0.000
AO(N1,3)=0.000
DO 444 N2=1,7
444 AN(N1,N2)=0.000
DO 110 IN1=1,10
DO 110 IN2=1,10
SINF(IN1,IN2)=0.000
STEMP(IN1,IN2)=0.000

```

00016160
 00016170
 00016180
 00016190
 00016200
 00016210
 00016220
 00016230
 00016240
 00016250
 00016260
 00016270
 00016280
 00016290
 00016300
 00016310
 00016320
 00016330
 00016340
 00016350
 00016360
 00016370
 00016380
 00016390
 00016400
 00016410
 00016420
 00016430
 00016440
 00016450
 00016460
 00016470
 00016480
 00016490
 00016500
 00016510
 00016520
 00016530
 00016540
 00016550
 00016560
 00016570
 00016580
 00016590
 00016600
 00016610
 00016620
 00016630
 00016640
 00016650
 00016660
 00016670
 00016680
 00016690
 00016700
 00016710
 00016720
 00016730
 00016740
 00016750

```

110 FINF(IN1,IN2)=0.000
200 CONTINUE
C
C
C
C
STRESSES AND DISPLACEMENTS ARE STORED
FOR EACH LOAD CASE
DO 2107 KLK=1,8
2107 AN(KLK,J)=UVOUT(KLK)
IF(NRNK.EQ.5) GO TO 2221
IF(J.EQ.1) UVOUT(5)=UVOUT(9)
IF(J.EQ.1) UVOUT(7)=UVOUT(9)
IF(J.EQ.2) UVOUT(2)=UVOUT(16)
IF(J.EQ.2) UVOUT(8)=UVOUT(16)
IF(J.EQ.3) UVOUT(1)=UVOUT(13)
IF(J.EQ.3) UVOUT(3)=UVOUT(16)
IF(J.EQ.4) UVOUT(4)=UVOUT(12)
IF(J.EQ.4) UVOUT(6)=UVOUT(12)
2221 CONTINUE
IF(NRNK.EQ.7 AND J.LT.5) GO TO 371
AN(9,J)=(UVOUT(9)+UVOUT(13))/2.
AN(10,J)=(UVOUT(12)+UVOUT(16))/2.
GO TO 372
371 AN(9,J)=UVOUT(7+2*J)
AN(10,J)=UVOUT(8+2*J)
372 CONTINUE
DO 15 IS=1,NSTS
15 STSA(IS,J)=STGV(IS)
DO 10 IS=1,NGPT
PHI(1,J,IS)=APSX(IS)
10 CONTINUE
DO 20 IS=1,NGPT
PHI(2,J,IS)=APSY(IS)
20 CONTINUE
DO 30 IS=1,NGPT
PHI(3,J,IS)=APSYX(IS)
30 CONTINUE
IF(J.LT.NRNK) RETURN
INTEGRATION OF STRESSES
DO 1010 III=1,10
1010 CONTINUE
NTR=NRNK+3
DO 45 IK=1,NTR
DO 45 JK=1,NTR
45 FINF(K,JK)=0.
H1=H(VJ)*NPLY(KJ)
DO 50 LI=1,NGPT
DO 50 LJ=1,3
DO 50 KI=1,NRNK
SUM=0.
DO 70 IL=1,3
SUM=SUM+H1*AINV(LJ,IL)*PHI(IL,KI,LI)
70 CONTINUE
STEMP(LJ,KI)=SUM
60 CONTINUE
DO 80 LK=1,NRNK
DO 80 LJ=1,NRNK
SUM=0.
DO 90 IL=1,3

```

```

00016760
00016770
00016780
00016790
00016800
00016810
00016820
00016830
00016840
00016850
00016860
00016870
00016880
00016890
00016900
00016910
00016920
00016930
00016940
00016950
00016960
00016970
00016980
00016990
00017000
00017010
00017020
00017030
00017040
00017050
00017060
00017070
00017080
00017090
00017100
00017110
00017120
00017130
00017140
00017150
00017160
00017170
00017180
00017190
00017200
00017210
00017220
00017230
00017240
00017250
00017260
00017270
00017280
00017290
00017300
00017310
00017320
00017330
00017340
00017350

```

SUM=SUM+PHI(IL,LK,LI)*STEMP(IL,LJ)	00017360
90 CONTINUE	00017370
FINF(LK,LJ)=FINF(LK,LJ)+SUM*WGH(LI)	00017380
80 CONTINUE	00017390
50 CONTINUE	00017400
DO 51 III=1, NRNK	00017410
DO 51 JJJ=1, NRNK	00017420
51 STMP(III, JJJ)=(FINF(III, JJJ)+FINF(JJJ, III))/2.0D0	00017430
DO 52 III=1, NRNK	00017440
DO 52 JJJ=1, NRNK	00017450
52 FINF(III, JJJ)=STEMP(III, JJJ)	00017460
CALL LINV2F(FINF, N, WK, 10, SINF, 4, WK, IER)	00017470
DO 410 IA=1, NTR	00017480
AO(IA, 1)=0.5+0.5*(-1.0)**(IA+1)	00017490
AO(IA, 2)=0.5+0.5*(-1.0)**(IA)	00017500
410 AO(IA, 3)=0.0D0	00017510
AO(1, 3)=DABS(YC(4))	00017520
AO(2, 3)=-DABS(XC(4))	00017530
AO(3, 3)=-DABS(YC(3))	00017540
AO(4, 3)=-DABS(XC(3))	00017550
AO(5, 3)=-DABS(YC(2))	00017560
AO(6, 3)=DABS(XC(2))	00017570
AO(7, 3)=DABS(YC(1))	00017580
AO(8, 3)=DABS(XC(1))	00017590
DO 420 KK=1, NTR	00017600
DO 420 LL=1, NRNK	00017610
SUM=0.0D0	00017620
DO 430 JJ=1, NRNK	00017630
430 SUM=SUM+AN(KK, JJ)*SINF(JJ, LL)	00017640
420 STMP(KK, LL)=SUM	00017650
DO 440 KK=1, NTR	00017660
N1=NTR-2	00017670
DO 440 JJ=N1, NTR	00017680
440 STMP(KK, JJ)=AO(KK, JJ-NRNK)	00017690
CALL LINV2F(STMP, NTR, 10, FINF, 4, WK, IER)	00017700
DO 450 II=1, NRNK	00017710
DO 450 JJ=1, NTR	00017720
SUM=0.0D0	00017730
DO 460 KK=1, NRNK	00017740
460 SUM=SUM+SINF(II, KK)*FINF(KK, JJ)	00017750
450 STMP(II, JJ)=SUM	00017760
DO 470 II=1, NTR	00017770
DO 470 JJ=1, NTR	00017780
SUM=0.0D0	00017790
DO 480 KK=1, NRNK	00017800
480 SUM=SUM+FINF(KK, II)*STEMP(KK, JJ)	00017810
ELSTFF(I, II, JJ)=SUM	00017820
A(II, JJ)=SUM	00017830
470 CONTINUE	00017840
DO 550 II=1, NRNK	00017850
DO 550 JJ=1, NTR	00017860
SUM=0.0D0	00017870
DO 560 KK=1, NRNK	00017880
560 SUM=SUM+SINF(II, KK)*FINF(KK, JJ)	00017890
550 STMP(II, JJ)=SUM	00017900
DO 570 II=1, NST3	00017910
DO 570 JJ=1, NTR	00017920
SUM=0.0D0	00017930
DO 580 KK=1, NRNK	00017940
580 SUM=SUM+A(II, KK)*STEMP(KK, JJ)	00017950

```

570 ELSTOP(I,II,1)=SUM
      RETURN
      END
C
C
C
      SUBROUTINE SMAX(HT,KJ,1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AINV(3,3),STULT(3,2),AVN(3)
      DIMENSION NV(3)
      DIMENSION PSMX(50,4),STM(3),CM(2)
      DIMENSION NPLY(2),NUMPLY(2),ANG(5,2),IPLY(100,2)
      DIMENSION E1(2),E2(2),Q12(2),V12(2),V21(2)
      COMMON/MD/E1,E2,Q12,V12,V21
      COMMON/LYP/NPLY,NUMPLY,ANG,IPLY
      COMMON/STMT/STM,CM
      COMMON/INV/AINV
      COMMON/SMX/PSMX
      COMMON/STN/STULT
      DATA CM/0,0/
      IF(CM(KJ).EQ.CM) GO TO 222
      PSMX(1,1)=STM(1)
      PSMX(1,2)=STM(2)
      PSMX(1,3)=STM(3)
      PSMX(1,4)=STM(2)
      RETURN
222 CONTINUE
C
C
C
      COMPUTE LAMINATE FAILURE LOADS BASED ON MAXIMUM
      FIBER STRAINS FOR EACH FAILURE MODE
      DO 100 K=1,3
      DO 10 II=1,3
      NV(II)=0
10  AVN(II)=0.000
      IF(K.EQ.1) NV(1)=1
      IF(K.EQ.2) NV(1)=-1
      IF(K.EQ.3) NV(3)=1
      DO 15 II=1,3
      DO 15 JJ=1,3
      AVN(II)=AVN(II)+AINV(II,JJ)*NV(JJ)
15  CONTINUE
      NP=NUMPLY(KJ)
      CMX=0.000
      RAD=DARCOS(-1.000)/180.00
      DO 25 II=1,NP
      TH=ANG(II,KJ)*RAD
      E1=DCOS(TH)*2*AVN(1)+AVN(2)*DSIN(TH)*2+
      *DCOS(TH)*DSIN(TH)*AVN(3)
      IF(K.EQ.1) GO TO 65
      EPRT=E1/STULT(2,KJ)
      GO TO 50
65  IF(K.EQ.2) GO TO 75
      EPRT=E1/STULT(1,KJ)
      GO TO 50
75  EPRT=E1/STULT(2,KJ)
50  CONTINUE
      IF(DABS(SMX).LT.DABS(EPRT)) SMX=EPRT
25  CONTINUE
      IF(DABS(SMX).GT.1.00-10) GO TO 555
00017960
00017970
00017980
00017990
00018000
00018010
00018020
00018030
00018040
00018050
00018060
00018070
00018080
00018090
00018100
00018110
00018120
00018130
00018140
00018150
00018160
00018170
00018180
00018190
00018200
00018210
00018220
00018230
00018240
00018250
00018260
00018270
00018280
00018290
00018300
00018310
00018320
00018330
00018340
00018350
00018360
00018370
00018380
00018390
00018400
00018410
00018420
00018430
00018440
00018450
00018460
00018470
00018480
00018490
00018500
00018510
00018520
00018530
00018540
00018550

```

	PSMX(I,K)=STULT(3,KJ)*G12(KJ)	00018560
	GO TO 100	00018570
555	CONTINUE	00018580
	PSMX(I,K)=DABS(1.0D0/SMX)	00018590
100	CONTINUE	00018600
	PSMX(I,4)=PSMX(1,2)	00018610
	RETURN	00018620
	END	00018630
		00018640
		00018650
		00018660
	SUBROUTINE POLY(W,AST,J,K,NCOL,LTNCH)	00018670
		00018680
		00018690
	IMPLICIT REAL*8(A-H,O-Z)	00018700
	DIMENSION XC(5),YC(5),A1(200),A2(200),XB(200)	00018710
	DIMENSION YB(200),T(200),A1A(4),A2A(4)	00018720
	COMMON/CHT1/XB,YB,A1,A2,T	00018730
	COMMON/XCYC/XC,YC	00018740
		00018750
	ARRAY COLLOCATION POINTS AROUND EXTERIOR	00018760
	BOUNDARY AND APPLY STRESS BOUNDARY	00018770
	CONDITION	00018780
		00018790
	DO 120 I=1,4	00018800
	A1A(I)=0.0	00018810
	A2A(I)=0.0	00018820
120	CONTINUE	00018830
	IF(LTNCH.EQ.1) A1A(I)=AST	00018840
	IF(LTNCH.EQ.2) A1A(I)=-AST	00018850
	J=0	00018860
	XC(5)=XC(1)	00018870
	YC(5)=YC(1)	00018880
	DO 10 I=1,4	00018890
	X=XC(I)-XC(I+1)	00018900
	Y=YC(I+1)-YC(I)	00018910
	IF(X.EQ.0.) X=1.D-6	00018920
	IF(Y.EQ.0.) Y=1.D-6	00018930
	TH=ATAN2(X,Y)	00018940
	TH=TH*180./DARCOS(-0.1D1)	00018950
	DX=(XC(I+1)-XC(I))/(NCOL+1)	00018960
	DY=(YC(I+1)-YC(I))/(NCOL+1)	00018970
	DO 20 II=1,NCOL	00018980
	J=J+1	00018990
	IF(I.EQ.1.OR.I.EQ.3) GO TO 23	00019000
	YB(J)=YC(I)	00019010
	XB(J)=XC(I)+DX*(II+.5)	00019020
	IF(II.EQ.1) XB(J)=XC(I)+(DX/2.)	00019030
	GO TO 24	00019040
23	CONTINUE	00019050
	YB(J)=YC(I)+DY*(II+.5)	00019060
	IF(II.EQ.1) YB(J)=YC(I)+(DY/2.)	00019070
	XB(J)=XC(I)	00019080
24	T(J)=TH	00019090
	A1(J)=A1A(I)	00019100
	A2(J)=A2A(I)	00019110
20	CONTINUE	00019120
10	CONTINUE	00019130
	RETURN	00019140
	END	00019150

000000000000

ARRAY COLLOCATION POINTS AROUND INNER BOUNDARY AND
APPLY BEARING LOAD IN A COSINUSOIDAL DISTRIBUTION

SUBROUTINE FIGECM(M,PHS,KJ,NOPT4,NCLL)

INTERNET BOARD - FUTURE SOCIETY ANALYSIS BY A
FUTURE COLLECTION TECHNOLOGY

00019160
00019170
00019180
00019190
00019200
00019210
00019220
00019230
00019240
00019250
00019260
00019270
00019280
00019290
00019300
00019310
00019320
00019330
00019340
00019350
00019360
00019370
00019380
00019390
00019400
00019410
00019420
00019430
00019440
00019450
00019460
00019470
00019480
00019490
00019500
00019510
00019520
00019530
00019540
00019550
00019560
00019570
00019580
00019590
00019600
00019610
00019620
00019630
00019640
00019650
00019660
00019670
00019680
00019690
00019700
00019710
00019720
00019730
00019740
00019750

C		00020360
C	CONSTANTS P1,P2,Q1,Q2 ARE NEEDED FOR STRESS CALCULATIONS	00020370
C		00020380
	P1=AI(1,1)*R1**2+AI(1,2)-AI(1,3)*R1	00020390
	P2=AI(1,1)*R2**2+AI(1,2)-AI(1,3)*R2	00020400
	Q1=AI(2,2)/R1+AI(1,2)*R1-AI(2,3)	00020410
	Q2=AI(2,2)/R2+AI(1,2)*R2-AI(2,3)	00020420
C		00020430
C	INPUTS AIN1(I),AIN2(I) ETC. REFER TO BOUNDARY CONDITIONS	00020440
		00020450
	NT4=4*NT	00020460
	NT8=8*NT	00020470
	NT8P4=8*NT+4	00020480
	NT8P2=8*NT+2	00020490
	NT8P1=8*NT+1	00020500
	NB2=2*NB	00020510
	NWK=NT8P1*(NT8P1+2)	00020520
	CALL CMAT(BC,CMCTCM,CMC,CM,RHS,GRHS,NT4,NT8,NT8P4,NT8P2,	00020530
	NT8P1,NB2,NWK,WA,NKK,NOPT4,KJ,NCLL)	00020540
	RETURN	00020550
	END	00020560
C		00020570
C		00020580
C	SUBROUTINE AMATRX(H,PHS,K)	00020590
C		00020600
C	ASSEMBLE THE A MATRIX	00020610
		00020620
	IMPLICIT REAL*8(A-H,O-Z)	00020630
	DIMENSION A(3,3),ANO(5,2),H(2),NPLY(2),NUMPLY(2)	00020640
	DIMENSION E1(2),E2(2),Q12(2),V12(2),V21(2)	00020650
	DIMENSION IPLY(100,2)	00020660
	COMMON/MD/E1,E2,Q12,V12,V21	00020670
	COMMON/AMT/A	00020680
	COMMON/LYP/NPLY,NUMPLY,ANO,IPLY	00020690
	THKNES=NPLY(K)*H(K)	00020700
	DENO=1.-E2(K)*V12(K)*H2/E1(K)	00020710
	Q11=E1(K)/DENO	00020720
	Q22=E2(K)/DENO	00020730
	Q12=V12(K)*Q22	00020740
	Q21=Q12	00020750
	Q33=Q12(K)	00020760
	DO 10 I=1,3	00020770
	DO 10 J=1,3	00020780
10	A(I,J)=0.0D0	00020790
	NN=NPLY(K)	00020800
	T=H(K)	00020810
	DO 20 I=1,NN	00020820
	LP=IPLY(I,K)	00020830
	THTAI=(ANO(LP,K)+PHS)*DARCOS(-1.D0)/180.D0	00020840
	C=DCOS(THTAI)	00020850
	S=DSIN(THTAI)	00020860
	A(1,1)=(Q11*CMH4+2.*(Q12+2.*Q33)*CMCHS*Q22*SMH4)*T+A(1,1)	00020870
	A(2,2)=(Q11*SMH4+2.*(Q12+2.*Q33)*CMCHS*Q22*CMH4)*T+A(2,2)	00020880
	A(1,2)=((Q11+Q12-4.*Q33)*CMCHS*Q12*(CMH4+SMH4))*T+A(1,2)	00020890
	A(2,1)=A(1,2)	00020900
	A(3,3)=((Q11+Q22-2.*Q12-2.*Q33)*CMCHS*Q33*(CMH4+SMH4))*T+A(3,3)	00020910
	A(1,3)=((Q11-Q12-2.*Q33)*CMCHS*(Q12-Q22+2.*Q33)*SMH3*CM)*T+A(1,3)	00020920
	A(2,3)=((Q11-Q12-2.*Q33)*SMH3*CM*(Q12-Q22+2.*Q33)*CMH3*SM)*T+A(2,3)	00020930
	A(3,2)=A(2,3)	00020940
		00020950

[illegible]

```

B=BX
CO=(0.0,1.0)
RB11=(Q1-P1*R1)/(A-CO*R1*B)
RB21=(Q2-P2*R2)/(A-CO*R2*B)
REALR1=R1
REALR2=R2
REALP1=P1
REALP2=P2
REALQ1=Q1
REALQ2=Q2
RRB11=RB11
RRB21=RB21
AIMGR1=CO*R1
AIMGR2=CO*R2
AIMGP1=CO*P1
AIMGP2=CO*P2
AIMGQ1=CO*Q1
AIMGQ2=CO*Q2
ARB11=CO*RB11
ARB21=CO*RB21
R1B=DCMPLX(REALR1,AIMGR1)
R2B=DCMPLX(REALR2,AIMGR2)
P1B=DCMPLX(REALP1,AIMGP1)
P2B=DCMPLX(REALP2,AIMGP2)
Q1B=DCMPLX(REALQ1,AIMGQ1)
Q2B=DCMPLX(REALQ2,AIMGQ2)
RB11B=DCMPLX(RRB11,ARB11)
RB21B=DCMPLX(RRB21,ARB21)
JJJ=0
DO 1000 I=1,NB
J=J+2
THTAI=THTA(I)*DARCO5(-1.000)/180.00
C=DCOS(THTAI)
S=DSIN(THTAI)
P11=C*P1+S*Q1
P12=C*P2+S*Q2
P21=-S*P1+C*Q1
P22=-S*P2+C*Q2
T11=(C*C*R1/R1+S*S-2.*C*S*R1)
T12=(C*C*R2/R2+S*S-2.*C*S*R2)
T21=(-C*C*R1/R1+C*S-(C*C-S*S)*R1)
T22=(-C*C*R2/R2+C*S-(C*C-S*S)*R2)
Z1=X(I)+R1*Y(I)
Z2=X(I)+R2*Y(I)
Z11=CDSORT(Z1*Z1-A*A-R1*R1*B*B)
Z22=CDSORT(Z2*Z2-A*A-R2*R2*B*B)
REAL1=Z11
AIMG1=-CO*Z11
IF(DABS(REAL1).LE.1.D-14)REAL1=0.000
IF(DABS(AIMG1).LE.1.D-16)AIMG1=0.000
Z11=DCMPLX(REAL1,AIMG1)
REAL2=Z22
AIMG2=-CO*Z22
IF(DABS(REAL2).LE.1.D-16)REAL2=0.000
IF(DABS(AIMG2).LE.1.D-16)AIMG2=0.000
Z22=DCMPLX(REAL2,AIMG2)
XETA1=(Z1+Z11)/(A-CO*R1*B)
IF(CDABS(XETA1).LT.0.999) GO TO 300
GO TO 310
300 Z11=-Z11
  
```

```

00021560
00021570
00021580
00021590
00021600
00021610
00021620
00021630
00021640
00021650
00021660
00021670
00021680
00021690
00021700
00021710
00021720
00021730
00021740
00021750
00021760
00021770
00021780
00021790
00021800
00021810
00021820
00021830
00021840
00021850
00021860
00021870
00021880
00021890
00021900
00021910
00021920
00021930
00021940
00021950
00021960
00021970
00021980
00021990
00022000
00022010
00022020
00022030
00022040
00022050
00022060
00022070
00022080
00022090
00022100
00022110
00022120
00022130
00022140
00022150
  
```

CCC

```

CM(I,6*NT+1)=-CM(I,1)*RB21B/R311+CM(I,6*NT+1)
CM(I,1)=(0.0,0.0)
IMPOSE SINGLE-VALUEDNESS CONDITION
CM(I,NT8+3)=CM(I,NT8+1)*SV11+CM(I,NT8+3)
CM(I,NT8+4)=CM(I,NT8+1)*SV12+CM(I,NT8+4)
CM(I,NT8+3)=CM(I,NT8+2)*SV21+CM(I,NT8+3)
CM(I,NT8+4)=CM(I,NT8+2)*SV22+CM(I,NT8+4)
CM(I,NT8+1)=(0.0,0.0)
CM(I,NT8+2)=(0.0,0.0)
139 CONTINUE
DO 141 I=1,NB2
DO 142 J=2,NT8
142 CM(I,J-1)=CM(I,J)
CM(I,NT8)=CM(I,NT8+3)
CM(I,NT8+1)=CM(I,NT8+4)
141 CONTINUE
DO 95 I=1,NB2
DO 96 J=1,NT8P1
REAL1=CM(I,J)
AIM01=-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
CM(I,J)=DCMPLX(REAL1,AIM02)
96 CONTINUE
95 CONTINUE
DO 100 I=1,NT8P1
DO 100 J=1,NT8P1
CSUM=(0.0,0.0)
DO 110 K=1,NB2
110 CSUM=CMC(K,I)*CM(K,J)+CSUM
CMCTCM(I,J)=CSUM
100 CONTINUE
3335 CONTINUE
DO 120 I=1,NB
J=I+2
BC(J-1)=AIN1(I)
120 BC(J)=AIN2(I)
DO 130 I=1,NT8P1
CSUM=(0.0,0.0)
DO 140 K=1,NB2
140 CSUM=CMC(K,I)*BC(K)+CSUM
130 RHS(I)=CSUM
IJOB=0
IF(NOPY4.EQ.5.AND.NCASE.GT.1) IJGB=2
M=1
CALL LEQ2C(CMCTCM,NT8P1,NT8P1,RHS,M,NT8P1,IJOB,WA,WKK,IER)
IF(IER.EQ.129) WRITE(6,11)
11 FORMAT(' TERMINAL ERROR(CMCTCM), IER = 129')
GRHS(1)=-(RHS(2*NT)*RB21+RHS(4*NT)*RB11B+RHS(6*NT)*RB21B)/RB11
GRHS(8*NT+1)=RHS(8*NT)*SV11+RHS(8*NT+1)*SV12
GRHS(8*NT+2)=RHS(8*NT)*SV21+RHS(8*NT+1)*SV22
DO 151 I=2,NT8
151 GRHS(I)=RHS(I-1)
C
C
C STRESS AND STRAIN CALCULATION

```

```

00022760
00022770
00022780
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

```

```

RAD=DARCOS(-1.000)/180.00
IC=1
IC2=1
SUMU1=0.000
SUMV1=0.000
SUMU2=0.000
SUMV2=0.000
NADD=0
IF(NOPT4.EQ.1) GO TO 1195
IF(NTYPE.NE.2.OR.NCASE.GT.4) GO TO 1196
NADD=4*NCUL
NIC=1
DO 197 II=1,NADD
  XOUT(NOUT+NSTS+II)=X(II)
197 YOUT(NOUT+NSTS+II)=Y(II)
  MN=NOUT+NSTS+NADD
  XOUT(MN+1)=AX
  YOUT(MN+1)=0.000
  XOUT(MN+2)=0.000
  YOUT(MN+2)=-AX
  XOUT(MN+3)=AX*DCOS(177.00*RAD)
  YOUT(MN+3)=AX*DSIN(177.00*RAD)
  XOUT(MN+4)=0.000
  YOUT(MN+4)=AX
  NADD=NADD+4
  GO TO 1195
1196 CONTINUE
  NADD=8
  NIC=2
  MN=NOUT+NSTS
  DO 199 III=1,4
    ICM=5-III
    XOUT(MN+III)=XC(ICM)
    YOUT(MN+III)=YC(ICM)
199 CONTINUE
    MN=MN+4
    XOUT(MN+1)=AX
    YOUT(MN+1)=0.000
    XOUT(MN+2)=0.000
    YOUT(MN+2)=-AX
    XOUT(MN+3)=AX*DCOS(177.00*RAD)
    YOUT(MN+3)=AX*DSIN(177.00*RAD)
    XOUT(MN+4)=0.000
    YOUT(MN+4)=AX
1195 CONTINUE
    NRCF=NOUT
    IF(NOPT4.EQ.5) NRCF=NOUT+NSTS+NADD
    NINC=NSTS/4
    DO 190 K=1,NRCF
      Z1=XOUT(K)+R1*YOUT(K)
      Z2=XOUT(K)+R2*YOUT(K)
      Z11=CDSQRT(Z1*Z1-AXA-R1*NR1*MB)
      Z22=CDSQRT(Z2*Z2-AXA-R2*NR2*MB)
      XETA1=(Z1+Z11)/(A-CO*NR1*MB)
      IF(CDABS(XETA1).LT.0.999) GO TO 400
      GO TO 410
400 Z11=-Z11
      XETA1=(Z1+Z11)/(A-CO*NR1*MB)
410 XETA2=(Z2+Z22)/(A-CO*NR2*MB)
      IF(CDABS(XETA2).LT.0.999) GO TO 420

```

```

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

```

```

GO T) 430
420 Z22=-Z22
XETA1=(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=NP*XETA1*NN*GRHS(N)/Z11+PHI1DP
PHI1DN=NN*XETA1*NN*GRHS(N)/Z11+PHI1DN
PHI2DP=NP*XETA2*NN*GRHS(2*NT+N)/Z22+PHI2DP
PHI2DN=NN*XETA2*NN*GRHS(3*NT+N)/Z22+PHI2DN
PHI1P=XETA1*NN*GRHS(N)+PHI1P
PHI1N=XETA1*NN*GRHS(NT+N)+PHI1N
PHI2P=XETA2*NN*GRHS(2*NT+N)+PHI2P
PHI2N=XETA2*NN*GRHS(3*NT+N)+PHI2N
170 CONTINUE
PHI1D=PHI1DP+PHI1DN+GRHS(3*NT+1)/Z11
PHI2D=PHI2DP+PHI2DN+GRHS(8*NT+2)/Z22
PHI1=PHI1P+PHI1N+GRHS(8*NT+1)*CDLOG(XETA1)
PHI2=PHI2P+PHI2N+GRHS(8*NT+2)*CDLOG(XETA2)
SGMAX=2.*(R1*R1*PHI1D+R2*R2*PHI2D)
SGMAY=2.*(PHI1D+PHI2D)
SGMAXY=-2.*(R1*PHI1D+R2*PHI2D)
EPSX=AMAT(1,1)*SGMAX+AMAT(1,2)*SGMAY+AMAT(1,3)*SGMAXY
EPSY=AMAT(2,1)*SGMAX+AMAT(2,2)*SGMAY+AMAT(2,3)*SGMAXY
EPSXY=AMAT(3,1)*SGMAX+AMAT(3,2)*SGMAY+AMAT(3,3)*SGMAXY
U=2.*(P1*PHI1+P2*PHI2)
V=2.*(Q1*PHI1+Q2*PHI2)
PI=DARCO5(-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
EPST=S**2*EPSX+C**2*EPSY-C*S*EPSXY
EPSRT=2.*(-C*S*EPSX+C*S*EPSY+(C**2-S**2)*(EPSXY/2.))
UR=UXC+VXS
IF(NOPT4.EQ.5) GO TO 3338
KTHT(K)=TETAA
REPX(K)=EPSX
REPY(K)=EPSY
REPLY(K)=EPSXY
ASX(K)=SGMAX
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

```

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
APSY(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
190 CONTINUE	00024960
DISPLACEMENTS ARE AVERAGED OVER ELEMENT SIDES FOR	00024970
CERTAIN LOAD CASES	00024980
IF(NIC.NE.1) RETURN	00024990
SUMU1=SUMU1/FLOAT(NCOL)	00025000
SUMV1=SUMV1/FLOAT(NCOL)	00025010
SUMU2=SUMU2/FLOAT(NCOL)	00025020
SUMV2=SUMV2/FLOAT(NCOL)	00025030
UVOUT(1)=SUMU2	00025040
UVOUT(2)=SUMV2	00025050
UVOUT(3)=SUMU2	00025060
UVOUT(4)=SUMV1	00025070
UVOUT(5)=SUMU1	00025080
UVOUT(6)=SUMV1	00025090
UVOUT(7)=SUMU1	00025100
UVOUT(8)=SUMV2	00025110
RETURN	00025120
END	00025130
	00025140
	00025150

00000000

SUBROUTINE FBOLT(ANGK,H,PSH,K)

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,HOUT
COMMON/FB1/BSTR,YSTR
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

WORK=0.
PLOADX=0.
IF(K.EQ.1) PLD=0.
DO 210 KK=1,NOUT
TH1=ATETAA(KK+1)-ATETAA(KK)
TH2=(ATETAA(KK)+ATETAA(KK+1))/2.
THETA=TH2*RAD
C=DCOS(THETA)
S=DSIN(THETA)
R=DSQRT(1./((C**2/AX**2+S**2/BX**2))
FORCE=((FSMR(KK)+FSMR(KK+1))/2.)*R*TH1*RAD*THKTOT
WORK=WORK+FORCE*.5*((UR(KK)+UR(KK+1))/2.)
PLOADX=PLOADX+FORCE*C

210 CONTINUE
PLD=PLD+PLOADX
DELEF=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

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

00000000

LI2=NOUT
 NCAS=1
 CALL QMATX(RAD,THETA,K,LI1,LI2,NCAS)

INTEGRATE AROUND CIRCULAR BOUNDARY FOR
 INDIVIDUAL PLY LOADS AND COMPUTE FOUNDATION
 MODULI

```

NNH=LI2-1
PLOADX=0.
WK=0.
DO 70 I=LI1,NNH
  TH1=ATETAA(I+1)-ATETAA(I)
  TH2=(ATETAA(I)+ATETAA(I+1))/2.
  THETA=TH2*H*AD
  C=DCOS(THETA)
  S=DSIN(THETA)
  R=DSQRT(1./(C**2/AX**2+S**2/BX**2))
  FORCR=((ASIGR(I)+ASIGR(I+1))/2.)*R*TH1*RAD*H(K)
  FORCRT=((ASIORT(I)+ASIORT(I+1))/2.)*R*TH1*RAD*H(K)
  PLOADX=PLOADX+FORCR*C-FORCRT*S
70 CONTINUE
FKI(J)=DABS(PLOADX/(H(K)*DELEFF))
PLX(J+(K-1)*NPLY(1))=I*PLOADX
100 CONTINUE
NT=NPLY(K)
NN=NPLY(K)
DO 310 I=1,NT
  DO 310 II=1,NN
    IF(IPLY(II,K).EQ.I) A'IGK(I,K)=FKI(II)
    IF(IPLY(II,K).EQ.I) PLXPT(II)=PLX(II+(K-1)*NPLY(1))
310 CONTINUE
NP=NPLY(K)
DO 311 I=1,NP
  AA=ANG(I,K)+PSH
311 CONTINUE
PLXTOT=0.000
IF(K.EQ.1) BLOAD=0.
TH=H(K)*NPLY(K)
BLOAD=(BSTR*DARCOS(-1.000)*BX*TH)/2.+BLOAD
IF(K.EQ.1) GO TO 611
NN=NPLY(1)+NPLY(2)
DO 212 I=1,NN
  PLXTOT=PLXTOT+PLX(I)
212 CONTINUE
611 CONTINUE
RETURN
END

```

00000000

SUBROUTINE QMATX(RAD,THETA,K,LI1,LI2,NCAS)

QMATX PERFORMS BASIC STRESS AND STRAIN
 TRANSFORMATIONS

IMPLICIT REAL*8(A-H,O-Z)
 DIMENSION ASIGR(400),ASIORT(400),ASIG1(400),ASIG2(400),ASIG6(400)

00025760
 00025770
 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

```

DIMENSION ATETAA(400),AEP5X(400),AEP5Y(400),AEP5XY(400) 00026360
DIMENSION E11(2),E22(2),ESS(2),PMU12(2),PMU21(2),SX(400),SXY(400) 00026370
DIMENSION AEP51(400) 00026380
DIMENSION ASX(400),ASXY(400) 00026390
COMMON/XXY1/ASX,ASXY 00026400
COMMON/MD0/E11,E22,ESS,PMU12,PMU21 00026410
COMMON/STRESS2/AEP51 00026420
COMMON/STRESS/ASIOR,ASIOR1,ASIOR2,ASIOR3 00026430
COMMON/CMT/ATETAA,AEP5X,AEP5Y,AEP5XY 00026440
COMMON/PSC3/SX,SXY 00026450
J=0 00026460
Q11=E11(K)/(1.0-PMU12(K)*PMU21(K)) 00026470
Q12=(PMU21(K)*E11(K))/(1.0-PMU12(K)*PMU21(K)) 00026480
Q22=E22(K)/(1.0-PMU12(K)*PMU21(K)) 00026490
Q66=E66(K) 00026500
C=DCOS(THETA) 00026510
S=DSIN(THETA) 00026520
BQ11=(Q11*(S**4))+(2.*(Q12*(2.*Q66))*(C**2)*(S**2))+(Q22*(S**4)) 00026530
BQ12=((Q11+Q22-(4.*Q66))*(S**2)*(C**2))+(Q12*(S**4+C**4)) 00026540
BQ14=((Q11-Q12-(2.*Q66))*(S*(C**3)))+((Q12-Q22+(2.*Q66))*(S**3)*C 00026550
*) 00026560
BQ22=(Q11*(S**4))+(2.*(Q12*(2.*Q66))*(S**2)*(C**2))+(Q22*(C**4)) 00026570
BQ26=((Q11-Q12-(2.*Q66))*C*(S**3))+(Q12-Q22+(2.*Q66))*S*(C**3)) 00026580
BQ66=((Q11+Q22-(2.*(Q12+Q66)))*(S**2)*(C**2))+(Q66*(C**4)+(S**4) 00026590
*) 00026600
DO 40 I=111,112 00026610
J=J+1 00026620
IF(NCAS.EQ.1) THETA=ATETAA(I)*RAD 00026630
C=DCOS(THETA) 00026640
S=DSIN(THETA) 00026650
SIGX=BQ11*AEP5X(I)+BQ12*AEP5Y(I)+BQ14*AEP5XY(I) 00026660
SIGY=BQ12*AEP5X(I)+BQ22*AEP5Y(I)+BQ26*AEP5XY(I) 00026670
SIGXY=BQ14*AEP5X(I)+BQ26*AEP5Y(I)+BQ66*AEP5XY(I) 00026680
SX(J)=SIGX 00026690
SXY(J)=SIGY 00026700
ASIGR(I)=SIGX*C**2+SIGY*S**2+2.*SIGXY*S*C 00026710
ASIGRT(I)=-SIGX*S*C+SIGY*C*S+SIGXY*(C**2-S**2) 00026720
ASIG1(J)=SIGX*C**2+SIGY*S**2+2.*S*C*SIGXY 00026730
ASIG2(J)=SIGX*S**2+SIGY*C**2-2.*S*C*SIGXY 00026740
ASIG6(J)=-C*S*SIGX+SIGY*C*S+(C**2-S**2)*SIGXY 00026750
AEP51(J)=AEP5X(I)*C**2+AEP5Y(I)*S**2+AEP5XY(I)*S*C 00026760
40 CONTINUE 00026770
RETURN 00026780
END 00026790

SUBROUTINE CENTD(H,FASSS,FASBS,P) 00026800
00026810
00026820
00026830
00026840
00026850
00026860
00026870
00026880
00026890
00026900
00026910
00026920
00026930
00026940
00026950
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION PLYK(100),BARK(100),BARU(100),F(100)
DIMENSION H(2),RF(2)
DIMENSION AII(100,100),A(2),B(2)
DIMENSION NPLY(2)
COMMON/PBB/PLYK,BARK,BARU
COMMON/RT/RF
COMMON/AFM/AII,F

```

	COMMON/LYP/NPLY	00026960
	NNN=NPLY(1)+NPLY(2)	00026970
		00026980
	SET UP THE CENTRAL DIFFERENCE EQUATIONS	00026990
		00027000
	DO 3 I=1,100	00027010
	DO 3 J=1,100	00027020
	3 AII(I,J)=0.	00027030
		00027040
	NECESSARY CONSTANTS ARE FORMED	00027050
		00027060
	DO 7 I=1,...	00027070
	A(I)=H(I)**2/FASSS	00027080
	7 B(I)=H(I)**4/FASBS	00027090
	H12=H(1)/H(2)	00027100
	A1=H(1)**2/FASSS	00027110
	A2=H(2)**2/FASSS	00027120
	NP=NPLY(1)+NPLY(2)	00027130
		00027140
		00027150
	SHEAR AT TOP OF PANEL EQUALS ZERO	00027160
		00027170
	AII(1,1)=1.	00027180
	AII(1,2)=-2.+A1*PLYK(2)	00027190
	AII(1,4)=2.+A1*PLYK(2)	00027200
	AII(1,5)=-1.	00027210
	F(1)=0.0	00027220
		00027230
	MOMENT CONDITION AT TOP	00027240
		00027250
	IF(RF(1).OE.1.D10) GO TO 50	00027260
	Z=1.	00027270
	R=RF(1)	00027280
	GO TO 60	00027290
50	Z=0.	00027300
	R=1.	00027310
60	AII(2,1)=R	00027320
	AII(2,2)=(Z**2.*H(1)*FASSS)*RM*(-2.-A1*PLYK(2)+(H(1)**2	00027330
	*FASSS)/FASBS)	00027340
	AII(2,3)=-Z*(4.*H(1)*FASSS+(2*H(1)**2*PLYK(1)*H(1)))	00027350
	AII(2,4)=Z**2.*H(1)*FASSS+RM*(2.+A1*PLYK(2)-(H(1)**2	00027360
	*FASSS)/FASBS)	00027370
	AII(2,5)=-R	00027380
	F(2)=Z**2.*H(1)**3*BARK(1)*BARU(1)	00027390
		00027400
	GOVERNING EQUATIONS FOR THE TOP PLATE	00027410
		00027420
	N2=NPLY(1)	00027430
	DO 55 J=1,N2	00027440
	I=J+2	00027450
	AII(I,J)=1.	00027460
	IF(J.EQ.1) GO TO 56	00027470
	AII(I,J+1)=-4.-A(1)*PLYK(J-1)	00027480
	GO TO 57	00027490
56	AII(I,J+1)=-4.-A(1)*PLYK(2)	00027500
57	AII(I,J+2)=6.+(2.*A(1)+B(1))*PLYK(J)	00027510
	IF(J.EQ.N2) GO TO 61	00027520
	AII(I,J+3)=-4.-A(1)*PLYK(J+1)	00027530
	GO TO 62	00027540
61	AII(I,J+3)=-4.-A(1)*PLYK(NPLY(1)-1)	00027550

	62	ALL(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
		M=(2.*A(1)+B(1))*BARK(J)*BARU(J)	00027600
		M+A(1)*BARK(J+1)*BARU(J+1)	00027610
		GO TO 59	00027620
	58	F(I)=2.*A(1)*BARK(2)*BARU(2)	00027630
		M=(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)	00027660
		M=(2.*A(1)+B(1))*BARK(J)*BARU(J)	00027670
	59	CONTINUE	00027680
	55	CONTINUE	00027690
C		INTERFACE SHEAR ON TOP PLATE = P	00027700
C		I=NPLY(1)+3	00027710
		J=NPLY(1)	00027720
		ALL(I,J)=1.	00027730
		ALL(I,J+1)=-((2.+A1*PLYK(NPLY(1)-1))	00027740
		ALL(I,J+3)=2.+A1*PLYK(NPLY(1)-1)	00027750
		ALL(I,J+4)=-1.	00027760
		F(I)=(-2.*H(1)*N3*(P))/FASBS	00027770
C		SLOPE CONTINUITY	00027780
C		I=NPLY(1)+4	00027790
		J=NPLY(1)	00027800
		ALL(I,J)=1.	00027810
		ALL(I,J+1)=-((2.+A1*PLYK(NPLY(1)-1)-H(1)*N2*FASBS/FASBS)	00027820
		ALL(I,J+3)=2.+A1*PLYK(NPLY(1)-1)-H(1)*N2*FASBS/FASBS	00027830
		ALL(I,J+4)=-1.	00027840
		ALL(I,J+5)=-H12*N3	00027850
		ALL(I,J+6)=H12*N3*(2.+A2*PLYK(NPLY(1)+2)-H(2)*N2*FASBS/FASBS)	00027860
		ALL(I,J+8)=-H12*N3*(2.+A2*PLYK(NPLY(1)+2)-H(2)*N2*FASBS/FASBS)	00027870
		ALL(I,J+9)=H12*N3	00027880
		F(I)=0.	00027890
C		MOMENT CONTINUITY	00027900
C		I=NPLY(1)+5	00027910
		J=NPLY(1)+1	00027920
		ALL(I,J)=1.	00027930
		ALL(I,J+1)=-((2.+A1*PLYK(NPLY(1)))	00027940
		ALL(I,J+2)=1.	00027950
		ALL(I,J+5)=-H12*N2	00027960
		ALL(I,J+6)=H12*N2*(2.+A2*PLYK(NPLY(1)+1))	00027970
		ALL(I,J+7)=-H12*N2	00027980
		F(I)=A1*(BARK(NPLY(1))*BARU(NPLY(1))-BARK(NPLY(1)+1)*	00027990
		BARU(NPLY(1)+1))	00028000
C		INTERFACE SHEAR ON BOTTOM PLATE	00028010
C		I=NPLY(1)+6	00028020
		J=NPLY(1)+5	00028030
		ALL(I,J)=-1.	00028040
		ALL(I,J+1)=(2.+A2*PLYK(NPLY(1)+2))	00028050
		ALL(I,J+3)=-((2.+A2*PLYK(NPLY(1)+2))	00028060
		ALL(I,J+4)=1.	00028070
			00028080
			00028090
			00028100
			00028110
			00028120
			00028130
			00028140
			00028150

```

C
C
C
F(I)=2.*H(2)*NPLY(1)*NPLY(2)
GOVERNING EQUATIONS FOR THE BOTTOM PLATE
N1=NPLY(1)+7
N2=NPLY(1)+NPLY(2)+6
DO 70 I=N1,N2
  J=I-2
  AII(I,J)=1.
  IF(I.EQ.N1) GO TO 71
  AII(I,J-1)=-4.-A(2)*PLYK(J-5)
  AII(I,J)=2.
71 AII(I,J+1)=-4.-A(2)*PLYK(NPLY(1)+2)
72 AII(I,J+2)=6.+(2.*A(2)+B(2))*PLYK(J-4)
  IF(I.EQ.N2) GO TO 73
  AII(I,J-3)=-4.-A(2)*PLYK(J-3)
  AII(I,J-2)=-4.-A(2)*PLYK(J-5)
75 AII(I,J+3)=4.-A(2)*PLYK(J-5)
75 AII(I,J+4)=1.
  IF(I.EQ.N1) GO TO 73
  IF(I.EQ.N2) GO TO 77
  F(I)=A(2)*BARK(J-5)*BARU(J-5)
  X=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
  Y=A(2)*BARK(J-3)*BARU(J-5)
  GO TO 74
73 F(I)=2.*A(2)*BARK(NPLY(1)+2)*BARU(NPLY(1)+2)
  X=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
  GO TO 74
77 F(I)=2.*A(2)*BARK(J-5)*BARU(J-5)
  X=(2.*A(2)+B(2))*BARK(J-4)*BARU(J-4)
74 CONTINUE
70 CONTINUE
C
C
C
SHEAR ON BOTTOM PLATE EQUALS ZERO
NP=NPLY(1)+NPLY(2)
I=NP+7
J=NP+4
AII(I,J)=-1.
AII(I,J+1)=(2.+A2)*PLYK(NP-1)
AII(I,J+3)=-2.+A2*PLYK(NP-1)
AII(I,J+4)=1.
F(I)=0.
C
C
C
MOMENT BOUNDARY CONDITION ON BOTTOM PLATE
I=NP+8
IF(RF(2).GE.1.D10) GO TO 85
Z=1.
R=RF(2)
GO TO 95
85 Z=0.
R=1.
95 AII(I,J)=-R
  AII(I,J+1)=Z*(2.*H(2)*FASSS)+R*(2.+A2*PLYK(NP-1))
  X=H(2)*M2*FASSS/FASBS
  AII(I,J+2)=-Z*(4.*H(2)*FASSS+2.*H(2)*M3*PLYK(NP))
  AII(I,J+3)=Z*2.*H(2)*FASSS+R*(-2.-A2*PLYK(NP-1))
  X=H(2)*M2*FASSS/FASBS
  AII(I,J+4)=R
  
```

00028140
 00028170
 00028130
 00028170
 00028200
 00028210
 00028220
 00028230
 00028240
 00028250
 00028260
 00028270
 00028280
 00028290
 00028300
 00028310
 00028320
 00028330
 00028340
 00028350
 00028360
 00028370
 00028380
 00028390
 00028400
 00028410
 00028420
 00028430
 00028440
 00028450
 00028460
 00028470
 00028480
 00028490
 00028500
 00028510
 00028520
 00028530
 00028540
 00028550
 00028560
 00028570
 00028580
 00028590
 00028600
 00028610
 00028620
 00028630
 00028640
 00028650
 00028660
 00028670
 00028680
 00028690
 00028700
 00028710
 00028720
 00028730
 00028740
 00028750

	F(I)=Z*(2.*H(2)*X3+JARK(NP)*BARU(NP))	00028760
	RETURN	00028770
	END	00028780
		00028790
	SUBROUTINE SOLVE(H,P,U1,U2)	00028800
		00028810
		00028820
		00028830
		00028840
		00028850
	IMPLICIT REAL*8(A-H,O-Z)	00028860
	DIMENSION A(100,100),B(100),NPLY(2),U(100),F(100)	00028870
	DIMENSION SX(100),PLYX(100),H(2)	00028880
	DIMENSION JARK(100),BARU(100)	00028890
	COMMON/PLP/NPLY	00028900
	COMMON/AFMA,A,F	00028910
	COMMON/P33/PLYX,JARK,BARU	00028920
		00028930
	SOLUTION OF THE SYSTEM: PA(U)=(3)	00028940
	NP=NPLY(1)+NPLY(2)+8	00028950
	DO 444 I=1,NP	00028960
444	B(I)=F(I)	00028970
		00028980
		00028990
	APPLYING GUASSIAN ELIMINATION TO THE	00029000
	MATRIX OF COEFFICIENTS	00029010
		00029020
	DO 2001 I=1,NP	00029030
	IR=I	00029040
2042	IF(A(IR,I).NE.0.) GO TO 2041	00029050
	IR=IR+1	00029060
	IF(IR.GT.NP) GO TO 2001	00029070
	GO TO 2042	00029080
2041	NN=IR+1	00029090
	DO 2002 L=NN,NP	00029100
	IF(DABS(A(L,I)).GT.1.D-30) GO TO 2009	00029110
	A(L,I)=0.	00029120
	GO TO 2002	00029130
2009	CF=-A(IR,I)/A(L,I)	00029140
	DO 2003 J=I,NP	00029150
	A(L,J)=A(L,J)+CF*A(IR,J)	00029160
	IF(DABS(A(L,J)).LT.1.D-30) A(L,J)=0.0	00029170
2003	CONTINUE	00029180
	B(L)=B(L)+CF*B(I)	00029190
2002	CONTINUE	00029200
2001	CONTINUE	00029210
		00029220
	BACK SUBSTITUTION	00029230
		00029240
	DO 2011 I=1,NP	00029250
	L=NP+1-I	00029260
	SUM=0.	00029270
	IF(A(L,L).EQ.0.) GO TO 2112	00029280
	N=L+1	00029290
	IF(N.GT.NP) GO TO 2013	00029300
	DO 2013 J=N,NP	00029310
	SUM=SUM-A(L,J)*SX(J)	00029320
2013	CONTINUE	00029330
	SX(L)=(B(L)+SUM)/A(L,L)	00029340
	GO TO 2011	00029350

```

2112 CONTINUE
      SX(L)=0.
2011 CONTINUE
      PT=P
      N1=NPLY(1)+2
      N2=NPLY(1)+7
      NN=NPLY(1)+NPLY(2)+6
      DO 1444 I=3,N1
      J=I-2
      U(J)=SX(I)
1444 CONTINUE
      DO 1555 I=N2,NN
      J=I-6
      U(J)=SX(I)
1555 CONTINUE
      NP=NPLY(1)+NPLY(2)

      COMPUTE AVERAGE RELATIVE DISPLACEMENTS
      IN TOP AND BOTTOM PLATES

      U1=DABS(U(1)-U(NPLY(1)))/2.
      U2=DABS(U(NPLY(1)+1)-U(NPLY(1)+NPLY(2)))/2.
      RETURN
      END

      SUBROUTINE FCRT(APP,NEL1,NEL2,NDAM,IN,LTNCM,NAVD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ELSTFF(50,10,10),ELCTSS(50,50,10),U(200)
      DIMENSION OSSX(20),OSSW(20)
      DIMENSION NELDIS(50,5,2)
      DIMENSION PSMX(50,4),AVES(50,3),STRSS(50),DLT(10)
      DIMENSION ELFAIL(50,3),NELTYP(50)
      DIMENSION NELCON(50,6),NELCHA(50,6),NPLY(2)
      DIMENSION ELWDTH(50),ELTHK(50),ELLOAD(50,2)
      DIMENSION NELPLS(2,50),LYPN(50)
      COMMON/ELS/ELSTFF,ELSTSS
      COMMON/OSXW/OSSX,OSSW
      COMMON/NPLS/NELPLS,LYPN
      COMMON/NCH/NELCON,NELCHA,NELDIS
      COMMON/FCC/ELWDTH,ELTHK,ELLOAD
      COMMON/SMX/PSMX
      COMMON/LAMF/ELFAIL
      COMMON/DISP/U
      COMMON/NTP/NELTYP
      COMMON/LYP/NPLY

      DETERMINE ELEMENT FAILURE LOADS IN NET SECTION,
      SHEAROUT AND BEARING, AND LOCATE THE CRITICAL
      FASTENER LOCATION. JOINT STRENGTH IS DETERMINED
      FROM LOWEST ELEMENT FAILURE LOAD

      NELTOT=NEL1+NEL2
      NS=NAVD
      NSTS=4*NAVD
      DO 10 I=1,NELTOT
      NRNK=10
      IF(NELTYP(I).EQ.3) NRNK=8
      KJ=I
  
```

```

00029360
00029370
00029380
00029390
00029400
00029410
00029420
00029430
00029440
00029450
00029460
00029470
00029480
00029490
00029500
00029510
00029520
00029530
00029540
00029550
00029560
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
  
```

IF(KJ.GT.NEL) KJ=KJ-NEL	C0029960
IF(NELTYP(I).EQ.1) GO TO 10	00029970
IC=0	00029980
DO 20 J=1,5	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=0.000	00030060
DO 40 K2=1,NRNK	00030070
SUM=SUM+ELSTS3(I,K,K2)*DLT(K2)	00030080
40 CONTINUE	00030090
30 STRSS(K)=SUM	00030100
SUM1=0.000	00030110
SUM2=0.000	00030120
SUM3=0.000	00030130
SUM4=0.000	00030140
DO 50 J=1,NS	00030150
SUM1=SUM1+STRSS(J)	00030160
SUM2=SUM2+STRSS(J+NS)	00030170
SUM3=SUM3+STRSS(J+2*NS)	00030180
50 CONTINUE	00030190
NR=2*NS	00030200
DO 51 II=1,NS	00030210
51 SUM4=SUM4+STRSS(II+NS)*MOSSW(II)	00030220
AVES(I,1)=SUM1/NS	00030230
AVES(I,2)=SUM2/NS	00030240
AVES(I,3)=SUM3/NS	00030250
IF(I.LE.NEL1) THK=ELTHK(I)*NELPLS(1,LYPN(I))	00030260
IF(I.GT.NEL1) THK=ELTHK(I)*NELPLS(2,LYPN(I))	00030270
ELD=(SUM4/2.00)*ELHDTH(I)*THK	00030280
PRATIO=DABS((ELLOAD(I,1)+ELLOAD(I,2))/ELD)	00030290
C C C SCALE AVERAGE STRESSES	00030300
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 C COMPUTE JOINT FAILURE LOADS BASED ON ELEMENT LOADS	00030350
DO 100 I=1,NELTOT	00030360
IF(NELTYP(I).EQ.1) GO TO 100	00030370
DO 110 J=1,3	00030380
N=J+1	00030390
IF(J.EQ.1.AND.LTNCM.EQ.1) N=1	00030400
IF(J.EQ.1.AND.LTNCM.EQ.2) N=2	00030410
ELFAIL(I,J)=DABS(APP*PSMX(I,N)/AVES(I,J))	00030420
110 CONTINUE	00030430
100 CONTINUE	00030440
C C C SEARCH FOR LOWEST JOINT FAILURE LOAD	00030450
INNS=0	00030460
FNS=1.0D10	00030470
INCO=0	00030480
	00030490
	00030500
	00030510
	00030520
	00030530
	00030540
	00030550

0000000000

C	SUBROUTINE LEQT2F (A,M,N,IA,B,IDOT,WKAREA,IER)	00031160
C	DIMENSION A(IA,1),B(IA,1),WKAREA(1)	00031170
C	DOUBLE PRECISION A,B,WKAREA,D1,D2,WA	00031180
C	FIRST EXECUTABLE STATEMENT	00031190
C	INITIALIZE IER	00031200
	IER=0	00031210
	JER=0	00031220
	J = NXN+1	00031230
	K = J+N	00031240
	MM = K+N	00031250
	KK = 0	00031260
	MM1 = MM-1	00031270
	JJ=1	00031280
	DO 5 L=1,N	00031290
	DO 5 I=1,N	00031300
	WKAREA(JJ)=A(I,L)	00031310
	JJ=JJ+1	00031320
5	CONTINUE	00031330
C	DECOMPOSE A	00031340
	CALL LUDATN (WKAREA,N,N,A,IA,IDOT,D1,D2,WKAREA(J),WKAREA(K),	00031350
	WA,IER)	00031360
	IF (IER.GT.128) GO TO 25	00031370
	IF (IDOT.EQ. 0 .OR. IER .NE. 0) KK = 1	00031380
	DO 15 I = 1,M	00031390
C	PERFORMS THE ELIMINATION PART OF	00031400
C	AX = B	00031410
C	CALL LUEL MN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))	00031420
C	REFINEMENT OF SOLUTION TO AX = B	00031430
	IF (KK .NE. 0)	00031440
	CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDOT,WKAREA(J),WKAREA(MM),	00031450
	WKAREA(K),WKAREA(K),JER)	00031460
	DO 10 II=1,N	00031470
	B(II,I) = WKAREA(MM1+II)	00031480
10	CONTINUE	00031490
	IF (JER.NE.0) GO TO 20	00031500
15	CONTINUE	00031510
	GO TO 25	00031520
20	IER = 131	00031530
25	JJ=1	00031540
	DO 30 J = 1,N	00031550
	DO 30 I = 1,M	00031560
	A(I,J)=WKARFA(JJ)	00031570
	JJ=JJ+1	00031580
30	CONTINUE	00031590
	IF (IER.EQ. 0) GO TO 9005	00031600
9000	CONTINUE	00031610
	CALL UERTST (IER,6HLEQT2F)	00031620
9005	RETURN	00031630
	END	00031640
C		00031650
C		00031660
C		00031670
C		00031680
C		00031690
	SUBROUTINE LUDATF (A,LU,N,IA,IDOT,D1,D2,IPVT,EQUIL,WA,IER)	00031700
	DIMENSION A(IA,1),IU(IA,1),IPVT(1),EQUIL(1)	00031710
	DOUBLE PRECISION A,LU,D1,D2,EQUIL,WA,ZERO,ONE,FOUR,SIXTN,SIXTH,	00031720
	RN,WREL,BIGA,BIG,P,SUM,AI,WI,T,TEST,Q	00031730
	DATA ZERO,ONE,FOUR,SIXTN,SIXTH/0.DO,1.DO,4.DO,	00031740
		00031750

```

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

```

```

00031760
00031770
00031780
00031790
00031800
00031810
00031820
00031830
00031840
00031850
00031860
00031870
00031880
00031890
00031900
00031910
00031920
00031930
00031940
00031950
00031960
00031970
00031980
00031990
00032000
00032010
00032020
00032030
00032040
00032050
00032060
00032070
00032080
00032090
00032100
00032110
00032120
00032130
00032140
00032150
00032160
00032170
00032180
00032190
00032200
00032210
00032220
00032230
00032240
00032250
00032260
00032270
00032280
00032290
00032300
00032310
00032320
00032330
00032340
00032350

```

```

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

```

00032360
 00032370
 00032380
 00032390
 00032400
 00032410
 00032420
 00032430
 00032440
 00032450
 00032460
 00032470
 00032480
 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

110	IER = 129	00032960
	D1 = ZERO	00032970
	D2 = ZERO	00032980
9000	CONTINUE	00032990
	PRINT ERRJR	00033000
	CALL UERTST(IER,6HLUDATF)	00033010
9005	RETURN	00033020
	END	00033030
C		00033040
C		00033050
	SUBROUTINE LUELMN (A,IA,N,B,APVT,X)	00033060
C		00033070
	DIMENSION A(IA,1),B(1),APVT(1),X(1)	00033080
	DOUBLE PRECISION A,B,X,SUM,APVT	00033090
C		00033100
	FIRST EXECUTABLE STATEMENT	00033110
	SOLVE LY = B FOR Y	00033120
	DO 5 I=1,N	00033130
5	X(I) = B(I)	00033140
	IW = 0	00033150
	DO 20 I=1,N	00033160
	IP = APVT(I)	00033170
	SUM = X(IP)	00033180
	X(IP) = X(I)	00033190
	IF (IW .EQ. 0) GO TO 15	00033200
	IM1 = I-1	00033210
	DO 10 J=IW,IM1	00033220
	SUM = SUM-A(I,J)*X(J)	00033230
10	CONTINUE	00033240
	GO TO 20	00033250
15	IF (SUM .NE. 0.DO) IW = I	00033260
20	X(I) = SUM	00033270
C		00033280
	SOLVE UX = Y FOR X	00033290
	DO 30 IB=1,N	00033300
	I = N+1-IB	00033310
	IP1 = I+1	00033320
	SUM = X(I)	00033330
	IF (IP1 .GT. N) GO TO 30	00033340
	DO 25 J=IP1,N	00033350
	SUM = SUM-A(I,J)*X(J)	00033360
25	CONTINUE	00033370
30	X(I) = SUM/A(I,I)	00033380
	RETURN	00033390
	END	00033400
C		00033410
C		00033420
	SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDGT,APVT,X,RES,DX,IER)	00033430
C		00033440
	DIMENSION A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1)	00033450
	DIMENSION APVT(1)	00033460
	DIMENSION ACCXT(2)	00033470
	DOUBLE PRECISION A,ACCXT,B,UL,X,RES,DX,ZERO,XNORM,DXNORM,APVT	00033480
	DATA ITMAX/75/,ZERO/0.DO/	00033490
C		00033500
	FIRST EXECUTABLE STATEMENT	00033510
	IER=0	00033520
	XNORM = ZERO	00033530
	DO 10 I=1,N	00033540
	XNORM = DMAX1(XNORM,DABS(X(I)))	00033550
10	CONTINUE	
	IF (XNORM .NE. ZERO) GO TO 20	
	IDGT = 50	

```

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

```

00033560
 00033570
 00033580
 00033590
 00033600
 00033610
 00033620
 00033630
 00033640
 00033650
 00033660
 00033670
 00033680
 00033690
 00033700
 00033710
 00033720
 00033730
 00033740
 00033750
 00033760
 00033770
 00033780
 00033790
 00033800
 00033810
 00033820
 00033830
 00033840
 00033850
 00033860
 00033870
 00033880
 00033890
 00033900
 00033910
 00033920
 00033930
 00033940
 00033950
 00033960
 00033970
 00033980
 00033990
 00034000
 00034010
 00034020
 00034030
 00034040
 00034050
 00034060
 00034070
 00034080
 00034090
 00034100
 00034110
 00034120
 00034130
 00034140
 00034150

```

    IF (IEQDF.EQ.1) WRITE(IUNIT,40) IER,NAMEQ,IEQ,NAMUPK      00034160
    IF (IEQDF.EQ.1) WRITE(IUNIT,40) IER,NAMUPK                00034170
    GO TO 30                                                    00034180
10 IF (IER.LE.32) GO TO 15                                     00034190
C   PRINT WARNING MESSAGE                                     00034200
    IF (LEVEL.LT.3) GO TO 30                                    00034210
    IF (IEQDF.EQ.1) WRITE(IUNIT,45) IER,NAMEQ,IEQ,NAMUPK      00034220
    IF (IEQDF.EQ.0) WRITE(IUNIT,45) IER,NAMUPK                 00034230
    GO TO 30                                                    00034240
15 CONTINUE                                                    00034250
C   CHECK FOR UERSET CALL                                     00034260
    DO 20 I=1,6                                                 00034270
        IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25                  00034280
20 CONTINUE                                                    00034290
    LEVOLD = LEVEL                                              00034300
    LEVEL = IER                                                 00034310
    IER = LEVOLD                                                00034320
    IF (LEVEL.LT.0) LEVEL = 4                                   00034330
    IF (LEVEL.GT.4) LEVEL = 4                                   00034340
    GO TO 30                                                    00034350
25 CONTINUE                                                    00034360
    IF (LEVEL.LT.4) GO TO 30                                    00034370
C   PRINT NON-DEFINED MESSAGE                                  00034380
    IF (IEQDF.EQ.1) WRITE(IUNIT,50) IER,NAMEQ,IEQ,NAMUPK      00034390
    IF (IEQDF.EQ.0) WRITE(IUNIT,50) IER,NAMUPK                 00034400
30 IEQDF = 0                                                    00034410
    RETURN                                                      00034420
35 FORMAT(19H *** TERMINAL ERROR,10X,7H( IER = ,I3,          00034430
1   20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                      00034440
40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H( IER = ,I3,    00034450
1   20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                      00034460
45 FORMAT(18H *** WARNING ERROR,11X,7H( IER = ,I3,           00034470
1   20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                      00034480
50 FORMAT(20H *** UNDEFINED ERROR,9X,7H( IER = ,I5,          00034490
1   20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                      00034500
C   SAVE P FOR P = R CASE                                     00034510
C   P IS THE PAGE NAMUPK                                     00034520
C   R IS THE ROUTINE NAMUPK                                  00034530
55 IEQDF = 1                                                    00034540
    DO 60 I=1,6                                                 00034550
60 NAMEQ(I) = NAMUPK(I)                                         00034560
65 RETURN                                                       00034570
    END                                                         00034580
C   SUBROUTINE UGETIO(IOP1,NIN,NOUT)                          00034590
C   SPECIFICATIONS FOR ARGUMENTS                             00034600
C   INTEGER IOP1,NIN,NOUT                                     00034610
C   SPECIFICATIONS FOR LOCAL VARIABLES                       00034620
C   INTEGER NIND,NOUTD                                         00034630
C   DATA NIND/S/,NOUTD/6/                                    00034640
C   FIRST EXECUTABLE STATEMENT                               00034650
    IF (IOPT.EQ.3) GO TO 10                                     00034660
    IF (IOPT.EQ.2) GO TO 5                                     00034670
    IF (IOPT.EQ.1) GO TO 9005                                  00034680
    NIN = NIND                                                  00034690
    NOUT = NOUTD                                                00034700
    GO TO 9005                                                  00034710
    9005                                                         00034720
    9005                                                         00034730
    9005                                                         00034740
    9005                                                         00034750

```

```

5 NIND = NIN
GO TO 9005
10 NOUTD = NOUT
9005 RETURN
END

```

C
C
C

SUBROUTINE VXADD(A,ACC)

C
C
C

DOUBLE PRECISION A,ACC(2)

SPECIFICATIONS FOR ARGUMENTS

C
C
C

DOUBLE PRECISION X,Y,Z,ZZ

SPECIFICATIONS FOR LOCAL VARIABLES

FIRST EXECUTABLE STATEMENT

X = ACC(1)

Y = A

IF (DABS(ACC(1)).GE.DABS(A)) GO TO 1

X = A

Y = ACC(1)

C
C
C

1 Z = X+Y

ZZ = (X-Z)+Y

COMPUTE Z+ZZ = ACC(1)+A EXACTLY

COMPUTE ZZ+ACC(2) USING DOUBLE
PRECISION ARITHMETIC

C
C
C

ZZ = ZZ+ACC(2)

COMPUTE ACC(1)+ACC(2) = Z+ZZ EXACTLY

C
C
C

ACC(1) = Z+ZZ

ACC(2) = (Z-ACC(1))+ZZ

RETURN

END

C
C
C

SUBROUTINE VXMUL (A,B,ACC)

C
C
C

DOUBLE PRECISION A,B,ACC(2)

SPECIFICATIONS FOR ARGUMENTS

SPECIFICATIONS FOR LOCAL VARIABLES

DOUBLE PRECISION X,HA,TA,HB,TB

INTEGER IX(2),I

LOGICAL L1

EQUIVALENCE

DATA

LX(8),LI(4)

(X,LX(1),IX(1)),(I,LI(1))

1/0

SPLIT A = HA+TA

B = HB+TB

FIRST EXECUTABLE STATEMENT

C
C
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

00034760
00034770
00034780
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

မာယာပုဂံမာယာပုဂံမာယာ

C		RADIX THE BASE OF THE FLOATING-POINT	00035960
C		NUMBER SYSTEM USED	00035970
	DATA	RINFP/27FFFFFFF/	00035980
	DATA	REPSP/200100000/	00035990
	DATA	RADIX/16.0/	00036000
	DATA	RE: 1/23410000000000000/	00036010
	DATA	ZERO/0.000/ONE/1.000/	00036020
C		ZRPOLY USES SINGLE PRECISION	00036030
C		CALCULATIONS FOR SCALING, BOUNDS	00036040
C		AND ERROR CALCULATIONS.	00036050
C		FIRST EXECUTABLE STATEMENT	00036060
	IER = 0		00036070
	IF (NDEG .GT. 100 .OR. NDEG .LT. 1) GO TO 165		00036080
	ETA = REPSR1		00036090
	AME = ETA		00036100
	RMRE = ETA		00036110
	RLO = REPSP/ETA		00036120
C		INITIALIZATION OF CONSTANTS FOR	00036130
C		SHIFT ROTATION	00036140
	XX = .7071068		00036150
	YY = -XX		00036160
	SINR = .9975641		00036170
	COSR = -.06975647		00036180
	N = NDEG		00036190
	NN = N+1		00036200
C		ALGORITHM FAILS IF THE LEADING	00036210
C		COEFFICIENT IS ZERO.	00036220
	IF (A(1).NE.ZERO) GO TO 5		00036230
	IER = 130		00036240
	GO TO 9000		00036250
C		REMOVE THE ZEROS AT THE ORIGIN IF	00036260
C		ANY	00036270
	5 IF (A(NN).NE.ZERO) GO TO 10		00036280
	J = NDEG-N+1		00036290
	JJ = J+NDEG		00036300
	Z(J) = ZERO		00036310
	Z(JJ) = ZERO		00036320
	NN = NN-1		00036330
	N = N-1		00036340
	IF (NN.EQ.1) GO TO 9005		00036350
	GO TO 5		00036360
C		MAKE A COPY OF THE COEFFICIENTS	00036370
	10 DO 15 I=1,NN		00036380
	P(I) = A(I)		00036390
	15 CONTINUE		00036400
C		START THE ALGORITHM FOR ONE ZERO	00036410
	20 IF (N.GT.2) GO TO 30		00036420
	IF (N.LT.1) GO TO 9005		00036430
C		CALCULATE THE FINAL ZERO OR PAIR OF	00036440
C		ZEROS	00036450
	IF (N.EQ.2) GO TO 25		00036460
	Z(NDEG) = -P(2)/P(1)		00036470
	Z(NDEG+NDEG) = ZERO		00036480
	GO TO 145		00036490
	25 CALL ZRPOLI (P(1),P(2),P(3),Z(NDEG-1),Z(NDEG+NDEG-1),Z(NDEG),		00036500
	1 Z(NDEG+NDEG))		00036510
	GO TO 145		00036520
		FIND LARGEST AND SMALLEST MODULI OF	00036530
		COEFFICIENTS.	00036540
	30 RMAX = 0.		00036550

[illegible]

22

C

CC

C

cc

ကဏ္ဍ	အမျိုးအမည်	အရက်	အကျယ်အဝန်း	အရေအတွက်
၁	အချောအမွှာ	၂၀	၃၀	၆၀၀
၂	အချောအမွှာ	၂၀	၃၀	၆၀၀
၃	အချောအမွှာ	၂၀	၃၀	၆၀၀
၄	အချောအမွှာ	၂၀	၃၀	၆၀၀
၅	အချောအမွှာ	၂၀	၃၀	၆၀၀
၆	အချောအမွှာ	၂၀	၃၀	၆၀၀
၇	အချောအမွှာ	၂၀	၃၀	၆၀၀
၈	အချောအမွှာ	၂၀	၃၀	၆၀၀
၉	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၀	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၁	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၂	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၃	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၄	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၅	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၆	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၇	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၈	အချောအမွှာ	၂၀	၃၀	၆၀၀
၁၉	အချောအမွှာ	၂၀	၃၀	၆၀၀
၂၀	အချောအမွှာ	၂၀	၃၀	၆၀၀

CC

**ZERO OR ZEROS AND RETURN TO THE
MAIN ALGORITHM.**

00037760
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
Y 00038130
00038140
00038150
00038160
00038170
00038180
00038190
00038200
00038210
00038220
00038230
00038240
00038250
00038260
00038270
00038280
00038290
00038300
00038310
00038320
00038330
00038340
00038350

1

22

CC

C

C

CC

C

ic

1

```

C      DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00038360
C      DOUBLE PRECISION SR,S1,U,V,RA,RD,C,D,A1,A2,A3, 00038370
1      A6,A7,E,F,O,H,SZR,SZI,RLZR,RLZI, 00038380
2      SVU,SVV,UI,VI,S,ZERO 00038390
C      LOGICAL VPASS,SPASS,VTRY,STRY 00038400
C      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,S1,U,V,RA,RD,C,D,A1,A2,A3,A6, 00038410
1      A7,E,F,O,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NH 00038420
C      DATA ZERO/0.0D0/ 00038430
C      NZ = 0 00038440
C      FIRST EXECUTABLE STATEMENT 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
C      L2 - LIMIT OF FIXED SHIFT STEPS 00038530
C      NZ -NUMBER OF ZEROS FOUND 00038540
C      BETAV = .25 00038550
C      BETAS = .25 00038560
C      OSS = SR 00038570
C      OVV = V 00038580
C      EVALUATE POLYNOMIAL BY SYNTHETIC 00038590
C      DIVISION 00038600
C      CALL ZRPQLH (NH,U,V,P,QP,RA,RD) 00038610
C      CALL ZRPQLE (ITYPE) 00038620
C      DO 40 J=1,L2 00038630
C      CALCULATE NEXT K POLYNOMIAL AND 00038640
C      ESTIMATE V 00038650
C      CALL ZRPQLF (ITYPE) 00038660
C      CALL ZRPQLE (ITYPE) 00038670
C      CALL ZRPQLO (ITYPE,UI,VI) 00038680
C      VV = VI 00038690
C      ESTIMATE S 00038700
C      SS = 0. 00038710
C      IF (RK(N).NE.ZERO) SS = -P(NN)/RK(N) 00038720
C      TV = 1. 00038730
C      TS = 1. 00038740
C      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
C      IF (VV.NE.0.) TV = ABS((VV-OVV)/VV) 00038780
C      IF (SS.NE.0.) TS = ABS((SS-USS)/SS) 00038790
C      IF DECREASING, MULTIPLY TWO MOST 00038800
C      RECENT CONVERGENCE MEASURES 00038810
C      TVV = 1. 00038820
C      IF (TV.LT.OTV) TVV = TV*OTV 00038830
C      TSS = 1. 00038840
C      IF (TS.LT.OTS) TSS = TS*OTS 00038850
C      COMPARE WITH CONVERGENCE CRITERIA 00038860
C      VPASS = TVV.LT.BETAV 00038870
C      SPASS = TSS.LT.BETAS 00038880
C      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
C      SVU = U 00038930
C      SVV = V 00038940
C      DO 5 I=1,N 00038950

```

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

```

C      INTEGER          NZ          00039560
      DOUBLE PRECISION  UU,VV      00039570
C      SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER          N,NN,J,I,ITYPE 00039580
      REAL             ARE,EE,ETA,OMP,RELSTP,RMP,RMRE,T,ZM 00039590
      DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00039600
      DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00039610
      1                A6,A7,E,F,O,H,SZR,SZI,RLZR,RLZI, 00039620
      2                UI,VI,ZERO,PT01,ONE 00039630
      LOGICAL          TRIED 00039640
      COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00039650
      1                A7,E,F,O,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00039660
      DATA            ZERO,PT01,ONE/0.000,0.0100,1.000/ 00039670
C      FIRST EXECUTABLE STATEMENT 00039680
      NZ = 0 00039690
C      VARIABLE-SHIFT K-POLYNOMIAL 00039700
C      ITERATION FOR A QUADRATIC FACTOR 00039710
C      CONVERGES ONLY IF THE ZEROS ARE 00039720
C      EQUIMODULAR OR NEARLY SO 00039730
C      UU,VV - COEFFICIENTS OF STARTING 00039740
C      QUADRATIC 00039750
C      NZ - NUMBER OF ZERO FOUND 00039760
      TRIED = .FALSE. 00039770
      U = UU 00039780
      V = VV 00039790
      J = 0 00039800
C      MAIN LOOP 00039810
      3 CALL ZRPQLI (ONE,U,V,SZR,SZI,RLZR,RLZI) 00039820
C      RETURN IF ROOTS OF THE QUADRATIC ARE 00039830
C      REAL AND NOT CLOSE TO MULTIPLE OR 00039840
C      NEARLY EQUAL AND OF OPPOSITE SIGN 00039850
      IF ( DABS(DABS(SZR)-DABS(RLZR)).GT.PT01*DABS(RLZR)) RETURN 00039860
C      EVALUATE POLYNOMIAL BY QUADRATIC 00039870
C      SYNTHETIC DIVISION 00039880
      CALL ZRPQLH (NN,U,V,P,QP,RA,RB) 00039890
      RMP = DABS(RA-SZR*RB)+DABS(SZI*RB) 00039900
C      COMPUTE A RIGOROUS BOUND ON THE 00039910
C      ROUNDING ERROR IN EVALUATING P 00039920
      ZM = SQRT(ABS(SNOL(V))) 00039930
      EE = 2.*ABS(SNOL(QP(1))) 00039940
      T = -SZR*RB 00039950
      DO 10 I=2,N 00039960
      10 EE = EE*ZM+ABS(SNOL(QP(I))) 00039970
      EE = EE*ZM+ABS(SNOL(RA)+T) 00039980
      EE = (5.*RMRE+4.*ARE)*EE-(5.*RMRE+2.*ARE)*M(ABS(SNOL(RA)+T)+ 00039990
      1 ABS(SNOL(RB))*ZM)+2.*ARE*ABS(T) 00040000
C      ITERATION HAS CONVERGED SUFFICIENTLY 00040010
C      IF THE POLYNOMIAL VALUE IS LESS 00040020
C      THAN 20 TIMES THIS BOUND 00040030
      IF (RMP.GT.20.*EE) GO TO 15 00040040
      NZ = NZ + 1 00040050
      RETURN 00040060
      15 J = J+1 00040070
C      STOP ITERATION AFTER 20 STEPS 00040080
      IF (J.GT.20) RETURN 00040090
      IF (J.LT.2) GO TO 25 00040100
      IF (RELSTP.GT..01.OR.RMP.LT.OMP.OR.TRIED) GO TO 25 00040110
C      A CLUSTER APPEARS TO BE STALLING THE 00040120
C      CONVERGENCE. FIVE FIXED SHIFT 00040130
C      STEPS ARE TAKEN WITH A U,V CLOSE 00040140
      00040150
  
```

```

C      IF (RELSTP.LT.EYA) RELSTP = ETA      TO THE CLUSTER      00040160
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.                        00040250
    J = 0                                  00040260
25 OMP = RMP                               00040270
                                           00040280
C                                           CALCULATE NEXT K POLYNOMIAL AND NEW
C                                           U AND V      00040290
                                           00040300
CALL ZRPQLE (ITYPE)                       00040310
CALL ZRPQLF (ITYPE)                       00040320
CALL ZRPQLE (ITYPE)                       00040330
CALL ZRPQLO (ITYPE,UI,VI)                 00040340
                                           00040350
C                                           IF VI IS ZERO THE ITERATION IS NOT
C                                           CONVERGING      00040360
                                           00040370
IF (VI.EQ.ZERO) RETURN                    00040380
RELSTP = DABS((VI-V)/VI)                  00040390
U = UI                                     00040400
V = VI                                     00040410
GO TO 5                                    00040420
END                                         00040430
                                           00040440
C                                           00040450
C                                           00040460
SUBROUTINE ZRPQLD (SSS,NZ,IFLAG)           00040470
C                                           SPECIFICATIONS FOR ARGUMENTS      00040480
INTEGER NZ,IFLAG                          00040490
DOUBLE PRECISION SSS                      00040500
C                                           SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER N,NN,J,X                          00040510
REAL ARE,EE,ETA,OMP,RMP,RMS,RMRE         00040520
DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101) 00040530
DOUBLE PRECISION SR,SI,U,V,RA,RB,C,D,A1,A2,A3, 00040540
A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,        00040550
1 PV,RKV,T,S,ZERO,PT001                  00040560
2 COMMON /ZRPQLJ/ P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6, 00040570
A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN 00040580
1 DATA ZERO/0.000/,PT001/0.00100/      00040590
                                           VARIABLE-SHIFT M POLYNOMIAL
                                           ITERATION FOR A REAL ZERO SSS -
                                           STARTING ITERATE      00040610
NZ - NUMBER OF ZERO FOUND                00040620
IFLAG - FLAG TO INDICATE A PAIR OF      00040630
ZEROS NEAR REAL AXIS                    00040640
FIRST EXECUTABLE STATEMENT              00040650
                                           00040660
NZ = 0                                   00040670
S = SSS                                  00040680
IFLAG = 0                                00040690
J = 0                                    00040700
C                                           MAIN LOOP      00040710
C 5 PV = P(1)                            00040720
                                           EVALUATE P AT S      00040730
QP(1) = PV                               00040740
DO 10 I=7,NN                             00040750
  
```

```

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

```

00040760
 00040770
 00040780
 00040790
 00040800
 00040810
 00040820
 00040830
 00040840
 00040850
 00040860
 00040870
 00040880
 00040890
 00040900
 00040910
 00040920
 00040930
 00040940
 00040950
 00040960
 00040970
 00040980
 00040990
 00041000
 00041010
 00041020
 00041030
 00041040
 00041050
 00041060
 00041070
 00041080
 00041090
 00041100
 00041110
 00041120
 00041130
 00041140
 00041150
 00041160
 00041170
 00041180
 00041190
 00041200
 00041210
 00041220
 00041230
 00041240
 00041250
 00041260
 00041270
 00041280
 00041290
 00041300
 00041310
 00041320
 00041330
 00041340
 00041350

```

GO TO 5
END

IMSL ROUTINE NAME - ZRPQLE

-----
COMPUTER - IBM/DOUBLE
LATEST REVISION - JANUARY 1, 1978
SUBROUTINE ZRPQLE (ITYPE)
  INTEGER ITYPE
  INTEGER N, NN
  REAL ARE, ETA, FMRE
  DOUBLE PRECISION P(101), QF(101), RK(101), QK(101), SVK(101)
  DOUBLE PRECISION SR, SI, U, V, RA, RB, C, D, A1, A2, A3,
  COMMON /ZRPQLJ/ A6, A7, E, F, G, H, SZR, SZI, RLZR, RLZI,
  P, QP, RK, QK, SVK, SP, SI, U, V, RA, RB, C, D, A1, A2, A3, A6,
  A7, E, F, G, H, SZR, SZI, RLZR, RLZI, ETA, ARE, RMRE, N, NN
  THIS ROUTINE CALCULATES SCALAR
  QUANTITIES USED TO COMPUTE THE
  NEXT K POLYNOMIAL AND NEW
  ESTIMATES OF THE QUADRATIC
  COEFFICIENTS
  ITYPE - INTEGER VARIABLE SET HERE
  INDICATING HOW THE CALCULATIONS
  ARE NORMALIZED TO AVOID OVERFLOW
  SYNTHETIC DIVISION OF K BY THE
  QUADRATIC 1, U, V
  FIRST EXECUTABLE STATEMENT

CALL ZRPQLH (N, U, V, RK, QK, C, D)
IF (DABS(C).GT.DABS(RK(N))*100.*ETA) GO TO 5
IF (DABS(D).GT.DABS(RK(N-1))*100.*ETA) GO TO 5
ITYPE = 3
TYPE=3 INDICATES THE QUADRATIC IS
ALMOST A FACTOR OF K

RETURN
5 IF (DABS(D).LT.DABS(C)) GO TO 10
ITYPE = 2
TYPE=2 INDICATES THAT ALL FORMULAS
ARE DIVIDED BY D

F = RA/D
F = C/D
G = U*RB
H = V*RB
A3 = (RA+G)*E+H*(RB/D)
A1 = RB*F-RA
A7 = (F+U)*RA+H
RETURN
10 ITYPE = 1
TYPE=1 INDICATES THAT ALL FORMULAS
ARE DIVIDED BY C

E = RA/C
F = D/C
G = U*E
  
```

```

H = VMRB                                00041960
A5 = RA*E+(H/C+G)*RB                    00041970
A1 = RB-RA*(D/C)                         00041980
A7 = RA+G*D+H*F                         00041990
RETURN                                   00042000
END                                       00042010
C                                         00042020
C                                         00042030
C                                         00042040
SUBROUTINE ZRPQLF (ITYPE)                00042050
C                                         00042060
C                                         00042070
C                                         00042080
C                                         00042090
C                                         00042100
C                                         00042110
C                                         00042120
C                                         00042130
C                                         00042140
C                                         00042150
C                                         00042160
C                                         00042170
C                                         00042180
C                                         00042190
C                                         00042200
C                                         00042210
C                                         00042220
C                                         00042230
C                                         00042240
C                                         00042250
C                                         00042260
C                                         00042270
C                                         00042280
C                                         00042290
C                                         00042300
C                                         00042310
C                                         00042320
C                                         00042330
C                                         00042340
C                                         00042350
C                                         00042360
C                                         00042370
C                                         00042380
C                                         00042390
C                                         00042400
C                                         00042410
C                                         00042420
C                                         00042430
C                                         00042440
C                                         00042450
C                                         00042460
C                                         00042470
C                                         00042480
C                                         00042490
C                                         00042500
C                                         00042510
C                                         00042520
C                                         00042530
C                                         00042540
C                                         00042550

H = VMRB
A5 = RA*E+(H/C+G)*RB
A1 = RB-RA*(D/C)
A7 = RA+G*D+H*F
RETURN
END

SUBROUTINE ZRPQLF (ITYPE)
C
C   SPECIFICATIONS FOR ARGUMENTS
C   SPECIFICATIONS FOR LOCAL VARIABLES
C
INTEGER          ITYPE
REAL             N,NN,I
DOUBLE PRECISION ARE,ETA,RMRE
DOUBLE PRECISION P(101),QP(101),RK(101),QK(101),SVK(101)
1  COMMON /ZRPQLJ/ A6,A7,E,F,G,H,SZR,SZI,RLZR,RLZI,TEMP,ZERO
1  DATA           P,QP,RK,QK,SVK,SR,SI,U,V,RA,RB,C,D,A1,A2,A3,A6,
A7,E,F,G,H,SZR,SZI,RLZR,RLZI,ETA,ARE,RMRE,N,NN
ZERO/0.0D0/

C   COMPUTES THE NEXT K POLYNOMIALS
C   USING SCALARS COMPUTED IN ZRPQLE
C   FIRST EXECUTABLE STATEMENT

IF (ITYPE.EQ.3) GO TO 20
TEMP = RA
IF (ITYPE.EQ.1) TEMP = RB
IF (DABS(A1).GT.DABS(TEMP)*ETA*10.) GO TO 10
IF A1 IS NEARLY ZERO THEN USE A
SPECIAL FORM OF THE RECURRENCE

RK(1) = ZERO
RK(2) = -A7*QP(1)
DO 5 I=3,N
5  RK(I) = A3*QK(I-2)-A7*QP(I-1)
RETURN

C   USE SCALED FORM OF THE RECURRENCE

10 A7 = A7/A1
A3 = A3/A1
RK(1) = QP(1)
RK(2) = QP(2)-A7*QP(1)
DO 15 I=3,N
15 RK(I) = A3*QK(I-2)-A7*QP(I-1)+QP(I)
RETURN

C   USE UNSCALED FORM OF THE RECURRENCE
C   IF TYPE IS 3

20 RK(1) = ZERO
RK(2) = ZERO
DO 25 I=3,N
25 RK(I) = QK(I-2)
RETURN
END

IMSL ROUTINE NAME - ZRPQLG

COMPUTER - IBM/DOUBLE
  
```

```

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

```

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

```

E = RB*(RB/DABS(C))-E
D = DSQRT(DABS(E))*DSQRT(DABS(C))
25 IF (E.LT.ZERO) GO TO 30
C REAL ZEROS
IF (RB.GE.ZERO) D = -D
RLR = (-RB+D)/RA
SR = ZERO
IF (RLR.NE.ZERO) SR = (C/RLR)/RA
GO TO 5
C COMPLEX CONJUGATE ZEROS
30 SR = -RB/RA
RLR = SR
SI = DABS(D/RA)
RLI = -SI
RETURN
END
C
C
C
C SUBROUTINE LEQ2C (A,N,IA,B,M,IB,IJOB,WA,WK,IER)
C COMPLEX*16 A(IA,1),B(IB,1),WA(N,1),TEMPA,TEMPC,TEMPC
C DOUBLE PRECISION WK(N),TA(2),TB(2),TC(2)
C DOUBLE PRECISION AR,AI,BR,BI,CR,CI,DXNORM,XNORM,ZERO
C DOUBLE PRECISION ACC(2)
C EQUIVALENCE (TA(1),TEMPA),(TB(1),TEMPB),(TC(1),TEMPC),
C (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI),
C (TC(1),CR),(TC(2),CI)
C DATA ZERO/0.0D0/
C DATA ITMAX/50/
C FIRST EXECUTABLE STATEMENT
IER = 0
N1 = N+1
N2 = N+2
IF (IJOB .EQ. 2) GO TO 15
C SAVE MATRIX A
DO 10 I = 1,N
DO 5 J = 1,N
WA(I,J) = A(I,J)
5 CONTINUE
10 CONTINUE
C FACTOR MATRIX A
CALL LEQ1C(WA,N,N,B,M,IB,1,WK,IER)
IF (IER .NE. 0) GO TO 9000
IF (IJOB .EQ. 1) GO TO 9005
C SAVE THE RIGHT HAND SIDES
15 DO 65 J = 1,M
DO 20 I = 1,N
WA(I,N1) = B(I,J)
20 CONTINUE
C OBTAIN A SOLUTION
CALL LEQ1C(WA,N,N,WA(1,N1),1,N,2,WK,IER)
C COMPUTE THE NORM OF THE SOLUTION
XNORM = ZERO
DO 25 I = 1,N
TEMPA = WA(I,N1)
XNORM = DMAX1(XNORM,DABS(AR),DABS(AI))
25 CONTINUE
IF (XNORM .EQ. ZERO) GO TO 65
C COMPUTE RESIDUALS

```

00043760
 00043770
 00043780
 00043790
 00043800
 00043810
 00043820
 00043830
 00043840
 00043850
 00043860
 00043870
 00043880
 00043890
 00043900
 00043910
 00043920
 00043930
 00043940
 00043950
 00043960
 00043970
 00043980
 00043990
 00044000
 00044010
 00044020
 00044030
 00044040
 00044050
 00044060
 00044070
 00044080
 00044090
 00044100
 00044110
 00044120
 00044130
 00044140
 00044150
 00044160
 00044170
 00044180
 00044190
 00044200
 00044210
 00044220
 00044230
 00044240
 00044250
 00044260
 00044270
 00044280
 00044290
 00044300
 00044310
 00044320
 00044330
 00044340
 00044350

DO 50 ITER = 1,ITMAX	00044360
DO 40 I = 1,N	00044370
TEMPB = B(I,J)	00044380
ACC(1) = 0.000	00044390
ACC(2) = 0.000	00044400
CALL VXADD(BR,ACC)	00044410
DO 30 JJ = 1,N	00044420
TEMPA = A(I,JJ)	00044430
TEMPB = WA(JJ,N1)	00044440
CALL VXMUL(-AR,BR,ACC)	00044450
CALL VXMUL(AI,BI,ACC)	00044460
30 CONTINUE	00044470
CALL VXSTO(ACC,CR)	00044480
TEMPB = B(I,J)	00044490
ACC(1) = 0.000	00044500
ACC(2) = 0.000	00044510
CALL VXADD(BI,ACC)	00044520
DO 35 JJ = 1,N	00044530
TEMPA = A(I,JJ)	00044540
TEMPB = WA(JJ,N1)	00044550
CALL VXMUL(-AR,BI,ACC)	00044560
CALL VXMUL(-BR,AI,ACC)	00044570
35 CONTINUE	00044580
CALL VXSTO(ACC,CI)	00044590
WA(I,N2) = TEMPC	00044600
40 CONTINUE	00044610
CALL LEQ1C(WA,N,N,WA(1,N2),1,N,2,WK,IER)	00044620
DXNORM = ZERO	00044630
C UPDATE THE SOLUTION	00044640
DO 45 I = 1,N	00044650
WA(I,N1) = WA(I,N1)+WA(I,N2)	00044660
TEMPA = WA(I,N2)	00044670
DXNORM = DMAX1(DXNORM,DABS(AR),DABS(AI))	00044680
45 CONTINUE	00044690
IF (XNORM+DXNORM .EQ. XNORM) GO TO 55	00044700
50 CONTINUE	00044710
IER = 130	00044720
C STORE THE SOLUTION	00044730
55 DO 60 JK = 1,N	00044740
B(JK,J) = WA(JK,N1)	00044750
60 CONTINUE	00044760
IF (IER .NE. 0) GO TO 9000	00044770
65 CONTINUE	00044780
GO TO 9005	00044790
9000 CONTINUE	00044800
CALL UERTST(IER,6HLEQ2C)	00044810
9005 RETURN	00044820
END	00044830
C	00044840
C	00044850
C	00044860
SUBROUTINE LEQ1C (A,N,IA,B,M,IB,IJOB,WA,IER)	00044870
C SPECIFICATIONS FOR ARGUMENTS	00044880
INTEGER N,IA,M,IB,IJOB,IER	00044890
COMPLEX*16 A(IA,N),B(IB,M)	00044900
DOUBLE PRECISION WA(N)	00044910
C SPECIFICATIONS FOR LOCAL VARIABLES	00044920
DOUBLE PRECISION P,Q,ZERO,ONE,T(2),RN,BIG	00044930
COMPLEX*16 SUM,TEMP	00044940
INTEGER I,J,JM1,IM1,K,IMAX,JF1,IW,N1	00044950

	EQUIVALENCE	(SUM,T(1))	00044960
	DATA	ZERO/0.000/,ONE/1.00/	00044970
C		INITIALIZATION	00044980
C		FIRST EXECUTABLE STATEMENT	00044990
	IER = 0		00045000
	IF (IJOB .EQ. 2) GO TO 75		00045010
	RN = N		00045020
C		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
	JM1 = J-1		00045160
	IF (JM1 .LT. 1) GO TO 25		00045170
C		COMPUTE U(I,J), I=1,...,J-1	00045180
	DO 20 I=1,JM1		00045190
	SUM = A(I,J)		00045200
	IM1 = I-1		00045210
	IF (IM1 .LT. 1) GO TO 20		00045220
	DO 15 K=1,IM1		00045230
	SUM = SUM-A(I,K)*A(K,J)		00045240
15	CONTINUE		00045250
	A(I,J) = SUM		00045260
20	CONTINUE		00045270
25	P = ZERO		00045280
C		COMPUTE U(J,J) AND L(I,J), I=J+1,...,	00045290
	DO 45 I=J,N		00045300
	SUM = A(I,J)		00045310
	IF (JM1 .LT. 1) GO TO 40		00045320
	DO 35 K=1,JM1		00045330
	SUM = SUM-A(I,K)*A(K,J)		00045340
35	CONTINUE		00045350
	A(I,J) = SUM		00045360
40	Q = WA(I)*CDABS(SUM)		00045370
	IF (P .GE. Q) GO TO 45		00045380
	P = Q		00045390
	IMAX = I		00045400
45	CONTINUE		00045410
C		TEST FOR ALGORITHMIC SINGULARITY	00045420
	Q = RN+P		00045430
	IF (Q .EQ. RN) GO TO 105		00045440
	IF (J .EQ. IMAX) GO TO 60		00045450
C		INTERCHANGE ROWS J AND IMAX	00045460
	DO 50 K=1,N		00045470
	TEMP = A(IMAX,K)		00045480
	A(IMAX,K) = A(J,K)		00045490
	A(J,K) = TEMP		00045500
50	CONTINUE		00045510
	WA(IMAX) = WA(J)		00045520
60	WA(J) = IMAX		00045530
	JP1 = J+1		00045540
	IF (JP1 .GT. N) GO TO 70		00045550

C	TEMP = A(J,J)	DIVIDE BY PIVOT ELEMENT U(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
105	IER = 129		00045930
9000	CONTINUE		00045940
C		PRINT ERROR	00045950
	CALL UERTST(IER,6HLE9T1C)		00045960
9005	RETURN		00045970
	END		00045980
			00045990

SUPPLEMENTARY

INFORMATION



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

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

1 May 1987

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

ALL ADDRESSES

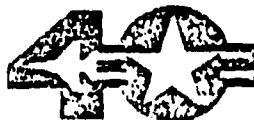
1. Please delete the second paragraph in the NOTICE page affixed to the inside cover of AFWAL-TR-86-3034, "Strength Analysis of Laminated and Metallic Plates Bolted Together by Many Fasteners" and AFWAL-TR-86-3035, "Design Guide for Bolted Joints in Composite Structures."
2. Please contact the undersigned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

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

cc: AFWAL/FIBRA
(V. Venkayya)

AD-B108123

UNITED STATES AIR FORCE



SEPTEMBER 18, 1947