



OUT-OF-PLANE ANALYSIS FOR COMPOSITE STRUCTURES VOLUME II SOFTWARE USERS MANUAL

Philip C. Paul
McDonnell Douglas Aerospace — East
MC DONNELL DOUGLAS CORPORATION
P.O. Box 516
St. Louis, MO 63166

Mary A. Mahler
NORTHROP CORPORATION
Aircraft Division
One Northrup Avenue
Hawthorne, CA 90250

15 SEPTEMBER 1994

FINAL REPORT

Distribution is Limited To U.S. Government Agencies and Their Contractors; Critical Technology: 15 September 1994. All Other Requests for this Document Shall be Referred to: Commanding Officer, NAWCADWAR (Code 0471).

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. Violations of these export laws are subject to severe criminal penalties.

DESTRUCTION NOTICE — For Unclassified, Limited Documents, Destroy by Any Method That Will Prevent Disclosure of Contents or Reconstruction of the Document.

Prepared for
Air Vehicle and Crew Systems Technology Department (Code 6043)
NAVAL AIR WARFARE CENTER, AIRCRAFT DIVISION
P.O. Box 5152
Warminster, PA 18974-0591



NOTICES

REPORT NUMBERING SYSTEM — The numbering of technical project reports issued by the Naval Air Warfare Center, Aircraft Division, Warminster is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Functional Department responsible for the report. For example: Report No. NAWCADWAR-92001-60 indicates the first Center report for the year 1992 and prepared by the Air Vehicle and Crew Systems Technology Department. The numerical codes are as follows:

CODE	OFFICE OR DEPARTMENT
00	Commanding Officer, NAWCADWAR
01	Technical Director, NAWCADWAR
05	Computer Department
10	AntiSubmarine Warfare Systems Department
20	Tactical Air Systems Department
30	Warfare Systems Analysis Department
50	Mission Avionics Technology Department
60	Air Vehicle & Crew Systems Technology Department
70	Systems & Software Technology Department
80	Engineering Support Group
90	Test & Evaluation Group

PRODUCT ENDORSEMENT — The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

Reviewed By: Edward F. Kant
Contracting Officer's Technical Representative (COTR)
or
Point Of Contact (POC)

Date: 10/28/94

Reviewed By: A. Mann
Branch Head

Date: 10/31/94

Reviewed By: J. E. Heer
Division Head

Date: 10/31/94

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 15 SEPT 1994	3. REPORT TYPE AND DATES COVERED FINAL		
4. TITLE AND SUBTITLE OUT-OF-PLANE ANALYSIS FOR COMPOSITE STRUCTURES VOLUME II SOFTWARE USER'S MANUAL		5. FUNDING NUMBERS N62269-87-C-0226		
6. AUTHOR(S) *PHILIP C. PAUL **MARY A. MAHLER				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) *McDonnell Douglas Aerospace East McDonnell Douglas Corporation P.O. Box 516 St. Louis, MO 63166 **Northrup Corporation Aircraft Division One Northrup Avenue Hawthorne, CA 90250		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Vehicle and Crew Systems Technology Department (Code 6043) NAVAL AIR WARFARE CENTER, AIRCRAFT DIVISION P.O. Box 5152 Warminster, PA 18974-0591		10. SPONSORING / MONITORING AGENCY REPORT NUMBER NAWCADWAR-94138-60 Vol. II		
11. SUPPLEMENTARY NOTES DOT/FAA Report No. DOT/FAA/CT-91/22-II NAWCADWAR P.O.C. — Edward Kautz — Code 6043				
12a. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS; CRITICAL TECHNOLOGY; 15 SEPTEMBER, 1994. OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO: COMMANDING OFFICER, NAWCADWAR (CODE 0471).		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) Composite airframe structures have recently experienced several unexpected failures due to the effects of out-of-plane loads. These loads are inherent to laminated, cocured, and adhesively bonded composite structures. There is great difficulty accounting for these loads and predicting their failures since the strength in the out-of-plane direction is commonly weak and inadequate design tools have been available. This report presents several design tools that will assist designers/analysts to produce strong, durable, integral composite designs. This report is divided into two volumes. Volume I contains a review of out-of-plane structural failures, the development and verification of the analysis methods from the program, the demonstration of the methodology to a full scale test article, and a design guide. Volume II consists of the software user's manual and the computer routines for the analysis methods developed in the program.				
14. SUBJECT TERMS COMPOSITE MATERIALS, COMPOSITE STRUCTURES, GRAPHITE EPOXY, OUT-OF-PLANE, INDIRECT LOADS, TRANSLAMINAR STRESS		15. NUMBER OF PAGES 299		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	ii
LIST OF FIGURES	v
LIST OF PAGES	vii
FOREWORD	viii
SECTION I INTRODUCTION	1-1
SECTION II DESIGN GUIDE FOR OUT-OF-PLANE ANALYSIS OF COMPOSITE STRUCTURES	2-1
1. Assessment of Loading Conditions	2-1
2. Structural Analysis Methods	2-3
2.1 Curved Laminate Analysis (CLA)	2-3
2.2 Ply Drop Off Procedures	2-8
2.3 Stiffener Pull-Off Analyses	2-9
2.3.1 Direct Loadings – Fuel Pressure	2-9
2.3.2 Indirect Loadings – Postbuckled Panels	2-13
2.4 Geometric Discontinuities	2-15
2.4.1 Stiffener Runouts	2-17
2.4.2 Example Analysis	2-19
3. Element Test Requirements	2-23
4. Allowables Determination	2-28
5. Preferred Design Techniques	2-31
SECTION III CURVED LAMINATE ANALYSIS (CLA) USER'S MANUAL	3-1
1. Introduction	3-1
1.1 Function	3-1
1.2 Limitations	3-1
2. Input Description	3-2
3. Output Description	3-4
4. Verification	3-4
5. Subroutine Functions	3-6
6. Example Problems	3-8
6.1 Problem #1	3-8
6.2 Problem #2	3-15

	Page
SECTION IV INTERFACIAL STRESS FIELD MODEL (ISFM) USER'S MANUAL ...	4-1
1. Introduction	4-1
1.1 Function	4-1
1.2 Limitations	4-2
2. Input Description	4-2
3. Output Description	4-3
4. Verification	4-4
5. Subroutine Functions	4-6
6. Example Problems	4-7
6.1 Problem #1	4-7
6.2 Problem #2	4-16
SECTION V ELASTICITY METHOD (WEBSTER) USER'S MANUAL	5-1
1. Introduction	5-1
1.1 Function	5-1
1.2 Limitations	5-1
2. Input Description	5-2
2.1 Errors and How to Avoid Them	5-8
3. Output Description	5-8
3.1 Output Warnings	5-8
4. Example Problem	5-9
SECTION VI RAYLEIGH-RITZ METHOD (BEAM) USER'S MANUAL	6-1
1. Introduction	6-1
1.1 Function	6-1
1.2 Limitations	6-1
2. Input Description	6-2
3. Output Description	6-4
4. Example Problem	6-5
SECTION VII SIMPLIFIED STIFFENER ANALYSIS MODEL (SSAM) USER'S MANUAL	7-1
1. Introduction	7-1
1.1 Function	7-1
1.2 Limitations	7-1

	Page
2. Description of SSAM Program	7-1
2.1 SSAM Program Input	7-1
2.2 SSAM Program Output	7-4
3. Description of SSAMGEN Program	7-4
3.1 SSAMGEN Program Inputs	7-4
3.2 Description of SSAMGEN Input Variables	7-7
3.3 SSAMGEN Program Output	7-9
4. Verification	7-9
5. Example Problem	7-12
SECTION VIII FORTRAN PROGRAM LISTINGS	8-1
1. Curved Laminate Analysis (CLA)	8-1
2. Interfacial Stress Field Model (ISFM)	8-20
3. Elasticity Method (WEBSTER)	8-35
4. Rayleigh-Ritz Method (BEAM)	8-86
5. Simplified Stiffener Analysis Model (SSAM & SSAMGEN)	8-107

LIST OF FIGURES

	Title	Page
Figure 1.	Potential Out-of-Plane Failures in Integrally Stiffened Torque Box Design and Recommended Analysis Routines	1-1
Figure 2.	Typical Out-Of-Plane Loads in Composite Structures	2-2
Figure 3.	CLA Uses Loads Defined at the Center of the Radial Arc	2-4
Figure 4.	Maximum Interlaminar Stresses in Corner Due to Axial Loadings	2-5
Figure 5.	Maximum Interlaminar Tension Stresses Due to Corner Moment	2-5
Figure 6.	Effect of Resin Rich Area on Maximum Interlaminar Tension Stress Due to Applied Moment	2-6
Figure 7.	Flange Gap Model	2-7
Figure 8.	Maximum Interlaminar Shear Stress as a Function of Gap Size	2-7
Figure 9.	MCAIR/Northrop Ply Drop Off Design Procedures	2-8
Figure 10.	Edge Loading Due to Lateral Pressure on a Panel	2-10
Figure 11.	Effect of Panel Geometry on Maximum Edge Load	2-10
Figure 12.	Spar Cap Design	2-11
Figure 13.	Comparison of ISFM and SSAM Results for Stresses at the End of a Spar Cap Flange	2-11
Figure 14.	Effect of Increased Flange Stiffness Under Pull-Off Load (SSAM Analysis)	2-12
Figure 15.	An Inner Wrapped Ply Does Not Improve Pull-Off Stresses Greatly	2-13
Figure 16.	Model Geometry for ISFM Analysis of Skin/Stiffener Separation	2-14
Figure 17.	Flange Geometry Affects Interface Stress Distributions	2-15
Figure 18.	Comparison of ISFM and SSAM Results for Hat Stiffener Loaded by Skin Displacement	2-16
Figure 19.	Effect of Flange EI for Prescribed Displacement Case	2-16
Figure 20.	Effect of Flange EI for Prescribed Displacement Case	2-17
Figure 21.	Stiffener Runout Analysis Technique	2-18
Figure 22.	Stiffness Distribution for Stiffener Runout	2-19
Figure 23.	Stiffener Flange Analysis	2-20
Figure 24.	Pull-Off Load Peaks at End of Cap for Open End Hat Stiffener	2-21
Figure 25.	Stiffness Distribution for Closed End Stiffener	2-22
Figure 26.	Pull-Off Loads Are Much Lower for Closed End Stiffener	2-22
Figure 27.	Interlaminar Tension Test Specimen	2-23
Figure 28.	Interlaminar Shear Test Specimen	2-24

	Title	Page
Figure 29.	Schematic of Curved Laminate Test Configuration	2-25
Figure 30.	MCAIR Cocured Joint Element Specimens	2-26
Figure 31.	Direct Out-of-Plane Loading of Hat Stiffened Element Specimen	2-26
Figure 32.	Recommended Skin/Stiffener Pull Off Specimen	2-27
Figure 33.	Schematic of Stiffener Runout Test Specimen	2-28
Figure 34.	The Additional Scatter in Out-of-Plane Strengths Is Reflected in a Reduced Factor on Allowables	2-29
Figure 35.	Translaminar Fatigue Test Results	2-30
Figure 36.	Effect of Fatigue on Pull-Off Strength of Hat Stiffener Panels	2-30
Figure 37.	Curved Laminate Analysis Geometry	3-3
Figure 38.	MCAIR/Northrop Angle Specimen Test Configuration	3-5
Figure 39.	Curved Laminate Analysis Correlation	3-5
Figure 40.	Effect of Layup and Inside Radius on the Maximum Radial Stress (as Predicted by CLA)	3-6
Figure 41.	CLA Flowchart	3-6
Figure 42.	Angle Specimen and the Resolution of the Applied Loads to the Center of Curvature	3-9
Figure 43.	L-Bracket Subjected to Combined Pressure and Tension Loads	3-16
Figure 44.	General Interfacial Stress Field Model	4-1
Figure 45.	Correlation of Predicted and Measured Pull Off Strengths	4-4
Figure 46.	Correlation of ISFM and Test Data of Panels Subjected to Combined Loads	4-5
Figure 47.	ISFM Stress Variations Resulting From Different Values for the Elastic Foundation Modulus (k)	4-5
Figure 48.	ISFM Flowchart	4-6
Figure 49.	ISFM Analysis of Blade Stiffener Pull Off Specimen	4-7
Figure 50.	ISFM Analysis of an Integral Blade Stiffened Panel	4-16
Figure 51.	Model of the Elasticity Method/WEBSTER	5-2
Figure 52.	Hat Stiffener Pull Off Configuration for WEBSTER	5-3
Figure 53.	Model of Blade Stiffener Pull Off Specimen	5-5
Figure 54.	Model of Hat Stiffener Pull Off Specimen	5-5
Figure 55.	Hat and Blade Stiffener Load Reduction Technique	5-7
Figure 56.	Model of the Rayleigh-Ritz Method/BEAM	6-1
Figure 57.	Geometry Description of BEAM Model	6-3
Figure 58.	The Simplified Stiffener Analysis Model (SSAM)	7-5

	Title	Page
Figure 59.	SSAMGEN Geometric and Loading Input Parameters	7-6
Figure 60.	Hat Stiffener Without an Internal Flange	7-10
Figure 61.	Hat Stiffener With an Internal Flange	7-11
Figure 62.	Comparison of SSAM Predictions With Test Data	7-12

LIST OF PAGES

Title Page
ii thru viii
1-1 thru 1-2
2-1 thru 2-32
3-1 thru 3-23
4-1 thru 4-22
5-1 thru 5-14
6-1 thru 6-8
7-1 thru 7-34
8-1 thru 8-156

FOREWORD

This report was prepared by McDonnell Aircraft Company (MCAIR), St. Louis, Missouri and Northrop Aircraft Division, Hawthorne, California under Contract N62269-87-C-0226. This report was prepared for the Naval Air Warfare Center (NAWC), Warminster, Pennsylvania and for the Federal Aviation Administration Technical Center, Atlantic City, New Jersey. The NAWC Project Engineer was Edward Kautz. The technical monitor for the FAA was Larry Neri with additional guidance provided by J. Soderquist, the FAA's National Resource Specialist for Composites.

While the Structural Research Department of MCAIR had primary responsibility for the performance of this program, the Structures Research Department of Northrop Aircraft Division had an equal share in the analytical development. The Program Manager for MCAIR was C.R. Saff. The Principal Investigator for MCAIR was Kenneth Sanger, and the Principal Investigator for Northrop was Dr. Han Pin Kan. The analysis development was performed by Philip C. Paul (MCAIR) and Mary Mahler (Northrop). Principal authors of this report were Philip C. Paul, C.R. Saff, and Kenneth Sanger of MCAIR and Mary Mahler and Dr. Han Pin Kan of Northrop.

SECTION I INTRODUCTION

Several analyses were developed and verified in Sections IV and VI of Volume I of this Final Report. This volume presents a design guide for the analysis of out-of-plane loading conditions and User's Manuals for the FORTRAN computer routines of these analyses. The routines and User's Manuals were created to facilitate the use of the analyses. The first paragraph of each Section contains a description of the fundamental approach used in each analysis. Some prior knowledge of the analyses is recommended to better understand the applications and limitations of the computer programs. Sections IV and VI of Volume I of this Final Report describe the theoretical development of each of the analyses in more depth.

The five programs described in this volume analyze four distinct out-of-plane loading conditions. Figure 1. The Curved Laminate Analysis (CLA) predicts the stresses and strengths of laminate corner radii. The stress states in joints of integrally stiffened structures subjected to direct (fuel pressure) and indirect (panel buckling) loads can be analyzed by the Interfacial Stress Field Model (ISFM) or the Simplified Stiffener Analysis Model

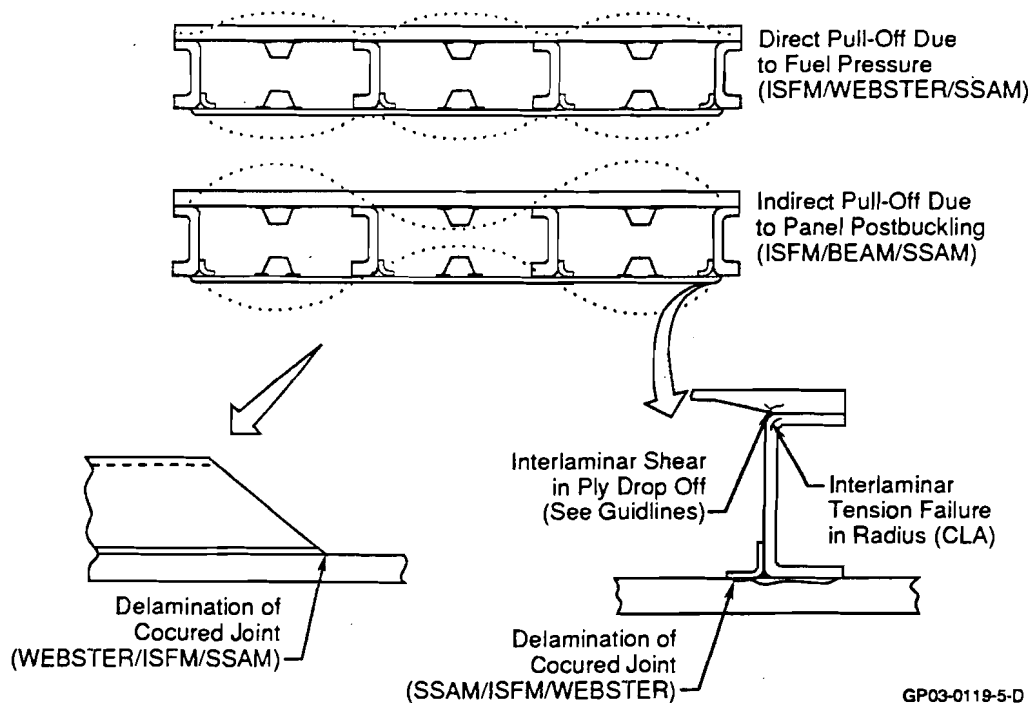


Figure 1. Potential Out-of-Plane Failures in Integrally Stiffened Torque Box Design and Recommended Analysis Routines

(SSAM). WEBSTER, WEB-Stiffener Termination – Elasticity Routine, solves for interlaminar failure of stiffener terminations. BEAM is based on a Rayleigh-Ritz approach, and assesses the stress state at stiffener flange terminations under direct loads. Users of the ISFM, WEBSTER, and BEAM analyses should be especially mindful of the assumed boundary conditions of each analysis.

This volume provides a brief description of each routine along with the input, the output, and example problems. The FORTRAN computer listing for each program is also provided.

SECTION II

DESIGN GUIDE FOR OUT-OF-PLANE ANALYSIS OF COMPOSITE STRUCTURES

Probably the most important facet of designing composite parts that are subjected to out-of-plane loads is determining the critical load conditions that lead to out-of-plane failures. These loads are dependent on the geometry of the structure and its use. Therefore, the few example cases described here can not address all possible geometries and loading conditions. Rather, the objective of this section is to provide some guidelines for the designer/analyst to use in reducing the risk of out-of-plane failures of composite parts.

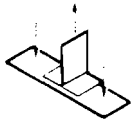
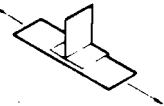
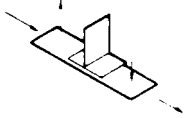
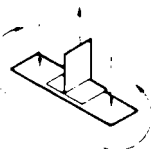
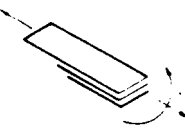
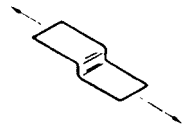
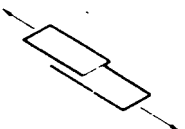
The emphasis of this design guide is to assess and summarize those geometries and load conditions that are known to produce out-of-plane failures in composite structures. The application of the analysis techniques developed in this program to the evaluation of the strength of composite structures is also addressed, as is the determination of local loads. Recommendations are also made concerning the type of element test data required to use these analysis techniques, and the number of such tests required to develop design allowables for out-of-plane load conditions. Wherever possible, preferred design techniques for minimizing out-of-plane loads in composite structures are outlined. Geometries that minimize these loads are identified for a few specific cases.

1. Assessment of Loading Conditions

Loading conditions that were found to induce out-of-plane loads were identified in the initial field survey summarized in Figure 2. The primary loading conditions that precipitate out-of-plane failures are internal and external pressures, changes in load paths (especially in joints), geometric discontinuities, and postbuckled or curved skins. Examples of these primary failures were illustrated in Figure 1.

Finite element models used for analyzing the airframe structure cannot be expected to provide much information about the local out-of-plane loading conditions. The loads that produce the majority of out-of-plane failures (i.e. fuel pressure loads and secondary bending loads) are not addressed by these coarse models. Rather, the analyst will have to examine the structure to identify the areas in which failures (such as those pictured in Figure 2) are possible.

Out-of plane failures must be anticipated in any area involving: fuel pressure loading, postbuckled stiffened skins, curved panels, abrupt changes in load paths (joggles/kinks), bonded joints, and skin splices.

Configuration	Description	Application
 Flatwise Tension	Skin/Stiffener Separation Due to Normal Pressure Loading	Wing Skin/Spar Flatwise Tension Due to Internal Fuel Pressure
 Transverse Tension	Stiffener Web Splitting Due to Transverse Tension Loading	Wing Skin/Spar Transverse Tension Due to Chordwise Loads
 Lateral Bending	Skin/Stiffener Separation Due to Lateral Stiffener Bending	Lateral Spar Bending Due to Asymmetrical Fuel Pressure
 Postbuckling	Skin/Stiffener Separation Due to Post-buckling Deformation and Loads	Stiffened Panels Subjected to Compressive and/or Shear Buckling
 Curved Panel Bending	Interlaminar Tension Due to Panel Bending	Fuselage Skins and Frames Subjected to Bending Loads
 Thickness Transitions	Interlaminar Tension and Shear Due to Axial Loading and Local Bending	Ply Drop Offs, Build-Ups, and Doublers Subjected to Axial Loads
 Irregular Loading	Interlaminar Tension and Local Bending Due to Axial Loading	Joggles and Kinks Subjected to Axial Loads
 Bonded Joints	Interlaminar Tension and Shear Due to Axial Loading and Local Bending	Single and Double Lap Bonded Joints

GP03-0119-35

Figure 2. Typical Out-Of-Plane Loads in Composite Structures

2. Structural Analysis Methods

Once the critical areas for out-of-plane failures have been identified, they must be modeled in sufficient detail that the out-of-plane loads can be accurately computed. This usually cannot be done within the overall structural model, because that modeling is too coarse to determine out-of-plane loads and stresses accurately. This does not necessarily imply that finite element models cannot be used to determine these loads. However, the displacements that produce out-of-plane stresses are generally very small compared to the displacements of the overall structure. These displacements can be so small in some cases, that they are within the error of the recovered displacements from the overall model. It is therefore very difficult to accurately determine critical out-of-plane stresses from large finite element models even when fine grid models are used. The alternative is to use local analyses, either finite element or closed form, to address out-of-plane load problems.

The methodology demonstration performed in this program used simple closed form analyses to model the critical out-of-plane loading conditions. It is these specialized out-of-plane analysis tools that are the primary product of this program. However, the in-plane loads that were applied to these models came from overall structural analyses such as those obtained from finite element analyses. The purpose of this design guide is to show how to use these local models to determine the failure modes and loads for composite structures.

The out-of-plane load conditions and their recommended solution methods can be summarized as

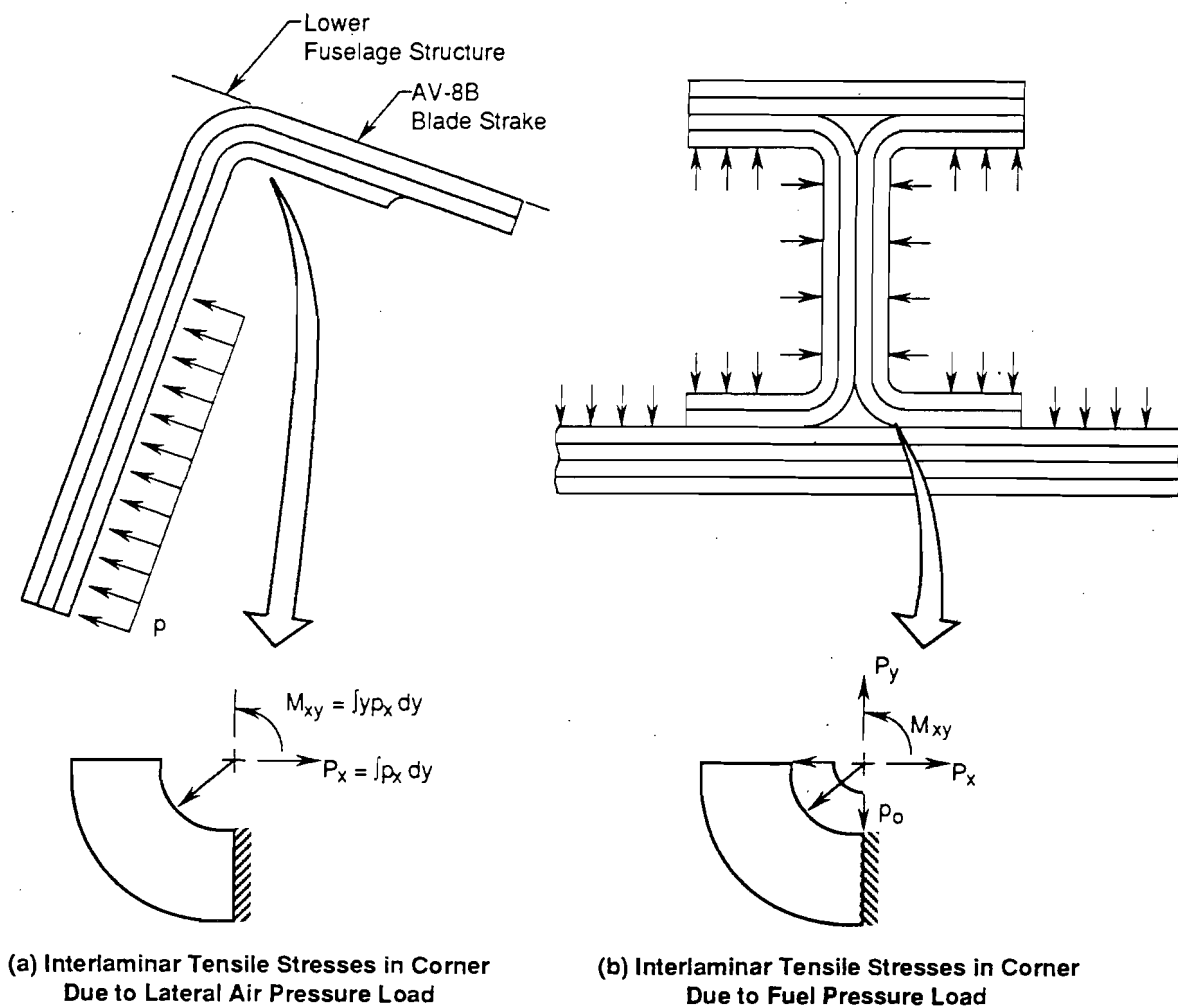
- curved laminates: CLA
- thickness variations: Ply Drop Off Procedures
- direct loadings (fuel pressure, etc.): ISFM, WEBSTER, SSAM
- indirect loadings (stiffener pull off due to impact or postbuckling of skins): BEAM, ISFM, WEBSTER
- indirect loadings (stiffener runout): WEBSTER along with ISFM or SSAM but are best described by Figure 1.

These analyses and their application to specific problems are described below.

2.1 Curved Laminate Analysis (CLA) – Curved laminates are most often used to attach spars and stiffeners to skins. The curved sections are used to redirect out-of-plane loads into the plane of the skin. These loads include various combinations of axial loads, shear loads, and moments. A few other applications of curved

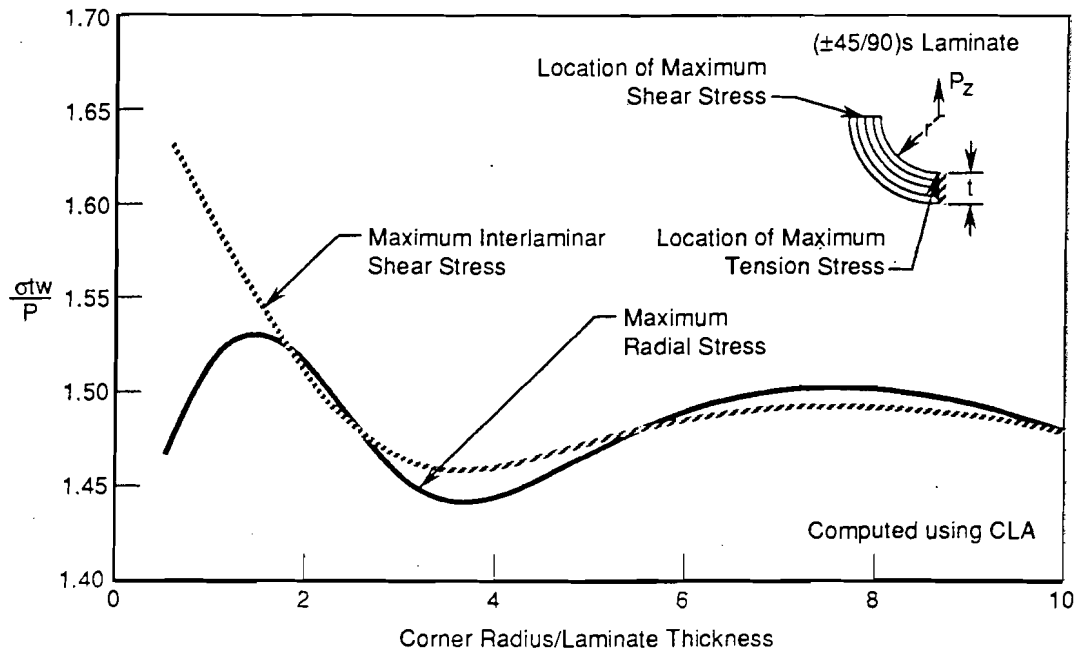
laminates were found, including the AV-8B blade strake discussed in Section III of this report. Even in this case, the purpose of this attachment is to transfer shear loads and moments into the skin and substructure.

The loads applied to curved laminates come from many sources: air pressure (blade strake), fuel pressure (skin to spar), and skin buckling (stiffener to skin). Once these loads are resolved into their normal and lateral load components and moments, as shown in Figure 3, the curved laminate analysis routine, CLA, can be used to determine the loads at which interlaminar shear or tension failures occur. Some examples of the results of these analyses are shown in Figures 4 and 5.

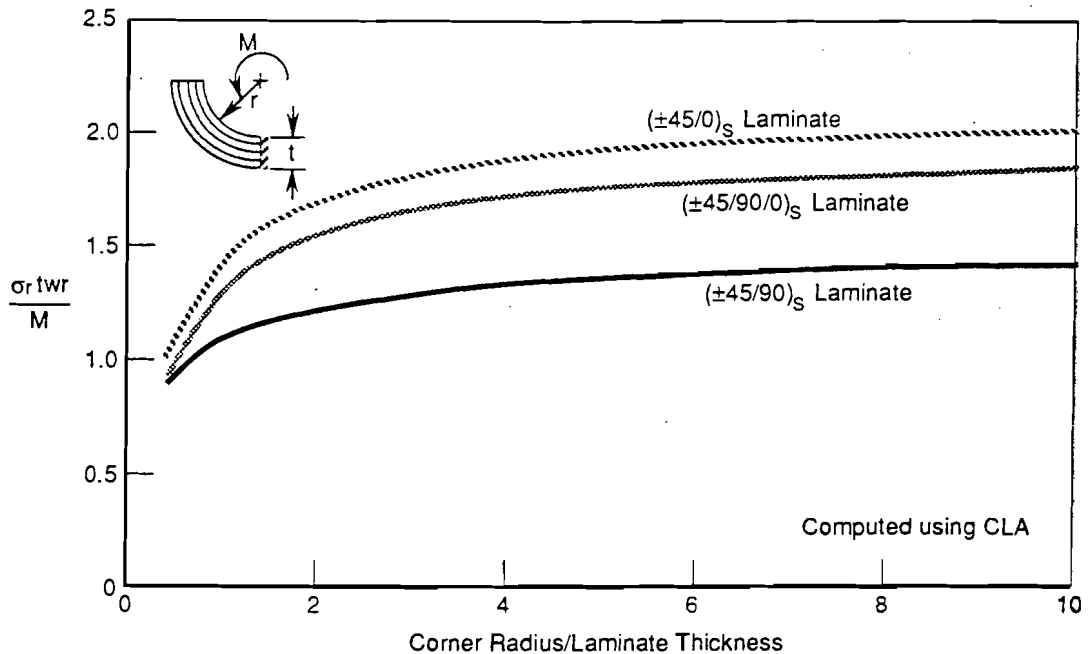


GP03-0119-7-D

Figure 3. CLA Uses Loads Defined at the Center of the Radial Arc



GP03-0119-58-D

Figure 4. Maximum Interlaminar Stresses in Corner Due to Axial Loadings

GP03-0119-59-D

Figure 5. Maximum Interlaminar Tension Stresses Due to Corner Moment

Often in fabricating curved laminates of small radii, resin rich areas are formed in the curved section. As shown in Figure 6, the resin rich area can be treated as a ply having neat resin properties. By choosing to represent the thickness of this layer as the average thickness of the resin rich area, CLA can be used to predict the effect of this resin rich area on interlaminar stresses. The results show that the effect of the resin rich area is to

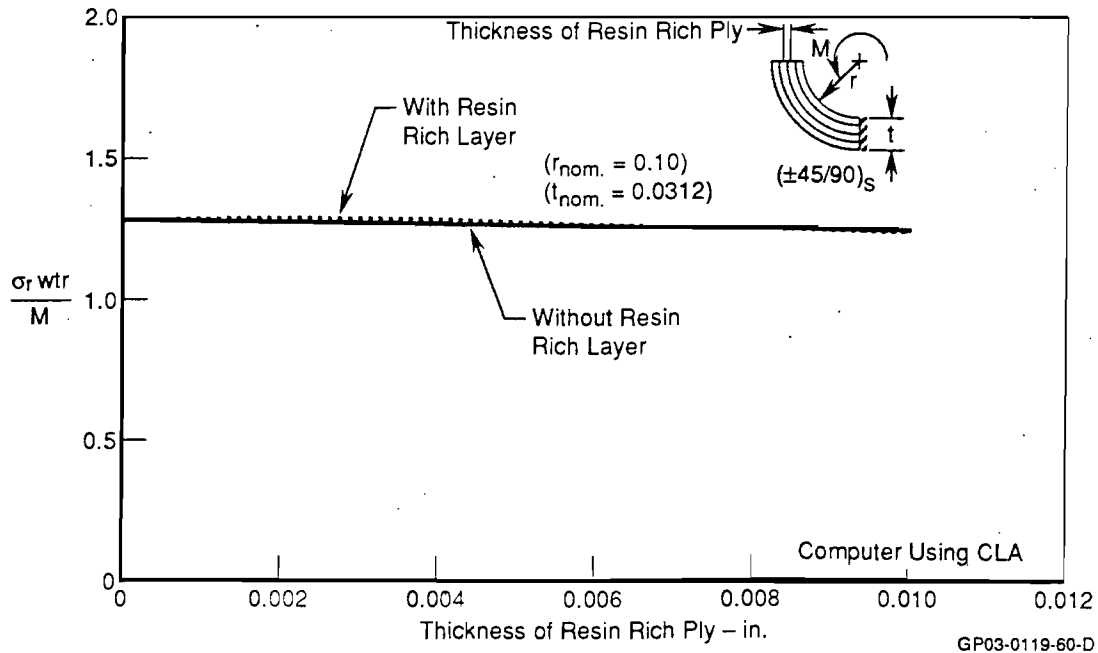


Figure 6. Effect of Resin Rich Area on Maximum Interlaminar Tension Stress Due to Applied Moment

reduce stresses slightly as its thickness increases. The effect is so small that just assuming the laminate thickness to increase by the thickness of the resin rich area gives nearly the same results, as shown in Figure 6. This analysis assumes that the resin rich area occurs in the center of the ply, which may not be the case in real parts. Regardless of where the resin rich area resides within the laminate, however, CLA can be used to estimate its effect on interlaminar stresses and failures.

Another important source of these transverse tension and shear loads is skin to substructure mismatches where the skin and the flange are required to deflect and close gaps created by the mismatch. One model of this type of loading is shown in Figure 7. Present requirements mandate gaps such as these be shimmed to minimize the out-of-plane loads due to mismatches. However, when skin and substructure have sufficiently low stiffnesses or gaps within tolerances, closure of the gap may be permitted through fastener torque-up. The permissible closure of a gap can generally be determined using the CLA method, as presented in Figure 8.

Failures in curved laminate radii are generally not catastrophic. The primary failure mode is delamination within the curved section, reducing the bending stiffness of the joint but rarely reducing its axial strength. These delaminations do change the load distributions in the flange attachments. Sometimes the axial joint strength is enhanced because of the reduced moment. Other times the strength is reduced if the primary loading is shear, as in the blade strike.

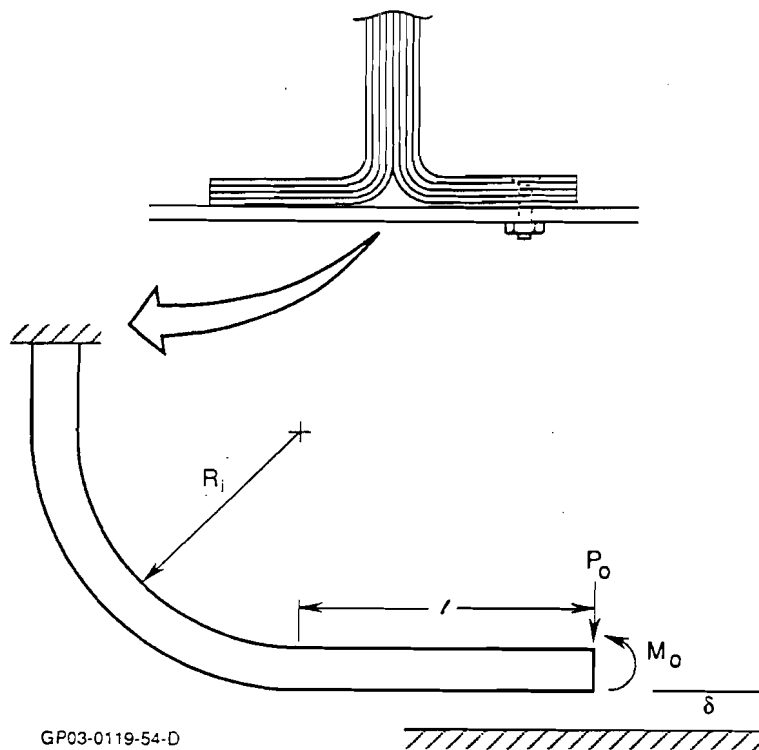


Figure 7. Flange Gap Model

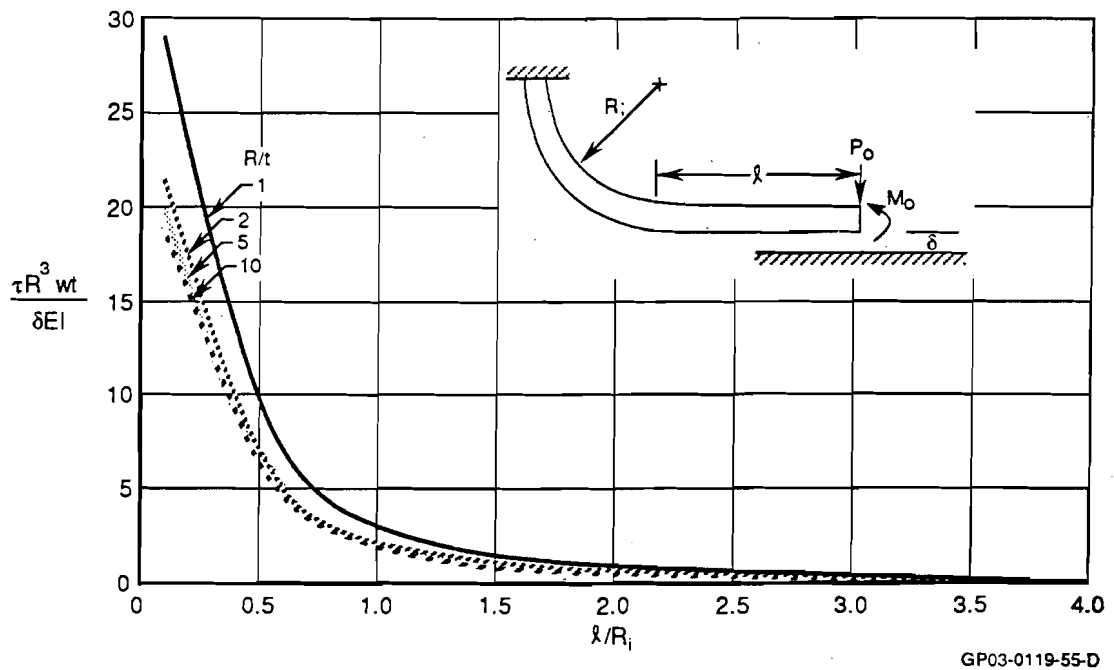
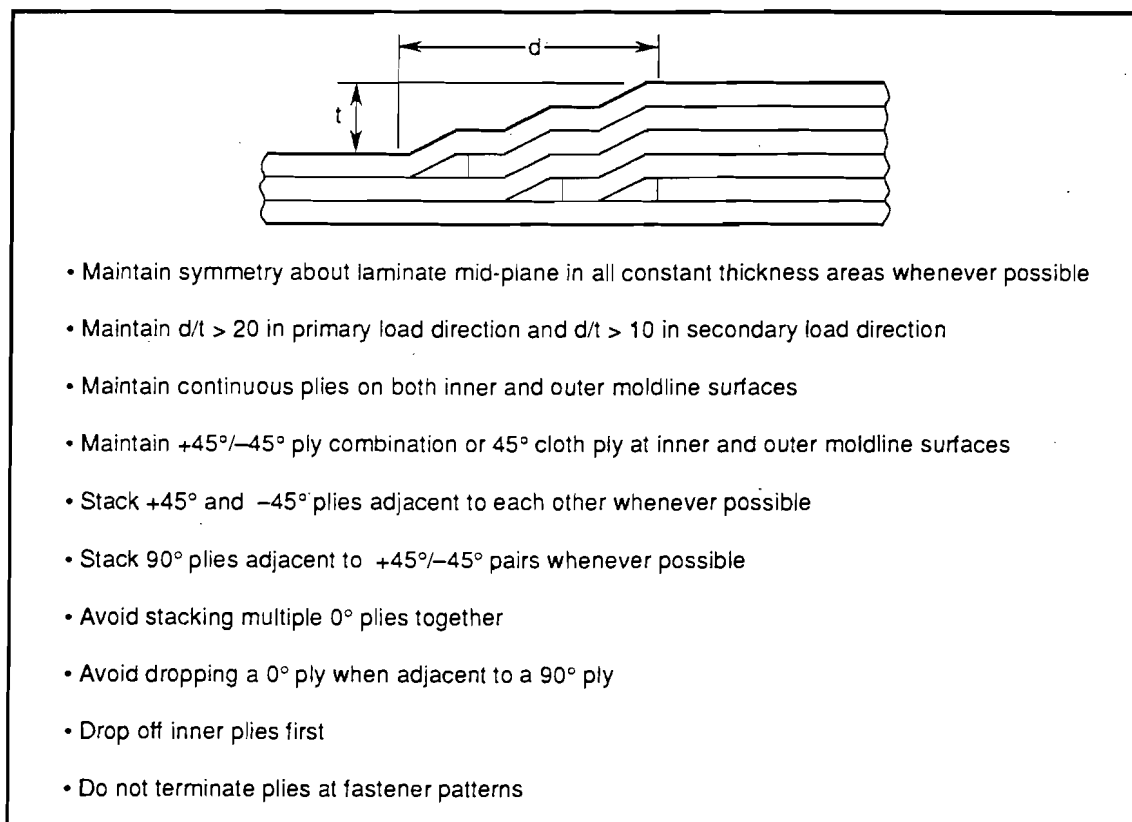


Figure 8. Maximum Interlaminar Shear Stress as a Function of Gap Size

2.2 Ply Drop Off Procedures – The thickness of a composite laminate is varied by adding or deleting plies from the primary layup. In Reference 5, Northrop used a fracture mechanics approach to predict the initiation and propagation of a delamination ahead of a ply drop off. It was predicted from this analysis that gradual changes in laminate thickness and stiffness generally do not initiate or perpetuate delamination growth. Other analyses and tests, performed both at MCAIR and Northrop, demonstrate that a single ply drop off does not generate sufficient strain energy to produce a delamination within a laminate.

Out-of-plane loads caused by thickness changes can be minimized by using established ply drop off design procedures. These have been based on many years of experience at both MCAIR and Northrop, as well as other aircraft companies. MCAIR and Northrop have developed some basic procedures from this experience. These guidelines/procedures, listed in Figure 9, have been proven to produce durable thickness variation in composite structures.



GP03-0119-28-D

Figure 9. MCAIR/Northrop Ply Drop Off Design Procedures

2.3 Stiffener Pull-Off Analyses – Most of the analyses developed in this program were derived primarily to examine pull off failures of skin/stiffener joints. Therefore, BEAM, ISFM, WEBSTER, and SSAM are all applicable to failure predictions of stiffener/skin joints under direct pull off loads. Each of these routines has its own set of assumed loads and boundary conditions, Figure 1, so selection of the analysis is determined by these conditions. BEAM is most appropriately used in cases where the flange end is the likely failure location. Stiffened skins that are stitched through the stiffener flanges to prevent significant stresses beneath the web provide an example of such a case. ISFM is most appropriate in cases in which failures can occur beneath a discontinuous web. WEBSTER is a more general method but requires the most input data to run. SSAM is a simple two dimensional finite element based technique that is the most general method. It was developed under MCAIR IRAD funding very late in the course of this program.

The most important part of these analyses is determining the loads and boundary conditions that most closely match the analysis and the structure of interest. The determination of the direct loads and examples of the application of these analyses to these loadings are addressed in the following paragraphs.

2.3.1 Direct Loadings – Fuel Pressure – Fuel pressure loads precipitated the failure of the MCAIR High Strain Wing Box. The computation of that fuel pressure load is described in detail in the Methodology Demonstration summary. However, the basic method for determining maximum pull off loads due to fuel pressures is a classical approach from Timoshenko, Reference 14.

Timoshenko shows that the restraining loads on the edges of a simply supported panel subjected to a uniform lateral pressure are represented by a sinusoidal distribution such as that shown in Figure 10. The peak load occurs in the middle of the shortest edge and has an approximate magnitude of $P=pa/2$, where P is the maximum pull off load, p is the lateral pressure, and a is the length of the edge. The magnitude of this load does not change appreciably with panel geometry, as shown in Figure 11. This load is also independent of panel stiffness.

To demonstrate the application of these analysis techniques to stiffener pull off under direct loads, let's examine the case of a bonded spar cap to skin joint under direct out-of-plane loads, like fuel pressure.

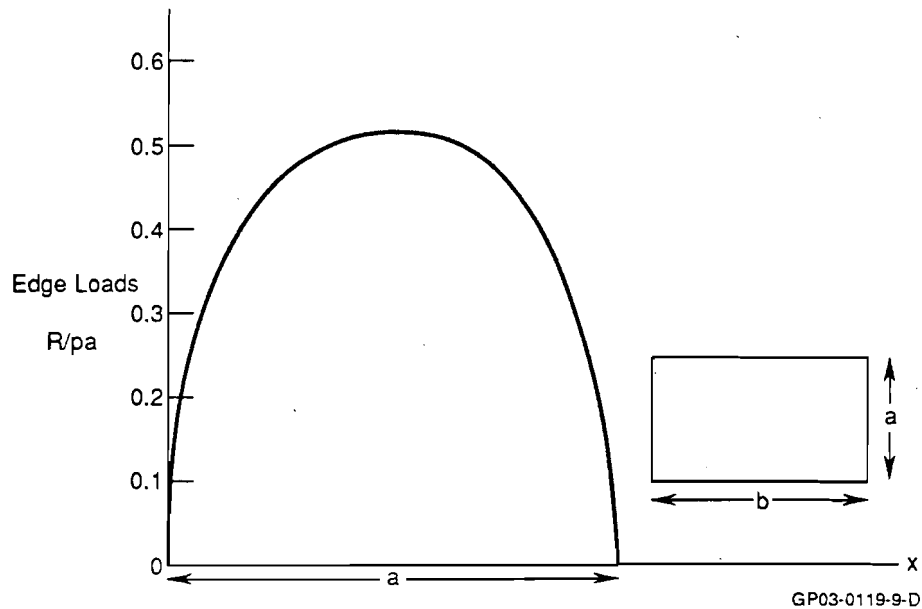


Figure 10. Edge Loading Due to Lateral Pressure on a Panel

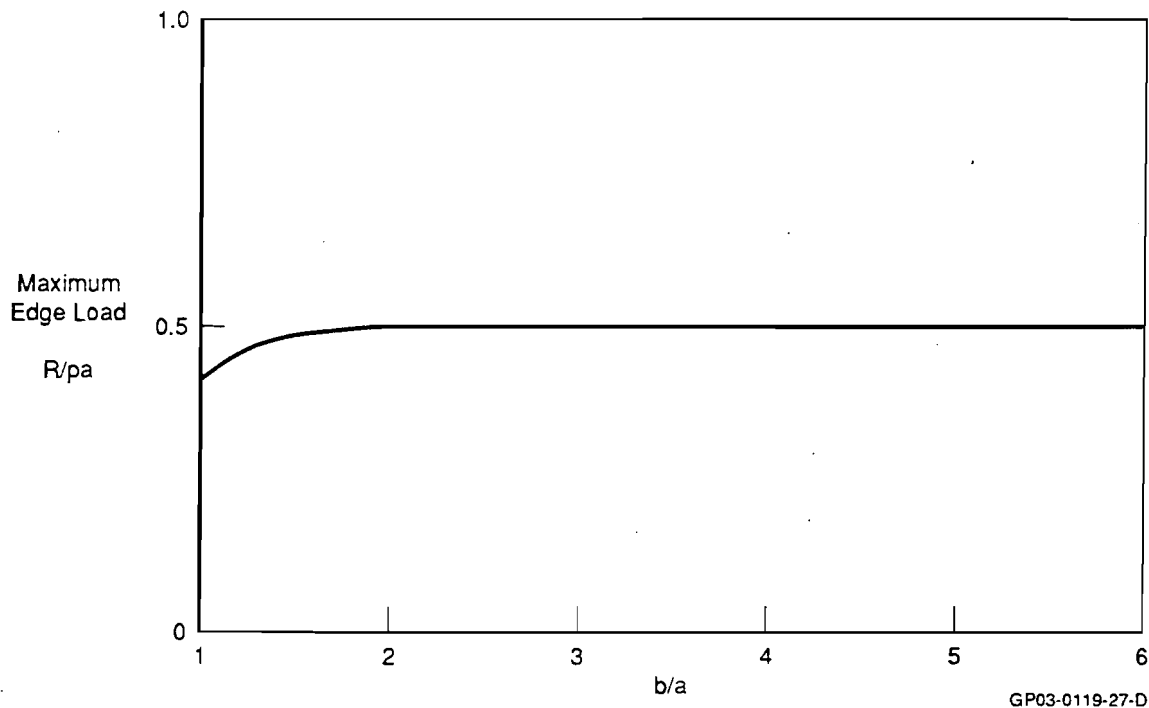
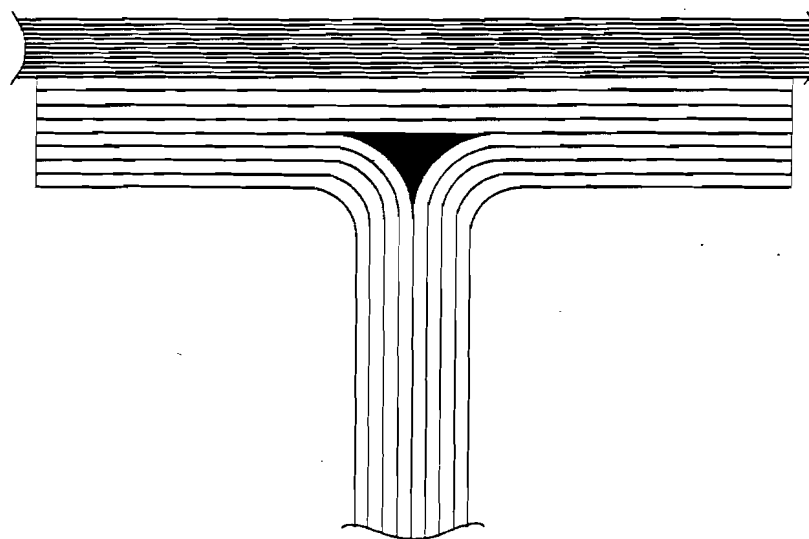


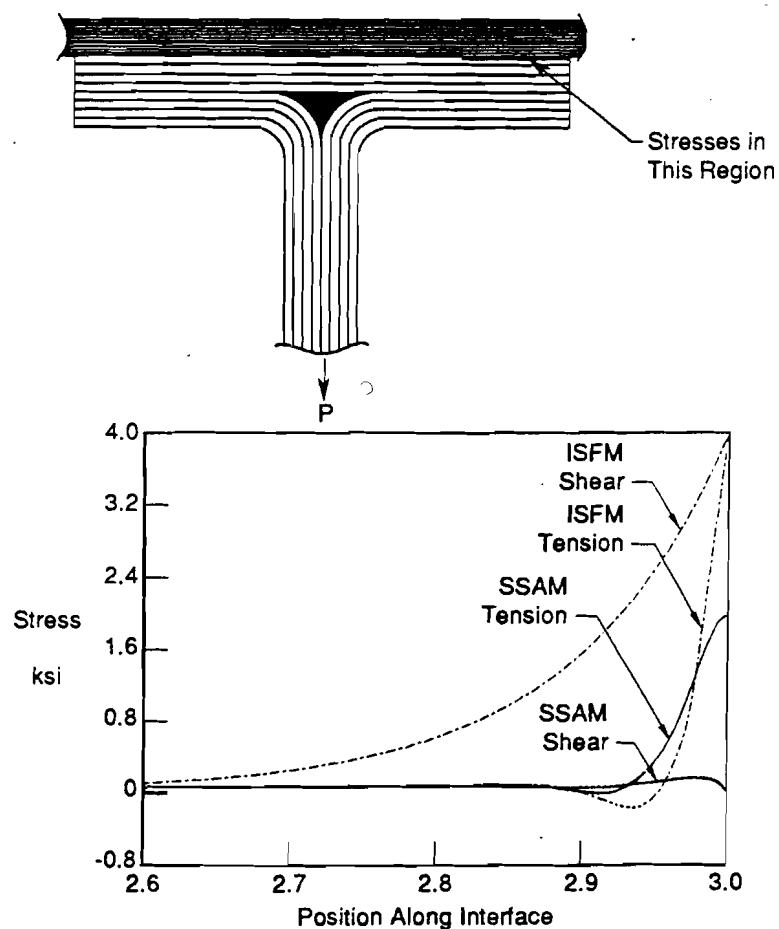
Figure 11. Effect of Panel Geometry on Maximum Edge Load

Let's assume the initial joint design looks like that shown in Figure 12 and that both skins and flanges are fabricated from AS4/3501-6 material. Both ISFM and SSAM analyses can be applied to this analysis and give the results shown in Figure 13. Note that tension stress distributions at the end of the bondline are similar, however, because ISFM uses a simple shear lag analysis to compute shear stresses (independent of the tension), higher shear stresses are predicted with ISFM than with SSAM. Because SSAM accounts for the coupled effects of tension and shear stresses in the adhesive, it is considered the preferred analysis for this case.



GP03-0119-62-D

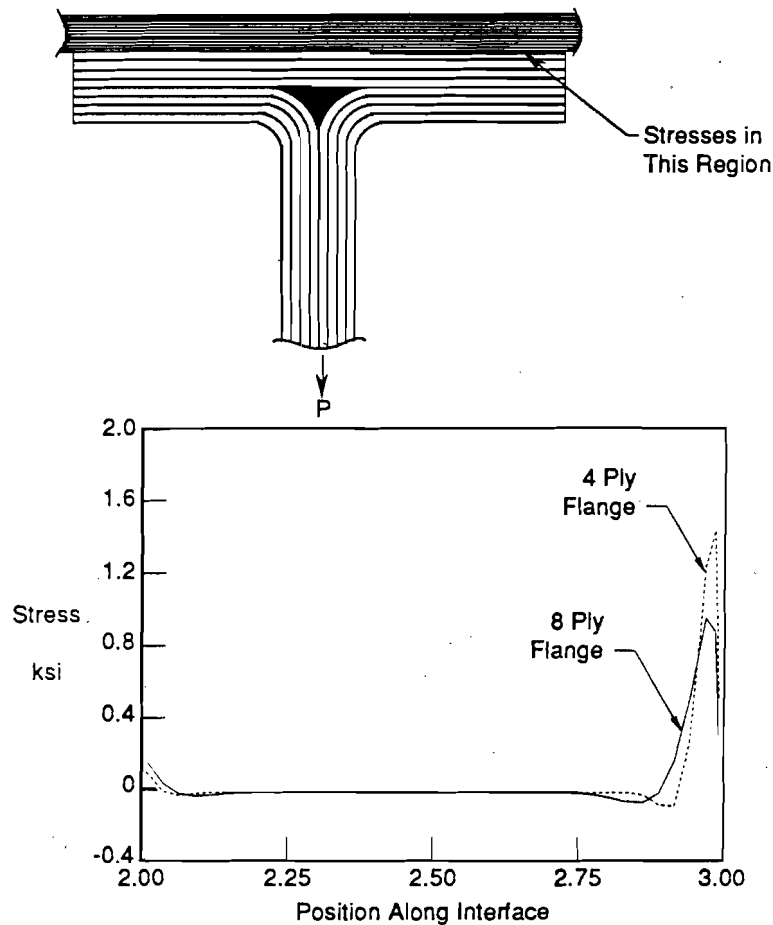
Figure 12. Spar Cap Design



GP03-0119-51-D

Figure 13. Comparison of ISFM and SSAM Results for Stresses at the End of a Spar Cap Flange

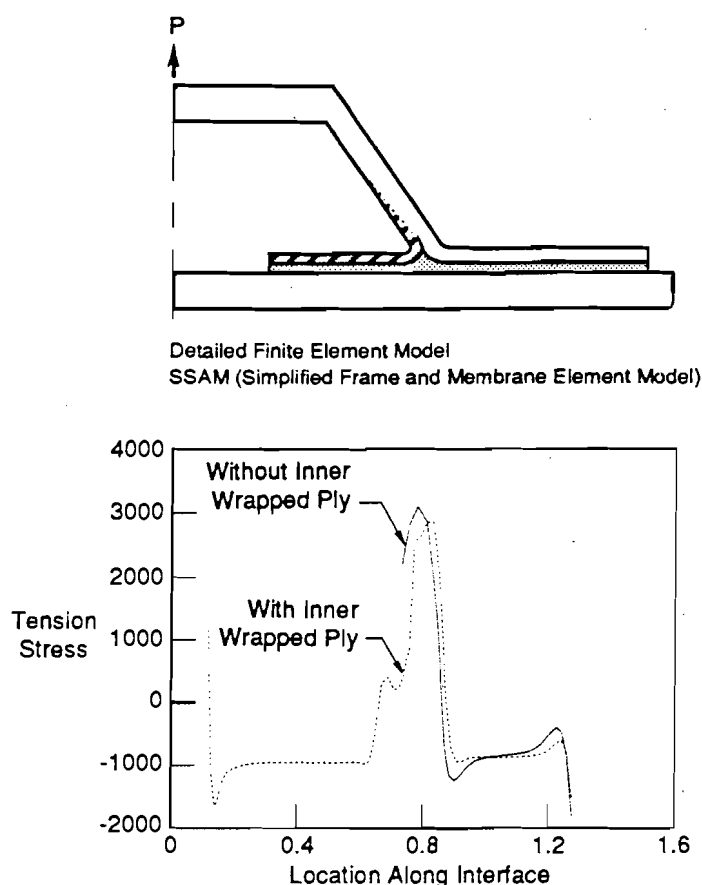
As flange stiffnesses are increased, the tension stress distributions become increasingly critical at the free end of the flange and less critical beneath the web, Figure 14. Eventually a flange can be designed that is so stiff that failures are initiated at the free end, but as shown in Figure 14, that stiffness would be much higher than normally used.



GP03-0119-52-D

Figure 14. Effect of Increased Flange Stiffness Under Pull-Off Load (SSAM Analysis)

During the analyses performed using SSAM, it was found that an interior wrapped ply does not substantially improve flange to skin joints for reasonably designed joints. Consider the hat stiffeners shown in Figure 15, one of which has an interior wrapped ply, the other not having such a ply. SSAM analyses show that the predicted tensile stresses for either geometry are not significantly different. Tests are required to verify this prediction, but, if true, one could justify a significant cost reduction in the fabrication of hat stiffened panels could such a ply be eliminated.



GP03-0119-45-D

Figure 15. An Inner Wrapped Ply Does Not Improve Pull-Off Stresses Greatly
(SSAM Analysis)

2.3.2 Indirect Loadings – Postbuckled Panels – Out-of-plane failures due to indirect load sources are among the most difficult to analyze. The primary causes of these loads are postbuckled structures and geometric discontinuities. Out-of-plane deformations of postbuckled panels induce loads on stiffeners, ribs, and spars. Geometric discontinuities, such as stiffener runouts, induce out-of-plane loads as cap loads are transferred to the skin. These cases will be addressed separately, because two separate methods are used to analyze the effects.

BEAM and ISFM are two dimensional beam models developed specifically to determine skin/stiffener separation for flat stiffened panels under combined compression and shear loadings. Both analyses account for panel geometry, material anisotropy, and stiffnesses of the structural components. However, early in the program ISFM was selected as the preferred technique because it permitted direct computation of both tension and shear stresses within the bondline between stiffener and skin.

ISFM is based on a beam on elastic foundation approach, Reference 6, and computes the out-of-plane interfacial stresses caused by the skin buckling away from supporting stiffeners. Model dimensions are determined by orienting the beam model along the buckle crest, as determined from a panel buckling analysis (Figure 16). The ISFM model development is described in Section IV of the Final Report. ISFM computes both interfacial tensile and shear stresses due to the deflection of the skin and the stiffener flange (Figure 17). The model predicts that increased flange stiffness will increase the interfacial stresses between the flange and skin, as shown in Figure 17.

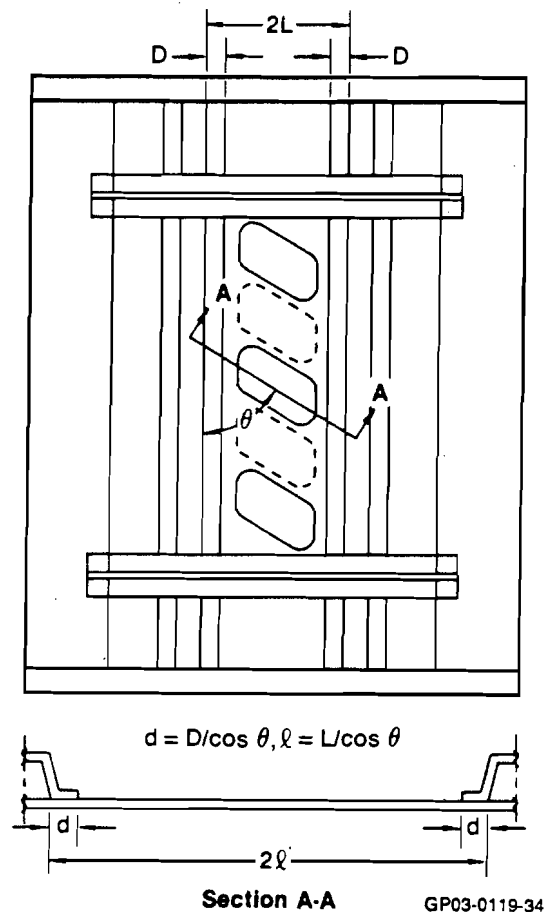


Figure 16. Model Geometry for ISFM Analysis of Skin/Stiffener Separation

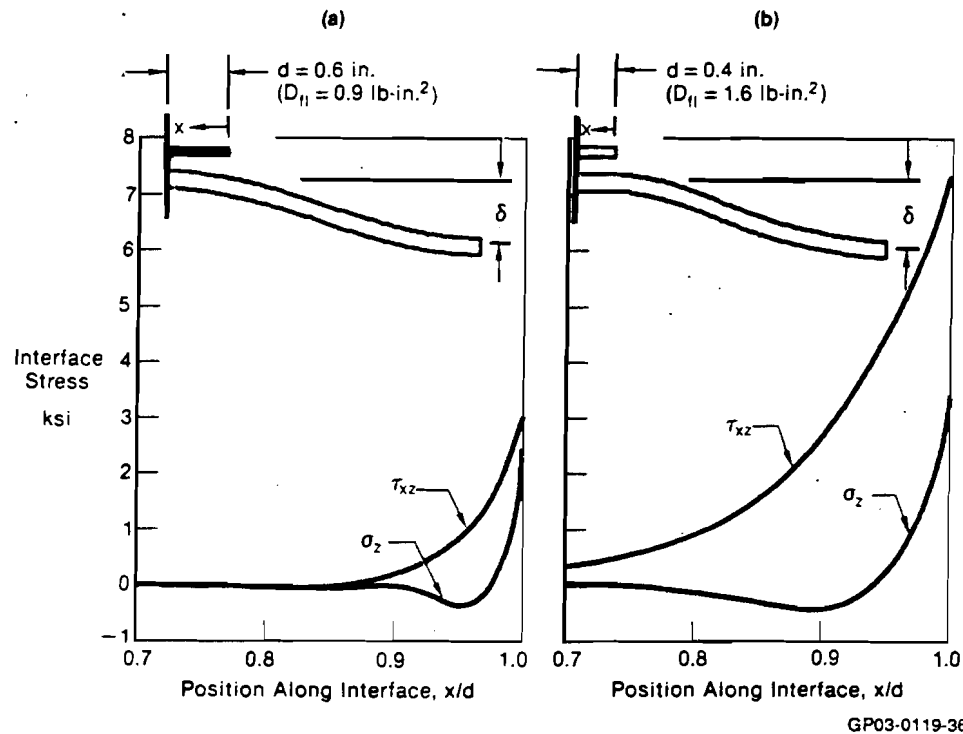


Figure 17. Flange Geometry Affects Interface Stress Distributions

Figures 18-20 show predictions of the effects of flange stiffness and skin stiffness on stress distributions in indirectly (displacement boundary conditions) loaded joints. Figure 18 shows a comparison between ISFM and the SSAM finite element analysis, developed late in the program under IRAD funding, for bondline stresses. For these loading conditions the two analyses disagree significantly, in contrast to the analogous comparison for direct pull off shown earlier in Figure 13. Because SSAM accounts for the bond continuity across the stiffener web, similar results were found for all the load and boundary conditions examined although only the SSAM results are shown for the parametric analyses shown in Figures 19 and 20.

2.4 Geometric Discontinuities – Geometric discontinuities, such as cut-outs, stiffener runouts, and thickness variations cause complex loading conditions that are inherent in almost every aircraft component. Integrally stiffened structures are designed to reduce the weight and number of parts used. However, these types of structures often introduce load path discontinuities, especially at stiffener runouts.

The geometric complexity of built up structure produces stress concentrations and load eccentricities, which must be analyzed to ensure that failure does not occur under design loads. Some areas, including stiffener and frame terminations, are susceptible to failure due to out-of-plane loads. These structural members are designed to

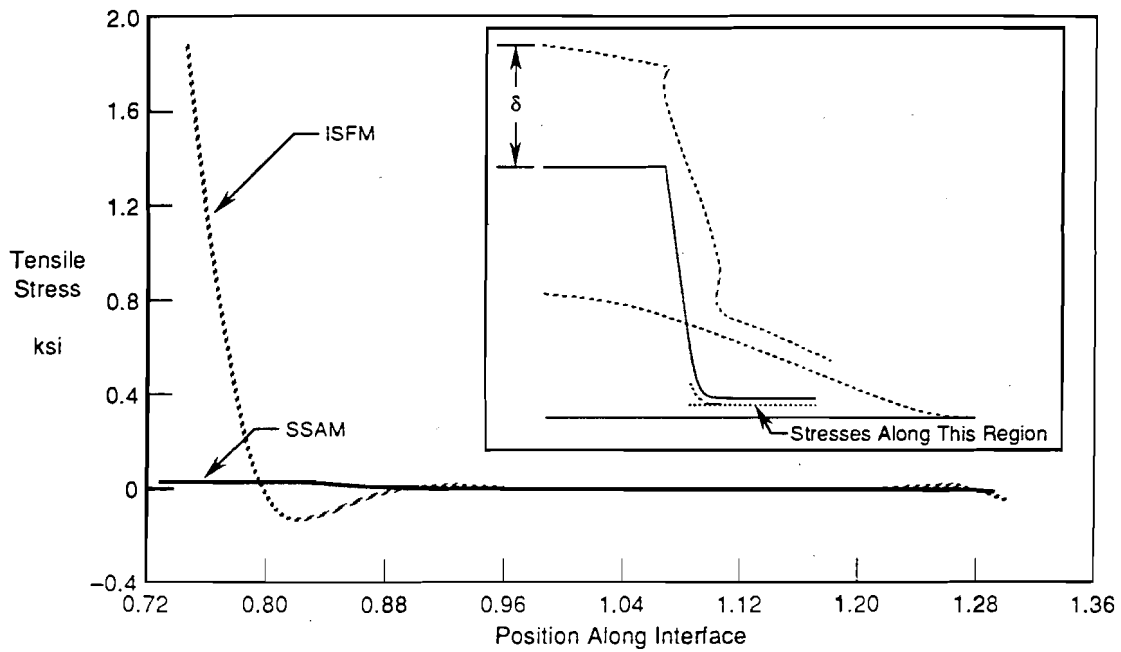


Figure 18. Comparison of ISFM and SSAM Results for Hat Stiffener Loaded by Skin Displacement

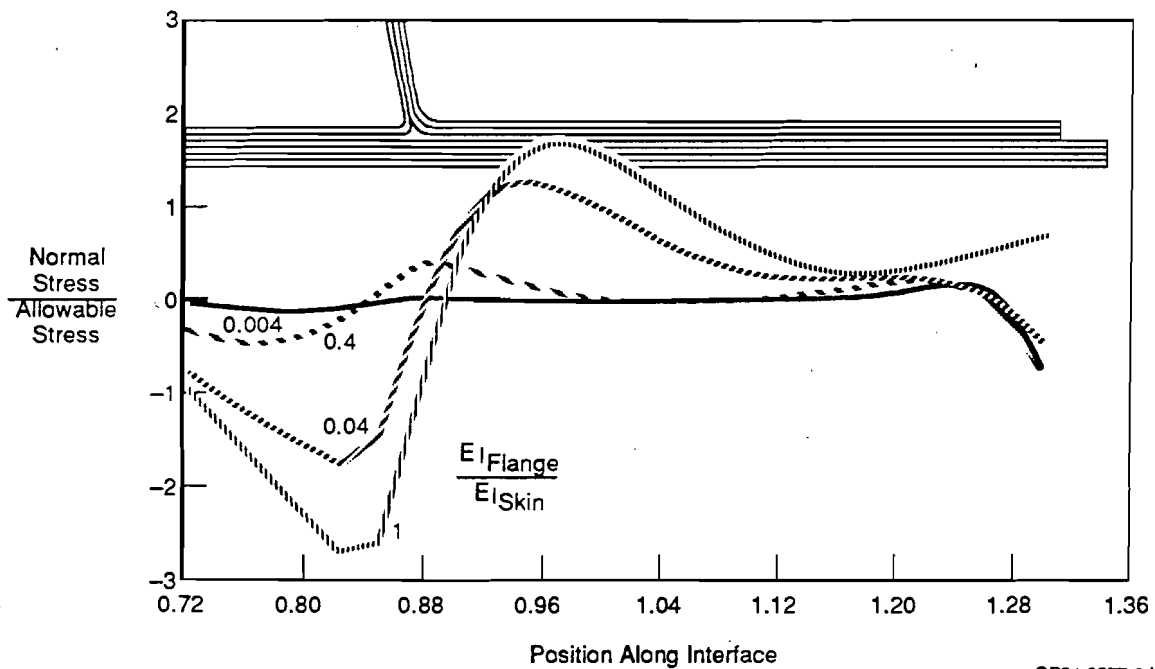
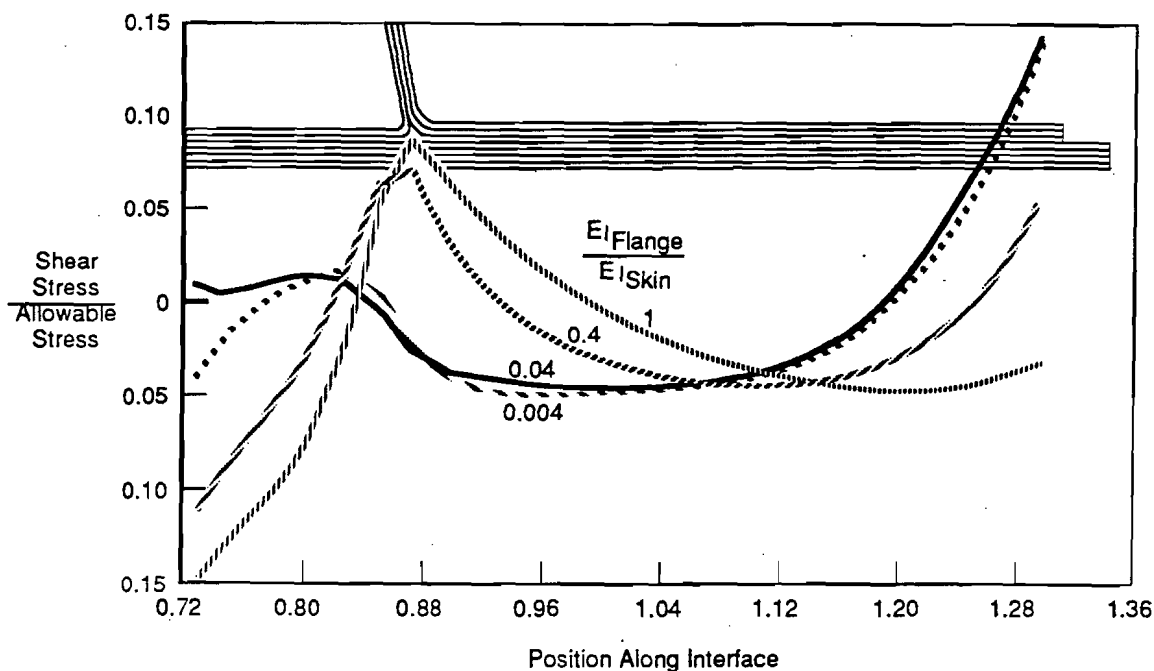


Figure 19. Effect of Flange EI for Prescribed Displacement Case



GP34-0277-5-V

Figure 20. Effect of Flange EI for Prescribed Displacement Case

carry a load or provide some form of support. At their termination point, the load being carried by the structure must be transferred into an adjoining member. This transfer of load usually causes a stress concentration in the joint and the potential for an out-of-plane failure of the structure.

The interface between two composite members may be cured, adhesively bonded, bolted, or any combination of these forms. This region could potentially fail due to the normal and shear stresses, regardless of the attachment technique. The ultimate strength of the interface, in terms of the combined ultimate normal and shear stress values, can be computed using WEBSTER or any other pull off analysis. However, the loads applied in the WEBSTER analysis must come from an analysis that accounts for the loading termination and geometry of the runout. Both analysis and test results show that the stresses at the very end of the stiffener/frame can exceed the point stress or strain failure criteria without causing failure. It is therefore assumed that failure occurs over a characteristic area (a_0). This averaging of stress in a high gradient area is typical of cut-out analyses.

2.4.1 Stiffener Runouts – Stiffener runouts present a difficult problem for two dimensional analyses in predicting the stress state in the runout. Loads carried by the stiffener are transferred to the adjoining skin at the runout, creating a stress concentration at this geometric discontinuity. The stiffener runout geometry and loading condition require that the problem be analyzed in three dimensions. Since only two dimensional analyses were derived for this program, the analytical approach to the stiffener runout problem combines two analyses to determine the stress state.

The stiffener runout is divided into two separate problems for this analysis. The first considers the stiffener/frame termination. The elasticity approach, which is the basis of WEBSTER, addresses these terminations. The stiffener runout is modeled along its longitudinal direction, represented as Cut A in Figure 21. The stiffener termination angle of the model (β) is assumed to be the same angle of the actual structure. If the stiffener experiences a nonlinear stiffness change in the termination, the termination angle may need to be altered to reflect this change. Generally, the termination angle may be varied to match the slowest and fastest changes in stiffness (and perhaps an intermediate value as well) to determine the maximum pull-off loads in the runout. The global loading condition is applied to the model to determine interfacial stress distributions.

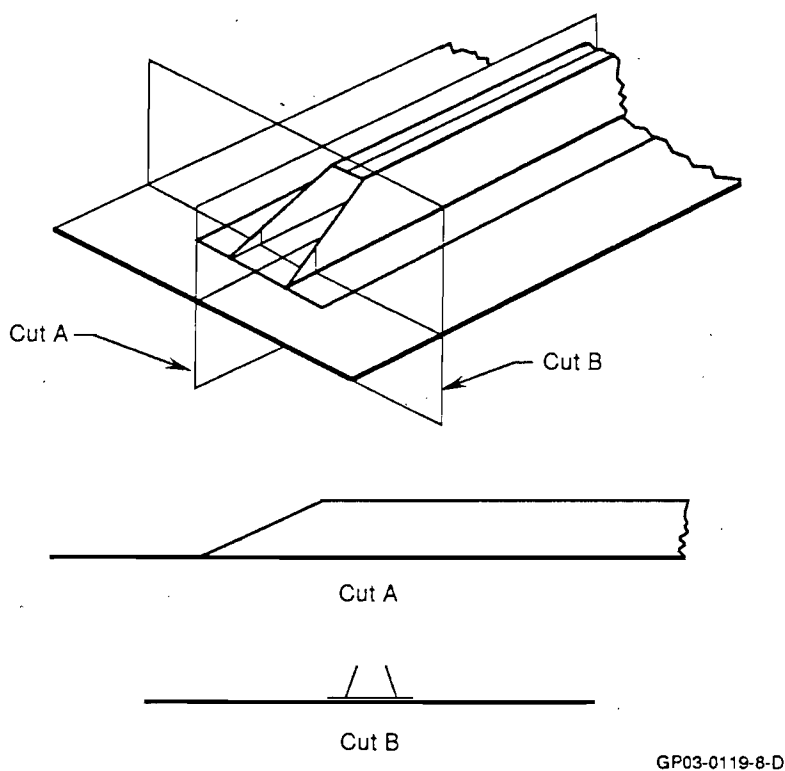


Figure 21. Stiffener Runout Analysis Technique

The second portion of the analysis considers the transverse direction of the runout, represented as Cut B in Figure 21, and determines the direct load pull off strength of the stiffener at any particular section. Any of the pull-off analyses, BEAM, ISFM, or WEBSTER, can then be used to determine the interfacial stress distributions of the skin/stiffener joint in the transverse direction. This analysis is performed using the pull off load determined from the normal stress distribution of the previous longitudinal analysis. The stiffener runout strength is determined by applying the quadratic failure criterion to the normal and shear stress distributions obtained from the transverse analyses according to the pull-off loads determined from the longitudinal analyses.

2.4.2 Example Analysis – In the following example, WEBSTER and ISFM are used to analyze a stiffener runout and to redesign the runout to provide better strength. WEBSTER is used to model the longitudinal loads and stresses and to determine pull off loads developed by the cap termination even under planar load conditions. ISFM is used to determine pull off loads for the critical sections identified in the WEBSTER analyses. WEBSTER analyses are linear, elastic and will be used to determine the critical section assuming displacement of the skin are small. When such terminations induce large moments, the skin can act as a beam column and a nonlinear, beam column analysis should be used. For small displacements, however, the WEBSTER analysis should be valid.

In this example, we address a structure like that shown in Figure 21. Overall finite element analyses are used to determine the load condition shown in the figure. The purpose of this analysis is to assure the structural integrity of the bond of the stiffener to the skin. As a first step, we compute the section properties over the length of the stiffener runout including an effective width of skin determined through a crippling analysis, as was done in the methodology demonstration presented Section V. Once the effective EIs are determined then the WEBSTER model is generated to simulate the stiffener termination. The slope of the termination (β) is selected to match the change in EI, as closely as possible, with distance through the stiffener termination. The section properties and WEBSTER model are compared in Figure 22.

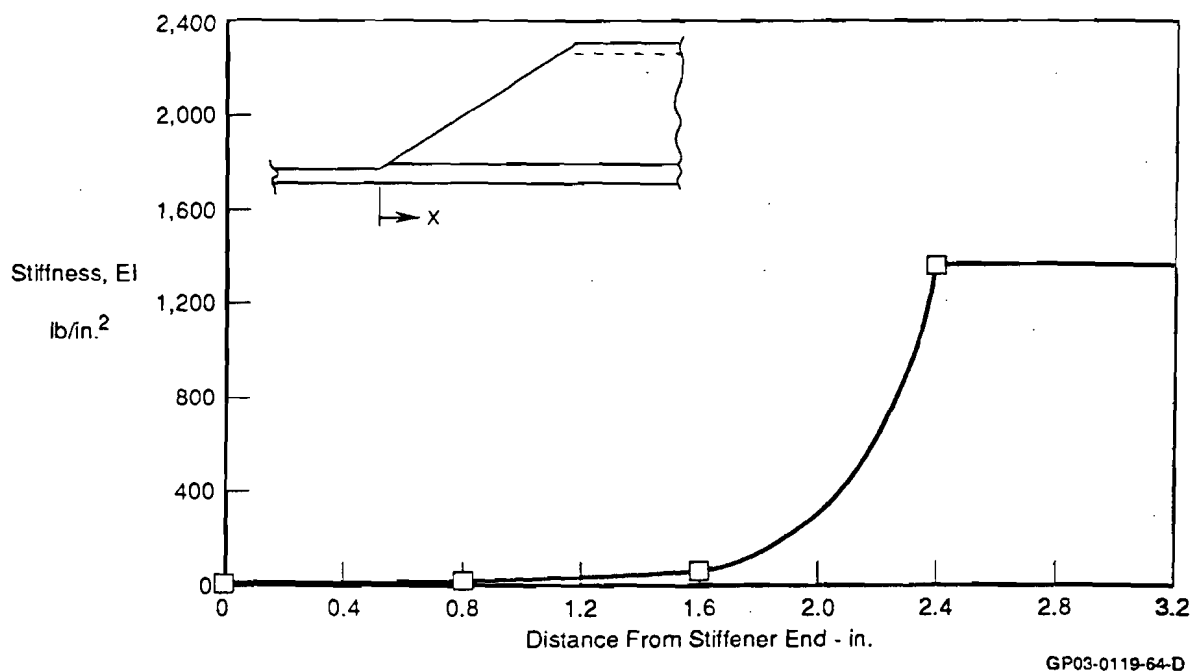
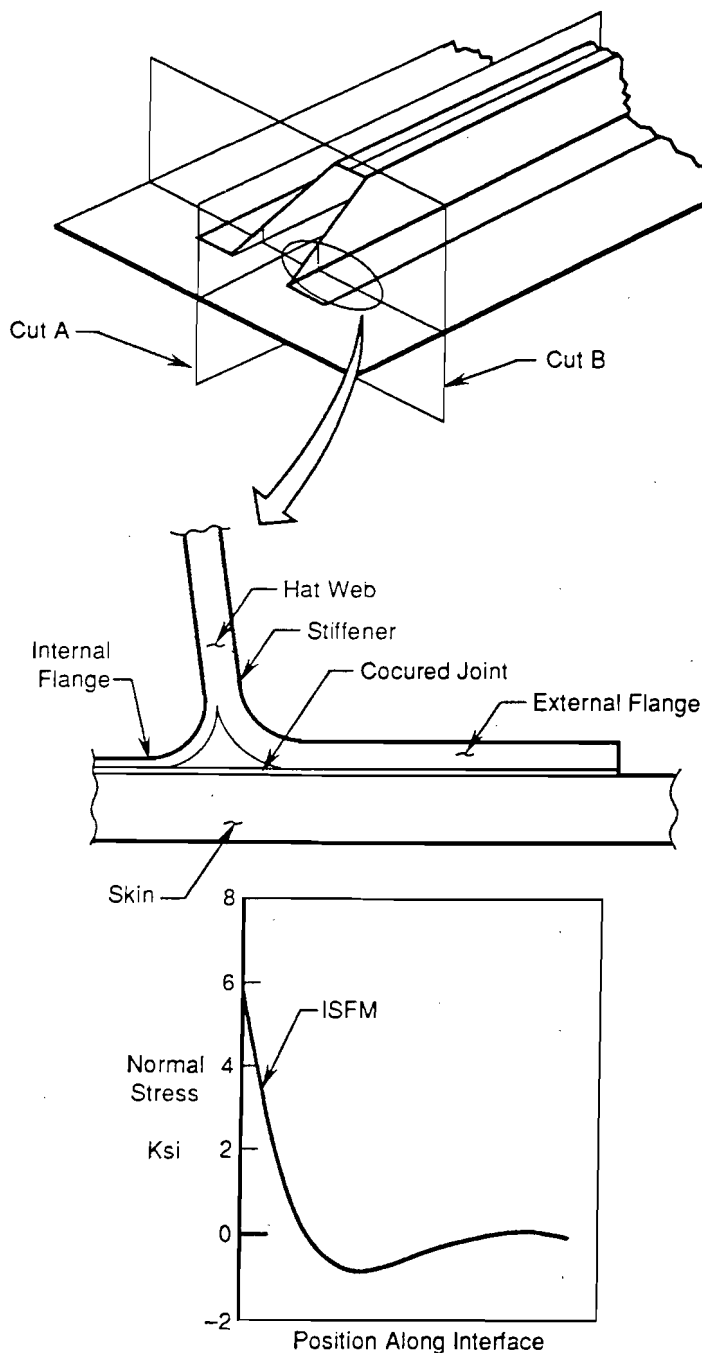


Figure 22. Stiffness Distribution for Stiffener Runout

The WEBSTER analysis predicts that the maximum out-of-plane loads will occur at the farthest end of the runout (Figure 23). ISFM analyses were performed at four sections throughout the runout to determine the pull off load carrying capability of the joint along the runout. The pull off loads, predicted by WEBSTER, are divided by the load carrying capability, predicted by ISFM, to determine the critical location in the stiffener



GP03-0119-67-D

Figure 23. Stiffener Flange Analysis

runout (Figure 24). Analysis of this critical section and comparison with the applied loads allows one to determine the margin of safety provided by this runout.

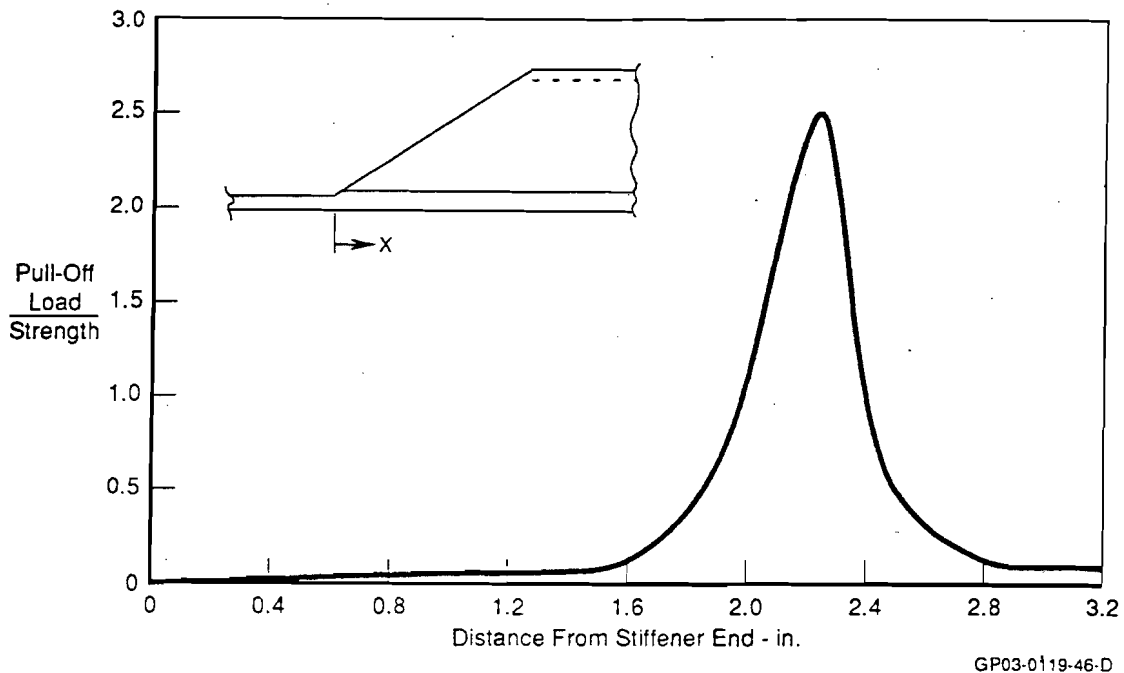


Figure 24. Pull-Off Load Peaks at End of Cap for Open End Hat Stiffener

A second analysis was performed to determine the effect of using a closed end stiffener runout to provide a better load transition between the stiff cap of the stiffener and the skin through the runout, Figure 25. Results of the WEBSTER and ISFM analyses are shown for this runout in Figure 26. This runout provides much higher strengths than the runout without a cap, but also induces higher pull off loads. The result is only a very small increase in load carrying capability with the closed end cap stiffener.

If failures predicted for these analyses occur at much higher loads than those for which the WEBSTER analysis was performed herein, a beam column analysis would have to be used to perform the initial assessment of critical pull off loads. The nonlinear nature of beam column displacements versus loads would not allow us to accurately predict failure via linear extrapolation of the WEBSTER results.

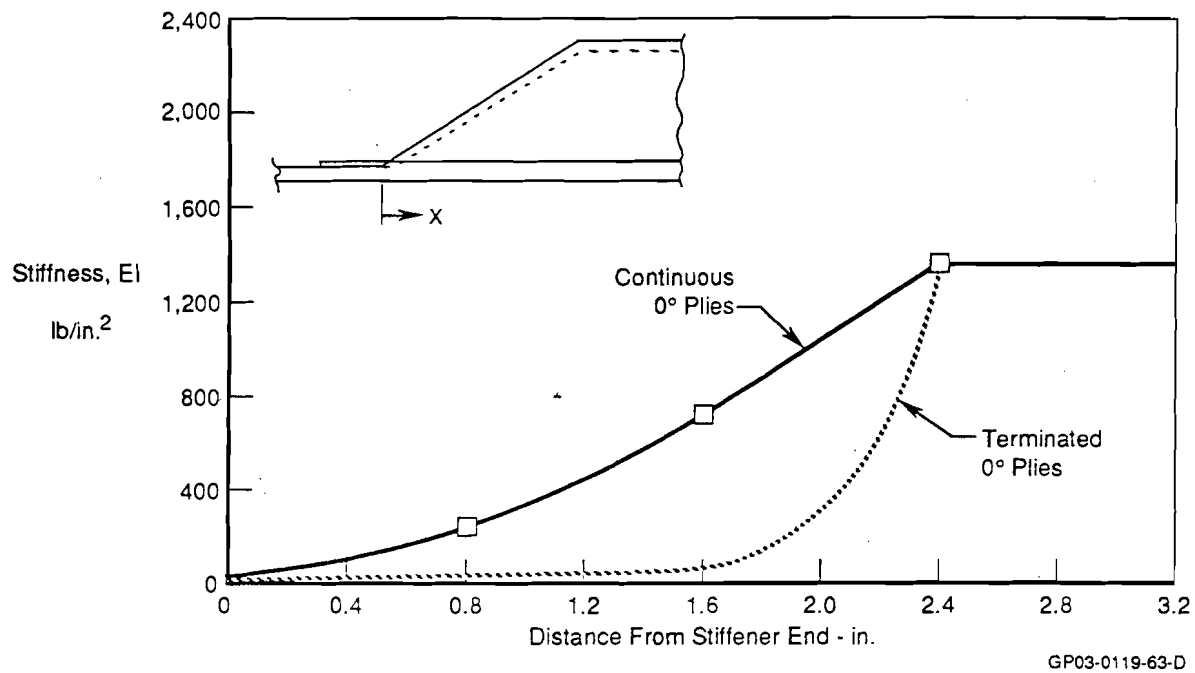


Figure 25. Stiffness Distribution for Closed End Stiffener

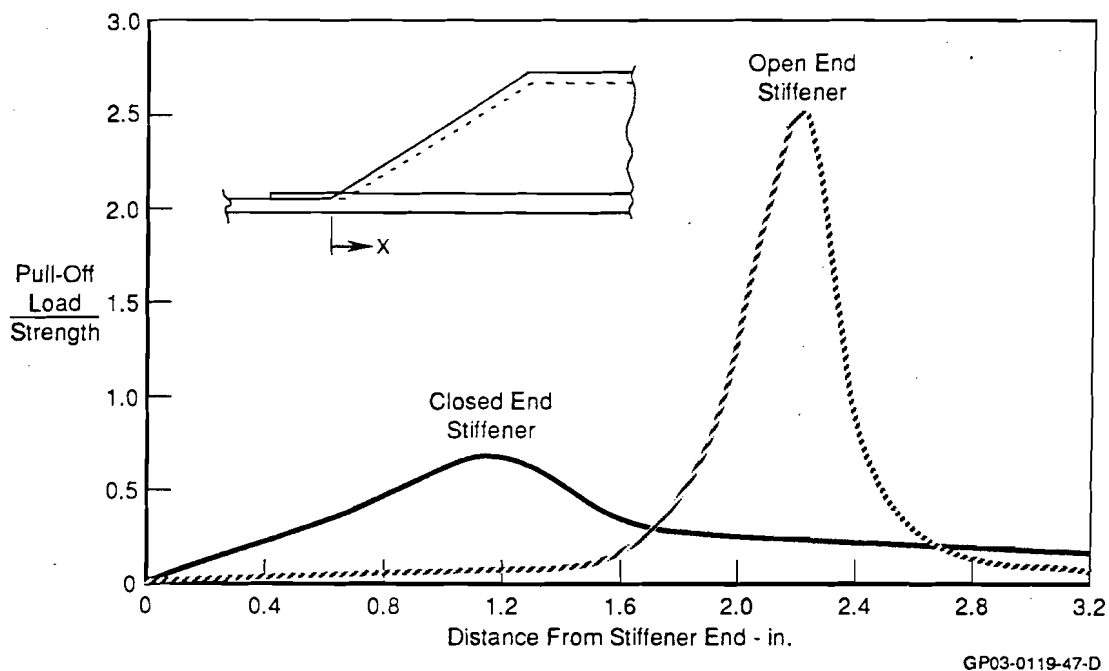
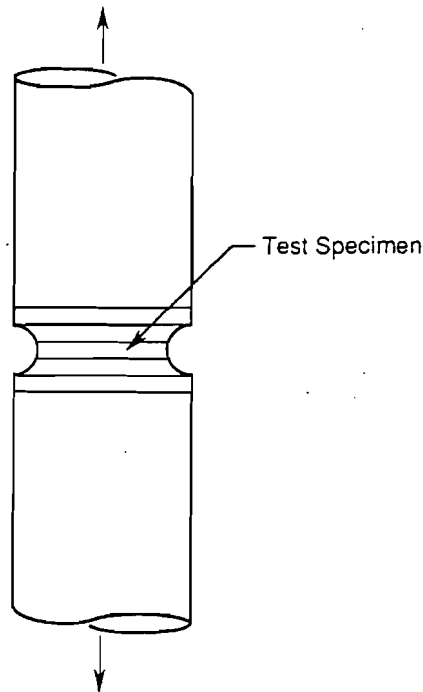


Figure 26. Pull-Off Loads Are Much Lower for Closed End Stiffener

3. Element Test Requirements

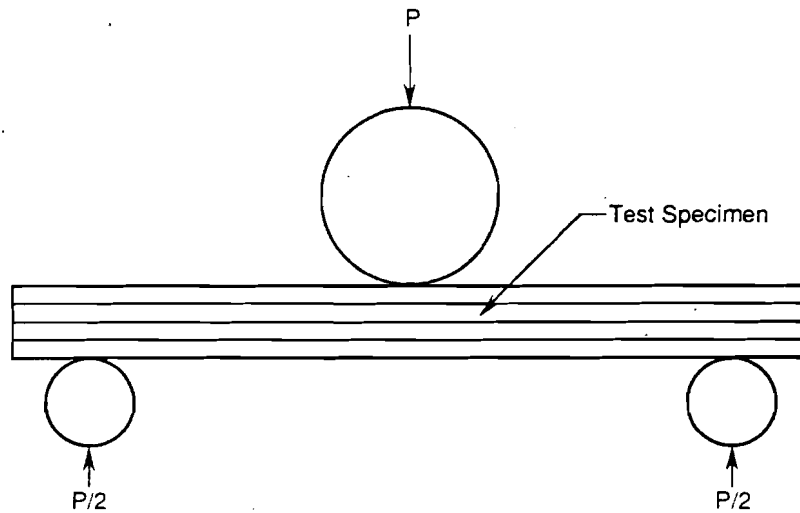
To accurately predict failure of out-of-plane loaded composite structures, coupon, element, and built-up specimens are required. This section describes some of the tests used in this report for analysis development and verification and some recommended tests that could provide better prediction capabilities for later analyses.

The purpose of coupon tests is to generate the allowables data required to predict the strength of structural components as described in previous sections. The primary parameters required in these analyses are the interlaminar tension and shear strengths. The coupon tests most often performed in the past to generate this data were flat, circular interlaminar tension specimens (Figure 27) and shear beam specimens (Figure 28). The round tensile specimens are bonded to metal rods, then pulled to failure in a tensile machine. Often the bondline fails, rather than the laminate. In such cases the specimen is machined to neck down to a smaller section to force the failure into the laminate. However, in these cases, the stress concentration provided by this notch is not well characterized. In addition, machining, bonding and machine alignment tolerances combine to produce significant scatter in the translaminar strength results.



GP03-0119-61-D

Figure 27. Interlaminar Tension Test Specimen



GP03-0119-48-D

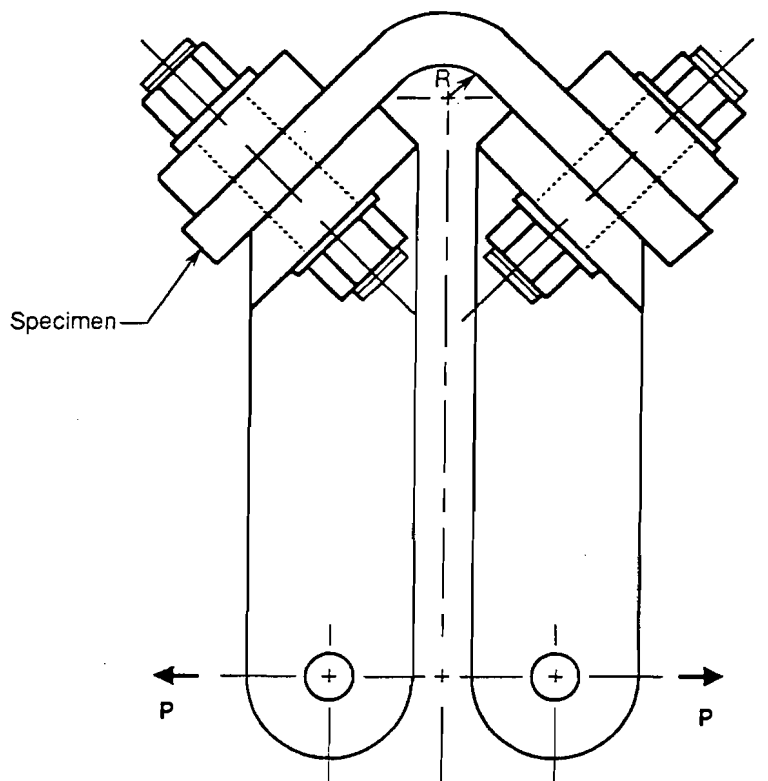
Figure 28. Interlaminar Shear Test Specimen

While the curved beam element specimen is far more difficult to fabricate, it is probably our preferred specimen for development of interlaminar tension strength data. The curved beam data must be interpreted using a routine like CLA but that analysis is relatively straightforward. The primary concern about using the curved beam specimen to determine interlaminar strengths is that the failure occurs in a curved section and not the flat portion of the laminate that one often wants to model. The shear strength data used in this program was generated using the standard ASTM shear beam specimen. One drawback to this specimen is that it permits only a strength assessment. WEBSTER requires stiffness data in both translaminar shear and tension. A good specimen for interlaminar tension modulus is not known today. For the analyses performed in this program transverse moduli were used to estimate the interlaminar moduli but the accuracy of this estimation is not known.

The purpose of element tests is to validate the out-of-plane analysis methodology and to provide data for the determination of strength allowables. Two small elements and one small subcomponent are recommended for this purpose, namely

- curved laminate specimens
- stiffener pull off tests
- stiffener runout specimen tests.

The curved laminate test specimen and fixture are shown schematically in Figure 29. This test specimen is similar to those analyzed in the current program. Excellent correlations have been achieved, as discussed in



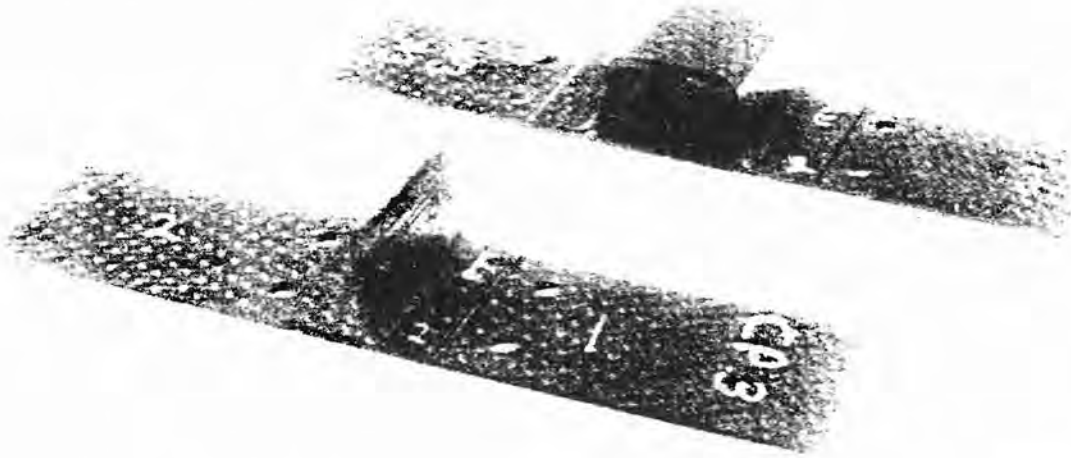
GP03-0119-29-D

Figure 29. Schematic of Curved Laminate Test Configuration

Section IV. This test is recommended for aircraft design development programs whenever curved laminates are used as primary load paths. The laminate layup, thickness, and radius at the corner of the specimen should be designed to satisfy the range of design applications of the specific structure. The test environment should simulate the service environment of the actual structure. A minimum of ten specimens should be tested for each specimen configuration and environment.

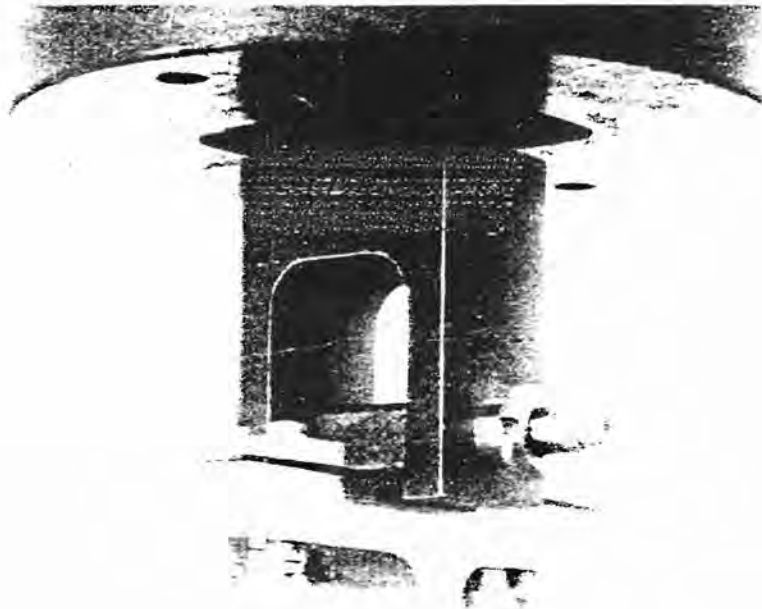
While examples relating element test results to **full scale panel** separation results appear successful based on the results presented in Volume I of this report, there is still some concern over the extrapolation of such results. For instance, hat stiffeners in pull off tests are subject to a considerable moment beneath the web of the hat, in addition to a pull off load. Yet this same loading condition does not occur in postbuckled panel tests. The moment is induced by skin deflections. However, postbuckling tests of panels show that there is far less skin panel deflection than is generated in the element tests.

Stiffener pull off specimens, shown in Figure 30, are designed to verify skin/stiffener separation strength. This specimen and its loading (Figure 31) are similar to those analyzed in the course of this program. However, a different loading condition, shown in Figure 32, is recommended for future tests. In the figure, the pull off



GP03-0119-38

Figure 30. MCAIR Cocured Joint Element Specimens



GP03-0119-39

Figure 31. Direct Out-of-Plane Loading of Hat Stiffened Element Specimen

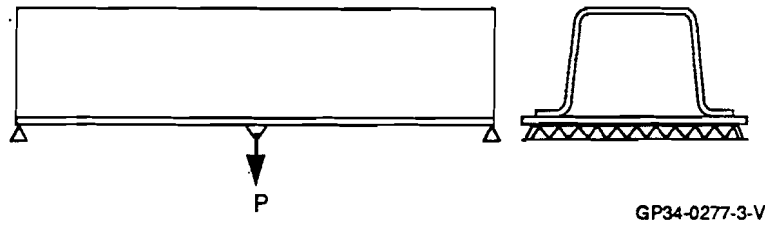
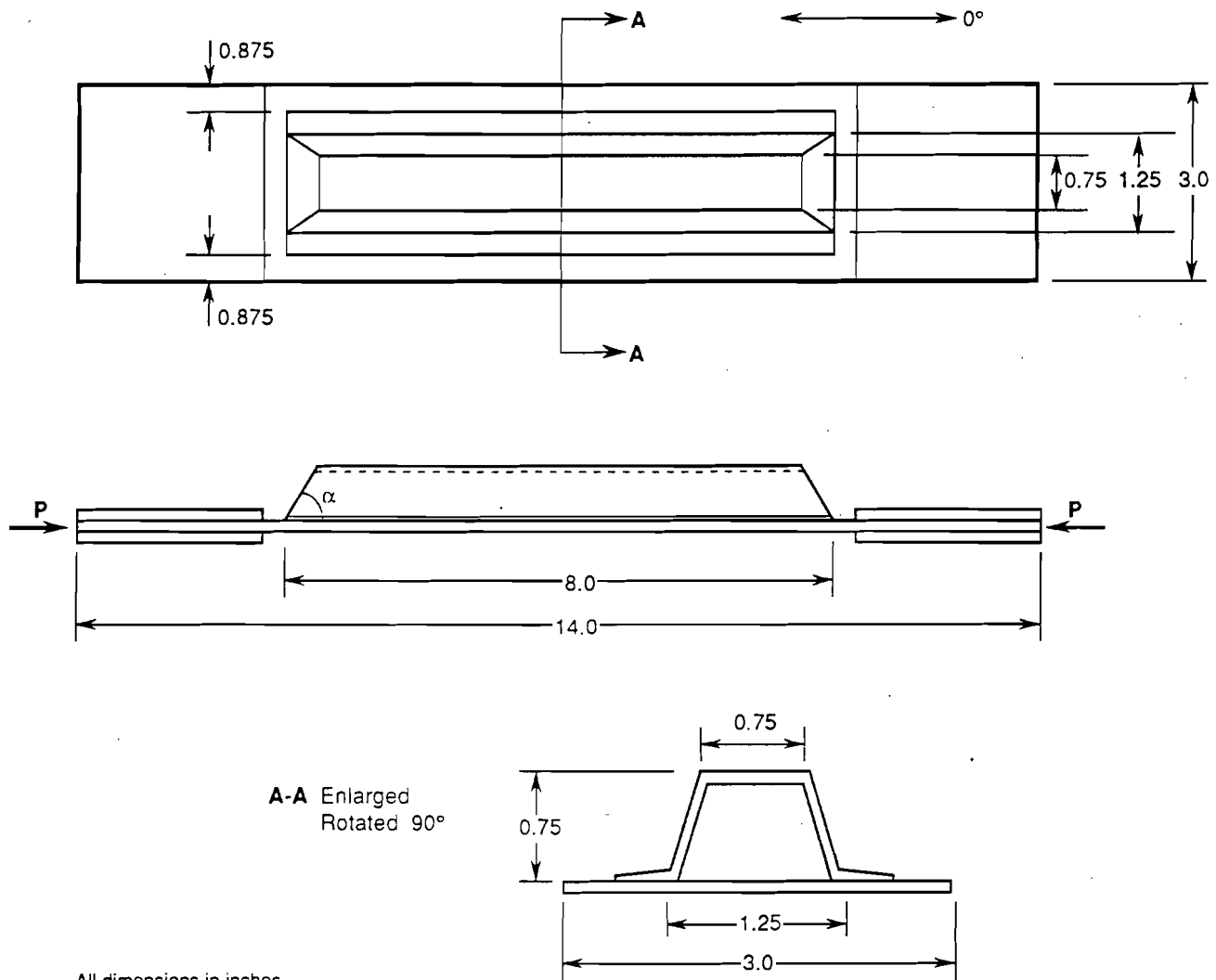


Figure 32. Recommended Skin/Stiffener Pull Off Specimen

strength is determined using a failure mode induced by the buckle shape of the skin, rather than direct pull off as performed previously in Figure 31. The length and loading conditions are selected to match the skin buckle shape of the skin as closely as possible. The previous tests involve considerable transverse skin bending that applies a large moment to the skin/stiffener joint that is not always experienced in actual postbuckled panels.

The stiffener runout specimen is shown schematically in Figure 33. The purpose of these tests is to evaluate designs that attempt to reduce interfacial stresses and to avoid out-of-plane failures. This test is used to verify the design and WEBSTER predictions. Tests should include a sufficient range of termination angles to optimize the structural design. Design features being considered for application to structure must be examined in the design development testing. One such option is to place "chicken rivets" (mechanical fasteners that provide a redundant load path for cocured or bonded joints) around the termination region such that an out-of-plane failure can be arrested. Other options include scarfing (tapering the stiffener termination) and increasing the skin thickness near the termination site. These must be included in the test article.



GP03-0119-31-D

Figure 33. Schematic of Stiffener Runout Test Specimen

4. Allowables Determination

Out-of-plane failures in composite structures are generally tension, shear, or mixed tension and shear failures (see Section IV). Allowables are therefore needed for interlaminar tension and shear stresses for out-of-plane designs. For mixed mode failure analysis, a quadratic criterion can be used with a reasonable degree of accuracy.

Interlaminar tension allowables can be determined from flatwise tension or curved beam tests. Interlaminar shear allowables can be determined from shear beam or other standard shear tests.

Composites exhibit more data scatter than metals. Based on the limited test data, out-of-plane properties are likely to have more data scatter than in-plane properties. This is demonstrated in the comparison of knock down factors for allowables for out-of-plane loadings versus in-plane loadings shown in Figure 34. Because of this

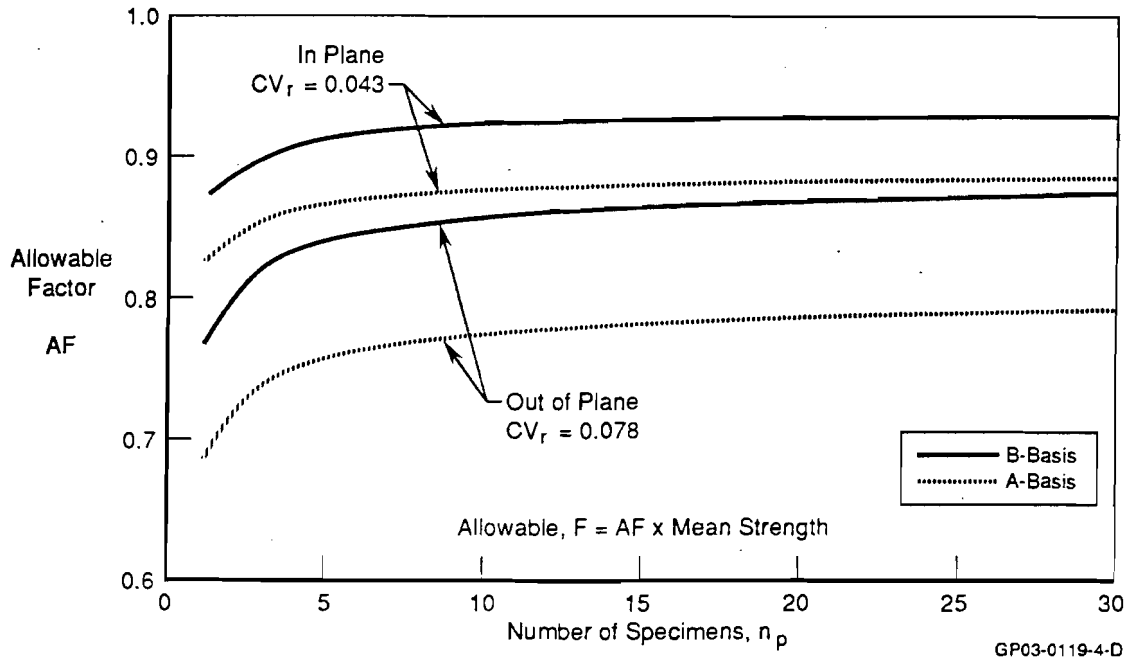


Figure 34. The Additional Scatter in Out-of-Plane Strengths Is Reflected in a Reduced Factor on Allowables

increased data scatter, an increase in the number of tests required to determine the B-basis design allowables from five to ten is recommended.

Out-of-plane properties are sensitive to environmental conditions. Design allowables should therefore be determined for the entire range of the environmental service envelope of the structure. Statistical methods must be used to compute the design allowables as shown in References 17-20.

In addition to environmental sensitivity, out-of-plane strengths can be significantly reduced by fatigue damage. Tests of flatwise tension coupons resulted in a rapid decrease in load carrying capability even at very short lives (Figure 35). On the other hand, fatigue limit stresses (those below which no failures occur for extremely large numbers of cycles) of up to half of the ultimate interlaminar tension strength have been found in test. Similarly, significant strength reductions have been found after fatigue loading of hat stiffened panels (Figure 36). Obviously, one must consider repeated loads in developing allowables for out-of-plane loaded structures subjected to repeated loads.

The current database of out-of-plane material properties is not as extensive as that of in-plane material properties. In-plane material properties (E_x , E_y , G_{xy} , ν_{xy} , t) of a material system are usually available, and will not be discussed in great detail. Other material properties (E_z , G_{xz} , G_{yz} , ν_{xz} , ν_{yz} , a_o) and interlaminar normal and shear strengths will be highlighted here.

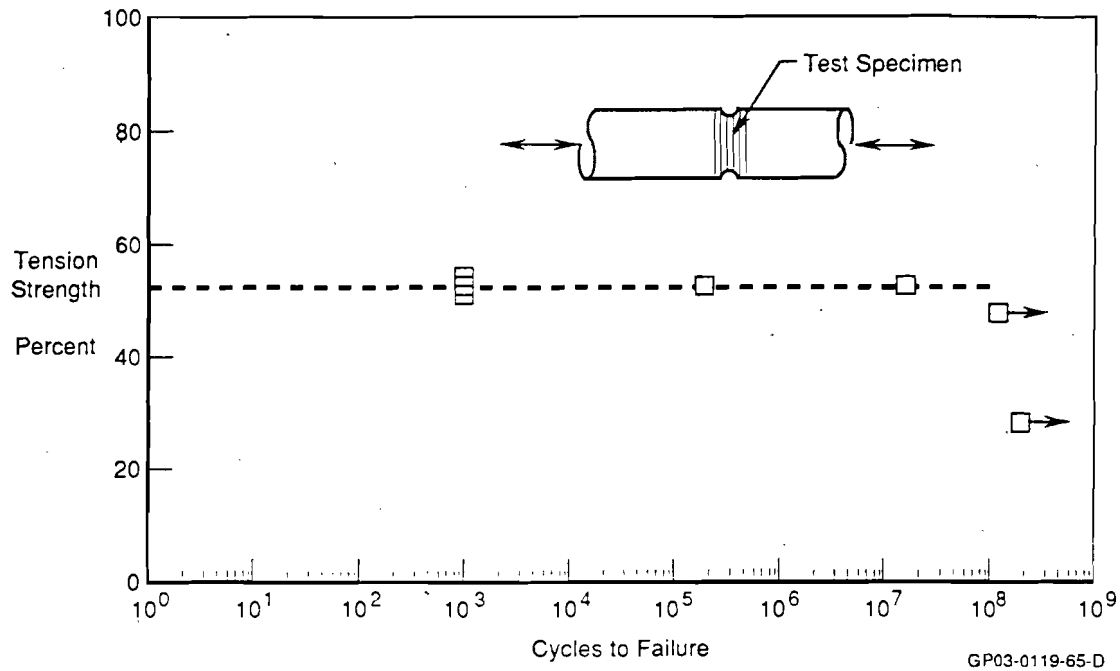


Figure 35. Translaminar Fatigue Test Results

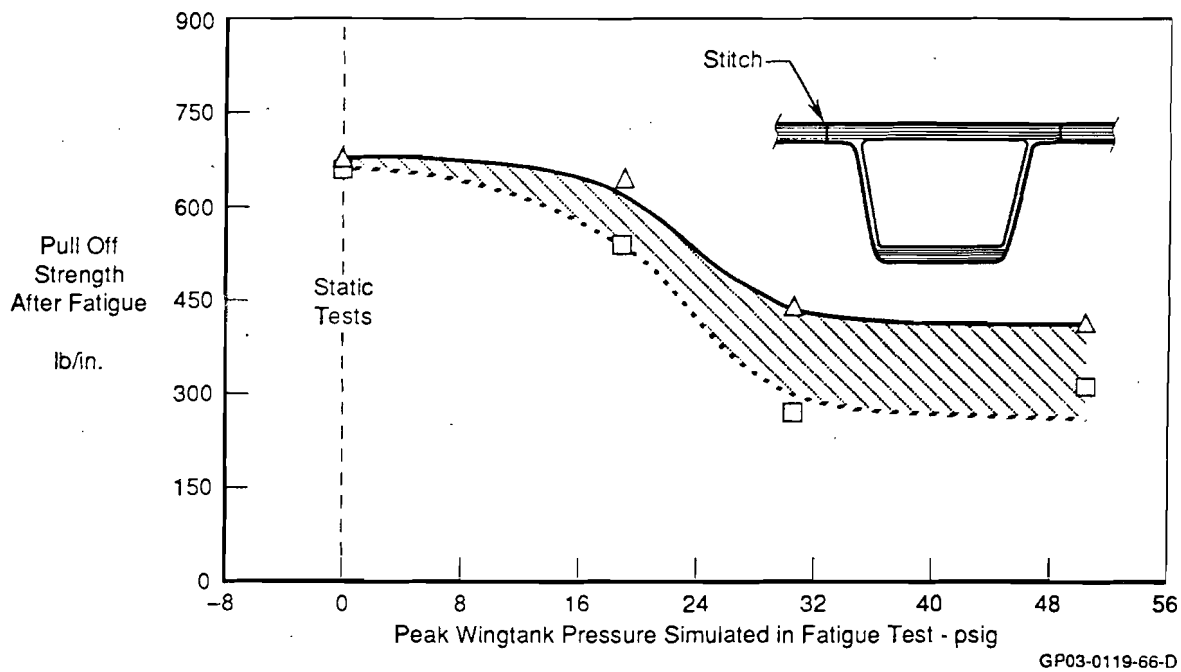


Figure 36. Effect of Fatigue on Pull-Off Strength of Hat Stiffener Panels

Out-of-plane material properties are especially needed for analyses such as WEBSTER and BEAM, including the attachment properties (E_z , G_{xz} , G_{yz} , ν_{xz} and ν_{yz}). Ideally, the exact values need to be known, though this is not always possible. Parametric studies of these variables using WEBSTER show that the resultant stress state is not heavily dependent on these properties. For example, if the value of E_z is altered by 20 percent, the maximum

interlaminar stresses in the model will not vary more than eight percent. If ν_{xz} and ν_{yz} increase (or decrease) by 50 percent, the maximum stress state varies by less than eight percent. Greater variations in these values alter the stress predictions more severely. Altering in-plane properties has a more drastic effect on the maximum stress. Varying E_x or E_y by 20 percent changes the stress state significantly.

The WEBSTER and BEAM analyses predict failure by averaging the stress over a characteristic length from the geometric discontinuity. This length has been well defined for geometric discontinuities such as circular cutouts. The definition of this characteristic length for stiffener terminations is the same as for circular cutouts (approximately 0.02 inches). Similar values were therefore used to validate each analysis. Although the geometric discontinuities are different, the characteristic lengths remained approximately equal. Characteristic dimensions can be decreased slightly to insure a conservative failure prediction in all cases.

5. Preferred Design Techniques

Cocured/adhesively bonded structures are anticipated to be used extensively in future aircraft to more fully utilize the potential benefits of composite materials. One of the key concerns in the application of these structures is out-of-plane failures. The potential for out-of-plane failures can be minimized through proper design, analysis, and test verification. The following design techniques are recommended to further avoid this type of failure.

1. Avoid sudden changes of load path within a structure.
2. Avoid sudden changes in stiffness. Design stiffener runouts so that bending stiffness is continuously runout into the skin, or optimize the termination angle at the runout to minimize the out-of-plane stress build-up.
3. For cases where out-of-plane loads cannot be reduced to acceptable levels, an arrestment mechanism for interfacial failures (delaminations) should be provided, such as fasteners at the end of stiffener runouts or stitching in stiffener flanges.
4. Closely follow ply drop off procedures in thickness variations
5. Avoid sharp corners ($R/t < 2$) to reduce interlaminar stresses. Place flange attachment fasteners more than two radii away from out-of-plane attachments to minimize shear stresses within the flange bondline.
6. Avoid large differences in attachment stiffnesses in joints

7. Reduce peel stresses by adding soft plies or additional adhesive near discontinuities
8. Use the softest possible flange stiffness to maximize pull-off strength in integrally stiffened panels
9. Use environmentally conditioned specimens to determine interlaminar strengths for design. In cases in which repeated out-of-plane loads are known to occur use fatigue cycled specimens to account for strength reductions found to occur after such cycling.

SECTION III

CURVED LAMINATE ANALYSIS (CLA) USER'S MANUAL

1. Introduction

1.1 Function – The Curved Laminate Analysis (CLA) was created as a tool for predicting the stress state and strength of curved orthotropic load carrying structures. This analysis is useful for assessing the strength of designs, and for conducting trade studies. The CLA program incorporates user– friendly features, and requires minimal input information (geometry, layup, material properties, and loads). The stress values are rapidly determined since CLA is a closed-form analysis. A Tsai-Hill (quadratic) failure criterion is used to predict strength by comparing the stress state to the given allowables. The program output includes an echo of the input information, the predicted failure loads, and the interlaminar stress state of the laminate.

The Curved Laminate Analysis is a strength of materials derivation and is based on equilibrium concepts. It accounts for the effects of applied axial, shear, and pressure loads, in addition to pure bending, unlike previous curved laminate derivations. The stress state can be determined at any radial cut through the laminate. A ply by ply inspection along each of the user specified radial cuts provides a failure prediction for the laminate.

1.2 Limitations – CLA assumes that the curved laminate exhibits linear elastic behavior. Plane sections are assumed to remain plane under bending and axial loads. This assumption is valid as long the ratio of the arc length of the curved laminate ($r\theta$, where r is inside radius and θ is the angle subtended by the laminate) to the laminate thickness (t) is not too small. This ratio ($r\theta/t$) should be greater than one for best results. Analytical results may be suspect in areas where the laminate changes from curved to straight, since the inside radius increases rapidly to infinity. Validity cannot be assured for geometries where the inside radius is grossly different from the laminate thickness.

The CLA method assumes the failure mode to be ply delamination resulting from interlaminar stresses. Other failure modes are not considered. The accuracy of Tsai–Hill failure predictions is dependent upon the values of the interlaminar tension and shear allowables. These allowables are not always readily available, and often need to be estimated.

2. Input Description

The input information is entered sequentially into the CLA program. The user is prompted to supply each item of information. User friendly features have been included to minimize the risk of "crashing" the program with incorrect input. The program inputs are described below.

1. Problem Description – Requests a problem title. A maximum of 60 alpha-numeric characters is accepted.

2. Material Properties – The lamina properties of up to three different materials are requested. The necessary properties for each material are:

- E_1 , modulus in fibre direction (msi)
- E_2 , modulus in transverse direction (msi)
- G_{12} , shear modulus of ply (msi)
- ν_{12} , Poisson's ratio of ply
- t , thickness (in.)

Note: 1.0 msi = 10^6 psi

3. Laminate Allowables – Requests the translaminar stress allowables for the laminate.

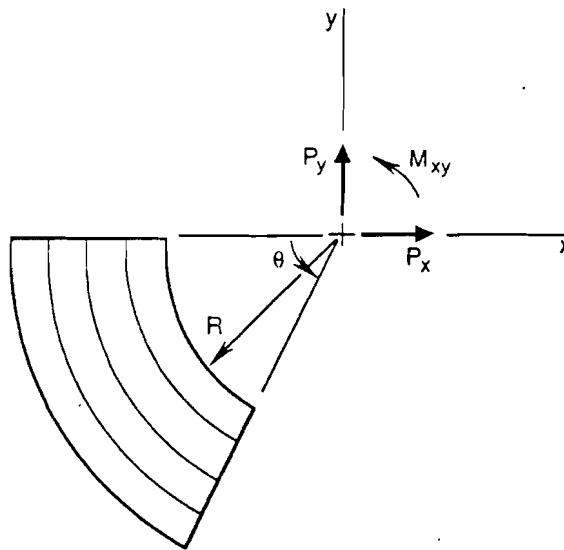
- Flatwise tension (ksi)
- Interlaminar shear (ksi)

4. Laminate Layup – Used to input the orientation (in degrees) and material identifier for a maximum of 100 plies. The user chooses from three layup types.

- Unsymmetric, $[a_1, a_2, \dots, a_n]$ $n < 100$
- Odd symmetric, $[a_1, a_2, a_3, \dots, \bar{a}_n]_s$ $n < 50$
- Even symmetric, $[a_1, a_2, a_3, \dots, a_n]_{ms}$ $m \times n < 50$

5. Angle Geometry – Requests the values that describe the curved laminate and the locations of stress inspection (Figure 37).

- Laminate width (in.)
- R_i , inside radius (in.)
- θ_{tot} , angle subtended by the laminate (degrees)
- θ_i , initial angle of inspection (degrees)
- θ_f , final angle of inspection (degrees)
- Number of inspections



GP03-0119-15-D

Figure 37. Curved Laminate Analysis Geometry

6. Applied Loads – The input section for entering the loads that are applied to the laminate. The loads must be resolved at the center of curvature. Refer to Figure 37 for the definition of the loads. The ratio of the loads is maintained in the failure predictions.

- P_x (lbs)
- P_y (lbs)
- M_{xy} (in-lbs)
- p_o (psi)

7. Output Options – Allows the user to request whether or not to use an output file. An option is given for either the full output or an abbreviated output.

- Output files (yes/no)
- Output file type (full/truncated)
- Output file name

3. Output Description

CLA provides the output of the program to the user on the terminal screen. The input data is given first to remind the user what problem parameters were chosen. The critical stress location is identified next. Both the angle and ply interface number of the most severe stress state are provided. The predicted failure loads are also provided. This prediction is based on the given loads, which are proportioned by the reciprocal of the failure index. The percentage of the failure due to each stress component is also given.

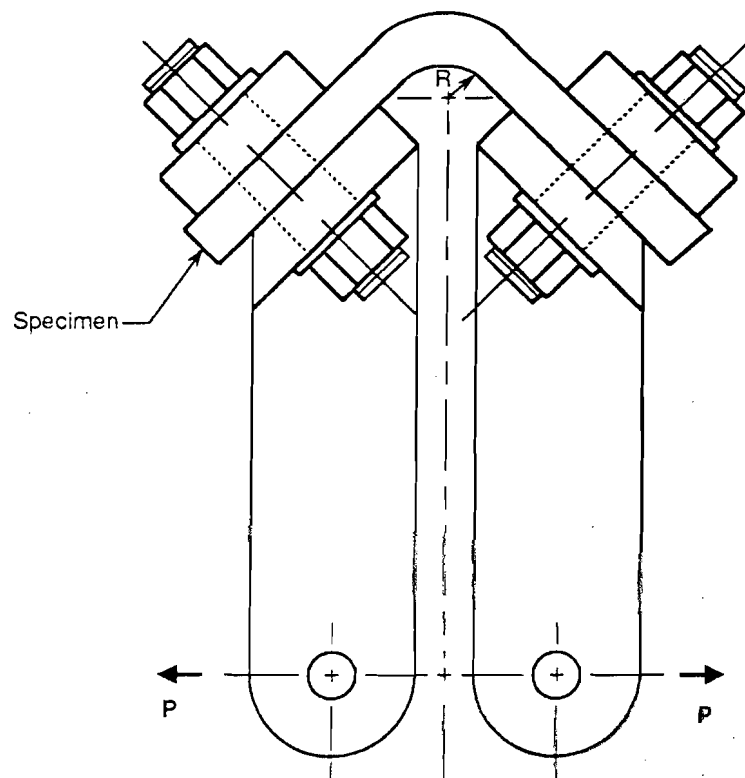
The information printed on the terminal screen is also sent to an output file, when requested. This is the only information provided when the truncated output is specified. The full output also includes this information, as well as the stress distributions and failure prediction at each inspection angle. Examples of each of the output options are given in the example problems section of this User's Manual.

4. Verification

The verification of CLA was done by comparing CLA strength predictions to actual test data. MCAIR and Northrop conducted a joint IRAD test program of curved laminates fabricated from a carbon/bismaleimide material system. The effects of varying the laminate thickness, inside radius, and test environment were evaluated using the configuration shown in Figure 38.

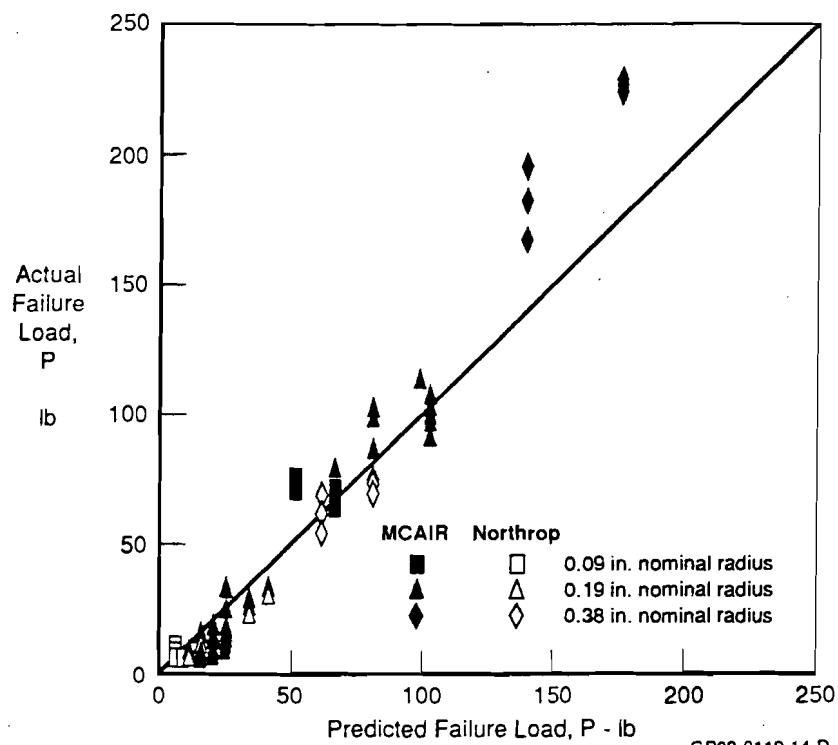
A good correlation was achieved, despite significant scatter in the test data. The comparison of the predicted and test failure loads is summarized in Figure 39.

Parametric studies were performed to verify the analysis can correctly predict some general trends. One of these investigated the effects of changing the layup and the inside radius of the curved section. The assumed laminate of this study consisted of ten carbon/epoxy tape plies, subjected to a 50 in-lb/in bending moment. Three laminates of increasing bending stiffness were analyzed, with the inside radius varied. The results are



GP03-0119-16-D

Figure 38. MCAIR/Northrop Angle Specimen Test Configuration



GP03-0119-14-D

Figure 39. Curved Laminate Analysis Correlation

depicted graphically in Figure 40. The laminate with the highest bending stiffness ($[0/0/0/90/90]_s$) exhibited the lowest radial stress. This is explained by the fact that a curved laminate of infinite stiffness will not bend, and therefore will not experience any interlaminar stresses. This study also revealed that an increase of the inside radius will decrease the radial stress. This agrees with theory, since a laminate of infinite radius (a straight laminate) will exhibit no radial stresses.

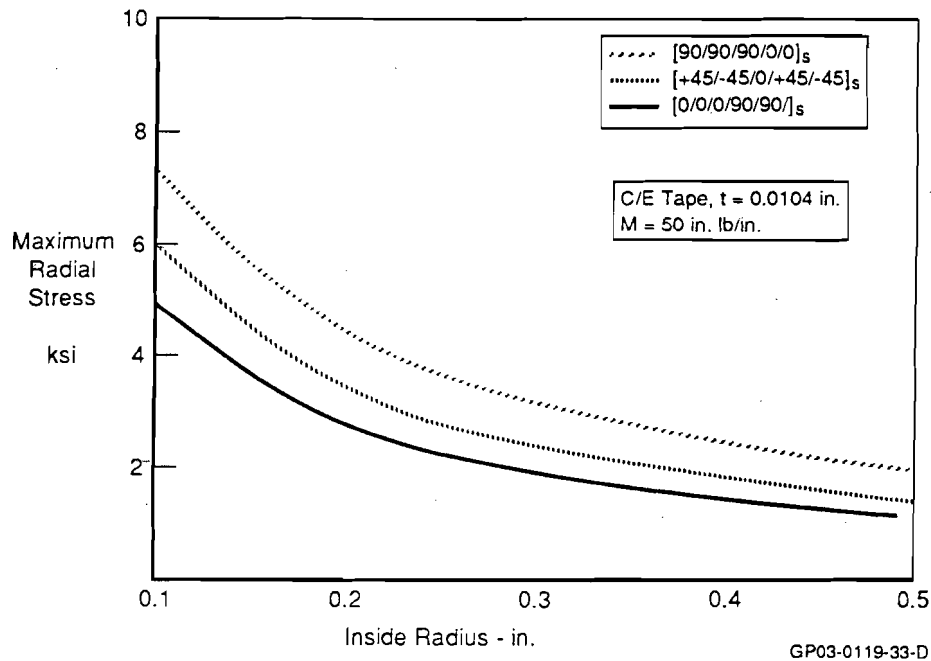
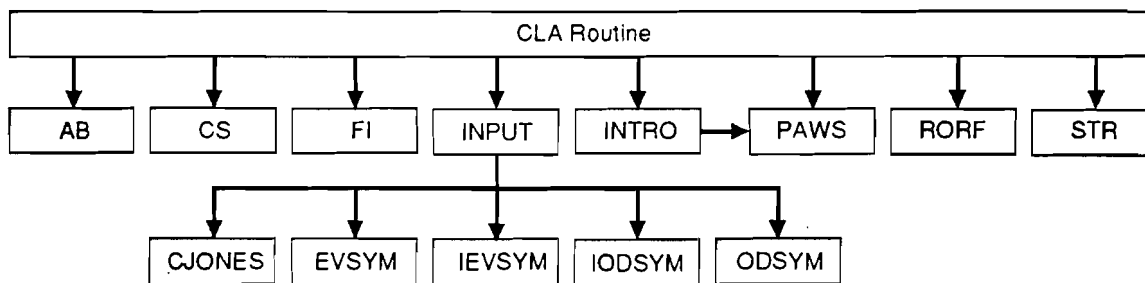


Figure 40. Effect of Layup and Inside Radius on the Maximum Radial Stress (as Predicted by CLA)

5. Subroutine Functions

Figure 41 provides the flowchart of the CLA FORTRAN computer program. The main program (CLA routine) performs several functions. It is responsible for calling each of the individual subroutines and performing



GP03-0119-32-D

Figure 41. CLA Flowchart

the necessary bookkeeping associated with the analysis. It also requests what type of an output file is wanted (if any). The output information is formatted and printed by the main program.

A brief description of each subroutine's functions follows.

AB: This routine produces the differential load quantities which are used later by STR.

CJONES: This function routine determines the x-direction elastic modulus for each ply of the laminate. The required input includes the orientation and material properties (E_1 , E_2 , G_{12} , ν_{12}) of the ply.

CS: Two constants that are later used by STR are produced by this subroutine. These constants are dependent upon the loads and geometry.

EVSYM: The ply orientation and thickness arrays for an even symmetric laminate are set up by this routine. This routine was created to minimize the amount of user input data required.

FI: A Tsai-Hill quadratic failure function is used in this function routine to determine the failure index (f_i) for a given stress state. The radial and shear stress values are each divided by their respective allowables, squared, and summed together. The square root of this value is f_i .

IEVSYM: This routine sets up the ply material identification array for an even symmetric laminate. This routine is similar to EVSYM, though it manipulates an integer array rather than real number arrays.

INPUT: The data for each run of the program is requested by this routine. A more complete description of the required input data was given previously in this User's Manual.

INTRO: The introductory information for CLA is printed to the computer terminal by this routine. The program title, the legal disclaimers, and a brief description of the analysis are presented within this introduction.

IODSYM: The ply material identification array for an odd symmetric laminate is set up by this routine. This routine was created to minimize the amount of user input data required.

ODSYM: This routine sets up the ply orientation and thickness arrays (real number arrays) for an odd symmetric laminate. This routine is similar to IODSYM, though it manipulates real number arrays rather than an integer array.

PAWS: This routine provides a pause for the user to view the data on the screen. A carriage return allows the program to proceed.

RORF: The radii for both the neutral axis and the centroid of the curved laminate are determined by this routine. This information is used in the stress calculations, and is given as part of the output.

STR: This routine is the core of CLA. It determines the stress components at the location of interest.

6. Example Problems

This section provides two example problems to illustrate the use of the CLA program. The first example is a typical test specimen that was used to correlate the analysis and test data. The second example is an "L-bracket" that is subjected to combined pressure and tension loads.

6.1 Problem #1

An angle specimen is tested in the configuration shown in Figure 42. The layup, material properties, and specimen geometry are given below. The goal is to predict the failure load of this specimen.

Laminate Layup: $[45_c/0/+45/-45/90/45_c]_{2S}$

Material Properties: AS4/3501-6 (cloth)

$$E_1 = 10.4 \text{ msi}$$

$$E_2 = 10.4 \text{ msi}$$

$$G_{12} = 1.0 \text{ msi}$$

$$\nu_{12} = 0.06$$

$$t_{\text{ply}} = 0.0140 \text{ in.}$$

Material Properties: AS4/3501-6 (tape)

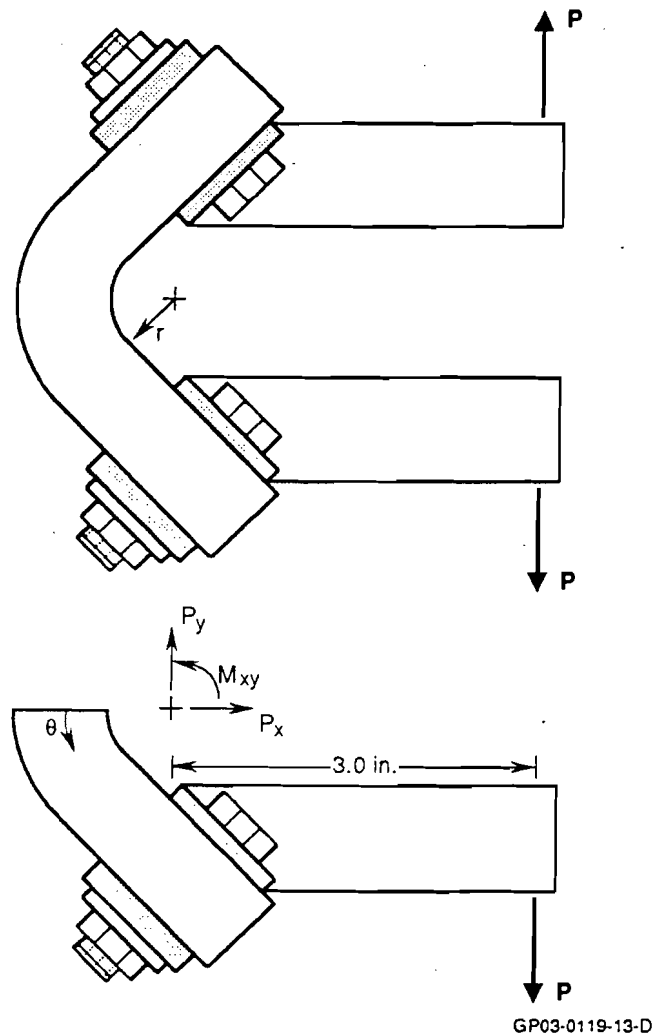
$$E_1 = 18.5 \text{ msi}$$

$$E_2 = 1.90 \text{ msi}$$

$$G_{12} = 0.85 \text{ msi}$$

$$\nu_{12} = 0.30$$

$$t_{\text{ply}} = 0.0052 \text{ in.}$$



GP03-0119-13-D

Figure 42. Angle Specimen and the Resolution of the Applied Loads to the Center of Curvature

Material Allowables: AS4/3501-6

$$\sigma_{fwt} = 3.00 \text{ ksi}$$

$$\tau_{ils} = 11.0 \text{ ksi}$$

Specimen Geometry:

$$\text{width} = 2.0 \text{ in.}$$

$$\text{inside radius} = 0.25 \text{ in.}$$

Solution:

The first step for predicting the failure load is to resolve the applied load (P) at the center of curvature such that the proportions of the load components (P_x , P_y , M_{xy}) can be determined as function of applied loads. This can be done in the following way.

$$P_x = 0$$

$$P_y = P$$

$$M_{xy} = 3.0 \times P$$

A unit load ($P = 1.0$ lb) can be assumed for predicting failures, since CLA is a linear analysis and requires only the proportions of the loads for failure predictions. The analysis needs to be performed over only half the curved section due to symmetry, as shown in Figure 42. The inspections are chosen to be performed at representative angles ($\theta = 0^\circ, 15^\circ, 30^\circ$, and 45°).

This gives all the necessary information for running CLA. The information from the interactive session is given on the following pages.

The failure load can be found directly from the output data. Failure is predicted at $P_y = 77.4$ lb, which corresponds to $P = 77.4$ lb.

CLA Example Problem #1 Input:

```

*****
CURVED LAMINATE ANALYSIS
VERSION 1.0

DEVELOPED BY MCAIR

ANALYSIS DEVELOPMENT: JOHN M. BUBB
                      PHILIP C. PAUL
*****
HIT <RETURN> TO CONTINUE
*****
THIS FORTRAN COMPUTER PROGRAM IS THE AUTOMATED
VERSION OF THE CURVED LAMINATE ANALYSIS METHOD.
THE METHOD IS BASED ON THE CURVED STIFFENER
ANALYSIS, CREATED BY JOHN BUBB. CLA DETERMINES
THE PLY BY PLY STRESS DISTRIBUTION AT ANY ANGULAR
POSITION IN THE CURVED LAMINATE. ORTHOTROPIC
MATERIALS, ARBITRARY LAYUPS, AND A GENERAL
LOADING ARE ALL PERMITTED BY CLA. THIS PROGRAM
WAS DEVELOPED UNDER THE NAVY CRAD "OUT-OF-PLANE
ANALYSIS FOR COMPOSITE STRUCTURES" (CONTRACT NO.
N62269-87-C-0226). FOR FURTHER INFORMATION, SEE
THE FINAL REPORT, OR CONTACT PHILIP C. PAUL.
*****
HIT <RETURN> TO CONTINUE

```

ENTER THE PROBLEM DESCRIPTION (60 CHARACTERS MAX.):

Example Problem #1

ENTER THE NUMBER OF MATERIALS (3 MAX.)

2

ENTER THE PROPERTIES FOR MATERIAL #1

- E1, MODULUS IN FIBRE DIRECTION (MSI)
- E2, MODULUS IN TRANSVERSE DIRECTION (MSI)
- G12, SHEAR MODULUS OF PLY (MSI)
- NU, POISSON RATIO OF PLY
- T, THICKNESS OF THE PLY

10.4 10.4 1.0 0.06 0.014

ENTER THE PROPERTIES FOR MATERIAL #2

- E1, MODULUS IN FIBRE DIRECTION (MSI)
- E2, MODULUS IN TRANSVERSE DIRECTION (MSI)
- G12, SHEAR MODULUS OF PLY (MSI)
- NU, POISSON RATIO OF PLY
- T, THICKNESS OF THE PLY

18.5 1.90 0.85 0.30 0.0052

ENTER THE LAMINATE ALLOWABLES.

- FLATWISE TENSION (KSI)
- INTERLAMINAR SHEAR (KSI)

3.0 11.0

ENTER THE TYPE OF LAYUP.

1: UNSYMMETRIC [A1,A2,...,AN]

2: ODD SYMMETRIC [A1,A2,A3,...,AN]S

3: EVEN SYMMETRIC [A1,A2,A3,...,AN]MS

3

EVEN SYMMETRIC LAYUP [A1,A2,A3,...,AN]MS

ENTER THE NUMBER OF ORIENTATIONS (N) AND THE
NUMBER OF SERIES (M) (M X N <= 50)

6 2

ENTERING A BACKSLASH (/) WITHIN THE FOLLOWING DATA
RECORDS ASSUMES THE REMAINING ITEMS TAKE THE VALUES
FROM THE PREVIOUS DATA RECORD.

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 1

45.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 2
0.0 2

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 3
45.0 /

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 4
-45.0 /

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 5
90.0 /

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 6
45.0 1

ENTER THE LAMINATE WIDTH (IN.)
2.0

ENTER THE INSIDE RADIUS OF THE CURVED LAMINATE (IN.)
0.25

ENTER THE NECESSARY ANGLES (DEGREES)
- THETA TOTAL: ANGLE SUBTENDED BY THE CURVED LAMINATE
- THETA INIT.: INITIAL ANGLE OF INSPECTION
- THETA FINAL: FINAL ANGLE OF INSPECTION
- INSP: NUMBER OF INSPECTIONS

(FOR ONLY ONE ANGLE OF INSPECTION AT THETA INITIAL,
ENTER INSP = 1)
45.0 0.0 45.0 4

ENTER THE LOADS RESOLVED AT THE CENTER OF CURVATURE.
- PX (LBS)
- PY (LBS)
- MXY (IN-LBS)
0.0 1.0 3.0

ENTER THE PRESSURE DIFFERENCE (PSI)
(POSITIVE VALUE FOR OUTWARD ACTING PRESSURE.)
0.0

DO YOU WANT OUTPUT FILES (Y/N)?
Y

ENTER THE OUTPUT FILE NAME (FILENAME.OUT) 9 CHARACTERS MAX.
EXAMPLE1

WHAT TYPE OF SCREEN OUTPUT DO YOU WANT?

- 1) FULL: STRESS STATE AT EVERY ANGLE
- 2) TRUNCATED: STRESS STATE ONLY AT FAILURE LOCATION

2

CLA Example Problem #1 Output:

**** CURVED LAMINATE ANALYSIS ****

INPUT DATA

Example Problem #1

PROPERTIES FOR MATERIAL #1:

E1 = 1.0400E+07 PSI
E2 = 1.0400E+07 PSI
G12 = 1.0000E+06 PSI
NU = .06
T = 1.4000E-02 IN.

HIT <RETURN> TO CONTINUE

PROPERTIES FOR MATERIAL #2:

E1 = 1.8500E+07 PSI
E2 = 1.9000E+06 PSI
G12 = 8.5000E+05 PSI
NU = .30
T = 5.2000E-03 IN.

LAMINATE ALLOWABLES:

- FLATWISE TENSION = 3.0000E+03 PSI
- INTERLAMINAR SHEAR = 1.1000E+04 PSI

HIT <RETURN> TO CONTINUE

PLY ORIENTATION MATERIAL PLY ORIENTATION MATERIAL

1	45.0	1	13	45.0	1
2	.0	2	14	90.0	2
3	45.0	2	15	-45.0	2
4	-45.0	2	16	45.0	2
5	90.0	2	17	.0	2
6	45.0	1	18	45.0	1
7	45.0	1	19	45.0	1
8	.0	2	20	90.0	2
9	45.0	2	21	-45.0	2
10	-45.0	2	22	45.0	2
11	90.0	2	23	.0	2
12	45.0	1	24	45.0	1

HIT <RETURN> TO CONTINUE

THE LAMINATE WIDTH: 2.000
 THE LAMINATE INNER RADIUS: .250
 THE RADIUS OF THE NEUTRAL AXIS: .338
 THE RADIUS OF THE CENTROID: .348

ANGLE INFORMATION:

- THETA TOTAL: 45.0
 - THETA INITIAL: .0
 - THETA FINAL: 45.0
 - INSPECTIONS: 4

APPLIED LOADS:

- PX: 0.0000E+00
 - PY: 1.0000E+00
 - MXY: 3.0000E+00
 - PO: 0.0000E+00

HIT <RETURN> TO CONTINUE

CRITICAL STRESS STATE AT THETA = .0

THE MAXIMUM FAILURE INDEX IS: .01

PREDICTED FAILURE INFORMATION

- PX: 0.0000E+00
- PY: 7.7423E+01
- MXY: 2.3227E+02
- PRESSURE: 0.0000E+00
- RADIAL STRESS (PSI): 3.000E+03
- SHEAR STRESS (PSI): 0.000E+00
- INTERFACE OF FAILURE: 8

CONTRIBUTIONS TO FAILURE

- RADIAL STRESS: 100.0%
- SHEAR STRESS: .0%

HIT <RETURN> TO CONTINUE

6.2 Problem #2

The "L-bracket" shown in Figure 43 is loaded by combined pressure and tension loads. The layup, material properties, structural geometry, and loads are given below. The objective is to determine the margin of safety of this structure.

Laminate Layup: $[45/0/45/0/45/\bar{0}]_s$

Material Properties: AS4/3501-6 (cloth)

$$E_1 = 10.4 \text{ msi}$$

$$E_2 = 10.4 \text{ msi}$$

$$G_{12} = 1.0 \text{ msi}$$

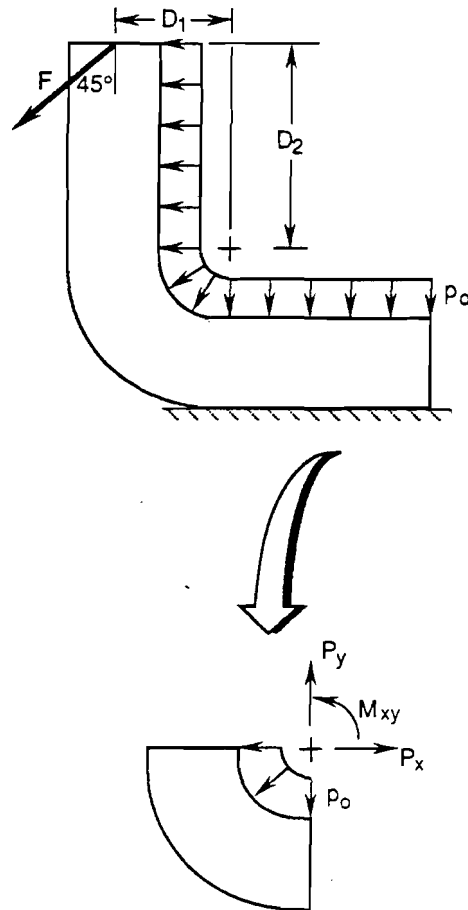
$$\nu_{12} = 0.06$$

$$t_{\text{ply}} = 0.0140 \text{ in.}$$

Material Allowables: AS4/3501-6

$$\sigma_{\text{fwt}} = 3.00 \text{ ksi}$$

$$\tau_{\text{lls}} = 11.0 \text{ ksi}$$



GP03-0119-12-D

Figure 43. L-Bracket Subjected to Combined Pressure and Tension Loads**Structural Geometry:**

- width = 1.0 in.
- inside radius = 0.20 in.
- $D_1 = 0.27$ in.
- $D_2 = 1.00$ in.

Structural Loading:

- $F = 100.0$ lb
- $p_0 = 10.0$ psig

Solution:

The first step is to resolve the applied loads (the tension and pressure loads) at the center of curvature. This can be done in the following way.

$$P_x = -F \sin 45^\circ - p_o D_2 w = -80.7 \text{ lb}$$

$$P_y = -F \cos 45^\circ = -70.7 \text{ lb}$$

$$M_{xy} = F D_1 \cos 45^\circ + F D_2 \sin 45^\circ + (0.5) p_o w D_2^2 = 94.8 \text{ in-lb}$$

This gives all the necessary information for running CLA. The input session and output data are given on the following pages.

CLA Example Problem #2 Input:

CURVED LAMINATE ANALYSIS
VERSION 1.0

DEVELOPED BY MCAIR

ANALYSIS DEVELOPMENT: JOHN M. BUBB
PHILIP C. PAUL

HIT <RETURN> TO CONTINUE

THIS FORTRAN COMPUTER PROGRAM IS THE AUTOMATED
VERSION OF THE CURVED LAMINATE ANALYSIS METHOD.
THE METHOD IS BASED ON THE CURVED STIFFENER
ANALYSIS, CREATED BY JOHN BUBB. CLA DETERMINES
THE PLY BY PLY STRESS DISTRIBUTION AT ANY ANGULAR
POSITION IN THE CURVED LAMINATE. ORTHOTROPIC
MATERIALS, ARBITRARY LAYUPS, AND A GENERAL
LOADING ARE ALL PERMITTED BY CLA. THIS PROGRAM
WAS DEVELOPED UNDER THE NAVY CRAD "OUT-OF-PLANE
ANALYSIS FOR COMPOSITE STRUCTURES" (CONTRACT NO.
N62269-87-C-0226). FOR FURTHER INFORMATION, SEE
THE FINAL REPORT, OR CONTACT PHILIP C. PAUL.

HIT <RETURN> TO CONTINUE

ENTER THE PROBLEM DESCRIPTION (60 CHARACTERS MAX.) :
Example Problem #2

ENTER THE NUMBER OF MATERIALS (3 MAX.)
1

ENTER THE PROPERTIES FOR MATERIAL #1

- E1, MODULUS IN FIBRE DIRECTION (MSI)
- E2, MODULUS IN TRANSVERSE DIRECTION (MSI)
- G12, SHEAR MODULUS OF PLY (MSI)
- NU, POISSON RATIO OF PLY
- T, THICKNESS OF THE PLY

10.4 10.4 1.0 0.06 0.0140

ENTER THE LAMINATE ALLOWABLES.

- FLATWISE TENSION (KSI)
- INTERLAMINAR SHEAR (KSI)

3.0 11.0

ENTER THE TYPE OF LAYUP.

- 1: UNSYMMETRIC [A1,A2,...,AN] _
 - 2: ODD SYMMETRIC [A1,A2,A3,...,AN]S
 - 3: EVEN SYMMETRIC [A1,A2,A3,...,AN]MS
- 2

ODD SYMMETRIC LAYUP [A1,A2,A3,...,AN]S

ENTER THE NUMBER OF PLY ORIENTATIONS (2 <= N <= 50)

6

ENTERING A BACKSLASH (/) WITHIN THE FOLLOWING DATA
RECORDS ASSUMES THE REMAINING ITEMS TAKE THE VALUES
FROM THE PREVIOUS DATA RECORD

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 1
45.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 2
0.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 3
45.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 4
0.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 5
45.0 1

ENTER THE ORIENTATION (IN DEGREES),
AND MATERIAL IDENTIFICATION FOR PLY # 6
0.0 1

ENTER THE LAMINATE WIDTH (IN.)
1.00

ENTER THE INSIDE RADIUS OF THE CURVED LAMINATE (IN.)
0.20

ENTER THE NECESSARY ANGLES (DEGREES)
- THETA TOTAL: ANGLE SUBTENDED BY THE CURVED LAMINATE
- THETA INIT.: INITIAL ANGLE OF INSPECTION
- THETA FINAL: FINAL ANGLE OF INSPECTION
- INSP: NUMBER OF INSPECTIONS

(FOR ONLY ONE ANGLE OF INSPECTION AT THETA INITIAL,
ENTER INSP = 1)
90.0 30.0 60.0 3

ENTER THE LOADS RESOLVED AT THE CENTER OF CURVATURE.
- PX (LBS)
- PY (LBS)
- MXY (IN-LBS)
-80.7 -70.7 94.8

ENTER THE PRESSURE DIFFERENCE (PSI)
(POSITIVE VALUE FOR OUTWARD ACTING PRESSURE.)
10.0

DO YOU WANT OUTPUT FILES (Y/N)?
Y

ENTER THE OUTPUT FILE NAME (FILENAME.OUT) 9 CHARACTERS MAX.
EXAMPLE2

WHAT TYPE OF SCREEN OUTPUT DO YOU WANT?
1) FULL: STRESS STATE AT EVERY ANGLE
2) TRUNCATED: STRESS STATE ONLY AT FAILURE LOCATION
1

CLA Example Problem #2 Output:

**** CURVED LAMINATE ANALYSIS ****

INPUT DATA

Example Problem #2

PROPERTIES FOR MATERIAL #1:

E1 = 1.0400E+07 PSI
 E2 = 1.0400E+07 PSI
 G12 = 1.0000E+06 PSI
 NU = .06
 T = 1.4000E-02 IN.

LAMINATE ALLOWABLES:

- FLATWISE TENSION = 3.0000E+03 PSI
 - INTERLAMINAR SHEAR = 1.1000E+04 PSI

HIT <RETURN> TO CONTINUE

PLY ORIENTATION MATERIAL PLY ORIENTATION MATERIAL

1	45.0	1	7	45.0	1
2	.0	1	8	.0	1
3	45.0	1	9	45.0	1
4	.0	1	10	.0	1
5	45.0	1	11	45.0	1
6	.0	1			

HIT <RETURN> TO CONTINUE

THE LAMINATE WIDTH: 1.000
 THE LAMINATE INNER RADIUS: .200
 THE RADIUS OF THE NEUTRAL AXIS: .270
 THE RADIUS OF THE CENTROID: .277

ANGLE INFORMATION:

- THETA TOTAL: 90.0
 - THETA INITIAL: 30.0
 - THETA FINAL: 60.0
 - INSPECTIONS: 3

APPLIED LOADS:

- PX: -8.0700E+01
 - PY: -7.0700E+01
 - MXY: 9.4800E+01
 - PO: 1.0000E+01

HIT <RETURN> TO CONTINUE

CRITICAL STRESS STATE AT THETA = 60.0

THE MAXIMUM FAILURE INDEX IS: 1.37

PREDICTED FAILURE INFORMATION

- PX: -5.8923E+01
 - PY: -5.1622E+01
 - MXY: 6.9219E+01
 - PRESSURE: 7.3015E+00
 - RADIAL STRESS (PSI): 2.992E+03
 - SHEAR STRESS (PSI): -8.180E+02
 - INTERFACE OF FAILURE: 4

CONTRIBUTIONS TO FAILURE

- RADIAL STRESS: 99.4%
 - SHEAR STRESS: .6%

HIT <RETURN> TO CONTINUE

STRESS STATE AT THETA = 30.0

INTERFACE NUMBER	RADIAL POSITION	RADIAL STRESS	SHEAR STRESS	FAILURE INDEX
1	2.1400E-01	9.4065E+02	-3.2719E+02	.31
2	2.2800E-01	2.7341E+03	-9.4258E+02	.92
3	2.4200E-01	3.0223E+03	-9.9874E+02	1.01
4	2.5600E-01	3.4952E+03	-1.1520E+03	1.17
5	2.7000E-01	3.3967E+03	-1.0791E+03	1.14
6	2.8400E-01	3.0736E+03	-9.7531E+02	1.03
7	2.9800E-01	2.7687E+03	-8.5021E+02	.93
8	3.1200E-01	1.9573E+03	-6.0093E+02	.65
9	3.2600E-01	1.5472E+03	-4.6115E+02	.52
10	3.4000E-01	4.3501E+02	-1.2974E+02	.15

HIT <RETURN> TO CONTINUE

THE MAXIMUM FAILURE INDEX IS: 1.17

PREDICTED FAILURE INFORMATION

- PX: -6.8988E+01
 - PY: -6.0440E+01
 - MXY: 8.1042E+01
 - PRESSURE: 8.5487E+00
 - RADIAL STRESS (PSI): 2.988E+03
 - SHEAR STRESS (PSI): -9.848E+02
 - INTERFACE OF FAILURE: 4

CONTRIBUTIONS TO FAILURE

- RADIAL STRESS: 99.2%
 - SHEAR STRESS: .8%

HIT <RETURN> TO CONTINUE

STRESS STATE AT THETA = 45.0

INTERFACE NUMBER	RADIAL POSITION	RADIAL STRESS	SHEAR STRESS	FAILURE INDEX
1	2.1400E-01	1.0266E+03	-3.3406E+02	.34
2	2.2800E-01	2.9818E+03	-9.6238E+02	1.00
3	2.4200E-01	3.2847E+03	-1.0197E+03	1.10
4	2.5600E-01	3.7978E+03	-1.1762E+03	1.27
5	2.7000E-01	3.6800E+03	-1.1017E+03	1.23
6	2.8400E-01	3.3297E+03	-9.9579E+02	1.11
7	2.9800E-01	2.9919E+03	-8.6808E+02	1.00
8	3.1200E-01	2.1150E+03	-6.1356E+02	.71
9	3.2600E-01	1.6682E+03	-4.7084E+02	.56
10	3.4000E-01	4.6905E+02	-1.3246E+02	.16

HIT <RETURN> TO CONTINUE

THE MAXIMUM FAILURE INDEX IS: 1.27

PREDICTED FAILURE INFORMATION

- PX: -6.3521E+01
 - PY: -5.5650E+01
 - MXY: 7.4620E+01
 - PRESSURE: 7.8713E+00
 - RADIAL STRESS (PSI): 2.989E+03
 - SHEAR STRESS (PSI): -9.258E+02
 - INTERFACE OF FAILURE: 4

CONTRIBUTIONS TO FAILURE

- RADIAL STRESS: 99.3%
 - SHEAR STRESS: .7%

HIT <RETURN> TO CONTINUE

STRESS STATE AT THETA = 60.0

INTERFACE NUMBER	RADIAL POSITION	RADIAL STRESS	SHEAR STRESS	FAILURE INDEX
1	2.1400E-01	1.1117E+03	-3.1817E+02	.37
2	2.2800E-01	3.2270E+03	-9.1660E+02	1.08
3	2.4200E-01	3.5444E+03	-9.7121E+02	1.18
4	2.5600E-01	4.0974E+03	-1.1203E+03	1.37
5	2.7000E-01	3.9604E+03	-1.0493E+03	1.32
6	2.8400E-01	3.5832E+03	-9.4842E+02	1.20
7	2.9800E-01	3.2128E+03	-8.2678E+02	1.07
8	3.1200E-01	2.2712E+03	-5.8437E+02	.76
9	3.2600E-01	1.7880E+03	-4.4844E+02	.60
10	3.4000E-01	5.0275E+02	-1.2616E+02	.17

HIT <RETURN> TO CONTINUE

THE MAXIMUM FAILURE INDEX IS: 1.37

PREDICTED FAILURE INFORMATION

- PX: -5.8923E+01
- PY: -5.1622E+01
- MXY: 6.9219E+01
- PRESSURE: 7.3015E+00
- RADIAL STRESS (PSI): 2.992E+03
- SHEAR STRESS (PSI): -8.180E+02
- INTERFACE OF FAILURE: 4

CONTRIBUTIONS TO FAILURE

- RADIAL STRESS: 99.4%
- SHEAR STRESS: .6%

HIT <RETURN> TO CONTINUE

The margin of safety (MS) is defined as

$$MS = \frac{\text{ultimate load}}{\text{applied load}} - 1.0$$

For this example, the predicted ultimate load, Px, is -58.9 lb., while the applied load is -80.7 lb. The margin of safety is then -0.27, indicating that failure has occurred under the applied loads.

SECTION IV

INTERFACIAL STRESS FIELD MODEL (ISFM) USER'S MANUAL

1. Introduction

1.1 Function – The stress state within the cocured or adhesively bonded joints of integrally stiffened composite structures is not easily defined. The Interfacial Stress Field Model (ISFM) was derived to predict this interfacial stress state. ISFM provides a tool for performing trade studies that compare joint designs, and for predicting the strength of the joints. ISFM rapidly determines the stress state of the joints since the model is based on a closed-form analysis. ISFM allows two types of out-of-plane loadings to be analyzed. Integrally stiffened structures subjected to direct out-of-plane loads and integrally stiffened postbuckled panels can be assessed by ISFM. The program output includes an echo of the input information, the interfacial normal stress distributions, and the predicted failure load.

The ISFM analysis is a strength of materials derivation. The skin and stiffener flange are modeled as separate beams, with the joint modeled as a common elastic foundation, as shown in Figure 44. The interfacial stresses are determined independent of each other. The normal stress is simply the elastic foundation modulus multiplied by the difference of the deflections of the two beams. The shear stress is determined using a shear lag analysis, and is based on the assumption that significant membrane loads can arise in the skin and stiffener flange. A more detailed discussion of these derivations can be found in the Analysis Development and Verification section of Volume I of this report.

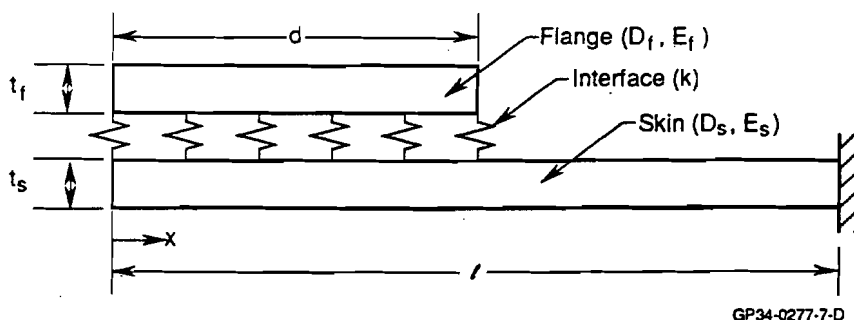


Figure 44. General Interfacial Stress Field Model

The program described in this section does not calculate the shear stress distribution since the shear lag analysis was found to be unreliable for all configurations. This behavior is discussed in Volume I of this report. The SSAM code, described in Section VII, does include the shear stress effects and is recommended for cases where significant shear effects may be present.

1.2 Limitations – ISFM is limited both by its assumed boundary conditions, and material properties that are often not well defined. The normal stress solution is based on eight assumed boundary conditions which may or may not be appropriate, depending upon the structure that is being analyzed. The present analysis incorporates two sets of boundary conditions, one set for direct load cases, the other set for postbuckling cases. These sets of boundary conditions were selected to address the most practical applications.

ISFM requires material dependent input values which are not available from standard tests. The elastic foundation modulus (k) is almost always an assumed value. When performing correlation of the analysis with test results, empiricism has been introduced to determine this value.

The accuracy of the failure predictions is dependent upon the interlaminar tension stress allowable value that is used in the failure criterion. This allowable is not always readily available, and often needs to be estimated.

2. Input Description

The input information for the ISFM program is entered in a straightforward manner. Features have been included to resist inputting erroneous data in the program. The description of each input item is given below.

1. Analysis Type – Requests either of the two following options.

- Direct loading analysis
- Indirect loading (postbuckling) analysis

2. Problem Description – Requests a problem title. A maximum of 55 alphanumeric characters is accepted.

3. Geometry Information – Requests the following data for the model shown in Figure 44.

- d , flange length
- l , skin length
- t_f , flange thickness
- t_s , skin thickness

4. Element Stiffnesses – Asks for the in-plane flexural rigidities in the x-direction.

- D_f , flange stiffness

- D_s , skin stiffness

5. Elastic Moduli – Requests the x-direction elastic moduli.

- E_f , flange modulus

- E_s , skin modulus

6. Interface Material Information – Asks for the elastic foundation modulus of the cocured/bonded joint.

- k , elastic foundation modulus

7. Stress Allowable – Requests the interface flatwise tension stress allowable.

- FWT, flatwise tension

8. Applied Load/Deflection – Asks for the imposed boundary value, depending upon the analysis type that was chosen.

- P , applied load (for direct loading analysis)

- M , applied moment (for direct loading analysis)

- δ , imposed buckle deflection (for indirect loading analysis)

9. Output Options – Allows the user to request whether or not to use an output file.

- Output file (yes/no)

- Output file name

3. Output Description

The output of ISFM consists of two sections. An echo of the input data is printed in the first section. This reminds the user what problem parameters were chosen. The second section presents the calculated values of the problem in tabular form. The program determines the flange deflection, skin deflection, and normal stress at equidistant locations along the skin/flange interface.

4. Verification

The verification of ISFM consisted of correlation with test data. Cocured blade and hat stiffeners were tested in pull off tension to simulate the effects of direct loads on integrally stiffened composite structures. The elastic foundation modulus (k) was estimated by finding the value of k necessary for ISFM to predict the failure of a set of data, and assume that value for the remaining predictions. The test configuration and correlation are summarized in Figure 45.

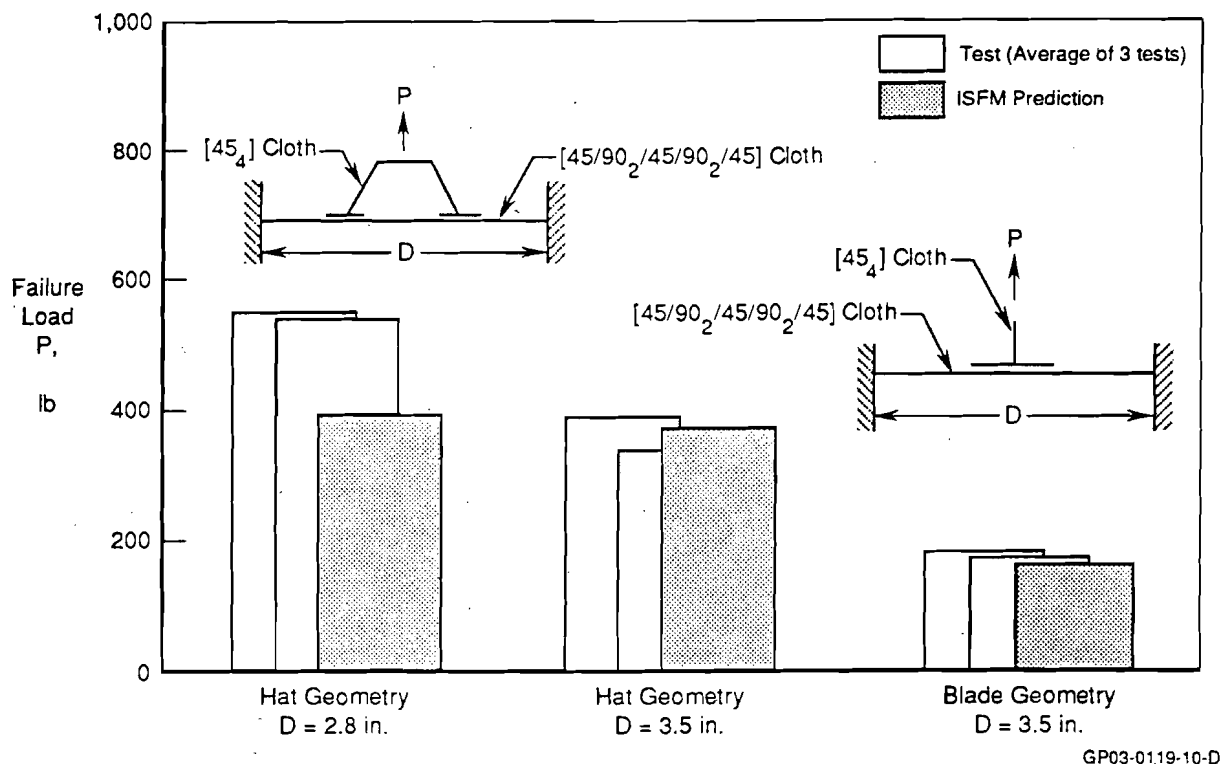
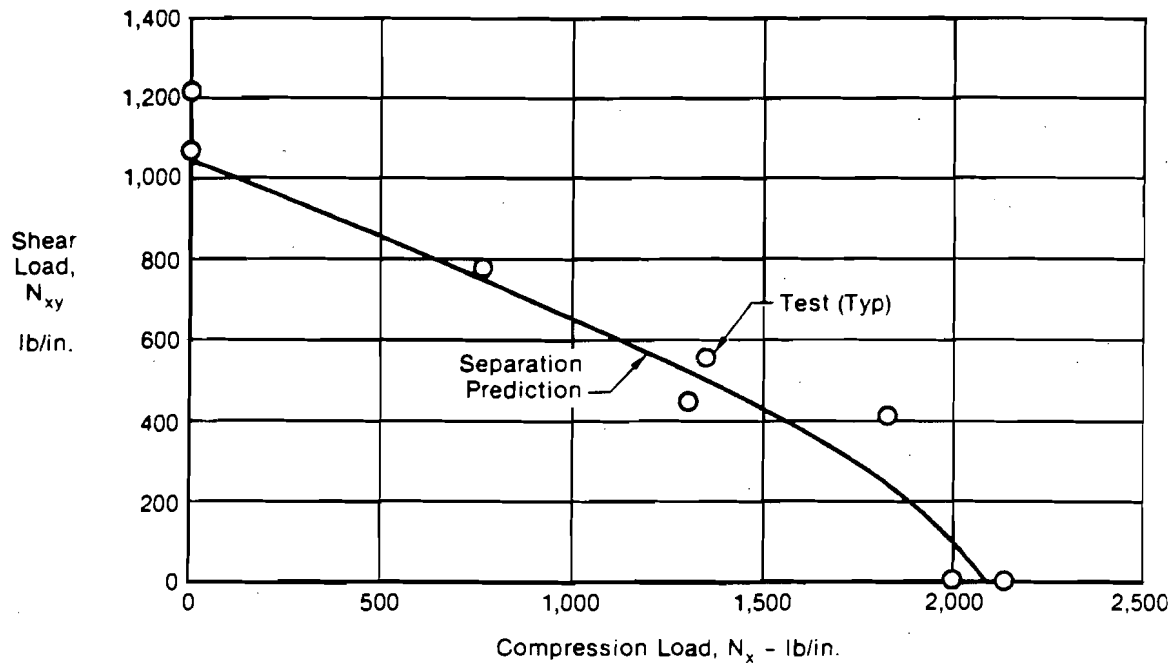


Figure 45. Correlation of Predicted and Measured Pull Off Strengths

This method has also been verified by correlating the ISFM predictions with results of postbuckling tests of flat composite panels. Multi-bay carbon/epoxy panels were tested at several different combinations of shear and compression loads. Skin/stiffener separation was the initiating mode of failure in all cases. Correlation between predicted and test separation loads is shown in Figure 46 for a range of combined loads cases.

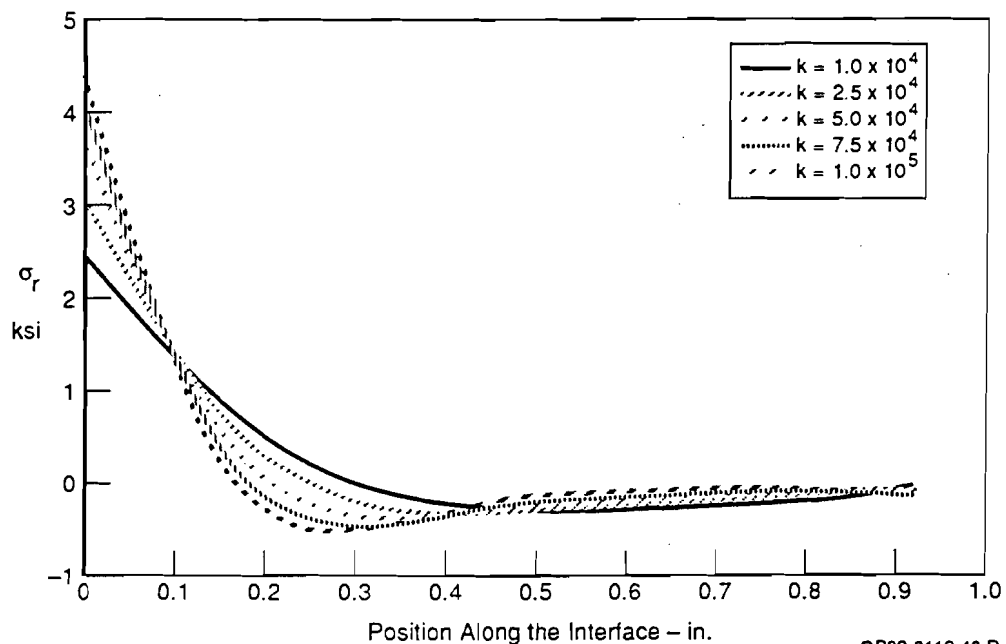
Parametric studies were also conducted to verify that the analysis demonstrates the proper trends. One study investigated the influence of the elastic foundation modulus value on the predicted stress. Figure 47 shows the variations in the normal stress distribution due to altering the k value. Increasing the elastic foundation modulus value by an order of magnitude can almost double the maximum predicted stress value. The gradient of the stress distribution also becomes more pronounced when k is increased. The fact that the maximum predicted normal

stress occurs at the web end of the flange is reassuring, since this is frequently the suspected location of failure initiation for direct load cases.



GP03-0119-37

Figure 46. Correlation of ISFM and Test Data of Panels Subjected to Combined Loads



GP03-0119-40-D

Figure 47. ISFM Stress Variations Resulting From Different Values for the Elastic Foundation Modulus (k)

5. Subroutine Functions

Figure 48 provides the flowchart of the ISFM FORTRAN computer program. The main program (ISFM routine) acts as a focal point for the modular construction of ISFM. The subroutines are accessed from this routine.

A brief description of each subroutine's functions follows.

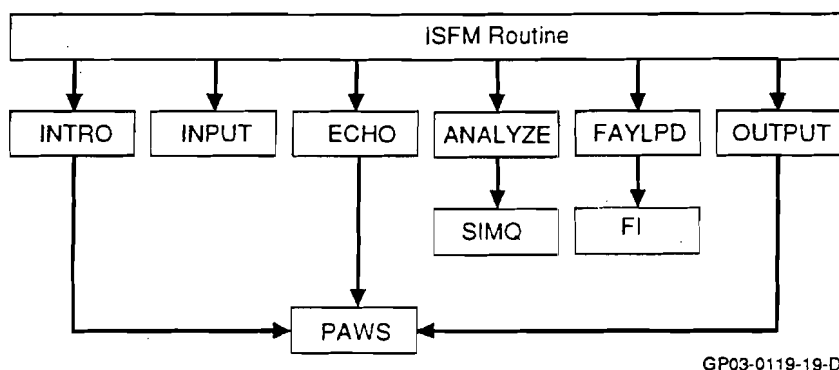


Figure 48. ISFM Flowchart

ANALYZ: This subroutine is the heart of the ISFM analysis, and is responsible for determining the interfacial normal stresses. The boundary condition equations necessary for finding the interfacial normal stresses are established in this routine. These are solved simultaneously to determine the coefficients of the equations that describe the deflections and the normal stresses.

ECHO: The input data is summarized as part of the output to the computer terminal (and output file, if requested) by this routine.

FAYLPD: The failure prediction is made by this routine by determining the maximum Tsai-Hill failure index for each position along the joint interface. The predicted failure loads/deflection is the applied loads/deflection proportioned by the inverse of the maximum failure index.

FI: This function routine determines the failure index for a given normal stress and its respective allowable value.

INPUT: The input data is entered into the program through this routine.

INTRO: This subroutine prints out an introductory message to the terminal at the start of each run. This introduction includes the program header, a legal disclaimer, and some general comments about the analysis.

OUTPUT: This routine presents the output information in tabular form. The flange deflection, skin deflection and normal stress are stated for each given position along the interface.

PAWS: This subroutine is called each time the program output to the screen needs to be paused. A carriage return allows the program to continue.

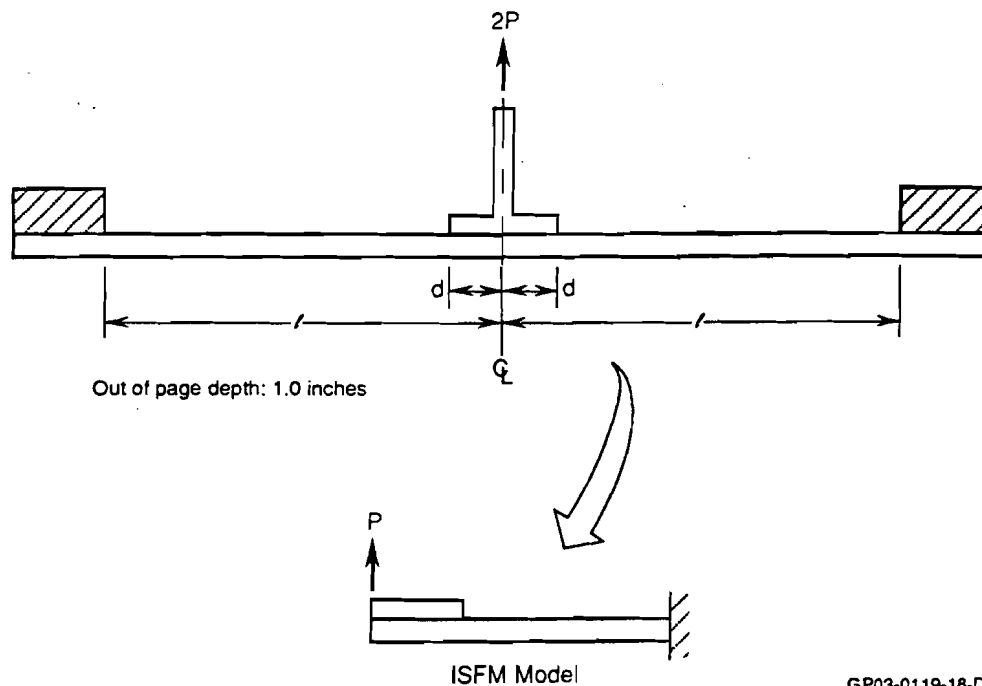
SIMQ: This routine solves a system of N linearly independent equations to find the N unknown coefficients. The subroutine ANALYZ calls this routine as part of its solution to determine the deflections and the normal stress distribution.

6. Example Problems

Two example problems are given in this section to illustrate the use of the ISFM program. Both analysis types and their application are presented.

6.1 Problem #1

An integral blade stiffener element (depth = 1.0 in.) is subjected to direct loads in a pull off test, as shown in Figure 49. The predicted strength of the element is desired. The information necessary for performing the analysis is given below.



GP03-0119-18-D

Figure 49. ISFM Analysis of Blade Stiffener Pull Off Specimen

Material: AS4/3501-6

$$E_1 = 18.5 \text{ msi}$$

$$E_2 = 1.90 \text{ msi}$$

$$G_{12} = 0.85 \text{ msi}$$

$$\nu_{12} = 0.3$$

$$G_{\text{resin}} = 0.25 \text{ msi}$$

Flatwise Tension Strength: 3.50 ksi

Flange

Length: 0.50 in.

Thickness: 0.028 in.

Stiffness: 6.169 lb-in²

Modulus: 3.39 msi

Skin

Length: 2.50 in.

Thickness: 0.080 in.

Stiffness: 169.2 lb-in²

Modulus: 5.85 msi

Solution:

The information for performing this analysis is all given above, except for the elastic foundation modulus and the applied load. The elastic foundation modulus is estimated as

$$k = \frac{E_2}{t_{\min}}$$

where

E_2 = the transverse modulus of a tape ply

$t_{\min}$ = the smaller of the flange or skin thickness

This value is found to be

$$k = \frac{1.90 \times 10^6}{0.028} = 67.9 \text{ msi/in.}$$

The applied load is assumed to be unity ($P = 1.0$) for making a failure prediction. Any value could be used, but this one allows the stresses to be easily proportioned by the predicted failure load to find the failure stress state. For this example, no moment is applied.

The interactive ISFM run of this problem is given on the following pages. The same information is contained in the associated output file. The output shows the predicted failure load to be $P = 15.298 \text{ lb.}$

ISFM Example Problem #1 Input:

Interfacial Stress Field Model
Version 1.1

Developed by MCAIR

Analysis Development: Ken B. Sanger
Philip C. Paul

HIT <RETURN> TO CONTINUE

COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS
RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
DOUGLAS CORPORATION.

THIS MATERIAL MAY BE REPRODUCED BY OR FOR THE
U.S. GOVERNMENT PURSUANT TO THE COPYRIGHT LICENSE
UNDER THE CLAUSE AT 52.227-7013 (MAY 1981).

HIT <RETURN> TO CONTINUE

This FORTRAN computer program is the automated version of the Interfacial Stress Field Model. This method was derived to determine the interfacial stress state of the cocured or adhesively bonded joints of integrally stiffened composite structures. The analysis can be used either on panels under direct loads or on panels under indirect (buckling) loads. This program was developed under the Navy CRAD "Out-of-Plane Analysis for Composite Structures" (Contract No.

N62269-87-C-0226). For further information, see the final report, or contact Philip C. Paul.

HIT <RETURN> TO CONTINUE

WHAT TYPE OF ANALYSIS DO YOU WISH TO PERFORM?

- 1) DIRECT LOADING ANALYSIS
- 2) INDIRECT LOADING (POSTBUCKLING) ANALYSIS

1

ENTER THE PROBLEM DESCRIPTION (55 CHAR. MAX.):

Example #1 - Blade Stiffened Panel Under Direct Loads

ENTER THE GEOMETRY INFORMATION

- D: FLANGE LENGTH
- L: TOTAL SKIN LENGTH
- TF: FLANGE THICKNESS
- TS: SKIN THICKNESS

0.50,2.50,0.0280,0.080

ENTER THE ELEMENT STIFFNESSES

- DF: FLANGE STIFFNESS
- DS: SKIN STIFFNESS

6.196,169.2

ENTER THE X-DIRECTION MODULI

- EF: FLANGE MODULUS
- ES: SKIN MODULUS

3.39e6,5.85e6

ENTER THE INTERFACE MATERIAL INFORMATION

- K: ELASTIC FOUNDATION MODULUS

67.9e6

ENTER THE INTERFACE MATERIAL STRESS ALLOWABLE

- FWT: FLATWISE TENSION ALLOWABLE

3500.0

ENTER THE APPLIED LOADS

- P: PULL OFF LOAD
- M: MOMENT

1.0,0.0

DO YOU WANT AN OUTPUT FILE (Y/N)?

Y

ENTER THE FILE NAME (9 CHARACTERS MAX.)

Examp11

ISFM Example Problem #1 Output:

INPUT DATA ECHO

Example #1 - Blade Stiffened Panel Under Direct Loads

DATE: 29-APR-91

TIME: 10:43:24

--- DIRECT LOADING ANALYSIS ---

GEOMETRY INFORMATION

- FLANGE LENGTH (D): 5.0000E-01
- TOTAL SKIN LENGTH (L): 2.5000E+00
- FLANGE THICKNESS (TF): 2.8000E-02
- SKIN THICKNESS (TS): 8.0000E-02

ELEMENT STIFFNESSES

- FLANGE STIFFNESS (DF): 6.1960E+00
- SKIN STIFFNESS (DS): 1.6920E+02

HIT <RETURN> TO CONTINUE

X-DIRECTION MODULI

- FLANGE MODULUS (EF): 3.3900E+06
- SKIN MODULUS (ES): 5.8500E+06

INTERFACE MATERIAL INFORMATION

- ELASTIC FOUNDATION MODULUS (K): 6.7900E+07

INTERFACE MATERIAL STRESS ALLOWABLE

- FLATWISE TENSION ALLOWABLE (FWT): 3.5000E+03

APPLIED LOADS

- PULL OFF LOAD (P): 1.0000E+00
- MOMENT (M): 0.0000E+00

HIT <RETURN> TO CONTINUE

PLEASE WAIT A MOMENT FOR THE OUTPUT DATA....

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
0.0000E+00	7.6046E-03	7.6013E-03	2.2878E+02
1.0000E-02	7.6024E-03	7.6009E-03	9.9549E+01
2.0000E-02	7.6001E-03	7.5998E-03	2.0432E+01
3.0000E-02	7.5977E-03	7.5980E-03	-1.8946E+01
4.0000E-02	7.5949E-03	7.5954E-03	-3.2036E+01
5.0000E-02	7.5917E-03	7.5922E-03	-3.0632E+01
6.0000E-02	7.5879E-03	7.5882E-03	-2.3166E+01
7.0000E-02	7.5834E-03	7.5836E-03	-1.4701E+01
8.0000E-02	7.5782E-03	7.5783E-03	-7.6901E+00
9.0000E-02	7.5723E-03	7.5723E-03	-2.8860E+00
1.0000E-01	7.5657E-03	7.5657E-03	-1.2702E-01
1.1000E-01	7.5584E-03	7.5584E-03	1.1153E+00
1.2000E-01	7.5505E-03	7.5505E-03	1.4125E+00

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
1.3000E-01	7.5419E-03	7.5419E-03	1.2274E+00
1.4000E-01	7.5327E-03	7.5327E-03	8.7155E-01
1.5000E-01	7.5228E-03	7.5228E-03	5.2010E-01
1.6000E-01	7.5123E-03	7.5123E-03	2.4917E-01
1.7000E-01	7.5012E-03	7.5012E-03	7.4279E-02
1.8000E-01	7.4894E-03	7.4894E-03	-1.9210E-02
1.9000E-01	7.4771E-03	7.4771E-03	-5.5902E-02
2.0000E-01	7.4641E-03	7.4641E-03	-5.9298E-02
2.1000E-01	7.4506E-03	7.4506E-03	-4.7176E-02
2.2000E-01	7.4364E-03	7.4364E-03	-3.0872E-02
2.3000E-01	7.4216E-03	7.4216E-03	-1.6441E-02
2.4000E-01	7.4063E-03	7.4063E-03	-6.3949E-03
2.5000E-01	7.3904E-03	7.3904E-03	-1.2937E-03

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
2.6000E-01	7.3739E-03	7.3739E-03	-8.7799E-04
2.7000E-01	7.3568E-03	7.3568E-03	-4.6527E-03
2.8000E-01	7.3392E-03	7.3392E-03	-1.1943E-02
2.9000E-01	7.3210E-03	7.3210E-03	-2.1480E-02
3.0000E-01	7.3023E-03	7.3023E-03	-3.0595E-02
3.1000E-01	7.2830E-03	7.2830E-03	-3.4185E-02

```

3.2000E-01 7.2632E-03 7.2632E-03 -2.3778E-02
3.3000E-01 7.2429E-03 7.2429E-03 1.2718E-02
3.4000E-01 7.2220E-03 7.2220E-03 8.9604E-02
3.5000E-01 7.2006E-03 7.2006E-03 2.1897E-01
3.6000E-01 7.1787E-03 7.1787E-03 4.0173E-01
3.7000E-01 7.1563E-03 7.1562E-03 6.1291E-01
3.8000E-01 7.1333E-03 7.1333E-03 7.8136E-01

```

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
3.9000E-01	7.1099E-03	7.1099E-03	7.6710E-01
4.0000E-01	7.0860E-03	7.0860E-03	3.4481E-01
4.1000E-01	7.0615E-03	7.0615E-03	-7.9023E-01
4.2000E-01	7.0366E-03	7.0366E-03	-2.9684E+00
4.3000E-01	7.0112E-03	7.0113E-03	-6.4107E+00
4.4000E-01	6.9853E-03	6.9854E-03	-1.0977E+01
4.5000E-01	6.9589E-03	6.9591E-03	-1.5776E+01
4.6000E-01	6.9321E-03	6.9324E-03	-1.8668E+01
4.7000E-01	6.9049E-03	6.9051E-03	-1.5755E+01
4.8000E-01	6.8774E-03	6.8775E-03	-1.1265E+00
4.9000E-01	6.8498E-03	6.8493E-03	3.2694E+01
5.0000E-01	6.8221E-03	6.8207E-03	9.2948E+01

THE ENTERED PULL OFF LOAD IS: 1.0000E+00
THE ENTERED MOMENT IS: 0.0000E+00

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
0.0000E+00	1.1634E-01	1.1629E-01	3.5000E+03
1.0000E-02	1.1630E-01	1.1628E-01	1.5229E+03
2.0000E-02	1.1627E-01	1.1626E-01	3.1257E+02
3.0000E-02	1.1623E-01	1.1624E-01	-2.8985E+02
4.0000E-02	1.1619E-01	1.1620E-01	-4.9009E+02
5.0000E-02	1.1614E-01	1.1615E-01	-4.6862E+02
6.0000E-02	1.1608E-01	1.1609E-01	-3.5440E+02
7.0000E-02	1.1601E-01	1.1602E-01	-2.2491E+02
8.0000E-02	1.1593E-01	1.1594E-01	-1.1765E+02
9.0000E-02	1.1584E-01	1.1584E-01	-4.4151E+01
1.0000E-01	1.1574E-01	1.1574E-01	-1.9432E+00
1.1000E-01	1.1563E-01	1.1563E-01	1.7062E+01
1.2000E-01	1.1551E-01	1.1551E-01	2.1609E+01

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
1.3000E-01	1.1538E-01	1.1538E-01	1.8778E+01
1.4000E-01	1.1524E-01	1.1524E-01	1.3333E+01
1.5000E-01	1.1509E-01	1.1509E-01	7.9567E+00
1.6000E-01	1.1493E-01	1.1493E-01	3.8119E+00
1.7000E-01	1.1476E-01	1.1476E-01	1.1364E+00
1.8000E-01	1.1458E-01	1.1458E-01	-2.9388E-01
1.9000E-01	1.1439E-01	1.1439E-01	-8.5521E-01
2.0000E-01	1.1419E-01	1.1419E-01	-9.0716E-01
2.1000E-01	1.1398E-01	1.1398E-01	-7.2172E-01
2.2000E-01	1.1376E-01	1.1376E-01	-4.7230E-01
2.3000E-01	1.1354E-01	1.1354E-01	-2.5153E-01
2.4000E-01	1.1330E-01	1.1330E-01	-9.7832E-02
2.5000E-01	1.1306E-01	1.1306E-01	-1.9791E-02

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
2.6000E-01	1.1281E-01	1.1281E-01	-1.3432E-02
2.7000E-01	1.1255E-01	1.1255E-01	-7.1178E-02
2.8000E-01	1.1228E-01	1.1228E-01	-1.8271E-01
2.9000E-01	1.1200E-01	1.1200E-01	-3.2861E-01
3.0000E-01	1.1171E-01	1.1171E-01	-4.6805E-01
3.1000E-01	1.1142E-01	1.1142E-01	-5.2298E-01
3.2000E-01	1.1112E-01	1.1112E-01	-3.6377E-01
3.3000E-01	1.1080E-01	1.1080E-01	1.9457E-01
3.4000E-01	1.1048E-01	1.1048E-01	1.3708E+00
3.5000E-01	1.1016E-01	1.1016E-01	3.3499E+00
3.6000E-01	1.0982E-01	1.0982E-01	6.1458E+00
3.7000E-01	1.0948E-01	1.0948E-01	9.3766E+00
3.8000E-01	1.0913E-01	1.0913E-01	1.1954E+01

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
3.9000E-01	1.0877E-01	1.0877E-01	1.1735E+01
4.0000E-01	1.0840E-01	1.0840E-01	5.2750E+00

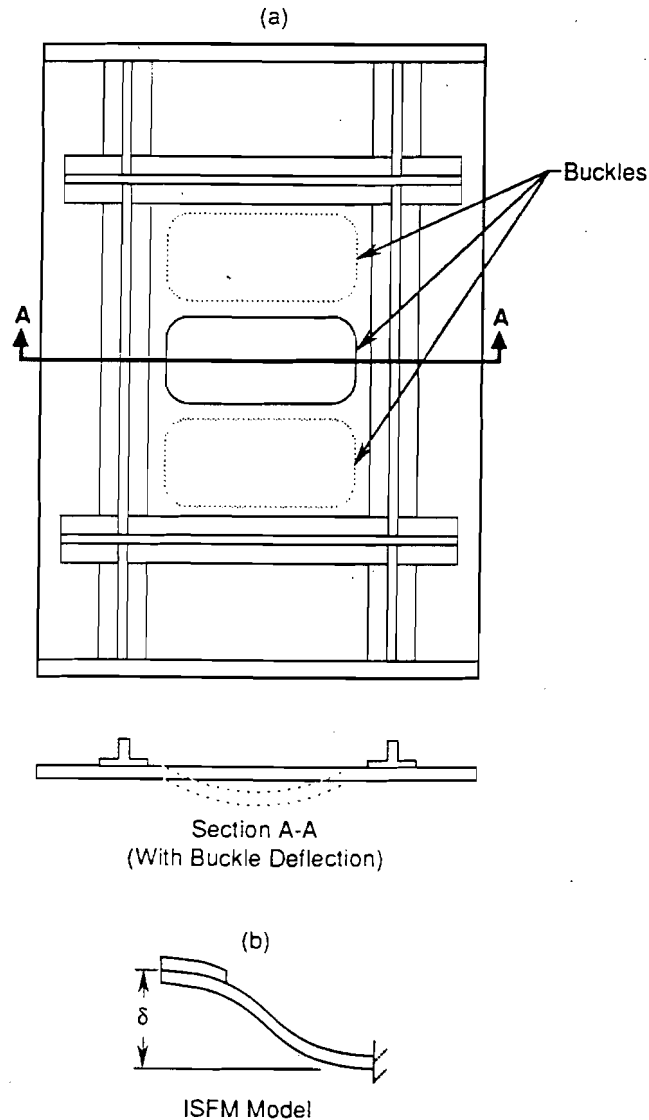
4.1000E-01	1.0803E-01	1.0803E-01	-1.2089E+01
4.2000E-01	1.0765E-01	1.0765E-01	-4.5412E+01
4.3000E-01	1.0726E-01	1.0726E-01	-9.8073E+01
4.4000E-01	1.0686E-01	1.0687E-01	-1.6793E+02
4.5000E-01	1.0646E-01	1.0646E-01	-2.4135E+02
4.6000E-01	1.0605E-01	1.0605E-01	-2.8560E+02
4.7000E-01	1.0563E-01	1.0564E-01	-2.4102E+02
4.8000E-01	1.0521E-01	1.0521E-01	-1.7233E+01
4.9000E-01	1.0479E-01	1.0478E-01	5.0016E+02
5.0000E-01	1.0437E-01	1.0435E-01	1.4220E+03

THE PREDICTED PULL OFF LOAD AT FAILURE IS: 1.5298E+01

THE PREDICTED MOMENT AT FAILURE IS: 0.0000E+00

6.2 Problem #2

An integral blade stiffened panel is subjected to compressive buckling loads. The panel is shown in Figure 50, and features the same blade stiffener that was used in Problem #1. The maximum buckling deflection at failure was found to be $\delta = 0.10$ in. The objective of this problem is to determine the predicted stress state at test failure, as well as the predicted failure deflection. Refer to Problem #1 for determining the input data.



GP03-0119-17-D

Figure 50. ISFM Analysis of an Integral Blade Stiffened Panel

Solution:

The analysis is similar to that run for Problem #1, except this is an indirect loads analysis. The complete interactive session is shown on the following pages. The stress state is given as part of the this session. The predicted failure deflection was determined to be $d = 0.084445$ in.

ISFM Example Problem #2 Input:

```

*****
Interfacial Stress Field Model
Version 1.1

Developed by MCAIR

Analysis Development: Ken B. Sanger
                    Philip C. Paul

*****

HIT <RETURN> TO CONTINUE

*****

COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS
RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
DOUGLAS CORPORATION.

THIS MATERIAL MAY BE REPRODUCED BY OR FOR THE
U.S. GOVERNMENT PURSUANT TO THE COPYRIGHT LICENSE
UNDER THE CLAUSE AT 52.227-7013 (MAY 1981).

*****

HIT <RETURN> TO CONTINUE

*****

This FORTRAN computer program is the automated
version of the Interfacial Stress Field Model.
This method was derived to determine the
interfacial stress state of the cocured or
adhesively bonded joints of integrally stiffened
composite structures. The analysis can be used
either on panels under direct loads or on panels
under indirect (buckling) loads. This program
was developed under the Navy CRAD "Out-of-Plane
Analysis for Composite Structures" (Contract No.
N62269-87-C-0226). For further information, see
the final report, or contact Philip C. Paul.

*****

HIT <RETURN> TO CONTINUE

```

WHAT TYPE OF ANALYSIS DO YOU WISH TO PERFORM?

1) DIRECT LOADING ANALYSIS

2) INDIRECT LOADING (POSTBUCKLING) ANALYSIS

2

ENTER THE PROBLEM DESCRIPTION (55 CHAR. MAX.) :

Example #2 - Buckled Blade Stiffened Panel

ENTER THE GEOMETRY INFORMATION

- D: FLANGE LENGTH
- L: TOTAL SKIN LENGTH
- TF: FLANGE THICKNESS
- TS: SKIN THICKNESS

0.50,2.50,0.0280,0.080

ENTER THE ELEMENT STIFFNESSES

- DF: FLANGE STIFFNESS
- DS: SKIN STIFFNESS

6.196,169.2

ENTER THE X-DIRECTION MODULI

- EF: FLANGE MODULUS
- ES: SKIN MODULUS

3.39e6,5.85e6

ENTER THE INTERFACE MATERIAL INFORMATION

- K: ELASTIC FOUNDATION MODULUS

67.9e6

ENTER THE INTERFACE MATERIAL STRESS ALLOWABLE

- FWT: FLATWISE TENSION ALLOWABLE

3500.0

ENTER THE BUCKLE DEFLECTION (DELTA)

0.1

DO YOU WANT AN OUTPUT FILE (Y/N)?

n

ISFM Example Problem #2 Output:

INPUT DATA ECHO

Example #2 - Buckled Blade Stiffened Panel

DATE: 29-APR-91

TIME: 11:08:49

**** INDIRECT LOADING (POSTBUCKLING) ANALYSIS ****

GEOMETRY INFORMATION

- FLANGE LENGTH (D): 5.0000E-01
- TOTAL SKIN LENGTH (L): 2.5000E+00
- FLANGE THICKNESS (TF): 2.8000E-02
- SKIN THICKNESS (TS): 8.0000E-02

ELEMENT STIFFNESSES

- FLANGE STIFFNESS (DF): 6.1960E+00

- SKIN STIFFNESS (DS): 1.6920E+02

HIT <RETURN> TO CONTINUE

X-DIRECTION MODULI

- FLANGE MODULUS (EF): 3.3900E+06
- SKIN MODULUS (ES): 5.8500E+06

INTERFACE MATERIAL INFORMATION

- ELASTIC FOUNDATION MODULUS (K): 6.7900E+07

INTERFACE MATERIAL STRESS ALLOWABLE

- FLATWISE TENSION ALLOWABLE (FWT): 3.5000E+03

BUCKLE DEFLECTION (DELTA): 1.0000E-01

HIT <RETURN> TO CONTINUE

PLEASE WAIT A MOMENT FOR THE OUTPUT DATA....

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
0.0000E+00	1.0000E-01	1.0007E-01	-4.5432E+03
1.0000E-02	1.0002E-01	1.0007E-01	-2.8903E+03
2.0000E-02	1.0004E-01	1.0006E-01	-1.5169E+03
3.0000E-02	1.0005E-01	1.0006E-01	-5.7348E+02
4.0000E-02	1.0005E-01	1.0005E-01	-3.0168E+01
5.0000E-02	1.0004E-01	1.0003E-01	2.1562E+02
6.0000E-02	1.0002E-01	1.0001E-01	2.7555E+02
7.0000E-02	9.9997E-02	9.9993E-02	2.4031E+02
8.0000E-02	9.9969E-02	9.9966E-02	1.7108E+02
9.0000E-02	9.9936E-02	9.9935E-02	1.0237E+02
1.0000E-01	9.9898E-02	9.9897E-02	4.9248E+01
1.1000E-01	9.9854E-02	9.9854E-02	1.4865E+01
1.2000E-01	9.9804E-02	9.9804E-02	-3.5865E+00

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
1.3000E-01	9.9748E-02	9.9748E-02	-1.0904E+01
1.4000E-01	9.9685E-02	9.9685E-02	-1.1686E+01
1.5000E-01	9.9614E-02	9.9614E-02	-9.4181E+00
1.6000E-01	9.9536E-02	9.9536E-02	-6.3151E+00

```

1.7000E-01 9.9449E-02 9.9450E-02 -3.5371E+00
1.8000E-01 9.9355E-02 9.9355E-02 -1.5210E+00
1.9000E-01 9.9251E-02 9.9251E-02 -2.8734E-01
2.0000E-01 9.9138E-02 9.9138E-02 3.3111E-01
2.1000E-01 9.9016E-02 9.9016E-02 5.4576E-01
2.2000E-01 9.8883E-02 9.8883E-02 5.3808E-01
2.3000E-01 9.8741E-02 9.8741E-02 4.3173E-01
2.4000E-01 9.8588E-02 9.8588E-02 2.9109E-01
2.5000E-01 9.8424E-02 9.8424E-02 1.3218E-01

```

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
2.6000E-01	9.8248E-02	9.8248E-02	-6.1759E-02
2.7000E-01	9.8061E-02	9.8061E-02	-3.2180E-01
2.8000E-01	9.7862E-02	9.7862E-02	-6.7078E-01
2.9000E-01	9.7650E-02	9.7650E-02	-1.0933E+00
3.0000E-01	9.7426E-02	9.7426E-02	-1.4962E+00
3.1000E-01	9.7188E-02	9.7188E-02	-1.6628E+00
3.2000E-01	9.6937E-02	9.6937E-02	-1.2105E+00
3.3000E-01	9.6672E-02	9.6672E-02	4.2011E-01
3.4000E-01	9.6393E-02	9.6393E-02	3.9040E+00
3.5000E-01	9.6099E-02	9.6099E-02	9.8330E+00
3.6000E-01	9.5790E-02	9.5790E-02	1.8309E+01
3.7000E-01	9.5466E-02	9.5466E-02	2.8270E+01
3.8000E-01	9.5127E-02	9.5126E-02	3.6543E+01

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR INITIAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
3.9000E-01	9.4771E-02	9.4770E-02	3.6777E+01
4.0000E-01	9.4398E-02	9.4398E-02	1.8620E+01
4.1000E-01	9.4009E-02	9.4009E-02	-3.2109E+01
4.2000E-01	9.3601E-02	9.3603E-02	-1.3109E+02
4.3000E-01	9.3175E-02	9.3180E-02	-2.8945E+02
4.4000E-01	9.2731E-02	9.2738E-02	-5.0227E+02
4.5000E-01	9.2268E-02	9.2279E-02	-7.3075E+02
4.6000E-01	9.1788E-02	9.1801E-02	-8.7871E+02
4.7000E-01	9.1292E-02	9.1303E-02	-7.6846E+02
4.8000E-01	9.0785E-02	9.0786E-02	-1.2737E+02
4.9000E-01	9.0270E-02	9.0250E-02	1.3944E+03
5.0000E-01	8.9753E-02	8.9692E-02	4.1447E+03

THE ENTERED DEFLECTION IS: 1.0000E-01

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
0.0000E+00	8.4445E-02	8.4502E-02	-3.8365E+03
1.0000E-02	8.4465E-02	8.4501E-02	-2.4408E+03
2.0000E-02	8.4479E-02	8.4498E-02	-1.2810E+03
3.0000E-02	8.4485E-02	8.4492E-02	-4.8428E+02
4.0000E-02	8.4484E-02	8.4484E-02	-2.5476E+01
5.0000E-02	8.4475E-02	8.4473E-02	1.8208E+02
6.0000E-02	8.4461E-02	8.4458E-02	2.3269E+02
7.0000E-02	8.4443E-02	8.4440E-02	2.0293E+02
8.0000E-02	8.4419E-02	8.4417E-02	1.4447E+02
9.0000E-02	8.4391E-02	8.4390E-02	8.6444E+01
1.0000E-01	8.4359E-02	8.4359E-02	4.1587E+01
1.1000E-01	8.4322E-02	8.4322E-02	1.2553E+01
1.2000E-01	8.4280E-02	8.4280E-02	-3.0287E+00

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
1.3000E-01	8.4233E-02	8.4233E-02	-9.2076E+00
1.4000E-01	8.4179E-02	8.4179E-02	-9.8682E+00
1.5000E-01	8.4120E-02	8.4120E-02	-7.9532E+00
1.6000E-01	8.4054E-02	8.4054E-02	-5.3328E+00
1.7000E-01	8.3981E-02	8.3981E-02	-2.9869E+00
1.8000E-01	8.3901E-02	8.3901E-02	-1.2844E+00
1.9000E-01	8.3813E-02	8.3813E-02	-2.4265E-01
2.0000E-01	8.3718E-02	8.3718E-02	2.7961E-01
2.1000E-01	8.3614E-02	8.3614E-02	4.6087E-01
2.2000E-01	8.3503E-02	8.3503E-02	4.5439E-01
2.3000E-01	8.3382E-02	8.3382E-02	3.6457E-01
2.4000E-01	8.3253E-02	8.3253E-02	2.4582E-01
2.5000E-01	8.3114E-02	8.3114E-02	1.1162E-01

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
2.6000E-01	8.2966E-02	8.2966E-02	-5.2153E-02
2.7000E-01	8.2808E-02	8.2808E-02	-2.7175E-01

```

2.8000E-01 8.2640E-02 8.2640E-02 -5.6644E-01
2.9000E-01 8.2461E-02 8.2461E-02 -9.2320E-01
3.0000E-01 8.2272E-02 8.2272E-02 -1.2635E+00
3.1000E-01 8.2071E-02 8.2071E-02 -1.4041E+00
3.2000E-01 8.1859E-02 8.1859E-02 -1.0222E+00
3.3000E-01 8.1635E-02 8.1635E-02 3.5477E-01
3.4000E-01 8.1399E-02 8.1399E-02 3.2967E+00
3.5000E-01 8.1151E-02 8.1151E-02 8.3035E+00
3.6000E-01 8.0891E-02 8.0890E-02 1.5461E+01
3.7000E-01 8.0617E-02 8.0617E-02 2.3873E+01
3.8000E-01 8.0330E-02 8.0330E-02 3.0859E+01

```

HIT <RETURN> TO CONTINUE

DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION

INTERFACE POSITION	FLANGE DEFLECTION	SKIN DEFLECTION	NORMAL STRESS
3.9000E-01	8.0030E-02	8.0029E-02	3.1056E+01
4.0000E-01	7.9715E-02	7.9715E-02	1.5724E+01
4.1000E-01	7.9386E-02	7.9387E-02	-2.7115E+01
4.2000E-01	7.9042E-02	7.9044E-02	-1.1070E+02
4.3000E-01	7.8682E-02	7.8686E-02	-2.4442E+02
4.4000E-01	7.8307E-02	7.8313E-02	-4.2415E+02
4.5000E-01	7.7916E-02	7.7925E-02	-6.1709E+02
4.6000E-01	7.7510E-02	7.7521E-02	-7.4203E+02
4.7000E-01	7.7092E-02	7.7102E-02	-6.4893E+02
4.8000E-01	7.6664E-02	7.6665E-02	-1.0756E+02
4.9000E-01	7.6229E-02	7.6212E-02	1.1775E+03
5.0000E-01	7.5792E-02	7.5741E-02	3.5000E+03

THE PREDICTED DEFLECTION AT FAILURE IS: 8.4445E-02

SECTION V

ELASTICITY METHOD (WEBSTER) USER'S MANUAL

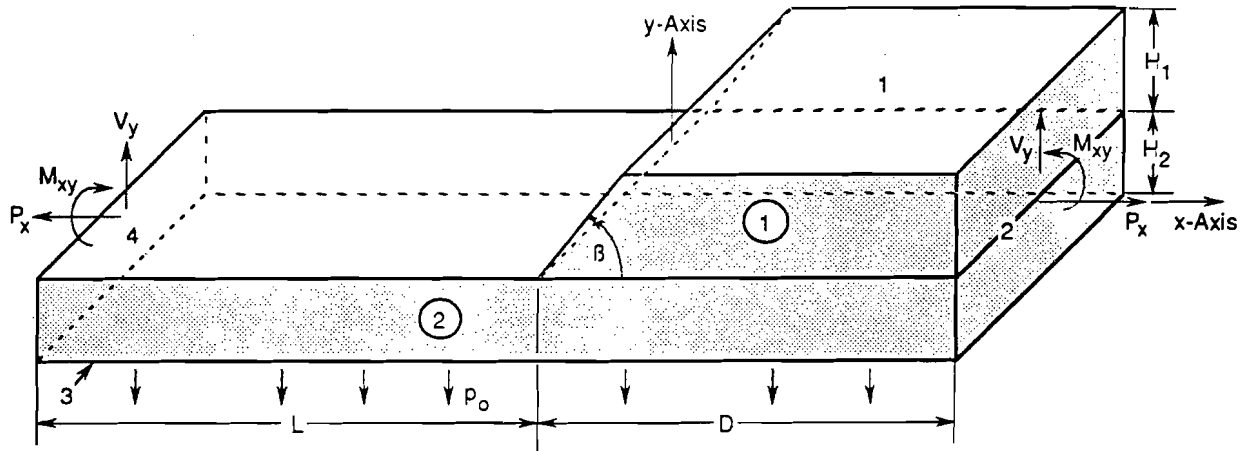
1. Introduction

1.1 Function – This method has been developed to consider the transfer of stresses between the stiffener and the adjacent structure. The analysis models the stiffener flange and the adjacent structure (skin) as orthotropic plates that are bonded together with a bondline of zero thickness (Figure 51). WEBSTER allows each of the plates to have its own mechanical properties. The solution method treats the analysis as a solution of an eigenvalue problem, and is described in more detail in the Analysis Development and Verification section of Volume I of this Final Report. Both the interlaminar normal and shear stresses are averaged over the specified characteristic lengths. The average stress values are applied to a Tsai-Hill failure criterion to assess the structure at the given loading condition.

1.2 Limitations – WEBSTER can be used interactively on VAX, IBM-PC, and IBM mainframe computers. The program will vary from computer to computer, since WEBSTER uses a few machine dependent subroutines. These differences are not evident to the user. The IMSL eigenvalue solver used in the provided FORTRAN program is most compatible with the IBM mainframe.

Due to the numerical solution of the boundary collocation method, there are certain limitations to the skin and flange lengths and the ratio of their lengths for the model. The number of eigenvalues found by the program may be insufficient for a valid stress distribution. With the number of eigenvalues found, the complexity of the solution might be too great for the operating system. All of these limits have warnings that are printed at the terminal and to the output file.

To retrieve the stresses at the collocation points and achieve a reasonable stress distribution throughout the model, neither length (D or L in Figure 51) should have a value less than 0.1, regardless of the units being used. Due to the large stress gradients which occurs at either end (Surfaces 2 and 4 of Figure 51) and the zero stress along the top surface of the model, a short length of either D or L results in erroneous stress distributions in this region and throughout the rest of the model. To retrieve the stress distribution at both ends of the model, it is necessary to limit the two lengths (D and L) so that they differ by no more than a factor of eight. A warning will appear if this is violated, since the eigenvalues determined with this error are not necessarily correct when checked at the collocation points.



GP03-0119-24-D

Figure 51. Model of the Elasticity Method/WEBSTER

The angle of stiffener termination (β) may range from 20° to 160° . A 90° angle is automatically offset by 2° to avoid a mathematical singularity. This offset was proven sufficient in the models used in the demonstration of the analysis. However, it may be necessary to input larger offsets (i.e. 87° or 93°) for right angles if the stress gradient is too high.

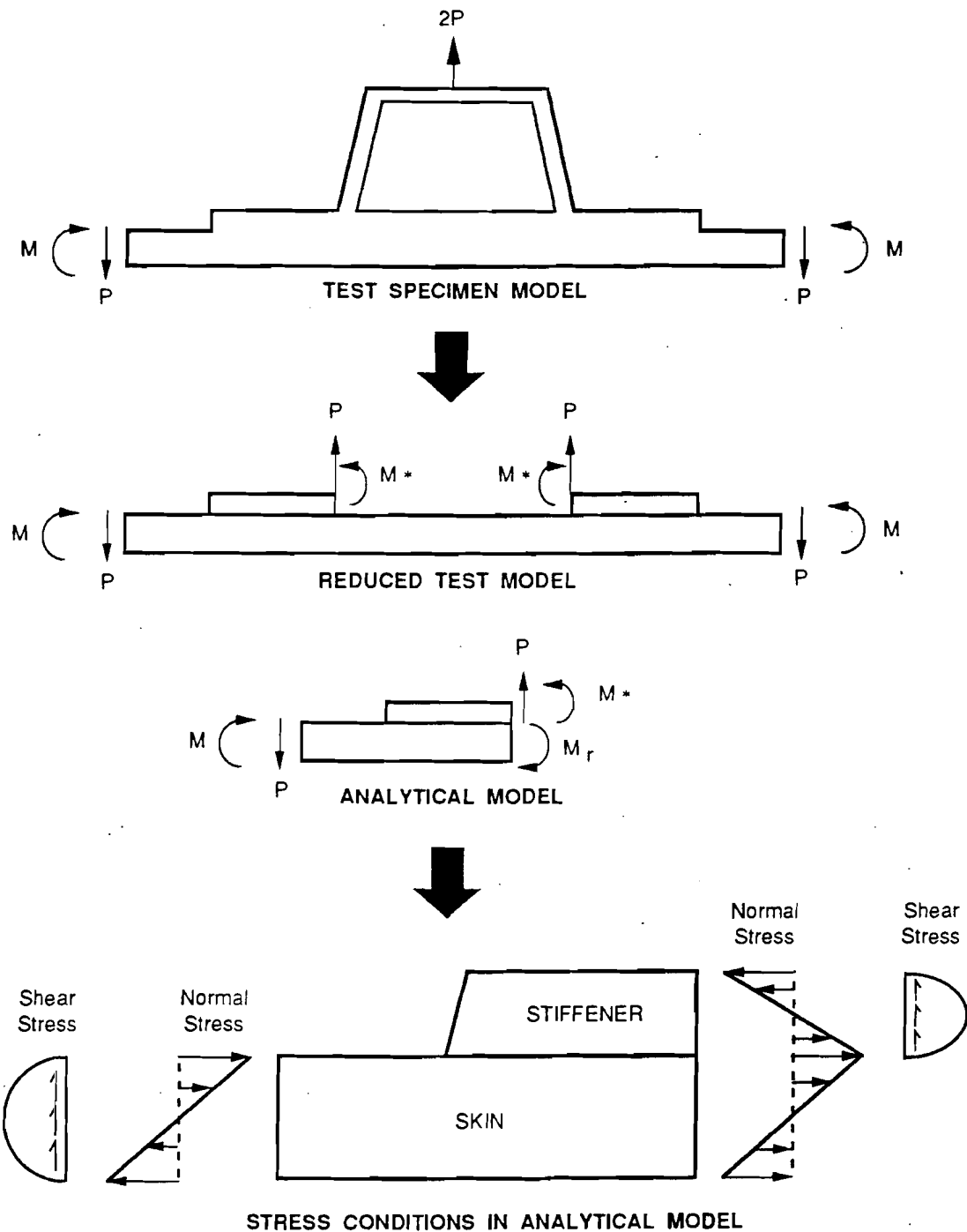
When three or less eigenvalues are found, the eigenvalue expansion is insufficient to accurately describe the stress state in the model. When four to six eigenvalues are found, the expansion series may be reasonable enough to define the stresses.

2. Input Description

WEBSTER runs interactively with each computer. The application of WEBSTER to a hat stiffener pull off configuration is shown in Figure 52. This is the same model that is used as the example problem for WEBSTER.

The structure of the model may be supplied by two methods. The first is by defining material properties, then giving the layup of both the skin and stiffener. Material properties needed are the Young's modulus in the fiber and transverse directions, in-plane shear modulus, Poisson's ratio, and lamina ply thickness. Material properties are stored in ".MAT" files that may be altered outside of the program. The program provides a listing of the material files in the working area. The skin and stiffener laminates may be hybrids or use the same material. The laminates for the skin and stiffener are generated by defining the material and orientation of each ply. The structure file, designated with the ".STR" extension, defines the resulting stiffness moduli in the x-y coordinate system. Values that are not often available (including the out-of-plane material properties and the characteristic length), are then defined. The characteristic length for a material is usually obtained from open hole computer

programs, such as BJSFM. These values are recommended for the stiffener termination failure criteria, until an improved value is available.



GP03-0119-23-D

Figure 52. Hat Stiffener Pull Off Configuration for WEBSTER

The geometric dimensions of the model are supplied by defining the length from the point of stiffener termination to the end of the "skin only" side, followed by the length of the stiffener (from the point of termination to the end of the "skin and stiffener side"), and the angle of termination of the stiffener. If a hat or blade specimen is being analyzed, additional dimensions are required in the load input section.

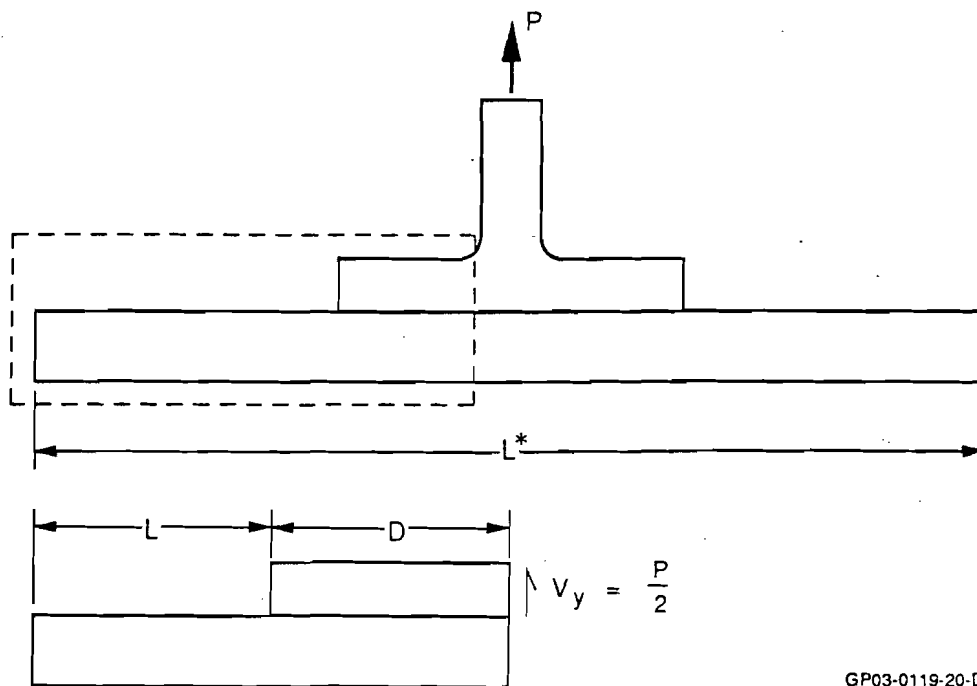
There are four types of loading conditions available for this analysis:

1. Forces applied to skin side, reactions occurring on skin/stiffener side
2. Hat stiffener pull off, clamped on the ends
3. Pressure as well as skin loading, reaction on skin/stiffener side.
4. Input each reaction stress individually

It is important to keep the pressure distributions simple and avoid sharp stress variations along the boundary. The user only needs to consider the forces acting on the model. The program assumes that the skin/stiffener edge reacts the applied forces in a constant strain manner. The load is entered as a force per unit width, except for normal pressure load cases. The program does not have an input for model width, since all forces and stresses are assumed to act uniformly through a unit width.

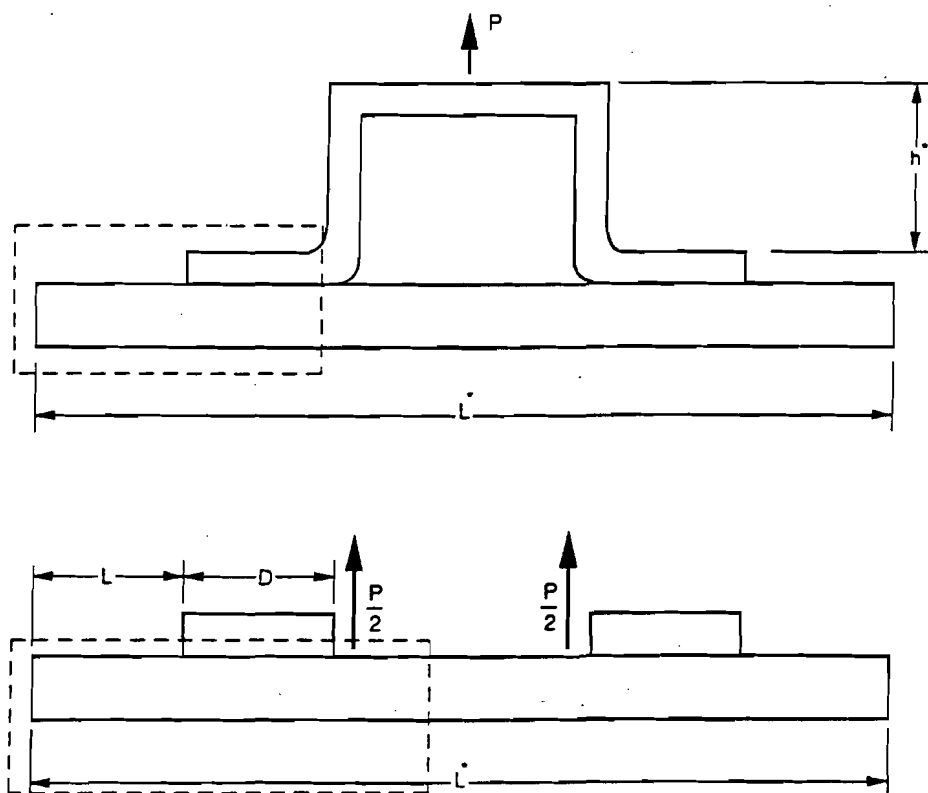
The first loading condition has a normal and shear force and bending moment acting on the neutral axis of the skin. These forces are obtained from the overall structural model. The determination of these forces is discussed in the Design Guide section of this Final Report. A zero force value is permitted for any of the applied loads. For instance, if there is only a bending moment acting on the skin, zero should be entered for the normal and shear force. The program will automatically determine the reactions acting on the neutral axis of the skin/stiffener side. Bending moments are modeled as linearly varying normal stresses acting on the edges. Axial normal forces are modeled as a uniform normal stress acting through the thickness, and proportionally on the skin and stiffener under a constant strain assumption. A shear force is modeled as a vertical parabolic stress distribution through the thickness of the edge.

The second loading condition is for blade and hat pull offs. The model assumes the skin to be clamped at each end, and a normal (vertical) load applied to the stiffener flange. Figures 53 and 54 illustrate the dimensions needed for this load condition. The input requires the overall length of the structure (the distance from one



GP03-0119-20-D

Figure 53. Model of Blade Stiffener Pull Off Specimen



GP03-0119-22-D

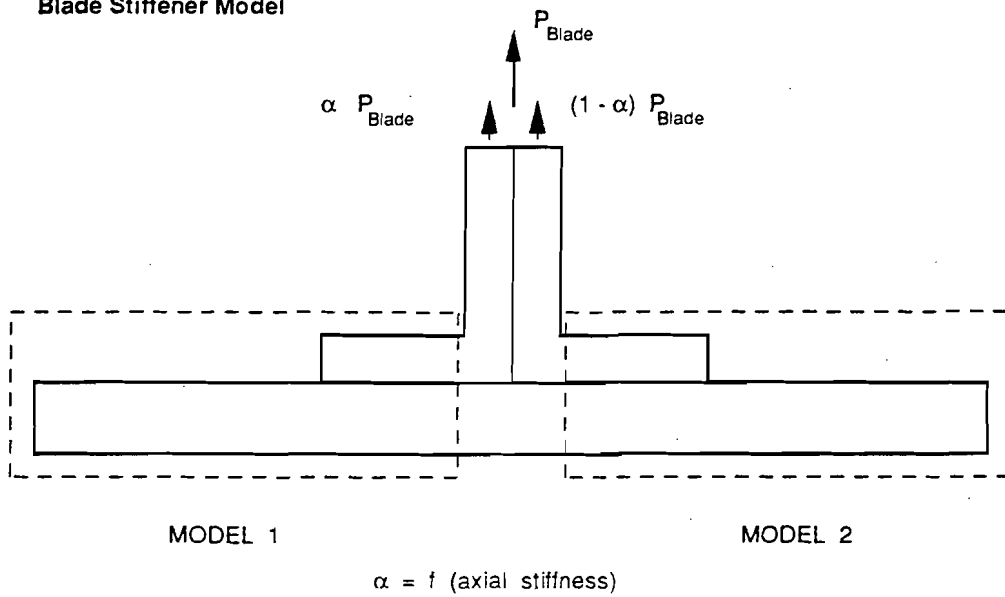
Figure 54. Model of Hat Stiffener Pull Off Specimen

clamped end to the other) and the height of the hat (the distance from the midplane of the hat flange to the midplane of the hat cap). The input load is the force per unit width that is considered applied to the flange being analyzed. The load breakdowns are shown in Figure 55. This force is proportioned with respect to the axial stiffness of each flange. For instance, a blade stiffener with identical flanges or a hat stiffener with only exterior flanges, the input load is one half the load applied to the stiffener. The input load for a hat stiffener with both exterior and interior flanges needs to be further proportioned by the axial stiffnesses of these flanges.

The third loading condition is similar to the first loading condition. A pressure loading is considered to act along the bottom of the model in addition to the forces applied to the skin. There are three options to input the pressure load. A constant or linearly varying pressure is modeled by prescribing the pressure at two points on the model. A parabolic stress distribution is modeled by identifying the locations and magnitudes of three pressures on the bottom surface, one of them being either the maximum or minimum. The final pressure distribution is a B-spline curve fit. This requires the position and magnitude of three pressure points on the bottom surface. It is considered good practice (but not essential) that the points used in the description of the pressure be within the range of the model.

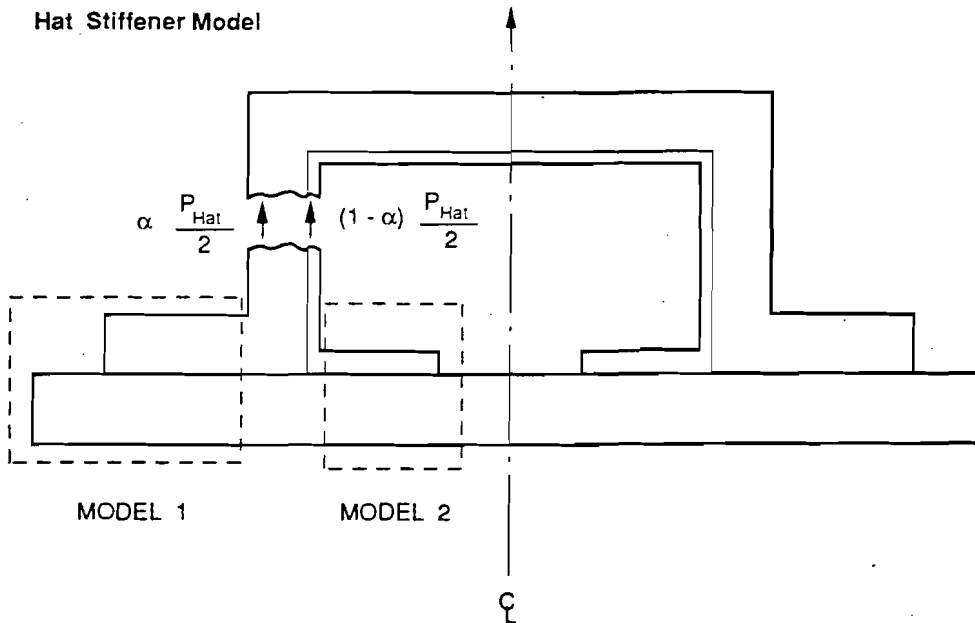
The positions given to describe the pressure distributions are with respect to the x-axis defined in Figure 51. Thus, points on the skin only side of the model are identified with negative values. The reactions for this loading condition are determined strictly on the skin/stiffener side of the model. If both ends react the pressure load, it is necessary to calculate and input the reactions of the skin only side (Surface 4 of Figure 51). In this manner, the load will not all be reacted by just one end, resulting in an incorrect loading situation. Examples of several loading conditions and how they relate to a complete structure are illustrated in the Design Guide section of this Final Report. It is advised to have some understanding of these examples prior to using the program.

The fourth loading condition allows the user to specify each individual reaction stress in the model.

Blade Stiffener Model

$$P_{\text{Model 1}} = \alpha P_{\text{Blade}}$$

$$P_{\text{Model 2}} = (1 - \alpha) P_{\text{Blade}}$$

Hat Stiffener Model

$$P_{\text{Model 1}} = \alpha \frac{P_{\text{Hat}}}{2}$$

$$P_{\text{Model 2}} = (1 - \alpha) \frac{P_{\text{Hat}}}{2}$$

GP03-0119-21-D

Figure 55. Hat and Blade Stiffener Load Reduction Technique

2.1 Errors and How to Avoid Them – WEBSTER has been written to avoid most errors. However, variances between operating systems may cause some errors to always occur. One such error is an under/overflow error where values exceed the lower or upper value limits of the system. In WEBSTER, this usually occurs during the iterative eigenvalue solution. If any intermediate value exceeds these limits during this process, the program will terminate execution. The solution to this problem is to modify the structure slightly.

The under/overflow error results from the mathematics of the problem, and is usually not due to the structural model. If the numerical problem is slightly altered, this problem may not occur. One modification is the material properties of the structure. This includes in-plane and out-of-plane material properties, and geometry values. The eigenvalues are heavily dependent upon E_x , E_y , G_{xy} , and ν_{xy} . Changing these properties by as much as 10 percent results in significant changes in the eigenvalues and stress state. The out-of-plane material properties (E_z , ν_{xz} , ν_{yz}) and the angle of stiffener termination (β) do not significantly affect the eigenvalues. Thus, increasing or decreasing E_z , ν_{xz} , or ν_{yz} from 5 to 50 percent may resolve the mathematical under/overflow problem. In parametric studies conducted, this modification did not significantly change the values of the eigenvalues nor the resulting stress state. The angle of termination may also be increased or decreased to resolve the error. However, the stress state near the point of termination might be significantly changed by this modification. The user should keep in mind that any modification to the problem results in a different problem and as a result different stresses are predicted.

3. Output Description

An output file is generated using the name of the file provided in the input section. A summary of the model is given, along with the material properties of the skin and stiffener. The average stresses are computed at a point near the stiffener termination. The hat/blade pull off model considers that a high stress region occurs under the hat/blade as well as the stiffener termination. Therefore, a failure function is computed at both locations for these loading conditions. Failure is based on the Tsai-Hill (quadratic) failure function. For the given ultimate shear and normal stresses, this failure value is calculated based on the stress values averaged over the characteristic length.

3.1 Output Warnings – Errors that generate warnings in the program will more than likely result in improper stress distributions in the model. Some of them are easily avoided and correctable by the user. Others are not so easily avoided. Warnings about the dimensions of the model are written to the terminal as well as in

the output file. However, the program continues operating with the flagged input. These warnings were discussed previously and should be avoided.

Warnings about the number of eigenvalues found by the solution routine are not as easy to avoid. The user does not have as many options to eliminate them. Having a solution with too few eigenvalues does not always mean the stresses will be erroneous. There will be times when numerical limits will only produce one to five eigenvalues for a model. There are a few variables that may increase the number of eigenvalues found without significantly altering the model. Since the out-of-plane properties are presently estimated, the values of E_z , ν_{xz} , and ν_{yz} can be increased or decreased over a range of 20 percent. This may allow more eigenvalues to be found by the IMSL routine. From parametric studies, it was shown that these parameters do not affect the stress state, but can affect the eigenvalue solution.

The angle of stiffener termination (β) may be slightly changed to increase the number of eigenvalues found. Some cases have demonstrated that the average stress is not significantly altered when the angle is varied from 85° to 95° . However, the eigenvalues found will vary slightly with this change and may result in a few more eigenvalues in the solution. Both of the remedies are not always satisfactory. Sometimes, no matter what slight modifications are made in the out-of-plane properties, the solution technique does not find a sufficient number of terms in the solution to define the stress state accurately. This problem seldom occurs on the PC and Micro-VAX and is even less likely on a mainframe. However, to date, there is no set way to avoid it nor a very clear explanation of why it occurs.

4. Example Problem

The following example illustrates the use of the WEBSTER analysis. This example is based on the model in Figure 52.

WEBSTER Example Problem Input:

```
THIS IS ***** WEBSTER *****
THE WEB-STIFFENER TERMINATION ELASTICITY ROUTINE
Developed by Northrop - Aircraft Division
Subcontractor to MCAIR - St. Louis.
Out-of-Plane Loading Navy/FAA Contract

For problems contact:
Mary Mahler (310) 332-9648 or Han-Pin Kan (310) 332-5285

WELCOME TO THE PROGRAM THAT SETS UP THE
LAMINATE PROPERTIES FOR THE SKIN AND STIFFENER.
```

THIS PART OF THE PROGRAM MAY BE BY-PASSED IF THE DESIRED PROPERTIES ARE ALREADY STORED IN A FILE FROM PREVIOUS MAKING. (IT MUST BE CORRECT, WITH ONLY CHANGES IN MATERIAL ALLOWABLES).

WILL YOU BE USING SUCH A FILE?
(IF NOT, YOU WILL BE PROMPTED TO MAKE ONE. (Y OR N)

N

BELOW ARE A LISTING OF MATERIALS THAT HAVE ALREADY DEFINED. YOU NOW HAVE THE OPTION TO ADD TO THE LIST IF YOU DESIRE OR IF DIFFERENT MATERIAL VALUES ARE NEEDED.

DO YOU WANT TO ADD TO THE LIST? (INPUT Y OR N)

ID NO.	ID NAME	EX	EY
1	AS4-3502	.9700E+07	.9700E+07

Y

FOR EACH MATERIAL SYSTEM ADDED, THE FOLLOWING INFO MUST BE SUPPLIED:

- 1) AN IDENTIFICATION NAME
 - 2) YOUNGS MODULUS (FIBER AND TRANSVERSE)--EX AND EY
 - 3) SHEAR MODULUS, POISSON RATIO, AND THICKNESS--GXY,VXY,T
- NOW ENTER THE NUMBER OF NEW MATERIALS TO BE ADDED TO THE LIST

1

FOR MATERIAL 2 INPUT AN IDENTIFYING NAME.

TAPE

FOR THE MATERIAL INPUT:

FIBER STIFFNESS, TRANSVERSE STIFFNESS, SHEAR MODULUS,
POISSONS RATIO, AND LAMINA THICKNESS.

18.7E6 1.5E6 0.78E6 0.32 0.0052

AVAILABLE MATERIALS ARE:

ID NO.	ID NAME	EX	EY
1	AS4-3502	.9700E+07	.9700E+07
2	TAPE	.1870E+08	.1500E+07

THE NEXT INPUT IS FOR THE STIFFENER.

A STIFFENER LAMINATE IS FORMED BASED ON THE FOLLOWING.

LAMINATE OPTIONS

- 1> SYMMETRICAL
 - 2> NON SYMMETRICAL
- *****

SELECT OPTION

1

ENTER TOTAL NUMBER OF LAYERS IN YOUR LAMINATE

4

INPUT MATERIAL TYPE FOR 2 LAYERS SEPARATED BY SPACES

2 2

YOUR MATERIAL TYPES ARE AS FOLLOWS :

(LAYER) MATERIAL TYPE

```

1) 2      2) 2
DO YOU WANT TO MODIFY MATERIAL TYPES (Y OR N)
N
YOUR THICKNESS FOR YOUR LAMINATES ARE :
(LAYER) THICKNESS
1) .0052      2) .0052
DO YOU WANT TO MODIFY PLY THICKNESSES (Y OR N)
N
INPUT ORIENTATION FOR 2 LAYERS SEPARATED BY A SPAC E
(ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE)
45 -45
PLY ORIENTATIONS ARE "(LAYER) ORIENTATION"
* (ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE) *
1) 45.00      2) -45.00
DO YOU WANT TO MODIFY PLY ORIENTATIONS (Y OR N)
N
THE X-DIRECTIONAL STIFFNESS IS      2721950.00000
THE Y-DIRECTIONAL STIFFNESS IS      2721950.00000
THE INPLANE SHEAR MODULUS IS      4849836.00000
THE POISSONS RATIO IS      .74484
THE STIFFENERS THICKNESS IS      .02080
DO YOU WANT ANY CHANGES TO THE STIFFENER LAMINATE?
N
INPUT OUT-OF-PLANE YOUNGS MODULUS (Ez)
1.5E6
THE NEXT INPUT IS FOR THE SKIN.
A SKIN LAMINATE IS FORMED BASED ON THE FOLLOWING.
      1                      AS4-3502          .9700E+07      .9700E+07
      2                      TAPE
.1870E+08 .1500E+07
      LAMINATE OPTIONS
      *****
      1> SYMMETRICAL
      2> NON SYMMETRICAL
      *****
      SELECT OPTION
1
ENTER TOTAL NUMBER OF LAYERS IN YOUR LAMINATE
16
INPUT MATERIAL TYPE FOR 8 LAYERS SEPARATED BY SPACES
2 2 2 2 2 2 2 2
YOUR MATERIAL TYPES ARE AS FOLLOWS :
(LAYER) MATERIAL TYPE
1) 2      2) 2      3) 2      4) 2      5) 2      6) 2
7) 2      8) 2
DO YOU WANT TO MODIFY MATERIAL TYPES (Y OR N)
N
YOUR THICKNESS FOR YOUR LAMINATES ARE :
(LAYER) THICKNESS

```


1) .0052 2) .0052 3) .0052 4) .0052 5) .0052 6) .0052
7) .0052 8) .0052

DO YOU WANT TO MODIFY PLY THICKNESSES (Y OR N)

N

INPUT ORIENTATION FOR 8 LAYERS SEPARATED BY A SPACE
(ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE)

45 -45 0 90 45 -45 90 0

PLY ORIENTATIONS ARE "(LAYER) ORIENTATION"

* (ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE) *

1) 45.00 2) -45.00 3) .00 4) 90.00 5) 45.00 6) -45.00

7) 90.00 8) .00

DO YOU WANT TO MODIFY PLY ORIENTATIONS (Y OR N)

N

THE X-DIRECTIONAL STIFFNESS IS 7370102.00000

THE Y-DIRECTIONAL STIFFNESS IS 7370102.00000

THE INPLANE SHEAR MODULUS IS 2814918.00000

THE POISSONS RATIO IS .30911

THE SKIN THICKNESS IS .08320

DO YOU WANT CHANGES TO THE SKIN LAMINATE?(Y OR N)

N

INPUT AVERAGE LENGTH FOR NORMAL STRESS

CONCENTRATION INTEGRATION, SHEAR STRESS CONCENTRATION LENGTH
AS WELL AS THE ULTIMATE INTERLAMINAR NORMAL AND SHEAR
STRESS VALUES.

0.15 0.15 3000 11000

THE PARAMETERS THAT MAY BE CHANGED AT
THIS TIME IS: THE INTERLAMINAR NORMAL AND SHEAR
STRESS ALLOWABLES AND THE CHARACTERISTIC LENGTHS
ASSOCIATED WITH AVERAGE NORMAL AND SHEAR STRESSES.

THE CURRENT VALUES FOR THESE PARAMETERS ARE:

NORMAL STRESS ALLOWABLE = 3000.00000

SHEAR STRESS ALLOWABLE = 11000.00000

AVERAGE NORMAL STRESS CHARACTERISTIC LENGTH = .150000

AVERAGE SHEAR STRESS CHARACTERISTIC LENGTH = .150000

DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y OR N)?

N

THE INFO. FROM THIS INPUT MAY BE STORED.

THIS IS STRONGLY RECOMMENDED UNLESS YOU HAVE JUST
RETRIEVED THE INFORMATION AND DO NOT WANT TO RE-WRITE
THE FILE. WOULD YOU LIKE TO STORE THIS INFO.? (Y OR N)

Y

INPUT THE NAME OF THE FILE YOU WOULD LIKE THE
DATA STORED UNDER. THE FILENAME MUST BE 8 LETTERS OR
LESS. THE EXTENSION ".STR" IS GIVEN TO ALL DATA FILES.
IF A PREVIOUS FILE ALREADY EXISTS WITH THIS NAME, THE
FILE WILL BE OVER-WRITTEN. (THE OLD ONE WILL BE LOST)

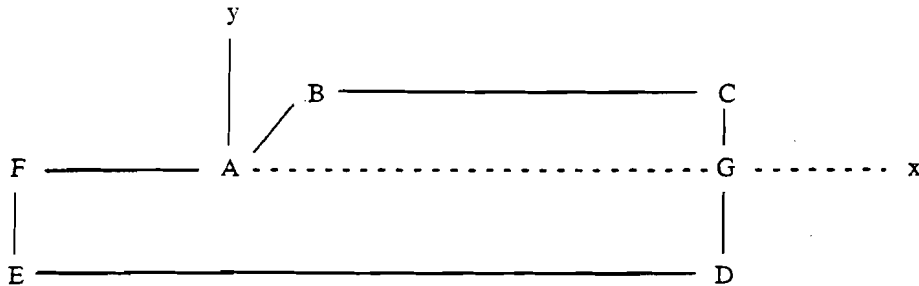
EXAMPLE

THE DATA HAS BEEN SAVED UNDER EXAMPLE .STR.

INPUT THE NAME OF THE OUTPUT FILE

EXAMPLE

The geometry of the structure is defined as follows:
The origin of the coordinate system is at point A.



L (distance from F to A), D (distance from A to G),
ANGLE (angle from the x-axis to line A-B (in degrees)):
Input L, D, ANGLE ?

0.6 0.7 90

THE FOLLOWING IS A DESCRIPTION OF THE LOADING
CONDITIONS AVAILABLE. THERE ARE 4 TYPES OF INPUT/
REACTION LOADS AVAILABLE. INPUT THE NUMBER DESIRED.

- 1) FORCES APPLIED TO SKIN SIDE, REACTIONS OCCURING ON SKIN/
STIFFENER SIDE.
 - 2) HAT-STIFFENER PULL-OFF, CLAMPED ON THE ENDS.
 - 3) PRESSURE AS WELL AS SKIN LOADING, REACTION ON
SKIN/STIFFENER SIDE.
 - 4) INPUT EACH REACTION STRESS INDIVIDUALLY
- INPUT 1, 2, 3, OR 4 (NSIDE)

2

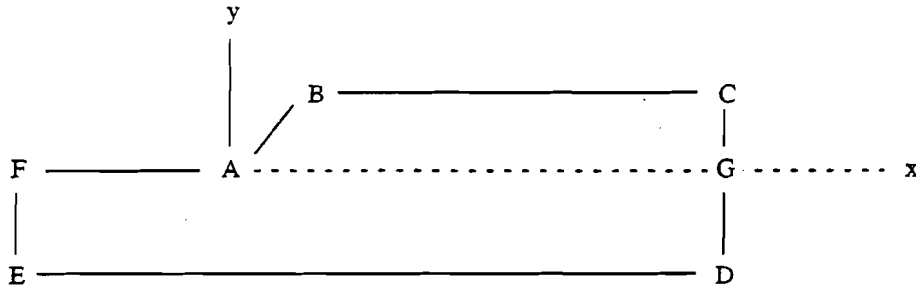
FOR THE HAT-PULLOFF SPECIMEN, INPUT THE FOLLOWING
FORCE APPLIED TO THE HAT, TOTAL LENGTH OF SPECIMEN, AND HEIGHT OF
HAT.

30 3.5 1.368

Stop - Program terminated.

NOTE: This program takes about 25 minutes to run on a 386-based PC.

WEBSTER Example Problem Output:



The length of the skin (F to A) is .60000
 The length of the stiffener (A to G) is .70000
 The thickness of the stiffener (G to C) is .02080
 The thickness of the skin (G to D) is .08320
 The angle surface A-B makes with the X-axis is 90.000 degrees.
 This run considers 29 collocation points.
 It uses 17 eigenvalues.

The model pull-off force is 30.000
 Total length of the structure is 3.500
 Height of the hat is 1.368

For an A0 (Normal) of .1500
 For an A0 (Shear) cf .1500
 GIVEN ULTIMATE NORMAL STRESS = 3000.00
 ULTIMATE SHEAR STRESS = 11000.00

Material properties of the stiffner and skin are:

	EX	EY	EZ
STIFFENER =	.2722E+07	.1500E+07	.2722E+07
SKIN =	.7370E+07	.1500E+07	.7370E+07

	GXZ
STIFFENER =	.4850E+07
SKIN =	.2815E+07

	VYZ	VXZ	VXY
STIFFENER =	.3000E+00	.7448E+00	.3000E+00
SKIN =	.3000E+00	.3091E+00	.3000E+00

NEAR THE POINT OF TERMINATION: AT POINT A:
 THE AVERAGE NORMAL STRESS IS 22.88
 THE AVERAGE SHEAR STRESS IS 1409.77
 FAILURE FUNCTION (QUADRATIC) IS .1284

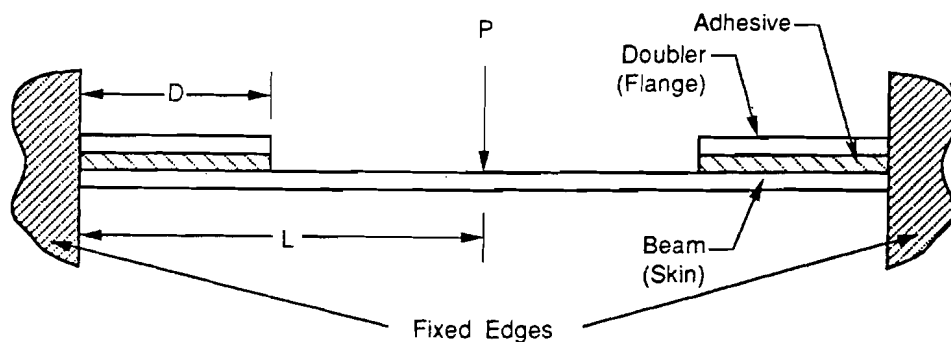
LENGTHS: A0N, A0S, A0N2, KCT 1.500000E-01 1.500000E-01
 7.950002E-02
 53

NEAR THE C-D SURFACE: AT POINT G:
 THE AVERAGE NORMAL STRESS IS 1307.18
 THE AVERAGE SHEAR STRESS IS 2647.79
 FAILURE FUNCTION (QUADRATIC) IS .4978

SECTION VI RAYLEIGH-RITZ METHOD (BEAM) USER'S MANUAL

1. Introduction

1.1 Function – The BEAM method assesses the stress state at stiffener flange terminations for structures under pure normal loads. This method is based on a Rayleigh-Ritz solution, and is described in more detail in Section IV of Volume I of this Final Report. The model, as illustrated in Figure 56, consists of a skin and stiffener flanges joined by an elastic adhesive layer. The skin and flanges are considered to be orthotropic plates. The solution provides the displacements of the skin and stiffener flange. The solution accounts for both the out-of-plane (z) and in-plane (x) displacements, and provides the interlaminar normal and shear stress distributions, in addition to the average stresses at the flange termination. The analysis is sensitive to the modeling of the adhesive layer, which is difficult to define in actual joints.



GP03-0119-26-D

Figure 56. Model of the Rayleigh-Ritz Method/BEAM

The BEAM analysis is a series of three computer programs. The first program requests the input data. The input information is transferred from one program to the next by the use of data sets.

1.2 Limitations – The BEAM analysis has several limitations. The analysis only predicts the stress state at the flange termination. It does not consider the stresses at the fixed end of the adhesive joint. This limits the analysis to situations where failure is expected at the flange termination.

This analysis is also limited in its single loading conditions. Only a vertical load may be applied at the mid-span of the skin. The functions used in the solution method account for the reactions at the clamped ends and for bending, and assume the flange and skin do not see relative motion at the clamped end. The program uses an incremental load solution since this is a nonlinear problem. It initially considers a unit load, and increments the

applied load for the solution up to the desired load. Since the load must be an integer value, interpolation should be used for noninteger load values. It is recommended to keep the load magnitude under 150, since the step load iteration process is time consuming process.

The properties of the adhesive are difficult to determine in most cases, and therefore become another limitation to using this analysis. BEAM considers the adhesive properties to be known material properties, though they are seldom found as such in reality. This part of the model is not well defined in actual structures. Most of the properties used for the adhesive joint are based on correlation to test data. The thickness of the adhesive has been shown to have a noticeable effect on the stress predictions, and must be a nonzero value due to the nature of the analysis.

2. Input Description

The model was created to allow for the material properties to change along the length of the skin and the flange. This allows the modeling of geometric details such as ply drop offs. The program inputs are described below. Figure 57 should be referenced for further assistance.

1. Model Geometry

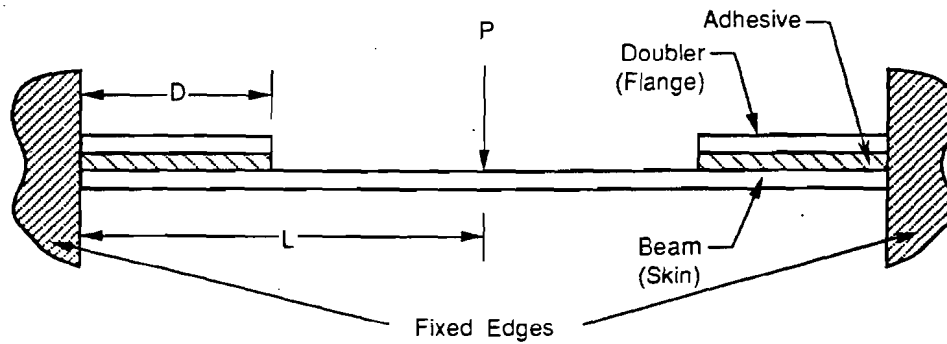
- XL (L), 1/2 the total skin (beam) length
- D, flange (doubler) length

2. Number of Steps

- NSL, number of steps in beam along L
- NSD, number of steps in doubler along D

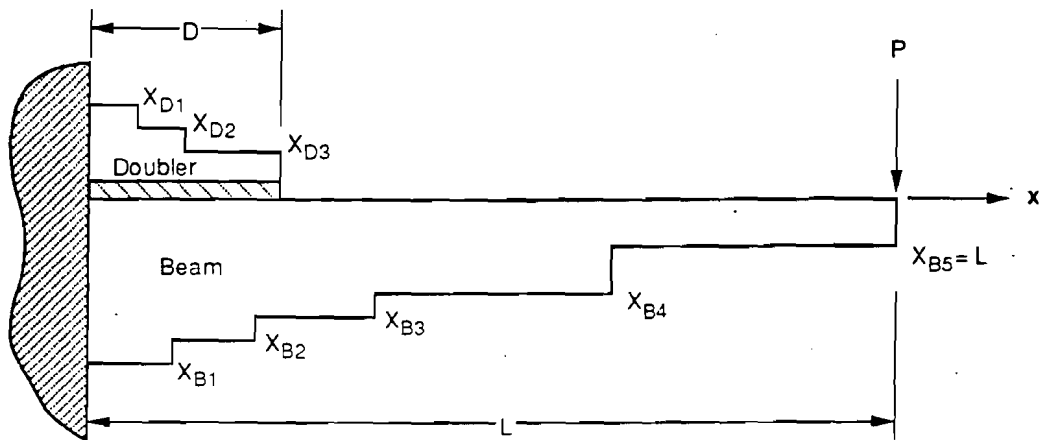
3. Beam Step Information – Enter this information for each step.

- SE1, end location of beam step i (last one should be at $x = L$)
- ET1, axial stiffness of beam step i per unit width ($E \times t$)
- EI1, bending stiffness of beam step i per unit width ($E \times I$)



Doubler Input :

$EI_{D1}, Et_{D1}, X_{D1}, EI_{D2}, Et_{D2}, X_{D2}, \dots$



Beam Input :

$EI_{B1}, Et_{B1}, X_{B1}, EI_{B2}, Et_{B2}, X_{B2}, \dots$

- Doubler has 3 changes (steps) in stiffness and/or thickness
- Beam has 5 changes (steps) in stiffness and/or thickness

GP03-0119-25-D

Figure 57. Geometry Description of BEAM Model

4. Doubler Step Information – Enter this information for each step.

- SE2, end location of doubler step i (last one should be at $x = L$)
- ET2, axial stiffness of doubler step i per unit width ($E \times t$)
- EI2, bending stiffness of doubler step i per unit width ($E \times I$)

5. Load Information

- NPMX (P), normal load applied at $x = L$ (must be an integer value)

6. Adhesive Information

- EX, Young's Modulus
- GXZ, shear modulus
- T, adhesive thickness
- SIGU, ultimate tensile strength
- TAUU, ultimate shear strength
- AO, characteristic length for average stress calculations

7. Program Completion Commands – The first portion of the program is complete at this point. The following commands should be entered to complete the analysis.

- MCDTWO: to complete the second portion of the analysis
- MCDTHREE: to complete the third portion of the analysis

3. Output Description

The output to BEAM is similar to that of WEBSTER. The objective of both programs is to predict failure at the stiffener termination. Therefore, the results of both analyses reflect this failure prediction. The first part of the output echoes the input and defines the problem. Following the echo of the input, the stress predictions are given. The output defines the stress components, failure function and displacement at several locations. The stress and failure function over the average characteristic length are also given.

For this program, there is only one form of output, that being given at the end of the subprogram "MCDTHREE." Intermediate data files used in the analysis can only be interpreted by the three subprograms within the analysis. All three subprograms must be run to achieve the final stress results.

The solution identifies the corresponding displacement of the structure as well as the stresses at the interface. This displacement may be needed to find the load associated with a given displacement of a structural member. This may be important when considering buckling or large known displacements of a skin panel.

4. Example Problem

The following example illustrates the use of the BEAM analysis.

BEAM Example Problem Input:

C:> MCDONE

This is program one of a three part series.
It simulates failure of stiffener/web separation.

Please input 1/2 the beam's length and
the length of the doubler. (XL, D)

6, 1

Input the number of steps in 1/2 of the beam
and the number of steps in the doubler. (NSL, NSD)
(The maximum of each is 15.)

2, 1

Input the x-coordinates of the beam steps.
Include the extensional and bending stiffness per
unit width for the step which ends at this point.
Only one step per line. The last one at $x = L/2$.
There should be -NSL- inputs. (SE1, ET1, EI1)

1.0 665000.0 400.0

6.0 581000.0 198.0

Input the x-coordinates of the doubler steps.
Include the extensional and bending stiffness per
unit width for the step which ends at this point.
Only one step per line with the last one at $x = D$.
There should be -NSD- inputs. (SE2, ET2, EI2)

1.0 290000.0 540.0

Please input the final load applied
to the structure. (NPMX)

10

Input the Young and Shear Modulus of adhesive,
the adhesive thickness, the ultimate tensile (peel)
and shear strength of the adhesive, and the average
characteristic length of the average stress.
(EX, GXZ, T, SIGU, TAUU, A0)

640000.0 217000.0 0.0015 3500.0 11000.0 0.15

The first portion of the program is finished.

The next step is to execute the second program. To do this type in : MCDTWO

Stop - Program terminated.

C:> MCDTWO

```

P =          1.00000000
P AND FXX      1.00000000 -1.461268E-002
P =          2.00000000
P AND FXX      2.00000000 -4.057334E-002
P =          3.00000000
P AND FXX      3.00000000 -6.653486E-002
P =          4.00000000
P AND FXX      4.00000000 -9.249476E-002
P =          5.00000000
P AND FXX      5.00000000 -1.184561E-001
P =          6.00000000
P =          7.00000000
P AND FXX      7.00000000 -4.318435E-001
P =          8.00000000
P AND FXX      8.00000000 -5.531115E-001
P =          9.00000000
P AND FXX      9.00000000 -6.468754E-001
P =         10.00000000
P AND FXX      10.00000000 -7.405267E-001

```

This is the end of the second program.

To finish executing the program - type: MCDTHREE

Stop - Program terminated.

C:> MCDTHREE

What name do you want the output listed under?
example.out

This information is also saved under a file called: example.out

```

Half length of the beam is      6.000
Length of the Doubler is       1.000
Adhesive thickness is          .00150

Youngs Modulus of the Adhesive is 640000.00
Shear Modulus of the Adhesive is  217000.00
Ultimate Peel (Normal) stress is  3500.00
Ultimate Shear Stress is         11000.00
Averager Characteristic Length is .15000

The beam is divided into  2 step(s)
The doubler is divided into 1 step(s)

```

The step(s) for the beam are at:

```

X =      1.00 ET = .665000E+06 EI = .400000E+03
X =      6.00 ET = .581000E+06 EI = .198000E+03

```

While the doubler has steps at:

```

X =      1.00 ET = .290000E+06 EI = .540000E+03

```

```

For an applied load of      1.00
Failure Function over a(0)   .00894

```

For an applied load of	2.00
Failure Function over a(0)	.00985
For an applied load of	3.00
Failure Function over a(0)	.00960
For an applied load of	4.00
Failure Function over a(0)	.00924
For an applied load of	5.00
Failure Function over a(0)	.00892
For an applied load of	7.00
Failure Function over a(0)	.04989
For an applied load of	8.00
Failure Function over a(0)	.05234
For an applied load of	9.00
Failure Function over a(0)	.05365
For an applied load of	10.00
Failure Function over a(0)	.05357

THIS IS THE END OF THE PROGRAM.

Stop - Program terminated.

BEAM Example Problem Output:

Half length of the beam is	6.000
Length of the Doubler is	1.000
Adhesive thickness is	.00150
Youngs Modulus of the Adhesive is	640000.00
Shear Modulus of the Adhesive is	217000.00
Ultimate Peel (Normal) stress is	3500.00
Ultimate Shear Stress is	11000.00
Averager Characteristic Length is	.15000

The beam is divided into 2 step(s)
The doubler is divided into 1 step(s)

The step(s) for the beam are at:

X =	1.00	ET =	.665000E+06	EI =	.400000E+03
X =	6.00	ET =	.581000E+06	EI =	.198000E+03

While the doubler has steps at:

X =	1.00	ET =	.290000E+06	EI =	.540000E+03
-----	------	------	-------------	------	-------------

For an applied load of	1.00
Failure Function over a(0)	.00894

X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.
6.00				-.02596079
1.00	323.9842	210.6196	.0089353	-.00069866

For an applied load of	2.00			
Failure Function over a(0)	.00985			
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.

NAWCADWAR-94138-60 (Volume II)

Report No. NADC
Volume II

6.00					-.02596079
1.00	340.0052	223.5557	.0098501		-.00069867
For an applied load of 3.00					
Failure Function over a(0) .00960					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.02596078		
1.00	335.4787	223.3118	.0095996	-.00069867	
For an applied load of 4.00					
Failure Function over a(0) .00924					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.02596079		
1.00	329.1619	218.2808	.0092385	-.00069867	
For an applied load of 5.00					
Failure Function over a(0) .00892					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.02596080		
1.00	323.2360	216.8400	.0089177	-.00069868	
For an applied load of 7.00					
Failure Function over a(0) .04989					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.08214336		
1.00	732.3618	859.2777	.0498861	-.00302763	
For an applied load of 8.00					
Failure Function over a(0) .05234					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.09371635		
1.00	712.6872	1147.1150	.0523381	-.00407804	
For an applied load of 9.00					
Failure Function over a(0) .05365					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.09371636		
1.00	720.8723	1165.8240	.0536536	-.00407804	
For an applied load of 10.00					
Failure Function over a(0) .05357					
X-location	Normal Stress	Shear Stress	Failure Function	Z-Displace.	
6.00			-.09371635		
1.00	719.7894	1167.9370	.0535670	-.00407804	

SECTION VII

SIMPLIFIED STIFFENER ANALYSIS MODEL (SSAM) USER'S MANUAL

1. Introduction

The Simplified Stiffener Analysis Model (SSAM) is an automated finite element program which uses frame and membrane elements to model a skin and cocured or adhesively bonded stiffener. The purpose of the program is to analyze the stress state in the interface between the skin and stiffener. The complete SSAM system is composed of two programs: a model generation program (SSAMGEN) which performs the preprocessing tasks and the main finite element analysis program (SSAM).

1.1 Function – The Simplified Stiffener Analysis Model (SSAM) is a simplified finite element model for analyzing the pull-off stresses in a cocured or adhesively-bonded skin/stiffener joint. A model of the skin and stiffener is generated using the SSAMGEN model generator and the analysis is performed by the SSAM finite element program. Any combination of loads, moments and boundary conditions can be specified on the skin or stiffener.

SSAM uses plane frame (ie 2-D beam) finite elements to model the skin and stiffener, and membrane elements are used to model the cocured or adhesively-bonded interface between the skin and stiffener flange(s). Truss elements are used to model any stitching which may be present. In order to account for nonlinearities which can be present in a postbuckling analysis, the frame elements are formulated to include the reduction in bending stiffness due to an axial compressive force.

The SSAM program calculates the distributions of normal and shear stress in the interface and applies an interactive failure criterion to determine the strength of the interface.

1.2 Limitations – SSAM assumes that the skin and stiffener flange can be idealized as beams which are connected by a common interface. This assumption is valid when the skin and flange stiffnesses are larger than the stiffness of the interface. This assumption allows computationally efficient plane frame finite elements to be used to model the skin and stiffener.

2. Description of SSAM Program

2.1 SSAM Program Input – The SSAM program assumes that a finite element model has already been generated and is stored in a text file. The required format of this file is described in the following paragraph.

The input file for SSAM contains all of the finite element mesh data which is needed in order to run the analysis. The file must be written in a specific format as follows:

DESCR = Alphanumeric mesh title

NBSK,NBFL,NPLATE,NUMBC,	NBSK = # of skin elements
NUMLDS,NSTITCH	NBFL = # of stiffener elements
	NPLATE = # of membrane elements
	NUMBC = # of restrained nodes
	NUMLDS = # of loaded nodes
	NSTITCH = # of truss elements

IEL,N1,N2,EI,EA

IEL = Element number
N1 = First node number
N2 = Second node number
EI = Bending stiffness
EA = Inplane stiffness

IEL,N1,N2,N3,N4,	IEL = Element number
PE,PG,SIG,TAU	N1 = First node
	N2 = Second node
	N3 = Third node
	N4 = Fourth node
	PE = Normal stiffness
	PG = Shear stiffness
	SIG = Normal stress allowable
	TAU = Shear stress allowable

Section E: Nodal Point Definitions (2I4,2e12.5)
one card for each node

N,NADJ,X,Y

N = Node number
NADJ = Node adjacent to N
(0 if no adjacent node exists)
X = X-coordinate
Y = Y-coordinate

Section F: Restrained Boundary Node Definitions (4I4,3E12.5)
one card per restrained node; only frame element
nodes may be restrained

N,IU,IV,IS,UDISP,VDISP,
SLOPE

N = Node number
IU = U-displacement flag
IU=0 if free, 1 if fixed
IV = V-displacement flag
IV=0 if free, 1 if fixed
IS = SLOPE displacement flag
IS=0 if free, 1 if fixed
UDISP = Fixed U-displacement
VDISP = Fixed V-displacement
SLOPE = Fixed slope

Section G: Nodal Point Load Definitions (I4,3E12.5)
one card per loaded node; only frame element
nodes may have loads applied

N,FX,FY,MOMENT

N = Node number
FX = Force in X-direction
FY = Force in Y-direction
MOMENT = Applied bending moment

Section H: Truss Element Definition (3I4,E12.5)
one card per truss element; truss elements connect a
frame element node on the skin to a frame element
node on the flange.

IEL,N1,N2,EA

IEL = Element number
N1 = First node (on skin)
N2 = Second node (on stiffener)
EA = EA of the element

2.2 SSAM Program Output – SSAM generates five output files. The first file contains tabular results of the analysis, and the remaining four files contain selected results which are written in a specific format for use with a general plotting routine. The five files are described below:

- 1) The analysis results are written in tabular form to the output file which was specified by the user. This file includes an echo of the input data, nodal point displacements, element force resultants, stress distributions in the interface and a strength analysis of the interface.
- 2) DISP.DAT – This file contains the undeformed and deformed shapes of the skin and stiffener. The deformed shape is generated by magnifying the nodal point displacements. The magnification factor used will be displayed in DISP.DAT.
- 3) NORM.DAT – This file contains the distribution of normal stress in the interface.
- 4) SHEAR.DAT – This file contains the distribution of shear stress in the interface.
- 5) FAIL.DAT – This file contains the failure function, k , along the interface. The failure function is calculated using an interactive failure criterion. Failure occurs if $k \geq 1$.

3. Description of SSAMGEN Program

SSAMGEN is a finite element model generation program specifically tailored to produce input data files for SSAM. This program greatly simplifies the stiffener analysis process by automatically generating a finite element mesh which is consistent with the assumptions made by the SSAM program. SSAMGEN generates this model from a minimum number of inputs by the user. The model generated by SSAMGEN is also optimized to produce a small stiffness matrix bandwidth, thereby allowing the SSAM program to run as quickly and efficiently as possible.

3.1 SSAMGEN Program Inputs – A detailed finite element model of a typical hat stiffened panel is shown in Figure 58 along with a typical SSAM model which could be generated to study this panel. The required geometric and loading input data for SSAMGEN to generate such a model is shown in Figure 59. Along with this data, SSAMGEN requires the axial and bending stiffnesses of the skin and stiffener components, and the normal and shear stiffnesses and strengths of the interface. The location and stiffness of any stitching is also required.

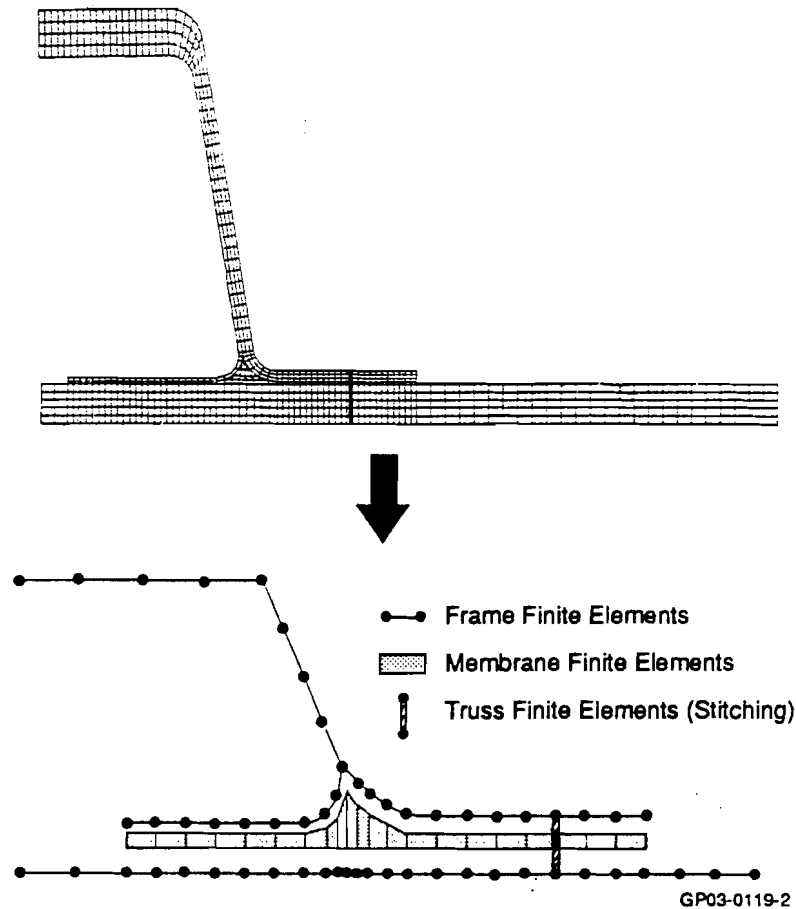
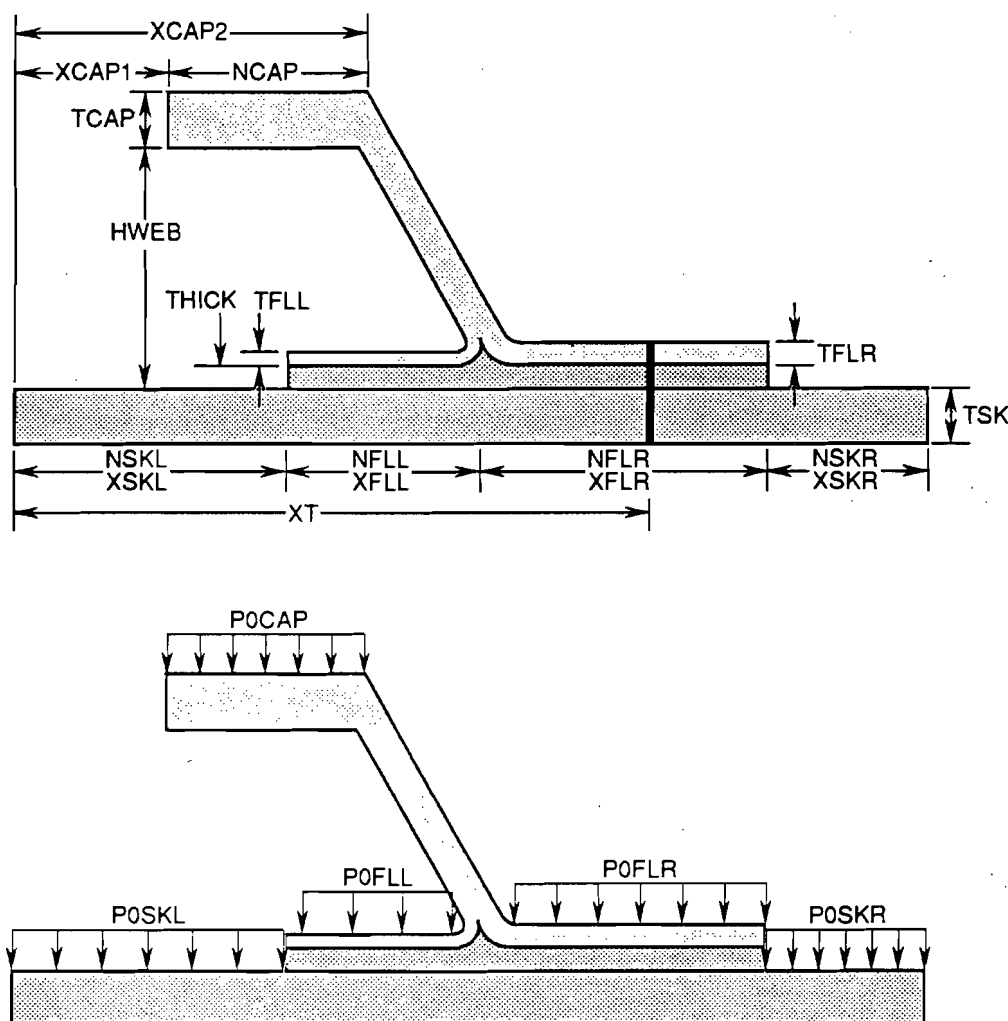


Figure 58. The Simplified Stiffener Analysis Model (SSAM)

SSAMGEN is capable of generating models for any stiffener which has a web and one or two flanges. An optional cap can be added to the top of the web. The geometric input data shown in Figure 59 is all that is required to simulate a stiffener and skin. A few ground rules apply:

- 1) If a certain section of the stiffener does not exist, this is modeled by simply setting the number of elements on that section equal to zero. For example, if the stiffener does not have a flange on the right-hand side of the web, then set the variable NFLR=0. All other variables associated with that flange will be ignored.
- 2) The flange/web inner moldline radius variable, RAD, takes on different meanings depending on the stiffener geometry. There are two possible cases:
 - a) If the stiffener has only one flange, then RAD is the inner moldline radius of that flange/web corner



GP03-0119-41-D

Figure 59. SSAMGEN Geometric and Loading Input Parameters

- b) If the stiffener has two flanges then RAD is the inner moldline radius of the side which makes the larger angle between the flange and web. For example, for the stiffener shown in Figure 59 the variable RAD would be the corner radius of the web and right-hand flange. For the special case where the web is vertical (as in a blade stiffener where the web makes a 90° angle with both flanges), then RAD is the radius of the corner to the right of the web.
- 3) SSAM allows the displacements and/or rotation to be fixed at any specified values at any point on the skin and stiffener. SSAMGEN will automatically apply boundary conditions at the two ends of the skin and at the end of the stiffener cap (or at the end of the web if the stiffener has no cap). No rotation or x-displacement will be allowed at these points and no vertical displacement will be allowed at the right-hand edge of the skin. Extra boundary conditions will be applied if the flange(s) run the entire length of the skin (i.e. if

NSKL=0 and/or NSKR=0). If boundary conditions other than these are needed the user must modify the mesh file manually before running SSAM.

Upon entry, SSAMGEN will prompt the user for the name of a file to be used to store the generated finite element mesh. After opening this file SSAMGEN will look in the user's default directory for a file named SSAMGEN.INP. If this file exists, SSAMGEN will attempt to read all of the input information from that file. If SSAMGEN.INP does not exist or does not contain the expected information, SSAMGEN will prompt the user for all inputs. The resulting mesh data file is ready to use with the SSAM program. The mesh data file is a text file which can be edited in order to make any desired modifications to the input.

3.2 Description of SSAMGEN Input Variables – The following table describes the input variables for SSAMGEN. These variables can either be typed in at the prompt or entered into a file named SSAMGEN.INP. A sample SSAMGEN.INP file is shown in Section 5 along with the resulting SSAM model and analysis output.

Variables which are grouped together below should be on the same line in the SSAMGEN.INP file.

Variable Names	Variable Descriptions
DESCR	DESCR = model description (up to 78 characters)
NSKR,NFLR,NFLL,NSKL, NWEB,NCAP	NSKR = # of elements in skin to the right of the stiffener NFLR = # of elements in the flange which is to the right of the web (usually $20 < NFLR < 50$ works well) NFLL = # of elements in the flange which is to the left of the web (usually $20 < NFLL < 50$ works) NSKL = # of elements in the skin to the left of the stiffener NWEB = # of elements in the web NCAP = # of elements in the cap
XSKR,XFLR,XFLL,XSKL, XCAP1,XCAP2,HWEB	XSKR = length of skin to the right of the stiffener (not used if NSKR=0) XFLR = length of flange to the right of the web (not used if NFLR=0) XFLL = length of flange to the left of the web (not used if NFLL=0) XSKL = length of skin to the left of the stiffener (not used if NSKL=0) XCAP1 = X-coordinate of the edge of the cap (not used if NCAP=0) XCAP2 = X-coordinate of the web/cap intersection (if NCAP=0 then this is

the X-coordinate of the end of the web)
HWEB = height of web measured from the
top of the skin

EIFLR,EIFLL,EIWEB,
EICAP,EISK

EIFLR = EI of flange to the right
of the web (not used if NFLR=0)
EIFLL = EI of flange to the left of the
web (not used if NFLL=0)
EIWEB = EI of the web
EICAP = EI of the cap (not used if
NCAP=0)
EISK = EI of the skin

EAFLR,EAFLLE,EAWEB,
EACAP,EASK

EAFLR = EA of flange to the right
of the web (not used if NFLR=0)
EAFLLE = EA of flange to the left of the
web (not used if NFLL=0)
EAWEB = EA of the web
EACAP = EA of the cap (not used if
NCAP=0)
EASK = EA of the skin

TFLR,TFLL,TCAP,TSK

TFLR = thickness of flange to the right
of the web (not used if NFLR=0)
TFLL = thickness of flange to the left
of the web (not used if NFLL=0)
TCAP = thickness of the cap (not used
if NCAP=0)
TSK = thickness of the skin

RAD

RAD = flange/web corner inner moldline radius

E,G,THICK,SIG,TAU

E = normal stiffness of interface
G = shear stiffness of interface
THICK = thickness of interface
SIG = Interface normal tensile stress allowable
TAU = Interface shear stress allowable

POSKR,POFLR,POCAP,
POFLL,POSKL

POSKR = uniform load on the skin
to the right of the stiffener (not used
if NSKR=0)
POFLR = uniform load on the flange to
the right of the web (not used if NFLR=0)
POCAP = uniform load on the cap (not

used if NCAP=0)
 POFL = uniform load on the flange to
 the left of the web (not used if NFLL=0)
 POSKL = uniform load on the skin to the
 left of the stiffener (not used if NSKL=0)

NSTITCH

NSTITCH = Number of stitches on the
 skin/flange interface

XT,EAST
 (use one line per stitch)

XT = X-coordinate of stitch
 EAST = EA of the stitch

3.3 SSAMGEN Program Output – The output of SSAMGEN is the mesh data file. This file can be modified if needed or used directly with the SSAM program.

4. Verification

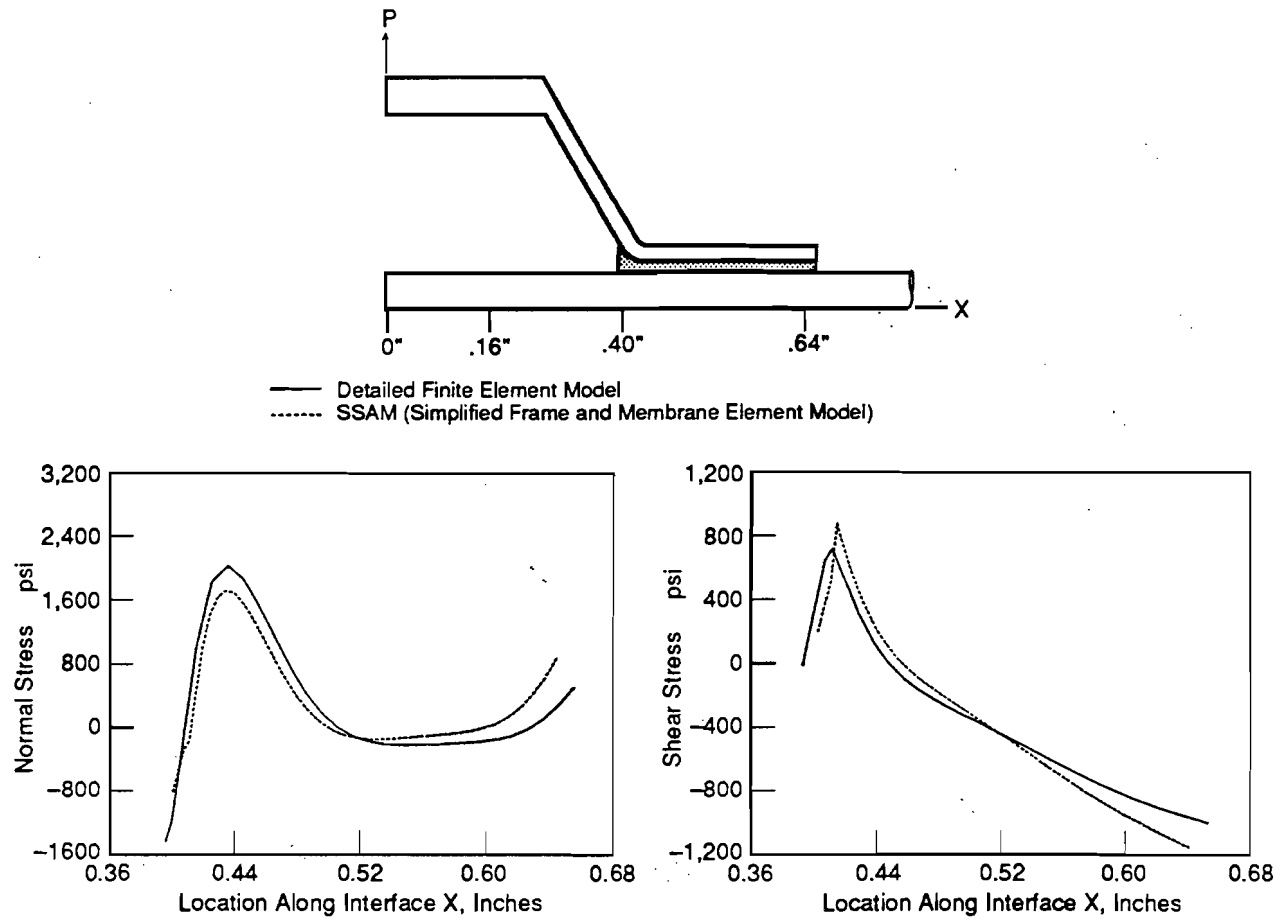
The verification of SSAM was accomplished by comparing SSAM predictions to those of a detailed finite element model and also by comparing SSAM strength predictions to hat pull off test data.

In order to validate the SSAM stress predictions, several example cases were run. Two examples are presented here. The interface stress fields are compared to results obtained from a detailed 2-D generalized plane strain finite element model.

The first example consists of a hat stiffened panel without an internal flange. A point load is applied to the middle of the hat cap, and the panel is fixed at the mid-bay position. The resulting interfacial normal and shear stress distributions are shown in Figure 60 as compared to the detailed finite element results. The stresses are shown to be in good agreement with the detailed model.

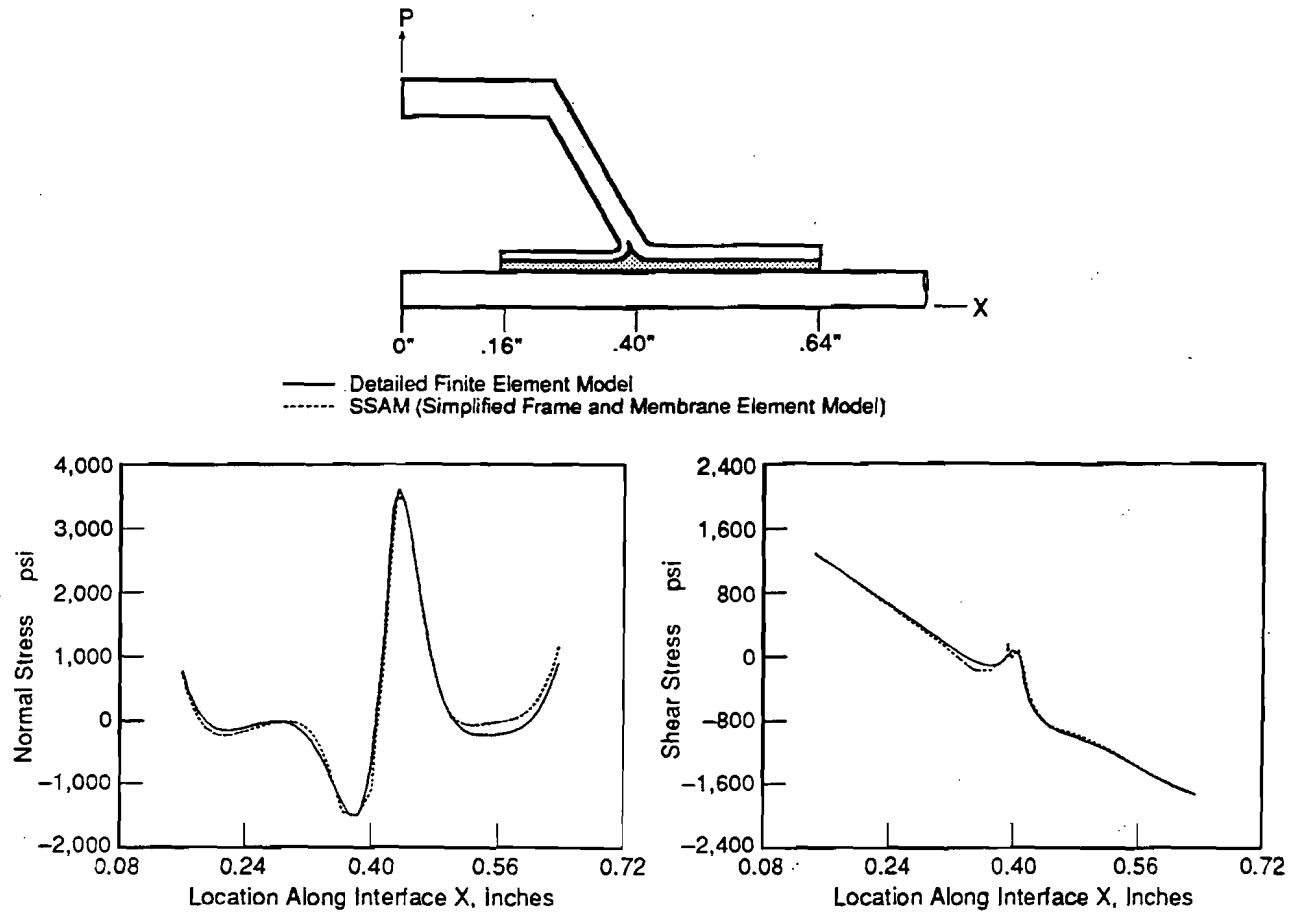
The second example consists of a similar hat stiffened panel this time with an internal flange. As before, a point load is applied to the middle of the hat cap, and the panel is fixed at the mid-bay position. The resulting interfacial normal and shear stress distributions are shown in Figure 61. The stresses are again shown to be in good agreement with the detailed model.

In order to further verify the the analysis, SSAM was used to model a set of hat pull off specimens. The correlation of the SSAM predictions of pull off load and the actual test data are shown in Figure 62. The SSAM predictions are shown to be within the scatter of the test data.



GP34-0277-1

Figure 60. Hat Stiffener Without an Internal Flange



GP34-0277-2

Figure 61. Hat Stiffener With an Internal Flange

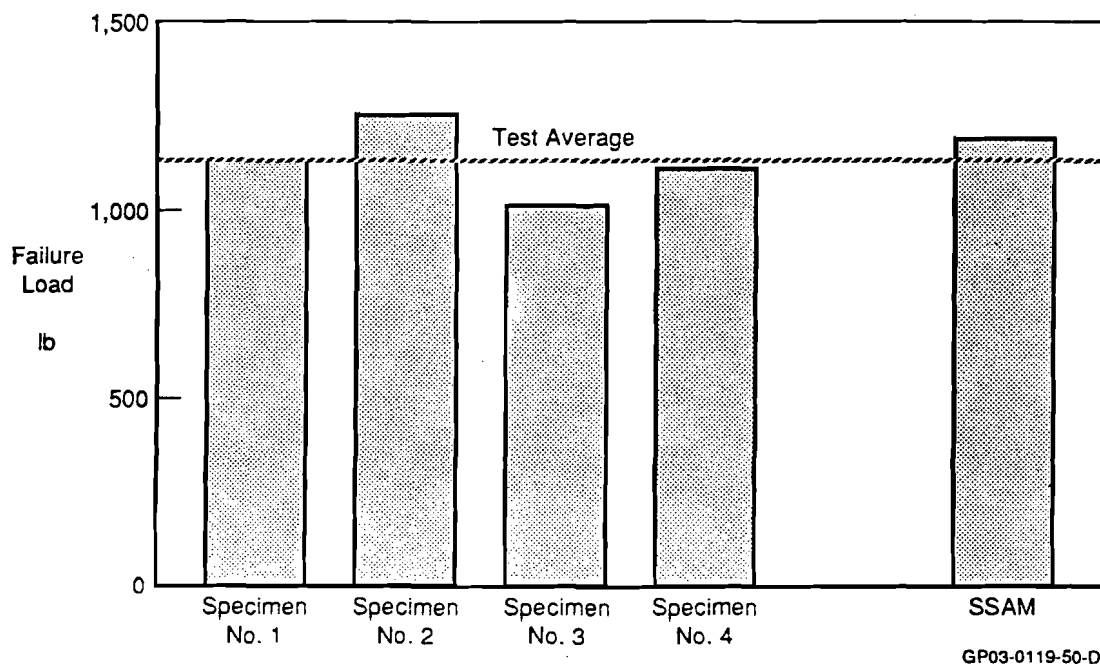


Figure 62. Comparison of SSAM Predictions With Test Data

5. Example Problem

The following pages detail a sample hat stiffener pull-off analysis using the SSAM and SSAMGEN codes. Shown below is a sample SSAMGEN.INP file which holds all the inputs for the SSAMGEN program. The following pages show the steps and inputs needed to run the SSAMGEN and SSAM codes using this input file. The interactive session is followed by a listing of the tabular results file generated by SSAM. The file SSAM-TEST.DAT, created by SSAMGEN, is not shown since its contents are echoed in the tabular output file.

Sample SSAMGEN.INP File:

```

IM7/8551-7A HAT PULL-OFF SPECIMEN WITH INNER FLANGE
10, 20, 35, 5, 6, 15, NSKR,NFLR,NFLL,NSKL,NWEB,NCAP
.70,.5494,0.6506,.1, 0.0, .586, 1.2778,
XSKR,XFLR,XFLL,XSKL,XCAP1,XCAP2,HCAP
2.3178, 0.2897, 7.823, 1313., 1784., EIFLR,EIFLL,EIWEB,EICAP,EISK
36.51E3, 18.26E3, 54.77E3, 1.454E6, 1.228E6, EAFLR,EAFLL,EAWEB,EACAP,EASK
.0276, .0138, .104, .132, TFLR, TFL, TCAP, TSK
0.090, RAD (FLANGE INNER MOLDLINE RADIUS)
.1482E6, 57.E3, 0.005, 3243.,5700., E,G,THICKNESS OF INTERFACE
0.0, 0.0, -750.0, 0.0, 0.0, POSKR,POFLR,POCAP,POFLL,POSKL
0, NSTITCH

```

Sample SSAMGEN and SSAM Input:

B:> ssamgen

```

*****
                      SSAMGEN
                    Version 1.10
*****

COPYRIGHT UNPUBLISHED - (1989/MCAIR).  ALL RIGHTS
RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
DOUGLAS CORPORATION.

```

Enter a model data file name (or STOP)>
ssamtest.dat

Stop - Program terminated.

B:> ssam

```

*****
          SSAM - the Simplified Stiffener Analysis Model
                    Version 1.10
*****

COPYRIGHT UNPUBLISHED - (1989/MCAIR).  ALL RIGHTS
RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
DOUGLAS CORPORATION.

```

Enter the input data file name(or STOP)>
ssamtest.dat

IM7/8551-7A HAT PULL-OFF SPECIMEN WITH INNER FLANGE

Enter an output file name>
ssamtest.out

Stiffness Matrix Bandwidth = 29
Number of Unrestrained DOF = 437

The program iterates until the maximum change in
axial force in the frame elements is less than .02 %

Iteration	Max. % Change
2	42.3393
3	1.2171
4	.0836
5	.0019

Summary of interface loading:
Total normal force is approximately = 459.483034562306400
Total shear force is approximately = -44.425367598647640
Maximum failure function = 1.178109892916511
Stop - Program terminated.

Contents of SSAMTEST.OUT File:

IM7/8551-7A HAT PULL-OFF SPECIMEN WITH INNER FLANGE

Number of Frame Elements = 146
Number of Plate Elements = 55
Number of Restraints = 7
Number of Loaded Nodes = 16

Beam Elements:

EL	N1	N2	EI	EA
1	2	1	.17840E+04	.12280E+07
2	3	2	.17840E+04	.12280E+07
3	4	3	.17840E+04	.12280E+07
4	5	4	.17840E+04	.12280E+07
5	6	5	.17840E+04	.12280E+07
6	7	6	.17840E+04	.12280E+07
7	8	7	.17840E+04	.12280E+07
8	9	8	.17840E+04	.12280E+07
9	10	9	.17840E+04	.12280E+07
10	11	10	.17840E+04	.12280E+07
11	13	11	.17840E+04	.12280E+07
12	15	13	.17840E+04	.12280E+07
13	17	15	.17840E+04	.12280E+07
14	19	17	.17840E+04	.12280E+07
15	21	19	.17840E+04	.12280E+07
16	23	21	.17840E+04	.12280E+07
17	25	23	.17840E+04	.12280E+07
18	27	25	.17840E+04	.12280E+07
19	29	27	.17840E+04	.12280E+07
20	31	29	.17840E+04	.12280E+07
21	33	31	.17840E+04	.12280E+07
22	35	33	.17840E+04	.12280E+07
23	37	35	.17840E+04	.12280E+07
24	39	37	.17840E+04	.12280E+07
25	41	39	.17840E+04	.12280E+07
26	43	41	.17840E+04	.12280E+07
27	45	43	.17840E+04	.12280E+07
28	47	45	.17840E+04	.12280E+07
29	49	47	.17840E+04	.12280E+07
30	51	49	.17840E+04	.12280E+07
31	53	51	.17840E+04	.12280E+07
32	56	53	.17840E+04	.12280E+07
33	59	56	.17840E+04	.12280E+07
34	62	59	.17840E+04	.12280E+07
35	65	62	.17840E+04	.12280E+07
36	68	65	.17840E+04	.12280E+07
37	71	68	.17840E+04	.12280E+07
38	74	71	.17840E+04	.12280E+07
39	77	74	.17840E+04	.12280E+07
40	80	77	.17840E+04	.12280E+07
41	83	80	.17840E+04	.12280E+07
42	86	83	.17840E+04	.12280E+07
43	89	86	.17840E+04	.12280E+07
44	92	89	.17840E+04	.12280E+07
45	95	92	.17840E+04	.12280E+07
46	98	95	.17840E+04	.12280E+07
47	101	98	.17840E+04	.12280E+07
48	104	101	.17840E+04	.12280E+07
49	107	104	.17840E+04	.12280E+07
50	110	107	.17840E+04	.12280E+07

Beam Elements:

	EL	N1	N2	EI	EA
51	113	110		.17840E+04	.12280E+07
52	116	113		.17840E+04	.12280E+07
53	118	116		.17840E+04	.12280E+07
54	120	118		.17840E+04	.12280E+07
55	122	120		.17840E+04	.12280E+07
56	124	122		.17840E+04	.12280E+07
57	126	124		.17840E+04	.12280E+07
58	128	126		.17840E+04	.12280E+07
59	130	128		.17840E+04	.12280E+07
60	132	130		.17840E+04	.12280E+07
61	134	132		.17840E+04	.12280E+07
62	136	134		.17840E+04	.12280E+07
63	138	136		.17840E+04	.12280E+07
64	140	138		.17840E+04	.12280E+07
65	142	140		.17840E+04	.12280E+07
66	144	142		.17840E+04	.12280E+07
67	145	144		.17840E+04	.12280E+07
68	146	145		.17840E+04	.12280E+07
69	147	146		.17840E+04	.12280E+07
70	148	147		.17840E+04	.12280E+07
71	14	12		.23178E+01	.36510E+05
72	16	14		.23178E+01	.36510E+05
73	18	16		.23178E+01	.36510E+05
74	20	18		.23178E+01	.36510E+05
75	22	20		.23178E+01	.36510E+05
76	24	22		.23178E+01	.36510E+05
77	26	24		.23178E+01	.36510E+05
78	28	26		.23178E+01	.36510E+05
79	30	28		.23178E+01	.36510E+05
80	32	30		.23178E+01	.36510E+05
81	34	32		.23178E+01	.36510E+05
82	36	34		.23178E+01	.36510E+05
83	38	36		.23178E+01	.36510E+05
84	40	38		.23178E+01	.36510E+05
85	42	40		.23178E+01	.36510E+05
86	44	42		.23178E+01	.36510E+05
87	46	44		.23178E+01	.36510E+05
88	48	46		.23178E+01	.36510E+05
89	50	48		.23178E+01	.36510E+05
90	52	50		.23178E+01	.36510E+05
91	54	52		.28970E+00	.18260E+05
92	57	54		.28970E+00	.18260E+05
93	60	57		.28970E+00	.18260E+05
94	63	60		.28970E+00	.18260E+05
95	66	63		.28970E+00	.18260E+05
96	69	66		.28970E+00	.18260E+05
97	72	69		.28970E+00	.18260E+05
98	75	72		.28970E+00	.18260E+05
99	78	75		.28970E+00	.18260E+05
100	81	78		.28970E+00	.18260E+05

Beam Elements:

EL	N1	N2	EI	EA
101	84	81	.28970E+00	.18260E+05
102	87	84	.28970E+00	.18260E+05
103	90	87	.28970E+00	.18260E+05
104	93	90	.28970E+00	.18260E+05
105	96	93	.28970E+00	.18260E+05
106	99	96	.28970E+00	.18260E+05
107	102	99	.28970E+00	.18260E+05
108	105	102	.28970E+00	.18260E+05
109	108	105	.28970E+00	.18260E+05
110	111	108	.28970E+00	.18260E+05
111	114	111	.28970E+00	.18260E+05
112	117	114	.28970E+00	.18260E+05
113	119	117	.28970E+00	.18260E+05
114	121	119	.28970E+00	.18260E+05
115	123	121	.28970E+00	.18260E+05
116	125	123	.28970E+00	.18260E+05
117	127	125	.28970E+00	.18260E+05
118	129	127	.28970E+00	.18260E+05
119	131	129	.28970E+00	.18260E+05
120	133	131	.28970E+00	.18260E+05
121	135	133	.28970E+00	.18260E+05
122	137	135	.28970E+00	.18260E+05
123	139	137	.28970E+00	.18260E+05
124	141	139	.28970E+00	.18260E+05
125	143	141	.28970E+00	.18260E+05
126	55	52	.78230E+01	.54770E+05
127	58	55	.78230E+01	.54770E+05
128	61	58	.78230E+01	.54770E+05
129	64	61	.78230E+01	.54770E+05
130	67	64	.78230E+01	.54770E+05
131	70	67	.78230E+01	.54770E+05
132	73	70	.13130E+04	.14540E+07
133	76	73	.13130E+04	.14540E+07
134	79	76	.13130E+04	.14540E+07
135	82	79	.13130E+04	.14540E+07
136	85	82	.13130E+04	.14540E+07
137	88	85	.13130E+04	.14540E+07
138	91	88	.13130E+04	.14540E+07
139	94	91	.13130E+04	.14540E+07
140	97	94	.13130E+04	.14540E+07
141	100	97	.13130E+04	.14540E+07
142	103	100	.13130E+04	.14540E+07
143	106	103	.13130E+04	.14540E+07
144	109	106	.13130E+04	.14540E+07
145	112	109	.13130E+04	.14540E+07
146	115	112	.13130E+04	.14540E+07

Plate Elements:

EL	N1	N2	N3	N4	E	G	FWT	ILS
1	151	149	150	152	.14820E+06	.57000E+05	.32430E+04	.57000E+04
2	153	151	152	154	.14820E+06	.57000E+05	.32430E+04	.57000E+04
3	155	153	154	156	.14820E+06	.57000E+05	.32430E+04	.57000E+04
4	157	155	156	158	.14820E+06	.57000E+05	.32430E+04	.57000E+04
5	159	157	158	160	.14820E+06	.57000E+05	.32430E+04	.57000E+04
6	161	159	160	162	.14820E+06	.57000E+05	.32430E+04	.57000E+04
7	163	161	162	164	.14820E+06	.57000E+05	.32430E+04	.57000E+04
8	165	163	164	166	.14820E+06	.57000E+05	.32430E+04	.57000E+04
9	167	165	166	168	.14820E+06	.57000E+05	.32430E+04	.57000E+04
10	169	167	168	170	.14820E+06	.57000E+05	.32430E+04	.57000E+04
11	171	169	170	172	.14820E+06	.57000E+05	.32430E+04	.57000E+04
12	173	171	172	174	.14820E+06	.57000E+05	.32430E+04	.57000E+04
13	175	173	174	176	.14820E+06	.57000E+05	.32430E+04	.57000E+04
14	177	175	176	178	.14820E+06	.57000E+05	.32430E+04	.57000E+04
15	179	177	178	180	.14820E+06	.57000E+05	.32430E+04	.57000E+04
16	181	179	180	182	.14820E+06	.57000E+05	.32430E+04	.57000E+04
17	183	181	182	184	.14820E+06	.57000E+05	.32430E+04	.57000E+04
18	185	183	184	186	.14820E+06	.57000E+05	.32430E+04	.57000E+04
19	187	185	186	188	.14820E+06	.57000E+05	.32430E+04	.57000E+04
20	189	187	188	190	.14820E+06	.57000E+05	.32430E+04	.57000E+04
21	191	189	190	192	.14820E+06	.57000E+05	.32430E+04	.57000E+04
22	193	191	192	194	.14820E+06	.57000E+05	.32430E+04	.57000E+04
23	195	193	194	196	.14820E+06	.57000E+05	.32430E+04	.57000E+04
24	197	195	196	198	.14820E+06	.57000E+05	.32430E+04	.57000E+04
25	199	197	198	200	.14820E+06	.57000E+05	.32430E+04	.57000E+04
26	201	199	200	202	.14820E+06	.57000E+05	.32430E+04	.57000E+04
27	203	201	202	204	.14820E+06	.57000E+05	.32430E+04	.57000E+04
28	205	203	204	206	.14820E+06	.57000E+05	.32430E+04	.57000E+04
29	207	205	206	208	.14820E+06	.57000E+05	.32430E+04	.57000E+04
30	209	207	208	210	.14820E+06	.57000E+05	.32430E+04	.57000E+04
31	211	209	210	212	.14820E+06	.57000E+05	.32430E+04	.57000E+04
32	213	211	212	214	.14820E+06	.57000E+05	.32430E+04	.57000E+04
33	215	213	214	216	.14820E+06	.57000E+05	.32430E+04	.57000E+04
34	217	215	216	218	.14820E+06	.57000E+05	.32430E+04	.57000E+04
35	219	217	218	220	.14820E+06	.57000E+05	.32430E+04	.57000E+04
36	221	219	220	222	.14820E+06	.57000E+05	.32430E+04	.57000E+04
37	223	221	222	224	.14820E+06	.57000E+05	.32430E+04	.57000E+04
38	225	223	224	226	.14820E+06	.57000E+05	.32430E+04	.57000E+04
39	227	225	226	228	.14820E+06	.57000E+05	.32430E+04	.57000E+04
40	229	227	228	230	.14820E+06	.57000E+05	.32430E+04	.57000E+04
41	231	229	230	232	.14820E+06	.57000E+05	.32430E+04	.57000E+04
42	233	231	232	234	.14820E+06	.57000E+05	.32430E+04	.57000E+04
43	235	233	234	236	.14820E+06	.57000E+05	.32430E+04	.57000E+04
44	237	235	236	238	.14820E+06	.57000E+05	.32430E+04	.57000E+04
45	239	237	238	240	.14820E+06	.57000E+05	.32430E+04	.57000E+04
46	241	239	240	242	.14820E+06	.57000E+05	.32430E+04	.57000E+04
47	243	241	242	244	.14820E+06	.57000E+05	.32430E+04	.57000E+04
48	245	243	244	246	.14820E+06	.57000E+05	.32430E+04	.57000E+04
49	247	245	246	248	.14820E+06	.57000E+05	.32430E+04	.57000E+04
50	249	247	248	250	.14820E+06	.57000E+05	.32430E+04	.57000E+04

Plate Elements::

EL	N1	N2	N3	N4	E	G	FWT	ILS
51	251	249	250	252	.14820E+06	.57000E+05	.32430E+04	.57000E+04
52	253	251	252	254	.14820E+06	.57000E+05	.32430E+04	.57000E+04
53	255	253	254	256	.14820E+06	.57000E+05	.32430E+04	.57000E+04
54	257	255	256	258	.14820E+06	.57000E+05	.32430E+04	.57000E+04
55	259	257	258	260	.14820E+06	.57000E+05	.32430E+04	.57000E+04

Node	Con	X-ord	Y-ord
1	0	.20000E+01	.66000E-01
2	0	.19300E+01	.66000E-01
3	0	.18600E+01	.66000E-01
4	0	.17900E+01	.66000E-01
5	0	.17200E+01	.66000E-01
6	0	.16500E+01	.66000E-01
7	0	.15800E+01	.66000E-01
8	0	.15100E+01	.66000E-01
9	0	.14400E+01	.66000E-01
10	0	.13700E+01	.66000E-01
11	149	.13000E+01	.66000E-01
12	150	.13000E+01	.15080E+00
13	151	.12740E+01	.66000E-01
14	152	.12740E+01	.15080E+00
15	153	.12479E+01	.66000E-01
16	154	.12479E+01	.15080E+00
17	155	.12219E+01	.66000E-01
18	156	.12219E+01	.15080E+00
19	157	.11959E+01	.66000E-01
20	158	.11959E+01	.15080E+00
21	159	.11698E+01	.66000E-01
22	160	.11698E+01	.15080E+00
23	161	.11438E+01	.66000E-01
24	162	.11438E+01	.15080E+00
25	163	.11178E+01	.66000E-01
26	164	.11178E+01	.15080E+00
27	165	.10917E+01	.66000E-01
28	166	.10917E+01	.15080E+00
29	167	.10657E+01	.66000E-01
30	168	.10657E+01	.15080E+00
31	169	.10397E+01	.66000E-01
32	170	.10397E+01	.15080E+00
33	171	.10136E+01	.66000E-01
34	172	.10136E+01	.15080E+00
35	173	.98760E+00	.66000E-01
36	174	.98760E+00	.15080E+00
37	175	.96157E+00	.66000E-01
38	176	.96157E+00	.15080E+00
39	177	.93553E+00	.66000E-01
40	178	.93553E+00	.15080E+00
41	179	.90950E+00	.66000E-01
42	180	.90950E+00	.15080E+00
43	181	.88347E+00	.66000E-01

Node	Con	X-ord	Y-ord
44	182	.88347E+00	.15080E+00
45	183	.85743E+00	.66000E-01
46	184	.85743E+00	.15080E+00
47	185	.81837E+00	.66000E-01
48	186	.80948E+00	.16254E+00
49	187	.78374E+00	.66000E-01
50	188	.77237E+00	.19511E+00
51	189	.75748E+00	.66000E-01
52	190	.75451E+00	.24114E+00
53	191	.74487E+00	.66000E-01
54	192	.74816E+00	.19541E+00
55	0	.72643E+00	.44689E+00
56	193	.72483E+00	.66000E-01
57	194	.73007E+00	.16834E+00
58	0	.69834E+00	.65263E+00
59	195	.69937E+00	.66000E-01
60	196	.70300E+00	.15025E+00
61	0	.67026E+00	.85837E+00
62	197	.67107E+00	.66000E-01
63	198	.67107E+00	.14390E+00
64	0	.64217E+00	.10641E+01
65	199	.65264E+00	.66000E-01
66	200	.65264E+00	.14390E+00
67	0	.61409E+00	.12699E+01
68	201	.63422E+00	.66000E-01
69	202	.63422E+00	.14390E+00
70	0	.58600E+00	.14618E+01
71	203	.61580E+00	.66000E-01
72	204	.61580E+00	.14390E+00
73	0	.54693E+00	.14618E+01
74	205	.59738E+00	.66000E-01
75	206	.59738E+00	.14390E+00
76	0	.50787E+00	.14618E+01
77	207	.57896E+00	.66000E-01
78	208	.57896E+00	.14390E+00
79	0	.46880E+00	.14618E+01
80	209	.56054E+00	.66000E-01
81	210	.56054E+00	.14390E+00
82	0	.42973E+00	.14618E+01
83	211	.54212E+00	.66000E-01
84	212	.54212E+00	.14390E+00
85	0	.39067E+00	.14618E+01
86	213	.52369E+00	.66000E-01
87	214	.52369E+00	.14390E+00
88	0	.35160E+00	.14618E+01
89	215	.50527E+00	.66000E-01
90	216	.50527E+00	.14390E+00
91	0	.31253E+00	.14618E+01
92	217	.48685E+00	.66000E-01
93	218	.48685E+00	.14390E+00
94	0	.27347E+00	.14618E+01

Node	Con	X-ord	Y-ord
95	219	.46843E+00	.66000E-01
96	220	.46843E+00	.14390E+00
97	0	.23440E+00	.14618E+01
98	221	.45001E+00	.66000E-01
99	222	.45001E+00	.14390E+00
100	0	.19533E+00	.14618E+01
101	223	.43159E+00	.66000E-01
102	224	.43159E+00	.14390E+00
103	0	.15627E+00	.14618E+01
104	225	.41317E+00	.66000E-01
105	226	.41317E+00	.14390E+00
106	0	.11720E+00	.14618E+01
107	227	.39474E+00	.66000E-01
108	228	.39474E+00	.14390E+00
109	0	.78133E-01	.14618E+01
110	229	.37632E+00	.66000E-01
111	230	.37632E+00	.14390E+00
112	0	.39067E-01	.14618E+01
113	231	.35790E+00	.66000E-01
114	232	.35790E+00	.14390E+00
115	0	-.24980E-15	.14618E+01
116	233	.33948E+00	.66000E-01
117	234	.33948E+00	.14390E+00
118	235	.32106E+00	.66000E-01
119	236	.32106E+00	.14390E+00
120	237	.30264E+00	.66000E-01
121	238	.30264E+00	.14390E+00
122	239	.28421E+00	.66000E-01
123	240	.28421E+00	.14390E+00
124	241	.26579E+00	.66000E-01
125	242	.26579E+00	.14390E+00
126	243	.24737E+00	.66000E-01
127	244	.24737E+00	.14390E+00
128	245	.22895E+00	.66000E-01
129	246	.22895E+00	.14390E+00
130	247	.21053E+00	.66000E-01
131	248	.21053E+00	.14390E+00
132	249	.19211E+00	.66000E-01
133	250	.19211E+00	.14390E+00
134	251	.17369E+00	.66000E-01
135	252	.17369E+00	.14390E+00
136	253	.15526E+00	.66000E-01
137	254	.15526E+00	.14390E+00
138	255	.13684E+00	.66000E-01
139	256	.13684E+00	.14390E+00
140	257	.11842E+00	.66000E-01
141	258	.11842E+00	.14390E+00
142	259	.10000E+00	.66000E-01
143	260	.10000E+00	.14390E+00
144	0	.80000E-01	.66000E-01
145	0	.60000E-01	.66000E-01

Node	Con	X-ord	Y-ord
146	0	.40000E-01	.66000E-01
147	0	.20000E-01	.66000E-01
148	0	.34694E-17	.66000E-01
149	11	.13000E+01	.13200E+00
150	12	.13000E+01	.13700E+00
151	13	.12740E+01	.13200E+00
152	14	.12740E+01	.13700E+00
153	15	.12479E+01	.13200E+00
154	16	.12479E+01	.13700E+00
155	17	.12219E+01	.13200E+00
156	18	.12219E+01	.13700E+00
157	19	.11959E+01	.13200E+00
158	20	.11959E+01	.13700E+00
159	21	.11698E+01	.13200E+00
160	22	.11698E+01	.13700E+00
161	23	.11438E+01	.13200E+00
162	24	.11438E+01	.13700E+00
163	25	.11178E+01	.13200E+00
164	26	.11178E+01	.13700E+00
165	27	.10917E+01	.13200E+00
166	28	.10917E+01	.13700E+00
167	29	.10657E+01	.13200E+00
168	30	.10657E+01	.13700E+00
169	31	.10397E+01	.13200E+00
170	32	.10397E+01	.13700E+00
171	33	.10136E+01	.13200E+00
172	34	.10136E+01	.13700E+00
173	35	.98760E+00	.13200E+00
174	36	.98760E+00	.13700E+00
175	37	.96157E+00	.13200E+00
176	38	.96157E+00	.13700E+00
177	39	.93553E+00	.13200E+00
178	40	.93553E+00	.13700E+00
179	41	.90950E+00	.13200E+00
180	42	.90950E+00	.13700E+00
181	43	.88347E+00	.13200E+00
182	44	.88347E+00	.13700E+00
183	45	.85743E+00	.13200E+00
184	46	.85743E+00	.13700E+00
185	47	.81837E+00	.13200E+00
186	48	.81837E+00	.14368E+00
187	49	.78374E+00	.13200E+00
188	50	.78374E+00	.16295E+00
189	51	.75748E+00	.13200E+00
190	52	.75748E+00	.19264E+00
191	53	.74487E+00	.13200E+00
192	54	.74487E+00	.17524E+00
193	56	.72483E+00	.13200E+00
194	57	.72483E+00	.15474E+00
195	59	.69937E+00	.13200E+00
196	60	.69937E+00	.14155E+00

Node	Con	X-ord	Y-ord
197	62	.67107E+00	.13200E+00
198	63	.67107E+00	.13700E+00
199	65	.65264E+00	.13200E+00
200	66	.65264E+00	.13700E+00
201	68	.63422E+00	.13200E+00
202	69	.63422E+00	.13700E+00
203	71	.61580E+00	.13200E+00
204	72	.61580E+00	.13700E+00
205	74	.59738E+00	.13200E+00
206	75	.59738E+00	.13700E+00
207	77	.57896E+00	.13200E+00
208	78	.57896E+00	.13700E+00
209	80	.56054E+00	.13200E+00
210	81	.56054E+00	.13700E+00
211	83	.54212E+00	.13200E+00
212	84	.54212E+00	.13700E+00
213	86	.52369E+00	.13200E+00
214	87	.52369E+00	.13700E+00
215	89	.50527E+00	.13200E+00
216	90	.50527E+00	.13700E+00
217	92	.48685E+00	.13200E+00
218	93	.48685E+00	.13700E+00
219	95	.46843E+00	.13200E+00
220	96	.46843E+00	.13700E+00
221	98	.45001E+00	.13200E+00
222	99	.45001E+00	.13700E+00
223	101	.43159E+00	.13200E+00
224	102	.43159E+00	.13700E+00
225	104	.41317E+00	.13200E+00
226	105	.41317E+00	.13700E+00
227	107	.39474E+00	.13200E+00
228	108	.39474E+00	.13700E+00
229	110	.37632E+00	.13200E+00
230	111	.37632E+00	.13700E+00
231	113	.35790E+00	.13200E+00
232	114	.35790E+00	.13700E+00
233	116	.33948E+00	.13200E+00
234	117	.33948E+00	.13700E+00
235	118	.32106E+00	.13200E+00
236	119	.32106E+00	.13700E+00
237	120	.30264E+00	.13200E+00
238	121	.30264E+00	.13700E+00
239	122	.28421E+00	.13200E+00
240	123	.28421E+00	.13700E+00
241	124	.26579E+00	.13200E+00
242	125	.26579E+00	.13700E+00
243	126	.24737E+00	.13200E+00
244	127	.24737E+00	.13700E+00
245	128	.22895E+00	.13200E+00
246	129	.22895E+00	.13700E+00
247	130	.21053E+00	.13200E+00

Node	Con	X-ord	Y-ord
248	131	.21053E+00	.13700E+00
249	132	.19211E+00	.13200E+00
250	133	.19211E+00	.13700E+00
251	134	.17369E+00	.13200E+00
252	135	.17369E+00	.13700E+00
253	136	.15526E+00	.13200E+00
254	137	.15526E+00	.13700E+00
255	138	.13684E+00	.13200E+00
256	139	.13684E+00	.13700E+00
257	140	.11842E+00	.13200E+00
258	141	.11842E+00	.13700E+00
259	142	.10000E+00	.13200E+00
260	143	.10000E+00	.13700E+00

Nodal Forces:

Node	X-Force	Y-Force	Moment
70	.00000E+00	.14650E+02	.95388E-01
73	.00000E+00	.29300E+02	.11574E-16
76	.00000E+00	.29300E+02	.11574E-16
79	.00000E+00	.29300E+02	.11574E-16
82	.00000E+00	.29300E+02	.11574E-16
85	.00000E+00	.29300E+02	.11574E-16
88	.00000E+00	.29300E+02	.11574E-16
91	.00000E+00	.29300E+02	.11574E-16
94	.00000E+00	.29300E+02	-.12560E-15
97	.00000E+00	.29300E+02	-.69796E-18
100	.00000E+00	.29300E+02	-.69796E-18
103	.00000E+00	.29300E+02	-.69796E-18
106	.00000E+00	.29300E+02	-.69796E-18
109	.00000E+00	.29300E+02	-.69796E-18
112	.00000E+00	.29300E+02	-.69796E-18
115	.00000E+00	.14650E+02	-.95388E-01

NODAL POINT DISPLACEMENTS:

Node	U	V	DV/DX
1	.00000000	.00000000	.00000000
2	-.0000053	.0004807	-.0135321
3	-.0000106	.0018664	-.0258606
4	-.0000159	.0040731	-.0369884
5	-.0000212	.0070168	-.0469185
6	-.0000265	.0106138	-.0556535
7	-.0000318	.0147805	-.0631954
8	-.0000371	.0194334	-.0695464
9	-.0000424	.0244892	-.0747079
10	-.0000477	.0298648	-.0786813
11	-.0000530	.0354769	-.0814676
12	.0069401	.0354588	-.0830383
13	-.0000551	.0376048	-.0821916
14	.0069445	.0376112	-.0827591
15	-.0000574	.0397576	-.0827456

NODAL POINT DISPLACEMENTS:

Node	U	V	DV/DX
16	.0069543	.0397666	-.0826121
17	-.0000596	.0419144	-.0831339
18	.0069637	.0419195	-.0829100
19	-.0000617	.0440793	-.0833667
20	.0069684	.0440820	-.0831695
21	-.0000636	.0462566	-.0834492
22	.0069657	.0462582	-.0832475
23	-.0000652	.0484258	-.0833839
24	.0069538	.0484265	-.0831654
25	-.0000665	.0505913	-.0831730
26	.0069317	.0505911	-.0829434
27	-.0000675	.0527578	-.0828159
28	.0068987	.0527568	-.0825809
29	-.0000681	.0549049	-.0823160
30	.0068547	.0549030	-.0820782
31	-.0000684	.0570371	-.0816726
32	.0067994	.0570343	-.0814329
33	-.0000684	.0591588	-.0808824
34	.0067324	.0591551	-.0806435
35	-.0000680	.0612500	-.0799515
36	.0066542	.0612455	-.0797167
37	-.0000672	.0633175	-.0788757
38	.0065645	.0633121	-.0786359
39	-.0000661	.0653558	-.0776550
40	.0064637	.0653490	-.0773735
41	-.0000647	.0673598	-.0762896
42	.0063521	.0673500	-.0759782
43	-.0000630	.0693263	-.0747774
44	.0062303	.0693159	-.0747977
45	-.0000610	.0712522	-.0731184
46	.0060983	.0712585	-.0741084
47	-.0000571	.0740559	-.0703844
48	.0067040	.0749154	-.0758856
49	-.0000530	.0764483	-.0677564
50	.0086793	.0777149	-.0647452
51	-.0000501	.0782001	-.0656545
52	.0106393	.0789709	-.0377719
53	-.0000485	.0790216	-.0646279
54	.0082077	.0791240	-.0425688
55	.0122295	.0808902	.0084482
56	-.0000464	.0803001	-.0629614
57	.0064930	.0800447	-.0604519
58	.0090205	.0821542	.0180589
59	-.0000440	.0818757	-.0608101
60	.0051212	.0816981	-.0618211
61	.0048561	.0832880	.0195405
62	-.0000414	.0835625	-.0583946
63	.0045189	.0835701	-.0577401
64	.0007597	.0844307	.0172912
65	-.0000398	.0846241	-.0568137

NODAL POINT DISPLACEMENTS:

Node	U	V	DV/DX
66	.0043926	.0846142	-.0561856
67	-.0020547	.0857493	.0045835
68	-.0000382	.0856561	-.0552315
69	.0042685	.0856418	-.0552916
70	.0000279	.0876464	-.0366418
71	-.0000366	.0866589	-.0536488
72	.0041453	.0866459	-.0537303
73	.0000261	.0890766	-.0364872
74	-.0000351	.0876325	-.0520658
75	.0040225	.0876200	-.0520773
76	.0000242	.0904910	-.0358589
77	-.0000336	.0885770	-.0504828
78	.0039001	.0885643	-.0504841
79	.0000224	.0918725	-.0347905
80	-.0000320	.0894923	-.0488998
81	.0037780	.0894796	-.0489042
82	.0000205	.0932042	-.0333162
83	-.0000305	.0903784	-.0473168
84	.0036559	.0903657	-.0473204
85	.0000186	.0944707	-.0314704
86	-.0000290	.0912359	-.0457330
87	.0035339	.0912232	-.0457352
88	.0000168	.0956586	-.0292860
89	-.0000274	.0920637	-.0441503
90	.0034121	.0920510	-.0441518
91	.0000149	.0967551	-.0267974
92	-.0000259	.0928624	-.0425676
93	.0032903	.0928497	-.0425688
94	.0000130	.0977488	-.0240392
95	-.0000244	.0936319	-.0409851
96	.0031685	.0936192	-.0409861
97	.0000112	.0986302	-.0210438
98	-.0000229	.0943723	-.0394027
99	.0030467	.0943596	-.0394036
100	.0000093	.0993905	-.0178458
101	-.0000213	.0950835	-.0378204
102	.0029250	.0950708	-.0378214
103	.0000075	.1000223	-.0144797
104	-.0000198	.0957656	-.0362382
105	.0028033	.0957529	-.0362394
106	.0000056	.1005200	-.0109777
107	-.0000183	.0964189	-.0346553
108	.0026816	.0964062	-.0346567
109	.0000037	.1008788	-.0073747
110	-.0000168	.0970426	-.0330733
111	.0025600	.0970299	-.0330753
112	.0000019	.1010954	-.0037043
113	-.0000152	.0976373	-.0314914
114	.0024384	.0976246	-.0314943
115	.0000000	.1011678	..0000000

NODAL POINT DISPLACEMENTS:

Node	U	V	DV/DX
116	-.0000137	.0982028	-.0299095
117	.0023170	.0981901	-.0299139
118	-.0000122	.0987391	-.0283276
119	.0021957	.0987264	-.0283345
120	-.0000107	.0992464	-.0267456
121	.0020746	.0992336	-.0267564
122	-.0000092	.0997247	-.0251626
123	.0019538	.0997119	-.0251797
124	-.0000077	.1001736	-.0235800
125	.0018337	.1001607	-.0236073
126	-.0000062	.1005934	-.0219967
127	.0017145	.1005804	-.0220398
128	-.0000047	.1009840	-.0204123
129	.0015968	.1009707	-.0204789
130	-.0000033	.1013454	-.0188260
131	.0014815	.1013318	-.0189355
132	-.0000019	.1016775	-.0172367
133	.0013699	.1016636	-.0174381
134	-.0000006	.1019803	-.0156425
135	.0012643	.1019663	-.0159466
136	.0000005	.1022538	-.0140397
137	.0011682	.1022371	-.0142329
138	.0000014	.1024976	-.0124246
139	.0010878	.1024714	-.0127639
140	.0000019	.1027114	-.0107893
141	.0010322	.1026939	-.0144916
142	.0000019	.1028948	-.0091273
143	.0010126	.1029582	-.0164024
144	.0000015	.1030591	-.0073018
145	.0000012	.1031869	-.0054763
146	.0000008	.1032782	-.0036508
147	.0000004	.1033329	-.0018254
148	.0000000	.1033512	.0000000
149	.0053239	.0354769	
150	.0057942	.0354588	
151	.0053695	.0376048	
152	.0058024	.0376112	
153	.0054038	.0397576	
154	.0058143	.0397666	
155	.0054272	.0419144	
156	.0058195	.0419195	
157	.0054405	.0440793	
158	.0058207	.0440820	
159	.0054440	.0462566	
160	.0058169	.0462582	
161	.0054381	.0484258	
162	.0058061	.0484265	
163	.0054229	.0505913	
164	.0057871	.0505911	
165	.0053983	.0527578	

NODAL POINT DISPLACEMENTS:

Node	U	V
166	.0057591	.0527568
167	.0053647	.0549049
168	.0057220	.0549030
169	.0053220	.0570371
170	.0056756	.0570343
171	.0052699	.0591588
172	.0056195	.0591551
173	.0052088	.0612500
174	.0055541	.0612455
175	.0051386	.0633175
176	.0054794	.0633121
177	.0050591	.0653558
178	.0053959	.0653490
179	.0049704	.0673598
180	.0053036	.0673500
181	.0048723	.0693263
182	.0051981	.0693159
183	.0047648	.0712522
184	.0050756	.0712585
185	.0045883	.0740559
186	.0052728	.0742408
187	.0044189	.0764483
188	.0065971	.0769787
189	.0042831	.0782001
190	.0088074	.0788587
191	.0042169	.0790216
192	.0073491	.0792641
193	.0041090	.0803001
194	.0056709	.0803614
195	.0039695	.0818757
196	.0045833	.0819225
197	.0038127	.0835625
198	.0041205	.0835701
199	.0037099	.0846241
200	.0040049	.0846142
201	.0036071	.0856561
202	.0038870	.0856418
203	.0035042	.0866589
204	.0037745	.0866459
205	.0034012	.0876325
206	.0036632	.0876200
207	.0032983	.0885770
208	.0035518	.0885643
209	.0031953	.0894923
210	.0034405	.0894796
211	.0030924	.0903784
212	.0033294	.0903657
213	.0029894	.0912359
214	.0032184	.0912232
215	.0028865	.0920637

NODAL POINT DISPLACEMENTS:

Node	U	V
216	.0031074	.0920510
217	.0027835	.0928624
218	.0029965	.0928497
219	.0026806	.0936319
220	.0028857	.0936192
221	.0025777	.0943723
222	.0027749	.0943596
223	.0024748	.0950835
224	.0026640	.0950708
225	.0023719	.0957656
226	.0025533	.0957529
227	.0022690	.0964189
228	.0024425	.0964062
229	.0021661	.0970426
230	.0023318	.0970299
231	.0020632	.0976373
232	.0022211	.0976246
233	.0019603	.0982028
234	.0021106	.0981901
235	.0018574	.0987391
236	.0020002	.0987264
237	.0017545	.0992464
238	.0018900	.0992336
239	.0016516	.0997247
240	.0017801	.0997119
241	.0015486	.1001736
242	.0016708	.1001607
243	.0014456	.1005934
244	.0015625	.1005804
245	.0013425	.1009840
246	.0014555	.1009707
247	.0012392	.1013454
248	.0013508	.1013318
249	.0011357	.1016775
250	.0012496	.1016636
251	.0010318	.1019803
252	.0011543	.1019663
253	.0009271	.1022538
254	.0010700	.1022371
255	.0008214	.1024976
256	.0009998	.1024714
257	.0007140	.1027114
258	.0009322	.1026939
259	.0006043	.1028948
260	.0008995	.1029582

NAWCADWAR-94138-60 (Volume II)

Report No. NADC
Volume II

ELEM	X-FORCE	Y-FORCE	MOMENT
1	92.94227	-439.50000	314.14101
2	92.94227	-439.50001	283.50481
3	92.94227	-439.50001	252.94490
4	92.94227	-439.50001	222.45350
5	92.94227	-439.50001	192.02281
6	92.94227	-439.50002	161.64507
7	92.94227	-439.50002	131.31252
8	92.94227	-439.50002	101.01742
9	92.94227	-439.50002	70.75203
10	92.94227	-439.50002	40.50863
11	100.68541	-446.97153	38.16245
12	106.89081	-445.34608	26.35834
13	106.04672	-438.07958	15.36943
14	99.53498	-432.95381	4.82452
15	88.90373	-429.90764	-5.48273
16	75.78674	-427.66236	-15.52115
17	61.26234	-425.73205	-25.47382
18	45.85695	-424.01304	-35.42330
19	29.93981	-422.45945	-45.25104
20	13.76583	-421.05131	-55.08323
21	-2.57436	-419.76238	-64.97027
22	-18.93756	-418.56467	-74.77603
23	-35.17864	-417.44870	-84.63479
24	-51.11060	-416.55638	-94.52506
25	-66.34741	-416.62314	-104.49587
26	-80.89720	-418.20880	-114.60128
27	-96.31397	-416.60768	-124.59898
28	-122.04662	-369.91715	-139.49202
29	-145.08547	-257.64303	-144.47735
30	-137.83835	-165.90726	-147.27112
31	-148.38734	-124.74412	-146.87795
32	-128.66578	-87.09032	-150.17550
33	-118.70327	-54.52731	-152.22981
34	-112.94517	-18.21199	-152.87932
35	-107.31207	2.99927	-153.02442
36	-104.34676	6.89202	-153.16502
37	-103.21217	6.53154	-153.22642
38	-102.78239	6.10311	-153.24638
39	-102.45543	6.01021	-153.25487
40	-102.21627	5.90566	-153.25640
41	-102.06480	5.74709	-153.25244
42	-101.96763	5.57273	-153.24516
43	-101.90168	5.39393	-153.23619
44	-101.85454	5.21041	-153.22636
45	-101.81895	5.02330	-153.21625
46	-101.78956	4.83359	-153.20627
47	-101.76213	4.64167	-153.19672
48	-101.73297	4.44734	-153.18791
49	-101.69756	4.24981	-153.18016
50	-101.65075	4.04785	-153.17397
51	-101.58362	3.83878	-153.17002

ELEM	X-FORCE	Y-FORCE	MOMENT
52	-101.48334	3.61889	-153.16940
53	-101.32942	3.38166	-153.17380
54	-101.08926	3.11691	-153.18595
55	-100.71071	2.80845	-153.21018
56	-100.11155	2.43058	-153.25338
57	-99.15943	1.94385	-153.32652
58	-97.64149	1.28095	-153.44736
59	-95.21886	.32549	-153.64447
60	-91.37177	-1.06953	-153.96127
61	-85.27212	-3.01317	-154.46307
62	-75.36505	-6.01312	-155.27603
63	-59.02691	-12.40179	-156.65599
64	-34.44117	-23.17150	-158.81202
65	-1.56300	-22.11830	-161.37999
66	23.60731	.00000	-162.83364
67	23.60731	.00000	-162.83062
68	23.60731	.00000	-162.82847
69	23.60731	.00000	-162.82717
70	23.60731	.00000	-162.82674
71	-6.20797	5.18966	.10336
72	-13.72492	5.33156	.11131
73	-13.15872	-.67990	-.00530
74	-6.65139	-4.84771	-.11010
75	3.84877	-7.28641	-.17907
76	16.73599	-9.17893	-.22782
77	31.00104	-10.90178	-.26970
78	46.13604	-12.49526	-.30839
79	61.77620	-13.97530	-.34195
80	77.67036	-15.34471	-.37383
81	93.72719	-16.61940	-.40464
82	109.80353	-17.82352	-.43149
83	125.76244	-18.97303	-.46044
84	141.45107	-19.95460	-.48831
85	156.53112	-20.04746	-.48994
86	170.80133	-18.54431	-.42038
87	185.02736	-19.86478	-.39919
88	198.76409	-63.20608	-.22772
89	198.91020	-179.12122	-.24795
90	162.86825	-303.26141	1.16617
91	34.00794	68.26619	-1.00244
92	85.47659	69.20617	-.79596
93	116.28622	50.00773	-.60427
94	125.72830	16.49945	-.18567
95	125.11270	-3.89608	-.03122
96	123.01352	-7.34281	-.08628
97	122.15445	-6.68798	-.08671
98	121.67732	-6.20636	-.08136
99	121.33414	-6.08824	-.08021
100	121.10622	-5.95615	-.07943
101	120.96187	-5.77895	-.07805
102	120.86731	-5.59398	-.07649

ELEM	X-FORCE	Y-FORCE	MOMENT
103	120.80309	-5.40874	-.07482
104	120.75747	-5.22122	-.07314
105	120.72307	-5.03180	-.07142
106	120.69461	-4.84104	-.06967
107	120.66813	-4.64908	-.06790
108	120.63992	-4.45576	-.06611
109	120.60576	-4.26052	-.06430
110	120.56051	-4.06268	-.06240
111	120.49564	-3.86050	-.06044
112	120.39871	-3.65183	-.05837
113	120.24988	-3.43268	-.05611
114	120.01767	-3.19693	-.05356
115	119.65160	-2.93490	-.05055
116	119.07200	-2.63123	-.04676
117	118.15085	-2.26312	-.04183
118	116.68337	-1.79039	-.03503
119	114.34148	-1.13805	-.02487
120	110.60962	-.21946	-.00950
121	104.67999	.96097	.00984
122	95.17831	2.65779	.03468
123	79.70438	6.94429	.11397
124	55.17693	15.24504	.31427
125	19.35334	10.95235	.23441
126	69.33496	-439.50000	.23564
127	69.33496	-439.50000	.81421
128	69.33496	-439.50000	.97296
129	69.33496	-439.50000	1.15242
130	69.33496	-439.50000	1.92386
131	69.33496	-439.50000	3.43866
132	69.33496	-424.85000	-21.74504
133	69.33496	-395.55000	-36.52280
134	69.33496	-366.25000	-50.16600
135	69.33496	-336.95000	-62.66593
136	69.33496	-307.65000	-74.02102
137	69.33496	-278.35000	-84.24295
138	69.33496	-249.05000	-93.32493
139	69.33496	-219.75000	-101.26599
140	69.33496	-190.45000	-108.07449
141	69.33496	-161.15000	-113.74552
142	69.33496	-131.85000	-118.27874
143	69.33496	-102.55000	-121.67915
144	69.33496	-73.25000	-123.94345
145	69.33496	-43.95000	-125.07303
146	69.33496	-14.65000	-125.06803

Interface Stresses:

X	NORMAL	SHEAR	FAILURE
1.28700	-322.64151	456.36712	.00641
1.26095	231.56817	102.89650	.00542
1.23490	234.69797	-148.10364	.00591
1.20890	132.65305	-339.69975	.00522
1.18285	86.54533	-461.19656	.00726
1.15680	73.79808	-531.75428	.00922
1.13080	66.67412	-573.23468	.01054
1.10475	60.67669	-598.33985	.01137
1.07870	55.68055	-613.05057	.01186
1.05270	51.09477	-620.80537	.01211
1.02665	47.59418	-623.65116	.01219
1.00060	44.90467	-622.36437	.01211
.97459	42.27153	-615.65693	.01184
.94855	25.34956	-597.80565	.01106
.92252	-34.16149	-565.88370	.00986
.89649	-67.96258	-549.37719	.00929
.87045	295.92507	-605.51140	.01961
.83790	2700.59659	-764.59424	.71146
.80106	3519.21972	-128.34365	1.17811
.77061	2749.73317	647.85936	.73185
.75118	2551.28031	1087.36882	.65529
.73485	1617.79689	542.11576	.25790
.71210	1236.16685	279.93786	.14771
.68522	1246.22698	235.39661	.14938
.66186	459.51438	179.35967	.02107
.64343	28.21175	90.15594	.00033
.62501	-44.49788	31.28733	.00003
.60659	-10.94931	20.64404	.00001
.58817	-4.29036	15.60725	.00001
.56975	-8.13531	10.09347	.00000
.55133	-9.54409	6.36729	.00000
.53291	-9.78615	4.24762	.00000
.51448	-9.97549	2.95824	.00000
.49606	-10.13106	2.15919	.00000
.47764	-10.30262	1.70137	.00000
.45922	-10.36187	1.48736	.00000
.44080	-10.44397	1.48133	.00000
.42238	-10.59108	1.68240	.00000
.40396	-10.71107	2.14023	.00000
.38553	-10.98384	2.96535	.00000
.36711	-11.33168	4.34236	.00000
.34869	-11.95490	6.58576	.00000
.33027	-12.86941	10.20126	.00000
.31185	-14.35236	16.00190	.00001
.29343	-16.73332	25.28981	.00002
.27500	-20.44227	40.12804	.00005
.25658	-26.26506	63.85203	.00013
.23816	-36.02082	101.88828	.00032
.21974	-52.23946	162.51184	.00081
.20132	-73.10681	257.79724	.00205

Interface Stresses:

X	NORMAL	SHEAR	FAILURE
.18290	-95.39534	410.43811	.00518
.16448	-176.83982	671.06683	.01386
.14605	-447.91302	1091.49923	.03667
.12763	-464.99934	1585.29270	.07735
.10921	1245.95591	2233.19115	.30111


```

COMMON/GEOM/NP1,R,W,T
COMMON/COMCHR/PDESC
COMMON/LOAD/FX,FY,MTX
COMMON/MATL/RCRIT,TAUCRT,E
COMMON/ANGLE/THTOT,THTINT,THTFIN,INSP
COMMON/LAYUP/ORIENT,MAT,NMAT
COMMON/MATPRP/E1,E2,G12,ANU,TPLY

C
  BLANK='
  AIROR='ILLEGAL INPUT.  TRY AGAIN.'
C
C.....
C
  CALL INTRO
C
  CALL INPUT
C
  OUTYPE=0
C
  WARN=BLANK
1000 WRITE(6,1010)WARN
1010 FORMAT('1',A,11('/),
&      ' DO YOU WANT OUTPUT FILES (Y/N)?')
C
  WARN=AIROR
  NCHAR=1
  READ(5,1020,ERR=1000)ANSWER
1020 FORMAT(A)
C  READ(5,1020,ERR=1000)NCHAR,ANSWER
C 1020 FORMAT(Q,A)
C
  IF (NCHAR.NE.1) GOTO 1000
  IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'Y')) THEN
    ANSWER='Y'
    WARN=BLANK
1030 WRITE(6,1040)WARN
1040 FORMAT('1',11('/),',A,/,/,
&      ' ENTER THE OUTPUT FILE NAME (FILENAME.OUT)',
&      ' 9 CHARACTERS MAX.')
    WARN=AIROR
    NCHAR=9
    READ(5,1020,ERR=1000)FILOUT
    IF (NCHAR.GT.9) NCHAR=9
    FILOUT=FILOUT(1:NCHAR)//'.OUT'
    WARN='ILLEGAL FILENAME.  TRY AGAIN.'
    OPEN(UNIT=13,FILE=FILOUT,STATUS='NEW',ERR=1030)
C
  ELSE IF ((ANSWER.EQ.'N').OR.(ANSWER.EQ.'N')) THEN
    ANSWER='N'
  ELSE
    GOTO 1000
  END IF
  WARN=BLANK
1022 WRITE(6,1024)WARN
1024 FORMAT('1',8('/),',A,/,/,
&      ' WHAT TYPE OF SCREEN OUTPUT DO YOU WANT?',/,
&      ' 1) FULL: STRESS STATE AT EVERY ANGLE',/,
&      ' 2) TRUNCATED: STRESS STATE ONLY AT FAILURE LOCATION')
C
  WARN=AIROR
  READ(5,*,ERR=1022)OUTYPE
C
  IF ((OUTYPE.LT.1).OR.(OUTYPE.GT.2)) GOTO 1022
C
  N=NP1
C

```

```

      IF (DP.GE.0.0) THEN
        KM=1
      ELSE
        KM=NP1
      END IF
C
      NRHO=NP1*2
C
      DRHO(1)=0.0
      DRHO(2)=0.95*T(1)
      TSUM=T(1)
      II=2
C
      DO 145 I=2,NP1
C
        R(I)=R(I-1)+T(I-1)
        II=II+1
        DRHO(II)=TSUM+(T(I)*0.05)
        II=II+1
        DRHO(II)=TSUM+(T(I)*0.95)
        TSUM=TSUM+T(I)
C
      145 CONTINUE
      DRHO(NRHO)=TSUM
      R(NP1+1)=R(NP1)+T(NP1)
C
C
C
      CALL RORF(X2,X1,AREA,N,T,W,E,R,RO,RF)
C
C
      WRITE(6,90)
      90 FORMAT(20(/))
C
      WRITE(6,100)PDESC
C
      IF (ANSWER.EQ.'Y') THEN
        WRITE(13,100)PDESC
      END IF
C
      100 FORMAT(' ',10X,'---- CURVED LAMINATE ANALYSIS ----',///,
&              ' INPUT DATA',/,34('-',),///,' ',A60,/)
C
      DO 105 I=1,NMAT
C
        WRITE(6,110)I,E1(I),E2(I),G12(I),ANU(I),TPLY(I)
C
        IF (ANSWER.EQ.'Y') THEN
          WRITE(13,110)I,E1(I),E2(I),G12(I),ANU(I),TPLY(I)
        END IF
C
        IF ((NMAT.GT.1).AND.(I.EQ.1)) CALL PAWS
C
      105 CONTINUE
C
      110 FORMAT(' PROPERTIES FOR MATERIAL #',I1,':',/,
&              ' E1 = ',1PE11.4,' PSI',/,
&              ' E2 = ',1PE11.4,' PSI',/,
&              ' G12 = ',1PE11.4,' PSI',/,
&              ' NU = ',0PF4.2,/,
&              ' T = ',1PE11.4,' IN.',/)
C
      WRITE(6,120)RCRIT,TAUCRT
C
      IF (ANSWER.EQ.'Y') THEN
        WRITE(13,120)RCRIT,TAUCRT
      END IF

```



```

C 120 FORMAT(//,' LAMINATE ALLOWABLES:',/,
&      ' - FLATWISE TENSION = ',1PE11.4,' PSI',/,
&      ' - INTERLAMINAR SHEAR = ',1PE11.4,' PSI',/)
C
C      CALL PAWS
C
C      NROWS=NP1/2
C      NFRAC=MOD(NP1,2)
C      J=NROWS+NFRAC
C      III=0
C
C      WRITE(6,125)
C
C      IF (ANSWER.EQ.'Y') THEN
C          WRITE(13,125)
C      END IF
C
C 125 FORMAT(' ',//,2(' PLY ORIENTATION MATERIAL '),/,
&      2(' ',28('-'),1X))
C
C      DO 130 I=1,NROWS
C          JPLUSI=J+I
C
C          WRITE(6,140) I, ORIENT(I), MAT(I),
&          JPLUSI, ORIENT(JPLUSI), MAT(JPLUSI)
C
C          IF (ANSWER.EQ.'Y') THEN
C              WRITE(13,140) I, ORIENT(I), MAT(I),
&              JPLUSI, ORIENT(JPLUSI), MAT(JPLUSI)
C          END IF
C
C          II=I-III
C          IV=NROWS-I
C          IF ((II.EQ.15).AND.(IV.GT.3)) THEN
C              III=I
C              CALL PAWS
C              WRITE(6,125)
C          END IF
C
C 130 CONTINUE
C
C      NROWS=NROWS+1
C      IF (NFRAC.EQ.1) WRITE(6,140) NROWS, ORIENT(NROWS), MAT(NROWS)
C
C      IF ((NFRAC.EQ.1).AND.(ANSWER.EQ.'Y')) WRITE(13,140) NROWS,
&      ORIENT(NROWS), MAT(NROWS)
C
C      CALL PAWS
C
C 140 FORMAT(2(' ',I3,6X,F5.1,8X,I1,6X))
C
C      WRITE(6,150) W, R(1), RO, RF
C
C      IF (ANSWER.EQ.'Y') THEN
C          WRITE(13,150) W, R(1), RO, RF
C      END IF
C
C 150 FORMAT(' ',//,
&      ' THE LAMINATE WIDTH: ',F6.3,/,
&      ' THE LAMINATE INNER RADIUS: ',F6.3,/,
&      ' THE RADIUS OF THE NEUTRAL AXIS: ',F6.3,/,
&      ' THE RADIUS OF THE CENTROID: ',F6.3)
C

```

```

C      WRITE(6,160)THTOT,THTINT,THTFIN,INSP
C
C      IF (ANSWER.EQ.'Y') THEN
C        WRITE(13,160)THTOT,THTINT,THTFIN,INSP
C      END IF
C
C      160 FORMAT(' ',//,
C        &      ' ANGLE INFORMATION:',//,
C        &      ' - THETA TOTAL: ',F5.1,//,
C        &      ' - THETA INITIAL: ',F5.1,//,
C        &      ' - THETA FINAL: ',F5.1,//,
C        &      ' - INSPECTIONS: ',I3)
C
C      WRITE(6,170)FX,FY,MXY,DP
C
C      IF (ANSWER.EQ.'Y') THEN
C        WRITE(13,170)FX,FY,MXY,DP
C      END IF
C
C      170 FORMAT(' ',//,
C        &      ' APPLIED LOADS:',//,
C        &      ' - PX: ',1PE11.4,//,
C        &      ' - PY: ',1PE11.4,//,
C        &      ' - MXY: ',1PE11.4,//,
C        &      ' - PO: ',1PE11.4)
C
C      CALL PAWS
C
C      IF (INSP.GT.1) THEN
C        DELTHT=(THTFIN-THTINT)/REAL(INSP-1)
C      ELSE
C        DELTHT=0.0
C      END IF
C
C      PI=ACOS(-1.0)
C      CONV=PI/180.
C      THTDEG=THTINT
C      THTOT=CONV*THTOT
C      THTINT=CONV*THTINT
C      THTFIN=CONV*THTFIN
C      DELTHT=CONV*DELTHT
C
C      180 FORMAT(' ',////,' ',50('-'),/,6X,
C        &      ' CRITICAL STRESS STATE AT THETA =',F5.1,//,
C        &      ' ',50('-'))
C
C      190 FORMAT(' ',////,50('-'),//,
C        &      ' STRESS STATE AT THETA =',F5.1,//,
C        &      ' INTERFACE RADIAL',8X,
C        &      ' RADIAL',9X,' SHEAR FAILURE',/,
C        &      ' NUMBER POSITION',7X,
C        &      ' STRESS',9X,' STRESS INDEX ',/,
C        &      ' ',61('-'))
C
C      MAXFI=0.0
C      THETA=THTINT
C      IF (DP.GE.0.0) THEN
C        PRSLOD=DP*W*R(1)
C      ELSE
C        PRSLOD=DP*W*R(NP1+1)
C      END IF
C

```

```

C      DO 350 ITHETA=1, INSP
C
C      F=FY*COS (THETA) -FX*SIN (THETA) +PRSLOD*(1.-COS (THETA))
C      V=FX*COS (THETA) +FY*SIN (THETA) -PRSLOD*SIN (THETA)
C      M=MX*Y+(RO*F)+(RO*PRSLOD*(1.-COS (THETA)))
C
C      FF=0.0
C      FMAX=0.0
C      INTN=1
C      INDIC=1
C
C      DO 300 IR=2, NRHO-1
C
C      RHO=R(1)+DRHO (IR)
C      CALL AB (F,M,V,R,W)
C      CALL CS (X2,X1,C1,C2,FA,MA)
C      CALL STR (C1,E,N,R,RHO,RM,C2,T,W,X1,X2)
C
C      IF (INDIC.EQ.1) THEN
C        INDIC=2
C        SIGR1=FWT
C        TAUAVE=SSRHO/2.0
C      ELSE IF (INDIC.EQ.2) THEN
C        INDIC=1
C        SIGRAV=0.5*(SIGR1+FWT)
C        TAUAVE=TAUAVE+(SSRHO/2.0)
C        TAUABS=ABS (TAUAVE)
C
C        IF (SIGRAV.GT.0.0) THEN
C          SR=SIGRAV
C        ELSE
C          SR=0.0
C        END IF
C
C        FF=FI (SR,TAUABS)
C        FMAX=MAX (FF,FMAX)
C
C        IF (FF.EQ.FMAX) THEN
C          RFRC=0.0
C          IF (SIGRAV.GT.0.0) RFRC=SIGRAV/RCRIT
C          TFRC=TAUAVE/TAUCRT
C          NINTF=INTN
C          SIGRF=SIGRAV
C          TAUF=TAUAVE
C        END IF
C
C        RADSTR (INTN,ITHETA)=SIGRAV
C        SHRSTR (INTN,ITHETA)=TAUAVE
C        FAINDX (INTN,ITHETA)=FF
C
C        MAXFI=MAX (MAXFI,FF)
C        IF (MAXFI.EQ.FF) THEN
C          FANGLE=THETA
C          FAINT=INTN
C          FCUT=ITHETA
C        END IF
C
C        IF (OUTYPE.EQ.1) THEN
C          WRITE (13,5555) INTN,R (INTN+1),SIGRAV,TAUAVE,FF
C        END IF
C        INTN=INTN+1
C      END IF
5555  FORMAT ('      ',I3,5X,3(1PE11.4,4X),0PF4.2)
C
C      300 CONTINUE

```

```

C      IF (FMAX.GT.0.0) THEN
          FCTR=1.0/FMAX
          SIGRF=SIGRF*FCTR
          TAUF=TAUF*FCTR
          FXF=FX*FCTR
          FYF=FY*FCTR
          MXYF=MXY*FCTR
          DPF=DP*FCTR
          RFRC=((RFRC*FCTR)**2)*100.
          TFRC=((TFRC*FCTR)**2)*100.
C
          PFI(1,ITHETA)=FMAX
          PFI(2,ITHETA)=FXF
          PFI(3,ITHETA)=FYF
          PFI(4,ITHETA)=MXYF
          PFI(5,ITHETA)=DPF
          PFI(6,ITHETA)=SIGRF
          PFI(7,ITHETA)=TAUF
          PFI(8,ITHETA)=RFRC
          PFI(9,ITHETA)=TFRC
          IPFI(ITHETA)=NINTF
C
7080  FORMAT(//,' THE MAXIMUM FAILURE INDEX IS: ',F4.2,///,
&      ' PREDICTED FAILURE INFORMATION',//,
&      ' - PX: ',1PE11.4,/,
&      ' - PY: ',1PE11.4,/,
&      ' - MXY: ',1PE11.4,/,
&      ' - PRESSURE: ',1PE11.4,/,
&      ' - RADIAL STRESS (PSI): ',1PE10.3,/,
&      ' - SHEAR STRESS (PSI): ',1PE10.3,/,
&      ' - INTERFACE OF FAILURE: ',I2,/,
&      ' CONTRIBUTIONS TO FAILURE',//,
&      ' - RADIAL STRESS: ',0PF5.1,'% ',/,
&      ' - SHEAR STRESS: ',F5.1,'% ')
C
C      IF (OUTYPE.EQ.1) THEN
C          WRITE(13,7080) FMAX,FXF,FYF,MXYF,DPF,SIGRF,
C      &      TAUF,NINTF,RFRC,TFRC
C      END IF
C
C      ELSE
C          PFI(1,ITHETA)=0.0
C      END IF
C
C      THETA=THETA+DELTH
C
350  CONTINUE
C
C      FANGLE=FANGLE/CONV
C
C      WRITE(6,180) FANGLE
C
C      IF (ANSWER.EQ.'Y') THEN
C          WRITE(13,180) FANGLE
C      END IF
C
C      IF (MAXFI.GT.0.0) THEN
          FCTR=1.0/MAXFI
          SIGRF=RADSTR(FAINT,FCUT)*FCTR
          TAUF=SHRSTR(FAINT,FCUT)*FCTR
          FXF=FX*FCTR
          FYF=FY*FCTR
          MXYF=MXY*FCTR
          DPF=DP*FCTR

```

```

RFRC=RADSTR(FAINT,FCUT)/RCRIT
IF (SIGRF.LT.0.0) RFRC=0.0
TFRC=SHRSTR(FAINT,FCUT)/TAUCRT
RFRC=((RFRC*FCTR)**2)*100.0
TFRC=((TFRC*FCTR)**2)*100.0
C
WRITE(6,7080)MAXFI,FXF,FYF,MXYF,DPF,
& SIGRF,TAUF,FAINT,RFRC,TFRC
C
IF (ANSWER.EQ.'Y') THEN
WRITE(13,7080)MAXFI,FXF,FYF,MXYF,DPF,
& SIGRF,TAUF,FAINT,RFRC,TFRC
END IF
ELSE
WRITE(6,375)
C
375 FORMAT(////' NO FAILURE PREDICTED IN THIS LAMINATE')
C
IF (ANSWER.EQ.'Y') THEN
WRITE(13,375)
END IF
END IF
C
CALL PAWS
IF (ANSWER.EQ.'Y') THEN
C
C
DO 425 J=1,INSP
C
IF (PFI(1,J).GT.0.0) THEN
C
IF (OUTYPE.EQ.1) WRITE(6,190)THTDEG
WRITE(13,190)THTDEG
DO 400 I=1,NP1-1
WRITE(13,5555)I,R(I+1),RADSTR(I,J)
& ,SHRSTR(I,J),FAINDX(I,J)
400 CONTINUE
IF (OUTYPE.EQ.1) THEN
DO 401 I=1,NP1-1
WRITE(6,5555)I,R(I+1),RADSTR(I,J),SHRSTR(I,J),FAINDX(I,J)
401 CONTINUE
CALL PAWS
ENDIF
IF (OUTYPE.EQ.1) THEN
WRITE(6,7080)(PFI(K,J),K=1,7),IPFI(J),(PFI(K,J),K=8,9)
CALL PAWS
ENDIF
WRITE(13,7080)(PFI(K,J),K=1,7),IPFI(J),(PFI(K,J),K=8,9)
C
ELSE
IF (OUTYPE.EQ.1) WRITE(6,450)THTDEG
WRITE(13,450)THTDEG
END IF
C
THTDEG=THTDEG+(DELTHT/CONV)
C
425 CONTINUE
C
END IF
C
450 FORMAT(////,' ',50(' - '),////,6X,
& ' NO FAILURE PREDICTED AT THETA =' ,F5.1)
C
IF (ANSWER.EQ.'Y') CLOSE(13)
C
END

```

```

C
C*****
C
C
C      SUBROUTINE RORF (X2,X1,AREA,N,T,W,E,R,RO,RF)
C
C      IMPLICIT REAL (A-H,O-Z)
C
C      DIMENSION T(101),E(101),R(101)
C
C      X2=0.
C      X3=0.
C      X5=0.
C      AREA=0.
C
C      DO 100 IA=1,N
C        R(IA+1)=R(IA)+T(IA)
C        AREA=AREA+W*T(IA)
C        X2=X2+E(IA)*W*T(IA)
C        X3=X3+E(IA)*W*ALOG(1.+T(IA)/R(IA))
100    X5=X5+E(IA)*W*T(IA)*(R(IA+1)+R(IA))/2.
C
C      RO=X2/X3
C      RF=X5/X2
C      X1=0.
C
C      DO 110 IA=1,N
C        B1=W*T(IA)
C        B2=R(IA)-RO+T(IA)/2.
C        B3=E(IA)*B1*B2
C        X1=X1+B3
110    CONTINUE
C
C      PRINT*, ' '
C      RETURN
C      END
C
C
C*****
C
C      SUBROUTINE AB (F,M,V,R,W)
C
C      IMPLICIT REAL (A-H,O-Z)
C      REAL M,MA
C
C      DIMENSION R(101)
C
C      COMMON/LAB001/DG,DV,DP,RF,FA,MA,G,RO,SSRHO,FWT
C      1,STRNTN,STRSTN,VA,KM
C
C      V1=V
C      F1=F
C
C      IF (DP.LT.0.0) THEN
C        DPRW=R(KM+1)*DP*W
C      ELSE
C        DPRW=R(KM)*DP*W
C      END IF
C      FF1=F1-DPRW
C
C      A=FF1
C      B=-1.0*V1
C      FA=A+DPRW
C      VA=-1.0*B
C      FM=0.0
C      VM=0.0

```

```

      DV=FA-DPRW
      G=V1
      DG=FF1
      MA=RF* (FM-VM) +M
C
      RETURN
      END
C
C
C*****
C
      SUBROUTINE CS (X2,X1,C1,C2,FA,MA)
C
      IMPLICIT REAL (A-H,O-Z)
      REAL MA
C
      C1=MA/X1
      C2=FA/X2
C
      RETURN
      END
C
C
C*****
C
      SUBROUTINE STR (C1,E,N,R,RHO,RM,C2,T,W,X1,X2)
C
      IMPLICIT REAL (A-H,O-Z)
      REAL LAMKS,LBR,LMDA,MA
C
      DIMENSION E(101),R(101),T(101),DRL(101),DRI(101),WE(101),
&Z(101),PHI(101),FQ(101),PHISUM(101),PHIS(101),PHISSM(101)
C
      COMMON/LAB001/DG,DV,DP,RF,FA,MA,G,RO,SSRHO,FWT
1,STRNTN,STRSTN,VA,KM
C
      RM=R(1)-.0001
      RN=R(N+1)+.0001
C
      IF (RHO.LT.RM) THEN
C
        WRITE(6,909)
909  FORMAT(T2,'RADIUS OF DESIRED RESULTS IS TOO SMALL!!!!')
C
        LQQ=1
        RETURN
C
      ELSE IF (RHO.GT.RN) THEN
C
        WRITE(6,910)
910  FORMAT(T2,'RADIUS OF DESIRED RESULTS IS TOO LARGE!!!!')
C
        LQQ=1
        RETURN
C
      END IF
C
      WK2=W
C
      WK=W
C
C
      IF (RHO.LT.R(2)) THEN
        K=1
      ELSE IF (RHO.EQ.R(2)) THEN
        K=2
      ELSE
        DO 700 I=2,N

```

```

C      IF (RHO.LT.R(I+1)) THEN
C          WK=W
C          K=I
C          GOTO 710
C      ELSE IF (RHO.EQ.R(I)) THEN
C          WK=W
C          K=I
C          GOTO 710
C      END IF
C 700  CONTINUE
C      IF (RHO.EQ.R(N+1)) THEN
C          WK=W
C          K=N
C          END IF
C      END IF
C 710 CONTINUE
C      IF (RHO.NE.R(K)) WK2=W
C      IF (DP.LT.0.) THEN
C          IF (RHO.LE.R(KM+1)) THEN
C              FDP=0.
C          ELSE
C              FDP=R(KM+1)*W*DP
C          END IF
C      ELSE IF (K.LT.KM) THEN
C          FDP=0.
C      ELSE
C          FDP=R(KM)*W*DP
C      END IF
C 153 YT1=0.
C     YT2=0.
C     YT3=0.
C     ALFA=RF*DG/X1
C     BTA=DV/X2
C     H=(BTA-ALFA)/2.
C     ALRO=ALFA*RO
C     ALS=RF*G/X1
C     BTS=VA/X2
C     HS=(BTS-ALS)/2.
C     ALRS=ALS*RO
C     DRK=RHO-R(K)
C     DRIK=1./RHO-1./R(K)
C     DRKL=ALOG(RHO/R(K))
C     DRKS=RHO**2-R(K)**2
C
C     DO 140 I=1,N
C         DRI(I)=1./R(I+1)-1./R(I)
C         DRL(I)=ALOG(R(I+1)/R(I))
C         WE(I)=W*E(I)
C         PHI(I)=WE(I)*T(I)*(ALRO+H*(R(I+1)+R(I)))
C         PHIS(I)=WE(I)*T(I)*(ALRS+HS*(R(I+1)+R(I)))
C         T1=ALRO*(DRL(I)+R(I)*DRI(I))
C         T2=H*(T(I)+DRI(I)*R(I)**2)
C         Z(I)=WE(I)*(T1+T2)
C 140 CONTINUE

```



```

C      TLA=ALRS*DRK
      TLH=HS*DRKS
      LAMKS=WE (K) * (TLA+TLH)
      TK1=ALRO* (DRKL+R (K) *DRIK)
      TK2=H* (DRK+DRIK*R (K) **2)
      ZR=WE (K) * (TK1+TK2)
      PHISUM(1)=PHI (1)
      PHISSM(1)=PHIS (1)

C      DO 142 I=1,N
      PHISUM(I+1)=PHISUM(I)+PHI (I+1)
      PHISSM(I+1)=PHISSM(I)+PHIS (I+1)
142 CONTINUE

C      FQ(1)=0.
      FQ(2)=Z(1)

C      DO 145 I=2,N
      FQ(I+1)=FQ(I)+Z(I)-DRI(I)*PHISUM(I-1)
145 CONTINUE

C      IF (K.GT.1) THEN
      FT=FQ(K)+ZR-DRIK*PHISUM(K-1)
      SSRHO=(PHISSM(K-1)+LAMKS)/(WK2*RHO**2)
      ELSE
      FT=ZR
      SSRHO=LAMKS/(W*RHO**2)
      END IF

C      FSTSUM=0.
      DO 162 I=1,K-1
      TM=C1*(RO*DRI(I)-T(I))
      TF=FA*T(I)/X2
      FST=WE(I)*(TM+TF)
162 FSTSUM=FSTSUM+FST
      TMK=C1*(RO*DRKL-(RHO-R(K)))
      TFK=FA*(RHO-R(K))/X2
      FSTR=WE(K)*(TMK+TFK)

C      IF (K.NE.1) THEN
      FWT=(FSTR+FSTSUM-FT-FDP)/(RHO*WK)

C      ELSE
180 FWT=(FSTR-FT-FDP)/(RHO*W)
      WK1=W
      WK2=W
      RK1=R(K)-.01
      RK2=R(K+1)+.01

C      IF ((RHO.NE.R(K)).AND.(RHO.NE.R(N+1)).AND.
& (K.NE.N).AND.(WK2.LT.WK1)) WRITE(6,906)RK2
906 FORMAT(T2,'CHECK RADIAL STRESS AT RADIUS =',F8.6)

C      END IF

C      S1=C1*(RO-RHO)/RHO
      STRNTN=S1+C2

C      IF (RHO.LE.R(N)) THEN
      STRSTN=STRNTN*E(K)
      ELSE
      STRSTN=STRNTN*E(N)
      END IF

C      303 CONTINUE

```

```

C      RETURN
C
C      END
C
C .....
C      REAL FUNCTION FI (SIGR,TAU)
C
C      IMPLICIT REAL (A-H,O-Z)
C
C      DIMENSION E(101)
C
C      COMMON/MATL/RCRIT,TAUCRT,E
C
C      FULT=(SIGR/RCRIT)**2+(TAU/TAUCRT)**2
C      FI=SQRT(FULT)
C      RETURN
C      END
C
C .....
C      SUBROUTINE INPUT
C
C      IMPLICIT REAL (A-H,O-Z)
C      REAL MA,MXY
C      CHARACTER*26 AIROR,BLANK,WARN
C      CHARACTER*60 PDESC
C
C      DIMENSION T(101),ORIENT(101),E(101),R(101),MAT(101)
C      DIMENSION E1(3),E2(3),G12(3),ANU(3),TPLY(3)
C
C      COMMON/LAB001/DG,DV,DP,RF,FA,MA,G,RO,SSRHO,FWT
C      1,STRNTN,STRSTN,VA,KM
C      COMMON/GEOM/NP1,R,W,T
C      COMMON/COMCHR/PDESC
C      COMMON/LOAD/FX,FY,MXY
C      COMMON/MATL/RCRIT,TAUCRT,E
C      COMMON/ANGLE/THTOT,THTINT,THTFIN,INSP
C      COMMON/LAYUP/ORIENT,MAT,NMAT
C      COMMON/MATPRP/E1,E2,G12,ANU,TPLY
C
C      AIROR='ILLEGAL INPUT. TRY AGAIN.'
C      BLANK='
C
C      WARN=BLANK
C      2 WRITE(6,4)WARN
C      4 FORMAT('1',20(/),' ',A,////,
C      &      ' ENTER THE PROBLEM DESCRIPTION (60 CHARACTERS MAX.)',
C      &      10X,':')
C
C      WARN=AIROR
C      NCHAR=1
C      READ(5,6,ERR=2)PDESC
C      6 FORMAT(A)
C      READ(5,6,ERR=2)NCHAR,PDESC
C      6 FORMAT(Q,A)
C      IF (NCHAR.GT.60) GOTO 2
C
C
C      WARN=BLANK
C      8 WRITE(6,10)WARN

```

```

10 FORMAT('1',20('/),' ',A,////,
& ' ENTER THE NUMBER OF MATERIALS (3 MAX.)')
WARN=AIOR
READ(5,*,ERR=8)NMAT
IF ((NMAT.LT.1).OR.(NMAT.GT.3)) GOTO 8
C
DO 16 I=1,NMAT
  WARN=BLANK
12  WRITE(6,14) WARN,I
14  FORMAT('1',20('/),' ',A,////,
& ' ENTER THE PROPERTIES FOR MATERIAL #',I1,/,
& ' - E1, MODULUS IN FIBRE DIRECTION (MSI)',/,
& ' - E2, MODULUS IN TRANSVERSE DIRECTION (MSI)',/,
& ' - G12, SHEAR MODULUS OF PLY (MSI)',/,
& ' - NU, POISSON RATIO OF PLY',/,
& ' - T, THICKNESS OF THE PLY',/)
  WARN=AIOR
  READ(5,*,ERR=12)ET1,ET2,GT12,ANUT,TT
  IF ((ET1.LE.0.0).OR.(ET2.LE.0.0).OR.(GT12.LE.0.0).OR.
& (ANUT.LE.0.0).OR.(ANUT.GT.0.5).OR.
& (TT.LE.0.0)) GOTO 12
C
  E1(I)=ET1*1.0E+06
  E2(I)=ET2*1.0E+06
  G12(I)=GT12*1.0E+06
  ANU(I)=ANUT
  TPLY(I)=TT
  WARN=BLANK
16 CONTINUE
C
C
270 WRITE(6,280) WARN
280 FORMAT('1',20('/),' ',A,////,
& ' ENTER THE LAMINATE ALLOWABLES.',/,
& ' - FLATWISE TENSION (KSI)',/,
& ' - INTERLAMINAR SHEAR (KSI)')
C
  WARN=AIOR
  READ(5,*,ERR=270)RCRIT,TAUCRT
C
  IF ((RCRIT.LE.0.0).OR.(TAUCRT.LE.0.0)) GOTO 270
C
  RCRIT=RCRIT*1000.
  TAUCRT=TAUCRT*1000.
C
C
  WARN=BLANK
18 WRITE(6,20) WARN
20 FORMAT('1',20('/),' ',A,////,
& ' ENTER THE TYPE OF LAYUP.',/,
& ' 1: UNSYMMETRIC [A1,A2,...,AN]',/,
& ' 2: ODD SYMMETRIC [A1,A2,A3,...,AN]S',/,
& ' 3: EVEN SYMMETRIC [A1,A2,A3,...,AN]MS')
C
  WARN=AIOR
  READ(5,*,ERR=18)IOPT
C
  IF ((IOPT.LT.1).OR.(IOPT.GT.3)) GOTO 18
  WARN=BLANK
C
C.....
C
  ORIENT(1)=0.0
  T(1)=0.0
  MAT(1)=1
C

```

```

      IF (IOPT.EQ.1) THEN
C... :
C      UNSYMMETRIC LAMINATE
      30  WRITE(6,40) WARN
      40  FORMAT('1',20(/),' ',A,////,
      &      ' UNSYMMETRIC LAYUP [A1,A2,...,AN]',/,/,
      &      ' ENTER THE NUMBER OF PLIES (2 <= N <= 100)')
      WARN=AIOR
      READ(5,*,ERR=30) NO
C
      IF ((NO.LT.2).OR.(NO.GT.100)) GOTO 30
      WARN=BLANK
C
      WRITE(6,155)
C
      DO 60 I=1,NO
      50  WRITE(6,70) WARN,I
      WARN=AIOR
C
      IF (I.GT.1) THEN
          ORIENT(I)=ORIENT(I-1)
          T(I)=T(I-1)
          MAT(I)=MAT(I-1)
      END IF
C
      READ(5,*,ERR=50) ORIENT(I),MAT(I)
      OI=ORIENT(I)
      IF ((OI.LE.-90.).OR.(OI.GT.90.).OR.(MAT(I).LT.1)
      &      .OR.(MAT(I).GT.NMAT)) GOTO 50
C
      T(I)=TPLY(MAT(I))
      WARN=BLANK
      60  CONTINUE
      70  FORMAT('0',A,/,/,
      &      ' ENTER THE ORIENTATION (IN DEGREES)',/,/,
      &      ' AND MATERIAL IDENTIFICATION FOR PLY #',I2)
      NP1=NO
C
C.....
      ELSE IF (IOPT.EQ.2) THEN
C... :
C      ODD SYMMETRIC LAMINATE
      80  WRITE(6,90) WARN
      90  FORMAT('1',20(/),' ',A,////,
      &      ' ',41X,' ',/,/,
      &      ' ODD SYMMETRIC LAYUP [A1,A2,A3,...,AN]S',/,/,
      &      ' ENTER THE NUMBER OF PLY ORIENTATIONS',
      &      ' (2 <= N <= 50)')
      WARN=AIOR
      READ(5,*,ERR=80) NO
C
      IF ((NO.LT.2).OR.(NO.GT.50)) GOTO 80
      WARN=BLANK
C
      WRITE(6,155)
C
      DO 110 I=1,NO
      100 WRITE(6,70) WARN,I
      WARN=AIOR
C
      IF (I.GT.1) THEN
          ORIENT(I)=ORIENT(I-1)
          T(I)=T(I-1)
          MAT(I)=MAT(I-1)
      END IF
C

```

```

      READ (5,*,ERR=100) ORIENT(I),MAT(I)
      OI=ORIENT(I)
      IF ((OI.LE.-90.).OR.(OI.GT.90.))
&      .OR.(MAT(I).LT.1).OR.(MAT(I).GT.NMAT)) GOTO 100
      T(I)=TPLY(MAT(I))
      WARN=BLANK
      CALL ODSYM(ORIENT,NO)
      CALL ODSYM(T,NO)
      CALL IODSYM(MAT,NO)
      NP1=(NO*2)-1
110  CONTINUE
C
C.....
      ELSE
C...      EVEN SYMMETRIC LAMINATE
C
120  WRITE(6,130) WARN
130  FORMAT('1',20(/),' ',A,////,
&      ' EVEN SYMMETRIC LAYUP [A1,A2,A3,...AN]MS',/,
&      ' ENTER THE NUMBER OF ORIENTATIONS (N) AND THE',/,
&      ' NUMBER OF SERIES (M) (M X N <= 50)')
      WARN=AIOR
      READ (5,*,ERR=120) NO,NS
      MN=NO*NS
      IF ((NO.LT.1).OR.(NS.LT.1).OR.(MN.GT.50)) GOTO 120
C
      WRITE(6,155)
C
      WARN=BLANK
      DO 150 I=1,NO
140  WRITE(6,70) WARN,I
      WARN=AIOR
C
      IF (I.GT.1) THEN
          ORIENT(I)=ORIENT(I-1)
          T(I)=T(I-1)
          MAT(I)=MAT(I-1)
      END IF
C
      READ (5,*,ERR=140) ORIENT(I),MAT(I)
      OI=ORIENT(I)
      IF ((OI.LE.-90.).OR.(OI.GT.90.))
&      .OR.(MAT(I).LT.1).OR.(MAT(I).GT.NMAT)) GOTO 140
      T(I)=TPLY(MAT(I))
      WARN=BLANK
150  CONTINUE
      CALL EVSYM(ORIENT,NO,NS)
      CALL EVSYM(T,NO,NS)
      CALL IEVSYM(MAT,NO,NS)
      NP1=2*NO*NS
      END IF
C
C.....
155  FORMAT('1',20(/),
&      ' ENTERING A BACKSLASH (/) WITHIN THE FOLLOWING DATA',/,
&      ' RECORDS ASSUMES THE REMAINING ITEMS TAKE THE VALUES',/,
&      ' FROM THE PREVIOUS DATA RECORD.')
C
C
C
      WARN=BLANK
160  WRITE(6,170) WARN
170  FORMAT('1',20(/),' ',A,////,
&      ' ENTER THE LAMINATE WIDTH (IN.)')
C
      WARN=AIOR

```

```

      READ(5,*,ERR=160)W
      IF (W.LE.0.0) GOTO 160
C
      WARN=BLANK
      180 WRITE(6,190)WARN
      190 FORMAT('1',20('/),' ',A,////,
&          ' ENTER THE INSIDE RADIUS OF THE CURVED LAMINATE (IN.)')
C
      WARN=AIOR
      READ(5,*,ERR=180)R(1)
      IF (R(1).LE.0.0) GOTO 180
      WARN=BLANK
C
      200 WRITE(6,210)WARN
      210 FORMAT('1',20('/),' ',A,////,
&          ' ENTER THE NECESSARY ANGLES (DEGREES)',/,
&          ' - THETA TOTAL: ANGLE SUBTENDED BY THE',
&          ' CURVED LAMINATE',/,
&          ' - THETA INIT.: INITIAL ANGLE OF INSPECTION',/,
&          ' - THETA FINAL: FINAL ANGLE OF INSPECTION',/,
&          ' - INSP: NUMBER OF INSPECTIONS',/,
&          ' (FOR ONLY ONE ANGLE OF INSPECTION AT THETA INITIAL,',
&          ' ENTER INSP = 1)')
C
      WARN=AIOR
C
      READ(5,*,ERR=200)THTOT,THTINT,THTFIN,INSP
C
      IF ((THTOT.LE.0.0).OR.(THTOT.GT.90.0).OR.(THTINT.LT.0.0)
&      .OR.(THTINT.GT.THTOT).OR.(THTFIN.LT.THTINT).OR.
&      (THTFIN.GT.THTOT).OR.(INSP.LT.1).OR.(INSP.GT.20)) GOTO 200
C
      WARN=BLANK
C
      DO 240 I=1,NP1
      MM=MAT(I)
      E(I)=CJONES(E1(MM),E2(MM),G12(MM),ANU(MM),ORIENT(I))
      240 CONTINUE
C
      290 WRITE(6,300)WARN
      300 FORMAT('1',20('/),' ',A,////,
&          ' ENTER THE LOADS RESOLVED AT THE CENTER OF CURVATURE.',/,
&          ' - PX (LBS)',/,
&          ' - PY (LBS)',/,
&          ' - MXY (IN-LBS)')
C
      WARN=AIOR
      READ(5,*,ERR=290)FX,FY,MXY
C
      WARN=BLANK
      310 WRITE(6,320)WARN
      320 FORMAT('1',20('/),' ',A,////,
&          ' ENTER THE PRESSURE DIFFERENCE (PSI)',/,
&          ' (POSITIVE VALUE FOR OUTWARD ACTING PRESSURE.)')
C
      WARN=BLANK
      READ(5,*,ERR=310)DP
C
      END
C
C
C-----
C
      SUBROUTINE ODSYM(VAR,NO)
C
      IMPLICIT REAL (A-H,O-Z)

```

```

C      DIMENSION VAR(101)
C
C      N2=NO*2
C      NP1=N2-1
C      N1=NP1-NO
C      J=NO-1
C
C      DO 30 I=N1,1,-1
C          VAR(N2-I)=VAR(J)
C          J=J-1
C      30 CONTINUE
C
C      RETURN
C      END
C
C-----
C      SUBROUTINE IODSYM(IVAR,NO)
C
C      IMPLICIT REAL (A-H,O-Z)
C
C      DIMENSION IVAR(101)
C
C      N2=NO*2
C      NP1=N2-1
C      N1=NP1-NO
C      J=NO-1
C
C      DO 30 I=N1,1,-1
C          IVAR(N2-I)=IVAR(J)
C          J=J-1
C      30 CONTINUE
C
C      RETURN
C      END
C
C-----
C      SUBROUTINE EVSYM(VAR,NO,NS)
C
C      IMPLICIT REAL (A-H,O-Z)
C
C      DIMENSION VAR(101)
C
C      NP1=2*NS*NO
C      K=1
C      M=NP1
C
C      DO 50 I=1,NS
C          DO 30 J=1,NO
C              VAR(K)=VAR(J)
C              K=K+1
C          30 CONTINUE
C
C          DO 40 J=1,NO
C              VAR(M)=VAR(J)
C              M=M-1
C          40 CONTINUE
C
C          50 CONTINUE
C
C      RETURN
C      END
C
C-----
C      SUBROUTINE IEVSYM(IVAR,NO,NS)

```

```

C      IMPLICIT REAL (A-H,O-Z)
C
C      DIMENSION IVAR(101)
C
C      NP1=2*NS*NO
C      K=1
C      M=NP1
C
C      DO 50 I=1,NS
C        DO 30 J=1,NO
C          IVAR(K)=IVAR(J)
C          K=K+1
30      CONTINUE
C
C        DO 40 J=1,NO
C          IVAR(M)=IVAR(J)
C          M=M-1
40      CONTINUE
C
C      50 CONTINUE
C
C      RETURN
C      END
C
C-----
C      REAL FUNCTION CJONES(E1,E2,G12,ANU,ORIENT)
C
C      IMPLICIT REAL (A-H,O-Z)
C
C
C      PI=3.141592654
C      ANGLE=(PI/180.)*ORIENT
C
C      CN=COS(ANGLE)
C      SN=SIN(ANGLE)
C
C      A1=(CN**4)/E1
C      A2=(1./G12)-(2.*ANU)/E1
C      A2=A2*(SN**2)*(CN**2)
C      A3=(SN**4)/E2
C
C      A4=A1+A2+A3
C
C      CJONES=1.0/A4
C
C      RETURN
C      END
C
C-----
C      SUBROUTINE PAWS
C
C      CHARACTER*1 ANSWER
C
C      WRITE(6,100)
100  FORMAT(/, 'HIT <RETURN> TO CONTINUE'./)
C
C      READ(5,110) ANSWER
110  FORMAT(A)
C
C      WRITE(6,120)
120  FORMAT(20(/))
C
C      RETURN

```


END

C

C-----

SUBROUTINE INTRO

C

INTEGER I

CHARACTER*54 TEXT(25)

C

```

TEXT(1)='-----'
TEXT(2)='          CURVED LAMINATE ANALYSIS'
TEXT(3)='          VERSION 1.0'
TEXT(4)=' '
TEXT(5)='          DEVELOPED BY MCAIR'
TEXT(6)=' '
TEXT(7)='          ANALYSIS DEVELOPMENT:  JOHN M. BUBB'
TEXT(8)='                                PHILIP C. PAUL'
TEXT(9)=' '
TEXT(10)='-----'
TEXT(11)=' '
TEXT(12)=' THIS FORTRAN COMPUTER PROGRAM IS THE AUTOMATED'
TEXT(13)=' VERSION OF THE CURVED LAMINATE ANALYSIS METHOD.'
TEXT(14)=' THE METHOD IS BASED ON THE CURVED STIFFENER'
TEXT(15)=' ANALYSIS, CREATED BY JOHN BUBB. CLA DETERMINES'
TEXT(16)=' THE PLY BY PLY STRESS DISTRIBUTION AT ANY ANGULAR'
TEXT(17)=' POSITION IN THE CURVED LAMINATE. ORTHOTROPIC'
TEXT(18)=' MATERIALS, ARBITRARY LAYUPS, AND A GENERAL'
TEXT(19)=' LOADING ARE ALL PERMITTED BY CLA. THIS PROGRAM'
TEXT(20)=' WAS DEVELOPED UNDER THE NAVY CRAD "OUT-OF-PLANE'
TEXT(21)=' ANALYSIS FOR COMPOSITE STRUCTURES" (CONTRACT NO.'
TEXT(22)=' N62269-87-C-0226). FOR FURTHER INFORMATION, SEE'
TEXT(23)=' THE FINAL REPORT, OR CONTACT PHILIP C. PAUL.'
TEXT(24)=' '
TEXT(25)='-----'

```

C

```

DO 10 I=1,25
  WRITE(6,20)TEXT(I)
  IF ((I.EQ.10).OR.(I.EQ.25)) THEN
    CALL PAWS
    IF (I.NE.25) WRITE(6,20)TEXT(I)
  END IF
10 CONTINUE
20 FORMAT(' ',A54)
RETURN
END

```

2. Interfacial Stress Field Model (ISFM)

PROGRAM ISFM

C

C-----

Interfacial Stress Field Model

Version 1.1

C

C

Developed by MCAIR

C

C

Analysis Development: Ken B. Sanger
Philip C. Paul

C

C-----

C

C

COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS
RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
DOUGLAS CORPORATION.

C

C

C

C

C This material may be reproduced by or for the
C U.S. Government pursuant to the copyright license
C under the clause at 52.227-7013 (May 1981).
C
C
C
C-----

C... This FORTRAN computer program is the automated
C version of the Interfacial Stress Field Model.
C This method was derived to determine the
C interfacial stress state of the cocured or
C adhesively bonded joints of integrally stiffened
C composite structures. The analysis can be used
C either on panels under direct loads or on panels
C under indirect (buckling) loads. This program
C was developed under the Navy CRAD "Out-of-Plane
C Analysis for Composite Structures" (Contract No.
C N62269-87-C-0226). For further information, see
C the final report, or contact Philip C. Paul.
C
C-----

C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C DOUBLE PRECISION K,L,MOMENT
C CHARACTER*55 PDESCR
C CHARACTER*13 FILNAM
C CHARACTER*1 OUTOPT
C
C DIMENSION XPOS(51),SIGMA(51),TAU(51),WF(51),WS(51)
C
C COMMON/ALFA/OUTOPT,FILNAM,PDESCR
C COMMON/NUMRC1/IANTYP,D,L,TF,TS,DF,DS
C COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAU1LS,P,MOMENT,DELTAF
C COMMON/NUMRC3/XPOS,SIGMA,TAU,WF,WS
C
C
C CALL INTRO
C
C CALL INPUT
C
C INUT=6
C CALL ECHO(INUT)
C
C IF (OUTOPT.EQ.'Y') THEN
C OPEN(UNIT=1,FILE=FILNAM,STATUS='NEW')
C INUT=1
C CALL ECHO(INUT)
C END IF
C
C J=0
C WRITE(6,1000)
1000 FORMAT(' PLEASE WAIT A MOMENT FOR THE OUTPUT DATA....')
C 1 CALL ANALYZ
C
C IF (J.EQ.1) GOTO 2
C INUT=6
C CALL OUTPUT(INUT,J)
C IF (OUTOPT.EQ.'Y') THEN
C INUT=1
C CALL OUTPUT(INUT,J)
C ENDIF
C J=1
C 2 CALL FAYLPD(FIMAX)
C
C IF (FIMAX.NE.1.0) GOTO 1
C INUT=6

```

      CALL OUTPUT (INUT, J)
C
      IF (OUTOPT.EQ.'Y') THEN
        INUT=1
        CALL OUTPUT (INUT, J)
        CLOSE (1)
      END IF
C
      STOP
      END
C
C-----
      SUBROUTINE INTRO
C
      INTEGER I
      CHARACTER*55 TEXT(36)
C
      TEXT(1)='-----'
      TEXT(2)='          Interfacial Stress Field Model'
      TEXT(3)='          Version 1.1'
      TEXT(4)=' '
      TEXT(5)='          Developed by MCAIR'
      TEXT(6)=' '
      TEXT(7)='          Analysis Development: Ken B. Sanger'
      TEXT(8)='          Philip C. Paul'
      TEXT(9)=' '
      TEXT(10)='-----'
      TEXT(11)=' '
      TEXT(12)='          COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS'
      TEXT(13)='          RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL'
      TEXT(14)='          DOUGLAS CORPORATION.'
      TEXT(15)=' '
      TEXT(16)=' '
      TEXT(17)='          THIS MATERIAL MAY BE REPRODUCED BY OR FOR THE'
      TEXT(18)='          U.S. GOVERNMENT PURSUANT TO THE COPYRIGHT LICENSE'
      TEXT(19)='          UNDER THE CLAUSE AT 52.227-7013 (MAY 1981).'
      TEXT(20)=' '
      TEXT(21)='-----'
      TEXT(22)=' '
      TEXT(23)='          This FORTRAN computer program is the automated'
      TEXT(24)='          version of the Interfacial Stress Field Model.'
      TEXT(25)='          This method was derived to determine the'
      TEXT(26)='          interfacial stress state of the cocured or'
      TEXT(27)='          adhesively bonded joints of integrally stiffened'
      TEXT(28)='          composite structures. The analysis can be used'
      TEXT(29)='          either on panels under direct loads or on panels'
      TEXT(30)='          under indirect (buckling) loads. This program'
      TEXT(31)='          was developed under the Navy CRAD "Out-of-Plane'
      TEXT(32)='          Analysis for Composite Structures" (Contract No.'
      TEXT(33)='          N62269-87-C-0226). For further information, see'
      TEXT(34)='          the final report, or contact Philip C. Paul.'
      TEXT(35)=' '
      TEXT(36)='-----'
C
      WRITE (6,1000)
1000 FORMAT(5(/))
C
      DO 1010 I=1,36
        WRITE (6,1020) TEXT(I)
        IF ((I.EQ.10).OR.(I.EQ.21).OR.(I.EQ.36)) THEN
          CALL PAWS
          IF (I.NE.36) WRITE (6,1020) TEXT(I)
        END IF
1010 CONTINUE
1020 FORMAT(' ',A55)
      RETURN

```

```

      END
C
C-----
C
      SUBROUTINE PAWS
C
      CHARACTER*1 ANSWER
C
      WRITE(6,100)
100  FORMAT(/,'          HIT <RETURN> TO CONTINUE',/)
C
      . READ(5,110)ANSWER
110  FORMAT(A)
C
      WRITE(6,120)
120  FORMAT(20(/))
      RETURN
      END
C
C-----
C
      SUBROUTINE INPUT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,L,MOMENT
      CHARACTER*55 PDESCR
      CHARACTER*26 AIROR,BLANK,WARN
      CHARACTER*13 FILNAM
      CHARACTER*1 OUTOPT
C
      COMMON/ALFA/OUTOPT,FILNAM,PDESCR
      COMMON/NUMRC1/IAN TYP,D,L,TF,TS,DF,DS
      COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAUILS,P,MOMENT,DELTAF
C
      * ENTER INPUT PARAMETERS *
C
      BLANK='
      AIROR='ILLEGAL INPUT.  TRY AGAIN.'
C
      WARN=BLANK
1000 WRITE(6,1010)WARN
1010 FORMAT('1',A26,////,
&          '  WHAT TYPE OF ANALYSIS DO YOU WISH TO PERFORM?',/,
&          '      1) DIRECT LOADING ANALYSIS',/,
&          '      2) INDIRECT LOADING (POSTBUCKLING) ANALYSIS')
      WARN=AIROR
C
      READ(5,*,ERR=1000)IAN TYP
      IF ((IAN TYP.LT.1).OR.(IAN TYP.GT.2)) GOTO 1000
C
      WARN=BLANK
1015 WRITE(6,1020)WARN
1020 FORMAT('1',A26,////,
&          '  ENTER THE PROBLEM DESCRIPTION (55 CHAR. MAX.)',
&          9X,':')
      WARN=AIROR
C
      READ(5,1180,ERR=1015)PDESCR
C
      WARN=BLANK
1025 WRITE(6,1030)WARN
1030 FORMAT('1',A26,////,
&          '  ENTER THE GEOMETRY INFORMATION',/,
&          '      - D: FLANGE LENGTH',/,
&          '      - L: TOTAL SKIN LENGTH',/,
&          '      - TF: FLANGE THICKNESS',/,

```

```

&      '      - TS: SKIN THICKNESS')
WARN=AIOR
C
READ(5,*,ERR=1025)D,L,TF,TS
IF ((D.LE.0.0).OR.(L.LE.0.0).OR.(TF.LE.0.0)
&    .OR.(TS.LE.0.0)) GOTO 1025
C
WARN=BLANK
1040 WRITE(6,1050)WARN
1050 FORMAT('1',A26,////,
&      ' ENTER THE ELEMENT STIFFNESSES',/,
&      '      - DF: FLANGE STIFFNESS',/,
&      '      - DS: SKIN STIFFNESS')
WARN=AIOR
READ(5,*,ERR=1040)DF,DS
IF ((DF.LE.0.0).OR.(DS.LE.0.0)) GOTO 1040
C
WARN=BLANK
1060 WRITE(6,1070)WARN
1070 FORMAT('1',A26,////,
&      ' ENTER THE X-DIRECTION MODULI',/,
&      '      - EF: FLANGE MODULUS',/,
&      '      - ES: SKIN MODULUS')
C
WARN=AIOR
READ(5,*,ERR=1060)EF,ES
IF ((EF.LE.0.0).OR.(ES.LE.0.0)) GOTO 1060
C
WARN=BLANK
1080 WRITE(6,1090)WARN
1090 FORMAT('1',A26,////,
&      ' ENTER THE INTERFACE MATERIAL INFORMATION',/,
&      '      - K: ELASTIC FOUNDATION MODULUS')
C
C SHEAR LAG FORMULATION DOES NOT PROPERLY CALCULATE SHEAR STRESS IGNORING
C IN THIS VERSION OF ISFM.
C
G=800000.
B=0.005
WARN=AIOR
READ(5,*,ERR=1080)K
IF ((K.LE.0.0).OR.(G.LE.0.0).OR.(B.LE.0.0)) GOTO 1080
C
WARN=BLANK
1100 WRITE(6,1110)WARN
1110 FORMAT('1',A26,////,
&      ' ENTER THE INTERFACE MATERIAL STRESS ALLOWABLE',/,
&      '      - FWT: FLATWISE TENSION ALLOWABLE')
WARN=AIOR
TAUCLS=11000000.
READ(5,*,ERR=1100)PULT
IF ((PULT.LE.0.0).OR.(TAUCLS.LE.0.0)) GOTO 1100
C
WARN=BLANK
IF (IANTYP.EQ.1) THEN
1120 WRITE(6,1130)WARN
1130 FORMAT('1',A26,////,
&      ' ENTER THE APPLIED LOADS',/,
&      '      - P: PULL OFF LOAD',/,
&      '      - M: MOMENT')
READ(5,*,ERR=1120)P,MOMENT
IF (P.LT.0.0) GOTO 1120
IF ((P.EQ.0.0).AND.(MOMENT.EQ.0.0)) GOTO 1120
ELSE IF (IANTYP.EQ.2) THEN
1140 WRITE(6,1150)WARN
1150 FORMAT('1',A26,////,

```

```

&      ' ENTER THE BUCKLE DEFLECTION (DELTA) '
      WARN=AIOR
      READ(5,*,ERR=1140)DELTA
      IF (DELTA.LE.0.0) GOTO 1140
      END IF
C
      WARN=BLANK
1160  WRITE(6,1170)WARN
1170  FORMAT('1',A26,////,
&      ' DO YOU WANT AN OUTPUT FILE (Y/N)?')
      WARN=AIOR
      READ(5,1180,ERR=1160)OUTOPT
1180  FORMAT(A)
      IF ((OUTOPT.NE.'Y').AND.(OUTOPT.NE.'y')).AND.
&      (OUTOPT.NE.'N').AND.(OUTOPT.NE.'n')) GOTO 1160
C
      IF ((OUTOPT.EQ.'Y').OR.(OUTOPT.EQ.'y')) THEN
        OUTOPT='Y'
        WARN=BLANK
1190  WRITE(6,1200)WARN
1200  FORMAT('1',A26,////,
&      ' ENTER THE FILE NAME (9 CHARACTERS MAX.)')
        WARN=AIOR
        READ(5,1180,ERR=1190)FILNAM
C
        DO 1210 I=9,1,-1
          IF (FILNAM(I:I).NE.' ') GOTO 1220
1210  CONTINUE
1220  FILNAM=FILNAM(1:I)//'.OUT'
        ELSE
          OUTOPT='N'
        END IF
C
      RETURN
      END
C
C-----
C
      SUBROUTINE ECHO(INUT)
C
C      * ECHO INPUT *
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,L,MOMENT
      INTEGER I
      CHARACTER TYME*8,DAYT*9
      CHARACTER*55 TEXT(4)
      CHARACTER*55 PDESCR
      CHARACTER*13 FILNAM
      CHARACTER*1 OUTOPT
C
      COMMON/ALFA/OUTOPT,FILNAM,PDESCR
      COMMON/NUMRC1/IANTYP,D,L,TF,TS,DF,DS
      COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAUILS,P,MOMENT,DELTA
C
      TEXT(1)='-----'
      TEXT(2)='          Interfacial Stress Field Model'
      TEXT(3)='          Version 1.1'
      TEXT(4)='-----'
C
      IF (INUT.NE.6) THEN
        DO 1000 I=1,4
          WRITE(INUT,1010)TEXT(I)
1000  CONTINUE
1010  FORMAT(' ',A55)
      ELSE

```

```

      TEXT(2)='                               INPUT DATA ECHO'
      WRITE(INUT,1010)TEXT(1)
      WRITE(INUT,1010)TEXT(2)
      END IF
C
      WRITE(INUT,1015)PDESCR
1015  FORMAT(////,' ',A55)
C
C...THIS SECTION MUST BE COMMENTED OUT
C   WHEN NOT COMPILED WITH VAX FORTRAN
C
C      CALL DATE(DAYT)
C      CALL TIME(TIME)
C      CALL TIMERS(TIME,DAYT)
C
      WRITE(INUT,1020)DAYT,TIME
1020  FORMAT(/,' ',40X,'DATE: ',A9,
&        /,' ',40X,'TIME: ',A8)
C
C...END OF VAX SPECIFIC SECTION
C
      IF (IANTYP.EQ.1) THEN
        WRITE(INUT,1030)
1030  FORMAT(//,' --- DIRECT LOADING ANALYSIS ---')
      ELSE
        WRITE(INUT,1040)
1040  FORMAT(//,' --- INDIRECT LOADING (POSTBUCKLING)',
&        ' ANALYSIS ---')
      END IF
C
      WRITE(INUT,1050)D,L,TF,TS
1050  FORMAT(/,' GEOMETRY INFORMATION',/,
&        ' - FLANGE LENGTH (D): ',1PE11.4,/,
&        ' - TOTAL SKIN LENGTH (L): ',1PE11.4,/,
&        ' - FLANGE THICKNESS (TF): ',1PE11.4,/,
&        ' - SKIN THICKNESS (TS): ',1PE11.4)
C
      WRITE(INUT,1060)DF,DS
1060  FORMAT(/,' ELEMENT STIFFNESSES',/,
&        ' - FLANGE STIFFNESS (DF): ',1PE11.4,/,
&        ' - SKIN STIFFNESS (DS): ',1PE11.4)
C
      IF (INUT.EQ.6) CALL PAWS
C
      WRITE(INUT,1080)EF,ES
1080  FORMAT(/,' X-DIRECTION MODULI',/,
&        ' - FLANGE MODULUS (EF): ',1PE11.4,/,
&        ' - SKIN MODULUS (ES): ',1PE11.4)
C
      WRITE(INUT,1090)K
1090  FORMAT(/,' INTERFACE MATERIAL INFORMATION',/,
&        ' - ELASTIC FOUNDATION MODULUS (K): ',1PE11.4)
C
      WRITE(INUT,1100)PULT
1100  FORMAT(/,' INTERFACE MATERIAL STRESS ALLOWABLE',/,
&        ' - FLATWISE TENSION ALLOWABLE (FWT): ',1PE11.4)
C
      IF (IANTYP.EQ.1) THEN
        WRITE(INUT,1110)P,MOMENT
1110  FORMAT(/,' APPLIED LOADS',/,
&        ' - PULL OFF LOAD (P): ',1PE11.4,/,
&        ' - MOMENT (M): ',1PE11.4)
      ELSE IF (IANTYP.EQ.2) THEN
        WRITE(INUT,1120)DELTAF
1120  FORMAT(/,' BUCKLE DEFLECTION (DELTA): ',1PE11.4)
      END IF

```

```

C      IF (INUT.EQ.6) CALL PAWS
C      RETURN
C      END
C-----
C      SUBROUTINE ANALYZ
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      DOUBLE PRECISION K,L,MOMENT,KAPPA
C      CHARACTER*13 FILNAM
C      CHARACTER*1 OUTOPT
C      DIMENSION A(8,8),BCOLM(8)
C      DIMENSION XPOS(51),SIGMA(51),TAU(51),WF(51),WS(51)
C      COMMON/NUMRC1/IANTYP,D,L,TF,TS,DF,DS
C      COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAU1LS,P,MOMENT,DELTAF
C      COMMON/NUMRC3/XPOS,SIGMA,TAU,WF,WS
C      * DEFINE FUNCTIONS *
C      RATD=DF/DS
C      BTA=(K*(DS+DF)/(4.*DS*DF))**.25
C      BTAD=BTA*D
C      CBD=COS(BTAD)
C      SBD=SIN(BTAD)
C      EBD=EXP(BTAD)
C      ENBD=EXP(-BTAD)
C      Z1=RATD*BTA
C      Z2=2.*(BTA**2)
C      Z3=RATD*(BTA**3)/3.
C      PI=ACOS(-1.)
C      EAF=EF*(TF*1.0)
C      EAS=ES*(TS*1.0)
C      KAPPA=(G/B)*((1.0/EAF)+(1.0/EAS))
C      KAPPA=SQRT(KAPPA)
C      T=((PI*DELTAF/(2.0*D))**2)*EAS
C      TAUCON=T*KAPPA*EAF/(EAS+EAF)
C      FOR POSTBUCKLING CASES, THE SKIN DEFLECTION AND ROTATION
C      AT THE FLANGE TERMINATION IS FOUND FROM RUNNING THE
C      DIRECT LOADS ANALYSIS, AND ASSUMING THE SKIN DEFLECTION
C      AND ROTATION AT THE TIP LOCATION TO BE ADEQUATE FOR THE
C      POSTBUCKLING CASE. - THE FIRST LOOP IS USED TO DETERMINE
C      THESE VALUES, THE SECOND IS USED TO PERFORM THE FINAL
C      SOLUTION.
C      DO 1020 I=1,8
C      DO 1010 J=1,8
C      A(I,J)=0.
C 1010 CONTINUE
C 1020 CONTINUE
C      DO 1050 LOOP=1,IANTYP
C      * CONSTRUCT 'A' MATRIX *
C      IF (LOOP.EQ.1) THEN
C      B.C. #1: PF(0) = P

```



```

C      A(1,1)=-2.*DF*(BTA**3)
      A(1,2)=2.*DF*(BTA**3)
      A(1,3)=2.*DF*(BTA**3)
      A(1,4)=2.*DF*(BTA**3)
      A(1,8)=6.*DF
C
      ELSE IF (LOOP.EQ.2) THEN
C
C      B.C. #1: WF(0)=DELTA F
C
      A(1,1)=1.0
      A(1,3)=1.0
      A(1,5)=1.0
C
      END IF
C
C      B.C. #2: MF(0) = 0
C
      A(2,2)=BTA**2
      A(2,4)=-(BTA**2)
      A(2,7)=1.
C
C      B.C. #3: PS(0) = 0
C
      A(3,1)=Z3
      A(3,2)=-Z3
      A(3,3)=-Z3
      A(3,4)=-Z3
      A(3,8)=1.
C
C      B.C. #4: WS'(0) = 0
C
      A(4,1)=-Z1
      A(4,2)=-Z1
      A(4,3)=Z1
      A(4,4)=-Z1
      A(4,6)=1.
C
C      B.C. #5: PF(D) = 0
C
      A(5,1)=-Z2*BTA*EBD*(CBD+SBD)
      A(5,2)=Z2*BTA*EBD*(CBD-SBD)
      A(5,3)=Z2*BTA*ENBD*(CBD-SBD)
      A(5,4)=Z2*BTA*ENBD*(CBD+SBD)
      A(5,8)=6.
C
C      B.C. #6: MF(D) = 0
C
      A(6,1)=-Z2*EBD*SBD
      A(6,2)=Z2*EBD*CBD
      A(6,3)=Z2*ENBD*SBD
      A(6,4)=-Z2*ENBD*CBD
      A(6,7)=2.
      A(6,8)=6.*D
C
      IF (LOOP.EQ.1) THEN
C
C      B.C. #7: WS(D) = DELTA(D)
C
      A(7,1)=-RATD*EBD*CBD
      A(7,2)=-RATD*(EBD*SBD-(BTA**2)*(L-D)**2)
      A(7,3)=-RATD*ENBD*CBD
      A(7,4)=-RATD*(ENBD*SBD+(BTA**2)*(L-D)**2)
      A(7,5)=1.
      A(7,6)=D

```

```

      A(7,7)=(D**2)-((L-D)**2)
      A(7,8)=D**3
C
C   B.C. #8: WS'(D) = W'(D)
C
      A(8,1)=-RATD*BTA*EBD*(CBD-SBD)
      A(8,2)=-RATD*BTA*EBD*(CBD+SBD)-Z2*RATD*(L-D)
      A(8,3)=RATD*BTA*ENBD*(CBD+SBD)
      A(8,4)=-RATD*BTA*ENBD*(CBD-SBD)+Z2*RATD*(L-D)
      A(8,6)=1.
      A(8,7)=2.*D+2.*(L-D)
      A(8,8)=3.*(D**2)
C
      ELSE IF (LOOP.EQ.2) THEN
C
C   B.C. #7: WS(D) = DELTA(D)
C
      A(7,1)=-RATD*EBD*CBD
      A(7,2)=-RATD*EBD*SBD
      A(7,3)=-RATD*ENBD*CBD
      A(7,4)=-RATD*ENBD*SBD
      A(7,5)=1.
      A(7,6)=D
      A(7,7)=D**2
      A(7,8)=D**3
C
C   B.C. #8: WS'(D) = W'(D)
C
      A(8,1)=-RATD*BTA*EBD*(CBD-SBD)
      A(8,2)=-RATD*BTA*EBD*(CBD+SBD)
      A(8,3)=RATD*BTA*ENBD*(CBD+SBD)
      A(8,4)=-RATD*BTA*ENBD*(CBD-SBD)
      A(8,6)=1.
      A(8,7)=2.*D
      A(8,8)=3.*(D**2)
C
      END IF
C
C   * CONSTRUCT 'BCOLM' MATRIX *
C
      IF (LOOP.EQ.1) THEN
        IF (IANTYP.EQ.2) P=1.0
        IF (IANTYP.EQ.2) MOMENT=0.0
C
        BCOLM(1)=P
        BCOLM(2)=MOMENT/(DF*2.0)
        BCOLM(3)=0.
        BCOLM(4)=0.
        BCOLM(5)=0.
        BCOLM(6)=0.
        BCOLM(7)=P*((2.*(L**3))-(3.*D*(L**2))+(D**3))/(6.*DS)
        BCOLM(8)=P*(-(L**2)+(D**2))/(2.*DS)
        BCOLM(7)=BCOLM(7)+(MOMENT*((L-D)**2)/(2.*DS))
        BCOLM(8)=BCOLM(8)+(MOMENT*(-(L-D)/DS))
C
      ELSE IF (LOOP.EQ.2) THEN
C
        BCOLM(1)=DELTAF
        BCOLM(2)=0.
        BCOLM(3)=0.
        BCOLM(4)=0.
        BCOLM(5)=0.
        BCOLM(6)=0.
        BCOLM(7)=DELTAD
        BCOLM(8)=THETAD

```

```

      END IF
C
C
C
C      * SOLVE 8 SIMULTANEOUS EQUATIONS *
C      CALL SIMQ(A,BCOLM,8,KS)
C
      IF(KS.EQ.1) THEN
        WRITE(6,1030)
1030      FORMAT('1',' - FATAL RUN TIME ERROR -',/,/,
      &          ' ALGORITHMIC SINGULARITY IN SUBROUTINE SIMQ')
        STOP
      END IF
C
C
C      * DEFINE COEFFICIENTS *
C
      A1=BCOLM(1)
      B1=BCOLM(2)
      C1=BCOLM(3)
      D1=BCOLM(4)
      A2=BCOLM(5)
      B2=BCOLM(6)
      C2=BCOLM(7)
      D2=BCOLM(8)
C
C
C      * CALCULATE WF(X), WS(X), PX(X) *
C
      X=0.0
C
      IF ((LOOP.EQ.1).AND.(IANTYP.EQ.2)) THEN
        INSPCT=2
        DELX=D
      ELSE
        INSPCT=51
        DELX=D/50.
      END IF
C
C
      DO 1040 I=1, INSPCT
        BTAX=BTAX*X
        CBX=COS(BTAX)
        SBX=SIN(BTAX)
        EBX=EXP(BTAX)
        ENBX=EXP(-BTAX)
        W1A=EBX*(A1*CBX+B1*SBX)
        W1B=ENBX*(C1*CBX+D1*SBX)
        W1C=A2+B2*X+C2*(X**2)+D2*(X**3)
        WS(I)=-RATD*(W1A+W1B)+W1C
        WF(I)=W1A+W1B+W1C
C
        IF ((IANTYP.EQ.1).AND.(I.EQ.1)) THEN
          DELTAF=WF(I)
          T=((PI*DELTAF/(2.0*D))**2)*EAS
          TAUCON=T*KAPPA*EAF/(EAS+EAF)
        END IF
C
        XPOS(I)=X
        SIGMA(I)=K*(WF(I)-WS(I))
        TAU(I)=TAUCON*EXP(-X*KAPPA)
C
        X=X+DELX
1040      CONTINUE
C
      IF ((LOOP.EQ.1).AND.(IANTYP.EQ.2)) THEN
        DELTAD=WS(2)*(DELTAF/WF(1))

```

```

W1A=EBX*((A1*(CBX-SBX))+(B1*(CBX+SBX)))
W1B=ENBX*((-C1*(CBX+SBX))+(-D1*(CBX-SBX)))
W1C=B2+(2.0*C2*X)+(3.0*D2*(X**2))
THETAD=(-RATD*(W1A+W1B))+W1C
THETAD=THETAD*(DELTA/WF(1))
END IF
C
1050 CONTINUE
C
RETURN
END
C
-----
C
SUBROUTINE SIMQ
C
PURPOSE
C   OBTAIN SOLUTION OF A SET OF SIMULTANEOUS
C   LINEAR EQUATIONS, AX=B
C
USAGE
C   CALL SIMQ(A,B,N,KS)
C
DESCRIPTION OF PARAMETERS
C   A - MATRIX OF COEFFICIENTS STORED COLUMNWISE. THESE ARE
C   DESTROYED IN THE COMPUTATION. THE SIZE OF MATRIX
C   A IS N BY N.
C   B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE
C   REPLACED BY FINAL SOLUTION VALUES, VECTOR X.
C   N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE > ONE.
C   KS - OUTPUT DIGIT
C       0 FOR A NORMAL SOLUTION
C       1 FOR A SINGULAR SET OF EQUATIONS
C
REMARKS
C   MATRIX A MUST BE GENERAL.
C   IF MATRIX IS SINGULAR, SOLUTION VALUES ARE MEANINGLESS.
C   AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX
C   INVERSION (MINV) AND MATRIX PRODUCT (GMPD).
C
SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C   NONE
C
METHOD
C   METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST
C   PIVOTAL DIVISOR. EACH STAGE OF ELIMINATION CONSISTS
C   OF INTERCHANGING ROWS WHEN NECESSARY TO AVOID
C   DIVISION BY ZERO OR SMALL ELEMENTS.
C   THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE
C   IN N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES
C   IS CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL
C   SOLUTION VALUES ARE DEVELOPED IN VECTOR B, WITH
C   VARIABLE 1 IN B(1), VARIABLE 2 IN B(2),.....,
C   VARIABLE N IN B(N). IF NO PIVOT CAN BE FOUND EXCEEDING
C   A TOLERANCE OF 0.0, THE MATRIX IS CONSIDERED SINGULAR
C   AND KS IS SET TO 1. THIS TOLERANCE CAN BE MODIFIED BY
C   REPLACING THE FIRST STATEMENT.
C
SUBROUTINE SIMQ(A,B,N,KS)
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION A(1),B(1)
TOL=0.0
KS=0
JJ=-N

```

```

      DO 65 J=1,N
      JY=J+1
      JJ=JJ+N+1
      BIGA=0.0
      IT=JJ-J
      DO 30 I=J,N
      IJ=IT+I
      IF (ABS (BIGA) -ABS (A (IJ))) 20,30,30
20  BIGA=A (IJ)
      IMAX=I
30  CONTINUE
      IF (ABS (BIGA) -TOL) 35,35,40
35  KS=1
      RETURN
40  I1=J+N* (J-2)
      IT=IMAX-J
      DO 50 K=J,N
      I1=I1+N
      I2=I1+IT
      SAVE=A (I1)
      A (I1)=A (I2)
      A (I2)=SAVE
50  A (I1)=A (I1)/BIGA
      SAVE=B (IMAX)
      B (IMAX)=B (J)
      B (J)=SAVE/BIGA
      IF (J-N) 55,70,55
55  IQS=N* (J-1)
      DO 65 IX=JY,N
      IXJ=IQS+IX
      IT=J-IX
      DO 60 JX=JY,N
      IXJX=N* (JX-1)+IX
      JJX=IXJX+IT
60  A (IXJX)=A (IXJX) - (A (IXJ) *A (JJX))
65  B (IX)=B (IX) - (B (J) *A (IXJ))
70  NY=N-1
      IT=N*N
      DO 80 J=1,NY
      IA=IT-J
      IB=N-J
      IC=N
      DO 80 K=1,J
      B (IB)=B (IB) -A (IA) *B (IC)
      IA=IA-N
80  IC=IC-1
      RETURN
      END

```

C

C

C

SUBROUTINE FAYLPD (FIMAX)

C

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DOUBLE PRECISION K,L,MOMENT

C

DIMENSION XPOS (51),SIGMA (51),TAU (51),WF (51),WS (51)

C

COMMON/NUMRC1/IANTYP,D,L,TF,TS,DF,DS

COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAUILS,P,MOMENT,DELTAF

COMMON/NUMRC3/XPOS,SIGMA,TAU,WF,WS

C

FIMAX=0.0

C

DO 1000 I=1,51

FINDEX=FI (SIGMA (I),TAU (I))

```

CCCC      WRITE(6,*)FINDEX
          : FIMAX=MAX(FIMAX,FINDEX)
          1000 CONTINUE
C
C...PUT THE PREDICTED FAILURE LOAD/DEFLECTION IN THE
C      APPLIED LOAD/DEFLECTION VARIABLE LOCATION
C
      IF (IANTYP.EQ.1) THEN
        IF (P.NE.0.) THEN
          IF (ABS(FIMAX-1.) .LT. 0.001) THEN
            FIMAX=1
            RETURN
          ELSE
            P=P/FIMAX
            MOMENT=MOMENT/FIMAX
          ENDIF
        ELSE
          IF (ABS(FIMAX-1.) .LT. 0.0001) THEN
            FIMAX=1
            RETURN
          ELSE
            MOMENT=MOMENT/FIMAX
          ENDIF
        ENDIF
      ELSE IF (IANTYP.EQ.2) THEN
        IF (ABS(FIMAX-1.) .LT. 0.000001) THEN
          FIMAX=1
          RETURN
        ELSE
          DELTAF=DELTAF/FIMAX
        ENDIF
      END IF
C
      RETURN
      END
C
C-----
C
      DOUBLE PRECISION FUNCTION FI(SIGMAZ,TAUXZ)
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,MOMENT
C
      COMMON/NUMRC2/EF,ES,K,G,B,PULT,TAU1LS,P,MOMENT,DELTAF
C
      IF (SIGMAZ.GT.0) THEN
        FULT=(SIGMAZ/PULT)**2
        FI=SQRT(FULT)
      ELSE
        FI=0.0
      ENDIF
C
      RETURN
      END
C
C-----
C
      SUBROUTINE OUTPUT(INUT,J)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,L,MOMENT
C
      DIMENSION XPOS(51),SIGMA(51),TAU(51),WF(51),WS(51)
C

```

```

COMMON/NUMRC1/ IANTYP, D, L, TF, TS, DF, DS
COMMON/NUMRC2/ EF, ES, K, G, B, PULT, TAU1LS, P, MOMENT, DELTAF
COMMON/NUMRC3/ XPOS, SIGMA, TAU, WF, WS

C
  IF (J.EQ.0) THEN
    WRITE (INUT, 998)
  998 FORMAT (///, ' DEFLECTIONS AND STRESSES FOR INITIAL CONDITION')
    WRITE (INUT, 1000)
  ELSE
    WRITE (INUT, 999)
  999 FORMAT (///, ' DEFLECTIONS AND STRESSES FOR CRITICAL CONDITION')
    WRITE (INUT, 1000)
  ENDIF
  1000 FORMAT (/, 2X, ' INTERFACE', 8X, ' FLANGE', 10X, ' SKIN',
    &          10X, ' NORMAL', 9X, /,
    &          2X, ' POSITION', 7X, ' DEFLECTION', 5X, ' DEFLECTION',
    &          7X, ' STRESS' 9X, /,
    &          1X, 58 (' -'))

C
  DO 1010 I=1, 51
    IF ((INUT.EQ.6) .AND. ((I.EQ.14) .OR. (I.EQ.27)
    &    .OR. (I.EQ.40))) THEN
      CALL PAWS
      IF (J.EQ.0) THEN
        WRITE (INUT, 998)
        WRITE (INUT, 1000)
      ELSE
        WRITE (INUT, 999)
        WRITE (INUT, 1000)
      ENDIF
    END IF
    WRITE (INUT, 1020) XPOS(I), WF(I), WS(I), SIGMA(I)
  1010 CONTINUE
  1020 FORMAT(' ', 1PE11.4, 3(4X, 1PE11.4))

C
  IF (J.EQ.0) THEN
    IF (IANTYP.EQ.1) THEN
      WRITE (INUT, 1031) P, MOMENT
    1031 FORMAT (/, ' THE ENTERED PULL OFF LOAD IS: ',
    &            1PE11.4, /,
    &            ' THE ENTERED MOMENT IS: ',
    &            1PE11.4)
    ELSE IF (IANTYP.EQ.2) THEN
      WRITE (INUT, 1041) DELTAF
    1041 FORMAT (/, ' THE ENTERED DEFLECTION IS: ', 1PE11.4)
    END IF
    IF (INUT.EQ.6) THEN
      CALL PAWS
    ENDIF
  ELSE
    IF (IANTYP.EQ.1) THEN
      WRITE (INUT, 1030) P, MOMENT
    1030 FORMAT (/, ' THE PREDICTED PULL OFF LOAD AT FAILURE IS: ',
    &            1PE11.4, /,
    &            ' THE PREDICTED MOMENT AT FAILURE IS: ',
    &            1PE11.4)
    ELSE IF (IANTYP.EQ.2) THEN
      WRITE (INUT, 1040) DELTAF
    1040 FORMAT (/, ' THE PREDICTED DEFLECTION AT FAILURE IS: ', 1PE11.4)
    END IF
  ENDIF

C
  RETURN
END
SUBROUTINE TIMERS (ATIME, ADATE)

C

```

```

C      INTEGER*2 IHR, IMIN, ISEC, I100TH
      INTEGER*2 IYR, IMON, IDAY
C
C      CHARACTER*1 ATIME*8, ADATE*9
      CHARACTER*3 MONTH(12)
C
C      DATA MONTH/'JAN','FEB','MAR','APR','MAY','JUN',
X      'JUL','AUG','SEP','OCT','NOV','DEC'/
C
C      *** MICROSOFT FORTRAN (IBM PC) TIME AND DATE ROUTINES ***
C
C      CALL GETTIM(IHR, IMIN, ISEC, I100TH)
      CALL GETDAT(IYR, IMON, IDAY)
C
C      IYR=IYR-100*(IYR/100)
C
C      WRITE(ATIME,101) IHR, IMIN, ISEC
      WRITE(ADATE,102) IDAY, MONTH(IMON), IYR
C
C      101 FORMAT(I2,':',I2,':',I2)
      102 FORMAT(I2,'-',A3,'-',I2)
C
C      RETURN
      END

```

3. Elasticity Method (WEBSTER)

```

C *****
C      **                      WEB                      **
C
C      Web-Stiffener Termination -- Elasticity Routine
C
C      Developed by: Northrop - Aircraft Division
C                  Strength & Life Assurance Dept.
C                  Hawthorne, CA 90250
C                  Mary Mahler
C                  Dr. Han-Pin Kan
C
C      This program is developed under the Navy/FAA contract
C      "Out-of-Plane Analysis of Composite Structures"
C      Contract No. N62269-87-C-0226. MCAIR-St. Louis being
C      the primary contractor, Northrop is the subcontractor.
C
C      For further information or problems contract:
C      M. Mahler (310) 332-9648 or Dr. Han-Pin Kan (310) 332-5285
C
C      This version is the final release of WEBSTER for the NAVY contract.
C      Finalized: December 1990.
C      Input/Output revision: January 1993.
C      One of 4 programs, linked with: (WEB-1),WEB-2,WEB-3,WEB-4,(WEB.LIB)
C
C      NOTE ON RUNNING WEBSTER: WEBSTER should be ran on the C: drive of
C      a PC with the WEBSTER directory made with sub-directories:
C      \MAT-DATA, \STR-DATA, \OUT-DATA in C:\WEBSTER
C
C      This is capable of running on the PC. (or VAX with the change to
C      the DIRM and DIRM1 subroutines)
C *****
C      DOUBLE PRECISION A(2,6,6), RHS(180), STX(2), STXY(2), STY(2),
C      WA(4), WKK(25), XC(300), BETA, BB, RCON,
C      CON, CON1, CON2, DZERO
C      COMPLEX*16 AL, AL2T, B(25,25), BL, B4(4,4), CCON, BV(40),
C      CMAT(180,25), CMAT5(180,25), CONST(25), CSMAT(25,25),

```



```

C          CZERO, MU, T2, ONE, T1(4,4), WAA(675),
C          CMATS(25,25), XDBL(60), FN1, FX(60)
DIMENSION RHS7(7,1), WK7(70), ZEX(2), ZEY(2), ZEZ(2),
C          ZGXY(2), RX(300)
INTEGER BOTLIM, IDUM(5), IJ(6), INFER(60), IRX(300),
C          ISVV(50), TOPLIM, TOTLEN
COMPLEX LAMDA(25), LAMDA2(25), PL, FFL, X(60)
REAL L, LMAT7(7,7), M1, M3, NAX, MD2T, M2,
C          NY3, NUXY(2), NUXZ(2), NUYZ(2), N4, styhld(46),
C          M2B, M2T, MD2B, M2A, N2T, N2B, M4, AD7(60),
C          stxhld(46), x2hld(46)
CHARACTER NMOUT*10, SFILE*35
COMMON/CP1/MU(2,4)
COMMON/LMAT/ARL(8,8), BIM(8,8)
COMMON/RL1/ANGLE
COMMON/RL2/BETA(2,6,6)
COMMON/STRS/AL(4,25), BL(4,25)
COMMON/TRANS/T2(4,4)
EQUIVALENCE (RX(1),IRX(1))
WRITE(*,900)
CALL STRMKR(H1, H2, ZEX, ZEY, ZEZ, ZGXY, NUXY, NUXZ, NUYZ,
            ULTN, ULTS, AON, AOS)
C *****
C INITIALIZE VARIABLES AND ZERO OUT MATRICES
C *****
CZERO = CMPLX(0.0D0,0.0D0)
EPS = 1.0E-08
NCP = 29
IER = 0
IJ(1) = 1
IJ(2) = 2
IJ(3) = 4
IJ(4) = 5
IJ(5) = 6
IJOB = 0
ITMAX = 200
KN = 0
NDRC = 25
NGUESS = 0
NSIG = 7
N1 = 1
N7 = 7
ONE = CMPLX(1.0D0,0.0D0)
PI = 4.0 * ATAN(1.0)
ZERO = 0.0E0
DZERO = 0.0D0
DO 5 I = 1, 4
DO 5 J = 1, 4
5 B4(I,J) = CZERO
DO 10 I = 1, NDRC
DO 10 J = 1, NDRC
10 B(I,J) = CZERO
DO 15 I = 1, NDRC
15 B(I,I) = ONE
DO 20 I = 1, 180
20 RHS(I) = ZERO
DO 30 I = 1, 2
DO 25 JK = 1, 6
DO 25 KJ = 1, 6
25 A(I,JK,KJ) = ZERO
A(I,1,1) = 1.0 / ZEX(I)
A(I,1,2) = -NUXY(I) / ZEX(I)
A(I,1,3) = -NUXZ(I) / ZEX(I)
A(I,2,1) = A(I,1,2)
A(I,2,2) = 1.0 / ZEY(I)
A(I,2,3) = -NUYZ(I) / ZEY(I)

```

```

      A(I,3,1) = A(I,1,3)
      A(I,3,2) = A(I,2,3)
      A(I,3,3) = 1.0 / ZEX(I)
      A(I,6,6) = 1.0 / ZGXY(I)
30  C *****
C      INPUT STATEMENTS
C *****
      WRITE(*,1050)
      READ(*,'(A)') nmout
      KHD = 10
      DO 2 K = 1, 9
      KK = 10 - K
2      IF (NMOUT(KK:KK).EQ.' ') KHD = KK
      SFILE(1:20) = 'C:\WEBSTER\OUT-DATA\'
      SFILE(21:20+KHD)=NMOUT(1:KHD-1)
      SFILE(21+KHD:24+KHD) = '.OUT'
      OPEN(31,FILE=SFILE)
      WRITE(31,11) SFILE
      N = 25
      WRITE(*,910)
      READ(*,*) L, D, ANGLE1
      ANGLE = ANGLE1
      IF (ANGLE.GT.88. AND .ANGLE.LT.92) ANGLE = 88.00
C
      IF (L.LT.0.10) WRITE(*,943) L
      IF (D.LT.0.10) WRITE(*,943) D
      IF (L/D.GT.7.OR.D/L.GT.7) WRITE(*,945) L,D
C
      ANGLE = ANGLE * PI / 180.00
35  WRITE(*,912)
      READ(*,*) NSIDE
      IF (NSIDE.EQ.1) THEN
      WRITE(*,913)
      READ(*,*) N4, VXY4, M4
      N4 = -N4
      N2T = N4 * H1 * ZEX(1) / ( H1 * (H1*ZEX(1) + H2*ZEX(2) ))
      N2B = N4 * H2 * ZEX(2) / ( H2 * (H1*ZEX(1) + H2*ZEX(2) ))
      N4 = N4 / H2
      M2A = M4 + (N2T*H1*H1/2.0) + (N4*H2*H2/2.0) -
      (N2B*H2*H2/2.0) + VXY4*(L+D)
      VM2T = 2.0 * VXY4 / (H1 + H2)
      MD2T = (H1 - H2) / 2.0
      IF (MD2T.LE.ZERO) VM2T = H1 * VM2T / (H1-MD2T)
      IF (MD2T.LE.ZERO) MD2T = ZERO
      VM2B = 2.0 * VXY4 / (H1 + H2)
      MD2B = (H1 - H2) / 2.0
      ELSE
      IF ( NSIDE.EQ.2) THEN
      WRITE(*,915)
      READ(*,*) PXY, TLL, HEIGHT
      DUM = 4.0 * (L+D) - HEIGHT
      IF (DUM.LE.TLL) THEN
      M4 = PXY * TLL / 4.0
      M2B = -PXY * (L+D) * (L+D) / TLL
      M2T = PXY * (TLL-2*L-2*D)*(TLL-2*L-2*D) / (4.0*TLL)
      ELSE
      M2B = -PXY * (L+D) * (L+D) / ( TLL * 4.0 )
      DUM = HEIGHT + 2*TLL - 4.0*L - 4.0*D
      IF (ABS(DUM).LT.0.001.OR.HEIGHT.LT.0.001) THEN
      M2T = 0.10 * PXY * D
      M4 = M2B + PXY*(L+D) + M2T
      ELSE
      M2T = PXY * (TLL-2*L-2*D)*(TLL-2*L-2*D) / (4.0*DUM)
      M4 = PXY * (L+D) + M2T + M2B
      END IF
      END IF

```

```

ELSE
  IF (NSIDE.EQ.3) THEN
    WRITE(*,916)
    READ(*,*) N4, VXY4, M4
    N4 = -N4
    N2T = N4 * H1 * ZEX(1) / ( H1 * (H1*ZEX(1) + H2*ZEX(2) ))
    N2B = N4 * H2 * ZEX(2) / ( H2 * (H1*ZEX(1) + H2*ZEX(2) ))
    N4 = N4 / H2
    M2A = M4 + (N2T*H1*H1/2.0) + (N4*H2*H2/2.0) -
              (N2B*H2*H2/2.0) + VXY4*(L+D)
  ELSE
    IF (NSIDE.EQ.4) THEN
      WRITE(*,1114)
    ELSE
      WRITE(*,1117)
      GO TO 35
    END IF
  END IF
  END IF
  END IF
  IF (NSIDE.NE.3) GO TO 45
C *****
C THE NEXT SEGMENT IS FOR PRESSURE LOADING ON THE BOTTOM SURFACE
C *****
40  WRITE(*,920)
    READ(*,*) NTYPE
    IF (NTYPE.EQ.1) THEN
      WRITE(*,925)
      READ(*,*) X1,P1, X2, P2
      A3 = ZERO
      B3 = ( P2 - P1 ) / ( X2 - X1 )
      C3 = P1 - B3 * X1
    ELSE
      IF (NTYPE.EQ.2) THEN
        WRITE(*,930)
        READ(*,*) XS, PS
        WRITE(*,935)
        READ(*,*) X1, P1
        Q3 = (X1-XS)*(X1-XS) / ( 4.0 * (P1-PS) )
        A3 = 1.0 / ( 4.0 * Q3 )
        B3 = -XS / ( 2.0 * Q3 )
        C3 = PS + (XS*XS) / (4.0 * Q3)
      ELSE
        IF (NTYPE.EQ.3) THEN
          WRITE(*,940)
          READ(*,*) X0,P0, X1,P1, X2, P2
          D0 = X0*X0 - (X1+X2)*X0 + X1*X2
          D1 = X1*X1 - (X0+X2)*X1 + X0*X2
          D2 = X2*X2 - (X0+X1)*X2 + X1*X0
          A3 = P0/D0 + P1/D1 + P2/D2
          B3 = - (X2+X1)*P0/D0 - (X0+X2)*P1/D1 - (X0+X1)*P2/D2
          C3 = P0*X1*X2/D0 + P1*X0*X2/D1 + P2*X0*X1/D2
        ELSE
          WRITE(*,955)
          GO TO 40
        END IF
      END IF
    END IF
  END IF
C *****
C DETERMINE THE LOCATION OF THE COLLOCATION POINTS
C *****
45  IF ( NCP.EQ.21 ) THEN
    NCP =21
    NJ = 3
    NJ1 = 5
    NJ2R = 5

```

```

      NJ2L = 5
    ELSE
      IF ( NCP.EQ.29 ) THEN
        NCP =29
        NJ = 4
        NJ1 = 7
        NJ2R = 7
        NJ2L = 7
      ELSE
        NJ = NCP/3.0
        NJ1 = NJ / 2.0
        NJ2R = NJ * 0.3
        NJ2L = NJ * 0.2
      END IF
    END IF
  CC
    DUM = H1 / ABS ( ATAN (ANGLE) )
    NR1S = NJ
    NR2NT = 2 * NJ
    NR2ST = 2 * NJ + NJ1
    NR2NB = 2 * ( NJ+NJ1 )
    NR2SB = 2 * ( NJ+NJ1 ) + NJ2R
    NR3N = 2 * ( NJ+NJ1+NJ2R )
    NR3S = 2 * ( NJ+NJ1+NJ2R ) + NJ
    NR4N = 2 * ( NJ+NJ1+NJ2R+NJ )
    NR4S = 2 * ( NJ+NJ1+NJ2R+NJ ) + NJ2L
    NRTL = NR4S + NJ2L
    DA1 = ( D - DUM ) / NJ
    DA2T = H1 / NJ1
    DA2B = H2 / NJ2R
    DA3 = (L+D) / NJ
    DA4 = H2 / NJ2L
    DO 50 I = 1, NJ
      XC(I) = DUM + (I-0.5) * DA1
    50  XC(180+I) = -L + (I-0.5) * DA3
      DO 55 I = 1, NJ1
        XC(100+I) = H1 - (I-0.5) * DA2T
    55  DO 60 I = 1, NJ2R
        XC(100+NJ1+I) = -(I-0.5) * DA2B
    60  DO 65 I = 1, NJ2L
        XC(280+I) = -H2 + (I-0.5) * DA4
    65  *****
  CC *****
  CC SET UP THE RIGHT HAND SIDE OF THE EQUATION CMAT*C=RHS
  CC THIS IS DONE BY RUNNING THE LAST 100 OR SO LINES IN THIS
  CC PORTION OF THE PROGRAM. THIS WAS FOUND TO BE EASIER
  CC THEN CALLING A SUBROUTINE AND TRANSFERING ALL NEEDED
  CC PARAMETERS.
  CC *****
  GO TO 2001
2000 CONTINUE
C *****
C EVALUATE CONSTANTS FOR ANALYSIS
C *****
76 DO 80 K = 1, 2
   DO 80 I = 1, 5
     II = IJ(I)
     DO 80 J = 1, 5
       JJ = IJ(J)
       BETA(K,II,JJ) = A(K,II,JJ) -
         ( A(K,II,3) * A(K,JJ,3) / A(K,3,3) )
80 CONTINUE
   DO 85 K = 1, 2
     BB = ( 2.0 * BETA(K,1,2) ) + BETA(K,6,6)
     RCON = ( BB * BB ) - ( 4.0 * BETA(K,2,2) * BETA(K,1,1) )
     IF (RCON.GT.0) THEN
       CON = SQRT(RCON)

```

```

      CON1 = (-BB+CON) / (BETA(K,1,1)*2.0)
      CON2 = (-BB-CON) / (BETA(K,1,1)*2.0)
      IF (CON1.LT.0) THEN
        CON1 = SQRT(-CON1)
        MU(K,1) = DCMPLX(DZERO,CON1)
      ELSE
        CON1 = DSQRT(CON1)
        MU(K,1) = DCMPLX(CON1,DZERO)
      END IF
      IF (CON2.LT.0) THEN
        CON2 = DSQRT(-CON2)
        MU(K,2) = DCMPLX(DZERO,CON2)
      ELSE
        CON2 = DSQRT(CON2)
        MU(K,2) = DCMPLX(CON2,DZERO)
      END IF
      MU(K,3) = - MU(K,1)
      MU(K,4) = - MU(K,2)
    ELSE
      CON = DSQRT(-RCON)
      CCON = DCMPLX(DZERO,CON)
      MU(K,1) = CDSQRT((-BB+CCON) / (BETA(K,1,1)*2.0) )
      MU(K,2) = CDSQRT((-BB-CCON) / (BETA(K,1,1)*2.0) )
      MU(K,3) = - MU(K,1)
      MU(K,4) = - MU(K,2)
    END IF
85    Continue
CC *****
CC          CALL IMSL ROUTINE TO SOLVE FOR LAMDA
CC *****
      NN4 = 4
      IJOB = 0
      DO 90 I = 1, NN4
        B4(I,I) = ONE
        T1(1,I) = ONE
        T1(2,I) = MU(1,I)
        T1(3,I) = PL(0.0,1,I)
90      T1(4,I) = FFL(0.0,1,I)
        CALL ZLEQTC(T1,NN4,NN4,B4,NN4,NN4,IJOB,WA,IER)
      DO 100 I = 1, 4
        T1(1,I) = ONE
        T1(2,I) = MU(2,I)
        T1(3,I) = PL(0.0,2,I)
100     T1(4,I) = FFL(0.0,2,I)
      DO 105 I = 1, NN4
        DO 105 J = 1, NN4
          DO 105 K = 1, NN4
105       T2(I,J) = T2(I,J) + B4(I,K) * T1(K,J)
        CALL ZZANLY(EPS, NSIG, KN, NGUESS, N, XDBL, ITMAX, INFER, IER)
CC *****
CC          THE LAMDA'S AND DELTA ARE SOLVED FOR.
CC          NOW IT IS DETERMINED WHICH LAMDA IS SATISFACTORY TO USE.
CC          THE REST OF THE PROGRAM IS DEPENDENT ON THE LAMDAS USED.
CC *****
      NRT = 0
      NRTZ = 0
      DO 110 I = 1, 50
        FX(KN+I) = FN1(XDBL(KN+I))
        ZRL = SNGL( DREAL(XDBL(KN+I)) )
        ZIM = SNGL( DIMAG(XDBL(KN+I)) )
        X(KN+I) = CMPLX(ZRL,ZIM)
110     CONTINUE
CC *****
CC *****
CC          NEW PART TO DETERMINE LAMDAS AUTOMATICALLY

```

```

NL2 = 0
ERR1 = 0.0010
  ZXXJ = 11.5
  ZXCV = 13.5
DO 125 I = 1, N+NGUESS
  IF (REAL(X(KN+I)).LT.-1) GO TO 125
  IF (ABS(REAL(X(KN+I))).GT.ZXXJ) GO TO 125
  IF (ABS(AIMAG(X(KN+I))).GT.ZXCV) GO TO 125
  IF (CDABS(FX(KN+I)).GT.ERR1) GO TO 125
  NL2 = NL2 + 1
  RX( (NL2*5)-4 ) = REAL (X(KN+I))
  RX( (NL2*5)-3 ) = AIMAG(X(KN+I))
  RX( (NL2*5)-2 ) = REAL (FX(KN+I))
  RX( (NL2*5)-1 ) = DIMAG(FX(KN+I))
  RX( NL2*5 ) = REAL(INFER(KN+I))
125 CONTINUE
  BOTLIM = 1
  TOPLIM = NL2*5
  TOTLEN = 5
  KEYPOS = 1
  CALL UTRSRT(IRX, IDUM, BOTLIM, TOPLIM, KEYPOS, TOTLEN)
  IK = 0
CC *****
CC NOW SET UP AN (NLAM x NLAM) MATRIX OF COEFFICIENTS TO EQUATIONS
CC USED TO SOLVE FOR THE UNKNOWN CONSTANTS. (DEPENDENT ON LAMDA)
CC *****
      DO 334 I = 1, NL2
      LAMDA(I) = CMPLX( RX((I*5)-4), RX((I*5)-3) )
      FX(I) = CMPLX( RX((I*5)-2), RX((I*5)-1) )
      INFER(I) = RX((I*5))
334 CONTINUE
      NSVV = 0
135 IF (NRT.NE.0) THEN
      NRT2 = 10
      END IF
      NLAM = NL2 - NSVV
      DO 140 JJ = 1, NRTL
      DO 140 II = 1, NLAM
140 CMAT(JJ, II) = CZERO
      NRT = 0
      ISKP = 0
      DO 225 JCT = 1, NLAM + NSVV
      J = JCT
      NZZY = 0
      IF (ISKP.GT.0) J = JCT - ISKP
      DO 145 IJL = 1, NSVV
      IF ( JCT.EQ.ISVV(IJL) ) NZZY = 550
145 IF ( JCT.EQ.ISVV(IJL) ) ISKP = ISKP + 1
      IF (NZZY.GT.1) GO TO 225
      CALL FNP(LAMDA(JCT))
CC
CC FOR EACH EIGENVALUE USED, THE EIGENVECTOR IS NOW COMPUTED
CC BL IS C(2,K) AND AL(J,K) = C(1,K)
CC FNP FOUND THE REAL AND IMAG PART OF THE F-MAT FOR THE EIGENVALUE
CC
      DO 155 IU = 1, 4
      DO 150 JU = 1, 3
      LMAT7(JU+4, IU+3) = ARL(JU+1, IU)
150 LMAT7(IU, JU) = ARL(IU, JU+1)
      RHS7(IU, 1) = -ARL(IU, 1)
      DO 155 JU = 1, 4
155 LMAT7(IU, JU+3) = -BIM(IU, JU)
      DO 160 IU = 1, 3
      RHS7(IU+4, 1) = -BIM(IU+1, 1)
      DO 160 JU = 1, 3
160 LMAT7(IU+4, JU) = BIM(IU+1, JU+1)

```

```

DO 165 IU = 1, 7
ZXC = LMAT7(4, IU)
LMAT7(4, IU) = LMAT7(5, IU)
165 LMAT7(5, IU) = ZXC
ZXC = RHS7(4, 1)
RHS7(4, 1) = RHS7(5, 1)
RHS7(5, 1) = ZXC
CC
CC SOLVE 7 X 7 NON-HOMOGENEOUS SET OF EQUATION
CC
IDGT = 0
N7 = 7
N77 = 7
N1 = 1
IER = 0
CALL ZLEQT2(LMAT7, N1, N77, N7, RHS7, IDGT, WK7, IER)
CC
CC FORM THE EIGENVECTOR ASSOCIATED WITH THE SKIN FOLLOWED BY THE
CC EIGENVECTOR ASSOCIATED WITH THE STIFFENER.
CC
BL(1, J) = CMPLX(1.0, RHS7(4, 1))
BL(2, J) = CMPLX(RHS7(1, 1), RHS7(5, 1))
BL(3, J) = CMPLX(RHS7(2, 1), RHS7(6, 1))
BL(4, J) = CMPLX(RHS7(3, 1), RHS7(7, 1))
CC
DO 170 IU = 1, 4
T1(IU, 1) = CZERO
AL(IU, J) = CZERO
DO 170 JU = 1, 4
T1(IU, 1) = T1(IU, 1) + CMPLX(ARL(IU, JU), BIM(IU, JU)) * BL(JU, 1)
170 AL(IU, J) = AL(IU, J) + T2(IU, JU) * BL(JU, J)
CC *****
CC *****
AD7(J) = 0
DO 185 IJU = 1, 4
AD7(J) = AD7(J) + DABS(DREAL(T1(IJU, 1)))
185 IF (IJU.NE.1) AD7(J) = AD7(J) + DABS(DIMAG(T1(IJU, 1)))
IF (AD7(J).GT.4.0E05.AND.IER.NE.0) THEN
NRT = 10
NSVV = NSVV + 1
ISVV(NSVV) = JCT
END IF
IF (NRT.EQ.0) LAMDA2(J) = LAMDA(JCT)
CC *****
DO 220 K = 1, 4
DO 190 I = 1, NJ
AL2T = ( ( XC(I) + MU(1, K) * H1 ) ** LAMDA(JCT) )
CMAT(I, J) = CMAT(I, J) + AL(K, J) * AL2T
190 CMAT(NR1S+I, J) = CMAT(NR1S+I, J) + MU(1, K) * AL(K, J) * AL2T
DO 195 I = 1, NJ1
AL2T = ( ( D + MU(1, K) * XC(100+I) ) ** LAMDA(JCT) )
CMAT(NR2NT+I, J) = CMAT(NR2NT+I, J) + MU(1, K) * MU(1, K) * AL(K, J) * AL2T
195 CMAT(NR2ST+I, J) = CMAT(NR2ST+I, J) + MU(1, K) * AL(K, J) * AL2T
CC
DO 200 I = 1, NJ2R
AL2T = ( ( D + MU(2, K) * XC(100+NJ1+I) ) ** LAMDA(JCT) )
CMAT(NR2NB+I, J) = CMAT(NR2NB+I, J) + MU(2, K) * MU(2, K) * BL(K, J) * AL2T
200 CMAT(NR2SB+I, J) = CMAT(NR2SB+I, J) + MU(2, K) * BL(K, J) * AL2T
CC
DO 205 I = 1, NJ
AL2T = ( ( XC(180+I) - MU(2, K) * H2 ) ** LAMDA(JCT) )
CMAT(NR3N+I, J) = CMAT(NR3N+I, J) + BL(K, J) * AL2T
205 CMAT(NR3S+I, J) = CMAT(NR3S+I, J) + MU(2, K) * BL(K, J) * AL2T
CC
DO 220 I = 1, NJ2L
AL2T = ( (-L + MU(2, K) * XC(280+I) ) ** LAMDA(JCT) )

```

```

      CMAT(NR4N+I,J) = CMAT(NR4N+I,J) + MU(2,K)*MU(2,K)*BL(K,J)*AL2T
      CMAT(NR4S+I,J) = CMAT(NR4S+I,J) + MU(2,K) * BL(K,J) * AL2T
220  CONTINUE
225  CONTINUE
      IF (NRT.NE.0) GO TO 135
      DO 338 I = 1, NLAM
338    LAMDA(I) = LAMDA2(I)
C *****
C *****
C NEW PART TO DETERMINE IF TWO EIGENVALUES ARE THE SAME (OR WITHIN
C 5 PERCENT OF EACH OTHER).
C *****
C *****
      LAMDA2(NLAM+1) = 22222.00
      IK = 0
      I = 2
129  CONTINUE
      RATIO = REAL(LAMDA2(I)) / REAL(LAMDA2(I-1))
      IF ( RATIO.GT.0.95.AND .RATIO.LT.1.05) THEN
        IF ( AD7(I-1).LT.AD7(I) ) THEN
          IK = IK+1
          LAMDA2(IK) = LAMDA2(I-1)
          FX(IK) = FX(I-1)
          INFER(IK) = INFER(I-1)
          AD7(IK) = AD7(I-1)
          I = I + 1
        ELSE
          I = I + 1
          IK = IK+1
          LAMDA2(IK) = LAMDA2(I-1)
          FX(IK) = FX(I-1)
          INFER(IK) = INFER(I-1)
          AD7(IK) = AD7(I-1)
        END IF
      ELSE
        IK = IK+1
        LAMDA2(IK) = LAMDA2(I-1)
        FX(IK) = FX(I-1)
        INFER(IK) = INFER(I-1)
        AD7(IK) = AD7(I-1)
      END IF
      I = I + 1
      IF (I.LE.NLAM+1) GO TO 129
      IF (NLAM.NE.IK) THEN
        NLAM = IK
        NL2 = NLAM
        NSVV = 0
        DO 248 I = 1, NLAM
248    ISVV(I) = 0
        LAMDA(I) = LAMDA2(I)
        GO TO 135
      ELSE
        NL2 = IK
        NLAM = IK
        NSVV = 0
      END IF
CC
CC USED TO DETERMINE EXACTLY WHICH LAMDAS TO USE
CC
      NLAMT = NLAM
      II = 0
      IF (NLAMT.LT.NLAM) THEN
        DO 279 I = 1, NLAMT
        WRITE(*,1155) I
        READ(*,*) IJ(1)
        DO 279 JJ = 1, NLAM

```



```

                IF (JJ.EQ.IJ(1)) LAMDA(I) = LAMDA2(JJ)
279          CONTINUE
                DO 283 I = 1, NLAMT
283          LAMDA2(I) = LAMDA(I)
                NLAM = NLAMT
                NL2 = NLAMT
                NSVV = 0
                II = 1
                END IF
                DO 274 I = 1, NLAM
274          ISVV(I) = 0
                IF (II.EQ.1) GO TO 135
CC *****
CC      WRITE OUT THE C MATRIX TO SEE WHAT IT CONSISTS OF
CC *****
                IF (NLAM.LT.5) WRITE(31,946) NLAM
                IF (NLAM.LT.5) WRITE(*,946) NLAM
                DO 230 I = 1, NRTL
                DO 230 JJ = 1, NLAM
230          CMAT5(I,JJ) = CMAT(I,JJ)
CC *****
CC      FIND THE TRANSPOSE TIMES THE MATRIX
CC *****
                LVN = NLAM
                DO 290 K = 1, LVN
                DO 290 I = 1, LVN
                CSMAT(K,I) = CZERO
                DO 290 J = 1, NRTL
290          CSMAT(K,I) = CSMAT(K,I) + CONJG(CMAT(J,K)) * CMAT(J,I)
                DO 295 I = 1, LVN
                B(I,1) = CZERO
                DO 295 J = 1, NRTL
                B(I,1) = B(I,1) + CONJG(CMAT(J,I)) * RHS(J)
295          B(I,2) = B(I,1)
CC *****
CC      CALL THE INVERSION SUBROUTINE AX=B WHERE A IS THE CMAT MATRIX
CC      B IS THE IDENTITY MATRIX AND X WITH BE THE OUTPUT C INVERSE...
CC *****
                DO 300 II = 1, LVN
                BV(II) = B(II,1)
                DO 300 JK = 1, LVN
300          CMATS(II,JK) = CSMAT(II,JK)
                IJOB = 0
                II = 1
                NDRC2 = NDRC
                CALL ZLEQ2C(CSMAT, LVN, NDRC2, B, II, NDRC, IJOB, WAA,WKK,IER)
                DO 340 I = 1, LVN
340          CONST(I) = B(I,1)
                CON1 = CZERO
                DO 353 I = 1, LVN
                AL2T = CZERO
                DO 352 JJ = 1, LVN
352          AL2T = AL2T + CMATS(I,JJ) * CONST(JJ)
                CON1 = CON1 + CDABS(AL2T-BV(I))
353          Continue
                IF (CON1.GT.100) WRITE(31,1099)
                IF (CON1.GT.100) WRITE(*,1099)
C *****
CC      BEGINNING OF WRITE STATEMENTS
CC *****
                IF (NSIDE.EQ.1.OR.NSIDE.EQ.3) WRITE(31,1105) L,D,
                    H1,H2,ANGLE1,NCP,NLAM,-N4,VXY4, M4
                IF (NSIDE.EQ.2) WRITE(31,1106) L,D,
                    H1,H2,ANGLE1,NCP,NLAM,PXY, TLL, HEIGHT
                IF (NSIDE.EQ.3) WRITE(31,1107) A3, B3, C3
                WRITE(31,1108) A0N, A0S, ULTN, ULTS

```

```

        WRITE(31,1110)
        WRITE(31,1115) ZEX(1), ZEZ(1), ZEY(1)
        WRITE(31,1120) ZEX(2), ZEZ(2), ZEY(2)
        WRITE(31,1125)
        WRITE(31,1116) ZGXY(1)
        WRITE(31,1121) ZGXY(2)
        WRITE(31,1130)
        WRITE(31,1115) NUYZ(1), NUXY(1), NUXZ(1)
        WRITE(31,1120) NUYZ(2), NUXY(2), NUXZ(2)
CC *****
CC      THE CONSTANTS NOW ARE ALL SOLVED FOR.  THE SUBROUTINE
CC      THAT CONTAINS THE EXPRESSIONS FOR STRESS ARE NOW USED
CC      TO REFIGURE STRESS AT COLLOCATION POINTS.
CC *****
      DO 375 I = 1, 300
      IF ( XC(I).EQ.ZERO ) GO TO 375
      X2 = -I
      Y2 = XC(I)
      IRT = 1
      IF ( I.GT.280 ) GO TO 365
      X2 = XC(I)
      Y2 = -H2
      IRT = 2
      IF ( I.GT.180 ) GO TO 365
      X2 = D
      Y2 = XC(I)
      IRT = 1
      IF ( I.GT.100 ) GO TO 365
      X2 = XC(I)
      Y2 = H1
      IRT = 2
365    CALL STRS1(CONST,LAMDA2,NLAM,X2,Y2,STX(1),STY(1),STXY(1))
375    CONTINUE
CC *****
CC      CALCULATE THE STRESS ALONG THE INTERFACE AS WELL AS THE AVERAGE
CC      NORMAL, AVERAGE SHEAR STRESS AND FAILURE FUNCTION. (QUADRATIC
CC      USING THE AVERAGE STRESSES.)
CC *****
      AVNSTR = ZERO
      AD7(1) = 0
      AON2 = 10000.
      NSIGN = 0
      KUP = 0
      KCT = 100
      AVSSTR = ZERO
C
      hldsy = 0.0
      zjxx = 0.0052
      DO 380 II = 1, 100
      jzj = 20
      X2N = II/ 100.0 * AON
      X2S = II/ 100.0 * AOS
      Y2 = -0.0000000001
      CALL STRS1(CONST,LAMDA2,NLAM,X2N,Y2,STX(1),STY(1),STXY(2))
      CALL STRS1(CONST,LAMDA2,NLAM,X2S,Y2,STX(1),STY(2),STXY(1))
      if (x2n.lt.zjxx) then
      x2hld(ii) = x2n
      styhld(ii) = sty(1)
      stxhld(ii) = stxy(1)
      iv = ii
      if (styhld(ii).gt.hldsy) then
      hldsy = styhld(ii)
      hldsx = stxhld(ii)
      hldx2 = x2hld(ii)
      end if
      jzj = 0

```

```

end if
  ivv = iv + 1
  if (jzj.eq.0) go to 380
  if (ii.eq.ivv) then
    do 379 iii = 1, iv
      if (hldx2.gt.x2hld(iii).or.styhld(iii).lt.0) then
        styhld(iii) = hldsy
        stxhld(iii) = hldsx
        x2hld(iii) = - x2hld(iii)
      end if
      if (styhld(iii).gt.0) then
        AVNSTR = AVNSTR + STYhld(iii) / 100.0
      end if
      AVSSTR = AVSSTR + ABS(STXhld(iii)) / 100.0
379    continue
    end if
    IF (STY(1).gt.0) then
      AVNSTR = AVNSTR + STY(1) / 100.0
    END IF
    AVSSTR = AVSSTR + ABS(STXY(1)) / 100.0
380    CONTINUE
    FFAIL = ((AVNSTR*AVNSTR) / (ULTN*ULTN)) + ((AVSSTR*AVSSTR) /
      (ULTS*ULTS))
    FFAIL = SQRT( FFAIL )
    WRITE(31,1163)
    WRITE(31,1166) AVNSTR, AVSSTR, FFAIL
CC
CC
    a0n2 = a0n
CC
    IF (NSIDE.GT.1) THEN
      AVNSTR = 0.0E0
      AVSSTR = 0.0E0
      AD7(1) = 0
      A0N2 = 10000.
      NSIGN = 0
      KUP = 0
      KCT = 100
      DO 390 II = 1, 100
        IF (A0N2.LT.A0N) GO TO 390
        X2N = D - (II/ 100.0 * A0N)
        X2S = D - (II/ 100.0 * A0S)
        Y2 = -0.0000000001
        AD7(1) = ABS(STY(1))
        CALL STRS1(CONST,LAMDA2,NLAM,X2N,Y2,STX(1),STY(1),STXY(2))
        CALL STRS1(CONST,LAMDA2,NLAM,X2S,Y2,STX(1),STY(2),STXY(1))
        IF (X2N.LT.D-0.065) NSIGN = ABS(STY(1)) - AD7(1)
        IF (NSIGN.GT.0) KUP = KUP + 1
        IF (NSIGN.LT.0) KUP = 0
        IF (KUP.GT.9) THEN
          A0N2 = D - X2N
          KCT = II
        ELSE
          AVNSTR = AVNSTR + ABS(STY(1)) / 100.0
          AVSSTR = AVSSTR + ABS(STXY(1)) / 100.0
        END IF
390      CONTINUE
      IF (A0N2.GT.A0N) A0N2 = A0N
      AVNSTR = AVNSTR * 100.0 / KCT
      AVSSTR = AVSSTR * 100.0 / KCT
      FFAIL = ((AVNSTR*AVNSTR) / (ULTN*ULTN)) + ((AVSSTR*AVSSTR) /
        (ULTS*ULTS))
      FFAIL = SQRT( FFAIL )
      WRITE(31,*) ' LENGTHS: A0N, A0S, A0N2, KCT ', A0N,A0S,A0N2,KCT
      WRITE(31,1164)
      WRITE(31,1166) AVNSTR, AVSSTR, FFAIL

```

```

      END IF
      STOP
CC *****
901  FORMAT(//)
1103 FORMAT(/,5X,'N = ',I3,6X,'EPS = ',F15.11)

900  FORMAT(/,13X,'THIS IS ***** WEBSTER *****',/,
      5X,'THE WEB-STIFFENER TERMINATION ELASTICITY ROUTINE ',/,
      5X,'Developed by Northrop - Aircraft Division ',/,
      5X,'Subcontractor to MCAIR - St. Louis.',/,
      5X,'Out-of-Plane Loading Navy/FAA Contract ',/,
      5X,'For problems contact: ',/,5X,' Mary Mahler ',
      '(310) 332-9648 or Han-Pin Kan (310) 332-5285',/)
903  FORMAT(/,3X,'COLLOCATION POINTS: NO, (X OR Y POSITION)')
910  FORMAT(/,8X,'The geometry of the structure is defined as follows:'
      /,8X,'The origin of the coordinate system is at point A.',/,8X,
      '
      y ',/,8X,
      | ',/,8X,'
      | B-----C ',/,8X,
      | ',/,8X,
      | ',/,8X,
      F-----A-----G-----x ',/,8X,
      | ',/,8X,
      | ',/,8X,
      E-----D ',/,8X,
      'L (distance from F to A), D (distance from A to G)',/,8X,
      'ANGLE (angle from the x-axis to line A-B (in degrees)):',/,8X,
      'Input L, D, ANGLE ?',/)
911  FORMAT(/,15X,'** ECHO INPUT **',/,2X,
      'NO OF COLLOCATION PTS.= ',I3,2X,'TERMINATION ANGLE = ',F9.2,/,
      2X,'LENGTH OF SKIN = ',F9.2,3X,'LENGTH OF SKIN AND STIFFENER = ',
      F9.2,/)
912  FORMAT(/,8X,'THE FOLLOWING IS A DESCRIPTION OF THE LOADING',
      /,8X,'CONDITIONS AVAILABLE. THERE ARE 4 TYPES OF INPUT',/,
      10X,'REACTION LOADS AVAILABLE. INPUT THE NUMBER DESIRED.',/,12X,
      '1) FORCES APPLIED TO SKIN SIDE, REACTIONS OCCURING ON SKIN',/,
      15X,'STIFFENER SIDE.',/,
      12X,'2) HAT-STIFFENER PULL-OFF, CLAMPED ON THE ENDS.',/,
      12X,'3) PRESSURE AS WELL AS SKIN LOADING, REACTION ON',/,
      15X,'SKIN/STIFFENER SIDE.',/,12X,'4) INPUT EACH REACTION',
      'STRESS INDIVIDUALLY',/,8X,'INPUT 1, 2, 3, OR 4 (NSIDE)')
913  FORMAT(/,8X,'FOR THE APPLIED FORCES ON THE SKIN SIDE: ',/,
      8X,'INPUT THE VALUE OF NORMAL FORCE, SHEAR FORCE, AND MOMENT.')
914  FORMAT(/,8X,'FOR THE SKIN/STIFFENER FLANGE SIDE INPUT:',/,
      8X,'THE NORMAL FORCE, SHEAR FORCE AND MOMENT PER UNIT LENGTH')
915  FORMAT(/,8X,'FOR THE HAT-PULLOFF SPECIMEN, INPUT THE FOLLOWING',
      /,8X,'FORCE APPLIED TO THE HAT, TOTAL LENGTH OF SPECIMEN, AND '
      'HEIGHT OF HAT.')
916  FORMAT(/,8X,'FOR THE APPLIED FORCES ON THE SKIN SIDE: ',/,
      8X,'INPUT THE VALUE OF NORMAL FORCE, SHEAR FORCE, AND MOMENT.')
917  FORMAT(/,8X,'The next input the Boundary Forces and Moments.',
      /,8X,'SURFACE C-D INPUT APPLIED: NORMAL FORCE, SHEAR FORCE(xy) '
      'AND MOMENT',/,8X,' (POSITIVE MOMENT GIVES AN UPWARD BENDING'
      'moment)',/,8X,' Input Nx, Vxy, Mz ?',/)
918  FORMAT(/,8X,'The next input the Boundary Forces and Moments.',
      /,8X,'SURFACE E-F INPUT APPLIED: NORMAL FORCE, SHEAR FORCE(xy) '
      'AND MOMENT',/,8X,' (POSITIVE MOMENT GIVES AN UPWARD BENDING'
      'moment)',/,8X,' Input Nx, Vxy, Mz ?',/)
920  FORMAT(/,8X,'1 of 3 pressure functions may be on Surface E-D:',
      /,8X,'1) Linear or Constant pressure ',/,8X,
      '2) Parabolic pressure function (where the vertex is known)',/,
      8X,'3) A spline (Lagrangian) fitted through any 3 known values',/,
      8X,'Input 1, 2 or 3 for the type of function desired. ',/)
921  FORMAT(/,8X,'Forces applied to the skin only side.',/,8X,
      '(Reactions occurring uniformly on the skin/stiffener edge)',/,
      8X,'Normal Force = ',F10.2,3X,'Shear Force = ',F10.2,/,8X,
      'Bending Moment = ',F10.2,/)

```

```

922  FORMAT(/,3X,'Hat Stiffener Pull-off Specimen.',/,3X,
    'Pull-off Force on Model = ',F10.2,3X,'Total specimen length = ',
    F10.2,/,3X,'Height of Hat = ',F10.2,/)
923  FORMAT(/,8X,'Forces applied to the skin side with pressure',
    ' loading.',/,8X,
    '(Reactions occurring uniformly on the skin/stiffener edge)',/,
    8X,'Normal Force = ',F10.2,3X,'Shear Force = ',F10.2,/,8X,
    'Bending Moment = ',F10.2,/,8X,
    'Normal Pressure function on the bottom surface is:',/,12X,F9.3,
    '*X*X + ',F9.3,' *X + ',F9.3,/)
925  FORMAT(/,8X,'For this loading condition on the bottom surface',/,
    8X,'there may be a linear, constant or zero load applied.',/,8X,
    'Input the x-location and force at any 2 points on the surface.',
    /,8X,'Input X1, P1, X2, P2 ',/)
930  FORMAT(/,8X,'For a parabolic loading, the x-location and',/,8X,
    'maximum (or minimum) force needs to be known. Input XS, PS',/)
935  FORMAT(/,8X,'Input the x-location and force at any other point',
    /,8X,'on the surface for the parabolic distribution to be',/,8X,
    'calculated.',/,8X,'Input X1, P1 :')
940  FORMAT(/,8X,'This function forms a spline between any three',/,
    8X,'forces at any three locations using a Lagrangian spline.',/,
    8X,'Input the x-location and force at the three locations.',/,8X,
    'Input: X1 ,P1, X2, P2, X3, P3 ',/)
943  FORMAT(/, ' ***** WARNING ***** INPUT WARNING *****',
    /, ' The length inputted is ',F9.4,'!',/,
    ' This is less then 0.10, which is a minimum value of flange',
    /, ' length that will produce a reasonable and valid stress',
    /, ' state. Try to redesign model to increase this length.',/)
945  FORMAT(/, ' ***** WARNING ***** INPUT WARNING *****',
    /, ' The lengths of ',F10.3,' and ',F10.3,/,
    ' have a ratio between each other which is greater then 7.',
    /, ' The model should be redesigned to reduce this ratio.',
    /, ' This could result in a faulty stress state.',/)
946  FORMAT(/, ' ***** EIGENVALUE WARNING ***** ',/,
    3X,'The number of eigenvalues found by the IMSL routine is ',I3,
    ' ',/,3X,'This is less then 5, which could result in an ',
    'improper stress state.',/,3X,'The differing magnitudes of ',
    'constants for each eigenvalue is a way to verify',/,3X,
    'that the number of eigenvalues obtained is appropriate.',/)
955  FORMAT(/, ' ATTENTION: NTYPE MUST EITHER EQUAL 1 OR 2 OR 3',/)
960  FORMAT(/, ' The total number of equations used is ',I3,/)
965  FORMAT(2(9X,I4,' ',F8.3))
969  FORMAT(/,3X,'Force Array (Right Hand Side of Equation):',
    ' No., Value No., Value')
970  FORMAT(2(9X,I4,' ',F15.3))
975  FORMAT(/,3X,'Material Properties: Stiffener/Skin',/,
    6X,'I ',J',13X,'BETA',16X,'A-MAT')
980  FORMAT(3X,I4,2X,I4,7X,E13.4,7X,E13.4)
984  FORMAT(/, ' MU of the stiffener = ',/,3X,I3,2(4X,E11.5,' ',E11.5),
    /,6X,2(4X,E11.5,' ',E11.5))
985  FORMAT(/, ' MU of the skin = ',/,3X,I3,2(4X,E11.5,' ',E11.5),
    /,6X,2(4X,E11.5,' ',E11.5))
986  FORMAT(/,5X,'THE MATRIX THAT RELATES ALPHAS TO BETAS IS (T2)',
    4(/,4(1X,E9.3,' ',E9.3)) )
990  FORMAT(/, ' THE LAMDA'S FOUND FROM THE IMSL ROUTINE ARE:',/,
    4X,'ID NO.',6X,'Value of Lamda',13X,'F(X)',8X,'Iter. Needed',/)
1000  FORMAT(3X,I4,2(3X,F10.4,' ',F10.4),5X,I5)
1001  FORMAT(3X,I4,2(3X,F10.4,' ',F10.4),5X,I5,5X,F10.2)
1025  FORMAT(/, ' Lamda equals ',3X,I4,5X,E11.5,' ',E11.5)
1050  FORMAT(' INPUT THE NAME OF THE OUTPUT FILE ')
1055  FORMAT(' Alpha & beta eigenvectors = ',2(3X,E11.5,' ',E11.5))
1060  FORMAT(' The Zero matrix check on F * Beta',/,
    2(4X,E11.5,' ',E11.5),/,2(4X,E11.5,' ',E11.5))
1061  FORMAT(/,3X,'The number of lamda (eigenvalues) to be used is',
    I3,/,3X,'Equal or fewer number of these may be used.',/,3X,
    'How many are desired in this run?',/)

```

```

1094  FORMAT(/,3X,'B of row ',I3,' for the system of eqtn = (',E14.7,
      ' ',E14.7,')')
1095  FORMAT(' THE C-SQUARE MATRIX = ',3X,I3,/,2(3X,E14.7,',' ,E14.7),/,
      2(3X,E14.7,',' ,E14.7),/,2(3X,E14.7,',' ,E14.7),/,
      2(3X,E14.7,',' ,E14.7))
1099  FORMAT(/,5X,'WARNING: Final square complex matrix did not',
      ' invert fully.',/,5X,'This could result in an error.',/,5X,
      ' Check the stresses with the collocation point stresses.',/)
1102  FORMAT(/,5X,'Cmats * Constant = ',E15.9,',' ,E15.9,/,5X,
      ' Cmat * RHS (B) = ',E15.9,',' ,E15.9)
11    FORMAT(9X,'The Name of this file is ',A35,/)
1105  FORMAT(/,21X,'y ',/,21X,'| ',/,21X,'| ',/,21X,'| ',/,21X,'| ',
      ' B-----C ',/,
      ' | ',/,
      ' - - - G - - - x ',/,
      ' | ',/,
      ' E-----D ',/,5X,
      ' The length of the skin (F to A) is ',F8.5,/,5X,
      ' The length of the stiffener (A to G) is ',F8.5,/,5X,
      ' The thickness of the stiffener (G to C) is ',F8.5,/,5X,
      ' The thickness of the skin (G to D) is ',F8.5,/,5X,
      ' The angle surface A-B makes with the X-axis is ',F8.3,
      ' degrees.',/,5X,' This run considers ',I3,
      ' collocation points.',/,6X,' It uses ',I3,' eigenvalues.',/,8X,
      ' THE NORMAL FORCE IN THE X-DIRECTION IS ',F16.3,/,8X,
      ' THE SHEAR FORCE ON SURFACE C-D IS ',F16.3,/,
      8X,' THE MOMENT ON THE LEFT FACE IS ',F16.3,/)
1106  FORMAT(/,21X,'y ',/,21X,'| ',/,21X,'| ',/,21X,'| ',/,21X,'| ',
      ' B-----C ',/,
      ' | ',/,
      ' - - - G - - - x ',/,
      ' | ',/,
      ' E-----D ',/,6X,
      ' The length of the skin (F to A) is ',F8.5,/,6X,
      ' The length of the stiffener (A to G) is ',F8.5,/,6X,
      ' The thickness of the stiffener (G to C) is ',F8.5,/,6X,
      ' The thickness of the skin (G to D) is ',F8.5,/,6X,
      ' The angle surface A-B makes with the X-axis is ',F8.3,
      ' degrees.',/,6X,' This run considers ',I3,
      ' collocation points.',/,6X,' It uses ',I3,' eigenvalues.',/,8X,
      ' The model pull-off force is ',F16.3,/,8X,
      ' Total length of the structure is ',F16.3,/,
      8X,' Height of the hat is ',F16.3,/)
1107  FORMAT(8X,'Normal Pressure function on the bottom surface is',
      ' described',/,12X,' by the quadratic equation ',/,12X,F9.3,
      ' *X*X + ',F9.3,' *X + ',F9.3,/)
1110  FORMAT(/,9X,'Material properties of the stiffner and skin are:',
      ' ',/,19X,' EX ',14X,' EY ',14X,' EZ')
1114  FORMAT(3X,'STRESSES WILL BE INPUTED FOR EACH COL. PT.')
1115  FORMAT(' STIFFENER = ',E12.4,4X,E12.4,4X,E12.4)
1116  FORMAT(' STIFFENER = ',E12.4,4X)
1117  FORMAT(3X,'THE NUMBER MUST BE 1, 2, OR 3 ONLY.')
1120  FORMAT(' SKIN = ',E12.4,4X,E12.4,4X,E12.4)
1121  FORMAT(' SKIN = ',E12.4,4X)
1122  FORMAT(4X,'WHAT IS THE LARGEST REAL PART OF LAMDA ACCEPTED? ',
      ' ( <12)')
1123  FORMAT(4X,'WHAT IS THE LARGEST IMAGINARY PART OF LAMDA ',
      ' ACCEPTED? ( < 15)')
1125  FORMAT(/,19X,' GX2 ')
1130  FORMAT(/,19X,' VYZ ',11X,' VXZ ',12X,' VXY')
1133  FORMAT(3X,'LAMDA = ',F11.6,',' ,F11.6,3X,'FOR I = ',I3,3X,'FX = ',
      ' F11.6,',' ,F11.6)
1135  FORMAT(/,5X,'The solution used ',I3,' eigenvalue(s).',
      ' ',5X,'The eigenvalues were as follows:',/,7X,
      ' Number',15X,' LAMDA',26X,' C(LAMDA)')
1140  FORMAT(8X,I4,7X,E12.5,',' ,E12.5,7X,E12.5,',' ,E12.5)

```

```

1144 FORMAT(/,5X,'THE NUMBERS OF BAD LAMDAS ARE',I3)
1145 FORMAT(' CONSTANTS OF THE EIGENVECTOR = ',I4,4X,E12.5,', ',E12.5)
1150 FORMAT(/,9X,'Stress at the collocation points:',/,7X,
        'X',10X,'Y',17X,'NORMAL',14X,'SHEAR',/)
1155 FORMAT(3X,'INPUT THE LAMDA ID NO FOR YOUR LAMDA = ',I3)
1160 FORMAT(3X,F8.4,3X,F8.4,5X,2(F16.2,4X) )
1165 FORMAT(/,' Interlaminar stress along the interface ',/,7X,
        'X',10X,'Y',17X,'NORMAL',14X,'SHEAR',/)
1108 FORMAT(/,' For an A0 (Normal) of ',F11.4,/,
        ' For an A0 (Shear) of ',F11.4,/,
        ' GIVEN ULTIMATE NORMAL STRESS = ',F12.2,/, ' ULTIMATE',
        ' SHEAR STRESS = ',F12.2,/)
1163 FORMAT(/,2X,'NEAR THE POINT OF TERMINATION: AT POINT A:')
1164 FORMAT(/,2X,'NEAR THE C-D SURFACE: AT POINT G:')
1166 FORMAT(' THE AVERAGE NORMAL STRESS IS ',
        F16.2,/, ' THE AVERAGE SHEAR STRESS IS ',F16.2,/,
        ' FAILURE FUNCTION (QUADRATIC) IS ',F16.4,/)
1167 FORMAT(3X,'INPUT THE STRESSES OF EACH BOUNDARY CONDITION')
1168 FORMAT(3X,'FOR ROW = ',I3)
1170 FORMAT(/,' How many other points do you wish the stresses at?',/)
1175 FORMAT(/,' Other points computed stress states.',/)
1180 FORMAT(/,' INPUT X, Y, AND (1 IF X IS NORMAL COMPONENT), OR ',/,
        ' (2 IF Y IS NORMAL COMPONENT) ',/)
1188 FORMAT(3X,'N = ',I3,3X,'NGUESS = ',I3,3X,'ITMAX = ',I3)
1189 FORMAT(3X,'N = ',I3,3X,'KN = ',I3,3X,'ITMAX = ',I3)
1196 FORMAT(3X,'J = ',I3,3X,'AD7(J) = ',F12.4,3X,'IER = ',I3)
1200 FORMAT(/,' The total load on the right face (prior to the moment
        being added)',/,5X,' in the X-direction is ',F13.4,/,5X,
        ' in the Y-direction is ',F13.4,/,5X,' Thus, the moment created
        from these forces is ',F13.4,/)
1205 FORMAT(' The total load on the left face (prior to a moment being
        added)',/,5X,' in the X-direction is ',F13.4,/,6X,
        ' in the Y-direction is ',F13.4,/,4X,' Thus, the moment created
        from these forces is ',F13.4,/)
1210 FORMAT(/,' The total load on the bottom surface is ',F13.4,/,
        ' Thus, the moment from this surface force is ',F13.4,/)
1235 FORMAT(3X,'The neutral axis (for the bending moment) is at Y = ',
        F13.5)
CC *****
CC SET UP THE RIGHT HAND SIDE OF THE EQUATION CMAT*C=RHS
CC *****
2001 CONTINUE
      IF ( NSIDE.EQ.1) THEN
C
C THIS IS FOR NSIDE = 1. IT DETERMINES THE NORMAL AND SHEAR
C STRESS ONLY...MOMENT FIGURED ON
C SURFACE 4 ONLY...NOT ON SURFACE 2
      DO 2005 I = 1, NJ2L
        RHS(NR4N+I) = N4 - (( 6.0 * M4 / (H2*H2*H2) ) *
          ( 2.0 * XC(280+I) + H2 ) )
        RHS(NR4S+I) = -4.0 * VXY4 / H2 * ( XC(280+I)/H2 + 1 )
        DUM = -H2 / 2.0
2005 IF (XC(280+I).GE.DUM) RHS(NR4S+I) = 4 * VXY4 * XC(280+I)/(H2*H2)
        DO 2010 I = 1, NJ1
          RHS(NR2ST+I) = VM2T * (XC(100+I)+H2) / (MD2T + H2)
          IF (XC(100+I).GE.MD2T)
            RHS(NR2ST+I) = VM2T * ( XC(100+I)-H1 ) / (MD2T-H1)
2010 RHS(NR2NT+I) = N2T
          DO 2015 I = 1, NJ2R
            RHS(NR2SB+I) = VM2B * (XC(100+NJ1+I)+H2) / (MD2B+H2)
            IF ( XC(100+NJ1+I) .GE. MD2B)
              RHS(NR2SB+I) = VM2B * ( 2.*MD2B-XC(100+NJ1+I)+H2) / (MD2B+H2)
2015 RHS(NR2NB+I) = N2B
          ELSE
            IF (NSIDE.EQ.2) THEN

```

```

C      NEXT IS FOR NSIDE = 2.    ALL BOUNDARY CONDITIONS
C
      DO 2070 I = 1, NJ2L
      RHS(NR4S+I) = -4.0 * PXY * ( H2 + XC(280+I) ) / (H2*H2)
      IF ( XC(280+I) .GE. (-H2/2.0) )
        RHS(NR4S+I) = 4.0 * PXY * XC(280+I) / (H2*H2)
2070   RHS(NR4N+I) = - (6.0*M4/(H2*H2*H2)) * ( 2.0*XC(280+I)+H2 )
      DO 2075 I = 1, NJ1
      RHS(NR2ST+I) = 4.0 * PXY * (H1 - XC(100+I) ) / (H1*H1)
      IF (XC(100+I) .LT. (H1/2))
        RHS(NR2ST+I) = 4.0 * PXY * XC(100+I) / (H1*H1)
2075   RHS(NR2NT+I) = - (6.0*M2T/(H1*H1*H1)) * ( 2.0*XC(100+I)-H1 )
      DO 2080 I = 1, NJ2R
2080   RHS(NR2NB+I) = - (6.0*M2B/(H2*H2*H2)) * ( 2.*XC(100+NJ1+I)+H2 )
      ELSE
C
C      THIS IS FOR NSIDE = 3. IT DETERMINES THE NORMAL AND SHEAR
C      STRESS ONLY....MOMENT FIGURED ON
C      SURFACE 4 ONLY...NOT ON SURFACE 2
      V3T = 0.0
      DO 2100 I = 1, NJ
      RHS(NR3N+I) = NY3(A3,B3,C3,XC(180+I))
      V3T = V3T - DA3 * RHS(NR3N+I)
      M3T = M3T + XC(180+I) * DA3 * RHS(NR3N+I)
2100   CONTINUE
      VM2T = 2.0 * (VXY4+V3T) / (H1 + H2)
      MD2T = (H1 - H2) / 2.0
      IF (MD2T.LE.ZERO) VM2T = H1 * VM2T / (H1-MD2T)
      IF (MD2T.LE.ZERO) MD2T = ZERO
      VM2B = 2.0 * (VXY4+V3T) / (H1 + H2)
      MD2B = (H1 - H2) / 2.0
      DO 2105 I = 1, NJ2L
      RHS(NR4N+I) = N4 - ( ( 6.0 * M4 / (H2*H2*H2) ) *
        ( 2.0 * XC(280+I) + H2 ) )
      RHS(NR4S+I) = -4.0 * VXY4 / H2 * ( XC(280+I)/H2 + 1 )
2105   IF (XC(280+I).GE.DUM) RHS(NR4S+I) = 4 * VXY4 * XC(280+I)/(H2*H2)
      DO 2110 I = 1, NJ1
      RHS(NR2ST+I) = VM2T * (XC(100+I)+H2) / (MD2T + H2)
      IF (XC(100+I) .GE. MD2T)
        RHS(NR2ST+I) = VM2T * ( XC(100+I)-H1 ) / (MD2T-H1)
2110   RHS(NR2NT+I) = N2T
      DO 2120 I = 1, NJ2R
      RHS(NR2SB+I) = VM2B * (XC(100+NJ1+I)+H2) / (MD2B+H2)
      IF ( XC(100+NJ1+I) .GE. MD2B)
        RHS(NR2SB+I) = VM2B * ( 2.*MD2B-XC(100+NJ1+I)+H2 ) / (MD2B+H2)
2120   RHS(NR2NB+I) = N2B
      END IF
      END IF
C *****
C      THE FOLLOWING IS FOR A MOMENT ACTING OVER THE ENTIRE SURFACE 2
C *****
      M2A = M2A + M3T
      IF (ABS(M2A).GT.0.001) THEN
      NAX = ( H1 * H1 / A(1,1,1) - H2 * H2 / A(2,1,1) ) / 2.0
      NAX = NAX / ( H1 / A(1,1,1) + H2 / A(2,1,1) )
      IF ( NAX.GE. ZERO ) THEN
        PP1A = ( (H1-NAX)*(H1-NAX) ) + ( NAX*NAX*NAX/(H1-NAX) )
        PP1B = (NAX*H2)/(H1-NAX) * ( (NAX/2.0) + (H2/6.0) )
        PP2A = H1 - NAX - (NAX*NAX/(H1-NAX)) - (NAX*H2/(H1-NAX))
        PP2B = NAX/2.0 + H2/3.0
        FA = -M2A / ( PP1A/3.0 + PP1B + PP2A*PP2B )
        FB = FA * PP2A / H2
        M1 = FA
        M2 = -( FA*NAX ) / ( H1 - NAX )
        M3 = -FB
      ELSE

```



```

PP1A = ( (H2+NAX)*(H2+NAX) ) - ( NAX*NAX*NAX/(H2+NAX) )
PP1B = (NAX*H1)/(H2+NAX) * ( (NAX/2.0) - (H2/6.0) )
PP2A = H2 + NAX - (NAX*NAX/(H2+NAX)) + (NAX*H1/(H2+NAX))
PP2B = -NAX/2.0 + H1/3.0
FB = -M2A / ( PP1A/3.0 + PP1B + PP2A*PP2B )
FA = FB * PP2A / H1
M1 = FA
M2 = -(FB*NAX) / (H2+NAX)
M3 = -FB
END IF
DO 2150 I = 1, NJ1
AD1 = ( XC(100+I) * (M1-M2) / H1 ) + M2
RHS(NR2NT+I) = RHS(NR2NT+I) + AD1
2150 CONTINUE
DO 2180 I = 1, NJ2R
AD2 = ( XC(100+NJ1+I) * ( M3-M2 ) / (-H2) ) + M2
RHS(NR2NB+I) = RHS(NR2NB+I) + AD2
2180 CONTINUE
ELSE
END IF
C *****
C DONE WITH CALCULATIONS, WRITE OUT BOUNDARY STRESSES.
C *****
IF (NSIDE.EQ.4) THEN
WRITE(*,1167)
DO 2188 I = 1, NRTL
RHS(I) = 0.0E0
WRITE(*,1168) I
2188 READ(*,*) RHS(I)
END IF
GO TO 2000
END
C *****
C
C One of the 4 programs: WEB-1, (WEB-2), WEB-3, WEB-4, WEB.LIB
C Last Revision: April 1990. (I/O up-dated at this time)
C
C * * * * *
C *****
C THIS PROGRAM IS THE "FRONT-END" FOR THE ELASTICITY STIFFENER/SKIN
C PROBLEM. ITS OBJECTIVE IS TO SET UP THE LAMINATE PROPERTIES OF THE
C SKIN AND STIFFENER. IT ALSO DETERMINES ALL MATERIAL CHARACTERISTICS
C (IE. ALLOWABLES AND CHARACTERISTIC LENGTH FOR STRESS CONCENTRATIONS.)
C *****
C * * * * *
SUBROUTINE STRMKR(H1, H2, ZEX, ZEY, ZEZ, ZGXY, NUXY, NUXZ, NUYZ,
SIGN, SIGS, AON, AOS)
COMMON/STIFF/MKIND(50), Q(4,50), QBAR(6), TT, NLayer, EX, EY, GXY, ENUXY
COMMON /MATT/ NU12, E1, E2, G12, TMat
REAL ZEX(2), ZEY(2), ZEZ(2), ZGXY(2), NUXY(2), TMat(50), H1, H2,
NUXZ(2), NUYZ(2), SIGN, SIGS, AON, AOS, E1(50), E2(50),
G12(50), NU12(50)
CHARACTER*1 ANS
CHARACTER*4 ESTR
CHARACTER*10 FNLT(40), FNMT(40), FNM
CHARACTER*35 SFILE
C *****
C DETERMINE IF STRUCTURE FILE ALREADY EXIST, AND IF SO, READ IT IN.
C *****
WRITE(*,200)
CALL YORN(ISTS,4)
IF (ISTS.EQ.1) GO TO 60
C *****
C THIS SECTION SETS UP THE MATERIAL PROPERTIES (IF NECESSARY) . IT
C ALLOWS FOR A VARIETY OF MATERIALS TO BE DEFINED AND USED WHEN FORMING
C A SKIN OR STIFFENER.

```

```

C *****
10      CONTINUE
        WRITE(*,205)
        CALL DIRM(FNMT,JFM)
12      ESTR = '.MAT'
        DO 20 I = 1, JFM
            KHD = 9
            DO 17 K = 1, 8
                KK = 9 - K
17          IF (FNMT(I)(KK:KK).EQ.' ') KHD = KK
                SFILE = '
                SFILE(1:20) = 'C:/WEBSTER/MAT-DATA/'
                SFILE(21:20+KHD) = FNMT(I)(1:KHD-1)
                SFILE(21+KHD:24+KHD) = ESTR
                OPEN(15,FILE=SFILE,STATUS='OLD')
                DO 15 J = 1, 5
15          READ(15,'(A)') ANS
                READ(15,*) EXX
                READ(15,*) EYY
                CLOSE(15)
20      WRITE(*,210) I, FNMT(I), EXX, EYY
        CALL YORN(IANS,4)
        IF (IANS.EQ.2) GO TO 30
        WRITE(*,215)
        READ(*,*) NMT
        ESTR = '.MAT'
        KHD = 9
        DO 25 I = 1, NMT
            WRITE(*,220) JFM+I
            READ(*,'(A)') FNMT(JFM+I)
            DO 18 K = 1, 8
                KK = 9 - K
18          IF (FNMT(JFM+I)(KK:KK).EQ.' ') KHD = KK
                SFILE = '
                SFILE(1:20) = 'C:\WEBSTER\MAT-DATA\'
                SFILE(21:20+KHD) = FNMT(JFM+I)(1:KHD-1)
                SFILE(21+KHD:24+KHD) = ESTR
                OPEN(15,FILE=SFILE,STATUS='NEW')
            WRITE(*,222)
            READ(*,*) EXX, EYY, GGXY, V12, T1
            WRITE(15,230)
            WRITE(15,235) EXX
            WRITE(15,240) EYY
            WRITE(15,245) GGXY
            WRITE(15,250) V12
            WRITE(15,255) T1
25      CLOSE(15)
C *****
C IN THIS SECTION, THE MATERIAL SYSTEMS USED IN THE STIFFENER AND SKIN
C ARE ALREADY DEFINED. THE LAMINATE IS NOW FORMED BASED ON THESE
C MATERIAL SYSTEMS. DEFINE THE STIFFENER FIRST.
C *****
30      WRITE(*,206)
        CALL DIRM2(FNMT,JFM)
        ESTR = '.MAT'
        DO 40 I = 1, JFM
            KHD = 9
            DO 27 K = 1, 8
                KK = 9 - K
27          IF (FNMT(I)(KK:KK).EQ.' ') KHD = KK
                SFILE = '
                SFILE(1:20) = 'C:/WEBSTER/MAT-DATA/'
                SFILE(21:20+KHD) = FNMT(I)(1:KHD-1)
                SFILE(21+KHD:24+KHD) = ESTR
                OPEN(15,FILE=SFILE,STATUS='OLD')
                DO 35 J = 1, 5

```

```

35      READ(15,'(A)') ANS
        READ(15,*) E1(I)
        READ(15,*) E2(I)
        READ(15,*) G12(I)
        READ(15,*) NU12(I)
        READ(15,*) TMAT(I)
        CLOSE(15)
        CALL QMAT(E1(I),E2(I),NU12(I),G12(I),Q,I)
40      WRITE(*,210) I, FNMT(I), E1(I), E2(I)
        WRITE(*,265)
C
C      SETS UP THE LAMINATE FOR THE STIFFENER
C
        CALL MAT
        WRITE(*,270) EX, EY, GXY, ENUXY, TT
        WRITE(*,275)
        CALL YORN(IANS,4)
        IF (IANS.EQ.1) GO TO 30
        WRITE(*,260)
        READ(*,*) ZEZ(1)
        NUXZ(1) = 0.3
        NUYZ(1) = 0.3
        H1 = TT
        ZEX(1) = EX
        ZEY(1) = EY
        ZGXY(1) = GXY
        NUXY(1) = ENUXY
C
C      THE NEXT INPUT IS FOR THE SKIN
C
50      WRITE(*,280)
        DO 55 I = 1, JFM
55      WRITE(*,210) I, FNMT(I), E1(I), E2(I)
        CALL MAT
C *****
C      THE SKIN AND STIFFENER PROPERTIES SHOULD NOW HAVE BEEN FORMED.
C      ASK IF THERE ARE ANY CHANGES AND PROCEED TO CHANGING THE ALLOWABLES
C      AND WRITING OUT THE LAMINATE.
C *****
        WRITE(*,300) EX, EY, GXY, ENUXY, TT
        WRITE(*,305)
        CALL YORN(IANS,4)
        IF (IANS.EQ.1) GO TO 50
        ZEZ(2) = ZEZ(1)
        NUXZ(2) = 0.3
        NUYZ(2) = 0.3
        H2 = TT
        ZEX(2) = EX
        ZEY(2) = EY
        ZGXY(2) = GXY
        NUXY(2) = ENUXY
        WRITE(*,306)
        READ(*,*) A0N,A0S,SIGN,SIGS
        GO TO 80
C *****
C      THE SECTION WHERE THE PROPERTIES ARE READ IN AND ONLY ALLOWABLES VARY
C *****
60      CONTINUE
        CALL DIR1(FNLT, JFL)
        WRITE(*,315)
C      read(*,'(A)') SFILE
        DO 65 I = 1, JFL
65      WRITE(*,320) I, FNLT(I)
        READ(*,*) JUFL
        ESTR = '.STR'
        FNM = FNLT(JUFL)

```

```

      KHD = 9
      DO 67 K = 1, 8
      KK = 9 - K
67    IF (FNM(KK:KK).EQ.' ') KHD = KK
      SFILE(1:20)='C:\WEBSTER\STR-DATA\'
      SFILE(21:20+KHD) = FNM(1:KHD-1)
      SFILE(21+KHD:24+KHD) = ESTR
      OPEN(11,FILE=SFILE,STATUS='OLD')
      DO 70 I = 1, 9
70    READ(11,'(A)') ESTR
      READ(11,*) SIGN
      READ(11,*) SIGS
      READ(11,*) AON
      READ(11,*) AOS
      READ(11,'(A)') ESTR
      READ(11,*) H1
      READ(11,*) ZEX(1)
      READ(11,*) ZEY(1)
      READ(11,*) ZEZ(1)
      READ(11,*) NUXZ(1)
      READ(11,*) NUXY(1)
      READ(11,*) NUYZ(1)
      READ(11,*) ZGXY(1)
      READ(11,'(A)') ESTR
      READ(11,*) H2
      READ(11,*) ZEX(2)
      READ(11,*) ZEY(2)
      READ(11,*) ZEZ(2)
      READ(11,*) NUXZ(2)
      READ(11,*) NUXY(2)
      READ(11,*) NUYZ(2)
      READ(11,*) ZGXY(2)
      CLOSE(11)
C *****
C THE POSSIBILITY TO CHANGE THE STRESS ALLOWABLES AND CHARACTERISTIC LENGTH
C *****
      NTCH = 0
80    WRITE(*,325) SIGN, SIGS, AON, AOS
      CALL YORN(IANS,4)
      IF (IANS.EQ.2) GO TO 90
      NTCH = 10
      WRITE(*,330)
      READ(*,*) SIGN, SIGS, AON, AOS
      GO TO 80
90    CONTINUE
C *****
C WRITES OUT THE PROPERTIES IN A FILE TO BE SAVED.
C *****
      IF (ISTS.EQ.1. AND .NTCH.EQ.0) GO TO 95
      WRITE(*,350)
      CALL YORN(IANS,4)
      IF (IANS.EQ.2) GO TO 95
      WRITE(*,355)
      READ(*,'(A10)') FNM
      ESTR = '.STR'
      KHD = 10
      DO 97 K = 1, 9
      KK = 10 - K
97    IF (FNM(KK:KK).EQ.' ') KHD = KK
      SFILE = '
      SFILE(1:20) = 'C:/WEBSTER/STR-DATA/'
      SFILE(21:20+KHD) = FNM(1:KHD-1)
      SFILE(21+KHD:24+KHD) = ESTR
      OPEN(12,FILE=SFILE,STATUS='NEW')
      WRITE(12,360)
      WRITE(12,365) SIGN

```

```

WRITE(12,370) SIGS
WRITE(12,375) AON
WRITE(12,380) AOS
WRITE(12,385) H1
WRITE(12,390) ZEX(1)
WRITE(12,395) ZEY(1)
WRITE(12,405) ZEZ(1)
WRITE(12,410) NUXZ(1)
WRITE(12,415) NUXY(1)
WRITE(12,420) NUYZ(1)
WRITE(12,430) ZGXY(1)
WRITE(12,440) H2
WRITE(12,390) ZEX(2)
WRITE(12,395) ZEY(2)
WRITE(12,405) ZEZ(2)
WRITE(12,410) NUXZ(2)
WRITE(12,415) NUXY(2)
WRITE(12,420) NUYZ(2)
WRITE(12,430) ZGXY(2)
WRITE(12,*) '
CLOSE(12)
WRITE(*,445) FNM
95 CONTINUE
C *****
C ALL NECESSARY FORMAT STATEMENTS
C *****
200 FORMAT(/,8X,'WELCOME TO THE PROGRAM THAT SETS UP THE ',/,
      8X,'LAMINATE PROPERTIES FOR THE SKIN AND STIFFENER.',/,
      8X,'THIS PART OF THE PROGRAM MAY BE BY-PASSED IF THE ',/,
      8X,'DESIRED PROPERTIES ARE ALREADY STORED IN A FILE',/,
      8X,'FROM PREVIOUS MAKING. (IT MUST BE CORRECT, WITH',/,
      8X,'ONLY CHANGES IN MATERIAL ALLOWABLES).',/,
      8X,' WILL YOU BE USING SUCH A FILE?',/,
      8X,' (IF NOT, YOU WILL BE PROMPTED TO MAKE ONE. (Y OR N)')
205 FORMAT(/,5X,'BELOW ARE A LISTING OF MATERIALS THAT HAVE ALREADY',
      /,5X,'DEFINED. YOU NOW HAVE THE OPTION TO ADD TO THE LIST IF',
      /,5X,'YOU DESIRE OR IF DIFFERENT MATERIAL VALUES ARE NEEDED.',/
      /,5X,'DO YOU WANT TO ADD TO THE LIST? (INPUT Y OR N)',/,
      3X,'ID NO.',24X,'ID NAME',14X,'EX',12X,'EY',/)
206 FORMAT(3X,'AVAILABLE MATERIALS ARE:',/,3X,'ID NO.',24X,'ID NAME',
      /,14X,'EX',12X,'EY')
210 FORMAT(3X,I3,4X,A35,3X,E12.4,3X,E12.4)
215 FORMAT(/,5X,'FOR EACH MATERIAL SYSTEM ADDED, THE FOLLOWING INFO',
      /,5X,'MUST BE SUPPLIED: ',/,8X,
      '1) AN IDENTIFICATION NAME ',/,8X,
      '2) YOUNGS MODULUS (FIBER AND TRANSVERSE)--EX AND EY',/,8X,
      '3) SHEAR MODULUS, POISSON RATIO, AND THICKNESS--GXY,VXY,T',/,5X,
      ' NOW ENTER THE NUMBER OF NEW MATERIALS TO BE ADDED TO THE LIST')
220 FORMAT(/,5X,'FOR MATERIAL ',I4,' INPUT AN IDENTIFYING NAME.',/)
222 FORMAT(/,5X,'FOR THE MATERIAL INPUT: ',/,8X,
      'FIBER STIFFNESS, TRANSVERSE STIFFNESS, SHEAR MODULUS, ',/,8X,
      'POISSONS RATIO, AND LAMINA THICKNESS.',/)
225 FORMAT(/,3X,'INPUT THE FRACTION OF IN-PLANE PROPERTIES',/,3X,
      'YOU WISH TO HAVE TO OUT-OF-PLANE PROPERTIES IN THE SKIN.')
230 FORMAT(' *****',
      /, ' THIS IS A FILE THAT DESCRIBES A PARTICULAR MATERIAL USED ',/,
      ' IN THE ELASTICITY PROGRAM. THESE FILES MAY BE DELETED OR ',/,
      ' CHANGED HERE EXTERNALLY, BUT NOT IN THE PROGRAM ITSELF',/,
      ' *****')
235 FORMAT(3X,E19.12,3X,' FIBER STRENGTH... EX ')
240 FORMAT(3X,E19.12,3X,' TRANSVERSE STRENGTH ... EY ')
245 FORMAT(3X,E19.12,3X,' IN-PLANE SHEAR STRENGTH... GXY ')
250 FORMAT(3X,E19.12,3X,' POISSONS RATIO ... NUXY ')
255 FORMAT(3X,E19.12,3X,' LAMINA THICKNESS ... T ')
260 FORMAT(/,3X,'INPUT OUT-OF-PLANE YOUNGS MODULUS (Ez)')
265 FORMAT(/,3X,'THE NEXT INPUT IS FOR THE STIFFENER.',/,

```

```

3X,'A STIFFENER LAMINATE IS FORMED BASED ON THE FOLLOWING.',/)
270 FORMAT(/,3X,'THE X-DIRECTIONAL STIFFNESS IS ',F16.5,/,3X,
      'THE Y-DIRECTIONAL STIFFNESS IS ',F16.5,/,3X,
      'THE INPLANE SHEAR MODULUS IS ',F16.5,/,3X,
      'THE POISSONS RATIO IS ',F12.5,/,3X,
      'THE STIFFNERS THICKNESS IS ',F12.5,/)
275 FORMAT(/,3X,'DO YOU WANT ANY CHANGES TO THE STIFFENER LAMINATE?')
280 FORMAT(/,3X,'THE NEXT INPUT IS FOR THE SKIN.',/,
      3X,'A SKIN LAMINATE IS FORMED BASED ON THE FOLLOWING.',/)
290 FORMAT(/,3X,'FOR THE SKIN, INPUT OUT-OF-PLANE PROPERTIES:',
      /,3X,'YOUNGS MODULUS (Ez) ')
300 FORMAT(/,3X,'THE X-DIRECTIONAL STIFFNESS IS ',F16.5,/,3X,
      'THE Y-DIRECTIONAL STIFFNESS IS ',F16.5,/,3X,
      'THE INPLANE SHEAR MODULUS IS ',F16.5,/,3X,
      'THE POISSONS RATIO IS ',F12.5,/,3X,
      'THE SKIN THICKNESS IS ',F12.5,/)
305 FORMAT(/,3X,'DO YOU WANT CHANGES TO THE SKIN LAMINATE?(Y OR N)')
306 FORMAT(/,3X,'INPUT AVERAGE LENGTH FOR NORMAL STRESS',/,3X,
      'CONCENTRATION INTEGRATION, SHEAR STRESS CONCENTRATION LENGTH',
      /,3X,'AS WELL AS THE ULTIMATE INTERLAMINAR NORMAL AND SHEAR',/,
      3X,'STRESS VALUES.',/)
310 FORMAT(/,8X,'INPUT THE NAME OF THE FILE THAT ALREADY',/
      8X,'EXISTS THAT YOU WANT TO USING. (8 LETTER OR LESS)',/,
      8X,'THE FOLLOWING ARE A LIST AVAILABLE IN THIS DIR.',/,
      8X,'THE EXTENSION IS ALWAYS DEFAULTED TO ".STR".',/)
315 FORMAT(/,5X,'THE FOLLOWING IS A LIST OF FILES THAT MAY BE USED',
      /,5X,'INPUT THE ID NO. THAT CORRESPONDS TO THE FILE ',
      /,5X,'DESIRED. (MULTIPLE NAMES MEAN MORE THEN ONE',/,
      5X,'VERSION OF THAT NAME EXISTS. VERSION NUMBERS',/,
      5X,'ARE LISTED IN DESCENDING ORDER.)',/,/,
      9X,'ID NO.',12X,'FILE NAME',/)
320 FORMAT(10X,I3,16X,A10)
325 FORMAT(/,8X,'THE PARAMETERS THAT MAY BE CHANGED AT',/,
      8X,'THIS TIME IS: THE INTERLAMINAR NORMAL AND SHEAR',/,
      8X,'STRESS ALLOWABLES AND THE CHARACTERISTIC LENGTHS',/,
      8X,'ASSOCIATED WITH AVERAGE NORMAL AND SHEAR STRESSES.',/,/,
      8X,'THE CURRENT VALUES FOR THESE PARAMETERS ARE:',/,/,
      8X,'NORMAL STRESS ALLOWABLE = ',F15.5,/,
      8X,'SHEAR STRESS ALLOWABLE = ',F15.5,/,
      8X,'AVERAGE NORMAL STRESS CHARACTERISTIC LENGTH = ',F10.6,/,
      8X,'AVERAGE SHEAR STRESS CHARACTERISTIC LENGTH = ',F10.6,/,/,
      8X,'DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y OR N)?')
330 FORMAT(8X,'INPUT THE VAUES DESIRED FOR:',/,
      8X,'NORMAL AND SHEAR STRESS ALLOWABLE AT THE INTERFACE',/,
      8X,'AVERAGE NORMAL AND SHEAR STRESS CHARACTERISTIC LENGTH',/)
350 FORMAT(/,8X,'THE INFO. FROM THIS INPUT MAY BE STORED.',/,
      8X,'THIS IS STRONGLY RECOMMENDED UNLESS YOU HAVE JUST',/,
      8X,'RETRIEVED THE INFORMATION AND DO NOT WANT TO RE-WRITE',/,
      8X,'THE FILE. WOULD YOU LIKE TO STORE THIS INFO.? (Y OR N)')
355 FORMAT(/,8X,'INPUT THE NAME OF THE FILE YOU WOULD LIKE THE ',/,
      8X,'DATA STORED UNDER. THE FILENAME MUST BE 8 LETTERS OR',/,
      8X,'LESS. THE EXTENSION ".STR" IS GIVEN TO ALL DATA FILES.',/,
      8X,'IF A PREVIOUS FILE ALREADY EXISTS WITH THIS NAME, THE',/,
      8X,'FILE WILL BE OVER-WRITTEN. (THE OLD ONE WILL BE LOST)',/)
360 FORMAT(' *****',/,
      ' THIS IS A DATA FILE USED FOR THE ELASTICITY PROGRAM.',/,
      ' IT IS FORMED IN SUBROUTINE - STRMKR -',/,
      ' IT MAY BE ALTERED WITHOUT USING THE COMPUTER PROGRAM',/,/,
      ' NOTE: ALL DATA IS FOLLOWED BY A DESCRIPTION.',/,
      ' IF THERE IS ANY UNCERTAINTY ABOUT CHANGES, USE THE',
      ' SUBROUTINE ',/,
      ' *****',/)
365 FORMAT(2X,F15.5,9X,'REAL: NORMAL STRESS ALLOWABLE ')
370 FORMAT(2X,F15.5,9X,'REAL: SHEAR STRESS ALLOWABLE ')
375 FORMAT(6X,F15.9,5X,'REAL: AVERAGE NORMAL STRESS CHAR. LENGTH')
380 FORMAT(6X,F15.9,5X,'REAL: AVERAGE SHEAR STRESS CHAR. LENGTH',/)

```

```

385     FORMAT(6X,F15.9,5X,'REAL: STIFFENER THICKNESS (TOTAL)')
390     FORMAT(2X,F15.5,9X,'REAL: EX (FIBER STIFFNESS MODULUS)')
395     FORMAT(2X,F15.5,9X,'REAL: EY (TRANSVERSE STIFFNESS MODULUS)')
405     FORMAT(2X,F15.5,9X,'REAL: EZ (OUT-OF-PLANE STIFFNESS MODULUS)')
410     FORMAT(6X,F15.9,5X,'REAL: POISON RATIO-XZ (FIBER/OUT-OF-PLANE)')
415     FORMAT(6X,F15.9,5X,'REAL: POISON RATIO-XY (INPLANE)')
420     FORMAT(6X,F15.9,5X,'REAL: POISON RATIO-ZY (TRANS/OUT-OF-PLANE)')
430     FORMAT(2X,F15.5,9X,'REAL: SHEAR MODULUS-XY (INPLANE)')
440     FORMAT(/,6X,F15.9,5X,'REAL: SKIN THICKNESS (TOTAL)')
445     FORMAT(/,8X,'THE DATA HAS BEEN SAVED UNDER ',A8,'.STR.',/)
        RETURN
        END
C *****
C     THE NEXT TWO SUBROUTINES (DIR1 AND DIRM) ARE EXCLUSIVELY
C     FOR THE IBM-PC. THE VAX VERSIONS FOLLOW & DO THE SAME THING.
C     THERE ARE MACHINE DEPENDENT CODE IN THESE TWO SUBROUTINES.
C
C     Copyright (C) KLS, 1988.
C
C Abstract:
C     Outputs the directory in 5 column format. The calling variable
C     'filein' has the conventional format: D:\path\filename.ext.
C     Wildcards "*" & "?" are permitted. 'filein' is an ASCII string.
C *****
C *****
        SUBROUTINE DIR1 (FLNT,N1)
            CHARACTER      FLNT(40)*10, FILOUT(5)*13, DD*28
            INTEGER*2      IERROR, ITYPE
            DD='C:\WEBSTER\STR-DATA\*.STR'
            ITYPE = 0
            CALL FILDIR(DD,ITYPE,FILOUT(1),IERROR)
            IF (IERROR.NE.0) RETURN
            N = 1
            N1 = 0
20          N = N+1
            ITYPE = ITYPE + 1
            CALL FILDIR(DD,ITYPE,FILOUT(N),IERROR)
            IF (N.EQ.5 .AND. IERROR.EQ.0) THEN
                DO 33 II = 1, 5
                    N1 = 1 + N1
                    FLNT(N1) = '
            DO 32 K = 1, 13
32          IF (FILOUT(II) (K:K).EQ.'.') FLNT(N1) = FILOUT(II) (1:K-1)
33          CONTINUE
            N=0
            END IF
30          FORMAT (1X,5(A12,4X))
            IF (IERROR.EQ.0) GOTO 20
            DO 43 II = 1, N-1
                N1 = 1 + N1
                FLNT(N1) = '
            DO 42 K = 1, 13
42          IF (FILOUT(II) (K:K).EQ.'.') FLNT(N1) = FILOUT(II) (1:K-1)
43          CONTINUE
            RETURN
            END
CCCC *****
        SUBROUTINE DIRM (FLNT,N1)
            CHARACTER      FLNT(40)*10, FILOUT(5)*13, DD*28
            INTEGER*2      IERROR, ITYPE
            DD='
            DD='C:\WEBSTER\MAT-DATA\*.MAT'
            ITYPE = 0
            IERROR = 0
            FILOUT(1) = '
            CALL FILDIR(DD,ITYPE,FILOUT(1),IERROR)

```

```

                IF (IERROR.NE.0) RETURN
                N = 1
                N1 = 0
20                N = N+1
                    ITYPE = ITYPE + 1
                    FILOUT(N) = '
CALL FILDIR(DD,ITYPE,FILOUT(N),IERROR)
                    IF (N.EQ.5 .AND.IERROR.EQ.0) THEN
DO 33      II = 1, 5
                    N1 = 1 + N1
                    write(*,*)' n1 = ',n1
                    FLNT(N1) = '
DO 32      K = 1, 13
32                IF (FILOUT(II)(K:K).EQ.'.') FLNT(N1) = FILOUT(II)(1:K-1)
33                CONTINUE
                    N=0
                    END IF
30                FORMAT (1X,5(A12,4X))
                    IF (IERROR.EQ.0) GOTO 20
DO 43      II = 1, N-1
                N1 = 1 + N1
                    FLNT(N1) = "
DO 42      K = 1, 13
42                IF (FILOUT(II)(K:K).EQ.'.') FLNT(N1) = FILOUT(II)(1:K-1)
43                CONTINUE
                    RETURN
                    END
CCCC *****
SUBROUTINE DIRM2(FLNT,N1)
CHARACTER FLNT(40)*10, FILOUT(5)*13, DD*28
INTEGER*2      IERROR, ITYPE
DD="C:\WEBSTER\MAT-DATA\*.*"
ITYPE = 0
IERROR = 0
FILOUT(1) = '
CALL FILDIR(DD,ITYPE,FILOUT(1),IERROR)
IF (IERROR.NE.0) RETURN
N = 1
N1 = 0
20                N = N+1
                    ITYPE = ITYPE + 1
                    FILOUT(N) = '
CALL FILDIR(DD,ITYPE,FILOUT(N),IERROR)
IF (N.EQ.5 .AND.IERROR.EQ.0) THEN
DO 33      II = 1, 5
                    N1 = 1 + N1
                    FLNT(N1) = '
DO 32      K = 1, 13
32                IF (FILOUT(II)(K:K).EQ.'.') FLNT(N1) = FILOUT(II)(1:K-1)
33                CONTINUE
                    N=0
                    END IF
30                FORMAT (1X,5(A12,4X))
                    IF (IERROR.EQ.0) GOTO 20
DO 43      II = 1, N-1
                    N1 = 1 + N1
                    FLNT(N1) = '
DO 42      K = 1, 13
42                IF (FILOUT(II)(K:K).EQ.'.') FLNT(N1) = FILOUT(II)(1:K-1)
43                CONTINUE
                    RETURN
                    END
C *****
C One of 4 programs, to be linked with: WEB-1,WEB-2,(WEB-3),WEB-4,WEB.lib
C Last Revision: April 1990 (I/O up-dated only)
C *****

```



```

      SUBROUTINE ERROR
C
C      ERROR MESSAGES
C
      WRITE(*,10)
10     FORMAT(//,10X,'YOU HAVE ENTERED AN ILLEGAL RESPONSE',/
           10X,'PLEASE RE-ENTER YOU ANSWER. ')
      END
      SUBROUTINE YORN(IANS,N)
      INTEGER IANS, N
      CHARACTER*1 CH(13)
      CH(2) = 'Y'
      CH(3) = 'y'
      CH(4) = 'N'
      CH(5) = 'n'
      CH(6) = 'A'
      CH(7) = 'a'
      CH(8) = 'D'
      CH(9) = 'd'
      CH(10) = 'H'
10     READ(*,50,ERR=30)CH(1)
      DO 20 I=1,N
      IF(CH(I+1).EQ.CH(1))GO TO 40
20     CONTINUE
30     CALL ERROR
      GO TO 10
40     IANS=(I+1)/2
      RETURN
50     FORMAT(A1)
      END
      SUBROUTINE YORNDN(IANS)
      INTEGER IANS
      CHARACTER*1 CH(7)
      CH(2) = 'Y'
      CH(3) = 'y'
      CH(4) = 'N'
      CH(5) = ' '
10     READ(*,50,ERR=30)CH(1)
      DO 20 I=1,5
      IF(CH(I+1).EQ.CH(1))GO TO 40
30     I = 4
40     IANS=(I+1)/2
      RETURN
50     FORMAT(A1)
      END
      SUBROUTINE ZERO (A,NM,AMAX)
      DIMENSION A(1)
      FMAX = .0001
      DO 10 I = 1,NM
      IF (ABS(A(I)) .GT. FMAX) FMAX = ABS(A(I))
10     CONTINUE
      DO 20 I = 1,NM
      XX = A(I) / FMAX
      IF (ABS(XX) .LT. 0.0005 .OR. (FMAX/AMAX) .LT. .0005) A(I) = 0.0
20     CONTINUE
      RETURN
      END
      SUBROUTINE QBMAT (Q,QBAR,THETA)
      DIMENSION Q(4),QBAR(6)
C THIS SUBROUTINE CALCULATES QBAR MATRIX
      T=THETA*3.14159265/180.
      S=SIN(T)
      C=COS(T)
      C4=C**4
      S4=S**4
      CS22=C*C*S*S

```

```

      SC3=S*C**3
      CS3=C*S**3
      A1=2.*(Q(2)+2.*Q(4))
      A2=Q(1)+Q(3)-4.*Q(4)
      A3=Q(1)+Q(3)-2.*(Q(2)+Q(4))
      A4=Q(1)-Q(2)-2.*Q(4)
      A5=Q(2)-Q(3)+2.*Q(4)
      QBAR(1)=Q(1)*C4+A1*CS22+Q(3)*S4
      QBAR(2)=A2*CS22+Q(2)*(C4+S4)
      QBAR(3)=Q(1)*S4+A1*CS22+Q(3)*C4
      QBAR(4)=A4*SC3+A5*CS3
      QBAR(5)=A4*CS3+A5*SC3
      QBAR(6)=A3*CS22+Q(4)*(C4+S4)
      RETURN
    END
    SUBROUTINE QMAT (E1,E2,ENU12,G12,Q,I)
      DIMENSION Q(4,1)
    C THIS SUBROUTINE CALCULATES Q MATRIX
      ENU21=ENU12*E2/E1
      DENOM=1.-ENU12*ENU21
      Q(1,I)=E1/DENOM
      Q(2,I)=ENU21*E1/DENOM
      Q(3,I)=E2/DENOM
      Q(4,I)=G12
      RETURN
    END
    SUBROUTINE MAT
      COMMON/PASS/AM(6), BM(6), DM(6), THICK(50), THETA(50),
        ENT(3), EMT(3), ALPHX(3)
      COMMON/STIFF/MKIND(50), Q(4,50), QBAR(6), TT,NLAYER,EX,EY,GXY,ENUXY
      COMMON /MATT/ NUI2, E1, E2, G12, TMTAT
      REAL NUI2(50), E1(50), E2(50), G12(50), TMTAT(50)
      INTEGER NLAYER
    CC
    10  WRITE(*,440)
      READ(*,*,ERR=10) ISYMM
    210  WRITE(*,580)
      READ(*,*,ERR=230) NLAYER
      IF (NLAYER .LT. 0 .AND. NLAYER .GT.200) GO TO 230
      IF (ISYMM .NE. 1) GO TO 270
      INTT = NLAYER / 2
      FLOAT = NLAYER / 2. - INTT
      ISET = 0
      IF (FLOAT .GT. 0.02) ISET = 1
      NLAYER = INTT + ISET
      GO TO 270
    230  CALL ERROR
      GO TO 210
    260  WRITE(*,590)
      WRITE(*,600) (J,MKIND(J),J=1,NLAYER)
      WRITE(*,610)
      CALL YORN(IANS,4)
      IF (IANS .EQ. 2) GO TO 290
      CALL PLYCHG(NLAYER,MKIND,THICK,2)
      GO TO 260
    270  CONTINUE
      WRITE(*,620) NLAYER
      READ(*,*,ERR=270) (MKIND(I),I=1,NLAYER)
      GO TO 260
    280  WRITE(*,630)
      WRITE(*,640) (I,THICK(I),I=1,NLAYER)
      WRITE(*,650)
      CALL YORN(IANS,4)
      IF (IANS .EQ. 2) GO TO 320
      CALL PLYCHG(NLAYER,MKIND,THICK,1)
      GO TO 280

```

```

290 DO 300 I = 1,NLAYER
300 THICK(I) = TMAT(MKIND(I))
    GO TO 280
310 WRITE(*,500 )
    WRITE(*,510 )(I,THETA(I),I=1,NLAYER)
    WRITE(*,660 )
    CALL YORN(IANS,4)
    IF (IANS.EQ. 2) GO TO 340
    CALL PLYCHG(NLAYER,MKIND,THETA,1)
    GO TO 310
320 CONTINUE
    WRITE(*,670 )NLAYER
    READ(*,*,ERR=320)(THETA(I),I=1,NLAYER)
    DO 330 J = 1,NLAYER
    IF (THETA(J) .GT. 360) THEN
    WRITE(*,680 )
    GO TO 320
    ENDIF
330 CONTINUE
    GO TO 310
340 IF (ISYMM .NE. 1) GO TO 360
    N1 = NLAYER + 1
    NLAYER = NLAYER + INTT
    DO 350 I=N1,NLAYER
    THICK(I)=THICK(NLAYER + 1 -I)
    MKIND(I)=MKIND(NLAYER + 1 -I)
350 THETA(I)=THETA(NLAYER + 1 -I)
360 CONTINUE
    CALL STIFFN
    N2=0
    N3=0
    N4=0
    DO 410 I=1,NLAYER
    J=MKIND(I)
    E1J=E1(J)*1.E-6
    E2J=E2(J)*1.E-6
    G12J=G12(J)*1.E-6
    ITH=THETA(I)
    IF (ITH.EQ.0 )N1=N1+1
    IF (ITH.EQ.45 )N2=N2+1
    IF (ITH.EQ.-45)N3=N3+1
    IF (ITH.EQ.90 )N4=N4+1
410 CONTINUE
    CALL ZERO(AM,6,AM(1))
    CALL ZERO(DM,6,DM(1))
    CALL ZERO(BM,6,DM(1))
440 FORMAT(13X,'LAMINATE OPTIONS',/,13X,'===== ',/,10X,
. ' 1> SYMMETRICAL ',/,10X, ' 2> NON SYMMETRICAL ',/,10X,
. ' ===== ',/,10X, ' SELECT OPTION ')
500 FORMAT(5X,'PLY ORIENTATIONS ARE "(LAYER) ORIENTATION"',/,5X,
. '* (ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE) *')
510 FORMAT(2X,6(I3,' '),F6.2,1X)/
580 FORMAT(3X,' ENTER TOTAL NUMBER OF LAYERS IN YOUR LAMINATE ')
590 FORMAT(5X,'YOUR MATERIAL TYPES ARE AS FOLLOWS :',/,5X,'(LAYER) MAT
. ERIAL TYPE')
600 FORMAT(2X,6(I3,' '),I2,2X))
610 FORMAT(5X,'DO YOU WANT TO MODIFY MATERIAL TYPES (Y OR N) ')
620 FORMAT(5X,'INPUT MATERIAL TYPE FOR',I3,' LAYERS SEPARATED BY SPACE
. S')
630 FORMAT(5X,'YOUR THICKNESS FOR YOUR LAMINATES ARE :',/,5X,'(LAYER)
. THICKNESS')
640 FORMAT(2X,6(I3,' '),F6.4,1X))
650 FORMAT(5X,'DO YOU WANT TO MODIFY PLY THICKNESSES (Y OR N) ')
660 FORMAT(5X,'DO YOU WANT TO MODIFY PLY ORIENTATIONS (Y OR N) ')
670 FORMAT(5X,'INPUT ORIENTATION FOR',I3,' LAYERS SEPARATED BY A SPAC
. E',/,5X,'(ANGLE IS MEASURED FROM X-AXIS; GOING COUNTER CLOCKWISE)')

```

```

680 .FORMAT(10X,/, ' YOUR ORIENTATION MAY NOT EXCEED 360')
      END
      SUBROUTINE PLYCHG(NLAYER, NN, A, IG)
      DIMENSION NC(5), A(1), NN(1)
10    WRITE(*,110)
      READ(*,*,ERR=20) NP
      GO TO 30
20    CALL ERROR
      GO TO 10
30    NPP=5
      IF(NPP.GT.NLAYER) NPP=NLAYER
      IF(NP.GE.NPP) GO TO 80
40    WRITE(*,120)
      READ(*,*,ERR=50) (NC(I), I=1, NP)
      GO TO 60
50    CALL ERROR
      GO TO 40
60    IF(IG.EQ.1) THEN
      WRITE(*,130) (NC(I), I=1, NP)
      READ(*,*,ERR=70) (A(NC(I)), I=1, NP)
      ELSE
      WRITE(*,130) (NC(I), I=1, NP)
      READ(*,*,ERR=70) (NN(NC(I)), I=1, NP)
      ENDIF
      GO TO 90
70    CALL ERROR
      GO TO 60
80    IF(IG.EQ.1) THEN
      WRITE(*,140) NLAYER
      READ(*,*,ERR=100) (A(I), I=1, NLAYER)
      ELSE
      WRITE(*,140) NLAYER
      READ(*,*,ERR=100) (NN(I), I=1, NLAYER)
      ENDIF
90    RETURN
100   CALL ERROR
      GO TO 80
110  .FORMAT(5X, '*** ENTER NO. OF PLIES YOU WANT TO CHANGE ***')
120  .FORMAT(5X, 'ENTER THE PLY NUMBERS THAT YOU WANT TO CHANGE')
130  .FORMAT(5X, 'ENTER NEW VALUES FOR PLIES', 3X, 5(I3, 2X))
140  .FORMAT(5X, 'INPUT ALL NEW VALUES FOR ', I4, 2X, 'PLIES')
      END
      SUBROUTINE STIFFN
      COMMON/PASS/AM(6), BM(6), DM(6), THICK(50), THETA(50),
      ENT(3), EMT(3), ALPHX(3)
      COMMON/STIFF/MKIND(50), Q(4, 50), QBAR(6), TT, NLAYER, EX, EY, GXY, ENUXY
      DIMENSION AMMM(6), DUM(9), DUM1(9)
      DO 10 I=1, 6
      AM(I)=0.
      BM(I)=0.
10    DM(I)=0.
      DO 20 I=1, 3
      ENT(I)=0
20    EMT(I)=0
      TT=0.
      DO 30 I=1, NLAYER
30    TT=TT+THICK(I)
      HK1=-TT/2.
      DO 70 I=1, NLAYER
      KK=MKIND(I)
      CALL QBMAT(Q(1, KK), QBAR, THETA(I))
      HK2 = HK1 + THICK(I)
      HPA=HK2-HK1
      HPB=(HK2**2-HK1**2)/2.
      HPD=(HK2**3-HK1**3)/3.
      DO 60 J=1, 6

```

```

      AM(J)=AM(J)+QBAR(J)*HPA
      BM(J)=BM(J)+QBAR(J)*HPB
60     DM(J)=DM(J)+QBAR(J)*HPD
70     HK1=HK2
      CALL LINV(AM,AMMM,IER)
      EX=1/TT/AMMM(1)
      EY=1/TT/AMMM(3)
      GXY=1/TT/AMMM(6)
      ENUXY=-AMMM(2)/AMMM(1)
      CALL VMULSS(AMMM,BM,3,DUM,3)
      CALL VMULSF(AMMM,3,ENT,1,3,ALPHX,3)
      CALL VMULSF(BM,3,DUM,3,3,DUM1,3)
      CALL VCVTFS(DUM1,3,3,DUM)
      DO 80 I=1,6
      DUM1(I)=DM(I)-DUM(I)
80     IF (IER.NE.0) DUM1(I)=DM(I)
      DO 90 I=1,6
90     AMMM(I)=AM(I)/TT
      CALL ZERO (AMMM,6,AMMM(1))
      RETURN
      END
      SUBROUTINE LINV(A,B,IER)
      DIMENSION A(*), B(*)
      DEL = A(1)*A(3)*A(6) + 2.0*A(2)*A(4)*A(5) - A(1)*A(5)**2 -
            A(3)*A(4)**2 - A(6)*A(2)**2
      IF (DEL.LE..0001) GO TO 10
      B(1) = ( A(3)*A(6) - A(5)**2 ) / DEL
      B(2) = ( A(4)*A(5) - A(2)*A(6) ) / DEL
      B(3) = ( A(1)*A(6) - A(4)**2 ) / DEL
      B(4) = ( A(5)*A(2) - A(3)*A(4) ) / DEL
      B(5) = ( A(2)*A(4) - A(1)*A(5) ) / DEL
      B(6) = ( A(1)*A(3) - A(2)**2 ) / DEL
      IER = 0
      RETURN
10     IER = 1
      RETURN
      END
C *****
      SUBROUTINE VCVTFS(A, N, M, B)
      DIMENSION A(N,1), B(1)
      DO 10 I = 1, M
      DO 10 J = 1, M
      IF (I.GT.J) GO TO 10
      L = J*(J-1)/2+I
10     B(L) = A(I,J)
      CONTINUE
      RETURN
      END
C *****
      SUBROUTINE VMULSF(A,N1,B,N2,M,C,L)
      DIMENSION A(1), B(M,1), C(L,1)
      DO 10 I= 1, N1
      DO 10 J= 1, N2
      C(I,J) = 0.0
      DO 10 K= 1, N1
      IF (I.LE.K) N = (K-1)*K/2+I
      IF (I.GT.K) N = (I-1)*I/2+K
10     C(I,J) = C(I,J) + A(N) * B(K,J)
      RETURN
      END
C *****
      SUBROUTINE VMULSS(A, B, N, C, M)
      DIMENSION A(1), B(1), C(M,1)
      DO 10 I = 1, N
      DO 10 J = 1, N
      C(I,J) = 0.

```

```

DO 10 K = 1, N
  IF (K.LE.J) N1 = (J-1)*J/2+K
  IF (K.GT.J) N1 = (K-1)*K/2+J
  IF (I.LE.K) N2 = (K-1)*K/2+I
  IF (I.GT.K) N2 = (I-1)*I/2+K
10  C(I,J) = C(I,J) + B(N1)*A(N2)
  RETURN
END
CC *****
CC The rest of the subroutines are part of a IMSL Library that
CC were used to solve a system of non-linear equatd
CC can by pass a load condition where a local minimum is
CC reached instead of the actual minimum.
CC *****
C
C IMSL ROUTINE NAME - ZZANLY
C
C An iterative routine to find various solutions to a given function.
C The function here is defined in Subroutine FN1.
C -----
C
C SUBROUTINE ZZANLY (EPS,NSIG,KN,NGUESS,N,X,ITMAX,INFER,IER)
C
C EPS - FIRST CRITERIA INVOLVING STEP SIZE (NOT DESIRED FOR TRUE RESULT)
C NSIG - STEPS OF ROOTS AGREE TO THAT MANY SIGNIFICANT DIGITS
C KN - NUMBER OF KNOWN ROOTS
C NGUESS - NUMBER OF GUESSED ROOTS
C N - NUMBER OF ROOTS TO BE FOUND
C X - ARRAY HAVING ROOTS FOUND (THE UNKNOWN)S
C ITMAX - MAXIMUM NUMBER OF ITERATIONS TO USE TO FIND ONE ROOT
C INFER - ARRAY CONTAINING HOW MANY ITERATIONS EACH ROOT TOOK
C IER -ERROR CODE
C
C DOUBLE COMPLEX X(60), BI, D, DD, DEN, FPRT, FRT, H, RT,RTM1,
C T1, T2, T3, TEM, X0, XL, XX, Y0, Y1, Y2,
C Z0, Z1, Z2, FN1
C DOUBLE PRECISION QZ, TPQ
C INTEGER INFER(60)
C REAL RZERO,AX,EPS1,EPS
C COMPLEX ZERO,P1,ONE,FOUR,P5
C
C START OF GROUP A
C
C DENA = 1.00E-36
C ZERO = CMPLX(0.0,0.0)
C P1 = CMPLX(0.1,0.0)
C ONE = CMPLX(1.0,0.0)
C FOUR = CMPLX(4.0,0.0)
C P5 = CMPLX(0.5,0.0)
C ZNR = 0.0
C ZNR1 = 0.0
C RZERO = 0.000
C ITXX = ITMAX
C THUND = 175.000
C AX = 0.0100
C RP01 = 0.0100
C ICKMAX = 3
C IER = 0
C IF (N .LT. 1) GO TO 9005
C EPS1 = (10.00)**(-NSIG)
C EPS1 = AMIN1(EPS1,RP01)
C SET NUMBER OF ITERATIONS
C KNP1 = KN+1
C KNPN = KN+N

```

```

      KNPNG = KN+NGUESS
      DO 5 I=1,KNPN
        INFER(I) = 0
        IF (I .GT. KNPNG) X(I) = ZERO
5     CONTINUE
      L= KNP1
C
C     GROUP A INITIALIZES VALUES, AND IS NEVER REPEATED IN PROGRAM
C
C     START OF GROUP B
C
10    JK = 0
      ICK = 0
      IF (ITMAX.NE.ITXX) ITMAX = ITXX
      IF ( L.GT.KNPNG ) X(L) = ZERO
      XL = X(L)
15    IC = 0
      IF (ZNR.NE.ZNR1) XL = DCMPLX(ZNR,ZNR/5.0)
      ZNR1 = ZNR
      H = AX
      H = P1*H
      IF (ABS(XL) .GT. AX) H = P1*XL
C
C                                     FIRST THREE POINTS ARE
C                                     XL+H, XL-H, XL
C
      RT = XL+H
      ASSIGN 20 TO NN
      GO TO 50
C
C     GROUP B,C - DEFINES FIRST X VALUE ( X(I-2) = 0.01)
C     - H = STEP SIZE (dx)
C     - RT = X(L) + dx = VALUE OF X USED IN ITERATION
C
C     START GROUP D
C
20    Z0 = FPRT
      Y0 = FRT
      X0 = RT
      RT = XL-H
      ASSIGN 25 TO NN
      GO TO 50
C
C     GROUP D - DEFINES F( X(I-2) ) AND X(I-1)
C     - Z0 = FPRT( X(I-2) ) -- SOLUTION WITH REDUCTION
C     - Y0 = FRT( X(I-2) ) -- EXACT SOLUTION
C     - RT = X(I-1) = -0.01 -- SECOND VALUE OF X
C
C     START GROUP E
C
25    Z1 = FPRT
      Y1 = FRT
      H = XL-RT
      D = H/(RT-X0)
      RT = XL
      ASSIGN 30 TO NN
      GO TO 50
C
C     GROUP E - DEFINES F( X(I-1) ) AND X(I)
C     - Z1 = FPRT( X(I-1) ) -- SOLUTION WITH REDUCTION
C     - Y1 = FRT( X(I-1) ) -- EXACT SOLUTION
C     - RT = X(I) = 0.00 -- THIRD VALUE OF X
C
C     START GROUP F
C
30    Z2 = FPRT

```

```

      Y2 = FRT
C
C      GROUP F - DEFINES F( X(I) )
C      - Z2 = FPRT( X(I) ) -- SOLUTION WITH REDUCTION
C      - Y2 = FRT( X(I) ) -- EXACT SOLUTION
C
C      START GROUP G --- BEGIN MAIN ALGORITHM
C
35 DD = ONE + D
   T1 = Z0*D*D
   T2 = Z1*DD*DD
   XX = Z2*DD
   T3 = Z2*D
   BI = T1-T2+XX+T3
   IF (ABS(BI) .LT. DENA) DEN = (0.0E0,0.0E0)
   IF (ABS(BI) .LT. DENA) GO TO 111
   DEN = BI*BI - FOUR*(XX*T1-T3*(T2-XX))
C
C      GROUP G - MUELLER FUNCTION PARAMETERS DEFINED
C      - DD = 1 + # (# = LAMDA IN PAPERS ON SOLUTION METHOD)
C      - T1 = ###*F(i-2)
C      - T2 = (1 + #)*(1 + #)*F(i-1)
C      - T3 = #*F(i)
C      - XX = (1 + #)*F(i)
C      - BI = B (or g)
C      - DEN = B^2 - 4AC
C
C      START GROUP H -- USE DENOMINATOR OF MAXIMUM AMPLITUDE
C
111 T1 = SQRT(DEN)
    QZ = 100.00*AMAX1(ABS(BI),ABS(T1))
    T2 = BI + T1
    TPQ = ABS(T2)+QZ
    IF (TPQ .EQ. QZ) T2 = ZERO
    T3 = BI - T1
    TPQ = ABS(T3) + QZ
    IF (TPQ .EQ. QZ) T3 = ZERO
    DEN = T2
    QZ = ABS(T3)-ABS(T2)
    IF (QZ .GT. RZERO) DEN = T3
C
C      TEST FOR ZERO DENOMINATOR
C
    ASSIGN 30 TO NN
    IF (ABS(DEN) .EQ. RZERO) GO TO 65
C
C      GROUP H - DETERMINE MAXIMUM OF [g + sqrt] OR [g - sqrt]
C      - DEN = MAX OF [g + sqrt] OR [g - sqrt]
C
C      START GROUP I -- DETERMINE STEP SIZE AND NEW X VALUE
C
    D = -XX/DEN
    D = D+D
    H = D*H
    RT = RT + H
C
C      SPECIAL INPUT .... IF THE X VALUE IS LESS THEN -9.0
C      THEN THE INITIAL VALUE IS INCREASED
C      AND THE PROGRAM RETURNS TO THE FIRST STEP
C      (CHECK PRETTY THOROUGHLY BUT IN SOME CASES COULD PRESENT ERRORS)
C      THIS WAS INPUTTED SPECIFICALLY FOR WEBSTER/PACSTER SINCE THE REAL PART
C      OF THE LAMDA (EIGENVALUE) TO BE USED MUST BE GREATER THEN -1.0.
C
    IF (DREAL(RT) .LT. -9.0) ZNR = ZNR + 0.750
    IF (DREAL(RT) .LT. -9.0) GO TO 15
C

```



```

C      GROUP I - PARAMETERS IMPORTANT FOR STEP SIZE
C      - H = STEP SIZE
C      - RT = NEW VALUE FOR X
C
C      START GROUP I' -- CHECK CONVERGENCE OF THE FIRST KIND
C
C      IF (ABS(H) .LE. EPS1*AMAX1(ABS(RT),AX)) GO TO 70
C      IF (IC .NE. 0) GO TO 15
C      ASSIGN 40 TO NN
C      GO TO 50
C
C      GROUP I' - IF STEP SIZE IS TOO SMALL, ALTER COURSE (GO TO GROUP XY)
C      - THIS IS CHECK MUST FAIL TWICE FOR PROGRAM TO IDENTIFY
C      CONVERGENCE --- DO NOT DESIRE THIS TYPE OF CONVERGENCE.
C
C      START GROUP J' -- DETERMINE IF F(X) IS INCREASING ARE DECREASING
C
C      40 QZ = ABS(FPRT)-ABS(Z2)*10.00
C      IF (QZ .GE. RZERO) GO TO 45
C
C      GROUP J' - IF F(X+1) > F(X) GO TO GROUP K
C      - IF F(X+1) < F(X) GO TO GROUP J
C
C      GROUP J - IF THE FUNCTION IS DECREASING (POSITIVE TREND/RESULTS)
C
C      Z0 = Z1
C      Z1 = Z2
C      Z2 = FPRT
C      Y0 = Y1
C      Y1 = Y2
C      Y2 = FRT
C      GO TO 35
C
C      GROUP J - CHANGE POINT NUMBERS TO PREPARE FOR NEXT STEP
C      - F(X(i-2)) = F(X(i-1))
C      - F(X(i-1)) = F(X(i))
C      - F(X(i)) = F(X(i+1))
C
C      GROUP K - F(X(i+1)) IS INCREASING .... ALTER STEP SIZE AND X(i+1)
C
C      45 CONTINUE
C      D = D*P5
C      H = H*P5
C      RT = RT-H
C
C      GROUP K - TAKE REMEDIAL ACTION TO INDUCE CONVERGENCE
C      - H = 0.5 * H (=dX)
C      - X(i+1) = X(i+1) - dX/2.0 = RT-H
C
C      START GROUP X
C
C      50 JK = JK+1
C      IF (ABS(RT).GT.THUND) ITMAX = JK-1
C      IF (ABS(RT).GT.THUND) RT = RTM1
C      IF (JK .GT. ITMAX) GO TO 75
C      FRT = FN1(RT)
C      FPRT = FRT
C
C      GROUP X - FOR NEW VALUE OF X [RT], FIND F(X) [FRT]
C
C      START GROUP XX - CHECK TO SEE IF FIRST ROOT IS BEING FOUND
C
C      IF (L .EQ. 1) GO TO 60
C      LM1 = L-1
C      DO 55 I=1,LM1
C      TEM = RT - X(I)

```

```

        IF (ABS(TEM) .LT. DENA) GO TO 65
        IF (ABS(FPRT) .LT. DENA) GO TO 65
55      FPRT = FPRT/TEM
60      CONTINUE

C
C      GROUP XX - COMPUTE DEFLATED FUNCTION      [FPRT]
C
C      START GROUP X' - CONVERGENCE OF SECOND KIND
C
        IF (ABS(FPRT) .LE. EPS .AND. ABS(FRT) .LE. EPS) GO TO 80
        GO TO NN, (20,25,30,40)
65      CONTINUE

C
C      GROUP X' - CHECK TO SEE IF F(X) IS LESS THEN TOLD EPSILON
C      - IF [FPRT] AND [FRT] < [EPS] .... GO TO GROUP Z
C      - IF [FPRT] AND [FRT] > [EPS] .... GO TO GROUP G
C
C      START GROUP XY
C
        IF (IC .NE. 0) GO TO 15
        TEM = 10.00*EPS1
        RTM1 = RT
        IF (ABS(RT) .GT. AX) TEM = TEM*RT
        RT = RT+TEM
        D = (H+TEM)*D/H
        H = H+TEM
        GO TO 50

C
C      GROUP XY - DETERMINANT USED IN STEP SIZE IS ALMOST ZERO
C      - DEFINE DIFFERENT STEP SIZE FOR NEW X [RT]
C
C      START GROUP Y - CHECK SOLUTION FOR TOO SMALL OF STEP SIZE
C
70      CONTINUE
        IF (IC .NE. 0) GO TO 80
        IC = 1
        Z0 = Y1
        Z1 = Y2
        Z2 = FN1(RT)
        XL = RT
        ICK = ICK+1
        IF (ICK .LE. ICKMAX) GO TO 35

C
C      GROUP Y - CHECK FOR TOO SMALL OF STEP SIZE OCCURS TWICE ONLY
C      - AFTER ONE FAILURE
C      - THE STEP SIZE IS ADDED TO THE X VALUE ONCE MORE
C      TO SEE IF A GREATER STEP SIZE RESULTS FROM THE NEW
C      SOLUTION. ( F(X(i-2))=F(X(i-1)), F(X(i-1))=F(X(i)
C                  F(X(i)) = F(X(i+1)), X = RT )
C
C      - AFTER TWO FAILURE --- CONVERGENCE IS ASSUMED.
C      - NOT ALWAYS A VALID SOLUTION
C
C      START GROUP Z - CONVERGED SOLUTION (ROOT FOUND)
C
C                  WARNING ERROR, ITMAX = MAXIMUM

        JK = ITMAX + JK
75      IER = 33
80      X(L) = RT
        INFER(L) = JK
        L = L+1
        IF (L .LE. KNP) GO TO 10
        IF (IER .EQ. 0) GO TO 9005

C
C      GROUP Z - X(L) = RT
C      - INFER(L) = JK

```

```

C          - L = L + 1
9000 CONTINUE
C      CALL ZUERTS (IER,'ZZANLY')
9005 RETURN
      END
C *****
C THIS IS THE 4TH CODE FOR WEBSTER:WEB-1,WEB-2,WEB-3,(WEB-4) AND WEB.LIB
C *****
C      IMSL ROUTINE NAME - ZLEQT2
C
C Solves a system of equation where all variables are real. The
C system is in the form: AX = B
C
C-----
      SUBROUTINE ZLEQT2 (A,M,N,IA,B,IDGT,WKAREA,IER)
C
C      DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
C                               FIRST EXECUTABLE STATEMENT
C                               INITIALIZE IER
C
      IER=0
      JER=0
      J = N*N+1
      K = J+N
      MM = K+N
      KK = 0
      MM1 = MM-1
      JJ=1
      DO 5 L=1,N
        DO 5 I=1,N
          WKAREA(JJ)=A(I,L)
          JJ=JJ+1
5    CONTINUE
C      DECOMPOSE A
C      CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
C                WA,IER)
      IF (IER.GT.128) GO TO 25
      IF (IDGT.EQ. 0 .OR. IER.NE. 0) KK = 1
      DO 15 I = 1,M
C      PERFORMS THE ELIMINATION PART OF
C      AX = B
      CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))
C      REFINEMENT OF SOLUTION TO AX = B
      IF (KK.NE. 0)
        CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDGT,WKAREA(J),WKAREA(MM),
        WKAREA(K),WKAREA(K),JER)
      DO 10 II=1,N
        B(II,I) = WKAREA(MM1+II)
10    CONTINUE
      IF (JER.NE.0) GO TO 20
15    CONTINUE
      GO TO 25
20    IER = 131
25    JJ=1
      DO 30 J = 1,N
        DO 30 I = 1,N
          A(I,J)=WKAREA(JJ)
          JJ=JJ+1
30    CONTINUE
      IF (IER.EQ. 0) GO TO 9005
9000 CONTINUE
9005 CONTINUE
      RETURN
      END
C
C      IMSL ROUTINE NAME - ZUERTS
C-----

```

```

SUBROUTINE ZUERTS (IER,NAME)
  INTEGER          IEQDF, IOUNIT, LEVEL, LEVOLD, NIN
  CHARACTER*6      NAME (1), NAMSET (1)
  COMMON/CONTRL/ICNTL, NW
  DATA            NAMSET/'UERSSET'/

C
  LEVEL = 4
  IEQDF = 0

C
  CALL UGETIO (1,NIN, IOUNIT)
C
  CHECK IER

  IF (IER.GT.999) GO TO 25
  IF (IER.LT.-32) GO TO 65
  IF (IER.LE.128) GO TO 5
  IF (LEVEL.LT.1) GO TO 30

C
  PRINT TERMINAL MESSAGE
  IF (NW.EQ.1) WRITE (31,35) IER,NAME
  GO TO 30
5  IF (IER.LE.64) GO TO 10
  IF (LEVEL.LT.2) GO TO 30

C
  PRINT WARNING WITH FIX MESSAGE
  IF (NW.EQ.1) WRITE (31,40) IER,NAME
  GO TO 30
10 IF (IER.LE.32) GO TO 15

C
  PRINT WARNING MESSAGE
  IF (LEVEL.LT.3) GO TO 30
  IF (NW.EQ.1) WRITE (31,66) IER,NAME
  GO TO 30
15 CONTINUE

C
  CHECK FOR UERSSET CALL
C
  IF ( NAME (1).NE.NAMSET (1) ) GO TO 25
  LEVOLD = LEVEL
  LEVEL = IER
  IER = LEVOLD
  IF (LEVEL.LT.0) LEVEL = 4
  IF (LEVEL.GT.4) LEVEL = 4
  GO TO 30
25 CONTINUE
  IF (LEVEL.LT.4) GO TO 30

C
  PRINT NON-DEFINED MESSAGE
  IF (NW.EQ.1) WRITE (31,50) IER,NAME
30 IEQDF = 0
35 FORMAT (19H *** TERMINAL ERROR,10X,7H (IER = ,I3,
      20H) FROM IMSL ROUTINE , ' = ',A6)
40 FORMAT (27H *** WARNING WITH FIX ERROR,2X,7H (IER = ,I3,
      20H) FROM IMSL ROUTINE , ' = ',A6)
66 FORMAT (18H *** WARNING ERROR,11X,7H (IER = ,I3,
      20H) FROM IMSL ROUTINE , ' = ',A6)
50 FORMAT (20H *** UNDEFINED ERROR,9X,7H (IER = ,I5,
      20H) FROM IMSL ROUTINE , ' = ',A6)
65 RETURN
END

C
C  IMSL ROUTINE NAME - UGETIO
C-----
SUBROUTINE UGETIO (IOPT,NIN,NOUT)
C
C  SPECIFICATIONS FOR ARGUMENTS
  INTEGER          IOPT,NIN,NOUT,NIND,NOUTD
  DATA            NIND/5/,NOUTD/6/
C
C  FIRST EXECUTABLE STATEMENT
  IF (IOPT.EQ.3) GO TO 10
  IF (IOPT.EQ.2) GO TO 5
  IF (IOPT.NE.1) GO TO 9005
  NIN = NIND
  NOUT = NOUTD

```

```

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

C
C     IMSL ROUTINE NAME      - GGUBFS
C -----
      REAL FUNCTION GGUBFS (DSEED)
C                                     SPECIFICATIONS FOR ARGUMENTS
      REAL    DSEED
C                                     SPECIFICATIONS FOR LOCAL VARIABLES
      REAL    D2P31M,D2P31
C                                     D2P31M=(2**31) - 1
C                                     D2P31 = (2**31) (OR AN ADJUSTED VALUE)
      DATA    D2P31M/2147483647.0/
      DATA    D2P31 /2147483648.0/
C                                     FIRST EXECUTABLE STATEMENT
      DSEED = AMOD(16807.0*DSEED,D2P31M)
      GGUBFS = DSEED / D2P31
      RETURN
      END

C
C     IMSL ROUTINE NAME      - LUDATN
C -----
      SUBROUTINE LUDATN (A, IA, N, LU, ILU, IDGT, D1, D2, APVT, EQUIL, WA, IER)
C
      DIMENSION    A(IA,1), LU(ILU,1), APVT(1), EQUIL(1)
      REAL          LU
      DATA          ONE, FOUR, SIXTN, SIXTH/1., 4., 16., .0625/
C                                     FIRST EXECUTABLE STATEMENT
C                                     INITIALIZATION
      IER = 0
      ZERO = 0.0E0
      RN = N
      WREL = ZERO
      D1 = ONE
      D2 = ZERO
      BIGA = ZERO
      DO 10 I=1,N
        BIG = ZERO
        DO 5 J=1,N
          P = A(I,J)
          LU(I,J) = P
          P = ABS(P)
          IF (P .GT. BIG) BIG = P
5         CONTINUE
          IF (BIG .GT. BIGA) BIGA = BIG
          IF (BIG .EQ. ZERO) GO TO 110
          EQUIL(I) = ONE/BIG
10        CONTINUE
        DO 105 J=1,N
          JM1 = J-1
          IF (JM1 .LT. 1) GO TO 40
C                                     COMPUTE U(I,J), I=1,...,J-1
          DO 35 I=1,JM1
            SUM = LU(I,J)
            IM1 = I-1
            IF (IDGT .EQ. 0) GO TO 25
C                                     WITH ACCURACY TEST
            AI = ABS(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+ABS(T)
15      CONTINUE
      LU(I,J) = SUM
20      WI = WI+ABS(SUM)
      IF (AI .EQ. ZERO) AI = BIGA
      TEST = WI/AI
      IF (TEST .GT. WREL) WREL = TEST
      GO TO 35
C      WITHOUT ACCURACY
25      IF (IM1 .LT. 1) GO TO 35
      DO 30 K=1,IM1
          SUM = SUM-LU(I,K)*LU(K,J)
30      CONTINUE
      LU(I,J) = SUM
35      CONTINUE
40      P = ZERO
C      COMPUTE U(J,J) AND L(I,J), I=J+1,...
      DO 70 I=J,N
          SUM = LU(I,J)
          IF (IDGT .EQ. 0) GO TO 55
C      WITH ACCURACY TEST
          AI = ABS(SUM)
          WI = ZERO
          IF (JM1 .LT. 1) GO TO 50
          DO 31 K=1,JM1
              T = LU(I,K)*LU(K,J)
              SUM = SUM-T
              WI = WI+ABS(T)
31      CONTINUE
          LU(I,J) = SUM
50      WI = WI+ABS(SUM)
          IF (AI .EQ. ZERO) AI = BIGA
          TEST = WI/AI
          IF (TEST .GT. WREL) WREL = TEST
          GO TO 65
C      WITHOUT ACCURACY TEST
55      IF (JM1 .LT. 1) GO TO 65
          DO 60 K=1,JM1
              SUM = SUM-LU(I,K)*LU(K,J)
60      CONTINUE
          LU(I,J) = SUM
65      Q = EQUIL(I)*ABS(SUM)
          IF (P .GE. Q) GO TO 70
          P = Q
          IMAX = I
70      CONTINUE
C      TEST FOR ALGORITHMIC SINGULARITY
      IF (P.EQ. ZERO) 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      APVT(J) = IMAX
      D1 = D1*LU(J,J)
85      IF (ABS(D1) .LE. ONE) GO TO 90
      D1 = D1*SIXTH
      D2 = D2+FOUR
      GO TO 85
90      IF (ABS(D1) .GE. SIXTH) GO TO 95
      D1 = D1*SIXTN

```

```

          D2 = D2-FOUR
          GO TO 90
95         CONTINUE
          JP1 = J+1
          IF (JP1 .GT. N) GO TO 105
C          DIVIDE BY PIVOT ELEMENT U(J,J)
          P = LU(J,J)
          DO 100 I=JP1,N
            LU(I,J) = LU(I,J)/P
100        CONTINUE
105        CONTINUE
C          PERFORM ACCURACY TEST
          IF (IDGT .EQ. 0) GO TO 9005
          P = 3*N+3
          WA = P*WREL
          IF (WA+10.0*(-IDGT) .NE. WA) GO TO 9005
          IER = 34
          GO TO 9000
C          ALGORITHMIC SINGULARITY
110        IER = 129
          D1 = ZERO
          D2 = ZERO
9000       CONTINUE
C          PRINT ERROR
          CALL ZUERTS(IER,'LUDATN')
9005       RETURN
          END
C
C      IMSL ROUTINE NAME      - LUELMN
C      -----
C      SUBROUTINE LUELMN (A,IA,N,B,APVT,X)
C
C      DIMENSION              A(IA,1),B(1),APVT(1),X(1)
C                               FIRST EXECUTABLE STATEMENT
C                               SOLVE LY = B FOR Y
C
C      DO 5 I=1,N
5      X(I) = B(I)
      IW = 0
      DO 20 I=1,N
        IP = APVT(I)
        SUM = X(IP)
        X(IP) = X(I)
        IF (IW .EQ. 0) GO TO 15
        IM1 = I-1
        DO 10 J=IW,IM1
          SUM = SUM-A(I,J)*X(J)
10      CONTINUE
        GO TO 20
15      IF (SUM .NE. 0.) IW = I
20      X(I) = SUM
C          SOLVE UX = Y FOR X
      DO 30 IB=1,N
        I = N+1-IB
        IP1 = I+1
        SUM = X(I)
        IF (IP1 .GT. N) GO TO 30
        DO 25 J=IP1,N
          SUM = SUM-A(I,J)*X(J)
25      CONTINUE
30      X(I) = SUM/A(I,I)
      RETURN
      END
C
C      IMSL ROUTINE NAME      - LUREFN
C      -----
C      SUBROUTINE LUREFN (A,IA,N,UL,IUL,B,IDGT,APVT,X,RES,DX,IER)

```

```

C      DIMENSION      A(IA,1),UL(IUL,1),B(1),X(1),RES(1),DX(1)
      DIMENSION      APVT(1)
      REAL            SUM
      DATA           ITMAX/500/,ZERO/0.0/
C                                     FIRST EXECUTABLE STATEMENT
      IER=0
      XNORM = ZERO
      DO 10 I=1,N
          XNORM = AMAX1(XNORM,ABS(X(I)))
10      CONTINUE
      IF (XNORM .NE. ZERO) GO TO 20
      IDGT = 50
      GO TO 9005
20      DO 77 ITER=1,ITMAX
          DO 30 I=1,N
              SUM = B(I)
              DO 25 J=1,N
                  SUM = SUM - A(I,J) * X(J)
25          CONTINUE
              RES(I) = SUM
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 = AMAX1(DXNORM,ABS(DX(I)))
              XNORM = AMAX1(XNORM,ABS(X(I)))
35          CONTINUE
          IF (ITER .NE. 1) GO TO 40
          IDGT = 50
          IF (DXNORM .NE. ZERO) IDGT = -ALOG10(DXNORM/XNORM)
40          IF (XNORM+DXNORM .EQ. XNORM) GO TO 9005
          IF (IDGT.GT.5) GO TO 9005
77      CONTINUE
C                                     ITERATION DID NOT CONVERGE
      IER = 129
9000      CONTINUE
9005      RETURN
      END

C      IMSL ROUTINE NAME      - UERSET
C      -----
C      SUBROUTINE UERSET (LEVEL,LEVOLD)
C                                     SPECIFICATIONS FOR ARGUMENTS
      INTEGER              LEVEL,LEVOLD
C                                     FIRST EXECUTABLE STATEMENT
      LEVOLD = LEVEL
      CALL ZUERTS (LEVOLD,'UERSET')
      RETURN
      END

C      IMSL ROUTINE NAME      - ZLEQTC
C      -----
C      SUBROUTINE ZLEQTC (A,N,IA,B,M,IB,IJOB,WA,IER)
C                                     SPECIFICATIONS FOR ARGUMENTS
      INTEGER              N,IA,M,IB,IJOB,IER
      COMPLEX*16           A(IA,N),B(IB,M)
      DOUBLE PRECISION     WA(N)
C                                     SPECIFICATIONS FOR LOCAL VARIABLES
      DOUBLE PRECISION     P,Q,ZERO,ONE,T(2),RN,BIG
      COMPLEX*16           SUM,TEMP
      INTEGER              I,J,JM1,IM1,K,IMAX,JP1,IW,N1
      EQUIVALENCE          (SUM,T(1))
      DATA                ZERO/0.0D0/,ONE/1.0D0/

```



```

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 = ABS(TEMP)
        IF (P .GT. BIG) BIG = P
      5 CONTINUE
      IF (BIG .EQ. ZERO) GO TO 105
      WA(I) = ONE/BIG
    10 CONTINUE

C                               L-U DECOMPOSITION
    DO 70 J = 1,N
      JM1 = J-1
      IF (JM1 .LT. 1) GO TO 25

C                               COMPUTE U(I,J), I=1,...,J-1
      DO 20 I=1,JM1
        SUM = A(I,J)
        IM1 = I-1
        IF (IM1 .LT. 1) GO TO 20
        DO 15 K=1,IM1
          SUM = SUM-A(I,K)*A(K,J)
        15 CONTINUE
        A(I,J) = SUM
      20 CONTINUE
      P = ZERO

C                               COMPUTE U(J,J) AND L(I,J), I=J+1,...,
      DO 45 I=J,N
        SUM = A(I,J)
        IF (JM1 .LT. 1) GO TO 40
        DO 35 K=1,JM1
          SUM = SUM-A(I,K)*A(K,J)
        35 CONTINUE
        A(I,J) = SUM
      40 Q = WA(I)*ABS(SUM)
        IF (P .GE. Q) GO TO 45
        P = Q
        IMAX = I
      45 CONTINUE

C                               TEST FOR ALGORITHMIC SINGULARITY
      Q = RN+P
      IF (Q .EQ. RN) GO TO 105
      IF (J .EQ. IMAX) GO TO 60

C                               INTERCHANGE ROWS J AND IMAX
      DO 50 K=1,N
        TEMP = A(IMAX,K)
        A(IMAX,K) = A(J,K)
        A(J,K) = TEMP
      50 CONTINUE
      WA(IMAX) = WA(J)
      WA(J) = IMAX
      JP1 = J+1
      IF (JP1 .GT. N) GO TO 70

C                               DIVIDE BY PIVOT ELEMENT U(J,J)
      TEMP = A(J,J)
      DO 65 I = JP1,N
        A(I,J) = A(I,J)/TEMP
      65 CONTINUE
      70 CONTINUE
      75 IF (IJOB .EQ. 1) GO TO 9005
      DO 103 K = 1,M

```

```

C                                     SOLVE UX = Y FOR X
      IW = 0
      DO 90 I = 1,N
        IMAX = WA(I)
        SUM = B(IMAX,K)
        B(IMAX,K) = B(I,K)
        IF (IW .EQ. 0) GO TO 85
        IM1 = I-1
        DO 80 J = IW,IM1
          SUM = SUM-A(I,J)*B(J,K)
80      CONTINUE
        GO TO 88
85      IF (T(1) .NE. ZERO .OR. T(2) .NE. ZERO) IW = I
88      B(I,K) = SUM
90      CONTINUE

C                                     SOLVE LY = B FOR Y
      N1 = N+1
      DO 100 IW = 1,N
        I = N1-IW
        JP1 = I+1
        SUM = B(I,K)
        IF (JP1 .GT. N) GO TO 98
        DO 95 J = JP1,N
          SUM = SUM-A(I,J)*B(J,K)
95      CONTINUE
98      B(I,K) = SUM/A(I,I)
100     CONTINUE
103     CONTINUE
      GO TO 9005

C                                     ALGORITHMIC SINGULARITY
105     IER = 129
9000     CONTINUE

C                                     PRINT ERROR
      CALL ZUERTS(IER,'ZLEQTC')
9005     RETURN
      END

C  IMSL ROUTINE NAME      - ZLEQ2C
C-----
      SUBROUTINE ZLEQ2C (A,N,IA,B,M,IB,IJOB,WA,WK,IER)
C
      COMPLEX*16          A(IA,1),B(IB,1),WA(N,1),TEMPA,TEMPB,TEMPC
      DOUBLE PRECISION    WK(N),TA(2),TB(2),TC(2)
      DOUBLE PRECISION    AR,AI,BR,BI,CR,CI,DXNORM,XNORM,ZERO
      DOUBLE PRECISION    ACC(2)
      EQUIVALENCE          (TA(1),TEMPA),(TB(1),TEMPB),(TC(1),TEMPC),
      .                     (TA(1),AR),(TA(2),AI),(TB(1),BR),(TB(2),BI),
      .                     (TC(1),CR),(TC(2),CI)
      DATA                ZERO/0.0D0/
      DATA                ITMAX/650/

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 ZLEQTC (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 ZLEQTC(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,ABS(AR),ABS(AI))
25 CONTINUE
IF (XNORM .EQ. ZERO) GO TO 65
C      COMPUTE RESIDUALS
DO 50 ITER = 1,ITMAX
  DO 40 I = 1,N
    TEMPB = B(I,J)
    ACC(1) = 0.0D0
    ACC(2) = 0.0D0
    CALL VXADD(BR,ACC)
    DO 30 JJ = 1,N
      TEMPA = A(I,JJ)
      TEMPB = WA(JJ,N1)
      CALL VXMUL(-AR,BR,ACC)
      CALL VXMUL(AI,BI,ACC)
30 CONTINUE
    CALL VXSTO(ACC,CR)
    TEMPB = B(I,J)
    ACC(1) = 0.0D0
    ACC(2) = 0.0D0
    CALL VXADD(BI,ACC)
    DO 35 JJ = 1,N
      TEMPA = A(I,JJ)
      TEMPB = WA(JJ,N1)
      CALL VXMUL(-AR,BI,ACC)
      CALL VXMUL(-BR,AI,ACC)
35 CONTINUE
    CALL VXSTO(ACC,CI)
    WA(I,N2) = TEMPC
40 CONTINUE
  CALL ZLEQTC(WA,N,N,WA(1,N2),1,N,2,WK,IER)
  DXNORM = ZERO
C      UPDATE THE SOLUTION
DO 45 I = 1,N
  WA(I,N1) = WA(I,N1)+WA(I,N2)
  TEMPA = WA(I,N2)
  DXNORM = DMAX1(DXNORM,ABS(AR),ABS(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 ZUERTS(IER,'LEQ2C ')
9005 RETURN
END
C
C  IMSL ROUTINE NAME      - VXADD
C-----
C
SUBROUTINE VXADD(A,ACC)

```

```

C
C      DOUBLE PRECISION    A,ACC(2)      SPECIFICATIONS FOR ARGUMENTS
C      DOUBLE PRECISION    X,Y,Z,ZZ      SPECIFICATIONS FOR LOCAL VARI
C                                          FIRST EXECUTABLE STATEMENT
C      X = ACC(1)
C      Y = A
C      IF (ABS(ACC(1)).GE.ABS(A)) GO TO 1
C      X = A
C      Y = ACC(1)
C      1  Z = X+Y                      COMPUTE Z+ZZ = ACC(1)+A EXACT
C      ZZ = (X-Z)+Y
C                                          COMPUTE ZZ+ACC(2) USING DOUBL
C                                          PRECISION ARITHMETIC
C      ZZ = ZZ+ACC(2)
C                                          COMPUTE ACC(1)+ACC(2) = Z+ZZ
C      ACC(1) = Z+ZZ
C      ACC(2) = (Z-ACC(1))+ZZ
C      RETURN
C      END

C
C      IMSL ROUTINE NAME    - VXSTO
C-----
C      SUBROUTINE VXSTO (ACC,D)
C      DOUBLE PRECISION    ACC(2),D      SPECIFICATIONS FOR ARGUMENTS
C                                          FIRST EXECUTABLE STATEMENT
C      D = ACC(1)+ACC(2)
C      RETURN
C      END

C
C      IMSL ROUTINE NAME    - VXMUL
C-----
C      SUBROUTINE VXMUL (A,B,ACC)
C      DOUBLE PRECISION    A,B,ACC(2)    SPECIFICATIONS FOR ARGUMENTS
C      DOUBLE PRECISION    X,HA,TA,HB,TB  SPECIFICATIONS FOR LOCAL VARI
C      INTEGER              IX(2),I
C      LOGICAL*1             LX(8),LI(4)
C      EQUIVALENCE           (X,LX(1),IX(1)),(I,LI(1))
C      DATA                 I/0/
C                                          SPLIT A = HA+TA
C                                          B = HB+TB
C                                          FIRST EXECUTABLE STATEMENT
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                                          COMPUTE HA*HB,HA*TB,TA*HB, AN
C                                          AND CALL VXADD TO ACCUMULAT
C                                          SUM

```

```

      X = TA*TB
      CALL VXADD (X,ACC)
      X = HA*TB
      CALL VXADD (X,ACC)
      X = TA*HB
      CALL VXADD (X,ACC)
      X = HA*HB
      CALL VXADD (X,ACC)
      RETURN
      END
      SUBROUTINE UTRSRT (Z, ZZ, BOTLIM, TOPLIM, KEYPOS, TOTLEN)
C*****
C      UTRSRT SUBROUTINE          SORT A TABLE BASED ON REAL KEY
C                                  (DJN)
C      THIS ROUTINE USES SUBROUTINE COPY - ENTRY POINTS: COPY & SWAP
C
C                                  LASTMOD          04 SEPTEMBER 1985 (DJN)
C*****
C      THIS SORT ROUTINE IS MODELLED AFTER THE QUICKSORT ALGORITHM. IT
C      IS A INCORE SORT. A CUTOFF LIMIT IS SET TO ELIMINATE THE WASTE OF
C      SORTING SMALL SUBSETS USING QUICKSORT. THESE SUBSETS ARE SORTED
C      BY INSERT SORT.
C      ALSO THIS ROUTINE NEEDS EXTRA SPACE OF THE
C      LENGTH = 2*LN(NUMBER OF TUPLES)/LN(2)
C      CURRENTLY A STACK OF (2,40) IS HARD CODED WHICH GIVES AN UPPER
C      LIMIT OF 1 TRILLION ENTRIES IN THE SORT. IN ADDITION AN ARRAY
C      (ZZ) OF LENGTH TOTLEN IS NEEDED AS A TEMPORARY TUPLE
C      STORAGE LOCATION.
C      D. E. KNUTH, VOL. 3, PAGE 116
C
      INTEGER          BOTLIM,          TOPLIM,
      .                BOT,             TOP,
      .                CUTOFF,          TOTLEN,
      .                KEY,             Z(*),
      .                ZZ(*),          STACK(2,40),
      .                KVAL
C
      REAL             RKEY,             VAL
C
      EQUIVALENCE      (RKEY,           KEY),
      .                (VAL,            KVAL)
C
      DATA            CUTOFF/15/
C
      BOT      = BOTLIM + KEYPOS - 1
      TOP      = TOPLIM - TOTLEN + KEYPOS
      ISTACK = 0
C
      TOP OF LOOP (Q2)
C
10  IF ( (TOP - BOT + 1)/TOTLEN .LT. CUTOFF ) THEN
C
      INSERTION SORT (Q8)
C
      JSTOP = BOT + TOTLEN
      DO 20 I = JSTOP, TOP, TOTLEN
      KEY = Z(I)
      CALL COPY ( Z, I, ZZ, KEYPOS, KEYPOS, TOTLEN )
C
      INNER LOOP
C
      DO 30 J = I, JSTOP, -TOTLEN
      KVAL = Z(J-TOTLEN)
      IF ( VAL .LE. RKEY ) GO TO 25
      CALL COPY ( Z, J-TOTLEN, Z, J, KEYPOS, TOTLEN )

```

```

30      CONTINUE
25      CALL COPY ( ZZ, KEYPOS, Z, J, KEYPOS, TOTLEN )
C
20      CONTINUE
C
C      CHECK STACK COUNT (Q9)
C
      IF ( ISTACK .LE. 0 ) THEN
C
          RETURN
      ELSE
          BOT      = STACK(1, ISTACK)
          TOP      = STACK(2, ISTACK)
          ISTACK = ISTACK - 1
      END IF
C
C      QUICKSORT LOOP (Q2 CONT)
C
      ELSE
          I      = BOT
          J      = TOP
          NENTRY = (TOP - BOT + 1) / TOTLEN
          IGET   = BOT + IFIX( XXRAND() * FLOAT(NENTRY) ) * TOTLEN
          KEY    = Z(IGET)
          CALL COPY ( Z, IGET, ZZ, KEYPOS, KEYPOS, TOTLEN )
          CALL SWAP ( Z, IGET, Z, I, KEYPOS, TOTLEN )
C
C      FIND UPPER BOUND (Q3)
C
200      KVAL = Z(J)
          IF ( RKEY .LT. VAL ) THEN
              J = J - TOTLEN
              GO TO 100
          END IF
C
C      TRANSFER ENTRIES (Q4)
C
          IF ( I .LT. J ) THEN
              CALL SWAP ( Z, I, Z, J, KEYPOS, TOTLEN )
              I = I + TOTLEN
          ELSE
              CALL COPY ( ZZ, KEYPOS, Z, I, KEYPOS, TOTLEN )
              GO TO 1000
          END IF
C
C      FIND LOWER BOUND (Q5)
C
200      KVAL = Z(I)
          IF ( VAL .LT. RKEY ) THEN
              I = I + TOTLEN
              GO TO 200
          END IF
C
C      TRANSFER ENTRIES (Q6)
C
          IF ( I .LT. J ) THEN
              CALL SWAP ( Z, I, Z, J, KEYPOS, TOTLEN )
              J = J - TOTLEN
              GO TO 100
          ELSE
              CALL COPY ( ZZ, KEYPOS, Z, J, KEYPOS, TOTLEN )
              I = J
          END IF
C
C      SET STACK (Q7)
C

```

```

1000      ISTACK = ISTACK + 1
          IF ( TOP-I .GE. I-BOT ) THEN
              STACK(1, ISTACK) = I + TOTLEN
              STACK(2, ISTACK) = TOP
              TOP = I - TOTLEN
          ELSE
              STACK(1, ISTACK) = BOT
              STACK(2, ISTACK) = I - TOTLEN
              BOT = I + TOTLEN
          END IF
      END IF
      GO TO 10
  END
  SUBROUTINE COPY (Z, I, ZZ, J, KEYPOS, TOTLEN )
C*****
C          COPY SUBROUTINE          COPY ROUTINE FOR UTSORT.
C                                     CONTAINS TWO ROUTINES
C      1 - COPY  SIMPLE COPY FROM TO
C      2 - SWAP  SIMPLE SWAP
C                                     (KLE)
C                                     LASTMOD      29 JULY, 1986 (DJN)
C*****
C      THIS ROUTINE IS COPY ROUTINE FOR THE TABLE SORTING ROUTINE.
C      INTEGER          TOTLEN,          KEYPOS,          TEMP, Z(*), ZZ(*)
C
C      IDIFF = I - J
C      KSTART = J - KEYPOS + 1
C      KSTOP = KSTART - 1 + TOTLEN
C
C      DO 10 K = KSTART, KSTOP, 1
C          L = IDIFF + K
C          ZZ(K) = Z(L)
10  CONTINUE
C
C      RETURN
C*****
C      ENTRY SWAP (Z, I, ZZ, J, KEYPOS, TOTLEN )
C
C      IDIFF = I - J
C      KSTART = J - KEYPOS + 1
C      KSTOP = KSTART - 1 + TOTLEN
C
C      DO 20 K = KSTART, KSTOP, 1
C          L = IDIFF + K
C          TEMP = ZZ(K)
C          ZZ(K) = Z (L)
C          Z (L) = TEMP
20  CONTINUE
C
C      RETURN
C      END
C
C      REAL FUNCTION XXRAND ( )
C*****
C      A SYSTEM UTILITY TO RETURN A REAL VALUE BETWEEN 0.0 AND 1.0
C*****
C      REAL FUNCTION XXRAND ( )
C      DATA          IY /1753/
C      IY          = IY * 659

```

```

      IF (IY .LT. 0) IY = IY + 32767
      YFL = IY
      XXRAND = YFL * 4.656613E-10
      IF (XXRAND.GT.1.OR.XXRAND.LT.0) THEN
        IJ = XXRAND
        XXRAND = XXRAND - IJ
        XXRAND = (XXRAND+1.0) / 2.5
      END IF
      RETURN
    END
  CC *****
  CC Quadratic function used for Stress function along the bottom surface
  CC *****
      REAL FUNCTION NY3(A3,B3,C3,Z)
      DOUBLE PRECISION Z
      NY3 = A3*Z*Z + B3*Z + C3
      RETURN
    END
  CC *****
  CC Sets up the T(2) matrix. A 4X4 matrix that will be inverted.
  CC *****
      SUBROUTINE FNP(LAMS)
      COMPLEX*16 MU, CZERO, F1(4,4), T2, J1, GFL, NL, LAM, L1, L2, ONE
      COMPLEX LAMS
      DOUBLE PRECISION BETA, Z1, Z2
      COMMON/CP1/MU(2,4)
      COMMON/LMAT/ARL(8,8),BIM(8,8)
      COMMON/RL1/ANGLE
      COMMON/RL2/BETA(2,6,6)
      COMMON/TRANS/T2(4,4)
      ONE = CMPLX(1.0D0,0.0D0)
      CZERO = CMPLX(0.0D0,0.0D0)
      Z1 = DBLE(REAL(LAMS))
      Z2 = DBLE(AIMAG(LAMS))
      LAM = DCMPLX(Z1,Z2)
      API = 3.15032
      L1 = LAM + ONE
      L2 = L1 + ONE
      DO 10 I1 = 1, 4
        J1 = SIN(API) * MU(2,I1)
        J1 = J1 + COS(API)
        J1 = J1 ** L2
        F1(1,I1) = GFL(API,L2,2,I1)
        F1(2,I1) = NL(API,2,I1) * GFL(API,L1,2,I1)
        F1(3,I1) = CZERO
        F1(4,I1) = CZERO
      DO 5 I2 = 1, 4
        F1(3,I1) = F1(3,I1) + GFL(ANGLE,L2,1,I2)*T2(I1,I2)
        F1(4,I1) = F1(4,I1)+NL(ANGLE,1,I2)*GFL(ANGLE,L1,1,I2)*T2(I1,I2)
5      CONTINUE
10      DO 20 I3 = 1, 4
        DO 20 I4 = 1, 4
        ARL(I3,I4) = REAL( F1(I3,I4) )
        BIM(I3,I4) = SNGL( DIMAG( F1(I3,I4)) )
20      RETURN
      END
  CC *****
  CC This subroutine is used to find the determinant of the multiplied
  CC MATRIX T(1) INVERSE * T(2). THIS IS A 4X4 MATRIX
  CC *****
      DOUBLE COMPLEX FUNCTION FN1(LAM)
      COMPLEX*16 MU, CZERO, FA, FB, FC, FD, FN1A, FN1B, FN1C,
        FN1D, T2, F1(4,4), FF, FE, LAM, L1, L2, GFL, NL, ONE
      DOUBLE PRECISION BETA
      COMMON/CP1/MU(2,4)
      COMMON/RL1/ANGLE

```



```

COMMON/RL2/BETA(2,6,6)
COMMON/TRANS/T2(4,4)
ONE = CMPLX(1.0,0.0D0)
CZERO = CMPLX(0.0D0,0.0D0)
FN1 = CZERO
ZZMX = 1.0E20
L1 = LAM + ONE
L2 = L1 + ONE
DO 10 I1 = 1, 4
  F1(3,I1) = CZERO
  F1(4,I1) = CZERO
DO 5 I2 = 1, 4
  F1(3,I1) = F1(3,I1) + GFL(ANGLE,L2,1,I2)*T2(I1,I2)
  F1(4,I1) = F1(4,I1)+NL(ANGLE,1,I2)*GFL(ANGLE,L1,1,I2)*T2(I1,I2)
5 CONTINUE
10 CONTINUE
FA = ( F1(3,3)*F1(4,4) - F1(4,3)*F1(3,4) )
FB = ( F1(3,1)*F1(4,4) - F1(4,1)*F1(3,4) )
FC = ( F1(3,1)*F1(4,3) - F1(4,1)*F1(3,3) )
FD = ( F1(3,2)*F1(4,4) - F1(4,2)*F1(3,4) )
FE = ( F1(3,2)*F1(4,3) - F1(4,2)*F1(3,3) )
FF = ( F1(3,1)*F1(4,2) - F1(4,1)*F1(3,2) )
FN1A = ( ((-ONE)**LAM) * MU(2,2) * ( FA - FB + FC ) )
FN1B = ( ((-ONE)**LAM) * MU(2,1) * ( FA - FD + FE ) )
FN1C = ( ((-ONE)**LAM) * MU(2,3) * ( FD - FB + FF ) )
FN1D = ( ((-ONE)**LAM) * MU(2,4) * ( FE - FC + FF ) )
FN1 = ((-ONE)**LAM) * ( FN1A - FN1B - FN1C + FN1D )
11 CONTINUE
RETURN
END
CC *****
CC Sets up the G(i,j) expression. See notes for true definition.
CC *****
DOUBLE COMPLEX FUNCTION GFL(THETA,LAA,I,J)
COMPLEX*16 MU, LAA
COMMON/CP1/MU(2,4)
GFL = COS(THETA) + MU(I,J) * SIN(THETA)
GFL = GFL ** LAA
RETURN
END
CC *****
CC Sets up n(i,j) expression. Similar to G(i,j).
CC See notes for definition.
CC *****
DOUBLE COMPLEX FUNCTION NL(THETA,I,J)
COMPLEX*16 MU
COMMON/CP1/MU(2,4)
NL = SIN(THETA) - MU(I,J) * COS(THETA)
RETURN
END
CC *****
CC Sets up P(i,j) expression. A function of Mu and Beta.
CC *****
COMPLEX FUNCTION PL(EPS,II,JJ)
COMPLEX*16 MU, Z1, Z2
DOUBLE PRECISION BETA
COMMON/CP1/MU(2,4)
COMMON/RL1/ANGLE
COMMON/RL2/BETA(2,6,6)
Z1 = MU(II,JJ) * MU(II,JJ) * BETA(II,1,1) + BETA(II,1,2)
Z2 = MU(II,JJ) * BETA(II,1,2) + BETA(II,2,2) / MU(II,JJ)
PL = Z1 * COS(EPS) + Z2 * SIN(EPS)
PL = PL / BETA(1,1,1)
RETURN
END
CC *****

```

```

CC  Sets-up similar function as PL. See notes for more information.
CC  *****
      COMPLEX      FUNCTION FFL(EPS,II,JJ)
      COMPLEX*16 MU, Z1, Z2
      DOUBLE PRECISION BETA
      COMMON/CP1/MU(2,4)
      COMMON/RL1/ANGLE
      COMMON/RL2/BETA(2,6,6)
      Z1 = MU(II,JJ) * MU(II,JJ) * BETA(II,1,1) + BETA(II,1,2)
      Z2 = MU(II,JJ) * BETA(II,1,2) + BETA(II,2,2) / MU(II,JJ)
      FFL = Z2 * COS(EPS) - Z1 * SIN(EPS)
      FFL = FFL / BETA(1,1,1)
      RETURN
      END
C  *****
C  Subroutine used to actually calculate the stress at a point given
C  all the necessary data.
C  *****
      SUBROUTINE STRS1 (COT, LMD, NLM, XP, YP, SX1, SY1, SKY1)
      COMPLEX*16 COT(25), MU, ZTL1, ZTL2, ZTL3, ZTL4,
      AL, BL
      COMPLEX LMD(25)
      DOUBLE PRECISION SKY1, SX1, SY1, TXY, TX, TY
      COMMON/CP1/MU(2,4)
      COMMON/STRS/AL(4,25), BL(4,25)
      ZERO = 0.0E0
      SX1 = ZERO
      SY1 = ZERO
      SKY1 = ZERO
      TX = ZERO
      TY = ZERO
      TXY = ZERO
      ISKP = 0
      DO 10 I = 1, NLM
        IF (YP.LT.ZERO) GO TO 5
        ZTL1 = ( XP + MU(1,1) * YP ) ** LMD(I)
        ZTL2 = ( XP + MU(1,2) * YP ) ** LMD(I)
        ZTL3 = ( XP + MU(1,3) * YP ) ** LMD(I)
        ZTL4 = ( XP + MU(1,4) * YP ) ** LMD(I)
        SX1 = SX1 + REAL( COT(I) * ( AL(1,I)*MU(1,1)*MU(1,1)*ZTL1 +
        AL(2,I)*MU(1,2)*MU(1,2)*ZTL2 + AL(3,I)*MU(1,3)*MU(1,3)*ZTL3 +
        AL(4,I)*MU(1,4)*MU(1,4)*ZTL4 ) )
        SY1 = SY1 + REAL( COT(I) * ( AL(1,I)*ZTL1 + AL(2,I)*ZTL2 +
        AL(3,I)*ZTL3 + AL(4,I)*ZTL4 ) )
        SKY1 = SKY1 - REAL( COT(I) * ( AL(1,I) * MU(1,1)*ZTL1 +
        AL(2,I) * MU(1,2) * ZTL2 +
        AL(3,I)*MU(1,3)*ZTL3 + AL(4,I)*MU(1,4)*ZTL4 ) )
        TX = TX + REAL( ( AL(1,I)*MU(1,1)*MU(1,1)*ZTL1 +
        AL(2,I)*MU(1,2)*MU(1,2)*ZTL2 + AL(3,I)*MU(1,3)*MU(1,3)*ZTL3 +
        AL(4,I)*MU(1,4)*MU(1,4)*ZTL4 ) )
        TY = TY + REAL( ( AL(1,I)*ZTL1 + AL(2,I)*ZTL2 +
        AL(3,I)*ZTL3 + AL(4,I)*ZTL4 ) )
        TXY = TXY - REAL( ( AL(1,I) * MU(1,1)*ZTL1 +
        AL(2,I) * MU(1,2) * ZTL2 +
        AL(3,I)*MU(1,3)*ZTL3 + AL(4,I)*MU(1,4)*ZTL4 ) )
        GO TO 10
      5
        ZTL1 = ( XP + MU(2,1) * YP ) ** LMD(I)
        ZTL2 = ( XP + MU(2,2) * YP ) ** LMD(I)
        ZTL3 = ( XP + MU(2,3) * YP ) ** LMD(I)
        ZTL4 = ( XP + MU(2,4) * YP ) ** LMD(I)
        SX1 = SX1 + REAL(COT(I) * (BL(1,I)*MU(2,1)*MU(2,1)*ZTL1 + ZTL2*
        BL(2,I)*MU(2,2)*MU(2,2) + BL(3,I)*MU(2,3)*MU(2,3)*ZTL3 +
        BL(4,I)*MU(2,4)*MU(2,4)*ZTL4 ) )
        SY1 = SY1 + REAL( COT(I) * ( BL(1,I)*ZTL1 + BL(2,I)*ZTL2 +
        BL(3,I)*ZTL3 + BL(4,I)*ZTL4 ) )

```

```

      SKY1= SKY1- REAL (COT (I) * (BL (1,I)*MU (2,1)*ZTL1 + BL (2,I)*
      MU (2,2)*ZTL2 + BL (3,I)*MU (2,3)*ZTL3 + BL (4,I)*MU (2,4)*ZTL4))
      TX = TX + REAL ((BL (1,I)*MU (2,1)*MU (2,1)*ZTL1 + ZTL2*
      BL (2,I)*MU (2,2)*MU (2,2) + BL (3,I)*MU (2,3)*MU (2,3)*ZTL3 +
      BL (4,I)*MU (2,4)*MU (2,4)*ZTL4 ) )
      TY = TY + REAL (( BL (1,I)*ZTL1 + BL (2,I)*ZTL2 +
      BL (3,I)*ZTL3 + BL (4,I)*ZTL4 ) )
      TXY = TXY - REAL ((BL (1,I)*MU (2,1)*ZTL1 + BL (2,I)*
      MU (2,2)*ZTL2 + BL (3,I)*MU (2,3)*ZTL3 + BL (4,I)*MU (2,4)*ZTL4))
10  CONTINUE
      RETURN
      END

```

4. Rayleigh-Ritz Method (BEAM)

```

C *****
C
C      This is the first of a set of three programs to solve the Out-
C      of-Plane loading analysis using a Rayleigh-Ritz series
C      approximation. The following program follows the derivation
C      that can be seen in the accompanying papers.
C
C      This program was written by Mary Mahler (Northrop Aircraft Div.)
C      and Dr. Han-Pin Kan (Northrop Aircraft Div.) as part of a contract
C      with McDonnell Douglas - Aircraft as the primary contractors.
C      Starting Nov. 1987 to Jan. 1988
C
C *****
C      REAL*4      SE1(25), SE2(25), ET1(25), EI1(25), ET2(25), EI2(25),
C      *          ZI(12,5,5), XX(2), ZII(4,5,5,5), ZIII(6,5,5,5,5), A(2)
C *****
C
C      These are the commands for the data input used for the program
C
C *****
C      WRITE(*,500)
C      WRITE(*,510)
C      READ(*,*) XL, D
C      WRITE(*,520)
C      READ(*,*) NSL, NSD
C      SE1(1) = 0.0
C      WRITE(*,540)
C      DO 10 I = 2, NSL+1
10  READ(*,*) SE1(I), ET1(I), EI1(I)
C      WRITE(*,550)
C      SE2(1) = 0.0
C      DO 20 I = 2, NSD+1
20  READ(*,*) SE2(I), ET2(I), EI2(I)
C
C      WRITE(*,555)
C      READ(*,*) NPMX
C      WRITE(*,560)
C      READ(*,*) EX, GXZ, T, SIGU, TAUU, AA0
C      NPST = 1
C      IS = 4
C      DDS = D * 2.0
C      NX = 2
C      XX(1) = XL
C      XX(2) = D
C      KMAX = 100
C *****
C
C      FINDING THE I'S ( ZI, ZII, ZIII) THAT ARE INTEGRALS IN THE RAYLEIGH-RITZ
C      SOLUTION --- THE FIRST LOOP FINDS THE I OF A PARTICULAR SUBSCRIPT.

```

```

C   THE SECOND RUNS THE LENGTH OF THE BEAM (OR DOUBLER), AND THE OTHERS
C   ARE INTERNAL SUMMATIONS.
C *****
C   XLD = XL - D
C
C   DO 100  M = 1, IS
C
C   The next 16 lines define the P, Q, and R values that are defined
C   in the derivation used as parameters in the sine and cosine expressions
C   These values are defined for each internal summation variable.
C
C   PM = PZ(M,XL)
C   RM = PZ(M,XLD)
C   QM = PZ(M,DDS)
C
C   DO 100  I = 1, IS
C   PI = PZ(I,XL)
C   RI = PZ(I,XLD)
C   QI = PZ(I,DDS)
C   RMI = RM - RI
C   RRI = RM + RI
C   PMI = PM - PI
C   PPI = PM + PI
C   PMR = PM - RI
C   PPR = PM + RI
C   PMQ = PM - QI
C   PQI = PM + QI
C   QMI = QM - QI
C   QQI = QM + QI
C   ZI(5,M,I) = D - SX(PM,D,0.0) - SX(PI,D,0.0)
C   ZI(6,M,I) = D - SX(PM,D,0.0) - SX(QI,D,0.0)
C   ZI(8,M,I) = D - SX(QM,D,0.0) - SX(QI,D,0.0)
C
C   DO 70  LK = 2, NSL+1
C
C   The (70) Do Loop runs along the beam only. Thus the values defined
C   within it are those integrals from 0 to L or from D to L.
C
C   A(1) = SE1(LK - 1)
C   A(2) = SE1(LK)
C   T1 = ET1(LK)
C   T11 = E11(LK)
C
C   DO 70  N = 1,2
C   S = (2 * N) - 3
C
C   ZI(1,M,I) = ZI(1,M,I) + T11/2.0 * PM * PM * PI * PI * S *
C   * ( SX(PMI,A(N),0.0) + SX(PPI,A(N),0.0) )
C
C   ZI(2,M,I) = ZI(2,M,I) + T1/2.0 * PM * PI * S *
C   * ( SX(PMI,A(N),0.0) + SX(PPI,A(N),0.0) )
C
C   IF (A(2).LE.D) GOTO 50
C   IF ( A(1).LT.D ) A(1) = D
C
C   ZI(3,M,I)=ZI(3,M,I) + T11 * PM * PM * RI * RI / 2.0 * S *
C   * ( SX(PMR,A(N),-D*RI ) + SX(PPR,A(N),RI*D ) )
C
C   ZI(4,M,I) = ZI(4,M,I) + T11 * RM * RM * RI * RI / 2.0 * S *
C   * ( SX(RMI,A(N),D*RMI ) + SX(RRI,A(N),D*RRI ) )
C
50 DO 70  J = 1, IS
C   A(1) = SE1(LK - 1)
C   PJ = PZ(J,XL)
C   RJ = PZ(J,XLD)

```

```

      RIJ = RI - RJ
      RRJ = RI + RJ
C
      ZII(1,M,I,J) = ZII(1,M,I,J) + T1 * PM * PI * PJ / 4.0 * S *
*      ( SX(PMI-PJ,A(N),0.0) + SX(PMI+PJ,A(N),0.0) -
*      SX(PPI-PJ,A(N),0.0) - SX(PPI+PJ,A(N),0.0) )
C
      IF (SEL(LK).LE.D) GOTO 60
      IF ( A(1).LT.D ) A(1) = D
C
      ZII(2,M,I,J) = ZII(2,M,I,J) + T1 * PM * RI * RJ / 4.0 * S *
*      ( SX(RIJ - PM,A(N),D*RIJ) + SX(RIJ + PM,A(N),D*RIJ) -
*      SX(RRJ - PM,A(N),D*RRJ) - SX(RRJ + PM,A(N),D*RRJ) )
C
      ZII(3,M,I,J) = ZII(3,M,I,J) + T1 * PM * RI * PJ / 4.0 * S *
*      ( SX(PMR - PJ,A(N),-D*RI) + SX(PMR + PJ,A(N),-D*RI) -
*      SX(PPR - PJ,A(N), D*RI) - SX(PPR + PJ,A(N), D*RI) )
C
60      DO 70 K = 1,IS
          PK = PZ(K,XL)
          RK = PZ(K,XLD)
          RJK = RJ - RK
          PJK = PJ - PK
          RRK = RJ + RK
          PPK = PJ + PK
          A(1) = SEL(LK - 1)
C
          ZIII(1,M,I,J,K) = ZIII(1,M,I,J,K) + T1 * PM * PI * PJ * PK
*          / 8.0 * S * ( SX(PMI+PJK,A(N),0.0) - SX(PMI+PPK,A(N),0.0)
*          - SX(PMI-PPK,A(N),0.0) + SX(PMI-PJK,A(N),0.0)
*          - SX(PPI+PJK,A(N),0.0) + SX(PPI+PPK,A(N),0.0)
*          + SX(PPI-PPK,A(N),0.0) - SX(PPI-PJK,A(N),0.0) )
C
          IF (A(2).LE.D) GOTO 70
          IF (A(1).LT.D) A(1) = D
C
          ZIII(2,M,I,J,K) = ZIII(2,M,I,J,K) + T1 * PM * PI * RJ * RK / 8.0 *
*          S * ( SX(PMI+RJK,A(N),RJK*D) - SX(PMI+RRK,A(N),RRK*D)
*          - SX(PMI-RRK,A(N),-RRK*D) + SX(PMI-RJK,A(N),-RJK*D)
*          - SX(PPI+RJK,A(N),RJK*D) + SX(PPI+RRK,A(N),RRK*D)
*          + SX(PPI-RRK,A(N),-RRK*D) - SX(PPI-RJK,A(N),-RJK*D) )
C
          ZIII(3,M,I,J,K) = ZIII(3,M,I,J,K) + T1 * PM * RI * RJ * RK / 8.0
*          S * ( SX(PMR+RJK,A(N),-(RIJ+RK)*D) -
*          SX(PMR+RRK,A(N),-(RIJ-RK)*D) - SX(PMR-RRK,A(N),-(RRJ+RK)*D)
*          + SX(PMR-RJK,A(N),-(RRJ-RK)*D) -
*          SX(PPR+RJK,A(N),(RRJ-RK)*D) + SX(PPR+RRK,A(N),(RRJ+RK)*D)
*          + SX(PPR-RRK,A(N),(RIJ-RK)*D) - SX(PPR-RJK,A(N),(RIJ+RK)*D) )
C
          ZIII(4,M,I,J,K) = ZIII(4,M,I,J,K) + T1 * PM * PI * PJ * RK / 8.0
*          S * ( SX(PMI+PJ-RK,A(N),-RK*D) - SX(PMI+PJ+RK,A(N),RK*D)
*          - SX(PMI-PJ-RK,A(N),-RK*D) + SX(PMI-PJ+RK,A(N),RK*D) -
*          SX(PPI+PJ-RK,A(N),-RK*D) + SX(PPI+PJ+RK,A(N),RK*D)
*          + SX(PPI-PJ-RK,A(N),-RK*D) - SX(PPI-PJ+RK,A(N),RK*D) )
C
          ZIII(5,M,I,J,K) = ZIII(5,M,I,J,K) + T1 * RM * RI * RJ * RK / 8.0
*          S * ( SX(RMI+RJK,A(N),(RMI+RJK)*D) -
*          SX(RMI+RRK,A(N),(RMI+RRK)*D) - SX(RMI-RRK,A(N),(RMI-RRK)*D)
*          + SX(RMI-RJK,A(N),(RMI-RJK)*D) -
*          SX(RRI+RJK,A(N),(RRI+RJK)*D) + SX(RRI+RRK,A(N),(RRI+RRK)*D) +
*          SX(RRI-RRK,A(N),(RRI-RRK)*D) - SX(RRI-RJK,A(N),(RRI-RJK)*D) )
C
70      CONTINUE
C
      DO 90 LK = 2, NSD + 1
C

```

```

C      The (90) Do Loop runs along the Doubler only. The following
C      expressions are for those integrals that go from 0 to D.
C
C      A(1) = SE2(LK-1)
C      A(2) = SE2(LK)
C      T1 = ET2(LK)
C      T11 = EI2(LK)
C
C      DO 90 N = 1, 2
C      S = (2 * N) - 3
C
C      ZI(5,M,I) = ZI(5,M,I) + S / 2.0 *
C      *      ( SX(PMI,A(N),0.0) + SX(PPI,A(N),0.0) )
C
C      ZI(6,M,I) = ZI(6,M,I) + S / 2.0 *
C      *      ( SX(PMQ,A(N),0.0) + SX(PQI,A(N),0.0) )
C
C      ZI(7,M,I) = ZI(7,M,I) + T11 * QM * QM * QI * QI * S / 2.0 *
C      *      ( SX(QMI,A(N),0.0) + SX(QQI,A(N),0.0) )
C
C      ZI(8,M,I) = ZI(8,M,I) + S / 2.0 *
C      *      ( SX(QMI,A(N),0.0) + SX(QQI,A(N),0.0) )
C
C      ZI(9,M,I) = ZI(9,M,I) + S / 2.0 *
C      *      ( SX(PMI,A(N),0.0) - SX(PPI,A(N),0.0) )
C
C      ZI(10,M,I) = ZI(10,M,I) + S / 2.0 *
C      *      ( SX(PMQ,A(N),0.0) - SX(PQI,A(N),0.0) )
C
C      ZI(11,M,I) = ZI(11,M,I) + T1 * QM * QI / 2.0 * S *
C      *      ( SX(QMI,A(N),0.0) + SX(QQI,A(N),0.0) )
C
C      ZI(12,M,I) = ZI(12,M,I) + S / 2.0 *
C      *      ( SX(QMI,A(N),0.0) - SX(QQI,A(N),0.0) )
C
C      DO 90 J = 1, IS
C      QJ = PZ(J,2.0*D)
C
C      ZII(4,M,I,J) = ZII(4,M,I,J) + T1 * QM * QI * QJ / 4.0 * S *
C      *      ( SX(QMI-QJ,A(N),0.0) + SX(QMI+QJ,A(N),0.0) -
C      *      SX(QQI-QJ,A(N),0.0) - SX(QQI+QJ,A(N),0.0) )
C
C      DO 90 K = 1, IS
C      QK = PZ(K,2.0*D)
C      QJK = QJ - QK
C      QQK = QJ + QK
C
C      ZIII(6,M,I,J,K) = ZIII(6,M,I,J,K) + T1 * QM * QI * QJ * QK / 8.0
C      *      * S * ( SX(QMI + QJK,A(N),0.0) - SX(QMI + QQK,A(N),0.0)
C      *      - SX(QMI - QQK,A(N),0.0) + SX(QMI - QJK,A(N),0.0)
C      *      - SX(QQI + QJK,A(N),0.0) + SX(QQI + QQK,A(N),0.0)
C      *      + SX(QQI - QQK,A(N),0.0) - SX(QQI - QJK,A(N),0.0) )
C
C      90 CONTINUE
C      100 CONTINUE
C      *****
C
C      NOW THAT THE Is ARE CALCULATED, THEY ARE STORED AND USED IN
C      THE SECOND PART. WHAT FOLLOWS ARE THE WRITE STATEMENTS THAT
C      FORM DATA FILES. AT THIS PART, NO DATA IS WRITTEN TO THE TERMINAL.
C      ONE MUST PROCEED TO ---- MCDTWO.FOR ----- TO CONTINUE THE PROGRAM.
C      *****
C
C      OPEN(41,FILE='IN2A.DAT',STATUS='NEW')
C

```

```

      WRITE(41,620) IS, EX, GXZ, T, KMAX, NPST, NPMX, D
      DO 110 NK = 1, 12
      DO 110 M = 1, IS
      DO 110 I = 1, IS
110    WRITE(41,630) ZI(NK, M, I)
      DO 120 NK = 1, 4
      DO 120 M = 1, IS
      DO 120 I = 1, IS
      DO 120 J = 1, IS
120    WRITE(41,630) ZII(NK, M, I, J)
      DO 130 NK = 1, 6
      DO 130 M = 1, IS
      DO 130 I = 1, IS
      DO 130 J = 1, IS
      DO 130 K = 1, IS
130    WRITE(41,630) ZIII(NK, M, I, J, K)
C
C * * * * *
C
      OPEN(42, FILE='IN3A.DAT', STATUS='NEW')
C
      WRITE(42,640) EX, GXZ, T, XL, D, IS, SIGU, TAUU, AA0
      WRITE(42,650) NX
      DO 150 I = 1, NX
150    WRITE(42,660) XX(I)
      WRITE(42,665) NSL, NSD
      DO 160 I = 2, NSL + 1
160    WRITE(42,670) SE1(I), ET1(I), EI1(I)
      DO 170 I = 2, NSD + 1
170    WRITE(42,670) SE2(I), ET2(I), EI2(I)
      WRITE(*,680)
C
500    FORMAT(/,2X,' This is program one of a three part series.',/,
*      ' It simulates failure of stiffener/web separation. ',/,)
510    FORMAT(/,2X,'Please input 1/2 the beam's length and ',/,9x,
*      ' the length of the doubler. ( XL, D) ',/,)
520    FORMAT(/,4X,'Input the number of steps in 1/2 of the beam ',/,
*      ' and the number of steps in the doubler. (NSL,NSD) ',/,
*      ' (The maximum of each is 15.) ',/,)
540    FORMAT(/,2X,' Input the x-coordinates of the beam steps. ',/,
*      ' Include the extensional and bending stiffness per ',/,
*      ' unit width for the step which ends at this point. ',/,
*      ' Only one step per line. The last one at x = L/2. ',/,
*      ' There should be -NSL- inputs. (SE1, ET1, EI1) ',/,)
550    FORMAT(/,2X,'Input the x-coordinates of the doubler steps.',/,
*      ' Include the extensional and bending stiffness per ',/,
*      ' unit width for the step which ends at this point. ',/,
*      ' Only one step per line with the last one at x = D. ',/,
*      ' There should be -NSD- inputs. ( SE2, ET2, EI2 ) ',/,)
555    FORMAT(/,2X,' Please input the final load applied ',/,
*      ' to the structure. (NPMX) ',/,)
560    FORMAT(/,2X,'Input the Young and Shear Modulus of adhesive.',/,
*      ' the adhesive thickness, the ultimate tensile (peel) ',/,
*      ' and shear strength of the adhesive, and the average ',/,
*      ' characteristic length of the average stress. ',/,
*      ' (EX, GXZ, T, SIGU, TAUU, A0) ',/,)
620    FORMAT(2X,I6,/,3(5X,E20.13,/,),5X,I8,/,3X,I6,/,3X,I6,/,5X,E20.13)
630    FORMAT(3X, E20.13)
640    FORMAT(5(3X,E20.13,/,),3X,I8, 3(/,3X,E20.13),/,4X,I8,/,4X,I8,/,
*      3X,E15.8)
650    FORMAT(3X, I8 ,/,)
660    FORMAT(3X, E20.13 , / )
665    FORMAT(3X, I8, /, 3X , I8 )
670    FORMAT(3X, E20.13, /, 3X, E20.13, / , 3X, E20.13 )
680    FORMAT(3X,' The first portion of the program is finished.',/,
*      ' The next step is to execute the second program. To',

```

```

      *      ' do this type in :      MCDTWO ',/)
C
      STOP
      END
      FUNCTION PZ(NUI,U)
C
C      THIS FUNCTION IS USED GENERICALLY TO TAKE A GIVEN INPUT AND
C      FIND P, Q, OR R OF THE PARTICULAR INPUT.
C       $P = (2N-1) * (PI/L)$ ,  $R = (2M-1)*PI/(L-D)$ ,  $Q = (2N-1) * (PI/(2*D))$ 
C
      PZ = 4.0 * ATAN(1.0) * (2*NUI - 1) / U
      RETURN
      END
C
      FUNCTION SX( ZC, ZA, ZD )
C
C      THIS FUNCTION IS SIMILUR TO : SIN(X)/X WITH THE EXCEPTION OF
C      THE CASE WHERE X IS ZERO.
C
      GAGE = 0.00005
      IF (ZC.NE.0) THEN
        SX = SIN( (ZC*ZA) - ZD) / ZC
      ELSE
        SX = ZA * COS(-ZD)
      END IF
      IF (ABS(SX).LT.GAGE) SX = 0.0
      RETURN
      END
C
C      THIS IS THE PROGRAM TO SOLVE THE NON-LINEAR EQUATIONS.
C      IT USES IMSL LIBRARY " ZSCNT " AND A SUBROUTINE TO
C      SET UP THE EQUATIONS.....NAME OF PROGRAM: MCDTWO.FOR
C
C      THIS PROGRAM USES THE LINEAR SOLUTION FOR THE INITIAL GUESS
C      OF THE INITIAL LOAD ONLY. AFTER THAT, THE PREVIOUS VALUES
C      OF THE SOLUTION ARE USED.
C      IT USES THE ' COMMON ' STATEMENT TO PASS VARIABLES TO FUNCTION
C
C      FILE 41 is the INPUT and was the OUTPUT from PROG1 ( IN2A.DATA )
C      FILE 45 is the OUTPUT and is the INPUT for PROG3 ( IN3B.DATA )
C
C *****
C      DIMENSION X(25), WK(2200), ZMM(15,15),
C      * BBB(350,25), RT(15), PHLD(350), FX(350), XHD(25)
C      INTEGER J1, J2, IS, IPP, IPS, KMAX, J22
C      EXTERNAL FN
C      COMMON/COEF/ZI(12,5,5),EZ,GXZ,T,P
C      COMMON/COEF1/ZII(4,5,5,5),ZIII(6,5,5,5,5)
C      COMMON/COEFF/ ZM(25,25)
C *****
C      READ IN DATA FROM THE FIRST PROGRAM AND INITIALIZE VALUES FOR THE
C      SOLUTION OF THE NON-LINEAR EQUATION SOLVER.
C
C *****
C      OPEN(41,FILE='IN2A.DAT',STATUS='OLD')
C      OPEN(45,FILE='IN3B.DAT',STATUS='NEW')
C      OPEN(48,FILE='ERROR.DAT',STATUS='NEW')
C      READ(41,*) IS, EZ, GXZ, T, KMAX, IPS, IPP, D
C      DO 5 NK = 1, 12
C      DO 5 N = 1, IS
C      DO 5 I = 1, IS
C      5 READ(41,*) ZI(NK,N,I)
C      J1 = IS * 5
C      J22 = 3 * IS
C *****

```



```

C
C SET UP THE M AND E MATRICES FOR LINEAR PART OF THE SOLUTION
C OF THE SYSTEM OF EQUATIONS --- M*B = Q OR E*BB = Q
C *****
C DO 100 N = 1, IS
C RT(N) = IPS
C RT(N+IS) = IPS
C RT(N+2*IS) = 0.000
C DO 100 K = 1, 5
C DO 100 J = 1, IS
C
C L = J + (K-1) * IS
C IF (K-2) 30, 40, 50
C
C 30 ZM(N,L) = 2*ZI(1,N,J) + 2*EZ*ZI(5,N,J) / T
C ZM(N+IS,L) = 2 * ZI(3,J,N)
C ZM(N+2*IS,L) = -2 * EZ * ZI(6,J,N) / T
C GOTO 100
C
C 40 ZM(N,L) = 2 * ZI(3,N,J)
C ZM(N+IS,L) = 2 * ZI(4,N,J)
C GOTO 100
C
C 50 IF (K-4) 60, 70, 80
C
C 60 ZM(N,L) = -2 * EZ * ZI(6,N,J) / T
C ZM(N+2*IS,L) = 2*ZI(7,N,J) + 2*EZ*ZI(8,N,J)/T
C GOTO 100
C
C 70 ZM(N+3*IS,L) = 2 * ZI(2,N,J) + 2 * GXZ * ZI(9,N,J) / T
C ZM(N+4*IS,L) = -2 * GXZ * ZI(10,J,N) / T
C GOTO 100
C
C 80 ZM(N+3*IS,L) = -2 * GXZ * ZI(10,N,J) / T
C ZM(N+4*IS,L) = 2 * ZI(11,N,J) + 2 * GXZ * ZI(12,N,J) / T
C 100 CONTINUE
C
C DO 99 I = 1, J22
C DO 99 J = 1, J22
C 99 ZMM(I,J) = ZM(I,J)
C
C CALL SOLV(ZMM, RT,J22)
C *****
C THE REST OF THE PROGRAM ARE VARIABLES THAT WILL BE DEPENDENT ON THE
C SOLUTION OF THE SYSTEM OF EQUATIONS. (IE. THE NON-LINEAR RIGHT-HAND-
C SIDE.) THE LOOP THAT IS USED TO SOLVE THE SYSTEM OF EQUATIONS
C BEGINS HERE. THE Q'S ARE THE NON-LINEAR PART OF THE SYSTEM.
C
C M*B = Q OR E*BB = Q
C *****
C
C ICT = (J1 * J1) + 1
C DO 10 NK = 1, 4
C DO 10 N = 1, IS
C DO 10 I = 1, IS
C DO 10 J = 1, IS
C ICT = ICT + 1
C 10 READ(41,*) ZII(NK,N,I,J)
C
C DO 15 NK = 1, 6
C DO 15 N = 1, IS
C DO 15 I = 1, IS
C DO 15 J = 1, IS

```

```

DO 15 K = 1, IS
  ICT = ICT + 1
15  READ(41,*) ZIII(NK,N,I,J,K)
    IW = 0
    DO 20 I = J22
20  X(I) = RT(I)
    DO 21 I = J22+1, J1
21  X(I) = 0.00005
    IV = 0
    DO 185 NA = IPS, IPP
      P = IPS + IV
C
C  CALL THE IMSL ROUTINE HERE
C
      NSIG = 2
      J2 = J1
      IER = 0
      WRITE(*,*) ' P = ', P
C
      CALL ZSCNT( FN, NSIG, J2, KMAX, P, X, FNORM, WK, IER )
C
      IV = IV + 1
      INTT = 0
      IF ( IER.NE. 0 ) GO TO 185
      CALL FNCT( J1, X, FXX )
      WRITE(*,*) ' P AND FXX ', P, FXX
C
188  IW = IW + 1
      PHLD(IW) = P
      FX(IW) = FXX
CC  WRITE(45,*) P, FXX
      DO 181 JP = 1, J1
CC  WRITE(45,*) X(JP)
181  BBB(IW,JP) = X(JP)
185  CONTINUE
C
      WRITE(45,*) IW
      DO 220 NA = 1, IW
      WRITE(45,530) PHLD(NA), FX(NA)
      DO 220 I = 1, J1
220  WRITE(45,530) BBB(NA,I)
      WRITE(*,540)
C *****
C 530  FORMAT( 3X, E20.13 )
540  FORMAT( 3X, ' This is the end of the second program.',/,
C  ' To finish executing the program - type: MCDTHREE ',//)
C
      STOP
      END
C
      SUBROUTINE FN( B, Q, J2, PP)
C
C  THIS SUBROUTINE FINDS THE EXPRESSIONS THAT MINIMIZE THE ENERGY
C  EQUATION. (IE dV/dA=0.0, dV/dB=0.0, etc. ) Used in the IMSL
C  routine (ZSCNT Subroutine)
C
      REAL B(25), Q(25), PP
      COMMON/COEFF/TM(25,25)
      COMMON/COEF1/ZII(4,5,5,5), ZIII(6,5,5,5,5)
C
      IR = J2 / 5
      ICC = 1
C
      DO 140 N = 1, IR
      Q1 = 0.0

```

```

Q2 = 0.0
Q3 = 0.0
Q4 = 0.0
Q5 = 0.0
Q11 = 0.0
Q22 = 0.0
Q33 = 0.0
Q111 = 0.0
Q222 = 0.0
Q333 = 0.0
Q444 = 0.0
Q555 = 0.0

```

C

```

DO 130 I = 1, IR
  Q111 = Q111 + TM(N,I) * B(I) + TM(N,I+IR) * B(I+IR)
  Q222 = Q222 + TM(N,I+2*IR) * B(I+2*IR)
  Q333 = Q333 + TM(N+IR,I) * B(I) + TM(N+IR,I+IR) * B(I+IR)
  Q444 = Q444 + TM(N+2*IR,I) * B(I) + TM(N+2*IR,I+2*IR)
  Q555 = Q555 + TM(N+3*IR,I+3*IR) * B(I+3*IR) +
  TM(N+3*IR,I+4*IR) * B(I+4*IR)
  Q555 = Q555 + TM(N+4*IR,I+3*IR) * B(I+3*IR) +
  TM(N+4*IR,I+4*IR) * B(I+4*IR)
DO 130 J = 1, IR

```

C

```

  Q1 = Q1 + 2.0 * B(I) * B(J+3*IR) * ZII(1,N,I,J) +
  2.0 * B(I+IR) * B(J+3*IR) * ZII(3,N,I,J)
  Q2 = Q2 + 2.0 * B(I+IR) * B(J+3*IR) * ZII(2,J,N,I) +
  2.0 * B(I) * B(J+3*IR) * ZII(3,I,N,J)
  Q3 = Q3 + 2.0 * B(I+2*IR) * B(J+4*IR) * ZII(4,N,I,J)
  Q4 = Q4 + B(I) * B(J) * ZII(1,J,I,N) +
  B(I+IR) * B(J+IR) * ZII(2,N,I,J) +
  2.0 * B(I) * B(J+IR) * ZII(3,I,J,N)
  Q5 = Q5 + B(I+2*IR) * B(J+2*IR) * ZII(4,J,I,N)

```

C

```

DO 130 K = 1, IR
  Q11 = Q11 + B(I) * B(J) * B(K) * ZIII(1,N,I,J,K) +
  3.0 * B(I) * B(J+IR) * B(K+IR) * ZIII(2,N,I,J,K) +
  B(I+IR) * B(J+IR) * B(K+IR) * ZIII(3,N,I,J,K) +
  3.0 * B(I) * B(J) * B(K+IR) * ZIII(4,N,I,J,K)
  Q22 = Q22 + B(I+IR) * B(J+IR) * B(K+IR) * ZIII(5,N,I,J,K) +
  3.0 * B(I+IR) * B(J) * B(K) * ZIII(2,K,J,I,N) +
  3.0 * B(I+IR) * B(J+IR) * B(K) * ZIII(3,K,N,I,J) +
  B(I) * B(J) * B(K) * ZIII(4,K,J,I,N)
  Q33 = Q33 + B(I+2*IR) * B(J+2*IR) * B(K+2*IR)
  * ZIII(6,N,I,J,K)

```

C

130

C

CONTINUE

```

Q(N) = 2.0 * PP + Q1 + Q11 + Q111
Q(N+IR) = 2.0 * PP + Q2 + Q22 + Q222
Q(N+2*IR) = Q3 + Q33 + Q333
Q(N+3*IR) = Q4 + Q444
Q(N+4*IR) = Q5 + Q555

```

140

C

CONTINUE

```

RETURN
END

```

C

SUBROUTINE SOLV (ZN,Q,J8)

C

C

This subroutine solves a system of linear equations.

C

DIMENSION ZN(15,15), E(325), Q(15)

C

DO 12 I = 1, J8

```

12      DO 12 J = 1, J8
C      E(J+(I-1)*J8) = -ZN(J,I)

      IT = J8
      TOL = 0.0
      KS = 0
      JJ = - IT
      DO 380 J = 1, IT
        JY = J + 1
        JJ = JJ + IT + 1
        BIGA = 0.0
        IL = JJ - J
        DO 320 I=J, IT
          IJ = IL + I
          IF ( ABS(BIGA) - ABS(E(IJ)) ) 310, 320, 320
310      BIGA = E(IJ)
          IMAX = I
320      CONTINUE
          IF ( ABS(BIGA) - TOL ) 330, 330 , 340
330      KS = 1
          WRITE(*,*) ' THE MATRIX IS SINGULAR '
          STOP
340      I1 = J + IT * (J-2)
          IL = IMAX - J
          DO 350 K = J, IT
            I1 = I1 + IT
            I2 = I1 + IL
            SAVE = E(I1)
            E(I1) = E(I2)
            E(I2) = SAVE
350      E(I1) = E(I1) / BIGA
            SAVE = Q(IMAX)
            Q(IMAX) = Q(J)
            Q(J) = SAVE / BIGA
            IF ( J-IT ) 360, 390, 360
360      IQS = IT * (J-1)
            DO 380 IX = JY, IT
              IXJ = IQS + IX
              IL = J - IX
              DO 370 JX = JY, IT
                IXJX = IT * ( JX-1 ) + IX
                JJX = IXJX + IL
370      E(IXJX) = E(IXJX) - ( E(IXJ) * E(JJX) )
380      Q(IX) = Q(IX) - ( Q(J) * E(IXJ) )
390      NY = IT - 1
          IL = IT * IT
          DO 400 J = 1, NY
            IA = IL - J
            IB = IT - J
            IC = IT
            DO 400 K = 1, J
              Q(IB) = Q(IB) - E(IA) * Q(IC)
              IA = IA - IT
              IC = IC - 1
400      CONTINUE
C *****
C      END OF EQUATION SOLVER.....
C *****
      RETURN
      END

C
C      SUBROUTINE FNCT( NN, Q,V )
C
C      This subroutine calculates the value of the total strain energy
C      Used as a comparison check when more then one initial guess is used.
C

```

```

      REAL    Q(25), V
      COMMON/COEF/ZI(12,5,5), EZ, GXZ,T, P
      COMMON/COEF1/ZII(4,5,5,5), ZIII(6,5,5,5,5)

C
      V = 0.000
      IS = NN/5
      DO 100  M = 1, IS

C
      V = V + 2 * P * ( Q(M) + Q(M+IS) )
      DO 100  I = 1, IS

C
      V = V + Q(M)*Q(I)*ZI(1,M,I) + Q(M+3*IS)*Q(I+3*IS)*ZI(2,M,I)
C      + 2.0*Q(M)*Q(I+IS)*ZI(3,M,I) + Q(M+IS)*Q(I+IS)*ZI(4,M,I)
C      + EZ/T*( Q(M)*Q(I)*ZI(5,M,I) - 2.0*Q(M)*Q(I+2*IS)*ZI(6,M,I) )
C      + Q(M+2*IS)*Q(I+2*IS)*( ZI(7,M,I) + EZ/T*ZI(8,M,I) )
C      + GXZ/T * ( Q(M+3*IS)*Q(I+3*IS)*ZI(9,M,I)
C      - 2 * Q(M+3*IS)*Q(I+4*IS)*ZI(10,M,I) )
C      + Q(M+4*IS)*Q(I+4*IS)*( ZI(11,M,I) + GXZ/T*ZI(12,M,I) )

C
      DO 100 J = 1, IS

C
      V = V + Q(M)*Q(I)*Q(J+3*IS)*ZII(1,M,I,J)
C      + Q(M+IS)*Q(I+IS)*Q(J+3*IS)*ZII(2,M,I,J)
C      + 2.0*Q(M)*Q(I+IS)*Q(J+3*IS)*ZII(3,M,I,J)
C      + Q(M+2*IS)*Q(I+2*IS)*Q(J+4*IS)*ZII(4,M,I,J)

C
      DO 100 K = 1, IS

C
      V = V + Q(M)*Q(I)*Q(J)*Q(K)*ZIII(1,M,I,J,K) / 4.0
C      + 1.5*Q(M)*Q(I)*Q(J+IS)*Q(K*IS)*ZIII(2,M,I,J,K)
C      + Q(M)*Q(I+IS)*Q(J+IS)*Q(K*IS)*ZIII(3,M,I,J,K)
C      + Q(M)*Q(I)*Q(J)*Q(K+IS)*ZIII(4,M,I,J,K)
C      + Q(M+IS)*Q(I+IS)*Q(J+IS)*Q(K+IS)*ZIII(5,M,I,J,K) / 4.0
C      + Q(M+2*IS)*Q(I+2*IS)*Q(J+2*IS)*Q(K+2*IS)*ZIII(6,M,I,J,K)/4.0

C
100  CONTINUE
      RETURN
      END

C *****
C
C      The rest of the subroutines are part of a IMSL Library that
C      were used to solve a system of non-linear equations and
C      can by pass a load condition where a local minimum is
C      reached instead of the actual minimum.
C *****
C
C      IMSL ROUTINE NAME      - ZSCNT
C -----
C
C      SUBROUTINE ZSCNT (FCN, NSIG, N, ITMAX, PAR, X, FNORM, WK, IER)
C                                SPECIFICATIONS FOR ARGUMENTS
C      INTEGER                IER, ITMAX, N, NSIG
C      DIMENSION              PAR(1), WK(1), X(N)
C                                SPECIFICATIONS FOR LOCAL VARIABLES
C      INTEGER                I1, I2, I3, I4, I5, LNEW, LOLD, N1
C      EXTERNAL                FCN

C
C                                FIRST EXECUTABLE STATEMENT
C
      N1=N+1
      I1 = N1*N1+1
      I2 = I1+N*N1
      I3 = I2+N1
      I4 = I3+N1
      I5 = I4+N1
      CALL UERSET(0, LOLD)

```

```

      CALL ZSCNU(X,N,FCN,NSIG,N1,WK(1),WK(I1),WK(I2),WK(I3),WK(I4)
* ,WK(I5),ITMAX,PAR,IER)
      CALL FCN(X,WK,N,PAR)
      FNORM = 0.0D0
      DO 5 I = 1,N
        FNORM = FNORM + WK(I)*WK(I)
5    CONTINUE
      CALL UERSET(LOLD,LNEW)
      IF (IER.EQ.0) GO TO 9005
9000  CONTINUE
      CALL UERTST(IER,'ZSCNT ')
9005  RETURN
      END

```

```

C
C   IMSL ROUTINE NAME   -  ZSCNU
C-----
C

```

```

      SUBROUTINE ZSCNU (X,N,FCN,NDIGIT,N1,A,Z,Y,XNORM,B,WK,
1  MAXIT,PAR,IER)

```

```

C                                     SPECIFICATIONS FOR ARGUMENTS
      INTEGER                      IER,MAXIT,N,NDIGIT,N1
      DIMENSION                    A(N1,1),B(1),PAR(1),WK(1),X(1),XNORM(1),
1  Y(1),Z(N,1)

```

```

C                                     SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER                      I,IBMAX,IBNORM,IDGT,IEVAL,ITER,J,JER,JI,JS,
1  MI,NRS,NSTART
      REAL*8    BIG,BNORM,CFACT,DX,EPS,HALF,HLMAX,RACC,REPS,
1  RRX,RX,SFACT,SMALL,TEST,TN,TR

```

```

      REAL*8    DSEED
      IER = 0
      SMALL = .2220446049E-15
      DSEED = 12345.0D0
      CFACT = 0.99D0
      BIG = 5.0D5
      SFACT = 0.1D0
      IBMAX = 50
      NRS = 2
      RACC = DMIN1(DMAX1(SMALL,10.0D0**(-NDIGIT)),0.1D0)
      REPS = DSQRT(SMALL)
      HLMAX = 3.0D0

```

```

C                                     INITIALIZATION OF OTHER VARIABLES

```

```

      ITER = 0
      IEVAL = 0
      NSTART = 0
      IBNORM = 0
      RX = 1.0D0
      RRX = 0.0D0
      EPS = 0.0D0
      DO 5 I=1,N
        B(I) = 0.0
        A(N1,I) = 1.00
        Z(I,N1) = X(I)

```

```

5    CONTINUE
      B(N1) = 1.0
      A(N1,N1) = 1.00
      JI = N1
      IEVAL = IEVAL+1
      CALL FCN (Z(1,N1),A(1,N1),N,PAR)
      BNORM = 0.0D0
      DO 10 I=1,N
        BNORM = BNORM+DBLE(A(I,N1)*A(I,N1))
10   CONTINUE
      XNORM(N1) = REAL(BNORM)

```

```

C
C
C

```

```

      STARTING PROCEDURE TO GENERATE N
      VECTORS IN AN EPS NEIGHBORHOOD OF
      X

```

```

15 IF (NSTART.EQ.NRS) EPS = EPS*10.0D0
   IF (NSTART.EQ.0) EPS = DMIN1(RX,REPS)
   IF (EPS.GT.BIG) GO TO 120
   NSTART = MOD(NSTART,NRS)+1
   DO 30 J=1,N
     DO 25 I=1,N
20       TR = (GGUBFS(DSEED)-0.5D0)*2.0D0
         IF (DABS(TR).LT.0.1D0) GO TO 20
         Z(I,J) = X(I)+DMAX1(DABS(DBLE(X(I))),0.1D0)*TR*EPS
25       CONTINUE
         IEVAL = IEVAL+1
         CALL FCN (Z(1,J),A(1,J),N,PAR)
30 CONTINUE
C                                     CALCULATE N NORMS
   DO 35 J=1,N
     XNORM(J) = 0.0
     DO 35 I=1,N
       XNORM(J) = XNORM(J)+A(I,J)*A(I,J)
35 CONTINUE
C                                     FIND JI SO THAT
C                                     XNORM(JI).LE.XNORM(J) FOR J=1,N1
C                                     FIND JS SO THAT
C                                     XNORM(JS).GE.XNORM(J) FOR J=1,N1
40 JI = N1
   JS = JI
   DO 37 J=1,N
     IF (XNORM(J).GT.XNORM(JS)) JS = J
     IF (XNORM(J).LT.XNORM(JI)) JI = J
37 CONTINUE
C                                     TEST FOR F(X)=0
   IF (XNORM(JI).EQ.0.0) GO TO 125
C                                     TEST FOR NON-CONVERGENCE
   IF (DBLE(XNORM(JI)).GT.SFACT*BNORM) GO TO 50
   BNORM = DBLE(XNORM(JI))
   IBNORM = ITER
50 IF ((ITER-IBNORM).GT.IBMAX) GO TO 120
C                                     INCREMENT AND CHECK ITERATION COUNT
   ITER = ITER+1
   IF (ITER.GE.MAXIT) GO TO 115
C                                     SOLVE FOR MULTIPLIERS
   DO 55 MI=1,N1
     Y(MI) = B(MI)
55 CONTINUE
   IDGT = 0
   CALL LEQT2F (A,1,N1,N1,Y,IDGT,WK,JER)
C                                     TEST FOR SINGULARITY OF A
   IF (JER.NE.0) GO TO 85
C                                     COMPUTE NEXT APPROXIMATION X
   DO 65 I=1,N
     DX = 0.0D0
     DO 60 J=1,N1
       DX = DX + DBLE(Y(J)*Z(I,J))
60   CONTINUE
     X(I) = REAL(DX)
65 CONTINUE
   HALF = 0.0D0
C                                     EVALUATE FUNCTION AT X
70 IEVAL = IEVAL+1
   CALL FCN (X,Y,N,PAR)
C                                     CALCULATE NORM OF F(X)
   TN = 0.0D0
   DO 75 I=1,N
     TN = TN+DBLE(Y(I)*Y(I))
75 CONTINUE
C                                     TEST FOR IMPROVEMENT
   IF (TN.LT.DBLE(XNORM(JS))) GO TO 95

```

```

C                                     EMPLOY HALVING TECHNIQUE
      HALF = HALF+1.D0
      IF (HALF.GT.HLMAX) GO TO 85
      DO 80 I=1,N
        X(I) = (X(I)+( REAL(HALF) ) *Z(I,JI))/( REAL(HALF+1.00) )
      80 CONTINUE
      GO TO 70

C                                     START AGAIN AT BEST APPROXIMATION
      85 IF (JI.EQ.N1) GO TO 15
      XNORM(N1) = XNORM(JI)
      DO 90 I=1,N
        Z(I,N1) = Z(I,JI)
        A(I,N1) = A(I,JI)
      90 CONTINUE
      GO TO 15

C                                     TEST FOR RELATIVE (COMPONENT)
C                                     CONVERGENCE OF APPROXIMATIONS
      95 IF ((HALF.NE.0.D0).OR.(ITER.EQ.1)) GO TO 105
      RX = SMALL
      DO 100 I=1,N
        RX = DMAX1( RX,DBLE( ABS(X(I)-Z(I,JI)) /
          * AMAX1( ABS(X(I)),0.1000 ) ) )
      100 CONTINUE
      RRX = DMAX1(-DLOG10(RX),0.0D0)
      IF (RX.LE.RACC) GO TO 125

C                                     REPLACE WORST APPROXIMATION WITH X
      105 IF (TN.LT.CFACT*DBLE(XNORM(JI))) NSTART = 0
      XNORM(JS) = REAL(TN)
      DO 110 I=1,N
        Z(I,JS) = X(I)
        A(I,JS) = Y(I)
      110 CONTINUE
      GO TO 40

C                                     SET IER AND RETURN
      115 IER = 129
      GO TO 125
      120 IER = 130
      125 DO 130 I=1,N
        X(I) = Z(I,JI)
      130 CONTINUE
      RETURN
      END

C
C   IMSL ROUTINE NAME    - GGUBFS
C   -----
C
C   REAL FUNCTION GGUBFS (DSEED)
C
C   REAL*8    DSEED                                     SPECIFICATIONS FOR ARGUMENTS
C
C   REAL*8    D2P31M,D2P31                               SPECIFICATIONS FOR LOCAL VARIABLES
C
C   DATA                                     D2P31M=(2**31) - 1
C   DATA                                     D2P31 =(2**31) (OR AN ADJUSTED VALUE)
C   DATA                                     D2P31M/2147483647.D0/
C   DATA                                     D2P31 /2147483648.D0/
C
C   FIRST EXECUTABLE STATEMENT
      DSEED = DMOD(16807.D0*DSEED,D2P31M)
      GGUBFS = DSEED / D2P31
      RETURN
      END
C   IMSL ROUTINE NAME    - LUDATN
C   -----
C
C   SUBROUTINE LUDATN (A,IA,N,LU,ILU,IDGT,D1,D2,APVT,EQUIL,WA,IER)

```


8-100

```

      WI = WI+ABS(T)
31      CONTINUE
      LU(I,J) = SUM
50      WI = WI+ABS(SUM)
      IF (AI .EQ. ZERO) AI = BIGA
      TEST = WI/AI
      IF (TEST .GT. WREL) WREL = TEST
      GO TO 65
C      WITHOUT ACCURACY TEST
55      IF (JMI .LT. 1) GO TO 65
      DO 60 K=1,JMI
          SUM = SUM-LU(I,K)*LU(K,J)
60      CONTINUE
      LU(I,J) = SUM
65      Q = EQUIL(I)*ABS(SUM)
      IF (P .GE. Q) GO TO 70
      P = Q
      IMAX = I
70      CONTINUE
C      TEST FOR ALGORITHMIC SINGULARITY
      IF (P.EQ. ZERO) 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      APVT(J) = IMAX
      D1 = D1*LU(J,J)
85      IF (ABS(D1) .LE. ONE) GO TO 90
      D1 = D1*SIXTH
      D2 = D2+FOUR
      GO TO 85
90      IF (ABS(D1) .GE. SIXTH) GO TO 95
      D1 = D1*SIXTN
      D2 = D2-FOUR
      GO TO 90
95      CONTINUE
      JP1 = J+1
      IF (JP1 .GT. N) GO TO 105
C      DIVIDE BY PIVOT ELEMENT U(J,J)
      P = LU(J,J)
      DO 100 I=JP1,N
          LU(I,J) = LU(I,J)/P
100      CONTINUE
105      CONTINUE
C      PERFORM ACCURACY TEST
      IF (IDGT .EQ. 0) GO TO 9005
      P = 3*N+3
      WA = P*WREL
      IF (WA+10.0**(-IDGT) .NE. WA) GO TO 9005
      IER = 34
      GO TO 9000
C      ALGORITHMIC SINGULARITY
110      IER = 129
      D1 = ZERO
      D2 = ZERO
9000      CONTINUE
C      PRINT ERROR
      CALL UERTST(IER,'LUDATN')
9005      RETURN
      END
C      IMSL ROUTINE NAME      - LUELMN

```

```

C -----
C
C      SUBROUTINE LUELMN (A, IA, N, B, APVT, X)
C
C      DIMENSION          A(IA,1), B(1), APVT(1), X(1)
C                          FIRST EXECUTABLE STATEMENT
C                          SOLVE LY = B FOR Y
C
C      DO 5 I=1,N
5      X(I) = B(I)
      IW = 0
      DO 20 I=1,N
          IP = APVT(I)
          SUM = X(IP)
          X(IP) = X(I)
          IF (IW .EQ. 0) GO TO 15
          IM1 = I-1
          DO 10 J=IW, IM1
              SUM = SUM - A(I, J) * X(J)
10          CONTINUE
          GO TO 20
15      IF (SUM .NE. 0.) IW = I
20      X(I) = SUM
C
C                          SOLVE UX = Y FOR X
C
C      DO 30 IB=1,N
      I = N+1-IB
      IP1 = I+1
      SUM = X(I)
      IF (IP1 .GT. N) GO TO 30
      DO 25 J=IP1,N
          SUM = SUM - A(I, J) * X(J)
25      CONTINUE
30      X(I) = SUM/A(I, I)
      RETURN
      END
C      IMSL ROUTINE NAME    - LUREFN
C -----
C
C      SUBROUTINE LUREFN (A, IA, N, UL, IUL, B, IDGT, 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      REAL*8              SUM
C      DATA               ITMAX/50/, ZERO/0.0/
C                          FIRST EXECUTABLE STATEMENT
C
C      IER=0
C      XNORM = ZERO
C      DO 10 I=1,N
          XNORM = AMAX1(XNORM, ABS(X(I)))
10      CONTINUE
      IF (XNORM .NE. ZERO) GO TO 20
      IDGT = 50
      GO TO 9005
20      DO 77 ITER=1, ITMAX
          DO 30 I=1,N
              SUM = DBLE(B(I))
              DO 25 J=1,N
                  SUM = SUM - DBLE(A(I, J)) * DBLE(X(J))
25          CONTINUE
              RES(I) = SUM
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 = AMAX1(DXNORM, ABS(DX(I)))
          35
      77

```

```

      XNORM = AMAX1(XNORM,ABS(X(I)))
35      CONTINUE
      IF (ITER.NE. 1) GO TO 40
      IDGT = 50
      IF (DXNORM.NE. ZERO) IDGT = -ALOG10(DXNORM/XNORM)
40      IF (XNORM+DXNORM.EQ. XNORM) GO TO 9005
77      CONTINUE
C      ITERATION DID NOT CONVERGE
      IER = 129
9000     CONTINUE
      CALL UERTST(IER,'LUREFN')
9005     RETURN
      END
C      IMSL ROUTINE NAME      - UERSET
C      -----
C
C      SUBROUTINE UERSET (LEVEL,LEVOLD)
C                                  SPECIFICATIONS FOR ARGUMENTS
C      INTEGER                     LEVEL,LEVOLD
C                                  FIRST EXECUTABLE STATEMENT
C      LEVOLD = LEVEL
C      CALL UERTST (LEVOLD,'UERSET')
C      RETURN
C      END
C
C      IMSL ROUTINE NAME      - UERTST
C      -----
C
C      SUBROUTINE UERTST(IER,NAME)
C
C      INTEGER                     I, IEQDF, IOUNIT, LEVEL, LEVOLD, NIN
C
C      CHARACTER*6                 NAME(1), NAMSET(1)
C
C      DATA                      NAMSET/'UERSET'/
C
C      LEVEL = 4
C      IEQDF = 0
C
C      CALL UGETIO(1,NIN,IOUNIT)      GET OUTPUT UNIT NUMBER
C
C      CHECK IER
C
C      IF (IER.GT.999) GO TO 25
C      IF (IER.LT.-32) GO TO 65
C      IF (IER.LE.128) GO TO 5
C      IF (LEVEL.LT.1) GO TO 30
C
C      PRINT TERMINAL MESSAGE
C      WRITE(48,35) IER,NAME
C      GO TO 30
C      5 IF (IER.LE.64) GO TO 10
C      IF (LEVEL.LT.2) GO TO 30
C
C      PRINT WARNING WITH FIX MESSAGE
C      WRITE(48,40) IER,NAME
C      GO TO 30
C      10 IF (IER.LE.32) GO TO 15
C
C      PRINT WARNING MESSAGE
C      IF (LEVEL.LT.3) GO TO 30
C      WRITE(48,66) IER,NAME
C      GO TO 30
C      15 CONTINUE
C
C      CHECK FOR UERSET CALL
C
C      IF ( NAME(1).NE.NAMSET(1) ) GO TO 25
C      LEVOLD = LEVEL
C      LEVEL = IER
C      IER = LEVOLD
C      IF (LEVEL.LT.0) LEVEL = 4

```

```

      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
25  CONTINUE
      IF (LEVEL.LT.4) GO TO 30
C
      PRINT NON-DEFINED MESSAGE
      WRITE(48,50) IER,NAME
30  IEQDF = 0
35  FORMAT(19H *** TERMINAL ERROR,10X,7H( IER = ,I3,
      1      20H) FROM IMSL ROUTINE , '      = ',A6)
40  FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H( IER = ,I3,
      1      20H) FROM IMSL ROUTINE , '      = ',A6)
66  FORMAT(18H *** WARNING ERROR,11X,7H( IER = ,I3,
      1      20H) FROM IMSL ROUTINE , '      = ',A6)
50  FORMAT(20H *** UNDEFINED ERROR,9X,7H( IER = ,I5,
      1      20H) FROM IMSL ROUTINE , '      = ',A6)
C
C
65  RETURN
      END
C
C  IMSL ROUTINE NAME      - UGETIO
C-----
C
      SUBROUTINE UGETIO(IOPT,NIN,NOUT)
C
C          SPECIFICATIONS FOR ARGUMENTS
      INTEGER          IOPT,NIN,NOUT,NIND,NOUTD
      DATA            NIND/5/,NOUTD/6/
C
C          FIRST EXECUTABLE STATEMENT
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
      NOUT = NOUTD
      GO TO 9005
5  NIND = NIN
      GO TO 9005
10 NOUTD = NOUT
9005 RETURN
      END
C
C  IMSL ROUTINE NAME      - LEQT2F
C-----
C
      SUBROUTINE LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
C          DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
C
C          FIRST EXECUTABLE STATEMENT
C          INITIALIZE IER
      IER=0
      JER=0
      J = N*N+1
      K = J+N
      MM = K+N
      KK = 0
      MM1 = MM-1
      JJ=1
      DO 5 L=1,N
          DO 5 I=1,N
              WKAREA(JJ)=A(I,L)
              JJ=JJ+1
5  CONTINUE
C
C          DECOMPOSE A
      CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
      *      WA,IER)
      IF (IER.GT.128) GO TO 25
      IF (IDGT.EQ. 0 .OR. IER.NE. 0) KK = 1

```

8-105

```

C      OPEN (UNIT=42, FILE=OUTPUT, STATUS='NEW' )
      WRITE(42,500) OUTPUT,L,DD,T,EZ, GXZ , SIGU, TAUU, A0, NSL, NSD
      DO 11 I = 1, NSL
11     WRITE(42,501) SE1(I), ET1(I), EI1(I)
      WRITE(42, 502)
      DO 12 I=1,NSD
12     WRITE(42, 501) SE2(I), ET2(I), EI2(I)
C
      OPEN (UNIT=41, FILE='IN3B.DAT', STATUS='OLD')
      READ(41,*) ITM
      DO 60 K = 1,ITM
      DO 13 JH = 1, NX
      U1(JH) = 0.00000000
      U2(JH) = 0.00000000
      W2(JH) = 0.00000000
13     W1(JH) = 0.00000000
      READ(41,*) P, FXZ
      READ(41,*) ( A(I), I=1, IS )
      READ(41,*) ( B(I), I=1, IS )
      READ(41,*) ( C(I), I=1, IS )
      READ(41,*) ( D(I), I=1, IS )
      READ(41,*) ( E(I), I=1, IS )
C
C      CALCULATE THE AVERAGE STRESS ON THE CHARACTERISTIC LENGHT
C
      DO 51 J = 1, 100
      XXX = DD - J * (A0/100)
      XD = XXX - DD
      W11 = 0.0
      W22 = 0.0
      U11 = 0.0
      U22 = 0.0
      DO 33 I = 1, IS
      PI = PZ(I,L)
      QI = PZ(I,2*DD)
      RI = PZ(I,L-DD)
C
      W11 = W11 + A(I) * ( 1 - COS( PI*XXX ) )
      IF (XXX.GT.DD) W11 = W11 + B(I) * ( 1 - COS(RI*XD) )
C
      W22 = W22 + C(I) * ( 1 - COS(QI*XXX ) )
      U11 = U11 + D(I) * SIN( PI*XXX )
      U22 = U22 + E(I) * SIN( QI*XXX )
33     CONTINUE
C
      SIG = SIG + EZ * ( W11 - W22 ) / T
      TAU = TAU + GXZ * ( U11 - U22 ) / T
51     CONTINUE
C
      SIG = SIG /100/A0
      if (sig.lt.0) sig = 0.0
      TAU = TAU/100/A0
      F = (SIG*SIG) / (SIGU*SIGU) + ( TAU*TAU ) / (TAUU*TAUU)
C
      WRITE(42,551) P, F
      WRITE(*,551) P, F
C
      WRITE(42,510)
      DO 50 J = 1, NX
      XD = X(J) - DD
      DO 30 I = 1, IS
C
      PI = PZ(I,L)
      QI = PZ(I,2*DD)
      RI = PZ(I,L-DD)

```

```

C      W1(J) = W1(J) + A(I) * ( 1 - COS( PI*X(J) ) )
      IF (X(J).GT.DD) W1(J) = W1(J) + B(I) * ( 1 - COS(RI*XD) )
C
      W2(J) = W2(J) + C(I) * ( 1 - COS(QI*X(J) ) )
      U1(J) = U1(J) + D(I) * SIN( PI*X(J) )
      U2(J) = U2(J) + E(I) * SIN( QI*X(J) )
30    CONTINUE
C
      SIG = EZ * ( W1(J) - W2(J) ) / T
      TAU = GXZ * ( U1(J) - U2(J) ) / T
      F = (SIG*SIG) / (SIGU*SIGU) + (TAU*TAU) / (TAUU*TAUU)
C
      IF (J.EQ.1) WRITE(42,522) X(J), W1(J)
      IF (J.EQ.2) WRITE(42,520) X(J), SIG,TAU, F, W1(J)
50    CONTINUE
60    CONTINUE
      WRITE(*,530)
C
500  FORMAT(2X,
      * ' This information is also saved under a file called: ',A64,/,
      * 7X,'Half length of the beam is ',F10.3,/,
      * 7X,'Length of the Doubler is ',F10.3,/,
      * 7X,'Adhesive thickness is ',F12.5,/,
      * 7X,'Youngs Modulus of the Adhesive is ',F12.2,/,
      * 7X,'Shear Modulus of the Adhesive is ',F12.2,/,
      * 7X,'Ultimate Peel (Normal) stress is ',F12.2,/,
      * 7X,'Ultimate Shear Stress is ',F12.2,/,
      * 7X,'Averager Characteristic Length is ',F12.5,/,
      * 7X,'The beam is divided into',I4,' step(s)',/,
      * 7X,'The doubler is divided into',I4,' step(s)',/,
      * ' The step(s) for the beam are at:')
501  FORMAT( 3X,' X = ',F9.2,' ET = ',E12.6, ' EI = ', E12.6)
502  FORMAT(' While the doubler has steps at: ')
510  FORMAT(' X-location Normal Stress Shear Stress ',
      * ' Failure Function Z-Displace.')
520  FORMAT( 2X,F7.2,6X,F14.4,3X, F13.4, 3X, F14.7, 2X, F14.8)
522  FORMAT( 2X,F7.2,55X,F14.8)
530  FORMAT( 3X,' THIS IS THE END OF THE PROGRAM.')
550  FORMAT( 3X,' What name do you want the output listed under?')
551  FORMAT(/,3X,'For an applied load of ',F9.2,/,3X,
      * ' Failure Function over a(0) ',F12.5)
      STOP
      END
C
      FUNCTION PZ (NU,U)
      PZ = 4.0 * ATAN(1.0) * (2*NU -1)/ U
      RETURN
      END

```

5. Simplified Stiffener Analysis Method (SSAM & SSAMGEN)

```

C
C*****
C***** THE FOLLOWING IS THE COMMON.INC FILE FOR SSAM & SSAMGEN *****
C*****
C
C
C INCLUDE file for the SSAM Finite Element programs.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MAXEL,MAXNOD, LDA,MAXBC,MAXLDS,MAXDOF,NFN,NFN2
C
C This INCLUDE file contains all of the major variable declarations
C needed by the SSAM system programs. The file is then included

```



```

C in each subroutine during the compilation process, thus insuring
C that the variables are dimensioned to the same size everywhere.
C
C First, some basic dimensions are established and assigned to constant
C variables. Other needed dimensions are then calculated from these, and
C finally the arrays are declared using these dimensions. Therefore, if
C you need to change the size of the arrays, you need only to change
C one of the PARAMETER statements below and re-compile the program.
C
C Define Four Basic Array Dimensions:
C MAXEL = Maximum number of frame elements allowed
C LDA = (MAXIMUM BANDWIDTH + 1) / 2
C NPN = Number of degrees-of-freedom per frame element node
C NPN2 = Number of degrees-of-freedom per membrane element
C
      PARAMETER (MAXEL = 200 )
      PARAMETER (LDA = 15 )
      PARAMETER (NPN = 3 )
      PARAMETER (NPN2 = 8 )
C
C Now calculate the other array dimensions from these basic ones
C
      PARAMETER (MAXNOD = 2*(MAXEL+2) )
      PARAMETER (MAXBC = MAXNOD )
      PARAMETER (MAXLDS = MAXNOD )
      PARAMETER (MAXDOF = NPN * MAXNOD )
C
C Ready to declare the array sizes
C
      INTEGER BC
      DIMENSION S(LDA,MAXDOF), NBCON (MAXEL, 2), NPCON (MAXEL, 4),
      &EI (MAXEL), IBC (MAXBC, NPN+1), AXFRC (MAXEL), AXLAST (MAXEL),
      &BDISP (MAXBC*NPN), IFRC (MAXLDS), FORCE (MAXLDS, NPN), F (MAXDOF),
      &DISP (MAXDOF), IBCS (MAXDOF), BC (MAXBC*NPN), WORK (MAXDOF),
      &EA (MAXEL), XORD (MAXNOD), YORD (MAXNOD), NPTOB (MAXNOD), TEA (MAXEL),
      &PE (MAXEL), PG (MAXEL), PFWT (MAXEL), PILS (MAXEL), NTCN (MAXEL, 2),
      &NFLAG (MAXNOD)
C
C Finally, put everything in COMMON blocks to that all subroutines have
C access to all of the data.
C
      COMMON /IOCTRL/ ISCRN, IKYBD, IFILE, IOUT
      COMMON /MESH/ NBEAM, NPLATE, NUMBC, NUMLDS, NSTITCH, NDOF, MBAND,
      & IEL, NUMNP, NCON, NFRNP
      COMMON /INDATA/ NBCON, NPCON, EI, EA, IBC, BC, AXFRC, AXLAST, BDISP,
      & FORCE, IFRC, PE, PG, XORD, YORD, NPTOB, NTCN, TEA,
      & PFWT, PILS, NFLAG
      COMMON /WORKING/ S, F, DISP, IBCS, WORK
C
C*****
C***** THE FOLLOWING IS THE SSAM.FOR PROGRAM *****
C*****
C
      PROGRAM SSAM
C
C SSAM - the Simplified Stiffener Analysis Model
C
C This program uses a combination of frame, membrane and truss finite elements
C to model the pull-off problem. The skin and stiffener are modeled as frames
C and the interface is modeled as membrane elements connecting the two frames.
C Optional stitching between the skin and flange can be modeled using truss
C bar finite elements. This version includes the reduction in frame bending
C stiffness due to an axial compressive force. Since this force is not known
C in advance, the model must be solved iteratively until the force resultants
C converge.
C

```

```

C Written by: Steve Wanthal
C             McDonnell Aircraft Co.
C             St. Louis, MO
C             October, 1989
C
C This program was developed under MCAIR Structural Research IRAD 7-331
C
C             COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS
C             RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
C             DOUGLAS CORPORATION.
C
C Dimension variables. Most variables are defined in file COMMON.INC
C This method allows the array sizes to be adjusted by altering only one
C line of code in COMMON.INC rather than having to change the main program
C and all subroutines.
C
C             INCLUDE 'COMMON.INC'
C Other variables that will be needed in the main program only
C             DIMENSION XD(NPN),SO(NPN2,NPN2),SOPRIME(12,12)
C             INTEGER PNODES(4)
C             CHARACTER INFIL*30,OUTFIL*30,DESCR*80,TODAY*24
C             CHARACTER*33 READERR(9)
C             DATA READERR /'WHILE OPENING MODEL FILE
C             &             'WHILE READING MODEL DESCRIPTION
C             &             'WHILE READING CONTROL INFORMATION'
C             &             'WHILE READING FRAME ELEMENTS
C             &             'WHILE READING PLATE ELEMENTS
C             &             'WHILE READING NODAL COORDINATES
C             &             'WHILE READING BOUNDARY CONDITIONS'
C             &             'WHILE READING NODAL POINT LOADS
C             &             'WHILE READING TRUSS INFORMATION'
C             DATA PERCONV,PERLAST / 0.02, 1.E20 /
C
C             VERSION=1.10
C
C Initialize I/O unit numbers to be used. Note that some computers won't like
C you to use these unit numbers. ISCRN is for writing to the terminal screen
C IKYBD is for reading input from the keyboard, and the others are all used
C for reading and writing to disk files. You can set these to any unit
C numbers that your system will accept.
C
C             DATA ISCRN,IKYBD,IFILE,IOUT,IDISP,INORM,ISHEAR,IFAIL
C             & /6, 5, 1, 2, 3, 4, 7, 8/
C
C Get the input mesh data file name
C
C             WRITE(ISCRN,2225)VERSION
C             1000 WRITE(ISCRN,2230)
C             READ(IKYBD,'(A30)',ERR=1000)INFIL
C             IF(INFIL.EQ.'STOP'.OR.INFIL.EQ.'stop')STOP
C
C Open the input file and read the data. Refer to the user's
C manual for a detailed description of the required input file format
C and descriptions of the input variables.
C
C             LINE=1
C             OPEN(UNIT=IFILE,FILE=INFIL,STATUS='OLD',ERR=1020)
C             REWIND IFILE
C
C Mesh description line
C
C             LINE=LINE+1
C             READ(IFILE,'(A79)',ERR=1010,END=1010)DESCR
C             WRITE(ISCRN,'(/,1X,A79,/)' )DESCR

```

C Control information

```

      LINE=LINE+1
      READ (IFILE,2000,ERR=1010,END=1010) NBSK,NBFL,NPLATE,NUMBC,
&NUMLDS,NSTITCH

```

C Frame element definitions. Each frame element is defined by the
 C two nodes at each end, EI and EA. Keep track of the maximum node
 C difference for later calculation of the stiffness matrix bandwidth.
 C NFLAG = 1 ==> frame element node
 C NFLAG = 2 ==> membrane element node

```

      NBEAM=NBSK+NBFL
      NUMNP=0
      NDIFF=0
      LINE=LINE+1
      DO I=1,NBEAM
        READ (IFILE,2010,ERR=1010,END=1010) K, (NBCON(K,J),J=1,2),
&
          EI(K),EA(K)
        NDIFF=MAX(NDIFF,IABS(NBCON(K,1)-NBCON(K,2)))
        NUMNP=MAX(NUMNP,NBCON(K,1),NBCON(K,2))
        NFLAG(NBCON(K,1))=1
        NFLAG(NBCON(K,2))=1
      END DO

```

C Membrane element connectivity. For each element, read the 4 nodes which
 C define the element corners, E, G, flatwise tensile and shear strengths.

```

      LINE=LINE+1
      DO I=1,NPLATE
        READ (IFILE,2020,ERR=1010,END=1010) K, (NPCON(K,J),J=1,4),
&
          PE(K),PG(K),PFWT(K),PILS(K)
        NUMNP=MAX(NUMNP,NPCON(K,1),NPCON(K,2),NPCON(K,3),NPCON(K,4))
      END DO

```

C Read nodal point information. This includes the adjacent node (if any)
 C and the x- and y-coordinates for each node. The finite element mesh
 C must be set up so that each node which defines a membrane element is
 C near a corresponding node which defines a frame element. These two nodes are
 C then considered "adjacent." The idea is that the degrees of freedom
 C of any membrane element node can be written in terms of the degrees
 C of freedom of the adjacent frame element node.

```

      LINE=LINE+1
      DO I=1,NUMNP
        READ (IFILE,2025,END=1010,ERR=1010) K,NPTOB(K),XORD(K),YORD(K)
        IF (NPTOB(K).EQ.0.AND.NFLAG(K).NE.1) WRITE (ISCRN,2500) K
      END DO

```

C Check the frame nodes which are adjacent to the membrane nodes
 C for the maximum numerical difference between them. This maximum
 C difference is used to find the stiffness matrix bandwidth

```

      DO K=1,NPLATE
        M1=0
        M2=NUMNP
        DO I=1,4
          NFLAG(NPCON(K,I))=2
          N=NPTOB(NPCON(K,I))
          IF (N.EQ.0) N=NPCON(K,I)
          M1=MAX(M1,N)
          M2=MIN(M2,N)
        END DO
        NDIFF=MAX(NDIFF,M1-M2)
      END DO

```

C Read the boundary conditions. The degrees of freedom for any
C node which defines a frame element can be fixed at any value.
C Note that boundary conditions cannot be applied to the nodes
C which define the interface.

```

NCON=0
LINE=LINE+1
DO I=1,NUMBC
  READ (IFILE,2030,ERR=1010,END=1010) (IBC(I,J),J=1,NPN+1),
& (XD(J),J=1,NPN)
  IF (NFLAG(IBC(I,1)).NE.1) THEN
    WRITE (ISCRN,*) 'WARNING - BOUNDARY CONDITIONS APPLIED ON'
    WRITE (ISCRN,*) 'THE INTERFACE ARE BEING IGNORED'
  ELSE
    DO J=1,NPN
      IF (IBC(I,J+1).NE.0) THEN
        NCON=NCON+1
        BDISP(NCON)=XD(J)
        BC(NCON)=NPN*(IBC(I,1)-1)+J
      END IF
    END DO
  END IF
END DO

```

C Read the nodal point loads. Forces in the X- and Y- directions and/or
C a moment can be specified at any node which defines a frame element.
C Note that loads cannot be specified on nodes which define membrane elements

```

LINE=LINE+1
KK=0
DO I=1,NUMLDS
  KK=KK+1
  READ (IFILE,2040,ERR=1010,END=1010) IFRC(KK), (FORCE(KK,J),J=1,NPN)
  IF (NFLAG(IFRC(KK)).NE.1) THEN
    WRITE (ISCRN,*) 'WARNING - LOADS & MOMENTS APPLIED ON'
    WRITE (ISCRN,*) 'THE INTERFACE ARE BEING IGNORED'
  KK=KK-1
  END IF
END DO
NUMLDS=KK

```

C Read in truss bar connectivity. Note that truss elements cannot
C connect membrane element nodes.

```

LINE=LINE+1
KK=0
DO IEL=1,NSTITCH
  KK=KK+1
  READ (IFILE,2050,ERR=1010,END=1010) K, (NTCON(KK,I),I=1,2),
& TEA(KK)
  IF (NFLAG(NTCON(KK,1))+NFLAG(NTCON(KK,2)).NE.2) THEN
    WRITE (ISCRN,*) 'WARNING - STITCHES ATTACHED TO THE'
    WRITE (ISCRN,*) 'INTERFACE ARE BEING IGNORED'
    KK=KK-1
  END IF
  IF (KK.GT.0) THEN
    M=IABS(NTCON(KK,2)-NTCON(KK,1))
    IF (NDIFF.LT.M) NDIFF=M
  END IF
END DO
NSTITCH=KK

```

C Finished reading. Close the file and skip to the next step.

```

CLOSE (IFILE,STATUS='KEEP')

```

```

      GO TO 1030

C Jump here if an error occurred while reading the input data file.
1010 CONTINUE
      CLOSE (IFILE)
1020 CONTINUE
      WRITE (ISCRN,*) 'ERROR READING INPUT FILE:',INFIL
      WRITE (ISCRN,*) READERR(LINE)
      GO TO 1000

C
C Open up the output file and echo the input data
C
1030 WRITE (ISCRN,2231)
      READ (IKYBD,'(A30)',ERR=1030)OUTFIL
      OPEN (UNIT=IOUT,FILE=OUTFIL,STATUS='UNKNOWN')
      REWIND IOUT
      WRITE (IOUT,'(A80,/)')DESCR

C
C The following line uses the TODAY function which is VAX/VMS-dependent.
C It simply prints the current date to the output file and can be
C omitted if your computer does not have an equivalent routine.
C
C      WRITE (IOUT,*) 'Date: ',TODAY()
C
      WRITE (IOUT,2060)NBEAM
      WRITE (IOUT,2070)NPLATE
      WRITE (IOUT,2080)NCON
      WRITE (IOUT,2090)NUMLDS
      WRITE (IOUT,2100)
      WRITE (IOUT,2010) (I,NBCON(I,1),NBCON(I,2),EI(I),EA(I),
&I=1,NBEAM)
      WRITE (IOUT,2110)
      WRITE (IOUT,2020) (I,(NPCON(I,J),J=1,4),PE(I),PG(I),PFWT(I),PILS(I),
&I=1,NPLATE)
      WRITE (IOUT,2125)
      WRITE (IOUT,2025) (I,NPTOB(I),XORD(I),YORD(I),I=1,NUMNP)
      IF (NUMLDS.GE.1)WRITE (IOUT,2130)
      WRITE (IOUT,2041) (IFRC(I),(FORCE(I,J),J=1,NPN),I=1,NUMLDS)
      IF (NSTITCH.GE.1)WRITE (IOUT,2131)
      WRITE (IOUT,2050) (I,(NTCON(I,J),J=1,2),TEA(I),I=1,NSTITCH)

C
C Find stiffness matrix bandwidth and number of unrestrained DOF's.
C
      NDOF=0
      DO I=1,NUMNP
          IF (NFLAG(I).EQ.1.OR.(NFLAG(I).NE.1.AND.NPTOB(I).EQ.0))NMAX=I
          IF (NFLAG(I).EQ.1)THEN
              NDOF=NDOF+3
          ELSE IF (NPTOB(I).EQ.0)THEN
              NDOF=NDOF+2
          END IF
      END DO
      MBAND=NDIFF*NPN+NPN
      WRITE (ISCRN,2240)2*MBAND-1,NDOF-NCON

C
C Calculate size of stiffness matrix and make sure the arrays are
C dimensioned large enough to continue.
C
      IF (NPN*NUMNP.GT.MAXDOF)
&WRITE (ISCRN,*) 'ERROR - MAXDOF MUST BE AT LEAST',NPN*NUMNP
      IF (MBAND.GT.LDA)WRITE (ISCRN,*) 'ERROR - LDA MUST BE AT LEAST',MBAND
      IF (NPN*NUMNP.GT.MAXDOF.OR.MBAND.GT.LDA) STOP

C
C Find number of equations we must keep
C

```

```

      NDOF=NPN*NMAX
C
C Initialize file DISP.DAT
C This file has the undeformed and deformed shape of the panel
C in GNFTPLOT format (GNFTPLOT is a general plotting routine in
C MCAIR's Structural Analysis Library).
C
      CALL INITDISP (NBSK,NBFL,IDISP)
C
C Begin iterations here. The program iterates until the axial
C force resultants in the frame elements converge. Note that the
C stiffness of the frame elements changes with the axial force, hence
C the stiffness changes from one iteration to the next. Variable
C PERLAST is the maximum percent change from the previous iteration;
C this is kept to check if the solution is converging or diverging.
C It is possible that a compressive force will be induced that is so
C large that it destabilizes the stiffness matrix. If this happens
C we can't solve the problem. Sorry.
C
      ITER=0
      WRITE (ISCRN,2300)PERCONV
200  CONTINUE
      ITER=ITER+1
C
C Start by zeroing the global stiffness matrix, [S]
C Note that S is stored in band-symmetric storage mode.
C
      DO I=1,NDOF
        DO J=1,MBAND
          S(J,I)=0.0
        END DO
      END DO
C
C Build global stiffness matrix. Add the frame elements first.
C KBSK generates the element stiffness matrix, and ADDB adds it
C to the global stiffness matrix. The stiffness matrix for one
C element is temporarily stored in array [S0].
C
      DO IEL=1,NBEAM
C
        CALL KBSK(S0)
        CALL ADDB(S0)
C
      END DO
C
C Add the membrane elements next.
C KPLA generates the element stiffness matrix, [S0]. MODP applied the
C constraints to rewrite [S0] in terms of the frame elements DOF's,
C and ADDP adds the rewritten stiffness matrix, [S0'], to the global
C stiffness matrix.
C
      DO IEL=1,NPLATE
C
        CALL KPLA(S0)
        CALL MODP (PNODES,S0,SOPRIME)
        CALL ADDP (PNODES,SOPRIME)
C
      END DO
C
C Add boundary conditions to account for the membrane element nodes.
C We always ignore the slope DOF for these nodes, and if the
C U- & V-displacements were condensed out in routine MODP then we
C ignore those also.
C
      CALL CONDEC
C

```

```

C Add the truss elements which represent the stitching
C
      IF (NSTITCH.GT.0) CALL ADDK
C
C Set up the global force vector, F
C
      CALL FVECTOR
C
C Apply the specified displacement boundary conditions
C
      CALL BOUND
C
C Check if any loads or displacements were applied (if not, there's no
C point in going on further since all results will be zero)
C
      CALL CHKLDS (NOLOAD)
      IF (NOLOAD.NE.0) THEN
        WRITE (ISCRN,*) 'ERROR - NO LOADS OR NONZERO DISPLACEMENT'
        WRITE (ISCRN,*) '          BOUNDARY CONDITIONS WERE SPECIFIED'
        STOP
      END IF
C
C Decompose the matrix equations
C
      CALL DECOM (IERR)
C
C Check possible error conditions that may arise in DECOM. There are
C two ways this can happen: 1) if insufficient boundary conditions were
C specified, or 2) if the stiffness loss due to a compressive force in one
C or more elements is so severe that it destabilizes the stiffness equations.
C The first error will always show up in the first iteration (ie first run
C thru DECOM). The second possible error will always show up after the first
C iteration because the axial forces are always zero for the first iteration
C and therefore they cause no loss in stiffness.
C
      IF (IERR.NE.0.AND.ITER.EQ.1) THEN
        WRITE (ISCRN,*) 'ERROR - THERE ARE INSUFFICIENT BOUNDARY'
        WRITE (ISCRN,*) '          CONDITIONS IN YOUR MODEL TO SOLVE'
        WRITE (ISCRN,*) '          THE PROBLEM'
        STOP
      ELSE IF (IERR.NE.0) THEN
        WRITE (ISCRN,*) 'ERROR - THE CALCULATED AXIAL FORCES ARE'
        WRITE (ISCRN,*) '          SO LARGE THAT PART OF THE SKIN'
        WRITE (ISCRN,*) '          AND/OR STIFFENER HAS BUCKLED'
        STOP
      END IF
C
C Solve the equations
C
      CALL SOLV
C
C Find the axial force resultant in each element and check for convergence of
C the force. PERCENT is the maximum percent change in the force between this
C iteration and the last one.
C
      CALL NEWP (PERCENT)
C
      IF (ITER.NE.1) WRITE (ISCRN, ' (I5,F18.4) ') ITER, PERCENT
      IF (PERCENT.LT.PERCONV) GO TO 300
      IF (ITER.GE.15.AND.PERCENT.GT.PERLAST) THEN
        WRITE (ISCRN,*) 'WARNING - THE AXIAL FORCES DO NOT SEEM TO BE'
        WRITE (ISCRN,*) '          CONVERGING IN ONE OR MORE ELEMENTS.'
        WRITE (ISCRN,*) '          WE'LL STOP THE ITERATIONS HERE, BUT'
        WRITE (ISCRN,*) '          THE SOLUTION MAY NOT BE ACCURATE.'
        GO TO 300
      END IF

```

```

      PERLAST=PERCENT
C
C Need another iteration.
C
      GO TO 200
C
C Convergence has been achieved. Write out the solution.
C Start by calculating the displacements at the plate nodes from
C the displacements of the frame nodes based on the constraint equations.
C
300 CALL PLDISP
C
C Print nodal point displacements to the output file
C
      CALL PRDISP
C
C Print the deformed shape of the panel to the DISP.DAT file for
C use in the GNFTPLOT routine.
C
      CALL DISPSHAPE(NBSK,NBFL,IDISP)
C
C Calculate and print frame element force and moment resultants
C
      CALL PRFORCE
C
C Calculate and print the interface stress in each plate element.
C Print the stresses to both the output file and to GNFTPLOT format files.
C
      CALL PRSTRESS(INORM,ISHEAR,IFAIL)
C
C All Done
C
      STOP
C
C Format Statements
C
2000 FORMAT(6I4)
2010 FORMAT(3I4,2E12.5)
2020 FORMAT(5I4,4E12.5)
2025 FORMAT(2I4,2E12.5)
2030 FORMAT(4I4,3E12.5)
2040 FORMAT(I4,3E12.5)
2050 FORMAT(3I4,E12.5)
2041 FORMAT(I4,3E14.5)
2060 FORMAT(/,' Number of Frame Elements =',I4,/)
2070 FORMAT(' Number of Plate Elements =',I4,/)
2080 FORMAT(' Number of Restraints =',I4,/)
2090 FORMAT(' Number of Loaded Nodes =',I4)
2100 FORMAT(/,' Beam Elements:',/,
&' EL N1 N2 EI EA')
2110 FORMAT(/,' Plate Elements:',/,
&' EL N1 N2 N3 N4 E G FWT ILS')
2120 FORMAT(/,' Boundary Conditions:',
&/', ' Node X Y M Y-Disp X-Disp. Slope')
2125 FORMAT(/,' Node Con X-ord Y-ord')
2130 FORMAT(/,' Nodal Forces:',/,
&' Node X-Force Y-Force Moment')
2131 FORMAT(/,' Truss Bars used to Model Stitching:',/,
&' # N1 N2 E')
C23456789012345678901234567890123456789012345678901234567890
2225 FORMAT(/,1X,59('*'),/,8X,
&'SSAM - the Simplified Stiffener Analysis Model',/,8X,
&' Version',F5.2,/,1X,59('*'),/,/,5X,
&'COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS',/,5X,
&'RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL',/,5X,
&'DOUGLAS CORPORATION.',/,/)

```



```

2230 FORMAT('$Enter the input data file name( or STOP)> ')
2231 FORMAT('$Enter an output file name> ')
2240 FORMAT(/,' Stiffness Matrix Bandwidth = ',I4,/,
&      ' Number of Unrestrained DOF = ',I4)
2300 FORMAT(/,' The program iterates until the maximum change in',/,
&      ' axial force in the frame elements is less than',f5.2,' %',/,
&      ' Iteration      Max. % Change')
2500 FORMAT(' WARNING - Membrane node',I4,
&      ' is not connected to a frame node')

C
      END
C
C*****
C
      SUBROUTINE KBEAM(THETA,S0)
C
C Routine to build the stiffness matrix for a frame element
C without any nonlinear effects. (The nonlinear effect is the
C loss in stiffness due to an axial compressive force).
C The derivation of these stiffness terms can be found in
C "Finite Element Structural Analysis" by T.Y.Yang
C
      INCLUDE 'COMMON.INC'
      DIMENSION S0(NPN2,NPN2)
C
      XL=COS(THETA)
      XM=SIN(THETA)
      XL2=XL*XL
      XM2=XM*XM
      XLM=XM*XL
C
      XLEN=BLN(IEL)
      R=EA(IEL)/EI(IEL)
C
      A1=12.0/XLEN/XLEN*EI(IEL)/XLEN
      A2=6.0/XLEN*EI(IEL)/XLEN
      A3=2.0*EI(IEL)/XLEN
      A4=R*EI(IEL)/XLEN
C
      S0(1,1)=A4*XL2+A1*XM2
      S0(1,2)=(A4-A1)*XLM
      S0(1,3)=-A2*XM
      S0(1,4)=-A4*XL2-A1*XM2
      S0(1,5)=(-A4+A1)*XLM
      S0(1,6)=-A2*XM
      S0(2,2)=A4*XM2+A1*XL2
      S0(2,3)=A2*XL
      S0(2,4)=(-A4+A1)*XLM
      S0(2,5)=-A4*XM2-A1*XL2
      S0(2,6)=A2*XL
      S0(3,3)=2.0*A3
      S0(3,4)=A2*XM
      S0(3,5)=-A2*XL
      S0(3,6)=A3
      S0(4,4)=A4*XL2+A1*XM2
      S0(4,5)=(A4-A1)*XLM
      S0(4,6)=A2*XM
      S0(5,5)=A4*XM2+A1*XL2
      S0(5,6)=-A2*XL
      S0(6,6)=2.0*A3
C
C Make the matrix symmetric
C
      DO I=2,NPN*2
      DO J=1,I-1
        S0(I,J)=S0(J,I)

```

```

      END DO
      END DO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KBSK(S0)
C
C Routine to build the stiffness matrix for a frame element
C including the stiffness due to an axial force.
C
C Refer to "Finite Element Structural Analysis" by T.Y. Yang
C {1st Ed., sections 5.2 (linear part) & 8.6 (nonlinear part)} for
C a description of this finite element and its derivation.
C
      INCLUDE 'COMMON.INC'
      DIMENSION S0(NPN2,NPN2),TMP(NPN2,NPN2)
C
C Find orientation angle of the frame element
C
      PI=ACOS(-1.000000000000)
      DX=XORD(NBCON(IEL,2))-XORD(NBCON(IEL,1))
      DY=YORD(NBCON(IEL,2))-YORD(NBCON(IEL,1))
      IF(DX.EQ.0.0) THEN
        THETA=PI/2.0*DY/ABS(DY)
      ELSE
        THETA=ATAN(DY/DX)
      END IF
C
C Get the stiffness matrix without nonlinear effects.
C The nonlinear effect is the change in bending stiffness
C due to the presence of an axial force in the element.
C
      CALL KBEAM(THETA,S0)
C
C Add the incremental stiffness due to the axial compressive force, AXFRC
C
      IF(AXFRC(IEL).EQ.0.0) RETURN
      IF(AXFRC(IEL).GE.0.0) RETURN
C
      XL=COS(THETA)
      XM=SIN(THETA)
      XL2=XL*XL
      XM2=XM*XM
      XLM=XM*XL
C
      P=-AXFRC(IEL)/10.0
      XLEN=BLN(IEL)
      A1=12.0/XLEN*P
      A2=XLEN/3.0*P
C
      TMP(1,1)=A1*XM2
      TMP(1,2)=-A1*XLM
      TMP(1,3)=-XM*P
      TMP(1,4)=-A1*XM2
      TMP(1,5)=A1*XLM
      TMP(1,6)=-XM*P
      TMP(2,2)=A1*XL2
      TMP(2,3)=XL*P
      TMP(2,4)=A1*XLM
      TMP(2,5)=-A1*XL2
      TMP(2,6)=P*XL
      TMP(3,3)=4.0*A2
      TMP(3,4)=XM*P

```

```

      TMP (3,5)=-P*XL
      TMP (3,6)=-A2
      TMP (4,4)=A1*XM2
      TMP (4,5)=-A1*XLM
      TMP (4,6)=XM*P
      TMP (5,5)=A1*XL2
      TMP (5,6)=-P*XL
      TMP (6,6)=4.0*A2
C
      DO I=1,2*NPN
      DO J=I,2*NPN
        TMP (J,I)=TMP (I,J)
      END DO
      END DO
C
C Add the incremental stiffness to the linear portion
C
      DO I=1,NPN*2
      DO J=1,NPN*2
        S0 (I,J)=S0 (I,J)-TMP (I,J)
      END DO
      END DO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE ADDB(S0)
C
C Routine to add the stiffness matrix for a frame element to the
C Global Stiffness Matrix, S. Note that S is stored in a
C band-symmetric storage mode. The terms are mapped as
C follows: term (I,J) in the full storage mode of S is
C mapped to term (J-I+1,I) in the band-symmetric storage mode.
C Note that terms with J < I are ignored since the matrix
C is symmetric.
C
      INCLUDE 'COMMON.INC'
      DIMENSION S0 (NPN2,NPN2)
C
      N1=(NBCON (IEL,1)-1)*NPN
      N2=(NBCON (IEL,2)-1)*NPN
C
      DO 2000 I=1,NPN
      DO 2000 J=1,NPN
C
C Top left quadrant of S0
C
        IS = N1 + I
        JS = N1 + J
        IF (IS.GT.JS) GO TO 1000
        IROW = JS - IS + 1
        ICOL = IS
        S (IROW,ICOL) = S (IROW,ICOL) + S0 (I,J)
1000    CONTINUE
C
C Top right quadrant of S0
C
        IS = N1 + I
        JS = N2 + J
        IF (IS.GT.JS) GO TO 1010
        IROW = JS - IS + 1
        ICOL = IS
        S (IROW,ICOL) = S (IROW,ICOL) + S0 (I,J + NPN)
1010    CONTINUE

```

```

C
C Bottom left quadrant of S0
C
      IS = N2 + I
      JS = N1 + J
      IF (IS.GT.JS) GO TO 1020
      IROW = JS - IS + 1
      ICOL = IS
      S(IROW,ICOL) = S(IROW,ICOL) + S0(I + NPN,J)
1020  CONTINUE
C
C Bottom right quadrant of S0
C
      IS = N2 + I
      JS = N2 + J
      IF (IS.GT.JS) GO TO 2000
      IROW = JS - IS + 1
      ICOL = IS
      S(IROW,ICOL) = S(IROW,ICOL) + S0(I + NPN,J + NPN)
2000 CONTINUE
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KPLA(S0)
C
C Build stiffness matrix for a 4-node isoparametric quadrilateral
C plane strain element.
C
C Refer to "Finite Element Structural Analysis" by T.Y. Yang
C (1st Ed., section 11.2.1) for a description of this
C finite element and its derivation.
C
      INCLUDE 'COMMON.INC'
C
      REAL J(2,2)
      DIMENSION S0(NPN2,NPN2),A(3,8),T1(2,4),T2(4,2),C(3,3),
      &TMP1(3,8),TMP2(8,8),WEIGHT(2),ABSCISSA(2)
C
C Define Gauss quadrature weight and abscissa values, use N=2 integration
C which is exact for polynomials up to third-degree polynomials.
C
      NGAUSS=2
      WEIGHT(1)=1.0
      WEIGHT(2)=1.0
      ABSCISSA(1)=1.0/SQRT(3.0)
      ABSCISSA(2)=-1.0/SQRT(3.0)
C
C Set up stress-strain matrix for plane strain, [C]
C
      PNU=PE(IEL)/2.0/PG(IEL)-1.0
      E=PE(IEL)/(1.0+PNU)/(1.0-2.0*PNU)
      C(1,1)=E*(1.0-PNU)
      C(1,2)=E*PNU
      C(1,3)=0.0
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=0.0
      C(3,1)=0.0
      C(3,2)=0.0
      C(3,3)=E*(1.0-2.0*PNU)/2.0
C
C Set matrix of nodal coordinates
C

```

```

      DO I=1,4
        T2(I,1)=XORD(NPCON(IEL,I))
        T2(I,2)=YORD(NPCON(IEL,I))
      END DO
C
C Initialize S0 matrix
C
      DO I=1,8
        DO K=1,8
          S0(I,K)=0.0
        END DO
      END DO
C
C Start integration
C
      DO N1=1,NGAUSS
        XI= ABSCISSA(N1)
        DO N2=1,NGAUSS
          ETA=ABSCISSA(N2)
C
C Find Jacobian matrix, [J] = [T1][T2]
C
          T1(1,1)=.25*(-1.0+ETA)
          T1(1,2)=.25*(1.0-ETA)
          T1(1,3)=.25*(1.0+ETA)
          T1(1,4)=.25*(-1.0-ETA)
          T1(2,1)=.25*(-1.0+XI)
          T1(2,2)=.25*(-1.0-XI)
          T1(2,3)=.25*(1.0+XI)
          T1(2,4)=.25*(1.0-XI)
C
          DO I=1,2
            DO K=1,2
              J(I,K)=0.0
              DO L=1,4
                J(I,K)=J(I,K)+T1(I,L)*T2(L,K)
              END DO
            END DO
          END DO
C
C Find determinant of Jacobian
C
          DETJ=J(1,1)*J(2,2)-J(1,2)*J(2,1)
C
C Find [A] matrix
C
          DO I=1,4
            A(1,2*I-1)=(J(2,2)*T1(1,I)-J(1,2)*T1(2,I))/DETJ
            A(1,2*I)=0.0
            A(2,2*I-1)=0.0
            A(2,2*I)=(-J(2,1)*T1(1,I)+J(1,1)*T1(2,I))/DETJ
            A(3,2*I-1)=(-J(2,1)*T1(1,I)+J(1,1)*T1(2,I))/DETJ
            A(3,2*I)=(J(2,2)*T1(1,I)-J(1,2)*T1(2,I))/DETJ
          END DO
C
C Ready to perform the integration
C
          DO I=1,3
            DO K=1,8
              TMP1(I,K)=0.0
              DO L=1,3
                TMP1(I,K)=TMP1(I,K)+C(I,L)*A(L,K)
              END DO
            END DO
          END DO
C

```

```

      DO I=1,8
      DO K=1,8
        TMP2(I,K)=0.0
        DO L=1,3
          TMP2(I,K)=TMP2(I,K)+A(L,I)*TMP1(L,K)
        END DO
      END DO
      END DO
C
      DO I=1,8
      DO K=1,8
        S0(I,K)=S0(I,K)+WEIGHT(N1)*WEIGHT(N2)*DETJ*TMP2(I,K)
      END DO
      END DO
C
C Ready for next Gauss point
C
      END DO
      END DO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE MODP(PNODES,S0,SOPRIME)
C
C Routine to apply constraint equations to
C the stiffness matrix for a plate element.
C
C The general relationship is  $[S0'] = [E]^T [S0] [E]$ 
C where the matrix  $[E]$  relates the nodal degrees-of-freedom of
C the plate element to the nodal degrees-of-freedom of the frame
C elements which are adjacent to the plate element.
C
C This process, in effect, takes the plate stiffness matrix in terms of
C the plate's nodes and generates a new plate stiffness matrix which
C is in terms of the nodal points of the frame elements above and
C below the plate element. The plate's nodal points can then be
C ignored for the remainder of the solution process.
C
      INCLUDE 'COMMON.INC'
      DIMENSION S0(NPN2,NPN2),SOPRIME(12,12),E(8,12),TMP(8,12),
& TLEN(4),BETA(4)
      INTEGER PNODES(4)
C
C Find distance between frame and membrane nodes and the
C orientation of the frame element.
C
      DO I=1,4
        N=NPCON(IEL,I)
        PNODES(I)=N
        M=NPFOB(N)
        IF (M.NE.0) THEN
          TLEN(I)=SQRT((XORD(N)-XORD(M))**2+
& (YORD(N)-YORD(M))**2)
          PNODES(I)=NPFOB(N)
          DY=YORD(M)-YORD(N)
          DX=XORD(N)-XORD(M)
          IF (DY.NE.0.0) THEN
            BETA(I)=ATAN(DX/DY)
          ELSE
            BETA(I)=ACOS(-1.000000)/2.0*DX/ABS(DX)
          END IF
        ELSE
          TLEN(I)=0.0
        END IF
      END DO

```

```

      END IF
    END DO
  C
  C Calculate the [E] matrix
  C
    CALL EMATRIX(TLEN,BETA,E)
  C
  C Multiply [S0] by [E]
  C
    DO I=1,8
      DO J=1,12
        TMP(I,J)=0.0
        DO K=1,8
          TMP(I,J)=TMP(I,J)+S0(I,K)*E(K,J)
        END DO
      END DO
    END DO
  C
  C Pre-multiply by transpose([E])
  C
    DO I=1,12
      DO J=1,12
        SOPRIME(I,J)=0.0
        DO K=1,8
          SOPRIME(I,J)=SOPRIME(I,J)+E(K,I)*TMP(K,J)
        END DO
      END DO
    END DO
  C
    RETURN
  END
C
C*****
C
  SUBROUTINE ADDP(PNODES,SOPRIME)
C
  C Routine to add the stiffness matrix for a plate element to the
  C Global Stiffness Matrix, S. Note that S is stored in a
  C band-symmetric storage mode. The terms are mapped as
  C follows: term (I,J) in the full storage mode of S is
  C mapped to term (J-I+1,I) in the band-symmetric storage mode.
  C Note that terms with J < I are ignored since the matrix
  C is symmetric.
C
  INCLUDE 'COMMON.INC'
  INTEGER PNODES(4)
  DIMENSION SOPRIME(12,12)
C
  C Add one 3x3 section at a time
  C
    DO 1000 I=1,4
      NI=PNODES(I)
      IR=NPN*(NI-1)
      DO 1000 J=1,4
        NJ=PNODES(J)
        IC=NPN*(NJ-1)
        DO 1000 II=1,3
          IS=3*(I-1)+II
          IS=IR+II
          DO 1000 JJ=1,3
            JS=3*(J-1)+JJ
            JS=IC+JJ
            IF(JS.LT.IS)GO TO 1000
            IROW=JS-IS+1
            ICOL=IS
            S(IROW,ICOL)=S(IROW,ICOL)+SOPRIME(IS,JS)
          END DO
        END DO
      END DO
    END DO
  END

```

```

1000 .CONTINUE
C
      RETURN
      END
C
C*****
C
      SUBROUTINE FVECTOR
C
C Routine to build the global force vector, F
C
      INCLUDE 'COMMON.INC'
C
C Initialize F to zero
C
      DO I=1,NDOF
        F(I) = 0.0
      END DO
C
C Add the nonzero forces to the proper locations
C
      DO I=1,NUMLDS
        DO J=1,NPN
          F(NPN*(IFRC(I)-1)+J) = FORCE(I,J)
        END DO
      END DO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE PRDISP
C
C Routine to print the nodal displacements
C
      INCLUDE 'COMMON.INC'
C
      WRITE(IOUT,2000)
      DO I=1,NUMNP
        IF (NFLAG(I) .EQ. 1) THEN
          WRITE(IOUT,2010) I, (DISP (NPN*(I-1)+J), J=1, NPN)
        ELSE
          WRITE(IOUT,2010) I, (DISP (NPN*(I-1)+J), J=1, NPN-1)
        END IF
      END DO
C
      RETURN
C
2000 FORMAT(///,'NODAL POINT DISPLACEMENTS:',//,
      &      ' Node      U      V      DV/DX')
2010 FORMAT(I5,4F15.7)
      END
C
C*****
C
      SUBROUTINE PRSTRESS(INORM,ISHEAR,IFAIL)
C
C Routine to print the interface stresses and failure function
C
      INCLUDE 'COMMON.INC'
      REAL JAC(2,2)
      DIMENSION Q(8),A(3,8),T1(2,4),T2(4,2)
C
C Files NORM.DAT and SHEAR.DAT have the normal and shear stress
C distributions written in GNFTPLOT input file format (GNFLPLOT is

```



```

C a general plotting routine in MCAIR's Stuctural Analysis Library)
C The format of a GNFTPLOT file is as follows:
C
C      TITLE OF FIRST CURVE
C      NUMBER OF DATA POINTS, N
C      X(1)      Y(1)
C      X(2)      Y(2)
C      :
C      :
C      X(N)      Y(N)
C      TITLE OF SECOND CURVE
C      NUMBER OF DATA POINTS, N
C      X(1)      Y(1)
C      :
C      :
C      :
C      :
C
C      FAILMAX=0.0
C      WRITE(IOUT,2000)
C      OPEN(UNIT=INORM,FILE='NORM.DAT',STATUS='UNKNOWN')
C      REWIND INORM
C      OPEN(UNIT=ISHEAR,FILE='SHEAR.DAT',STATUS='UNKNOWN')
C      REWIND ISHEAR
C      OPEN(UNIT=IFAIL,FILE='FAIL.DAT',STATUS='UNKNOWN')
C      REWIND IFAIL
C      WRITE(INORM,*)'SSAM'
C      WRITE(ISHEAR,*)'SSAM'
C      WRITE(IFAIL,*)'SSAM'
C      WRITE(INORM,*)NPLATE
C      WRITE(ISHEAR,*)NPLATE
C      WRITE(IFAIL,*)NPLATE
C
C      DO IEL=1,NPLATE
C
C          N1=NPCON(IEI,1)
C          N2=NPCON(IEI,2)
C          N3=NPCON(IEI,3)
C          N4=NPCON(IEI,4)
C          XA=(XORD(N1)+XORD(N2)+XORD(N3)+XORD(N4))/4.0
C
C          Q(1)=DISP(NPN*N1-2)
C          Q(2)=DISP(NPN*N1-1)
C          Q(3)=DISP(NPN*N2-2)
C          Q(4)=DISP(NPN*N2-1)
C          Q(5)=DISP(NPN*N3-2)
C          Q(6)=DISP(NPN*N3-1)
C          Q(7)=DISP(NPN*N4-2)
C          Q(8)=DISP(NPN*N4-1)
C
C          DO I=1,4
C              T2(I,1)=XORD(NPCON(IEI,I))
C              T2(I,2)=YORD(NPCON(IEI,I))
C          END DO
C
C          XI=0.0
C          ETA=0.0
C
C      Find Jacobian matrix, [J]
C
C          T1(1,1)=.25*(-1.0+ETA)
C          T1(1,2)=.25*(1.0-ETA)
C          T1(1,3)=.25*(1.0+ETA)
C          T1(1,4)=.25*(-1.0-ETA)
C          T1(2,1)=.25*(-1.0+XI)

```

```

      T1(2,2)=.25*(-1.0-XI)
      T1(2,3)=.25*(1.0+XI)
      T1(2,4)=.25*(1.0-XI)
C
      DO I=1,2
      DO K=1,2
        JAC(I,K)=0.0
        DO L=1,4
          JAC(I,K)=JAC(I,K)+T1(I,L)*T2(L,K)
        END DO
      END DO
      END DO
C
C Find determinant of Jacobian
C
      DETJ=JAC(1,1)*JAC(2,2)-JAC(1,2)*JAC(2,1)
C
C Find [A] matrix
C
      DO I=1,4
        A(1,2*I-1)=(JAC(2,2)*T1(1,I)-JAC(1,2)*T1(2,I))/DETJ
        A(1,2*I)=0.0
        A(2,2*I-1)=0.0
        A(2,2*I)=(-JAC(2,1)*T1(1,I)+JAC(1,1)*T1(2,I))/DETJ
        A(3,2*I-1)=(-JAC(2,1)*T1(1,I)+JAC(1,1)*T1(2,I))/DETJ
        A(3,2*I)=(JAC(2,2)*T1(1,I)-JAC(1,2)*T1(2,I))/DETJ
      END DO
C
C Find the strains
C
      EPSX=0.0
      EPSY=0.0
      EPSXY=0.0
      DO II=1,8
        EPSX=EPSX + A(1,II)*Q(II)
        EPSY=EPSY + A(2,II)*Q(II)
        EPSXY=EPSXY+ A(3,II)*Q(II)
      END DO
C
C Find the stress
C
      PNU=PE(IEL)/PG(IEL)/2.0-1.0
      PP=PE(IEL)/(1.0+PNU)/(1.0-2.0*PNU)
      SIGY=PP*(PNU*EPSX+(1.0-PNU)*EPSY)
      SIGXY=PP*(1.0-2.0*PNU)*EPSXY/2.0
C
C Find the failure funtion at this stress state
C
      IF(SIGY.GE.0.0) THEN
        SY=SIGY
      ELSE
        SY=0.0
      END IF
      FAILURE=(SY/PFWT(IEL))**2+(SIGXY/PILS(IEL))**2
C
C Print the results
C
      WRITE(IOUT,2010)XA,SIGY,SIGXY,FAILURE
      WRITE(INORM,*)XA,SIGY
      WRITE(ISHEAR,*)XA,SIGXY
      WRITE(IFAIL,*)XA,FAILURE
      STOT=STOT+SIGXY*PLEN(IEL)
      YTOT=YTOT+SIGY*PLEN(IEL)
      IF(FAILURE.GT.FAILMAX) FAILMAX=FAILURE
      END DO
      WRITE(ISCRN,*)' '

```

```

WRITE(ISCRN,*)'Summary of interface loading:'
WRITE(ISCRN,*)'Total normal force is approximately = ',YTOT
WRITE(ISCRN,*)'Total shear force is approximately = ',STOT
WRITE(ISCRN,*)'Maximum failure function = ',FAILMAX
C
CLOSE(INORM)
CLOSE(ISHEAR)
CLOSE(IFAIL)
RETURN
C23456789012345678901234567890123456789012345678901234567890123456789012
2000 FORMAT(// ' Interface Stresses:',//,
& ' X NORMAL SHEAR FAILURE')
2010 FORMAT(4F15.5)
C
END
C
C*****
C
SUBROUTINE BOUND
C
C ROUTINE TO APPLY DISPLACEMENT BOUNDARY CONDITIONS
C THANKS TO R. M. PIDAPARTI FOR SUPPLYING THIS ROUTINE.
C
INCLUDE 'COMMON.ING'
C
C INITIALIZE DISPLACEMENT ARRAY, DISP, AND BC FLAG ARRAY, IBCS.
C
DO 100 I=1,NDOF
IBC(I)=0
100 DISP(I)=0.0
C
C PUT PRESCRIBED BC'S INTO THE DISPLACEMENT VECTOR AND SET BC FLAG VECTOR
C
NON0=0
DO 110 I=1,NCON
J=BC(I)
DISP(J)=BDISP(I)
IBC(J)=1
110 IF(BDISP(I).NE.0.0)NON0=1
C
C IF ALL BC'S ARE ZERO, WE CAN STOP HERE
C
IF(NON0.NE.0)GO TO 120
GO TO 4
120 CONTINUE
C
C HAVE ONE OR MORE NON-ZERO BC'S -- MUST GO THROUGH THE LOOPS
C TO MODIFY THE STIFFNESS MATRIX
C
DO 1 I=1,NDOF
DO 2 J=1,NCON
IF(BDISP(J).EQ.0.0)GO TO 2
K=BC(J)
IF(K.GE.I) JJ=K-I+1
IF(K.LT.I) JJ=I-K+1
IF(JJ.GT.MBAND) GO TO 2
IF(K.GE.I) F(I)=F(I)-S(JJ,I)*BDISP(J)
IF(K.LT.I) F(I)=F(I)-S(JJ,K)*BDISP(J)
2 CONTINUE
1 CONTINUE
4 CONTINUE
RETURN
END
C
C*****

```

```

C      SUBROUTINE DECOM(IERR)
C
C      SUBROUTINE TO DECOMPOSE THE STIFFNESS MATRIX
C      THANKS TO R. M. PIDAPARTI FOR SUPPLYING THIS ROUTINE.
C
C      INCLUDE 'COMMON.INC'
C      INTEGER NUMDOF, IBAND
C      NUMDOF=NDOF
C      IBAND=MBAND
C
C      IERR=0
C      N=IBAND
C      NN=NUMDOF-IBAND+1
C      KK=0
C
C      DO 11 I=1,NUMDOF
C      IF(IBCS(I).EQ.1) GO TO 11
C      KK=KK+1
C      IF(I.LE.NN) GO TO 2
C      N=N-1
2      DO 1 J=1,N
C      ID1=I+J-1
C      IF(ID1.GT.NUMDOF) GO TO 11
C      IF(BCS(ID1).EQ.1) GO TO 1
C      SUMX=0.
C      IF(KK.EQ.1) GO TO 4
C      L1=I-1
C      L2=L1+1
C      DO 5 L=1,L1
C      L2=L2-1
C      JJ=J+L
C      IF(JJ.GT.IBAND) GO TO 4
C      II=L+1
C      ID1=II+L2-1
C      ID2=JJ+L2-1
C      IF(ID1.GT.NUMDOF.OR.ID2.GT.NUMDOF.OR.L2.GT.NUMDOF) THEN
C      WRITE(ISCRN,*) 'DECOM--LINE 2 ID1,ID2,L2=',ID1,ID2,L2
C      GO TO 5
C      END IF
C      KO=BCS(ID1)+BCS(ID2)+BCS(L2)
C      IF(KO.NE.0) GO TO 5
C      SUMX=SUMX+S(II,L2)*S(JJ,L2)
5      CONTINUE
4      IF(J.NE.1) GO TO 3
C      UCK=S(1,I)-SUMX
C      IF(UCK.LE.1.E-10) THEN
C      WRITE(ISCRN,*) 'ERROR - MATRIX DECOMPOSITION FAILED, UCK =',UCK
C      IERR=-1
C      RETURN
C      END IF
C      S(1,I)=SQRT(UCK)
C      GO TO 1
3      CONTINUE
C      S(J,I)=(S(J,I)-SUMX)/S(1,I)
1      CONTINUE
11     CONTINUE
C      RETURN
C      END
C
C*****
C
C      SUBROUTINE SOLV
C
C      SUBROUTINE TO SOLVE THE SYSTEM OF EQUATIONS
C      THANKS TO R. M. PIDAPARTI FOR SUPPLYING THIS ROUTINE.

```

```

C      INCLUDE 'COMMON.INC'
C
      IN=0
      IN=IN+1
5      IF (IBCS(IN).EQ.1) GO TO 5
      WORK(IN)=F(IN)/S(1,IN)
      NN=IN+1
      DO 3 I=NN,NDOF
      IF (IBCS(I).EQ.1) GO TO 3
      I1=I-1
      SUMX=0.
      N=1
      II=I
      IF (I.LT.MBAND) GO TO 2
      N=I-MBAND+1
      II=MBAND
2      DO 4 J=N,I1
      IF (IBCS(J).EQ.1) GO TO 6
      SUMX=SUMX+WORK(J)*S(II,J)
6      II=II-1
4      CONTINUE
      WORK(I)=(F(I)-SUMX)/S(1,I)
3      CONTINUE
C
      IN=NDOF+1
1      IN=IN-1
      IF (IBCS(IN).EQ.1) GO TO 1
      DISP(IN)=WORK(IN)/S(1,IN)
      NN=IN-1
      DO 12 I=1,NN
      JJ=IN-I
      IF (IBCS(JJ).EQ.1) GO TO 12
      SUMX=0.
      II=1
      N=IN
      J1=JJ+1
      IF (I.LT.MBAND) GO TO 15
      N=MBAND+JJ-1
15      DO 14 J=J1,N
      II=II+1
      IF (IBCS(J).EQ.1) GO TO 14
      SUMX=SUMX+DISP(J)*S(II,JJ)
14      CONTINUE
      DISP(JJ)=(WORK(JJ)-SUMX)/S(1,JJ)
12      CONTINUE
      RETURN
      END
C
C*****
C
      INTEGER FUNCTION MAXI(N1,N2,N3,N4)
C
C      Find maximum of four integer numbers
C
      M=N1
      IF (N2.GT.M) M=N2
      IF (N3.GT.M) M=N3
      IF (N4.GT.M) M=N4
      MAXI=M
      RETURN
      END
C
C*****
C
      INTEGER FUNCTION MINI(N1,N2,N3,N4)

```

```

C
C Find minimum of four integer numbers
C
      M=N1
      IF (N2.LT.M) M=N2
      IF (N3.LT.M) M=N3
      IF (N4.LT.M) M=N4
      MINI=M
      RETURN
      END

C
C*****
C
      REAL*8 FUNCTION BLEN(I)
C find length of frame element "I"
C
      INCLUDE 'COMMON.INC'
C
      DX=XORD(NBCON(I,1))-XORD(NBCON(I,2))
      DY=YORD(NBCON(I,1))-YORD(NBCON(I,2))
      BLEN=SQRT(DX*DX+DY*DY)
C
      RETURN
      END

C
C*****
C
      REAL*8 FUNCTION PLEN(I)
C Find length of plate element "I"
C
      INCLUDE 'COMMON.INC'
C
      PLEN=ABS(XORD(NPCON(I,2))-XORD(NPCON(I,1)))
C
      RETURN
      END

C
C*****
C
      SUBROUTINE DISPSHAPE(NBSK,NBFL,IDISP)
C
C Routine to write the deformed shape of the
C panel to file DISP.DAT. DISP.DAT is written in
C the required format for use with the GNFTPLOT plotting routine.
C The format of a GNFTPLOT file is as follows:
C
C      TITLE OF FIRST CURVE
C      N {N = NUMBER OF (X,Y) DATA POINTS IN THE CURVE}
C      X(1)      Y(1)
C      X(2)      Y(2)
C      :         :
C      :         :
C      X(N)      Y(N)
C      TITLE OF SECOND CURVE
C      NUMBER OF DATA POINTS, N
C      X(1)      Y(1)
C      :         :
C      :         :
C      :         :
C      :         :
C
C      INCLUDE 'COMMON.INC'
C

```

```

C Find a displacement magnification factor
C to make the GNFTPLOT graph look better.
C
  N1=NBCON (NBSK+NBFL,2)
  N2=NBCON (NBSK,2)
  DIST=0.5*ABS (YORD (N1) -YORD (N2))
  DYMAX=0.0
  DO I=1,NUMNP
    DD=ABS (DISP (NPN*(I-1)+2))
    IF (DD.GT.DYMAX) THEN
      DYMAX=DD
      IDYMAX=I
    END IF
  END DO
  IF (DYMAX.NE.0.0) THEN
    FACTOR=DIST/DYMAX
  ELSE
    WRITE (ISCRN,*) 'WARNING - No vertical displacements anywhere,'
    WRITE (ISCRN,*) 'so no good displacement magnification'
    WRITE (ISCRN,*)
    & WRITE (ISCRN,*) 'factor, MAGN, was found for file DISP.DAT'
    & WRITE (ISCRN,*) ' (MAGN=1.0 will be used)'
    FACTOR=1.0
  END IF
C
C Stiffener
C
  WRITE (IDISP, ' (A13,F10.3)') ' SSAM, MAGN.=' ,FACTOR
  WRITE (IDISP,*) NBFL+2
  NODE=NBCON (NBSK+1,1)
  WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
&YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
  DO I=NBSK+1,NBSK+NBFL
    NODE=NBCON (I,2)
    WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
& YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
  END DO
  NODE=NBCON (NBSK+NBFL,1)
  WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
&YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
C
C Skin
C
  WRITE (IDISP, ' (A13,F10.3)') ' SSAM, MAGN.=' ,FACTOR
  WRITE (IDISP,*) NBSK+2
  NODE=NBCON (1,1)
  WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
&YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
  DO I=1,NBSK
    NODE=NBCON (I,2)
    WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
& YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
  END DO
  NODE=NBCON (NBSK,1)
  WRITE (IDISP,*) XORD (NODE)+FACTOR*DISP (3*(NODE-1)+1),
&YORD (NODE)+FACTOR*DISP (3*(NODE-1)+2)
C
  CLOSE (IDISP)
  RETURN
  END
C
C*****
C
  SUBROUTINE PRFORCE
C
C Calculate and print force resultants for frame elements

```

```

C
  INCLUDE 'COMMON.INC'
  DIMENSION S0 (NPN2,NPN2), Q (NPN2), YY (NPN2)
C
  WRITE(IOUT,1000)
C
  DO IEL=1,NBEAM
C
C Find the forces in the global axes
C
    CALL KBSK(S0)
C
    NODE1=NBCON (IEL,1)
    NODE2=NBCON (IEL,2)
C
    DO I=1,NPN
      Q(I)=DISP (NPN*(NODE1-1)+I)
      Q(I+NPN)=DISP (NPN*(NODE2-1)+I)
    END DO
C
    DO I=1,NPN*2
      YY(I)=0.0
      DO J=1,NPN*2
        YY(I)=YY(I)+S0(I,J)*Q(J)
      END DO
    END DO
C
C Find the forces and moment at the center of the element (ie x=L/2)
C
    PI=ACOS(-1.000000000000)
    DX=XORD(NBCON (IEL,2))-XORD(NBCON (IEL,1))
    DY=YORD(NBCON (IEL,2))-YORD(NBCON (IEL,1))
    IF (DX.EQ.0.0) THEN
      THETA=PI/2.0*DY/ABS(DY)
    ELSE
      THETA=ATAN(DY/DX)
    END IF
    AXF=-YY(1)
    SHF=-YY(2)
    XMOM=SHF*BLEN(IEL)/2.0*COS(THETA)
    & -AXF*BLEN(IEL)/2.0*SIN(THETA)-YY(3)
    WRITE(IOUT,1010) IEL,AXF,SHF,XMOM
C
  END DO
C
  1000 FORMAT(//,' ELEM          X-FORCE          Y-FORCE          MOMENT')
  1010 FORMAT(I5,3F15.5)
C
  RETURN
  END
C
C*****
C
  SUBROUTINE NEWP (PERCENT)
C
C Calculate axial force in each element and keep
C track of the maximum percent change from the
C forces calculated in the previous iteration
C
  INCLUDE 'COMMON.INC'
  DIMENSION S0 (NPN2,NPN2), Q (NPN2)
C
  PERCENT=0.0
  DO IEL=1,NBEAM
C
C Find the axial force in global coordinates

```



```

C Use {F} = {S0}{Q}
C
      CALL KBSK(S0)
C
      NODE1=NBCON(IEL,1)
      NODE2=NBCON(IEL,2)
      DO I=1,NPN
        Q(I)=DISP(NPN*(NODE1-1)+I)
        Q(I+NPN)=DISP(NPN*(NODE2-1)+I)
      END DO
C
      X1=0.0
      Y1=0.0
      DO J=1,NPN*2
        X1=X1+S0(1,J)*Q(J)
        Y1=Y1+S0(2,J)*Q(J)
      END DO
C
C Transform forces to the local axes to get the axial component
C
      PI=ACOS(-1.000000000000)
      DX=XORD(NBCON(IEL,2))-XORD(NBCON(IEL,1))
      DY=YORD(NBCON(IEL,2))-YORD(NBCON(IEL,1))
      IF (DX.EQ.0.0) THEN
        THETA=PI/2.0*DY/ABS(DY)
      ELSE
        THETA=ATAN(DY/DX)
      END IF
      AXFRC(IEL)=- (Y1*SIN(THETA)+X1*COS(THETA))
C
      END DO
C
C Find the largest axial force
C
      AXMAX=0.0
      DO IEL=1,NBEAM
        IF (ABS(AXFRC(IEL)).GT.AXMAX) AXMAX=ABS(AXFRC(IEL))
      END DO
C
C Find the % change from the last iteration, but ignore elements where
C the axial force is less than 1% of the max force. This is done in order
C to handle cases where the stiffener does not have any loads or boundary
C conditions applied on it, so the axial forces are almost zero. The effect
C of these small forces on the overall solution is negligible, so there is
C no reason to include them in the convergence test. Besides, they can
C sometimes fool the convergence test into thinking that the solution
C has not converged when in fact it has.
C
      AXMAX=AXMAX/100.0
      DO IEL=1,NBEAM
        IF (ABS(AXFRC(IEL)).GT.AXMAX) THEN
          DELTA=AXFRC(IEL)-AXLAST(IEL)
          IF (AXLAST(IEL).NE.0.0) THEN
            P=ABS(DELTA/AXLAST(IEL))*100.0
          ELSE IF (DELTA.EQ.0.0) THEN
            P=0.0
          ELSE
            P=100000.
          END IF
          IF (P.GT.PERCENT) PERCENT=P
          AXLAST(IEL)=AXFRC(IEL)
        END IF
      END DO
C
      RETURN
      END

```

```

C
C*****
C
      SUBROUTINE PLDISP
C
C Calculate the displacements of the plate nodes using the
C displacement solution of the frame nodes. The same constraint
C equations which were used in MODP are used here.
C
      INCLUDE 'COMMON.INC'
      DIMENSION E(8,12),Q(12),QPRIME(8),TLEN(4),BETA(4)
C
      DO IEL=1,NPLATE
C
C Find distance between adjacent frame and membrane nodes
C
      DO I=1,4
        N=NPCON(IEL,I)
        M=NPTOB(N)
        IF (M.NE.0) THEN
          TLEN(I)=SQRT((XORD(N)-XORD(M))**2+
& (YORD(N)-YORD(M))**2)
          DY=YORD(M)-YORD(N)
          DX=XORD(N)-XORD(M)
          IF (DY.NE.0.0) THEN
            BETA(I)=ATAN(DX/DY)
          ELSE
            BETA(I)=ACOS(-1.000000)/2.0*DX/ABS(DX)
          END IF
        ELSE
          TLEN(I)=0.0
        END IF
      END DO
C
C Calculate the [E] matrix of constraint equations
C
      CALL EMATRIX(TLEN,BETA,E)
C
C Find the {q} vector of frame element displacements
C
      DO I=1,4
        NODE=NPTOB(NPCON(IEL,I))
        IF (NODE.EQ.0) NODE=NPCON(IEL,I)
        DO J=1,NPN
          K=NPN*(NODE-1)+J
          Q(NPN*(I-1)+J)=DISP(K)
        END DO
      END DO
C
C Multiply {Q} by [E] to get {Q'}, the membrane element displacements
C
      DO I=1,8
        QPRIME(I)=0.0
        DO K=1,12
          QPRIME(I)=QPRIME(I)+E(I,K)*Q(K)
        END DO
      END DO
C
C Move plate displacements back into DISP array in the proper locations
C
      DO I=1,4
        NODE=NPCON(IEL,I)
        DO J=1,2
          DISP(NPN*(NODE-1)+J)=QPRIME(2*(I-1)+J)
        END DO
      END DO

```

```

C
C Next element
C
      END DO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE EMATRIX(TLEN,BETA,E)
C
C Routine to find the [E] matrix which relates the degrees of freedom
C of the plate elements to those of the two frame elements which are
C directly above and below the plate element.
C
      INCLUDE 'COMMON.INC'
      DIMENSION TLEN(4),BETA(4),E(8,12)
C
C Initialize the [E] matrix
C
      DO I=1,8
        DO J=1,12
          E(I,J)=0.0
        END DO
      END DO
C
C Calculate the [E] matrix. The general relation is {q} = [E] {q'}
C where {q} are the degrees of freedom of the two frame elements directly
C above and below the plate element, and {q'} are the degrees of freedom
C of the plate element.
C
      E(1,1)=1.0
      E(1,3)=-TLEN(1)*COS(BETA(1))
      E(2,2)=1.0
      E(2,3)=-TLEN(1)*SIN(BETA(1))
      E(3,4)=1.0
      E(3,6)=-TLEN(2)*COS(BETA(2))
      E(4,5)=1.0
      E(4,6)=-TLEN(2)*SIN(BETA(2))
      E(5,7)=1.0
      E(5,9)=TLEN(3)*COS(BETA(3))
      E(6,8)=1.0
      E(6,9)=TLEN(3)*SIN(BETA(3))
      E(7,10)=1.0
      E(7,12)=TLEN(4)*COS(BETA(4))
      E(8,11)=1.0
      E(8,12)=TLEN(4)*SIN(BETA(4))
C
      RETURN
      END
C
C*****
C
      SUBROUTINE ADDK
C
C Routine to add truss elements to the global stiffness matrix
C to simulate the effect of stitching in the skin/stiffener
C
      INCLUDE 'COMMON.INC'
      DIMENSION S0(NPN2,NPN2)
C
      DO I=1,NSTITCH
C Find the truss bar orientation

```

```

      N1=NTCON(I,1)
      N2=NTCON(I,2)

      DY=YORD(N2)-YORD(N1)
      DX=XORD(N2)-XORD(N1)

      IF (DX.NE.0.0) THEN
        ANGLE=ATAN(DY/DX)
      ELSE IF (DY.NE.0.0) THEN
        ANGLE=ACOS(-1.00000)/2.0*DY/ABS(DY)
      ELSE
        WRITE(ISCRN,*)'WARNING - ZERO LENGTH TRUSS ELEMENT IGNORED'
        GO TO 10
      END IF

```

C Build the truss bar stiffness matrix

```

      EAL=TEA(I)/SQRT(DX*DX+DY*DY)

      XL=COS(ANGLE)
      XL2=XL*XL
      XM=SIN(ANGLE)
      XM2=XM*XM
      XLM=XL*XM

      S0(1,1)=EAL*XL2
      S0(1,2)=EAL*XLM
      S0(1,3)=-EAL*XL2
      S0(1,4)=-EAL*XLM
      S0(2,1)=EAL*XLM
      S0(2,2)=EAL*XM2
      S0(2,3)=-EAL*XLM
      S0(2,4)=-EAL*XM2
      S0(3,1)=-EAL*XL2
      S0(3,2)=-EAL*XLM
      S0(3,3)=EAL*XL2
      S0(3,4)=EAL*XLM
      S0(4,1)=-EAL*XLM
      S0(4,2)=-EAL*XM2
      S0(4,3)=EAL*XLM
      S0(4,4)=EAL*XM2

```

C Add the truss stiffness to the global stiffness matrix

```

      DO II=1,2
      DO JJ=1,2

        IROW=(N1-1)*NPN+II
        ICOL=(N1-1)*NPN+JJ
        K=ICOL-IROW+1
        IF (ICOL.GE.IROW) S(K,IROW)=S(K,IROW)+S0(II,JJ)

        IROW=(N1-1)*NPN+II
        ICOL=(N2-1)*NPN+JJ
        K=ICOL-IROW+1
        IF (ICOL.GE.IROW) S(K,IROW)=S(K,IROW)+S0(II,JJ+2)

        IROW=(N2-1)*NPN+II
        ICOL=(N1-1)*NPN+JJ
        K=ICOL-IROW+1
        IF (ICOL.GE.IROW) S(K,IROW)=S(K,IROW)+S0(II+2,JJ)

        IROW=(N2-1)*NPN+II
        ICOL=(N2-1)*NPN+JJ
        K=ICOL-IROW+1

```

```

      IF (ICOL.GE.IROW) S (K, IROW) = S (K, IROW) + S0 (II+2, JJ+2)

      END DO
      END DO

10    CONTINUE
      END DO

      RETURN
      END

C
C*****
C
      SUBROUTINE INITDISP (NBSK, NBFL, IDISP)
C
C Routine to initialize the DISP.DAT file which contains
C the undeformed and deformed shapes of the panel in GNFTPLOT format.
C This routine writes out the undeformed information only. The rest
C is written in routine DISPSHAPE.
C
      INCLUDE 'COMMON.INC'
C
      OPEN (UNIT=IDISP, FILE='DISP.DAT', STATUS='UNKNOWN')
      REWIND IDISP
C
      WRITE (IDISP, *) 'STIFFENER NEUTRAL AXIS'
      WRITE (IDISP, *) NBFL+2
      WRITE (IDISP, *) XORD (NBCON (NBSK+1, 1)), YORD (NBCON (NBSK+1, 1))
      DO I=NBSK+1, NBSK+NBFL
        WRITE (IDISP, *) XORD (NBCON (I, 2)), YORD (NBCON (I, 2))
      END DO
      WRITE (IDISP, *) XORD (NBCON (NBSK+NBFL, 1)), YORD (NBCON (NBSK+NBFL, 1))
C
      WRITE (IDISP, *) 'SKIN NEUTRAL AXIS'
      WRITE (IDISP, *) NBSK+2
      WRITE (IDISP, *) XORD (NBCON (1, 1)), YORD (NBCON (1, 1))
      DO I=1, NBSK
        WRITE (IDISP, *) XORD (NBCON (I, 2)), YORD (NBCON (I, 2))
      END DO
      WRITE (IDISP, *) XORD (NBCON (NBSK, 1)), YORD (NBCON (NBSK, 1))
C
      WRITE (IDISP, *) 'BOTTOM OF INTERFACE'
      WRITE (IDISP, *) NPLATE+2
      WRITE (IDISP, *) XORD (NPCON (1, 2)), YORD (NPCON (1, 2))
      DO I=1, NPLATE
        WRITE (IDISP, *) XORD (NPCON (I, 1)), YORD (NPCON (I, 1))
      END DO
      WRITE (IDISP, *) XORD (NPCON (NPLATE, 2)), YORD (NPCON (NPLATE, 2))
C
      WRITE (IDISP, *) 'TOP OF INTERFACE'
      WRITE (IDISP, *) NPLATE+2
      WRITE (IDISP, *) XORD (NPCON (1, 3)), YORD (NPCON (1, 3))
      DO I=1, NPLATE
        WRITE (IDISP, *) XORD (NPCON (I, 4)), YORD (NPCON (I, 4))
      END DO
      WRITE (IDISP, *) XORD (NPCON (NPLATE, 3)), YORD (NPCON (NPLATE, 3))

      RETURN
      END

C
C*****
C
      SUBROUTINE CONDBC
C
C Routine to add boundary conditions for the membrane element nodes.
C If the node was condensed out using routine MODP, then add BCs for

```

```

C all three degrees of freedom.  If not, then add a BC for the slope only.
C
C   INCLUDE 'COMMON.INC'
C
C   DIMENSION IDONE (MAXNOD)
C
C   DO IEL=1,NPLATE
C
C     DO I=1,4
C       N=NPCON (IEL,I)
C       IF (IDONE (N) .NE. -1) THEN
C
C         IF (NPJOB (N) .NE. 0) THEN
C Add BCs for U- and V-displacements
C         NCON=NCON+1
C         BDISP (NCON)=0.0
C         BC (NCON)=NPN* (N-1)+1
C         NCON=NCON+1
C         BDISP (NCON)=0.0
C         BC (NCON)=NPN* (N-1)+2
C       END IF
C Add a BC for the slope
C       NCON=NCON+1
C       BDISP (NCON)=0.0
C       BC (NCON)=NPN*N
C       IDONE (N)=-1
C     END IF
C   END DO
C
C   RETURN
C   END
C
C *****
C
C   SUBROUTINE CHKILDS (NOLOAD)
C
C   Check if any loads and/or displacements were entered
C
C   INCLUDE 'COMMON.INC'
C
C   NOLOAD=0
C   DO I=1,NDOF
C     IF ( F (I) .NE. 0.0 .AND. IBCS (I) .EQ. 0) RETURN
C     IF (DISP (I) .NE. 0.0) RETURN
C   END DO
C
C   NOLOAD=-1
C
C   RETURN
C   END
C
C *****
C
C   CHARACTER*24 FUNCTION TODAY ()
C
C   Function to return the current date in a presentable format
C   NOTE: THIS IS A VAX/VMS-DEPENDENT ROUTINE!
C
C   Use this function in the following way:
C   CHARACTER*24 TODAY
C   WRITE (*,*) 'Date: ', TODAY ()
C
C   This prints something like:

```

```

C      Date: Wednesday Sept. 27, 1989
C
C      CHARACTER      CTIME*23, JUNK*5
C      CHARACTER*5    MONTH(12)
C      CHARACTER*3     INDX(12)
C      CHARACTER*10    WDAY(0:6)
C      INTEGER        STATUS, M, D, Y, NDAYS(12)
C
C      DATA NDAYS /31,28,31,30,31,30,31,31,30,31,30,31/
C      DATA WDAY/' Sunday ',' Monday ',' Tuesday ',' Wednesday ',
C      & ' Thursday ',' Friday ',' Saturday '/
C      DATA MONTH /' Jan. ',' Feb. ',' March ',' April ',' May ',' June ',
C      & ' July ',' Aug. ',' Sept. ',' Oct. ',' Nov. ',' Dec. '/
C      DATA INDX  /' JAN ',' FEB ',' MAR ',' APR ',' MAY ',' JUN ',
C      & ' JUL ',' AUG ',' SEP ',' OCT ',' NOV ',' DEC '/
C
C      Get date and time from the system. Note that this is a VAX-dependent routine
C      This returns something like: CTIME = 27-SEP-1989 08:41:56.21
C
C      STATUS=LIB$DATE_TIME(CTIME)
C
C      Put date into a more presentable format
C
C      TODAY = '
C      TODAY(17:18) = CTIME(1:2)
C      TODAY(21:24) = CTIME(8:11)
C
C      Find the right month
C
C      DO I=1,12
C      IF( CTIME(4:6) .EQ. INDX(I) ) THEN
C      TODAY(11:15) = MONTH(I)
C      M=I
C      END IF
C      END DO
C
C      Now find the corresponding weekday
C
C      JUNK=CTIME(1:2)
C      READ(JUNK,*)D
C      JUNK=CTIME(8:11)
C      READ(JUNK,*)Y
C      Y=Y-1
C      M=M-1
C      N=365*Y+Y/4-Y/100+Y/400
C      N=N+D
C      DO I=1,M
C      N=N+NDAYS(I)
C      END DO
C      IDAY=MOD(N,7)
C
C      TODAY(1:10)=WDAY(IDAY)
C
C      All Done
C
C      RETURN
C      END
C
C*****
C***** THE FOLLOWING IS THE SSAMGEN.FOR PROGRAM *****
C*****
C
C      PROGRAM SSAMGEN
C
C      This program generates a finite element mesh file for the

```

C Simplified Stiffener Analysis Model (SSAM) program.
C
C The model generated here is optimized to produce a small
C stiffness matrix bandwidth in the SSAM program.
C
C
C Written by: Steve Wanthal
C McDonnell Aircraft Co.
C St. Louis, MO
C October, 1989
C
C This program was developed under MCAIR Structural Research IRAD 7-331

C COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS
C RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL
C DOUGLAS CORPORATION.
C

INCLUDE 'COMMON.INC'
DIMENSION XT(25)
CHARACTER MESHFIL*30,DESCR*80
LOGICAL LFLANGE,RFLANGE

COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
& XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
& EIFLL,EIFLR,EIWEB,EICAP,EISK,
& EAFLL,EAFLR,EAWEB,EACAP,EASK,
& TFLL,TFLR,TCAP,TSK,RAD,
& E,G,THICK,FWT,XILS,
& POFLL,POFLR,POCAP,POSKL,POSKR,XT,
& XL,YL,XR,YR,XBAR,YBAR
COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

C DATA ISCRN,IKYBD,IFILE,IOUT/6,5,1,2/
C

VERSION=1.10

C Open the mesh data file
C

WRITE(ISCRN,2225)VERSION
1000 WRITE(ISCRN,2230)
READ(IKYBD,'(A30)',ERR=1000)MESHFIL
IF(MESHFIL.EQ.'STOP'.OR.MESHFIL.EQ.'stop')STOP
OPEN(UNIT=IOUT,FILE=MESHFIL,STATUS='UNKNOWN',ERR=1000)
REWIND IOUT

C Get input information from file SSAMGEN.INP or from user
C

CALL MGENIN

XCAP=XCAP2-XCAP1
IF(NSKR.EQ.0) XSKR=0.0
IF(NFLR.EQ.0) THEN
XFLR=0.0
TFLR=0.0
END IF
IF(NFLL.EQ.0) THEN
XFLL=0.0
TFLL=0.0
END IF
IF(NSKL.EQ.0) XSKL=0.0
IF(NCAP.EQ.0) THEN
XCAP=0.0
TCAP=0.0
END IF


```

C
C RFLANGE = TRUE if the stiffener has a flange to the right of the web
C LFLANGE = TRUE if the stiffener has a flange to the left of the web
C
      LFLANGE=(NELL.GT.0)
      RFLANGE=(NFLR.GT.0)

      PI = ACOS(-1.0000000)
C
C Calculate overall length of the panel
C
      XLEN=XSKR+XFLR+XFLL+XSKL
C
C Calculate the corners of the web
C
      X0=XSKL+XFLL
      Y0=TSK

      X1=XCAP2
      Y1=TSK+HWEB+TCAP/2.0

      YL=TSK+THICK+TFLL/2.0
      YR=TSK+THICK+TFLR/2.0
C
C This is the complex part. We need to figure out the properties of the
C web/flange corners and the starting and ending points of the corners.
C The corners are defined by the angles they curve through, the radii, and
C coordinates of both ends of the corner arcs. (XBAR,YBAR) is the point
C where the web intersects the corner arcs. (XR,YR) is the point where the
C right-hand flange begins to curve up into the web, and (XL,YL) is the
C point where the left-hand flange begins to curve up. The meaning of RAD
C depends on the geometry of the web and flange(s).
C
C There are nine possible flange/web combinations:
C
C 1) vertical web with a flange on the right
C 2) vertical web with a flange on the left
C 3) vertical web with flanges on the right and left
C 4) web of positive slope with a flange on the left
C 5) web of negative slope with a flange on the left
C 6) web of positive slope with a flange on the right
C 7) web of negative slope with a flange on the right
C 8) web of positive slope with flanges on the left and right
C 9) web of negative slope with flanges on the left and right
C
      ICASE=0
      IF (X1-X0.EQ.0.0) THEN
        IF (LFLANGE.AND.RFLANGE) THEN
          ICASE=3
          CALL CASE3
        ELSE IF (RFLANGE) THEN
          ICASE=1
          CALL CASE1
        ELSE IF (LFLANGE) THEN
          ICASE=2
          CALL CASE2
        END IF
      ELSE
        SLOPE=(Y1-Y0)/(X1-X0)
        IF (SLOPE.GT.0.0) THEN
          IF (LFLANGE.AND.RFLANGE) THEN
            ICASE=8
            CALL CASE8
          ELSE IF (RFLANGE) THEN
            ICASE=6
            CALL CASE6
          
```

```

        ELSE IF (LFLANGE) THEN
            ICASE=4
            CALL CASE4
        END IF
    ELSE
        IF (LFLANGE.AND.RFLANGE) THEN
            ICASE=9
            CALL CASE9
        ELSE IF (RFLANGE) THEN
            ICASE=7
            CALL CASE7
        ELSE IF (LFLANGE) THEN
            ICASE=5
            CALL CASE5
        END IF
    END IF
END IF
C Error in input
IF (ICASE.EQ.0) THEN
    WRITE (ISCRN,*) 'ERROR - YOU HAVE NO FLANGES SPECIFIED'
    STOP
END IF

C
C NFRNP = Number of nodes defining the frame elements
C NUMNP = Total number of nodes
C NOTE: Nodes 1-NFRNP define frame elements, and
C       nodes NFRNP+1 - NUMNP define membrane elements.
C       This convention allows a minimal stiffness matrix size and bandwidth.
C
        NFRNP=NSKR+2*(NFLR+NFLR)+NSKL+NWEB+NCAP+2
C
C Generate frame element connectivity
C
        IEL=0
C
C NSKR Connectivity
C
        IF (NSKR.GT.0) DEL=XSKR/FLOAT(NSKR)
        DO I=1,NSKR
            IEL=IEL+1
            NBCON(I,1)=I
            NBCON(I,2)=I+1

            EI(I)=EISK
            EA(I)=EASK

            XORD(I)=XLEN-FLOAT(I-1)*DEL
            XORD(I+1)=XORD(I)-DEL
            YORD(I)=TSK/2.0
            YORD(I+1)=TSK/2.0
        END DO
C
C NFLR Connectivity - use at least 3 elements to model the corner radius
C
        IF (RFLANGE) THEN
            NCORR=NFLR*(XR-XBAR)/XFLR
            IF (NCORR.LT.3) NCORR=3
            ANGLE=0.0
            DO I=1,NFLR
C skin
                IEL=IEL+1
                NBCON(IEL,1)=(NSKR+1)+2*(I-1)
                NBCON(IEL,2)=NBCON(IEL,1)+2
                EI(IEL)=EISK
                EA(IEL)=EASK
            END DO
        END IF
    END IF
END IF

```

```

C flange
  IELF=NSKR+NFLR+NFL+NSKL+I
  NBCON (IELF, 1) =NBCON (IEL, 1) +1
  NBCON (IELF, 2) =NBCON (IEL, 2) +1
  EI (IELF) =EIFLR
  EA (IELF) =EAFLR

  IF (I.LE.NFLR-NCORR) THEN
    DEL=(XSKL+XFLL+XFLR-XR)/FLOAT(NFLR-NCORR)
    XORD(NBCON(IELF, 1))=XLEN-XSKR-FLOAT(I-1)*DEL
    XORD(NBCON(IELF, 2))=XORD(NBCON(IELF, 1))-DEL
    YORD(NBCON(IELF, 1))=TSK+THICK+TFLR/2.0
    YORD(NBCON(IELF, 2))=TSK+THICK+TFLR/2.0
    XORD(NBCON(IEL, 1))=XORD(NBCON(IELF, 1))
    XORD(NBCON(IEL, 2))=XORD(NBCON(IELF, 2))
    YORD(NBCON(IEL, 1))=TSK/2.0
    YORD(NBCON(IEL, 2))=TSK/2.0

  ELSE
    ANGLE=ANGLE+THETAR/FLOAT(NCORR)

    XORD(NBCON(IELF, 2))=XR-RADR*SIN(ANGLE)
    YORD(NBCON(IELF, 2))=YR+RADR*(1.0-COS(ANGLE))
    YORD(NBCON(IEL, 2))=YORD(NBCON(IEL, 1))

C      IF (LFLANGE) THEN
C        XORD(NBCON(IEL, 2))=XORD(NBCON(IELF, 2))
C      ELSE
C        DEL=(XR-XFLL-XSKL)/FLOAT(NCORR)
C        XORD(NBCON(IEL, 2))=XORD(NBCON(IEL, 1))-DEL
C      ENDIF
    END IF

  END DO
END IF

C Now do NFLL connectivity
C
  NEXTRA=NWEB+NCAP
  IF (LFLANGE) THEN
    NCORL=NFL*(XBAR-XL)/XFLL
    IF (NCORL.LT.3) NCORL=3
    ANGLE=THETAL
    DO I=1, NFLL
C skin
      IEL=IEL+1
      IF (IEL.EQ.1) THEN
        NBCON (IEL, 1) =1
      ELSE
        NBCON (IEL, 1) =NBCON (IEL-1, 2)
      END IF
      NBCON (IEL, 2) =NBCON (IEL, 1) +2
      IF (I.LE.NEXTRA+1.AND.I.NE.1) NBCON (IEL, 2) =NBCON (IEL, 2) +1
      EI (IEL) =EISK
      EA (IEL) =EASK
C flange
      IELF=NSKR+NFLR+NFL+NSKL+NFLR+I
      NBCON (IELF, 1) =NBCON (IEL, 1) +1
      NBCON (IELF, 2) =NBCON (IEL, 2) +1
      EI (IELF) =EIFLL
      EA (IELF) =EAFLL

      IF (I.EQ.1.AND.(.NOT.RFLANGE)) THEN
        XORD(NBCON(IELF, 1))=XBAR
        YORD(NBCON(IELF, 1))=YBAR
        YORD(NBCON(IEL, 1))=TSK/2.0

```

```

      DEL=(XFL-L-XL)/FLOAT(NCORL)
      XORD(NBCON(IEL,1))=XORD(NBCON(IEL-1,2))-DEL
    END IF

    IF (I.LE.NCORL) THEN
      ANGLE=ANGLE-THETAL/FLOAT(NCORL)

      XORD(NBCON(IEL,2))=XL+RADL*SIN(ANGLE)
      YORD(NBCON(IEL,2))=YL+RADL*(1.0-COS(ANGLE))
      YORD(NBCON(IEL,2))=YORD(NBCON(IEL,1))

C      IF (RFLANGE) THEN
C        XORD(NBCON(IEL,2))=XORD(NBCON(IEL,2))
C      ELSE
C        DEL=(XFL-L+XSKL-XL)/FLOAT(NCORL)
C        XORD(NBCON(IEL,2))=XORD(NBCON(IEL,1))-DEL
C      ENDIF

      ELSE

      DEL=(XL-XSKL)/FLOAT(NFLL-NCORL)
      XORD(NBCON(IEL,2))=XL-FLOAT(I-NCORL)*DEL
      YORD(NBCON(IEL,2))=TSK/2.0

      XORD(NBCON(IEL,2))=XORD(NBCON(IEL,2))
      YORD(NBCON(IEL,2))=TSK+THICK+TFLL/2.0
    END IF

  END DO
END IF

C
C NSKL Connectivity
C
  IF (NSKL.NE.0) DEL=XSKL/FLOAT(NSKL)
  DO I=1,NSKL
    IEL=IEL+1
    NBCON(IEL,1)=NBCON(IEL-1,2)
    IF (I.EQ.1) THEN
      NBCON(IEL,2)=NBCON(IEL,1)+2
    ELSE
      NBCON(IEL,2)=NBCON(IEL,1)+1
    END IF
    IF (I+NFLL.LE.NEXTRA+1) NBCON(IEL,2)=NBCON(IEL,2)+1
    IF (NFLL.EQ.0.AND.I.EQ.1) NBCON(IEL,2)=NBCON(IEL,2)-1
    EI(IEL)=EISK
    EA(IEL)=EASK
    XORD(NBCON(IEL,2))=XSKL-FLOAT(I)*DEL
    YORD(NBCON(IEL,2))=TSK/2.0
  END DO

C
C Web connectivity
C
  IF (NWEB.NE.0) THEN
    DELX=(X1-XBAR)/FLOAT(NWEB)
    DELY=(Y1-YBAR)/FLOAT(NWEB)
  END IF

  DO I=1,NWEB
    IEL=NSKR+NFLR+NFLL+NSKL+NFLR+NFLL+I
    IF (I.EQ.1.AND.RFLANGE) THEN
      NBCON(IEL,1)=NBCON(NSKR+NFLR+NFLL+NSKL+NFLR,2)
    ELSE IF (I.EQ.1) THEN
      NBCON(IEL,1)=NBCON(NSKR+NFLL+NSKL+1,1)
    ELSE
      NBCON(IEL,1)=NBCON(IEL-1,2)
    END IF
  END DO

```

```

      END IF
      IF (I.LE.NFLL) THEN
        NBCON(IEL,2)=NBCON(IEL,1)+3
      ELSE IF (I.LE.NFLL+NSKL) THEN
        NBCON(IEL,2)=NBCON(IEL,1)+2
      ELSE
        NBCON(IEL,2)=NBCON(IEL,1)+1
      END IF
      EA(IEL)=EAWEB
      EI(IEL)=EIWEB
      XORD(NBCON(IEL,2))=XORD(NBCON(IEL,1))+DELX
      YORD(NBCON(IEL,2))=YORD(NBCON(IEL,1))+DELY
    END DO
C
C Generate cap connectivity
C
      IF (NCAP.GT.0) DEL=XCAP/FLOAT(NCAP)

      DO I=1,NCAP
        IEL=IEL+1
        NBCON(IEL,1)=NBCON(IEL-1,2)
        IF (I+NWEB.LE.NFLL) THEN
          NBCON(IEL,2)=NBCON(IEL,1)+3
        ELSE IF (I+NWEB.LE.NFLL+NSKL) THEN
          NBCON(IEL,2)=NBCON(IEL,1)+2
        ELSE
          NBCON(IEL,2)=NBCON(IEL,1)+1
        END IF
        EA(IEL)=EACAP
        EI(IEL)=EICAP
        XORD(NBCON(IEL,2))=XORD(NBCON(IEL,1))-DEL
        YORD(NBCON(IEL,2))=Y1
        IF (I.EQ.1) YORD(NBCON(IEL,1))=Y1
      END DO
C
C Now add the stitching. We may need to move some
C node locations in order to get the stitching in the
C right places.
C
      DO I=1,NSTITCH
        NODE1=0
        NODE2=0
        DXMIN=1.0E20
        DO K=NSKR+1,NSKR+NFLR+NFLL-1
C
C Find the node nearest the x-coordinate of this stitch, but
C don't use the ends of the flanges.
C
          DX=ABS( XORD(NBCON(K,2))-XT(I) )
          IF (DX.LT.DXMIN) THEN
            NODE1=NBCON(K,2)
            NODE2=NBCON(K+NSKL+NFLR+NFLL,2)
            DXMIN=DX
          C
          WRITE(ISCRN,*) K,NODE1,NODE2,DXMIN
        END IF
      END DO
C
C Move the nearest node to the stitch location
C
      XORD(NODE1)=XT(I)
      XORD(NODE2)=XT(I)
C
C Add the stitch connectivity

```

```

C
      NTCN(I,1)=NODE1
      NTCN(I,2)=NODE2

      END DO

C
C Now generate plate element connectivity - ignore the corner radius
C for now.
C
      NPCL=0
      NEWNOD=NFRNP
      DO I=1,NFLR+NFL
        IF (XORD(NBCN(NSKR+I,1)).GT.XORD(NBCN(NSKR+I,2))) THEN
          NPCL=NPCL+1
          IF (NPCL.EQ.1) THEN
            NEWNOD=NEWNOD+1
            NPCN(NPCL,2)=NEWNOD
            NEWNOD=NEWNOD+1
            NPCN(NPCL,3)=NEWNOD

            XORD(NPCN(NPCL,2))=XORD(NBCN(NSKR+I,1))
            XORD(NPCN(NPCL,3))=XORD(NBCN(NSKR+NFLR+NFL+NSKL+I,1))
            YORD(NPCN(NPCL,2))=TSK
            YORD(NPCN(NPCL,3))=TSK+THICK
            NPLOB(NPCN(NPCL,2))=NBCN(NSKR+I,1)
            NPLOB(NBCN(NSKR+I,1))=NPCN(NPCL,2)
            NPLOB(NPCN(NPCL,3))=NBCN(NSKR+NFLR+NFL+NSKL+I,1)
            NPLOB(NBCN(NSKR+NFLR+NFL+NSKL+I,1))=NPCN(NPCL,3)
          ELSE
            NPCN(NPCL,2)=NPCN(NPCL-1,1)
            NPCN(NPCL,3)=NPCN(NPCL-1,4)
          END IF

          NEWNOD=NEWNOD+1
          NPCN(NPCL,1)=NEWNOD
          NEWNOD=NEWNOD+1
          NPCN(NPCL,4)=NEWNOD
          XORD(NPCN(NPCL,1))=XORD(NBCN(NSKR+I,2))
          XORD(NPCN(NPCL,4))=XORD(NBCN(NSKR+NFLR+NFL+NSKL+I,2))
          YORD(NPCN(NPCL,1))=YORD(NPCN(NPCL,2))
          YORD(NPCN(NPCL,4))=YORD(NPCN(NPCL,3))
        C
        C NPLOB associates plate element nodes with frame element nodes.
        C
          NPLOB(NPCN(NPCL,4))=NBCN(NSKR+NFLR+NFL+NSKL+I,2)
          NPLOB(NBCN(NSKR+NFLR+NFL+NSKL+I,2))=NPCN(NPCL,4)
          NPLOB(NPCN(NPCL,1))=NBCN(NSKR+I,2)
          NPLOB(NBCN(NSKR+I,2))=NPCN(NPCL,1)
        END IF

      END DO

C
C Fix the node coordinates under the radius
C
      IF (RFLANGE.AND.LFLANGE) THEN
C
C Both right and left hand flanges
C First find the point right under the intersection of the flanges
C
        RL=RADL+TFLL/2.0
        RR=RADR+TFLR/2.0
        DELX=XR-XL
        DELY=RR-RL
        ANGLE=ATAN(DELY/DELX)*180.0/PI
        IF (ANGLE.GT.1.0) THEN
          AA=DELX/DELY

```

```

      BB=(RL*RL-RR*RR+DELX*DELX+DELY*DELY)/(2.0*DELY)
      XX=(AA*BB+SQRT(RL*RL-BB*BB+AA*AA*RL*RL))/(1.0*AA*AA)
      YY=-AA*XX+BB+YL+RADL
      XX=XX+XL
    ELSE
      XX=DELX/2.0+XL
      YY=-SQRT(RL*RL-DELX*DELX/4.0)+YL+RADL
    END IF
    XXX=(XR-XX)/RR
    IF (ABS(XXX).LT.1.0) THEN
      ANGLR=ASIN(XXX)
    ELSE
      C      WRITE(6,*)XR,XX,RR,XXX
      ANGLR=PI/2.0
    END IF
    XXX=(XX-XL)/RL
    IF (ABS(XXX).LT.1.0) THEN
      ANGLL=ASIN(XXX)
    ELSE
      C      WRITE(6,*)XX,XL,RL,XXX
      ANGLL=PI/2.0
    END IF
    C      WRITE(6,*)ANGLR*180./PI,ANGLL*180./PI

    ANGLE=0.0
    DO I=1,NCORR

      ANGLE=ANGLE+ANGLR/FLOAT(NCORR)
      IEL=NSKR+NFLR+NFL+NSKL+NFLR-NCORR+I
      N2=NBCON(IEL,2)
      N3=NPTOB(N2)

      XORD(N3)=XR-RR*SIN(ANGLE)
      YORD(N3)=YR+RADL-RR*COS(ANGLE)
      IEL2=NSKR+NFLR-NCORR+I
      N2=NBCON(IEL2,2)
      N4=NPTOB(N2)
      XORD(N2)=XORD(N3)
      YORD(N4)=YORD(N3)

    END DO

    ANGLE=0.0
    DO I=NCORL,2,-1

      ANGLE=ANGLE+ANGLL/FLOAT(NCORL)
      IEL=NSKR+NFLR+NFL+NSKL+NFLR+I
      N2=NBCON(IEL,1)
      N3=NPTOB(N2)

      XORD(N3)=XL+RL*SIN(ANGLE)
      YORD(N3)=YL+RADL-RL*COS(ANGLE)
      IEL2=NSKR+NFLR+I
      N2=NBCON(IEL2,1)
      N4=NPTOB(N2)
      XORD(N2)=XORD(N3)
      YORD(N4)=YORD(N3)

    END DO
  C
  C Left hand flange only
  C
    ELSE IF (LFLANGE) THEN
      RADI=RADL+TFLL/2.0

```

```

DO I=NCORL,1,-1

    IEL=NSKR+NFLR+NELL+NSKL+NFLR+I

    N2=NBCON (IEL,1)
    N3=NPTOB (N2)

    ANGLE=ASIN ( (XORD (N2) -XL) /RADL)

    IF (N3.NE.0) THEN
        XORD (N3)=XL+RADI*SIN (ANGLE)
        YORD (N3)=TSK+THICK+RADI*(1.0-COS (ANGLE))
        NN3=N3
    END IF

END DO

C
DELY=YORD (NN3) -TSK
FILANG=15.6*PI/180.0
DELX=DELY*TAN (FILANG)
DX=XSKL+XFLL+DELX-XL
DO I=NCORL,1,-1

    IEL=NSKR+NFLR+I

    N2=NBCON (IEL,1)
    N3=NPTOB (N2)
    XORD (N2)=XORD (NBCON (IEL,2)) +DX/FLOAT (NCORL)
    N4=NPTOB (NBCON (NSKR+NFLR+NELL+NSKL+NFLR+I,1))
    IF (N4.NE.0.AND.XORD (N2) .LT.XORD (N4)) XORD (N2)=XORD (N4)
    IF (N3.NE.0) XORD (N3)=XORD (N2)
END DO
DX=(XLEN-XORD (N2)) /FLOAT (NSKR)
DO IEL=1,NSKR
    N2=NBCON (IEL,1)
    XORD (N2)=XLEN-DX*FLOAT (IEL-1)
END DO

C
C Right hand flange only
C
ELSE IF (RFLANGE) THEN

    RADI=RADR+TFLR/2.0
    DO I=NFLR-NCORR+1,NFLR

        IEL=NSKR+NFLR+NELL+NSKL+I

        N2=NBCON (IEL,2)
        N3=NPTOB (N2)

        ANGLE=ASIN ( (XR-XORD (N2)) /RADR)
        IF (N3.NE.0) THEN
            XORD (N3)=XR-RADI*SIN (ANGLE)
            YORD (N3)=TSK+THICK+RADI*(1.0-COS (ANGLE))
            NN3=N3
        END IF
    END DO

C
DELY=YORD (NN3) -TSK
FILANG=15.6*PI/180.0
DELX=DELY*TAN (FILANG)
DX=XR+DELX-XSKL-XFLL
DO I=1,NCORR

    IEL=NSKR+NFLR-NCORR+I

```



```

      N2=NBCON (IEL, 2)
      N3=NPTOB (N2)
      XORD (N2)=XORD (NBCON (IEL, 1)) -DX/FLOAT (NCORR)
      N4=NPTOB (NBCON (NSKR+NFLR+NELL+NSKL+NFLR-NCORR+I, 2))
      IF (N4.NE.0.AND.XORD (N2).GT.XORD (N4)) XORD (N2)=XORD (N4)
      IF (N3.NE.0) XORD (N3)=XORD (N2)

      END DO

      DX=(XSKL-DELX)/FLOAT (NSKL)
      DO I=1, NSKL
        IEL=NSKR+NFLR+NELL+I
        N2=NBCON (IEL, 2)
        XORD (N2)=XORD (NBCON (IEL, 1)) -DX
      END DO

      END IF
      NUMNP=NEWNOD
C
C Generate boundary conditions:
C assume it is fixed at the right hand side, and
C zero slope at the left hand side.
C Also, no axial displacement at these locations.
C
      IBC (1,1)=1
      IBC (1,2)=1
      IBC (1,3)=1
      IBC (1,4)=1

      IBC (2,1)=NBCON (NSKR+NFLR+NELL+NSKL, 2)
      IBC (2,2)=1
      IBC (2,3)=0
      IBC (2,4)=1

      IBC (3,1)=NBCON (NSKR+2*(NFLR+NELL)+NSKL+NWEB+NCAP, 2)
      IBC (3,2)=1
      IBC (3,3)=0
      IBC (3,4)=1

      NUMBC=3
C
C Extra BC if the left-hand flange runs the length of the skin
C
      IF (NSKR.EQ.0.AND.RFLANGE) THEN
        NUMBC=NUMBC+1
        IBC (NUMBC, 1)=NBCON (NFLR+NELL+NSKL+1, 1)
        IBC (NUMBC, 2)=1
        IBC (NUMBC, 3)=1
        IBC (NUMBC, 4)=1
      END IF
C
C Extra BC if the right-hand flange runs the length of the skin
C
      IF (NSKL.EQ.0.AND.LFLANGE) THEN
        NUMBC=NUMBC+1
        IBC (NUMBC, 1)=NBCON (NSKR+2*(NFLR+NELL), 2)
        IBC (NUMBC, 2)=1
        IBC (NUMBC, 3)=0
        IBC (NUMBC, 4)=1
      END IF
C
C Generate work equivalent point loads on cap,
C flanges and skin due to the specified uniform loads
C
      MM=NSKR+2*(NFLR+NELL)+NSKL+NWEB

```

```

IF (POSKL.NE.0.0) CALL WEQIV (POSKL,1,NSKR)
IF (POFLL.NE.0.0) CALL WEQIV (POFLL,NSKR+NFLR+NELL+NSKL+1,
& NSKR+NFLR+NELL+NSKL+NFLR-NCORR)
IF (POCAP.NE.0.0) CALL WEQIV (POCAP,MM+1,MM+NCAP)
IF (POFLR.NE.0.0) CALL WEQIV (POFLR,NSKR+2*NFLR+NELL+NSKL+NCORL+1,
& NSKR+2*NFLR+2*NELL+NSKL)
IF (POSKR.NE.0.0) CALL WEQIV (POSKR,NSKR+NFLR+NELL+1,
&NSKR+NFLR+NELL+NSKL)

```

C

C Check if we have any inside-out skin elements.. This may happen if there
C is only one flange and it is underneath the web. (ICASE=5 OR 6)

C

```

IF (ICASE.EQ.5.AND.NELL.NE.NPEL) THEN
  N=NELL-NPEL
  XX1=XORD (NBCON (NSKR+N,2))
  IF (NSKR.GT.0) THEN
    NN=N+1
    XX2=XORD (NBCON (NSKR,1))
  ELSE
    NN=N
    XX2=XLEN
  END IF
  DEL = (XX2-XX1)/FLOAT (NN)
  DO I=NSKR+N,NSKR+1,-1
    XORD (NBCON (I,1))=XORD (NBCON (I,2))+DEL
  END DO
END IF
IF (ICASE.EQ.6.AND.NFLR.NE.NPEL) THEN
  N=NFLR-NPEL
  XX2=XORD (NBCON (NSKR+NPEL,2))
  IF (NELL+NSKL.GT.0) THEN
    NN=N+1
    XX1=XORD (NBCON (NSKR+NFLR+1,2))
  ELSE
    NN=N
    XX1=0.0
  END IF
  DEL = (XX2-XX1)/FLOAT (NN)
  DO I=NSKR+NPEL+1,NSKR+NFLR
    XORD (NBCON (I,2))=XORD (NBCON (I,1))-DEL
  END DO
END IF

```

C

C Ready to write the mesh

C

```

1254 WRITE (IOUT,' (A80)')DESCR
  WRITE (IOUT,2000) NSKR+NSKL+NFLR+NELL,NFLR+NELL+NWEB+NCAP,NPEL,
&NUMBC,NUMLDS,NSTITCH
  WRITE (IOUT,2010) (I, (NBCON (I,JJ),JJ=2,1,-1),EI (I),EA (I),
&I=1,NSKR+NSKL+2* (NFLR+NELL)+NWEB+NCAP)
  WRITE (IOUT,2020) (I, (NPCON (I,J),J=1,4),E,G,FWT,XILS,I=1,NPEL)
  DO I=1,NUMNP
    WRITE (IOUT,2025) I,NPTOB (I),XORD (I),YORD (I)
  END DO
  DO I=1,NUMBC
    WRITE (IOUT,2030) (IBC (I,JJ),JJ=1,NPN+1), (0.0,JJ=1,NPN)
  END DO
  IF (NUMLDS.GT.0)
&WRITE (IOUT,2040) (IFRC (I), (FORCE (I,JJ),JJ=1,NPN),I=1,NUMLDS)
  IF (NSTITCH.GT.0)
&WRITE (IOUT,2050) (I, (NTCON (I,JJ),JJ=1,2),TEA (I),I=1,NSTITCH)

```

C

```

CLOSE (IOUT)
STOP

```

C

C Formats

```

C
2000 FORMAT(6I4)
2010 FORMAT(3I4,2E12.5)
2020 FORMAT(5I4,4E12.5)
2025 FORMAT(2I4,2E12.5)
2030 FORMAT(4I4,3E12.5)
2040 FORMAT(I4,3E12.5)
2050 FORMAT(3I4,E12.5)
2041 FORMAT(I4,3E14.5)
2225 FORMAT(/,1X,58('*'),/,24X,
&'SSAMGEN',/,22X,'Version',F5.2,/,1X,58('*'),///,5X,
&'COPYRIGHT UNPUBLISHED - (1989/MCAIR). ALL RIGHTS',/,5X,
&'RESERVED UNDER THE COPYRIGHT LAWS BY MCDONNELL',/,5X,
&'DOUGLAS CORPORATION.',///)
2230 FORMAT(/,'$Enter a model data file name (or STOP)> ')
END
SUBROUTINE MGENIN

```

```

C
C Routine to input the mesh generation information
C

```

```

INCLUDE 'COMMON.INC'
DIMENSION XT(25)
CHARACTER DESCR*80
COMMON /MGENINP/DESCR,NSKR,NFLR,NFLR,NSKL,NWEB,NCAP,
& XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
& EIFLL,EIFLR,EIWEB,EICAP,EISK,
& EAFLL,EAFLR,EAWEB,EACAP,EASK,
& TFLR,TFLR,TCAP,TSK,RAD,
& E,G,THICK,FWT,XILS,
& POFLL,POFLR,POCAP,POSKL,POSKR,XT,
& XL,YL,XR,YR,XBAR,YBAR

```

```

C
IIN=0
OPEN(UNIT=IFILE,FILE='SSAMGEN.INP',STATUS='OLD',ERR=5000)
REWIND IFILE
IIN=IFILE
1005 CONTINUE
IF(IIN.NE.IFILE)
&WRITE(ISCRN,*)'ENTER A DESCRIPTION'
READ(IIN,'(A80)',ERR=5000)DESCR
IF(IIN.NE.IFILE)
&WRITE(ISCRN,*)'ENTER NSKR, NFLR, NFLR, NSKL, NWEB AND NCAP'
READ(IIN,*,ERR=5000)NSKR,NFLR,NFLR,NSKL,NWEB,NCAP

```

```

C
C Make sure at least 4 elements are used for the flange(s)
C

```

```

IF(NFLR.GT.0.AND.NFLR.LT.4)THEN
WRITE(ISCRN,*)
&'WARNING - USING FOUR ELEMENTS IN RIGHT-HAND FLANGE'
NFLR=4
END IF
IF(NFLL.GT.0.AND.NFLL.LT.4)THEN
WRITE(ISCRN,*)
&'WARNING - USING FOUR ELEMENTS IN LEFT-HAND FLANGE'
NFLL=4
END IF

```

```

C
IF(IIN.NE.IFILE)
&WRITE(ISCRN,*)'ENTER XSKR, XFLL, XFLR, XSKL, XCAP1, XCAP2, HWEB'
READ(IIN,*,ERR=5000)XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB
IF(IIN.NE.IFILE)WRITE(ISCRN,*)
&'ENTER EI FOR R.H. FLANGE, L.H. FLANGE, WEB, CAP & SKIN'
READ(IIN,*,ERR=5000)EIFLL,EIFLR,EIWEB,EICAP,EISK
IF(IIN.NE.IFILE)WRITE(ISCRN,*)
&'ENTER EA FOR R.H. FLANGE, L.H. FLANGE, WEB, CAP & SKIN'
READ(IIN,*,ERR=5000)EAFLL,EAFLR,EAWEB,EACAP,EASK

```

```

      IF(IIN.NE.IFILE)WRITE(ISCRN,*)
      &'ENTER THICKNESS FOR RH & LH FLANGES, CAP AND SKIN'
      READ(IIN,*,ERR=5000)TFLR, TFLR, TCAP, TSK
      IF(IIN.NE.IFILE)
      &WRITE(ISCRN,*)'ENTER FLANGE INNER MOLDLINE RADIUS'
      READ(IIN,*,ERR=5000)RAD
      IF(IIN.NE.IFILE)
      &WRITE(ISCRN,*)'ENTER E, G, THICKNESS OF INTERFACE, FWT & ILS'
      READ(IIN,*,ERR=5000)E,G,THICK,FWT,XILS
      IF(IIN.NE.IFILE)WRITE(ISCRN,*)
      &'ENTER P0 ON RH SKIN AND FLANGE, CAP, LH FLANGE & SKIN'
      READ(IIN,*,ERR=5000)POSKR, POFLR, POCAP, POFLR, POSKL
      IF(IIN.NE.IFILE)
      &WRITE(ISCRN,*)'ENTER NUMBER OF STITCHES'
      READ(IIN,*,ERR=5000)NSTITCH
      IF(IIN.NE.IFILE.AND.NSTITCH.GT.0)WRITE(ISCRN,*)
      &'STITCH COORDINATES ARE MEASURED FROM THE FLANGE EDGE'
      NGOOD=0
      DO I=1,NSTITCH
        NGOOD=NGOOD+1
        IF(IIN.NE.IFILE)
        & WRITE(ISCRN,*)'ENTER X-COORD AND EA FOR STITCH #',NGOOD
        READ(IIN,*,ERR=5000)XT(NGOOD),TEA(NGOOD)
        IF(XT(NGOOD).LT.XSKL.OR.XT(NGOOD).GT.XSKL+XFLL+XFLR)THEN
          WRITE(ISCRN,*)'WARNING - STITCH IS NOT ON THE FLANGE AREA'
          WRITE(ISCRN,*)'AND WILL BE IGNORED'
          NGOOD=NGOOD-1
        END IF
      END DO
      NSTITCH=NGOOD
C
      IF(IIN.EQ.IFILE)CLOSE(IFILE)
      RETURN
C
5000 CONTINUE
      IF(IIN.EQ.IKYBD)THEN
        WRITE(ISCRN,*)'ERROR READING INPUT - YOU MUST START OVER'
      ELSE
        WRITE(ISCRN,*)'ERROR READING INPUT FILE SSAMGEN.INP'
        WRITE(ISCRN,*)'YOU MUST ENTER THE INFORMATION MANUALLY:'
      END IF
      IIN=IKYBD
      GO TO 1005
C
      END
      SUBROUTINE WEQIV(P0,IEL1,IEL2)
C
C Generate work equivalent loads on elements IEL1 thru IEL2
C due to a uniformly distributed load of magnitude P0.
C
      INCLUDE 'COMMON.INC'
C
      IF(P0.EQ.0.0)RETURN
      DO IEL=IEL1,IEL2
        N1=NBCON(IEL,1)
        N2=NBCON(IEL,2)
        DL=SQRT((XORD(N1)-XORD(N2))**2+(YORD(N1)-YORD(N2))**2)
C
C Do loads at first node
C
        NSKLD=0
        DO I=1,NUMLDS
          IF(IFRC(I).EQ.N1)NSKLD=I
        END DO
        IF(NSKLD.EQ.0)THEN
          NUMLDS=NUMLDS+1

```

```

      NSKLD=NUMLDS
      END IF
      IFRC (NSKLD)=N1
      FORCE (NSKLD,2)=FORCE (NSKLD,2)-P0*DL/2.0
      FORCE (NSKLD,3)=FORCE (NSKLD,3)-P0*DL*DL/12.0
C
C Do loads at second node
C
      NSKLD=0
      DO I=1,NUMLDS
        IF (IFRC (I).EQ.N2) NSKLD=I
      END DO
      IF (NSKLD.EQ.0) THEN
        NUMLDS=NUMLDS+1
        NSKLD=NUMLDS
      END IF
      IFRC (NSKLD)=N2
      FORCE (NSKLD,2)=FORCE (NSKLD,2)-P0*DL/2.0
      FORCE (NSKLD,3)=FORCE (NSKLD,3)+P0*DL*DL/12.0
      END DO
C
      RETURN
      END
      SUBROUTINE CASE1
C
C Routine to find corner radius information for Case #1
C
      INCLUDE 'COMMON.INC'
      DIMENSION XT(25)
      CHARACTER*80 DESCR
      COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&      XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&      EIFLL,EIFLR,EIWEB,EICAP,EISK,
&      EAFLL,EAFLR,EAWEB,EACAP,EASK,
&      TFLR,TFLR,TCAP,TSK,RAD,
&      E,G,THICK,FWT,XILS,
&      POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&      XL,YL,XR,YR,XBAR,YBAR
      COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

      THETAR=PI/2.0
      THETAL=0.0
      RADR=RAD+TFLR/2.0
      RADL=0.0
      XBAR=X0
      YBAR=YR+RADR
      XL=0.0
      XR=XBAR+RADR
C
      RETURN
      END
      SUBROUTINE CASE2
C
C Routine to find corner radius information for Case #2
C
      INCLUDE 'COMMON.INC'
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&      XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&      EIFLL,EIFLR,EIWEB,EICAP,EISK,
&      EAFLL,EAFLR,EAWEB,EACAP,EASK,
&      TFLR,TFLR,TCAP,TSK,RAD,
&      E,G,THICK,FWT,XILS,
&      POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&      XL,YL,XR,YR,XBAR,YBAR

```

```

COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

THETAL=PI/2.0
THETAR=0.0
RADL=RAD+TFLL/2.0
RADR=0.0
XBAR=X0
YBAR=Y1+RADL
XR=0.0
XL=XBAR-RADL

C
  RETURN
  END
  SUBROUTINE CASE3

```

```

C
C Routine to find corner radius information for Case #3
C

```

```

  INCLUDE 'COMMON.INC'
  CHARACTER*80 DESCR
  DIMENSION XT(25)
  COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&                XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&                EIFLL,EIFLR,EIWEB,EICAP,EISK,
&                EAFLL,EAFLR,EAWEB,EACAP,EASK,
&                TFLL,TFLR,TCAP,TSK,RAD,
&                E,G,THICK,FWT,XILS,
&                POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&                XL,YL,XR,YR,XBAR,YBAR
  COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

  THETAR=PI/2.0
  THETAL=PI/2.0
  RADR=RAD+TFLR/2.0
  XBAR=X0
  YBAR=YR+RADR
  RADL=YBAR-YL
  XL=XBAR-RADL
  XR=XBAR+RADR

```

```

C
  RETURN
  END
  SUBROUTINE CASE4

```

```

C
C Routine to find corner radius information for Case #4
C

```

```

  INCLUDE 'COMMON.INC'
  CHARACTER*80 DESCR
  DIMENSION XT(25)
  COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&                XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&                EIFLL,EIFLR,EIWEB,EICAP,EISK,
&                EAFLL,EAFLR,EAWEB,EACAP,EASK,
&                TFLL,TFLR,TCAP,TSK,RAD,
&                E,G,THICK,FWT,XILS,
&                POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&                XL,YL,XR,YR,XBAR,YBAR
  COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

  B=(Y0*X1-X0*Y1)/(X1-X0)
  THETAL=ATAN(SLOPE)
  THETAR=0.0
  RADL=RAD+TFLL/2.0
  RADR=0.0
  XBAR=(YL+RADL*(1.0-COS(THETAL))-B)/SLOPE
  YBAR=XBAR*SLOPE+B
  XL=XBAR-RADL*SIN(THETAL)

```

```

      XR=0.0
C
      RETURN
      END
      SUBROUTINE CASE5
C
C Routine to find corner radius information for Case #5
C
      INCLUDE 'COMMON.INC'
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR, NSKR, NFLL, NFLR, NSKL, NWEB, NCAP,
&              XSKR, XFLL, XFLR, XSKL, XCAP1, XCAP2, HWEB,
&              EIFLL, EIFLR, EIWEB, EICAP, EISK,
&              EAFLL, EAFLR, EAWEB, EACAP, EASK,
&              TFLL, TFLR, TCAP, TSK, RAD,
&              E, G, THICK, FWT, XILS,
&              POFLL, POFLR, POCAP, POSKL, POSKR, XT,
&              XL, YL, XR, YR, XBAR, YBAR
      COMMON /CORNERS/SLOPE, PI, X0, Y0, X1, Y1, THETAL, THETAR, RADL, RADR

      B=(Y0*X1-X0*Y1)/(X1-X0)
      THETAL=PI+ATAN(SLOPE)
      THETAR=0.0
      RADL=RAD+TFLL/2.0
      RADR=0.0
      XBAR=(YL+RADL*(1.0-COS(THETAL))-B)/SLOPE
      YBAR=XBAR*SLOPE+B
      XL=XBAR-RADL*SIN(THETAL)
      XR=0.0
C
      RETURN
      END
      SUBROUTINE CASE6
C
C Routine to find corner radius information for Case #6
C
      INCLUDE 'COMMON.INC'
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR, NSKR, NFLL, NFLR, NSKL, NWEB, NCAP,
&              XSKR, XFLL, XFLR, XSKL, XCAP1, XCAP2, HWEB,
&              EIFLL, EIFLR, EIWEB, EICAP, EISK,
&              EAFLL, EAFLR, EAWEB, EACAP, EASK,
&              TFLL, TFLR, TCAP, TSK, RAD,
&              E, G, THICK, FWT, XILS,
&              POFLL, POFLR, POCAP, POSKL, POSKR, XT,
&              XL, YL, XR, YR, XBAR, YBAR
      COMMON /CORNERS/SLOPE, PI, X0, Y0, X1, Y1, THETAL, THETAR, RADL, RADR

      B=(Y0*X1-X0*Y1)/(X1-X0)
      THETAR=PI-ATAN(SLOPE)
      THETAL=0.0
      RADR=RAD+TFLR/2.0
      RADL=0.0
      XBAR=(YR+RADR*(1.0-COS(THETAR))-B)/SLOPE
      YBAR=XBAR*SLOPE+B
      XR=XBAR+RADR*SIN(THETAR)
      XL=0.0
C
      RETURN
      END
      SUBROUTINE CASE7
C
C Routine to find corner radius information for Case #7
C

```

```

      INCLUDE 'COMMON.INC'
      REAL IFT
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&                XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&                EIFLL,EIFLR,EIWEB,EICAP,EISK,
&                EAFLL,EAFLR,EAWEB,EACAP,EASK,
&                TFLR,TFLR,TCAP,TSK,RAD,
&                E,G,THICK,FWT,XILS,
&                POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&                XL,YL,XR,YR,XBAR,YBAR
      COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

C THIS CASE IS THE SAME AS THE CMARS SYSTEM. DO COORDINATES THE SAME WAY
      HRS=X0
      HCS=XCAP2
      IFT=TFLR
      EFT=TFLR
      HH=HWEB
      THETAR=ATAN(HH/(HRS-HCS))
      THETAS=ATAN((HRS-HCS)/HH)
C      WRITE(ISCRN,*) THETAR,THETAS
      POINTX=HRS
      POINTY=TSK+THICK
      EFOMLR=RAD+EFT
      TEMPY=EFOMLR-EFOMLR*SIN(THETAS)
      XBAR=(2.0*(POINTX-TEMPY*TAN(THETAS))+EFT*COS(THETAS))/2.0
      YBAR=(2.0*(POINTY+TEMPY)+EFT*SIN(THETAS))/2.0
      CEFX=POINTX-TEMPY*TAN(THETAS)+EFOMLR*COS(THETAS)
      CEFY=POINTY+EFOMLR

C      B=(Y0*X1-X0*Y1)/(X1-X0)
      THETAL=0.0
      RADR=RAD+TFLR/2.0
      RADL=0.0
      XR=CEFX
      XL=0.0
C      WRITE(ISCRN,*) XBAR,YBAR,XR,YR
C
      RETURN
      END
      SUBROUTINE CASE8
C
C Routine to find corner radius information for Case #8
C
      INCLUDE 'COMMON.INC'
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&                XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&                EIFLL,EIFLR,EIWEB,EICAP,EISK,
&                EAFLL,EAFLR,EAWEB,EACAP,EASK,
&                TFLR,TFLR,TCAP,TSK,RAD,
&                E,G,THICK,FWT,XILS,
&                POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&                XL,YL,XR,YR,XBAR,YBAR
      COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR

      B=(Y0*X1-X0*Y1)/(X1-X0)
      THETAL=ATAN(SLOPE)
      THETAR=PI/2.0
      RADL=RAD+TFLR/2.0
      XBAR=(YL+RADL*(1.0-COS(THETAL))-B)/SLOPE
      YBAR=XBAR*SLOPE+B
      XL=XBAR-RADL*SIN(THETAL)

```



```

      RADR=YBAR-YR
      XR=XBAR+RADR
C
      RETURN
      END
      SUBROUTINE CASE9
C
C Routine to find corner radius information for Case #9
C
      INCLUDE 'COMMON.INC'
      REAL IFT
      CHARACTER*80 DESCR
      DIMENSION XT(25)
      COMMON /MGENINP/DESCR,NSKR,NFLL,NFLR,NSKL,NWEB,NCAP,
&              XSKR,XFLL,XFLR,XSKL,XCAP1,XCAP2,HWEB,
&              EIFLL,EIFLR,EIWEB,EICAP,EISK,
&              EAFLL,EAFLR,EAWEB,EACAP,EASK,
&              TFLL,TFLR,TCAP,TSK,RAD,
&              E,G,THICK,FWT,XILS,
&              POFLL,POFLR,POCAP,POSKL,POSKR,XT,
&              XL,YL,XR,YR,XBAR,YBAR
      COMMON /CORNERS/SLOPE,PI,X0,Y0,X1,Y1,THETAL,THETAR,RADL,RADR
C THIS CASE IS THE SAME AS THE CMARS SYSTEM. DO COORDINATES THE SAME WAY
      HRS=X0
      HCS=XCAP2
      IFT=TFLL
      EFT=TFLR
      HH=HWEB-THICK-TFLL
      B=(Y0*X1-X0*Y1)/(X1-X0)
      THETAR=ATAN(HH/(HRS-HCS))
      THETAS=ATAN((HRS-HCS)/HH)
      THETAL=PI/2.0
      POINTX=HRS+IFT*TAN(THETAS)+IFT/COS(THETAS)
      POINTY=TSK+THICK
      EFOMLR=RAD+EFT/2.0
      TEMPY=EFOMLR-EFOMLR*SIN(THETAS)
      XBAR=POINTX-TEMPY*TAN(THETAS)
      YBAR=POINTY+TEMPY
      CEFX=XBAR+EFOMLR*COS(THETAS)
      CEFY=POINTY+EFOMLR
C
      RADR=RAD+TFLR/2.0
      XR=CEFX
      RADL=YBAR-YL
      XL=XBAR-RADL
C
      RETURN
      END

```

DISTRIBUTION LIST

No. of Copies

NAVAIRWARCENACDIVWAR.....	2
Code 8131	
NAVAIRWARCENACDIVWAR.....	10
Code 6043	

DISTRIBUTION LIST

No. of Copies

One Northrop Avenue
Hawthorne, CA 90250

Rockwell International/North American Aircraft Division..... 1
Attn: Technical Library
P.O. Box 92098
Los Angeles, CA 90009

Rockwell International/North American Aircraft Operations.....1
Attn: Gary G. Steward
P.O. Box 92098
Los Angeles, CA 90009

Rockwell International/North American Aircraft 1
Attn: Dr. T.R. Logan
P.O. Box 7009
Downey, CA 90241-7009

Rockwell International/North American Aircraft Operations..... 1
Attn: Technical Library
P.O. Box 582808
Tulsa, OK 74158

Rockwell International/North American Aircraft Operations..... 1
Attn: Mr. F. Kaufman
P.O. Box 582808
Tulsa, OK 74158

Rockwell International/Tulsa Division..... 1
Attn: Mr. G. Sherrick
P.O. Box 51308
Tulsa, OK 741511

Sikorsky Aircraft..... 1
Attn: Technical Library
North Main Street
Stratford, CT 06601-1381

Sikorsky Aircraft..... 1
Attn: Mr. Samuel P. Garbo, M/S S314A2
North Main Street
Stratford, CT 06601-1381

Textron Specialty Materials..... 1
Attn: Mr. William F. Grant
2 Industrial Avenue
Lowell, MA 01851

NAVAIRWARCENACDIVWAR..... 1
Tom Hess, Code 60C1

NAVAIRWARCENACDIVWAR..... 1
Code 6064

DISTRIBUTION LIST

No. of Copies

McDonnell Douglas Aerospace.....	1
Attn: Mr. R. Palmer	
3855 Lakewood Blvd.	
Long Beach, CA 90846	
McDonnell Douglas Aerospace.....	1
Attn: Mr. Amos Hoggard, M/C 36-90	
3855 Lakewood Blvd.	
Long Beach, CA 90846	
McDonnell Douglas Aerospace.....	1
Attn: Mr. A. Hawley, Dept. ELH, M/S 212-10	
3855 Lakewood Blvd.	
Long Beach, CA 90846	
McDonnell-Douglas Helicopter Company.....	1
Attn: Technical Library	
5000 E. McDowell Road	
Mesa, AZ 85205	
McDonnell-Douglas Astronautics.....	1
Attn: Technical Library	
5301 Bolsa Avenue	
Huntington Beach, CA 92647	
Northrop Aircraft Corporation.....	1
Attn: Technical Library	
One Northrop Avenue	
Hawthorne, CA 90250	
Northrop Aircraft Corporation.....	1
Attn: Dr. H.P. Kan	
One Northrop Avenue	
Hawthorne, CA 90250	
Northrop Aircraft Corporation.....	1
Attn: Dr. Ravi Deo	
One Northrop Avenue	
Hawthorne, CA 90250	
Northrop Aircraft Corporation.....	1
Attn: Dr. S.P. Agrawal	
One Northrop Avenue	
Hawthorne, CA 90250	
Northrop Aircraft Corporation.....	1
Attn: Dr. M. Ratwani	
One Northrop Avenue	
Hawthorne, CA 90250	
Northrop Aircraft Corporation.....	1
Attn: Dr. R. White	

DISTRIBUTION LIST

No. of Copies

LTV Aerospace and Defense Company.....	1
Vought Aeroproducts Division	
Attn: Technical Library	
P.O. Box 655907	
Dallas, TX 75265-5907	
LTV Aerospace and Defense Company.....	1
Vought Missiles and Advanced Programs Division	
Attn: C. Foreman, Engineering Project Manager	
P.O. Box 655907	
Dallas, TX 75265-5907	
McDonnell Douglas Aerospace.....	1
Attn: Technical Library	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. H. Dill, M/S 1022137	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. C. Pingle, Dept 337, M/S 0644324	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. D.J. Dorr, Dept 390, M/S 1021310	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. R. Ehlen, Dept. 337, M/S 0644434	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. C. Saff, Dept. 337, M/S 0341160	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Mr. J. Goering, Dept. 337, M/S 0341160	
P.O. Box 516	
St. Louis, MO 63166	
McDonnell Douglas Aerospace.....	1
Attn: Technical Library, M/S 212-10	
3855 Lakewood Blvd.	
Long Beach, CA 90846	

DISTRIBUTION LIST

No. of Copies

Grumman Aerospace Corporation..... Aerospace and Electronics Group Attn: Mr. J. Bruno, B44-35 South Oyster Bay Road Bethpage, Long Island, NY 11714	1
Grumman Aerospace Corporation..... Aerospace and Electronics Group Attn: Mr. S. Dastin South Oyster Bay Road Bethpage, Long Island, NY 11714	1
Grumman Aerospace Corporation..... Aerospace and Electronics Group Attn: Technical Library South Oyster Bay Road Bethpage, Long Island, NY 11714	1
Lockheed/Fort Worth Division..... Attn: Technical Library P.O. Box 748 Fort Worth, TX 76101	1
Lockheed/Fort Worth Division..... Attn: Dr. G. Law P.O. Box 748 Fort Worth, TX 76101	1
Lockheed-Georgia Company..... Attn: Technical Library 86 South Cobb Drive Marietta, GA 30063	1
Lockheed-Georgia Company..... Attn: Technical Information Dept. 72-34, Zone 26 Marietta, GA 30063	1
Lockheed Missile and Space Company..... Attn: Technical Library 1111 Lockheed Way Sunnyvale, CA 94088	1
LTV Aerospace and Defense Company..... Vought Aeroproducts Division Attn: Mr. J. Pimm, M/S 194-51 P.O. Box 655907 Dallas, TX 75265-5907	1

DISTRIBUTION LIST

No. of Copies

Boeing Defense & Space Group.....	1
Helicopter Division	
Attn: Dr. C.K. Gunther, M/S P30-30	
P.O. Box 16858	
Philadelphia, PA 19142-0858	
Boeing Defense & Space Group.....	1
Helicopter Division	
Attn: Technical Library	
P.O. Box 16858	
Philadelphia, PA 19142-0858	
Boeing Defense & Space Group.....	1
Attn: Mr. J. Howitt, Jr. M/S 4H-79	
P.O. Box 3707	
Seattle, WA 98124-2207	
Boeing Defense & Space Group.....	1
Attn: Mr. J. Quinlivan, M/S 6N-21	
P.O. Box 3707	
Seattle, WA 98124-2207	
Boeing Defense & Space Group.....	1
Attn: Dr. W.J. Renton, M/S 4H-79	
P.O. Box 3707	
Seattle, WA 98124-2207	
Boeing Defense & Space Group	1
Attn: Technical Library	
P.O. Box 20746	
Wichita, KS 67277-7730	
Boeing Defense & Space Group	1
Attn: Mr. J. Avery	
P.O. Box 20746	
Wichita, KS 67277-7730	
General Electric Company.....	1
Attn: Technical Library	
1 Neumann Way	
Cincinnati, OH 45215	
General Electric Company.....	1
Attn: Technical Library	
P.O. Box 8555	
Philadelphia, PA 19101	
General Electric Company.....	1
Attn: Mr. C. Zweben	
P.O. Box 8555-M4018	
Philadelphia, PA 19101	

DISTRIBUTION LIST

No. of Copies

FAA Atlanta ACO.....	1
1669 Phoenix Parkway	
Suite 210C	
Atlanta, GA 30349	
FAA Wichita ACO.....	1
1801 Airport Road	
Room 100	
Wichita, KS 67209	
FAA Chicago ACO.....	1
2300 E. Devon, Room 232	
Des Plains, IL 60011	
FAA Airplane Certification Office.....	1
4400 Bluemound Rd. ASW-150	
Fort Worth, TX 76193	
Beech Aircraft Corporation.....	1
Attn: Mr. M.P. Djuric	
Kellogg St. and Webb Road	
Wichita, KS 67201	
Bell Helicopter/Textron Inc.....	1
Attn: Mr. M.K. Stevenson, M/S 12	
P.O. Box 482	
Fort Worth, TX 76101	
Bell Helicopter/Textron Inc.....	1
Attn: Technical Library	
P.O. Box 482	
Fort Worth, TX 76101	
Bell Helicopter/Textron Inc.....	1
Attn: Mr. D. Reisdorfer	
P.O. Box 482	
Fort Worth, TX 76101	
Boeing Defense & Space Group.....	1
Helicopter Division	
Attn: W. Kesack	
P.O. Box 16858	
Philadelphia, PA 19142-0858	
Boeing Defense & Space Group.....	1
Helicopter Division	
Attn: Mr. R.L. Pinckney	
P.O. Box 16858	
Philadelphia, PA 19142-0858	

DISTRIBUTION LIST

	No. of Copies
FAA Technical Center.....	25
Attn: D. Oplinger	
MS ACD-210, Bldg. 201A	
Atlantic City Int'l Airport, NJ 08405	
FAA Technical Center.....	57
Attn: Wm. Bagot	
MS ACD-205, Bldg. 210	
Atlantic City Int'l Airport, NJ 08405	
FAA Field Engineering Office - ACD-20.....	1
6 E. Taylor St., MS 250	
NASA/Langley Research Center	
Hampton, VA 23665-5225	
FAA NASA/Ames Research Center.....	1
Attn: Barry Scott, ACD-30	
P.O. Box 25	
Moffitt Field, CA 94035	
FAA Civil Aeromedical Inst.	1
AAM-400A	
P.O. Box 25082	
Oklahoma City, OK 73125	
FAA Boston ACO.....	1
12 New England Exec. Park	
Burlington, MA 01803-5299	
FAA New York ACO.....	1
181 S. Franklin Avenue	
Room 202	
Valley Stream, NY 11581	
FAA Anchorage ACO.....	1
605 W. 4th Avenue	
Room 214	
Anchorage, AK 99513-7587	
FAA Seattle ACO.....	1
1601 Lind Avenue, S.W.	
Renton, WA 98055-4056	
FAA Los Angeles ACO.....	1
3229 E. Spring Street	
Los Angeles, CA 90808-2425	
FAA Rotor Craft ACO.....	1
4400 Bluemound Rd. AEW-170	
Fort Worth, TX 76193	

DISTRIBUTION LIST

No. of Copies

Administrator..... National Aeronautics and Space Administration Langley Research Center Attn: Mr. J.W. Deaton, M/S 188A Hampton, VA 23665-5225	1
Administrator..... National Aeronautics and Space Administration Langley Research Center Attn: Mr. J. Davis, M/S 243/STPO Hampton, VA 23665-5225	1
Administrator..... National Aeronautics and Space Administration Langley Research Center Attn: Dr. G.L. Roderick Hampton, VA 23665-5225	1
Administrator..... National Aeronautics and Space Administration George C. Marshall Space Flight Center Attn: Technical Library Huntsville, AL 35812	1
Administrator..... National Aeronautics and Space Administration Lewis Research Center Attn: Technical Library 21000 Brookpark Road Cleveland, OH 44153	1
Administrator..... National Aeronautics and Space Administration Lewis Research Center Attn: Dr. C. Chamis, M/S 49-6 21000 Brookpark Road Cleveland, OH 44153	1
Administrator..... Defense Technical Information Center Bldg. #5, Cameron Station Alexandria, VA 22314	2
Federal Aviation Administration..... Attn: AIR-121 800 Independence Avenue, S.W. Washington, D.C. 20591	1
Federal Aviation Administration..... Attn: Mr. J.R. Soderquist, AIR-103 800 Independence Avenue, S.W. Washington, D.C. 20591	1

DISTRIBUTION LIST

No. of Copies

Commanding Officer..... 1
 U.S. Army Air Mobility R&D Lab
 Attn: H. Reddick
 Fort Eustis, VA 23604

Commanding Officer..... 1
 U.S. Army Aviation Applied Technology Directorate
 Attn: T.E. Condon
 SAVRT/TY-ASR
 Fort Eustis, VA 23604-5577

Commanding Officer..... 1
 U.S. Army Aviation Applied Technology Directorate
 Attn: Drew Orlino
 SAVRT/TY-ATS
 Fort Eustis, VA 23604-5577

Commanding Officer..... 1
 Advanced Systems Research and
 Analysis Office-ASRAO
 Attn: Library, M/S 219-3
 Ames Research Center
 Moffett Field, CA 94035-1000

NASA Headquarters..... 1
 Attn: G. Seidel
 OAST-Code RM
 Washington, D.C. 20546

Adminstrator
 National Aeronautics and Space Administration..... 1
 Attn: Airframes Branch, FS 120
 Washington, D.C. 20546

Administrator..... 1
 National Aeronautics and Space Administration
 Langley Research Center
 Attn: Dr. C.P. Blankenship, M/S-188M
 Hampton, VA 23665-5225

Administrator..... 1
 National Aeronautics and Space Administration
 Langley Research Center
 Attn: Dr. J. Starnes, M/S 190
 Hampton, VA 23665-5225

Administrator..... 1
 National Aeronautics and Space Administration
 Langley Research Center
 Attn: Mr. W.T. Freeman, M/S 243
 Hampton, VA 23665-5225

DISTRIBUTION LIST

No. of Copies

Commander.....	1
U.S. Naval Postgraduate School	
Attn: Technical Library	
Monterey, CA 93943	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIBAC, R. Holzwarth	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIBAC, L. R. Phillips	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIB, D. Paul	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIBA, G. Holderby	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIBC, C. Ramsey	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: FIBEC, Dr. G. Sendeckyj	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: MLSE, T. Reinhart	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Wright Laboratory	
Attn: MLTN, R.C. Tomashot	
Wright-Patterson Air Force Base, OH 45433-6553	
Commanding Officer.....	1
Picatinny Arsenal	
PLASTECH	
Attn: Librarian, Code DRDAR-SCM-0, Bldg. 351N	
Dover, NJ 07801	

DISTRIBUTION LIST

No. of Copies

Commander.....	1
Naval Surface Weapons Center	
Attn: Dr. J.M. Augl	
10901 New Hampshire Avenue	
Silver Spring, MD 20903-5000	
Commander.....	1
Naval Surface Weapons Center	
Attn: Dr. J. Goff, R34	
White Oak Laboratory	
Silver Spring, MD 20910	
Commander.....	1
Naval Aviation Depot	
Attn: J. Myers, Code 35110	
MCAS, Cherry Point, NC 28533-5030	
Commander.....	1
Naval Aviation Depot	
Attn: J. Fuss, Code 35430	
MCAS, Cherry Point, NC 28533-5030	
Commander.....	1
Naval Aviation Depot	
Attn: J. Gresham, Code 35210	
MCAS, Cherry Point, NC 28533-5030	
Commander.....	1
Naval Aviation Depot	
Attn: M. Linn, Code 340	
NAS, Jacksonville, FL 32212-0016	
Commander.....	1
Naval Aviation Depot	
Attn: D. Perl, Code 343	
NAS, North Island, San Diego, CA 92135-5112	
Commander.....	1
Naval Aviation Depot	
Attn: M. Williams, Code 31327	
NAS, North Island, San Diego, CA 92135-5112	
Commander.....	1
Naval Aviation Depot	
Attn: D. Knapp, Code 342	
NAS, Pensacola, FL 32508-5300	
Commander.....	1
United States Naval Academy	
Attn: Mechanical Engineering Dept.	
Annapolis, MD 21402	

DISTRIBUTION LIST

No. of Copies

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-5304
 Washington, D.C. 20361

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-530T
 Washington, D.C. 20361

Commander..... 1
 Naval Sea Systems Command
 Attn: Sea-05R
 Washington, D.C. 20362-5101

Commander..... 1
 Naval Air Warfare Center Weapons Division
 Attn: K. Bailey, Code 338
 China Lake, CA 93355

Commanding Officer..... 1
 Naval Ship Engineering Center
 Attn: NSEC 6101E
 Arlington, VA 20360

Commander..... 1
 David Taylor Research Center
 Attn: R. Rockwell, Code 1720-6
 Bethesda, MD 20084

Commander..... 1
 David Taylor Research Center
 Attn: A. Macander, Code 1720
 Bethesda, MD 20084

Commander..... 1
 David Taylor Research Center
 Attn: E.T. Camponeschi, Code 2844
 Annapolis, MD 21402

Commander..... 1
 David Taylor Research Center
 Attn: R. Crane, Code 2802
 Annapolis, MD 21402

Commander..... 1
 David Taylor Research Center
 Attn: H. Edelstein, Code 2870
 Annapolis, MD 21401

DISTRIBUTION LIST

No. of Copies

Office of Naval Technology..... 1
 Attn: W. King, ONT-212
 800 N. Quincy Street
 Arlington, VA 22217

Office of Naval Research..... 1
 Attn: A.K. Vasudevan, Code 1216
 800 N. Quincy Street
 Arlington, VA 22217

Office of Naval Research..... 1
 Attn: Y. Rajapakse, Code 1132SM
 800 N. Quincy Street
 Arlington, VA 22217

Director..... 1
 Naval Research Laboratory
 Attn: L. Gause
 4555 Overlook Avenue, S.W.
 Washington, D.C. 20375-5000

Director..... 1
 Naval Research Laboratory
 Attn: Dr. I. Wolock, Code 6383
 4555 Overlook Avenue, S.W.
 Washington, D.C. 20375-5000

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-530
 Washington, D.C. 20361

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-5302
 Washington, D.C. 20361

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-53021
 Washington, D.C. 20361

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-53022
 Washington, D.C. 20361

Commander..... 1
 Naval Air Systems Command
 Attn: AIR-53023
 Washington, D.C. 20361