

AD 744505

AD

USAAMRDL TECHNICAL REPORT 72-17
A TETRA-CORE STRESS ANALYSIS MODEL

By

Alan Lee Dobyns
David C. Jack

April 1972

EUSTIS DIRECTORATE
U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
FORT EUSTIS, VIRGINIA

CONTRACT DAAJ02-71-C-0056
THE BOEING COMPANY
SEATTLE, WASHINGTON

Approved for public release;
distribution unlimited.



Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U.S. Department of Commerce
Springfield, VA 22151



261

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the US Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Trade names cited in this report do not constitute an official endorsement or approval of the use of such commercial hardware or software.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

ACCESSION FOR	
CPSTI	WHITE SECTION ☒
DDC	BUFF SECTION ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODE	
DIST.	AVAIL. AND/OR SPECIAL
A	



DEPARTMENT OF THE ARMY
U. S. ARMY AIR MOBILITY RESEARCH & DEVELOPMENT LABORATORY
EUSTIS DIRECTORATE
FORT EUSTIS, VIRGINIA 23604

This study was performed under Contract DAAJ02-71-C-0056 with The Boeing Company, Seattle, Washington. The technical monitor for this contract was Mr. I. E. Figge, Structures Division.

This report contains the results of a study to develop a three-dimensional finite-element computer program for "Tetra-Core" elements.

This report has been reviewed by the Eustis Directorate, US Army Air Mobility Research and Development Laboratory and is considered to be technically sound. It is published for the exchange of information and the stimulation of future research.

ic

TASK 1F162203A17003
Contract DAAJ02-71-C-0056
USAAMRDL Technical Report 72-17
April 1972

A TETRA-CORE STRESS ANALYSIS MODEL

D180-14257-1

By

Alan Lee Dobyms
David C. Jack

Prepared by

The Boeing Company
Seattle, Washington

for

EUSTIS DIRECTORATE
U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
FORT EUSTIS, VIRGINIA

Approved for public release; distribution unlimited.

ABSTRACT

The results of a five-month analytical program are presented, describing a stress analysis method for a three-dimensional fiber composite structure. The intersections of the elements or webs of this structure form polyhedra in the form of triangular prisms, Tetra-core, and truncated forms of these.

The finite-element model generates the elements and polyhedra automatically with an optimization routine which selects the minimum weight structure that will meet the stiffness and loads requirements.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author)		25. REPORT SECURITY CLASSIFICATION
The Boeing Company Seattle, Washington		Unclassified
		26. GROUP
3. REPORT TITLE		
A TETRA-CORE STRESS ANALYSIS MODEL		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
Final Report		
5. AUTHOR(S) (First name, middle initial, last name)		
Alan L. Dobyns David C. Jack		
6. REPORT DATE	7A. TOTAL NO. OF PAGES	7B. NO. OF REV
April 1972	258	8
8A. CONTRACT OR GRANT NO.	8B. ORIGINATOR'S REPORT NUMBER(S)	
DAAJ02-71-C-0056	USAAMRDL Tech Report 72-17	
a. PROJECT NO.	9B. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
Task 1F162203A17003	D180-14257-1	
c.		
d.		
10. DISTRIBUTION STATEMENT		
Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY
		Eustis Directorate, U.S Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia
13. ABSTRACT		
The results of a five-month analytical program are presented, describing a stress analysis method for a three-dimensional fiber composite structure. The intersections of the elements or webs of this structure form polyhedra in the form of triangular prisms, Tetra-core, and truncated forms of these. The finite element model generates the elements and polyhedra automatically with an optimization routine which selects the minimum weight structure that will meet the loads requirement.		

DD FORM 1473

REPLACES DD FORM 1473, 1 JAN 60, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

1a

Unclassified
Security Classification

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Tetra-Core Fiber Composite Stress Analysis Structural Optimization						

Unclassified
Security Classification

FOREWORD

This report was prepared by The Boeing Company, Seattle, Washington, under U. S. Army Contract DAAJ02-71-C-0056 (DA Task 1F162203A17003), and contains an analytical stress analysis model of a three-dimensional fiber composite, Tetra-Core. The contract was started 28 May 1971 and was completed 17 December 1971. Project Engineer for this program was Mr. I. E. Figge, Sr., Structures Division, Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory.

The authors acknowledge the following Boeing personnel for their support: David L. Beste and Robert Blaisdell.

Preceding page blank

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT.....	iii
FOREWORD.....	v
LIST OF ILLUSTRATIONS.....	ix
INTRODUCTION.....	1
MODEL DESCRIPTION.....	7
INPUTS.....	7
AUTOMATIC MODEL GENERATOR.....	22
Flat Plate Nodal Generator.....	22
Cylinder Nodal Generator.....	23
Airfoil Nodal Generator.....	23
Plate Element Generator.....	24
Flat Plate.....	24
Cylinder and Airfoil.....	25
Flat Plate Loads.....	25
Cylinder Loads Generation.....	28
Airfoil Loads Generation.....	29
Nodal Fixity.....	29
FINITE ELEMENT STRESS ANALYSIS.....	31
Linear Strain Element.....	31
Quadrilateral Element.....	39
Constant Strain Triangle.....	39
Consistent Surface Loads Vector.....	40
Calculation of Deflections.....	41
OPTIMIZATION.....	45
NONLINEAR ANALYSIS.....	53
BUCKLING ANALYSIS.....	57
MARGIN OF SAFETY.....	58
CALCULATION OF ULTIMATE LOADS.....	59
CALCULATION OF STIFFENESSES.....	59
OUTPUT DESCRIPTION.....	61
SAMPLE PROBLEMS.....	65
COMPARISON WITH UNIVERSITY OF KANSAS TESTS.....	69
MODEL CONSTRAINTS.....	70
COMPUTER PROGRAM.....	71
LIST OF SUBROUTINES.....	90

Preceding page blank

TABLE OF CONTENTS (Cont)

	<u>Page</u>
DESCRIPTION OF VARIABLES USED IN COMMON.....	92
DESCRIPTION OF VARIABLES USED IN SUBROUTINES.....	95
LITERATURE CITED.....	178
GLOSSARY.....	179
APPENDIXES.....	181
I. ADDITIONAL GENERATION EXAMPLES.....	181
II. EXAMPLE OF COMPUTER PROGRAM OUTPUTS.....	204
DISTRIBUTION.....	251

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Typical Applications of Tetra-Core.....	2
2	Basic Tetra-Core Element.....	3
3	Sequence of Operation in Tetra-Core Analysis Program.....	6
4	Basic Tetrahedron Geometry.....	8
5	Tetra-Core Input Geometry.....	9
6	Material Properties Required as Input.....	11
7	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	12
8	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	13
9	Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	14
10	Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	15
11	Skew A Plates in Cylinder (Computer Plot).....	16
12	Flat Plate Applied Loads.....	18
13	True Tetra-Core Cylinder With Torsion Load Applied, $L_X = 10$	20
14	True Tetra-Core Airfoil With Torsion Load Applied, $L_X = 20$	21
15	Structural Stiffness Matrix.....	42
16	Design Space.....	46
17	Two-Dimensional Optimization Example.....	48
18	Composite Constraint Gradient.....	48
19	Constraint Surfaces of Three-Dimensional Problem..	50
20	Global Optimum.....	50

LIST OF ILLUSTRATIONS (Cont)

<u>Figure</u>		<u>Page</u>
21	Example of Tetra-Core Optimization.....	52
22	Method Used To Determine Moduli for Plastic Analysis.....	54
23	Example of Nonlinear Analysis.....	56
24	Elements Used in Tetra-Core Program.....	61
25	Added Mid-Side Nodes, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	62
26	Input Data for Nonlinear Analysis.....	66
27	Input Data for Airfoil Analysis.....	67
28	Input Data for Optimization Run.....	68
29	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	182
30	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 20$, $L_Y = 6$	183
31	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 6$	184
32	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 8$, $L_Y = 4$	185
33	Vertical Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$	186
34	Skew A Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$	187
35	Skew B Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$	188
36	Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	189
37	Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	190
38	Load N_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	191

LIST OF ILLUSTRATIONS (Cont)

<u>Figure</u>		<u>Page</u>
39	Load N_y Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$	192
40	In-Plane Shear Load N_{xy} Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$	193
41	Moment M_x Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$	194
42	Moment M_y Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$	195
43	Twisting Moment M_{xy} Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$	196
44	Vertical Shear Load $XQSHR$ Applied to Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	197
45	Vertical Shear Load $YQSHR$ Applied to Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	198
46	Load N_x Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	199
47	Load N_y Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	200
48	In-Plane Shear Load N_{xy} Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	201
49	Moment M_x Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	202
50	Moment M_y Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$	203
51	Output From Optimization Run.....	205

INTRODUCTION

This document contains a description of a three-dimensional finite-element stress analysis computer model developed to provide a tool for defining the geometry of Tetra-core structures. The model uses material properties and applied loads in determining the final composite configuration. The primary polyhedron shape is a tetrahedron; one form of the model can produce a Tetra-core, where all elements intersect to form an equilateral tetrahedron.

The objective of the program was to develop a finite-element stress model that would provide a method of analysis of a complex nonhomogenous structure. The model was developed to run on the IBM 360/44. The size of problem that can be run is limited only by the number and size of high-speed discs available.

Tetra-core is a new construction concept developed for use on helicopters, airplanes, and other vehicles where lightweight and damage-tolerant structures are desirable. A typical application is shown in Figure 1. It is made by filament winding fiber glass or advanced composite fibers into an interlocking group of tetrahedrons, as shown in Figure 2. This arrangement appears to give a lightweight, truss-type structure that will be able to sustain damage and still carry a load, because of the high degree of structural indeterminacy. Tetra-core is constructed using a standard filament-winding machine. The fibers are laid in legs oriented in one direction, then the fibers of the next direction are laid over them, thus providing good load transfer between legs. Adhesive is applied to the fiber as it is being wound. The final composite is then ready for the curing process.

Several stress analysis methods have been considered for use with Tetra-core. A simple stress analysis, in which the load in each leg is resolved into its components by a summation of forces, does not account for the structural indeterminacy of Tetra-core, and will probably give conservative results.

The classical laminated plate theory widely used with boron and graphite-layered panels does not apply to Tetra-core since it is not a homogenous material. Each leg of a Tetra-core can be analyzed using laminated plate theory, but a method that accounts for the redundancy caused by the interweaving of the legs should be used in determining the strength of a Tetra-core panel.

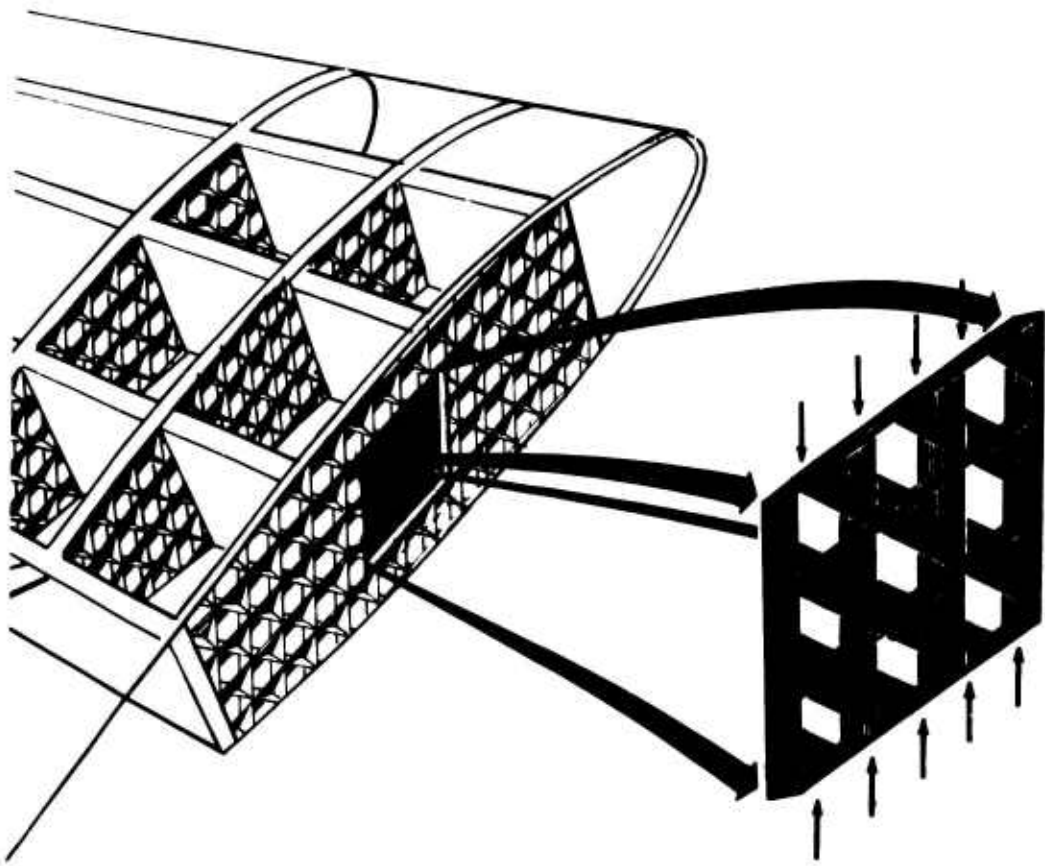


Figure 1. Typical Application of Tetra-Core.

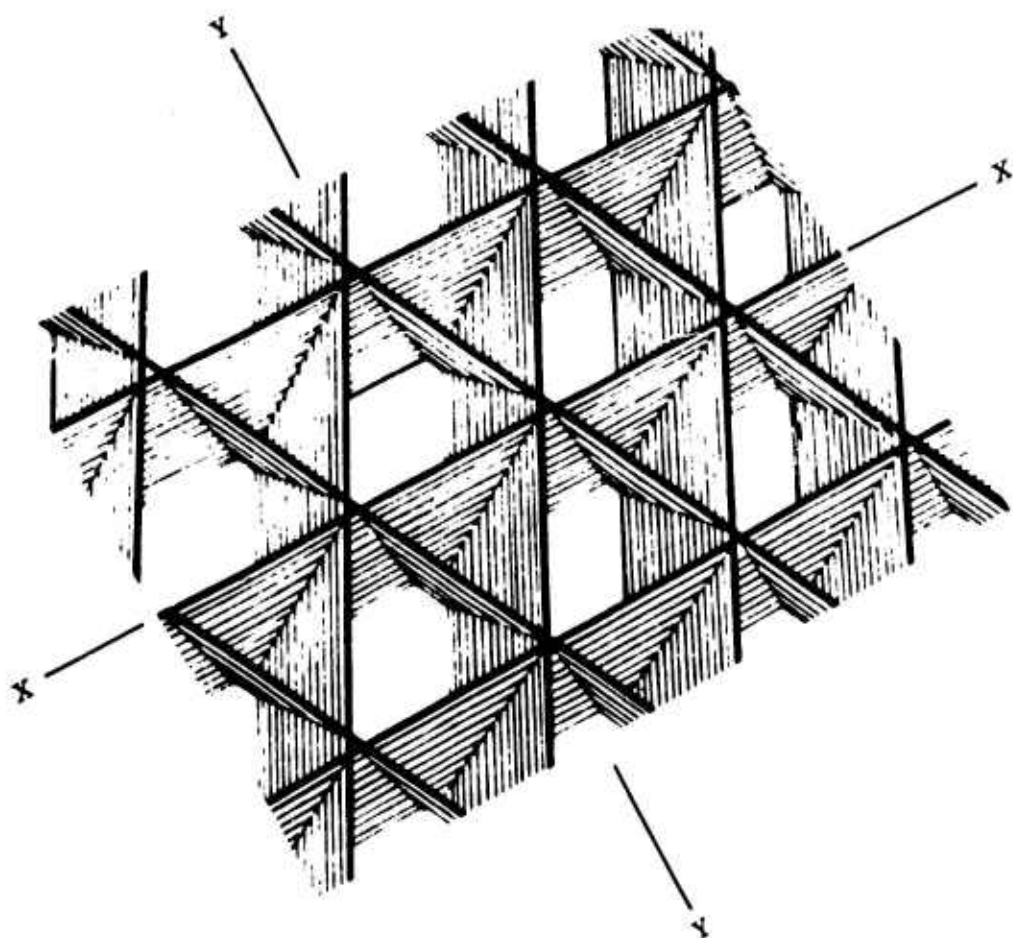


Figure 2. Basic Tetra-Core Element.

The finite-element method was used for this contract, along with an automatic grid generator to automatically idealize the structure and apply loads to it.

Two types of elements are used in this program: a linear strain triangle (6 nodes) for the legs of the Tetra-core, and a constant strain triangle (3 nodes) for the face sheets attached to the core. The linear strain triangle is required for the legs because of the change of strain from tension to compression between top and bottom of the tetrahedron when the panel is subjected to a bending moment. If a constant strain triangular element were used in this strain field, the result would be an averaging of the maximum and minimum stresses since the element would enforce a constant strain over the triangle. A constant strain triangle may be used for the face sheets, however, since the stress field does not contain changes of sign or large strain gradients. This allows a more efficient use of the available core storage space since additional nodes do not have to be generated at the midpoint of each side.

The program has been written to automatically generate finite-element models for Tetra-core structures made up of flat plates, cylinders, or airfoils. Two basic types of Tetra-core may be generated: a true Tetra-core, where the sides of the Tetra-core meet at a point to form a tetrahedron; and a truncated Tetra-core, in which the tops have been "cut off" of the tetrahedron. The basic tetrahedron geometry can be varied to give a skewed tetrahedron. Self-equilibrating nodal loads are automatically applied to the model based on the input combination of in-plane and out-of-plane loads. Multiple load cases may be run. Boundary conditions (nodal fixities) are applied to the model to hold it from translating or rotating without providing reactions for the nodal loads. This combination of loads and boundary conditions results in the idealization of a coupon of Tetra-core to which loads corresponding to those in some location of a larger structure are applied. A conventional finite-element analysis would be run of the total structure or a part of the structure. Stresses from this run would then be used to apply loads to the Tetra-core model corresponding to critical locations on the total structure.

Deflections are calculated using a Choleski triangularization method with substitution to solve the equation $[K] \{\delta\} = \{P\}$. Stresses and margin of safety are calculated at several locations in the linear strain elements. A buckling analysis of a triangle in each leg is done using the buckling equation of an orthotropic rectangle of the same size as a triangle in that leg, since no orthotropic triangle buckling equation are available. An optimization of the Tetra-core geometry can be made

using a steepest descent/side-step method to find a minimum weight design for the given loading conditions. A nonlinear analysis can be run using stress-strain curves of the material in each leg to account for the change in stiffness of each plate element with increasing load. The effect of a hole in a Tetra-core structure can be analyzed. The program will give a zero stiffness to plates connecting to a node at the center of the hole. This causes the loads to be redistributed into plates around the hole, creating the effect of a stress concentration.

The program is now configured for the IBM 360/44, but it can be easily modified to run on any 360. Converting to a CDC 6600 would be more difficult, since direct-access disc drives are required.

Figure 3 shows the elements of the program and the sequence of calculation. Each program element, input and output data, and model usage are described in the following chapters. This includes input format, output format, and a complete listing of the program.

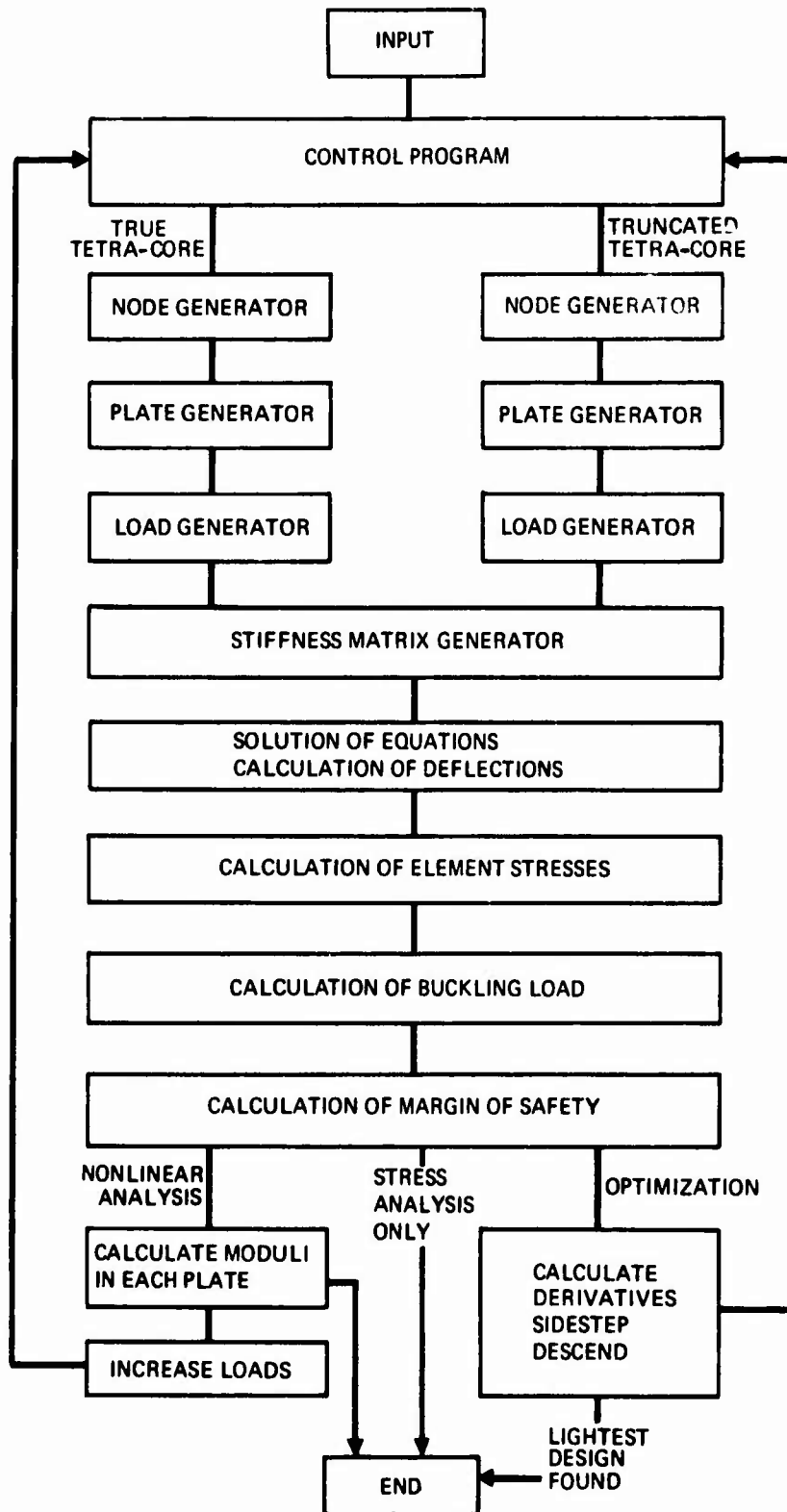


Figure 3. Sequence of Operations in Tetra-Core Analysis Program.

MODEL DESCRIPTION

This program has been written to automatically generate a finite-element model as a series of node points in space and connect them with plate elements into an idealization of a Tetra-core structure (Figure 1). It will automatically apply nodal loads to the model to represent uniform applied loads. Tetra-core flat plates, cylinders, and airfoils can be analyzed. The program uses the finite-element model it has generated to calculate stresses and deflections for each element and each node. A margin of safety is calculated for each plate element. A simple buckling analysis is performed on a plate in each leg. An optimization can be done to determine the least-weight Tetra-core design for a given set of loads. A nonlinear analysis can be performed by inputting a set of material stress-strain curves and applying loads in increments to compute the degradation in each plate element. The effect of a hole in the model can be simulated by giving a low stiffness to the plates connected to a specified node. A diagram of the sequence of computations performed by the program is given in Figure 3.

INPUTS

Input data required to run the Tetra-core analysis program has been kept to a minimum to simplify the work required for trade-off studies on the effect of geometry, materials, and other variables. Basic inputs are:

- o Type of analysis to be run--nonlinear, optimization, damage, face sheets, flat plate, cylinder, airfoil, and true or truncated tetrahedron
- o Optimization controls
- o Size of specimen to be analyzed and tetrahedron geometry
- o Material data--moduli and Poisson's ratio, allowable stresses, and stress-strain curves
- o Loading to be imposed on the model

Using these data, the program generates a set of node points and plates to model the Tetra-core element and applies the specified loadings as nodal loads.

The basic tetrahedron geometry, shown in Figures 4 and 5, is input as the length of a side, the actual tetrahedron height, the theoretical height at which the apexes would meet, and the offsets which determine the amount of deviation from a true tetrahedron.

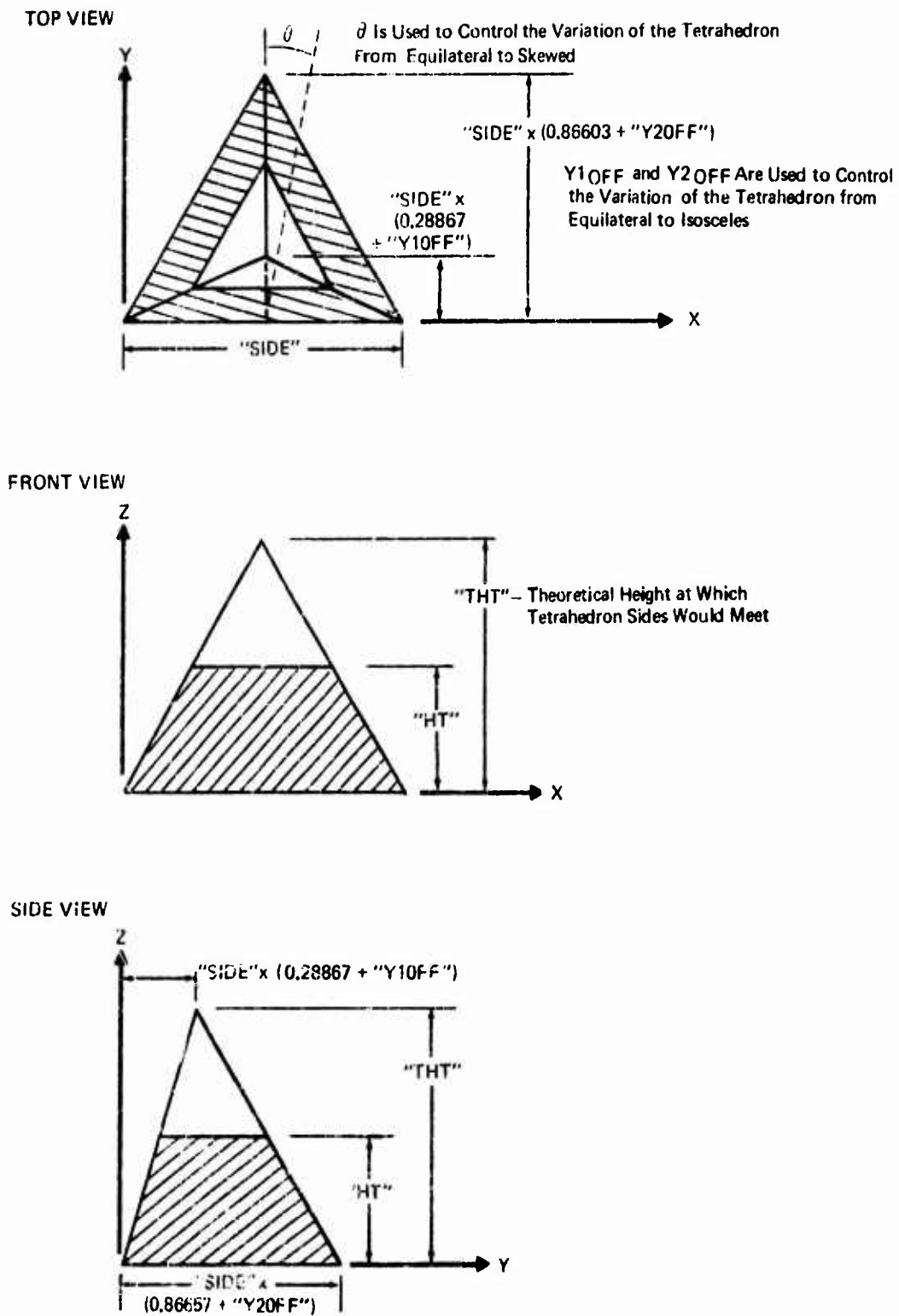


Figure 4. Basic Tetrahedron Geometry.

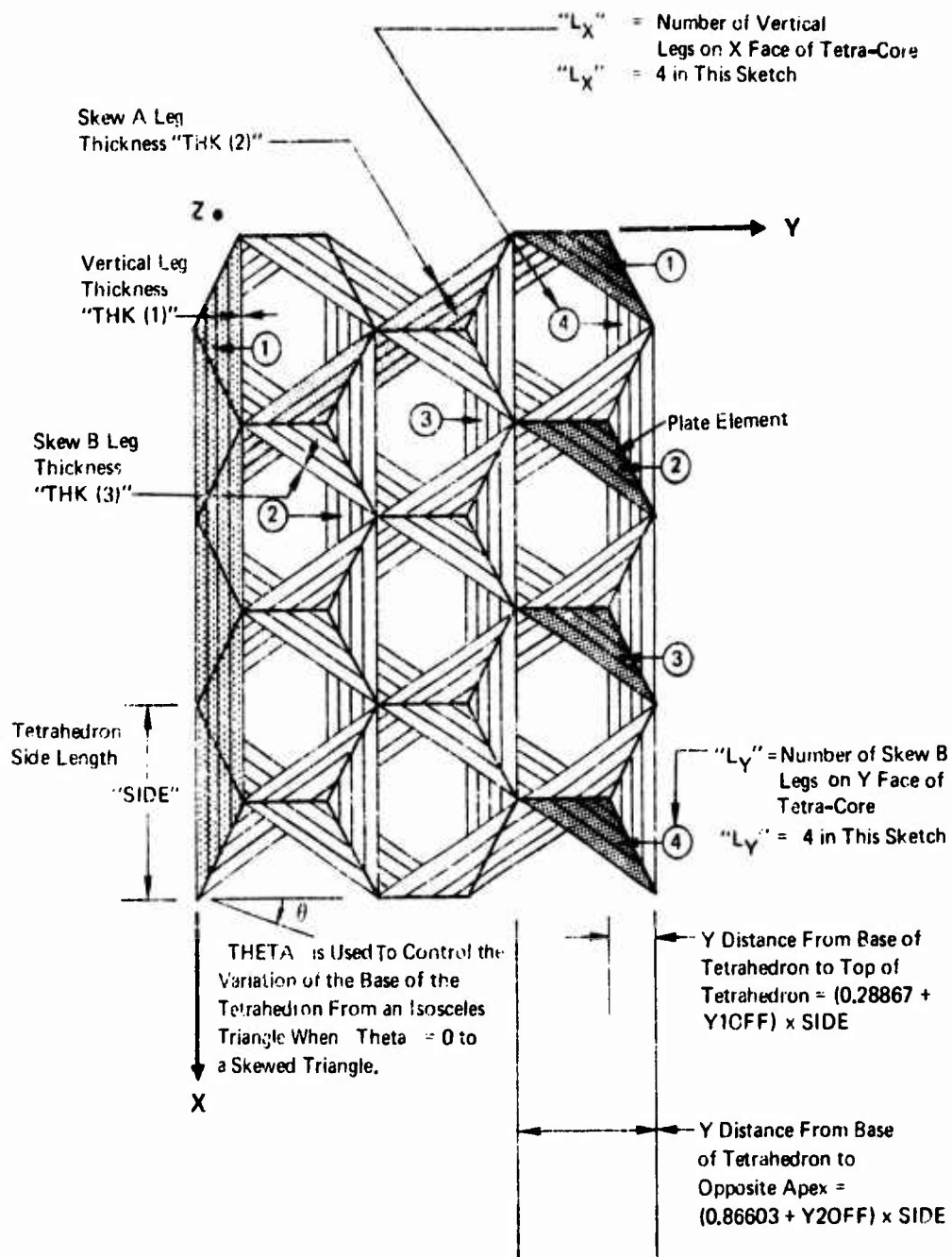


Figure 5. Tetra-Core Input Geometry.

If the actual height and the theoretical height are identical, then a true tetrahedron will result. Two generation options are available in the program: a true tetrahedron and a truncated tetrahedron. The true tetrahedron option should be used whenever possible, since it requires fewer nodal points and plates to be generated, and uses a more efficient nodal numbering system that allows a smaller bandwidth when used in a cylinder or an airfoil. If the theoretical height for a truncated tetrahedron is input as a large number (such as 10 inches), then a three-sided figure with near-vertical walls will result. The truncated Tetra-core model cannot be used to generate a true Tetra-core, since three node points at the top of a tetrahedron would have the same coordinates. This would cause a singular stiffness matrix and result in a bad solution. Offsets can be used to change the shape of the figure from a true tetrahedron with angles of 60 degrees, if no offsets are used, to an isosceles tetrahedron of any angle, if offset is used, to a skewed tetrahedron, if θ is used. The number of legs on the X and Y faces (LX and LY) must be even numbers (2, 4, 6, etc.).

Plates to connect the node points and form a Tetra-core model are generated so that the local X-axis is always in the direction of the fibers in that leg, as shown in Figure 6. For sandwich faces, the local X-axis is in the global X direction. Elastic moduli, allowable stresses, and stress-strain curves are input separately for each leg and each face sheet, thus allowing each leg to be a different material if desired. Material stress-strain curves are input only if a nonlinear analysis is run.

Using the input data LX and LY and the geometry, the tetrahedrons are automatically integrated into a Tetra-core node system as shown in Figure 7 for a true tetrahedron and in Figure 8 for a truncated tetrahedron. The node point numbering system is shown for each type. Other examples are shown in Appendix I.

The number of plates generated depends on the input number of legs. Plates are divided into Vertical, Skew A, and Skew B legs by the program. In Figures 9 and 10, the Skew A and B plates generated for a true tetrahedron with 10 legs on the X face and 6 legs on the Y face are shown. Plates are numbered in sequence with plates in the Vertical legs generated first, then plates in Skew A legs, then plates in Skew B legs. When cylinders and airfoils are generated, several additional plates are added to "zip up" the seam and connect the two sides with more Skew A and Skew B plates. Figure 11 shows the Skew A plates making up a cylinder. Face sheets can be automatically added to the true Tetra-core model to give a sandwich element. It was not considered feasible to generate a face sheet for the truncated Tetra-core element due to the complex shapes which would be required.

VERTICAL LEG

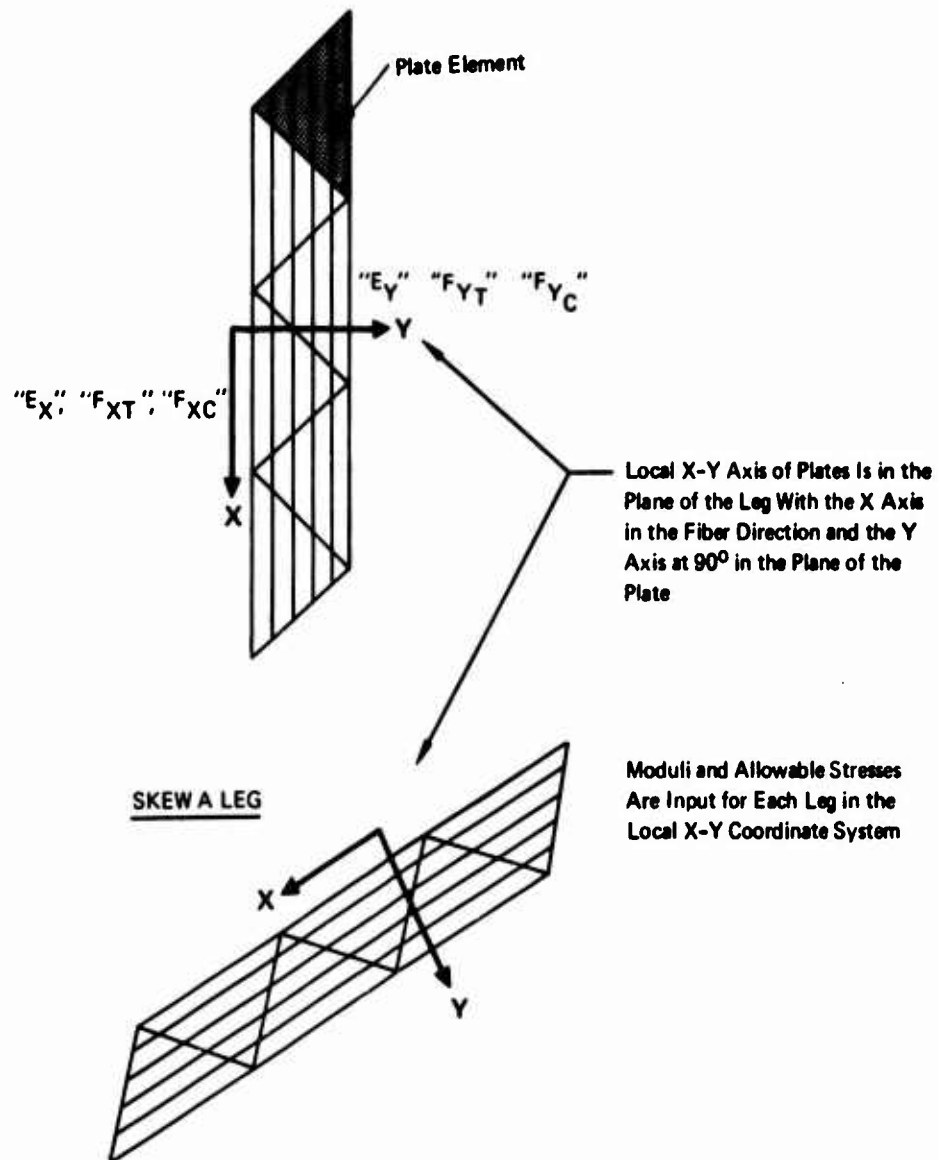


Figure 6. Material Properties Required as Input.

- Nodes Connected by Dark Lines Are on Upper Surface of Plate
- Nodes Connected by Light Lines Are on Lower Surface of Plate

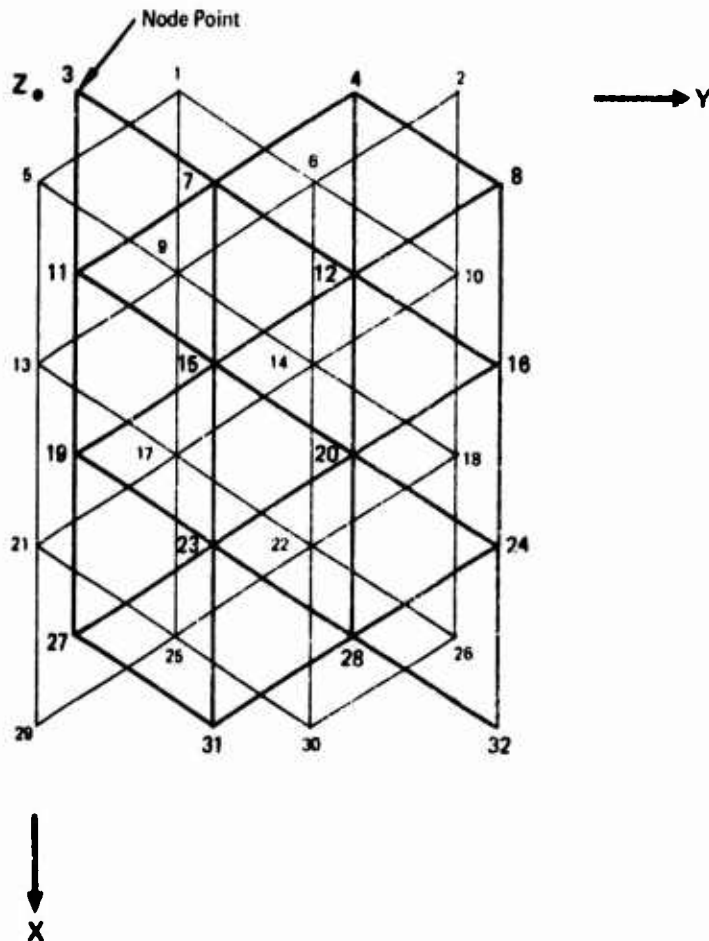


Figure 7. Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

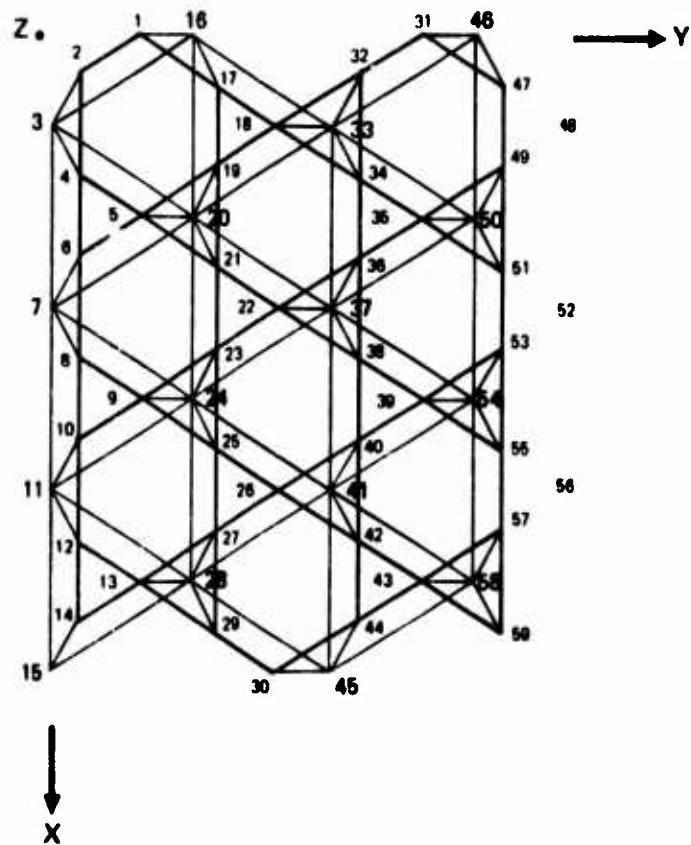


Figure 8. Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

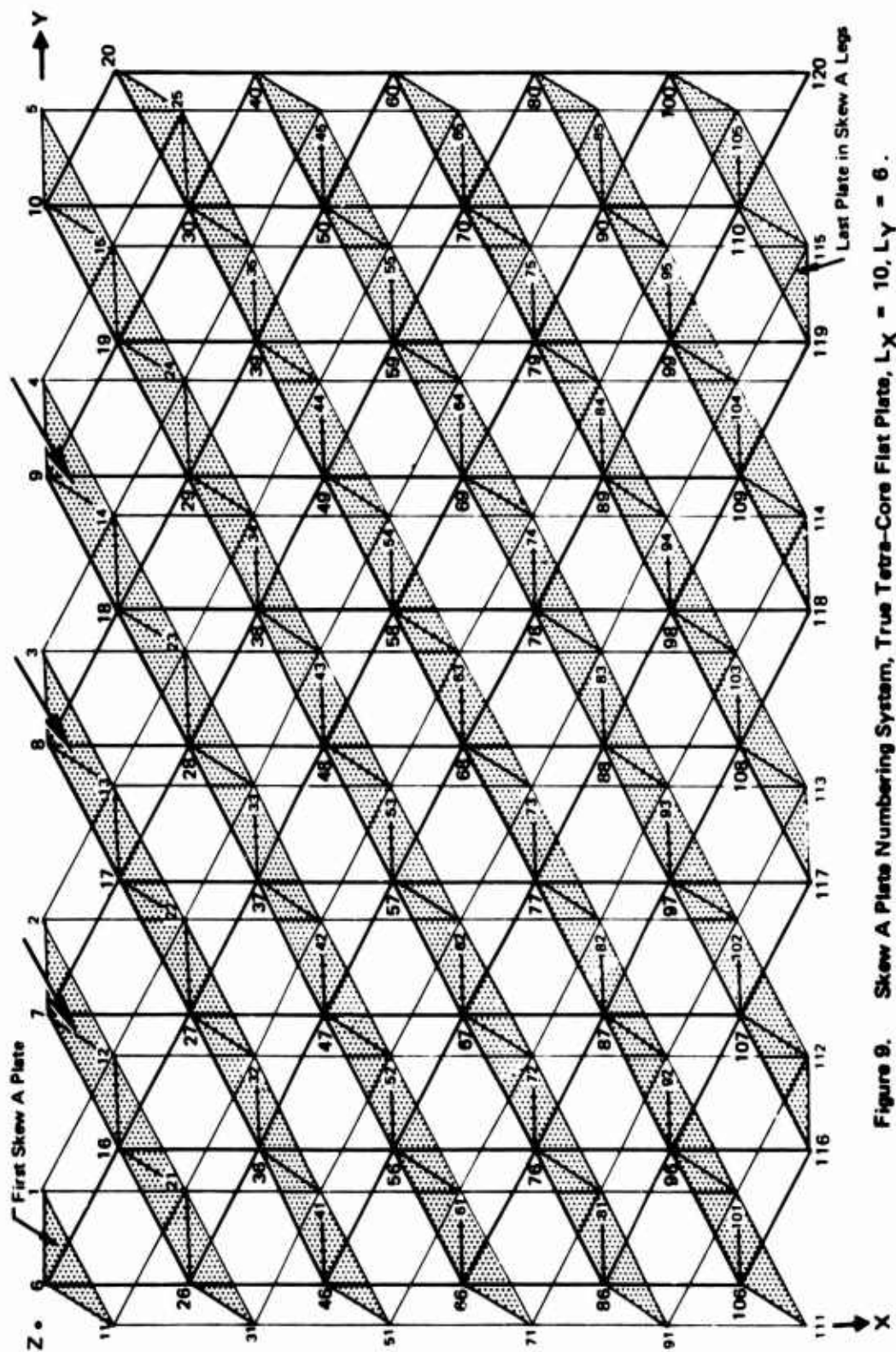
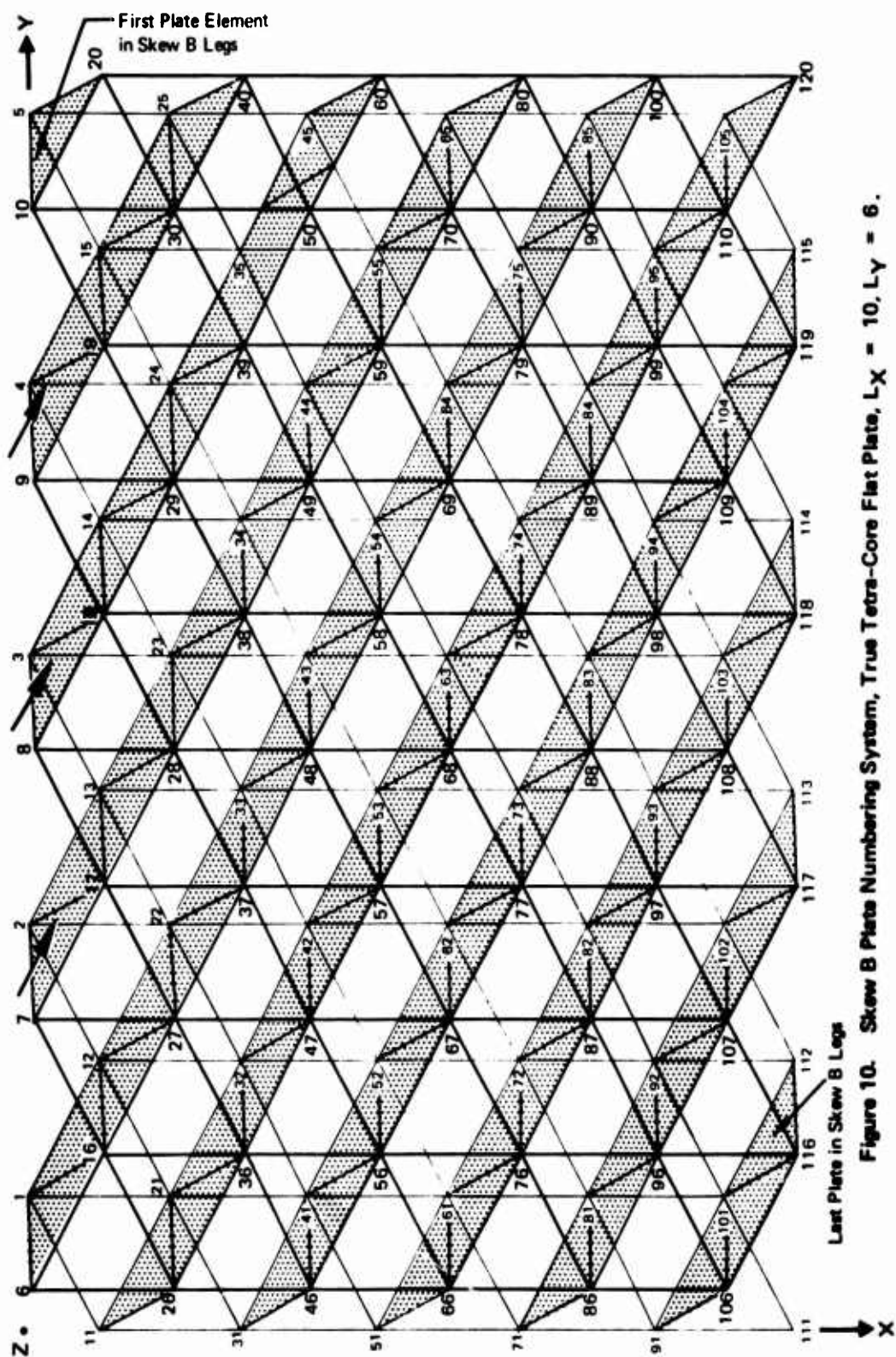


Figure 9. Skew A Plate Numbering System, True Tetra-Core Flat Plates, $L_x = 10$, $L_y = 6$.



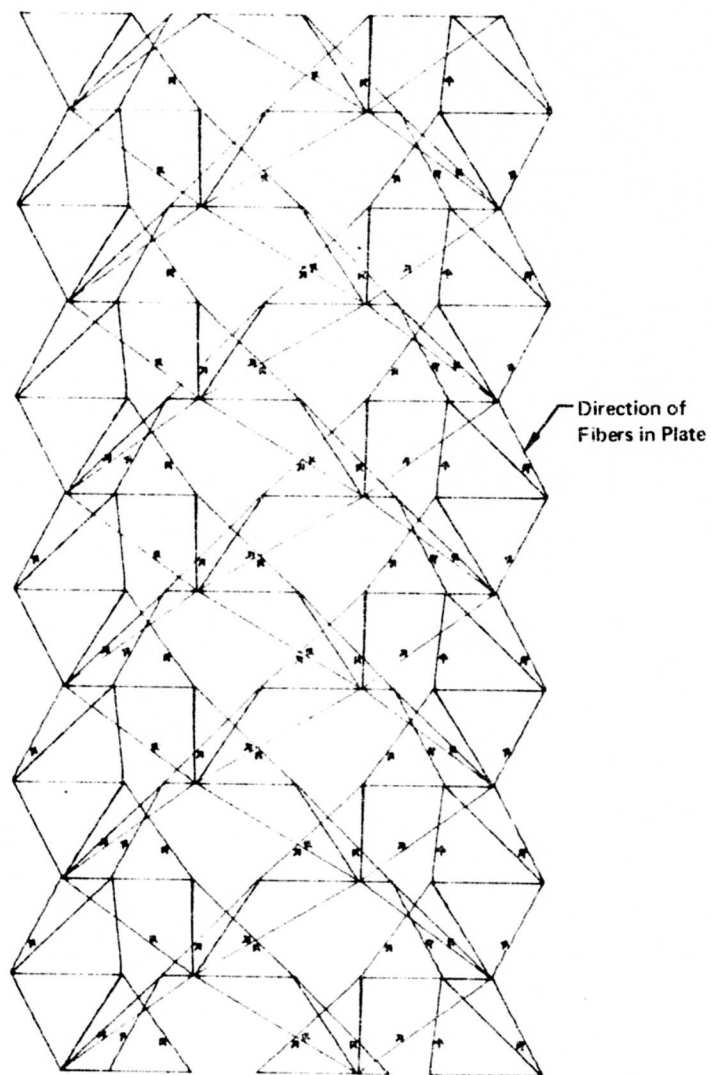


Figure 11. Skew A Plates in Cylinder (Computer Plot) .

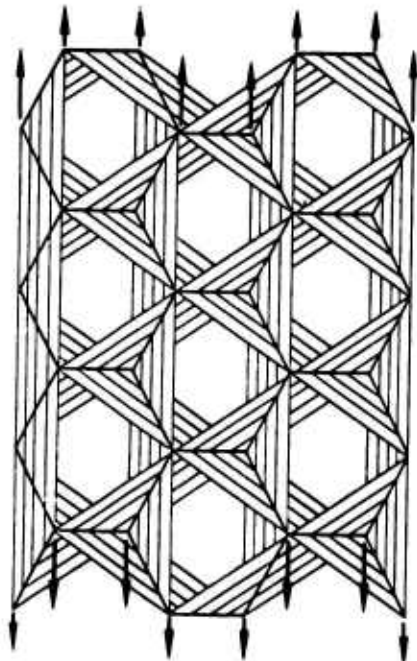
When generating a cylinder, the program transforms the flat plate Y and Z coordinates into a circular shape with a circumference equal to the Y dimension of the plate. Thus the outside radius of the cylinder is the X dimension of the plate divided by 2π . The airfoil section is generated with an input set of thickness/chord and X/chord ratios to define its shape. Vertical legs are defined to be parallel to the X-axis for cylinders and airfoils to give the maximum bending stiffness.

Node point loads are automatically applied to the finite-element model based on the input loading conditions. Loads are automatically applied in self-equilibrating sets; i.e., a load applied to one side is reacted by an equal and opposite load on the opposite side. Loading of the linear strain triangle, which is used for the Vertical, Skew A, and Skew B legs, includes a load on the mid-node point between the vertex node points. This load is four times the load at a vertex for a uniform applied load. The vertex node point loadings generated by inputting flat plate end loads in the X and Y directions and shear loads for the true tetrahedron case are shown in Figures 12a, b, and c. For moment loads, the load on the center node point of the triangle side is zero, since the two vertex loads cancel out. Node point loadings generated from input moments in the X, Y, and XY directions are shown in Figures 12d, e, and f for the true tetrahedron. Out-of-plane shear loads can be applied to the model. A moment is automatically applied to react the moment caused by the shear loads as shown in Figures 12g and h for vertical shear loads on the X and Y faces.

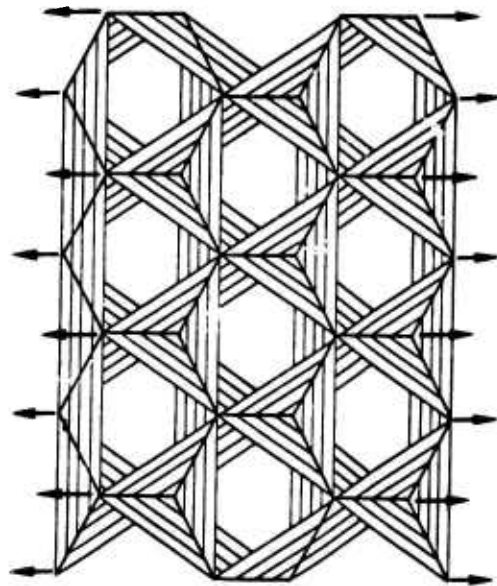
The truncated tetrahedron is loaded in a similar manner, except that the lower node points in an edge are given twice the load of the upper node points, since there are twice as many upper node points. The mid-side node point load is four times the upper node point load. More detailed examples of loads applied to Tetra-core flat plates are shown in Appendix I.

The cylinder model can be loaded with moment, torque, and end loads. The input load is used as the total load on the section and is divided among all nodes on that end. The nodal loads generated from an input torsion load on a cylinder are shown in Figure 13.

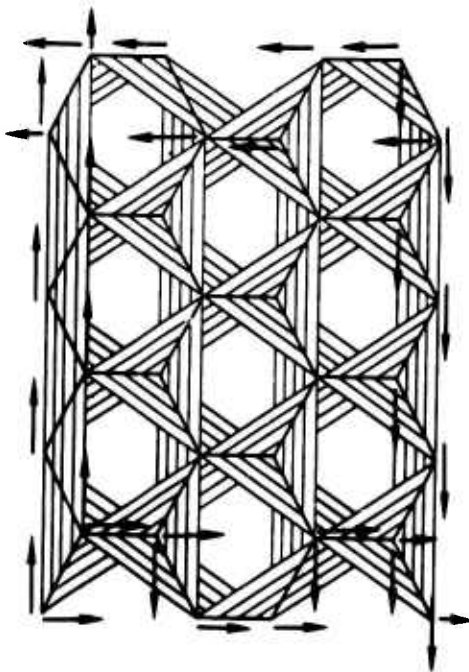
The airfoil model can be loaded with moment, torque, and end loads in a similar manner to the cylinder. A torsion load applied to a diamond airfoil section is shown in Figure 14.



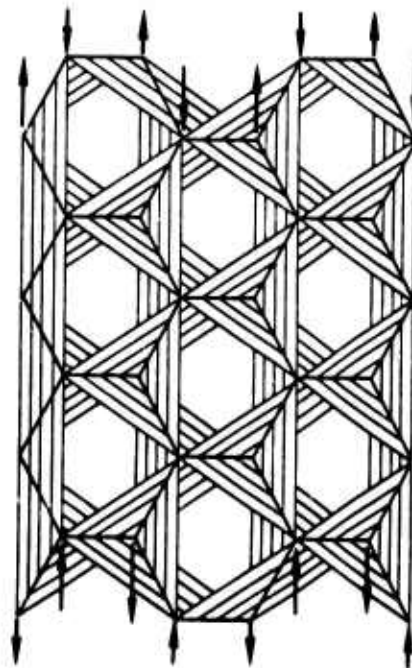
(a) X LOAD, P_X



(b) Y LOAD, P_Y

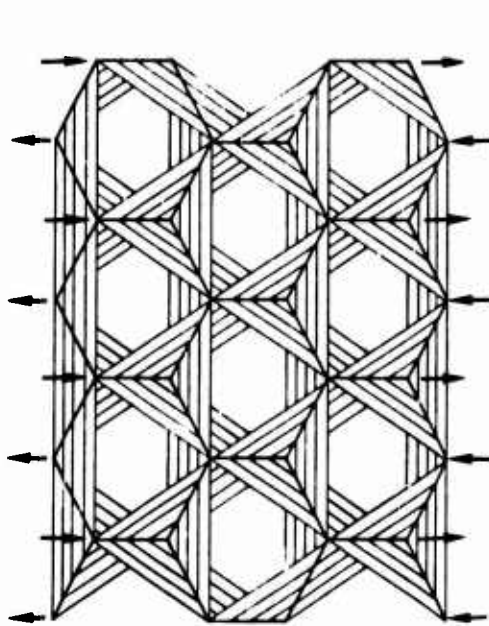


(c) SHEAR LOAD, P_{XY}

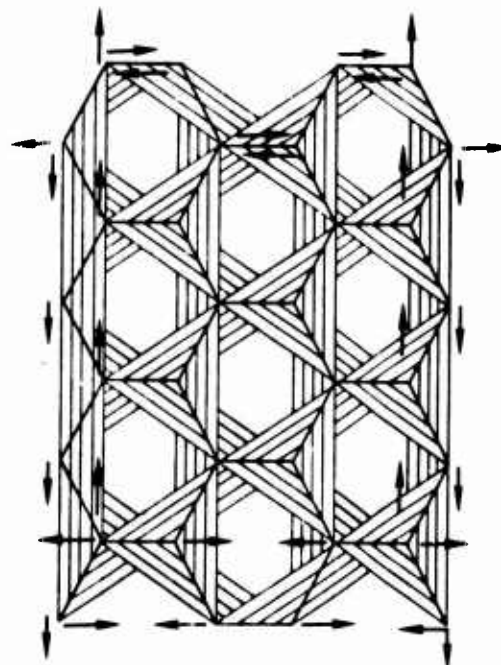


(d) X MOMENT, P_{MX}

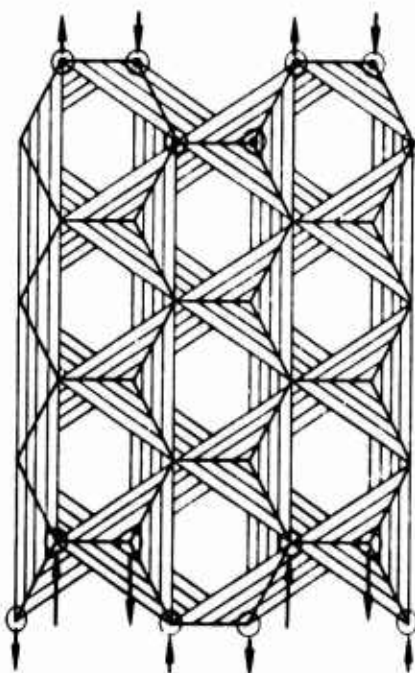
Figure 12. Flat Plate Applied Loads .



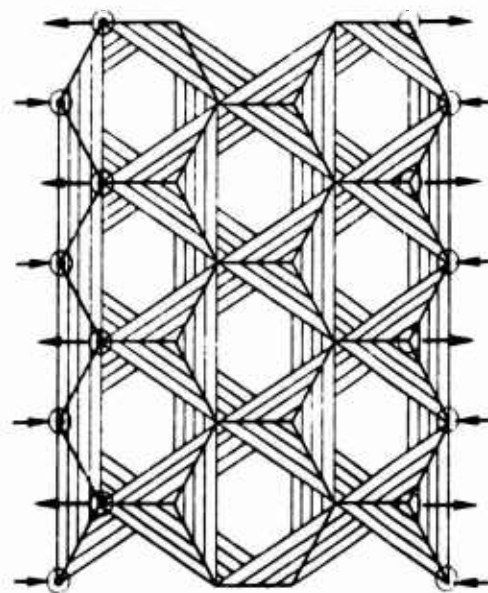
(e) Y MOMENT, PMY



(f) SHEAR MOMENT, PMXY



(g) X SHEAR, XQSHR



(h) Y SHEAR, YQSHR

Figure 12. Continued.

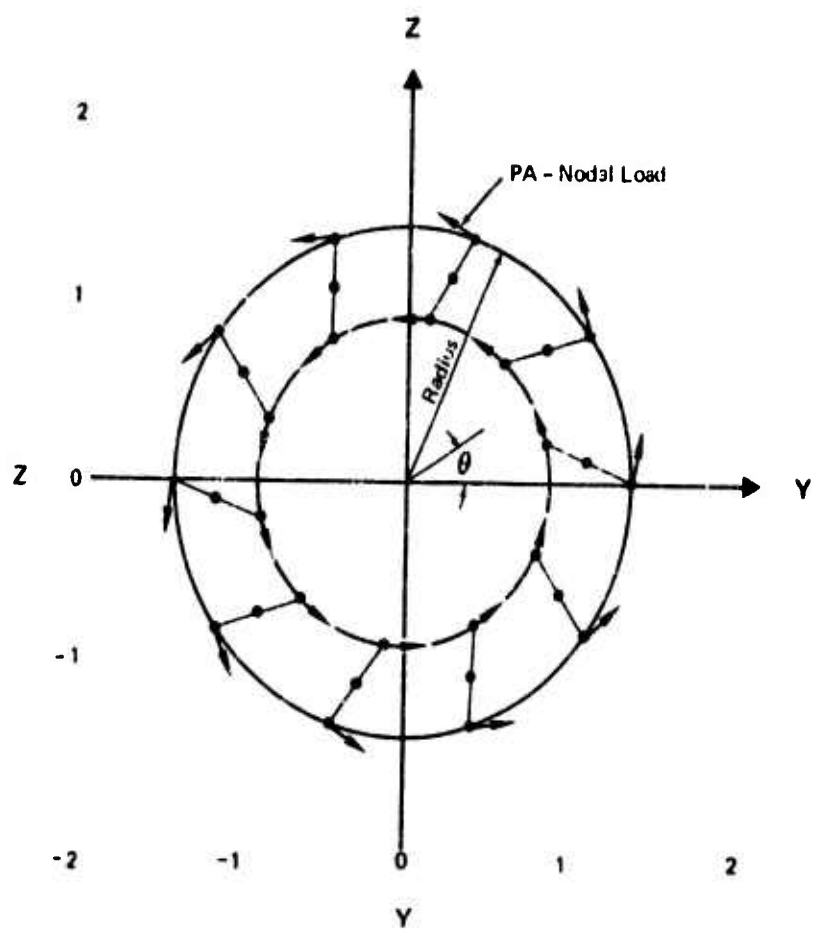


Figure 13. True Tetra-Core Cylinder With Torsion Load Applied, $L_X = 10$.

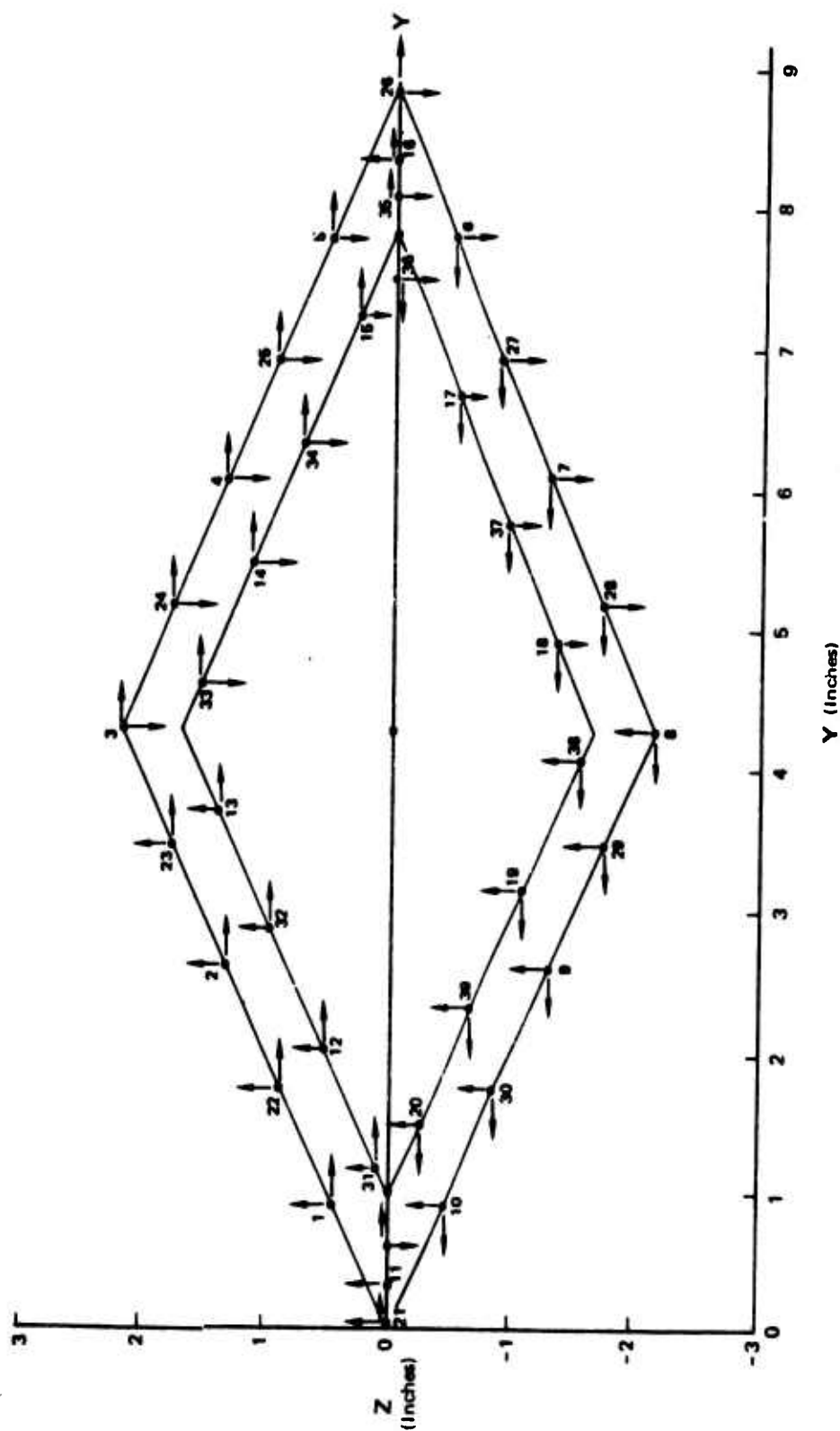


Figure 14. True Tetra-Core Airfoil With Torsion Load Applied, $L_X = 20$.

Loads can be applied to the true Tetra-core specimen being analyzed in any combination (end loads, moments, vertical shears, etc.). The truncated Tetra-core specimen can be loaded with end loads and moments in any combination, but if a vertical shear load is applied, then only certain combinations of loads may be applied at the same time, because of the fixities that are applied to the four sides of the panel to keep the model stable. If no vertical shear loads are applied, the Z-freedom is fixed for all nodes on the panel sides. If a vertical shear is applied on the X face, the side nodes are fixed in the Y direction, which permits movement in the Z direction. A vertical shear load on the Y face causes edge nodes to be fixed in the X direction. Thus, if a load in the Y direction is applied to a panel with a vertical shear load on the X face, the effect of the Y loads will be lost, since they are applied to nodes that cannot move in the direction of the load.

Moment, torsion, and end loads can be applied in any combination to both true and truncated Tetra-core cylinders and airfoils.

The effect of a hole in the Tetra-core can be simulated by giving the plates connected to a node in the center of the panel a zero stiffness. This unloads those plates and transfers the load to the remaining plates, providing the effect of a stress concentration around the hole.

AUTOMATIC MODEL GENERATOR

The Tetra-core analysis program has been written so that a finite-element model will be generated from the input data.

- o Type of Analysis--Card Type 1
- o Optimization Controls--Card Type 2
- o Geometry--Card Types 5, 6, and 7
- o Material Data--Card Types 4, 8, 9, and 10
- o Loading--Card Types 12 and 13

Flat Plate Nodal Generator

A different nodal generator subroutine has been written for each type of Tetra-core, true and truncated. Subroutines NODGEN and NODGAN are used to generate nodal coordinates for the true and truncated Tetra-core models, respectively. True Tetra-core nodes are generated starting at the Y-axis, with numbers increasing as the X distance increases, as shown in Figure 7. Nodal coordinates (X, Y, Z) on the bottom surface of the Tetra-core at a given X distance are generated first,

using the repeating geometry of Tetra-core, then those on the upper surface. Then the X distance is increased and another cycle of numbering is begun. For the truncated Tetra-core, a different system is used because of its more difficult geometry. Numbering is started at the Y-axis on the first vertical leg, as shown in Figure 8. Node numbering follows the first vertical leg to its end, then switches to the second vertical leg, etc.

Nodal coordinates are generated in order of their appearance on the leg, resulting in a mixing of nodes from the upper and lower surfaces. Nodes in the last vertical leg are generated in the same manner, but every fourth node is unconnected to any other node in the flat plate and must be given fixity in the X, Y, Z directions. If a cylinder or airfoil is generated, these extra nodes are used to connect the two sides of the plate after it has been folded so that the two edges almost meet.

Cylinder Nodal Generator

Cylinders are generated from the flat plate coordinates previously generated. The cylinder is assumed to have a circumference equal to the length of the X face of the flat plate plus the width of one tetrahedron. This gives the cylinder an outside radius that is the circumference divided by 2π . The new Y and Z coordinates of each node are calculated:

$$\theta_i = y_i \text{ flat plate} * 2\pi / \text{circumference}$$

$$y_{\text{cyl}} = \text{radius} * \cos(\theta_i)$$

$$z_{\text{cyl}} = \text{radius} * \sin(\theta_i)$$

The flat plate is essentially rolled into a cylinder so that the two edges of the flat plate are separated by the width of the one tetrahedron side in the Y direction. The same method is used for both true Tetra-core and truncated Tetra-core. An example of the end of a true Tetra-core cylinder is shown in Figure 13.

Airfoil Nodal Generator

Airfoil nodal coordinates are generated by transformation of flat plate coordinates using a curve-fitting subroutine. The same basic method is used for both true Tetra-core and truncated Tetra-core. The flat plate already generated is essentially folded over to make the upper and lower surfaces of the airfoil. The chord length of the airfoil is half the length of the X face of the original flat plate. The Z coordinate of each node is determined using the curve-fitting subroutine with the input t/C and X/C values. The distance

of a node from the X-axis and the airfoil chord length is used to calculate the X/chord ratio of that node. This X/chord ratio is used in the curve-fitting subroutine to determine the thickness/chord ratio of a point on the surface of the airfoil from the input t/C ratios of the airfoil. The Z coordinate of the node is then the thickness/chord ratio for that point times the chord length. See Figure 14 for an example of the end of a diamond-shaped airfoil generated by the program.

Plate Element Generator

Plate elements are generated automatically to connect the nodes from the nodal generator and form either a true Tetra-core or a truncated Tetra-core plate. True Tetra-core plate elements are generated in subroutine PLATGN, with truncated Tetra-core plate elements generated in subroutine PLATGA. The same method is used for both models, but a different subroutine is needed for each, since the nodal numbering sequence is different for each.

Flat Plate

Plate element generation starts with the upper plate in the left vertical leg. Linear strain triangular and quadrilateral plates are used for the Vertical and Skew A and B legs, since they can be expected to encounter compression strain on one side of the leg and tension strain on the other for some load cases. The mid-side node on the outer face of the element is reduced out, since it will not be connected to another element and would cause the merged stiffness matrix to be singular if left in. Two linear strain triangles are added to get the quadrilateral linear strain element used in the truncated Tetra-core model. See Figure 24 for the triangle and quadrilateral element node numbering sequence.

Plates are numbered so that the local X-axis of each plate element is in the fiber direction of that leg. Vertical plates are generated by following the vertical leg from top to bottom, then moving to the next vertical leg to the right, and starting at the top again. Each vertical leg has the same number of vertical elements.

Skew plate elements are then generated, starting with the plate in the top left of the flat plate segment, as shown in Figure 9. Skew A legs are generated from upper left to lower right. Generally, each Skew A leg has a different number of plate elements. Next, Skew B linear strain plate elements are generated, starting at the upper right corner of the segment, as shown in Figure 10. Leg generation proceeds from upper right to lower left, with a different number of plate elements in each leg.

If the face sheet option has been specified ("IFACE" = 1 on Card 1), an upper and lower face sheet will be generated for the segment.

Cylinder and Airfoil

A cylinder is generated by folding a flat plate model into a circular cylindrical segment. The two edges of the flat plate facing in the Y direction end up facing each other, separated by the width of a tetrahedron side. The result is a cylinder with a strip missing down the side. Plates are added to connect the two sides of the cylinder, resulting in a complete cylinder, shown in Figure 11. The plates making up a Tetra-core airfoil are generated in the same way, with the flat plate model being folded over into the desired shape and plate elements added to "zip up" the seam where the two ends meet.

Flat Plate Loads

The load generator takes input loads, which are uniform loads (given in either pounds/inch or inch-pounds/inch), and applies them as nodal loads to give an equivalent total load. The same method is used for both true Tetra-core panels and truncated Tetra-core panels. Since a different nodal numbering system was used for each type of Tetra-core, a different subroutine was used for each. Subroutine LODGEN generates loads for the true Tetra-core model, while loads for the truncated Tetra-core model are generated in subroutine LODGAN. When generating nodal loads for the linear strain triangle, the mid-side node must be loaded to give an equivalent uniform load across the element. Theoretically, the load should be divided so that one-sixth of the total load on that plate goes into each corner node and two-thirds goes into the mid-side node¹. This is done automatically in the program.

In the case of a uniform load in the X direction, the input load (pounds/inch) is multiplied by the panel width and divided by the number of nodes on that face, taking account of the fact that some nodes have more load than others.

$$PX = \frac{XLOAD \times XLGT}{NXND}$$

where XLOAD = Input load on the X face (lb/in.)

XLGT = Length of X face

NXND = Number of nodes on X face

NXND = 2 x LX + (2 x LX - 1) x 4 for true Tetra-core model

This gives the X direction load to be applied to each node (PX). Loads are applied to the apex nodes only in the LODGEN and LODGAN subroutines since mid-side nodes are generated in the stress analysis overlay. Nodal loads applied to give the effect of a uniform load in the X direction are shown in Figure 12a for a true Tetra-core flat plate.

Figure 12b shows a load in the Y direction applied to the true Tetra-core panel. The nodal load PY is calculated:

$$PY = \frac{YLOAD \times YLGT}{NYND}$$

where YLOAD = input load on the Y face (lb/in.)

YLGT = Length of Y face (in.)

NYND = $2 \times LY + (2 \times LY - 1) \times 4$ for a true Tetra-core model

LY = Number of legs on Y face

For a uniform in-plane shear load, the loads shown in Figure 12c are applied. The nodal loads are calculated in the same way as for a uniform load in the Y direction, except that a different nodal load (PXYA or PXYB) is used for the X and Y faces.

$$PXYA = \frac{XYLOD \times XLGT}{NXND}$$

where XYLOD = input in-plane shear load (lb/in.)

A moment load in the X direction is applied, as shown in Figure 12d, for a true Tetra-core flat plate. For a moment load, the mid-side nodes are not loaded. Nodal loads are calculated using the moment load per inch, panel width, panel depth, and number of nodes on the side being loaded.

$$PMX = \frac{XMOM \times XLGT}{HT \times LX}$$

where XMOM = Input moment on X face (in.-lb/in.)

A moment load is applied in the Y direction, as shown in Figure 12e. The nodal loads are calculated:

$$PMY = \frac{YMOM \times YLGT}{HT \times LY}$$

where YMOM = Input moment on Y face (in.-lb/in.)

A twisting shear moment is applied as shown in Figure 12f. Nodal loads PMXYA and PMXYB are calculated separately for the X and Y faces of the panel:

$$PMXYA = \frac{XYMOM \times YLGT}{HT \times LY}$$

$$PMXYB = \frac{XYMOM \times XLGT}{HT \times LX}$$

where XYMOM = Input twisting moment (in.-lb/in.)

Out-of-plane shear loads can be applied to flat panels using the input loads XQSHR and YQSHR. A combination of a vertical shear load and a moment load is applied to each loaded end of the panel, as shown in Figures 12g and 12h. The moment load PMX is applied to balance the moment caused by the two vertical shear loads PZ. The resulting set of loads is self equilibrating, so no reactions will result at the fixed nodes.

Calculations of nodal loads for a given X and Y vertical shear are:

$$PZ = \frac{XQSHR \times XLGT}{NXND}$$

$$PMY = \frac{XQSHR \times SIDE \times (LY - 1) \times XLGT}{2 \times HT \times LX}$$

where XQSHR = Input vertical shear load on X face (lb/in.)

$$PZ = \frac{YQSHR \times YLGT}{NYND}$$

$$PMX = \frac{YQSHR \times SIDE \times (0.86603 + Y2 \text{ OFF}) \times (LX-1) \times YLGT}{2 \times HT \times LY}$$

where YQSHR = Input vertical shear load on Y face (lb/in.)

For a true Tetra-core flat plate, these vertical shear loads can be applied with any other loads, such as moment loads and in-plane shear loads; etc. for the truncated Tetra-core flat panel, all loads cannot be applied simultaneously. A vertical shear load YQSHR can be applied with loads and moments on the X face, but cannot be used with in-plane shears or twisting moments or with a moment on the Y face. Similarly, the vertical shear load XQSHR can be applied with loads and moments on the Y face, but not with shear loads or moment loads on the X face. The reason is that the truncated Tetra-core model is

unstable on the edges and must be supported to prevent rigid body rotation in some of the plates. This is done by fixing nodes on the flat plate edges in the Z direction. This allows the flat plate to deflect in the X-Y plane but not in the Z direction. With this fixity, no vertical shear load can be applied to the model because nodes that would be loaded in the Z direction are fixed in that direction and cannot be loaded. When a vertical shear load on the X face is specified, the program will fix the edge nodes in the Y direction, instead of the Z direction, which allows Z nodal loads to be applied. However, loads can no longer be applied to these nodes in the Y direction. Therefore, shear loads and moments on the Y face cannot be applied at the same time that vertical shear loads are applied to the X face. Similarly, if a vertical shear load is applied to the Y face, all edge nodes are fixed in the X direction, and shears and moments on the X face cannot be applied. For the same reason, a twisting shear moment cannot be applied in conjunction with an out-of-plane shear load.

Cylinder Loads Generation

Total end loads, bending moments, and torsion loads are applied to cylinders as nodal loads. End loads are applied to the cylinder by dividing the input X load (in pounds) by the number of nodes on one end of the cylinder, taking into account the fact that some nodes have a larger load than others.

$$PX = \frac{XLOAD}{NXND}$$

where XLOAD = input X total end load on cylinder (lb)

A torsion load is applied to the cylinder as nodal loads tangent to the surface of the cylinder at the node being loaded, as shown in Figure 13. Loads on the inner surface of the cylinder have a smaller load than those on the outer surface by the ratio of their radii.

$$PA = \frac{TORQ}{LX \times (r_A^2 / r_A) \times 4}$$

$$PB = \frac{PA \ r_B}{r_A}$$

where TORQ = Input total torsion load on cylinder (in.-lb)

r_A = Radius of outer surface of cylinder

r_B = Radius of inner surface of cylinder

A bending moment is applied to the cylinder as a tension load on nodes below the Y-axis of the cylinder and a compression load on nodes above the Y-axis of the cylinder. The magnitude of the load on each node depends on the distance of that node from the Y-axis. That is, the neutral axis of the cylinder is assumed to be the Y-axis, and a straight-line variation of load with distance from the neutral axis is used.

$$PMX = \frac{XMOM}{SUM}$$

where XMOM = Input total moment on end of cylinder (in.-lb)

$$SUM = 4\pi \sum_{j=1}^{nodes} (Y_i)^2$$

Airfoil Loads Generation

End loads and bending moments are applied to the airfoil in the same way they were applied to the cylinder. A torsion load is applied to the airfoil by applying a load to each end node, in the direction of the airfoil surface at that node, as shown in Figure 14. Since no prediction could be made of the location of the shear center of the airfoil, it was necessary to apply the same total load to each node. This was done instead of ratioing the magnitude of each load by its distance from the shear center, as was done for the cylinder.

Nodal Fixity

In a finite-element analysis, it is necessary to restrain the model from translating or rotating as a rigid body, as this will result in a singular stiffness matrix. This is done by removing the ability of some nodes to move in the X, Y, or Z direction, thus "fixing" them in that direction.

For the Tetra-core flat plate analysis, nodes on the Y face next to the X-axis (nodes 3, 5, 11, 13, ..., 29 in Figure 7) are fixed in the Y direction. The first two nodes on the first vertical leg (nodes 3 and 5 in Figure 7) are fixed in the X direction. The lower of the two nodes is fixed in the Z direction. This keeps the model from translating or rotating as a rigid body.

The boundary conditions and system of applied loads used for the flat plate Tetra-core model result in a coupon type analysis. That is, the plate segment being analyzed can be thought of as a small coupon cut out of the larger structure, with loads corresponding to the loads in the larger structure at that point applied to the coupon sides. This allows the

use of a coarse-grid finite-element analysis, using a standard finite-element program, to determine the loads at any point in the structure. Tetra-core analyses are then run of several points in the structure, using loads found from the coarse-grid analysis.

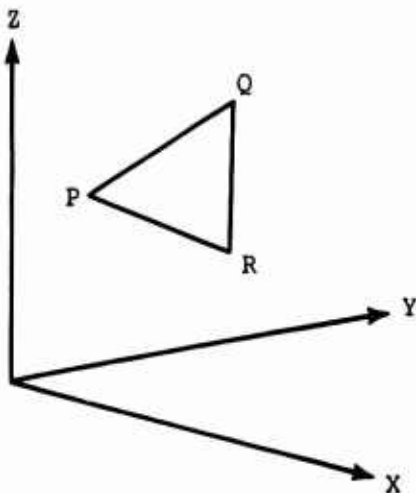
For the truncated Tetra-core model, additional fixities are applied to make the model stable, as discussed in the previous section. Nodes on the panel edge are given fixity in the Z direction to restrain rigid-body rotation of some edge plates. This can be illustrated by looking at the Skew B leg in the upper right corner of a panel (Figure 8). It can be seen that this plate element can pivot about its lower surface node (No. 46) without inducing any in-plane strains into the Skew A or vertical plates supporting it. This results in a singular stiffness matrix, but can be prevented by restraining movement of the edge nodes in the X, Y, or Z direction. If a vertical shear load is applied on the X faces of the panel, then edge nodes are given a fixity in the Y direction instead of the Z direction. If a shear load is applied on the Y faces, then edge nodes are given a fixity in the X direction, as discussed in the Load section.

For Tetra-core cylinders and airfoils, nodes on the end next to the X-axis are given fixity in the X, Y, and Z directions, providing the effect of a cylinder with one end fixed into a stiff foundation.

FINITE-ELEMENT STRESS ANALYSIS

Linear Strain Element

To best derive the stiffness matrix for a linear strain, six-node triangular element, we first require the geometry expressed in terms of some local in-plane coordinate system.



If $X_{QP} = X(Q) - X(P)$, etc.,

$$Y_{QP} = Y(Q) - Y(P)$$

$$Z_{QP} = Z(Q) - Z(P)$$

$$\text{and } l_1 = \sqrt{X_{QP}^2 + Y_{QP}^2 + Z_{QP}^2}$$

where l_1 is the length of the line PQ.

Then the direction cosines of PQ are

$$\lambda_1 = X_{QP}/l_1 \text{ for } l_1 > 0$$

$$\lambda_2 = Y_{QP}/l_1$$

$$\lambda_3 = Z_{QP}/l_1$$

$$\text{or } \lambda_1 = \lambda_2 = \lambda_3 \text{ for } l_1 \leq 0$$

The direction cosines of the normal to PQ, through R are obtained as follows:

with $RR = \lambda_1 X_{RP} + \lambda_2 Y_{RP} + \lambda_3 Z_{RP}$

$$X_1 = X_{RP} - \lambda_1 RR$$

$$Y_1 = Y_{RP} - \lambda_2 RR$$

$$Z_1 = Z_{RP} - \lambda_3 RR$$

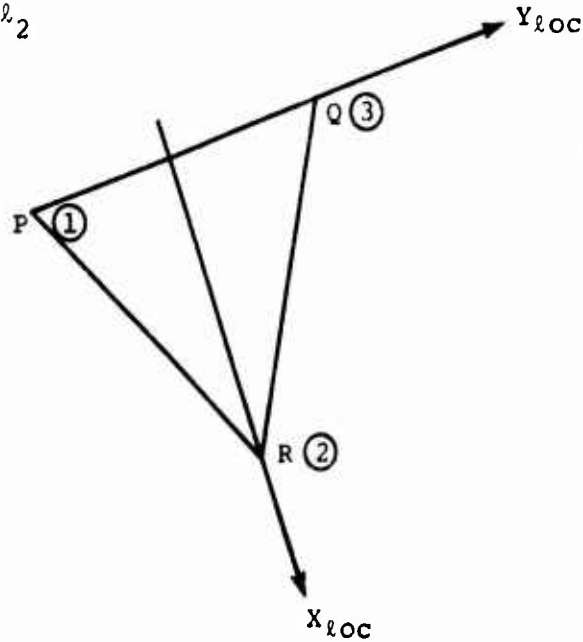
and $\ell_2 = \sqrt{X_1^2 + Y_1^2 + Z_1^2}$

and the direction cosines of the normal are

$$\mu_1 = X_1/\ell_2$$

$$\mu_2 = Y_1/\ell_2$$

$$\mu_3 = Z_1/\ell_2$$



Referring the triangle to local coordinates shown previously and denoting $X_{loc_i} - X_{loc_j}$ by X_{ij} , we have

$$a_3 = x_{21} = x_{QP}^{\lambda_1} + y_{QP}^{\lambda_2} + z_{QP}^{\lambda_3}$$

$$-b_3 = y_{21} = x_{QP}^{\mu_1} + y_{QP}^{\mu_2} + z_{QP}^{\mu_3}$$

$$a_1 = x_{32} = x_{RQ}^{\lambda_1} + y_{RQ}^{\lambda_2} + z_{RQ}^{\lambda_3}$$

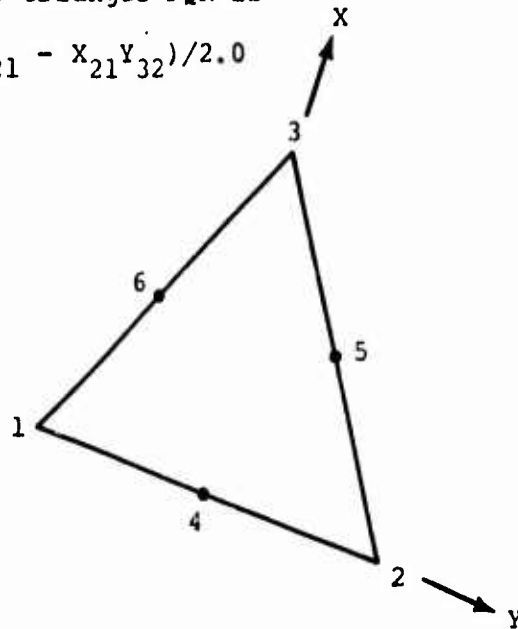
$$-b_1 = y_{32} = x_{RQ}^{\mu_1} + y_{RQ}^{\mu_2} + z_{RQ}^{\mu_3}$$

$$-a_2 = x_{31} = x_{RP}^{\lambda_1} + y_{RP}^{\lambda_2} + z_{RP}^{\lambda_3}$$

$$b_2 = y_{31} = x_{RP}^{\mu_1} + y_{RP}^{\mu_2} + z_{RP}^{\mu_3}$$

and the area of triangle PQR is

$$\Delta = (x_{32}y_{21} - x_{21}y_{32})/2.0$$



Following the method of Felippa,² the stiffness matrix of the six-node linear strain triangle may be expressed as

$$[B^T] \begin{bmatrix} c_{11Q} & c_{12Q} & c_{13Q} \\ c_{21Q} & c_{22Q} & c_{23Q} \\ c_{31Q} & c_{32Q} & c_{33Q} \end{bmatrix} [B]$$

where B relates the strain at any point within the triangle to the nodal deflections,

$$\text{and } Q = \frac{At}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (3 \times 3)$$

$$\text{and } \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{Bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \alpha_{12} \end{Bmatrix}$$

Using Hooke's Law to relate stress and strain, $C_{23} = C_{32} = C_{31} = C_{13} = 0$, and further, C_{ij} is symmetric.

Matrix B may be expressed as follows:

$$\text{with } \phi_X = \frac{1}{2\Delta} \begin{bmatrix} 3b_1 & -b_2 & -b_3 & 4b_2 & 0 & 4b_3 \\ -b_1 & 3b_2 & -b_3 & 4b_1 & 4b_3 & 0 \\ -b_1 & -b_2 & 3b_3 & 0 & 4b_2 & 4b_1 \end{bmatrix} \quad (3 \times 6)$$

$$\text{and } \phi_Y = -\frac{1}{2\Delta} \begin{bmatrix} 3a_1 & -a_2 & -a_3 & 4a_2 & 0 & 4a_3 \\ -a_1 & 3a_2 & -a_3 & 4a_1 & 4a_3 & 0 \\ -a_1 & -a_2 & 3a_3 & 0 & 4a_2 & 4a_1 \end{bmatrix} \quad (3 \times 6)$$

$$\text{then } B = \begin{bmatrix} \phi_X & 0 \\ 0 & \phi_Y \\ \phi_Y & \phi_X \\ (9 \times 12) \end{bmatrix}$$

So

$$k = \begin{bmatrix} \phi_x^T & 0 & \phi_y^T \\ 0 & \phi_y^T & \phi_x^T \end{bmatrix} \begin{bmatrix} C_{11}Q & C_{12}Q & 0 \\ C_{12}Q & C_{22}Q & 0 \\ 0 & 0 & C_{33}Q \end{bmatrix} \begin{bmatrix} \phi_x & 0 \\ 0 & \phi_y \\ \phi_y & \phi_x \end{bmatrix}$$

which may be expanded to

$$k = \left[\begin{array}{c|c} \phi_x^T C_{11}Q \phi_x + \phi_y^T C_{33}Q \phi_y & \phi_x^T C_{12}Q \phi_y + \phi_y^T Q C_{33} \phi_x \\ \hline \phi_y^T C_{22}Q \phi_x + \phi_x^T C_{33}Q \phi_y & \phi_y^T C_{22}Q \phi_y + \phi_x^T C_{33}Q \phi_x \end{array} \right]$$

(6 X 6) (6 X 6)

(6 X 6) (6 X 6)

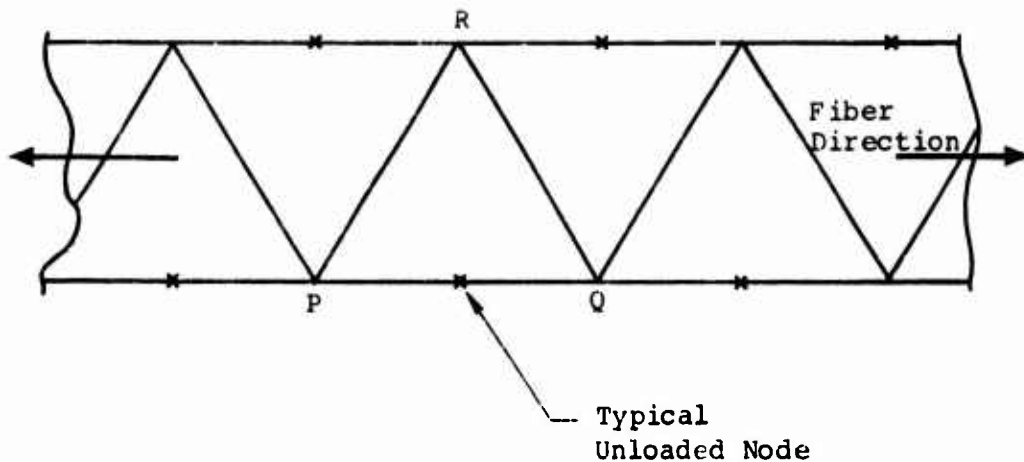
When k is partitioned as above, a small increase in efficiency of computing results as

- 1/ Multiplication by null matrices is avoided
- 2/ Symmetry of the stiffness matrix can be used
- 3/ Storage for a full B matrix is not required.

With

$$\begin{Bmatrix} P_{X_1} \\ P_{X_2} \\ P_{X_3} \\ \vdots \\ P_{Y_5} \\ P_{Y_6} \end{Bmatrix} = [k] \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ V_5 \\ V_6 \end{Bmatrix}$$

In the Tetra-core model, if P-Q is the fiber direction, then the node at the mid-side of P-Q is never loaded or supported out-of-plane.



Upon generating the structural stiffness matrix, this node would, in general, create linear dependence in the stiffness matrix. The linear dependence could be removed by giving the triangle bending stiffness, but this would greatly reduce program efficiency. As the node is never loaded, the simplest and most direct solution is to remove the node at the elemental stage.

This is done as follows: The node to be removed corresponds to node 6. So, when k is rearranged so that it corresponds to the usual order, the rows and columns are rearranged so that those corresponding to U_6 , V_6 occur at the bottom and right-hand side of the matrix.

$$\begin{Bmatrix} P_{X_1} \\ P_{Y_2} \\ P_{X_2} \\ P_{Y_2} \\ P_{X_3} \\ P_{Y_3} \\ P_{X_4} \\ P_{Y_4} \\ P_{X_5} \\ P_{Y_5} \\ \hline P_{X_6} \\ P_{Y_6} \end{Bmatrix} = \begin{bmatrix} & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ \hline & & & & & & & & & & \\ & & & & & & & & & & \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ U_2 \\ V_2 \\ U_3 \\ V_3 \\ U_4 \\ V_4 \\ U_5 \\ V_5 \\ \hline U_6 \\ V_6 \end{Bmatrix}$$

and the matrix is partitioned so that

$$\begin{Bmatrix} P_{1-5} \\ P_6 \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} U_{1-5} \\ U_6 \end{Bmatrix}$$

P_{X_6}, P_{Y_6} are each zero, so

$$\{P_{1-5}\} = [K_{11}] [U_{1-5}] + [K_{12}] [U_6]$$

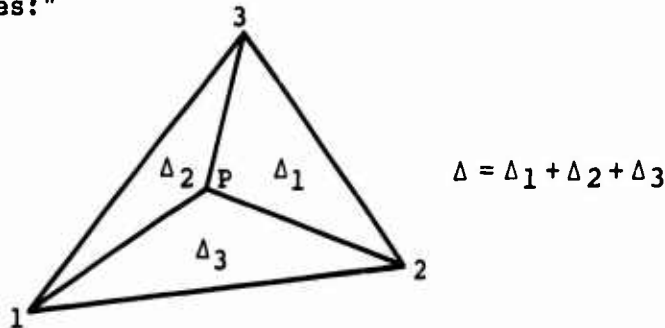
$$\{0\} = [K_{21}] [U_{1-5}] + [K_{22}] [U_6]$$

$$\text{i.e., } \{P_{15}\} = [K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}] \quad U_{1-5}$$

and we have the stiffness of the element expressed in terms of the five remaining nodes as

$$[K_{11}] - [K_{12}]^T [K_{22}] [K_{21}] = K_{\text{RED}}$$

In calculating the stress matrix, as in Reference 1, we define "area coordinates:"



$$\xi_1 = \frac{\Delta_1}{\Delta}; \quad \xi_2 = \frac{\Delta_2}{\Delta}; \quad \xi_3 = \frac{\Delta_3}{\Delta}$$

Coordinates ξ_i uniquely determine location P, and the stress matrix is

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} \xi_i & 0 & 0 \\ 0 & \xi_i & 0 \\ 0 & 0 & \xi_i \end{bmatrix} [B] \begin{Bmatrix} U_1 \\ V_1 \\ U_2 \\ V_2 \\ \vdots \\ U_6 \\ V_6 \end{Bmatrix} \quad i = 1 \rightarrow 3$$

In the program this has been broken down into a more efficient form as

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} \langle \psi_X \rangle & 0 \\ 1 \times 6 & \\ 0 & \langle \psi_Y \rangle \\ & 1 \times 6 \\ \langle \psi_X \rangle & \langle \psi_X \rangle \\ 1 \times 6 & \\ 3 \times 12 & \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ \vdots \\ U_6 \\ V_6 \end{Bmatrix}$$

where

$$\begin{aligned} \langle \psi_X \rangle = \frac{1}{2A} & \langle (4\xi_1-1)b_1, (4\xi_2-1)b_2, (4\xi_3-1)b_3, 4(\xi_2b_1+\xi_1b_2), \\ 1 \times 6 & 4(\xi_3b_2+\xi_2b_3), 4(\xi_1b_3+\xi_3b_1) \rangle \end{aligned}$$

$$\begin{aligned} \langle \psi_Y \rangle = \frac{1}{2A} & \langle (4\xi_1-1)a_1, (4\xi_2-1)a_2, (4\xi_3-1)a_3, 4(\xi_2a_1+\xi_1a_2), \\ 1 \times 6 & 4(\xi_3a_2+\xi_2a_3), 4(\xi_1a_3+\xi_3a_1) \rangle \end{aligned}$$

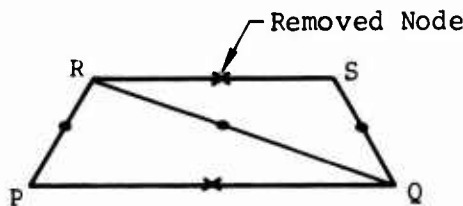
Multiplication by the C matrix is carried out after element strains have been determined in subroutine DEFL. The sixth node is reduced out of the stress matrix in the same manner as for the stiffness matrix.

$$[S T_{RED}] = [S T] - [S T_{12}]^T [K_{22}]^{-1} [K_{21}]$$

The stress matrix is determined for four locations in the linear strain triangle: Nodes P, Q, and R and the triangle centroid.

Quadrilateral Element

The linear strain quadrilateral element used in the truncated Tetra-core model is formed by adding two linear strain triangles from which the sixth nodes have been removed.



The mid-side node between R and Q is then reduced out as the top and bottom mid-side nodes had been, since it is never loaded or attached to another element. This results in a six-node linear strain quadrilateral. The stress matrices are similarly formed, with stress now being computed at the four corner nodes.

Constant Strain Triangle

The constant strain triangle used for face sheets is generated using the method of Przemieniecki³. The local in-plane coordinate system set up for the linear strain triangle

with $a_1 = x_{32}$, etc., is used. A [B] matrix relating strains within the element to deflections of the node points is defined.

$$[B] = \frac{1}{2A} \begin{bmatrix} b_1 & 0 & b_3 & 0 & b_2 & 0 \\ 0 & a_1 & 0 & a_3 & 0 & a_2 \\ a_1 & b_1 & a_3 & b_3 & a_2 & b_2 \end{bmatrix}$$

The stiffness matrix [K] is then

$$[K] = \int_V [B]^T [C] [B] dV$$

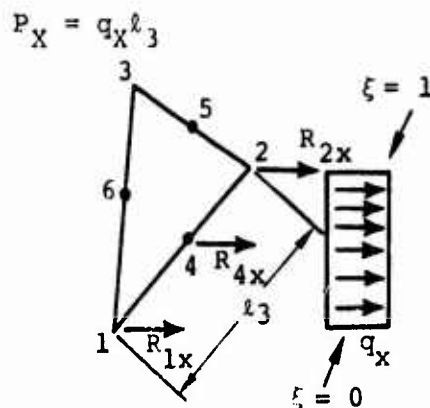
$$[K] = \Delta t [B]^T [C] [B]$$

and

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_3 \end{bmatrix} [B] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}$$

Consistent Surface Loads Vector

The uniform in-plane loading acting on an edge of the linear strain triangle must be resolved into stress resultants acting at the two corner nodes and the midpoint. The derivation shown here is from a document by Tocher.¹ The total force on the plate is



The equilibrium equations can be written using virtual work as

$$\langle \delta U_1 \quad \delta U_4 \quad \delta U_2 \rangle \begin{Bmatrix} R_{1X} \\ R_{4X} \\ R_{2X} \end{Bmatrix} = \int_0^1 \delta U(\xi) P_X \ell_3 d\xi$$

The virtual deflection $\delta U(\xi)$ can be obtained as (Ref. 1)

$$\delta U = \langle \delta U_1 \quad \delta U_4 \quad \delta U_2 \rangle \begin{Bmatrix} 1-3\xi + 2\xi^2 \\ 4\xi - 4\xi^2 \\ 2\xi^2 - \xi \end{Bmatrix}$$

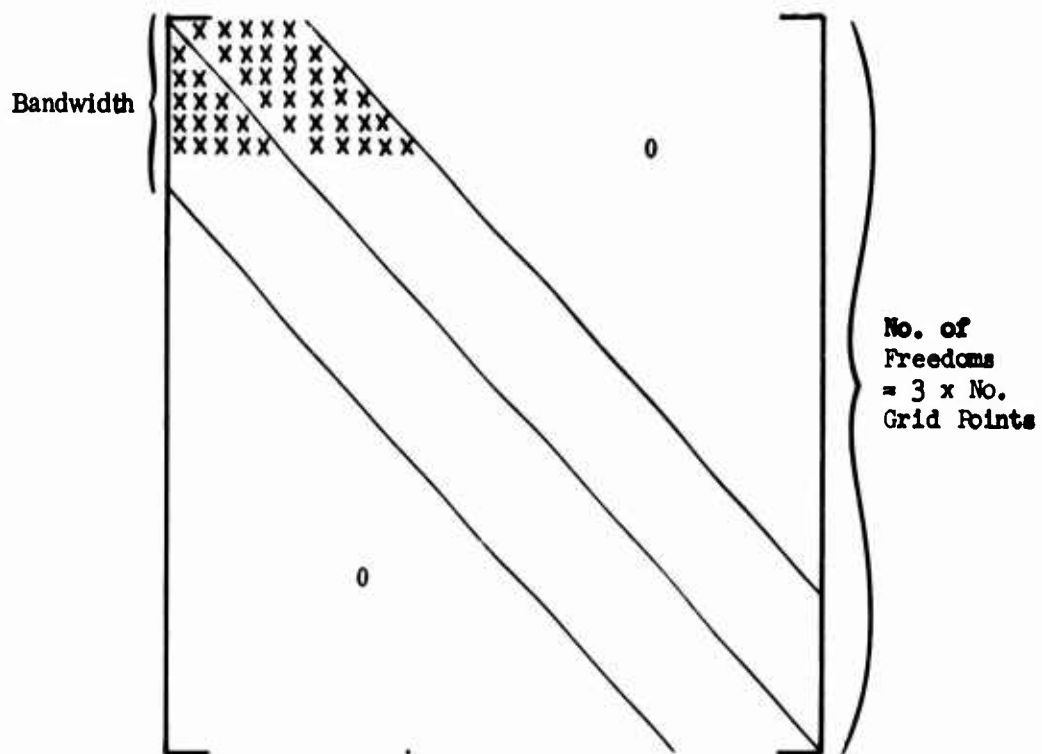
Thus

$$\begin{Bmatrix} R_{X1} \\ R_{X4} \\ R_{X2} \end{Bmatrix} = q_X \ell_3 \int_0^1 \begin{Bmatrix} 1-3\xi + 2\xi^2 \\ 4\xi - 4\xi^2 \\ 2\xi^2 - \xi \end{Bmatrix} d\xi = \frac{q_X \ell_3}{6} \begin{Bmatrix} 1 \\ 4 \\ 1 \end{Bmatrix}$$

and the load distribution is simply two-thirds of the load through the middle node and one-sixth through each end node.

Calculation of Deflections

A solution of the equation $k\Delta = P$ is used to calculate deflections (Δ) from the stiffness matrix (k) and the loads vector (P). A banded form of the stiffness matrix is assumed, in which only the nonzero portion of the stiffness matrix, which is inside the bandwidth, is stored (Figure 15). The stiffness matrix is stored in blocks on the disc, with each block being three rows. Thus, only one block is in core at any one time during generation of the stiffness matrix. Each elemental stiffness matrix is generated and added into the overall stiffness matrix using the direct stiffness method. The location of the block which each member of the elemental stiffness matrix is added into is determined from the nodal connectivity of that element. Each element will add its stiffness to several blocks, thus requiring the ability to turn the needed block by means of a direct-access disc.



Core Storage Requirement = Bandwidth x No. of Freedoms

Figure 15. Structural Stiffness Matrix.

Once the stiffness matrix has been formed, it is reduced to a triangular form using the Choleski methods³. This results in all members of the matrix below the diagonal being zero.

$$\bar{k}_{ii} = k_{ii} - \sum_{r=1}^{i-1} \bar{k}_{ir}^2 \quad 1/2 \quad i = 1, n$$

$$\bar{k}_{ij} = k_{ij} - \sum_{r=1}^{j-1} \bar{k}_{ir} \bar{k}_{rj} / \bar{k}_{jj} \quad i < j$$

$$\bar{k}_{ij} = 0 \quad i > j$$

where n is the order of the stiffness matrix. The loads vector is modified by the same factors that were used to triangularize the stiffness matrix. With the triangularized form of the stiffness matrix, the deflections of the last node can be calculated

$$\bar{k}_{1n} \Delta_1 + \bar{k}_{2n} \Delta_2 + \dots + \bar{k}_{nn} \Delta_n = P_n$$

where $\bar{k}$ is the triangularized stiffness matrix.

Since $\bar{k}_{1n}, \bar{k}_{2n} \dots \bar{k}_{n-1,n} = 0$

$$\Delta_n = P_n / \bar{k}_{nn}$$

Using this method, $\Delta_{n-1}, \Delta_{n-2}, \dots, \Delta_1$ can be solved, in turn, by back substituting the previously found deflections into the calculations for the next higher row. For example, Δ_{n-1} can be calculated

$$\bar{k}_{n-1,n-1} \Delta_{n-1} + \bar{k}_{n,n-1} \Delta_n = P_{n-1}$$

where k_n had been calculated in the step before and

$$\bar{k}_{n-1,n-1}, \bar{k}_{n,n-1} \text{ and } P_{n-1} \text{ are known}$$

This method of calculating deflections has proved to be much faster than the traditional method of inverting the stiffness matrix and multiplying $k^{-1}p$ to get Δ . The method allows a large problem to be solved on a computer with small core space,

since only three rows and three columns of the stiffness matrix are in core at one time during solution. However, it requires many accesses to the disc during a solution, since rows and columns must be shuffled in and out of core many times. On the 360/44 at Fort Eustis, the disc space available is the limiting factor on the size of problem that can be run.

Fixities are applied to the stiffness matrix by multiplying the diagonal term of the row to be fixed by a factor of 10^{12} . This has the effect of giving that freedom a much greater stiffness than other freedoms, effectively fixing it. This is equivalent to putting a one on the diagonal and setting elements of that row and column to zero, but it is more efficient.

OPTIMIZATION

A modified form of the steepest-descent optimization method is used to find a minimum weight Tetra-core design for a given set of loads⁴. Variables that can be optimized are tetrahedron geometry (side length, height, theoretical height, Y1 offset, Y2 offset, THETA) and the thicknesses of each leg. See Figures 4 and 5 for a description of the tetrahedron geometry. Optimization controls have been set up so that any combination of variables can be held constant during optimization. Also, any three variables can be forced to be the same during optimization; i.e., Vertical and Skew A and B legs can be kept to the same thickness during optimization. Input Card Type 2 is used to specify which variables will be optimized.

The optimization method will be illustrated using a two-dimensional example so that the variation of design variables can be plotted graphically. For the example, assume that a Tetra-core run is being made in which all variables are held constant except for the vertical leg thickness and the skew leg thickness, and that the Skew A and B legs will have the same thicknesses. Using this model, a design space can be plotted for a given set of loads in which the X and Y axes correspond to the variation in vertical and skew leg thicknesses as shown in Figure 16. For any point in the design space, the Tetra-core element will have a given weight and margin of safety. A series of curves of constant weight and another series of curves of constant margin of safety can be drawn as shown in the figure. The margin-of-safety curve for a margin of safety of zero is used to divide the design space into a feasible region, in which the margin of safety is greater than zero, and an unfeasible region, with a margin of safety less than zero. The optimization program does not know the location or shape of these curves, except for the location of the present design, but it can calculate the gradient of these curves; that is, the direction of travel in design space which will result in the maximum change in either the weight or the margin of safety. The gradient can be calculated by changing each variable, by a small amount, while holding all other variables constant and calculating the weight and margin of safety for the new design point. This is done separately for all variables, resulting in a set of partial derivatives:

$$\frac{\partial MS}{\partial t_i} = \frac{MS_i - MS_{BASE}}{t_i - t_{iBASE}} \quad i = 1, N, N = \text{No. of variables}$$

$$\frac{\partial WT}{\partial t_i} = \frac{WT_i - WT_{BASE}}{t_i - t_{iBASE}} \quad i = 1, N$$

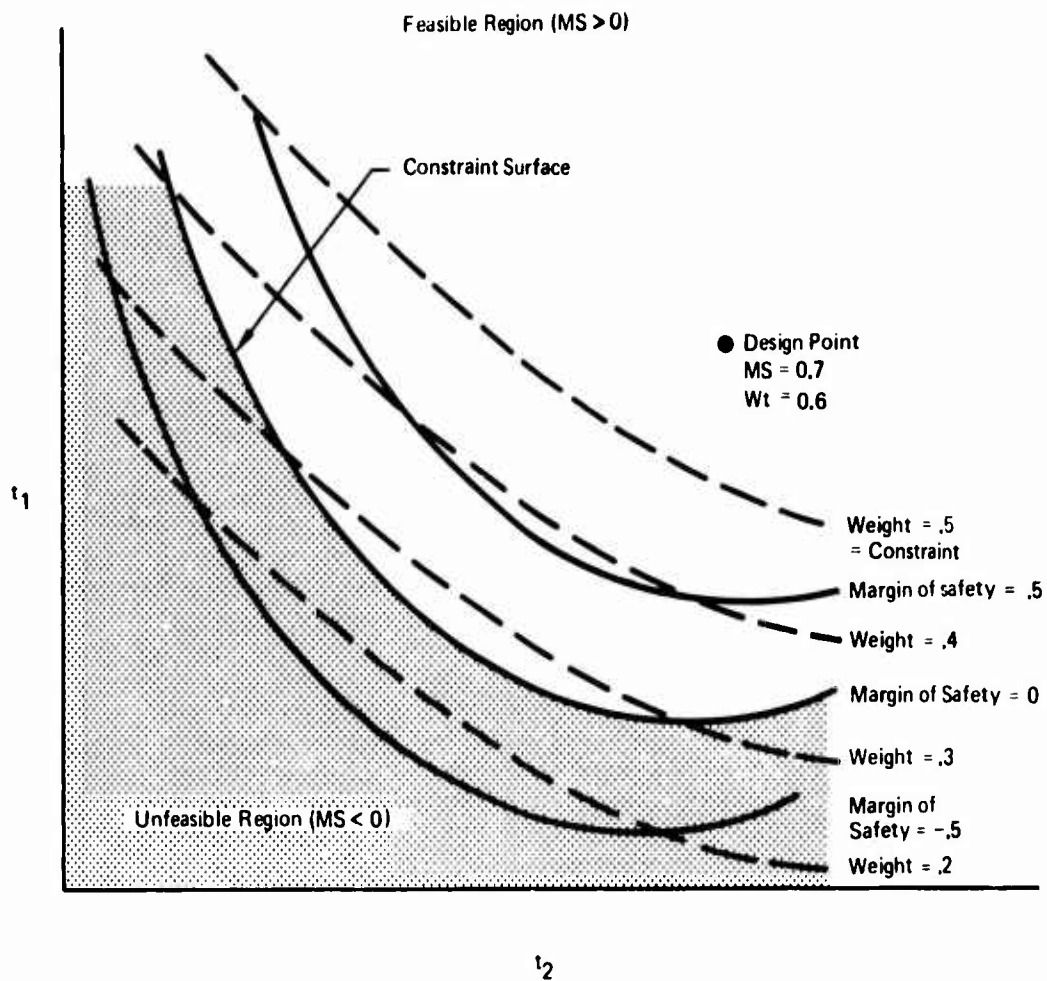


Figure 16. Design Space.

where MS_{BASE} , WT_{BASE} , and t_{iBASE} refer to the design point at the start of calculation of derivatives, before changing any thickness. Then these sets of partial derivatives can be used to calculate the required gradients.

$$\beta_{i_k} = \frac{\partial MS_k}{\partial t_i} / \sum_{j=1}^n \left(\frac{\partial MS_k}{\partial t_j} \right)^2 \quad \begin{array}{l} i = 1, N \\ k = 1, M \\ M = \text{number of constraints} \end{array}$$

and

$$\alpha_i = \frac{\partial WT}{\partial t_i} / \sum_{j=1}^n \left(\frac{\partial WT}{\partial t_j} \right) \quad i = 1, N$$

These gradients are actually the direction cosines of the normal to each curve at that design point.

For example, assume that the starting design point is chosen at Point A in Figure 17 and that the weight and margin of safety have been calculated for this point. As yet, no partial derivatives have been calculated, so the thickness will be decreased in steps in the same ratio to get to the constraint curve ($MS=0$). In the general case, the program changes only the thicknesses in first reaching the constraint curve, leaving the side, height, etc., dimensions constant. Then the program calculates the gradients of the weight and the margin-of-safety curves as previously described. There is a different margin-of-safety curve for each leg, since changing the thickness of one leg will change the margin of safety in that leg more than in another leg. This requires the calculation of a gradient of the margin of safety in each leg, as shown in Figure 18. A composite constraint gradient is computed using the gradient to each constraint curve and the margin of safety in each leg.

$$\{\gamma\} = \{\beta_1\} (1-MS_1)^{15} + \{\beta_2\} (1-MS_2)^{15} + \dots + \{\beta_M\} (1-MS_M)^{15}$$

The proportion in which each constraint gradient is added into the composite constraint gradient is determined by the distance of the design point from that constraint.

In Figure 18 for case A, in which the design point is at the intersection of two constraints, each gradient is added in equally, resulting in a new direction. In case B, in which the design point is on one constraint and far from the other, the gradient of the constraint which the design is on will have the major influence on the new direction. This composite

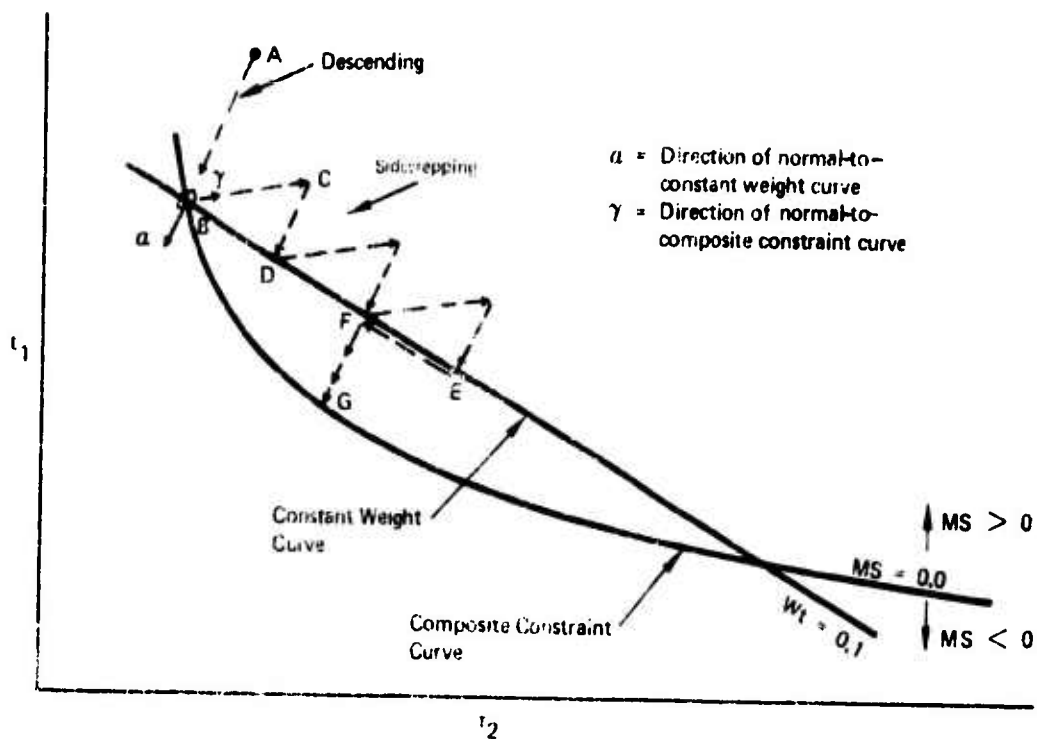


Figure 17. Two-Dimensional Optimization Example.

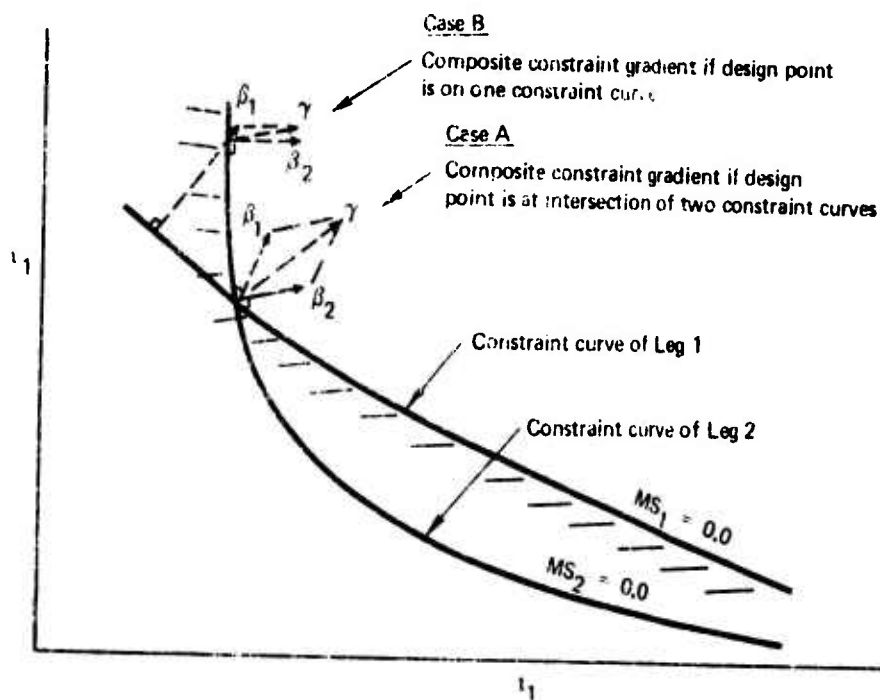


Figure 18. Composite Constraint Gradient.

constraint gradient is used by the program to make a step in the direction of maximum change in margin of safety, to Point C in Figure 17. The step is calculated:

$$t_{i+1} = t_i + \gamma_i X$$

where X is the step length. Next, the design variables are decreased along the gradient to the constant weight curve to reach the starting weight (Point D). This cycle B-C-D results in a sidestep away from the constraint curve in the direction of maximum increase in margin of safety, while maintaining a constant weight. The margin of safety is calculated at Point D. If the margin of safety is now greater than it was at Point B, then the bucket in the curve has not been reached and another sidestep is taken (D-F). Margin of safety is recalculated and another sidestep is taken (F-E). When the margin of safety is calculated at Point E, it is found to be less than at Point F. Therefore, the bucket in the constraint curve has been passed and a backward step is taken (E-F). Now the variables are decreased along the gradient to the constant weight curve (this is known as the direction of steepest descent) in steps, to reach the constraint curve (Point G). This series of operations (B-G) is called one cycle in the program. At the completion of one cycle (Point G), gradients are recalculated, the step length is reduced, and another series of sidesteps is begun. The number of optimization cycles to be run is input as "NSTP" on card type one. If a zero or blank is input for "NSTP," the program will set "NSTP" = 3. If the program makes ten sidesteps (B-D, D-F, etc.) without reaching the bucket, the program will descend to the constraint curve, recalculate gradients, and begin sidestepping again with a larger step length to reach the optimum in a fewer number of steps. At the end of each cycle (B-G), the weight of the design is checked. If the new weight is less than the weight at the end of the previous cycle, the variables are stored as a new optimum design. Generally, the program will find a series of designs, each lighter than the previous one. However, it is possible that the optimum will be found on the first or second cycle, and the rest of the cycles will result in heavier designs. This makes it necessary to store the lightest design found, since the program may not return to it.

In general, the constraint and constant weight curves are not two dimensional but are N -dimensional hypersurfaces, N being the number of variables. Figure 19 illustrates the set of constraint surfaces for a three-dimensional problem. Generally, several optimum designs will be possible for a given set of variables, and the program will only find the one that is

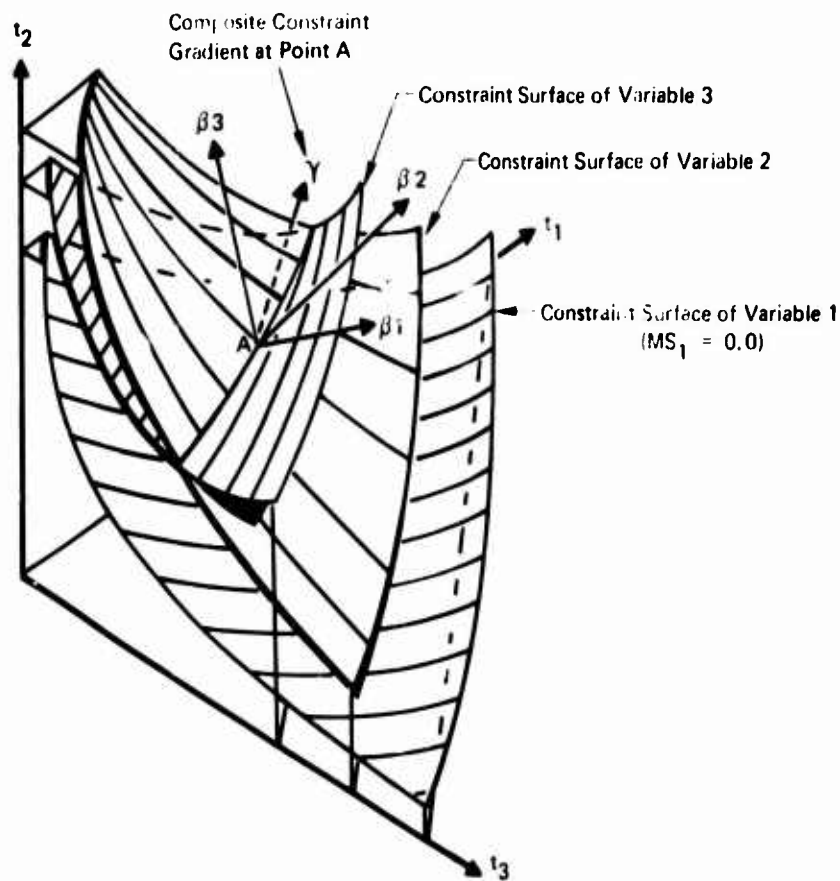


Figure 19. Constraint Surfaces of Three-Dimensional Problem.

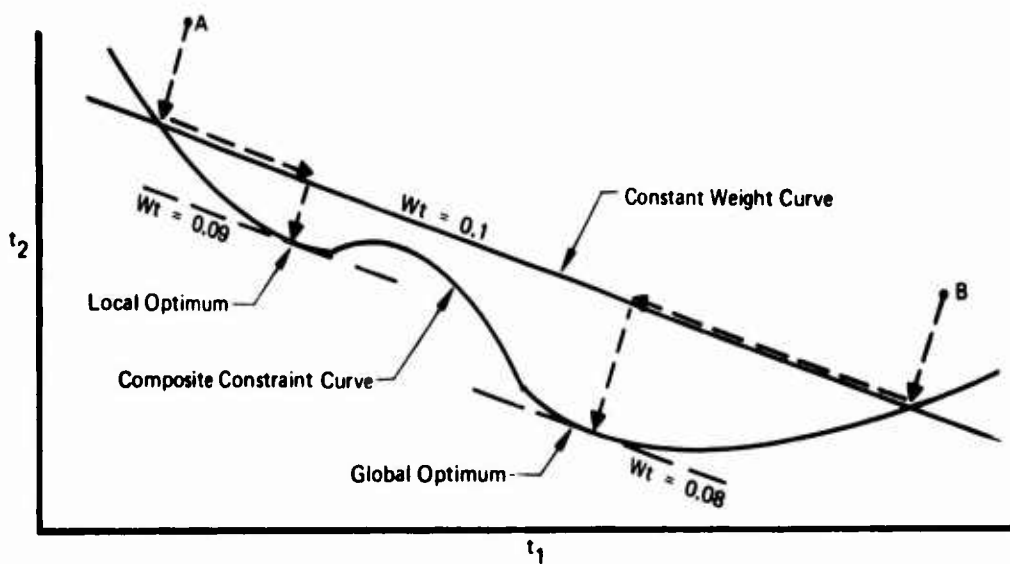


Figure 20. Global Optimum.

nearest to the input starting design. In Figure 20, if optimization is begun from Point A, an optimum design with a weight of 0.9 pound/square inch will be found, while starting at Point B will result in a weight of 0.08 pound/square inch. Thus it is necessary to make several optimization runs to determine if the "global" or most optimum design has been found.

The steepest descent/sidestep optimization method can be thought of as a way of getting down a mountain pass to the lowest altitude possible. The sides of the mountain represent constraints, and lines of constant altitude represent lines of constant weight. Once a given altitude has been reached by the design point, the altitude cannot be increased--only decreased.

A plot of the weight versus number of cycles for a typical case is shown in Figure 21. For this run a true Tetra-core flat plate was optimized with the side length held constant, the height and vertical leg thickness allowed to vary, and the Skew A and B leg thicknesses held the same. The largest change in weight was in the first cycle, indicating that the optimum region was found quickly, with the rest of the cycles being taken up with searching the immediate region to find the exact location of the optimum.

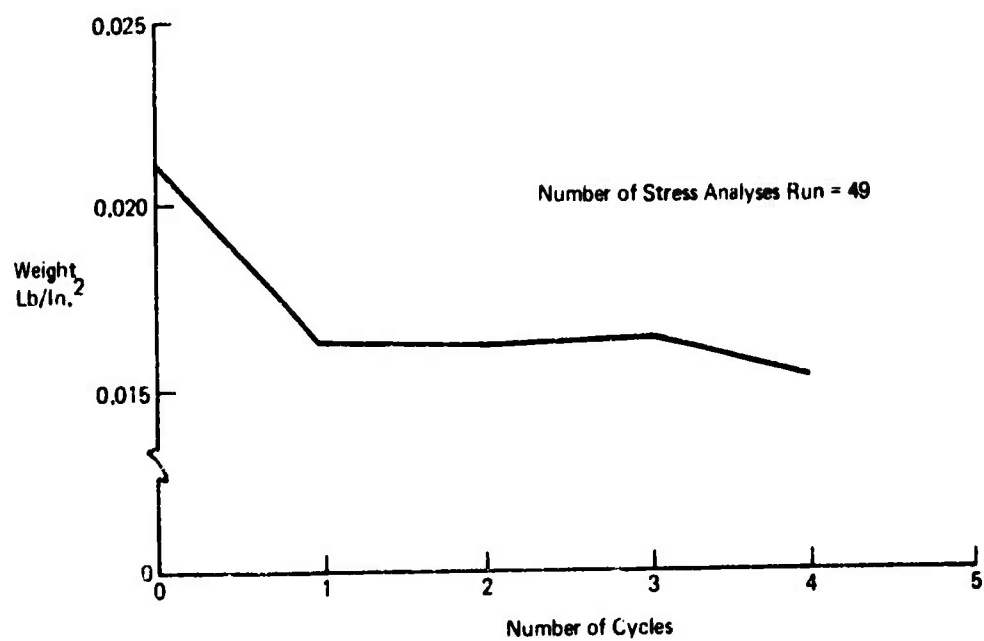


Figure 21. Example of Tetra-Core Optimization.

NONLINEAR ANALYSIS

An analysis of the effect of the nonlinear stress-strain behavior of the fiber-glass-epoxy composite used in Tetra-core legs can be conducted with this program. Stress-strain curves of the composite, determined by cutting out legs from the panel and testing them separately, are used as input, as is the loading increment to be applied⁵. First, the program calculates the stresses and strains in each plate, using the tangent moduli and Poisson's ratios input on Card Type 4 and the loads input on Card Types 12 and 13. Then, a new modulus is calculated for each plate, using the stress-strain curves input for each leg and the strains already calculated. As an example, suppose that the strain in the fiber direction (ϵ_1) of one plate had been + 0.0011 at the end of the first load increment, and that the tension stress-strain curve shown in Figure 22 had been input on Card Type 8 as a series of points for the leg which this plate was in, then the program takes the strain of 0.0011 and uses a linear curve fitting method to determine the stress corresponding to the given strain. This stress is used to compute a secant modulus to be used by the plate for the second iteration:

$$E_{\text{sec}} = \sigma_1 / \epsilon_1$$

The secant modulus is determined in the longitudinal, transverse, and shear directions from the stress-strain curves input for those directions. A different stress-strain curve is input for tension and compression for the longitudinal and transverse directions, and the correct one is chosen by referring to the sign of the strain in that direction.

Once moduli and Poisson's ratio in each plate have been determined, the loads are increased by the amount input on Card Types 12 and 13 and a new stress analysis is run. Then the stresses and strains in each plate are used to determine a new set of secant moduli and Poisson's ratio for each plate. The loads are increased again and the stresses are recalculated. This loading in increments continues until the number of steps specified on Card Type 1 as "NSTP" are completed. As the loads are increased, some plates will reach their failing strain in some direction. When this happens, the secant modulus in that direction is reduced to a small number, which unloads that plate without allowing the nodes to which it is attached to become free, as would happen if the modulus were reduced to zero. On the next loading increment, strains in the adjoining plates will be higher than would be caused by the increase in end load only, since the loads in the failed plate have been dumped into them. This will result in the failure of more plates in

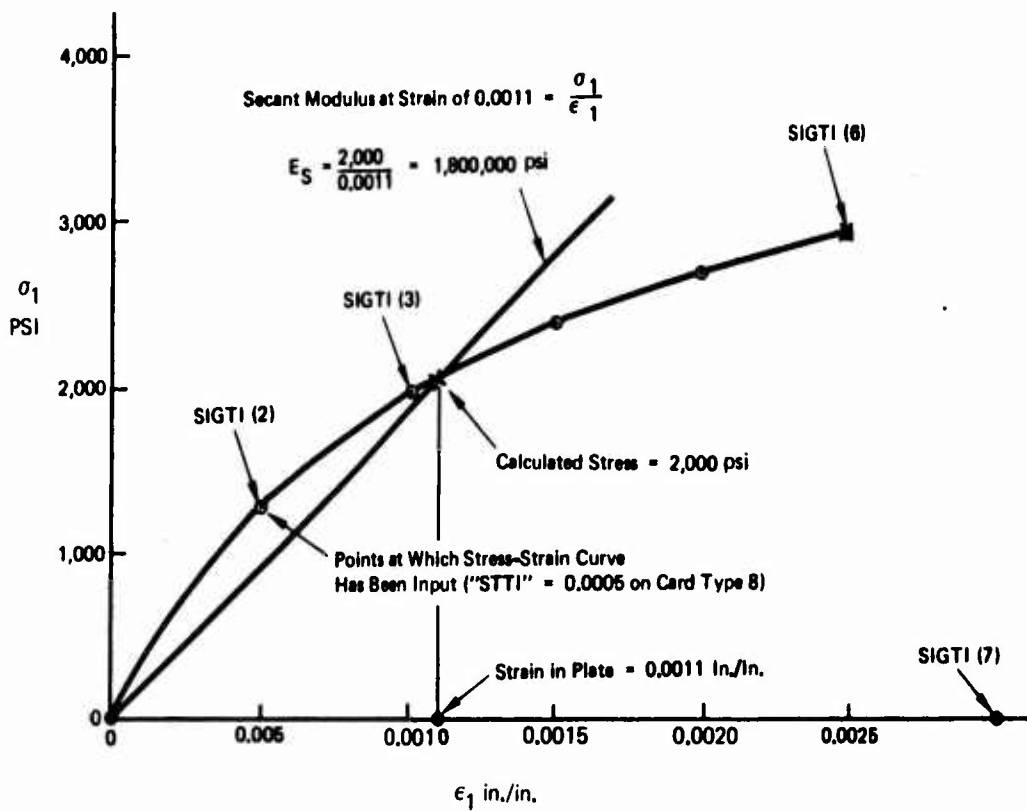


Figure 22. Method Used To Determine Moduli for Plastic Analysis.

the same area. Finally, enough plates will have failed that there is no load path through the structure, resulting in complete failure. Figure 23 illustrates a nonlinear analysis run on a Tetra-core panel with several plates removed to simulate the effect of a hole. Initially, all plates except those that have been removed are intact, but as the load is increased, several plates around the hole fail; and finally at a load of 3,000 pounds/inch, total failure occurs. If a lower increment of load had been used, the spread of failure around the hole would have been more visible.

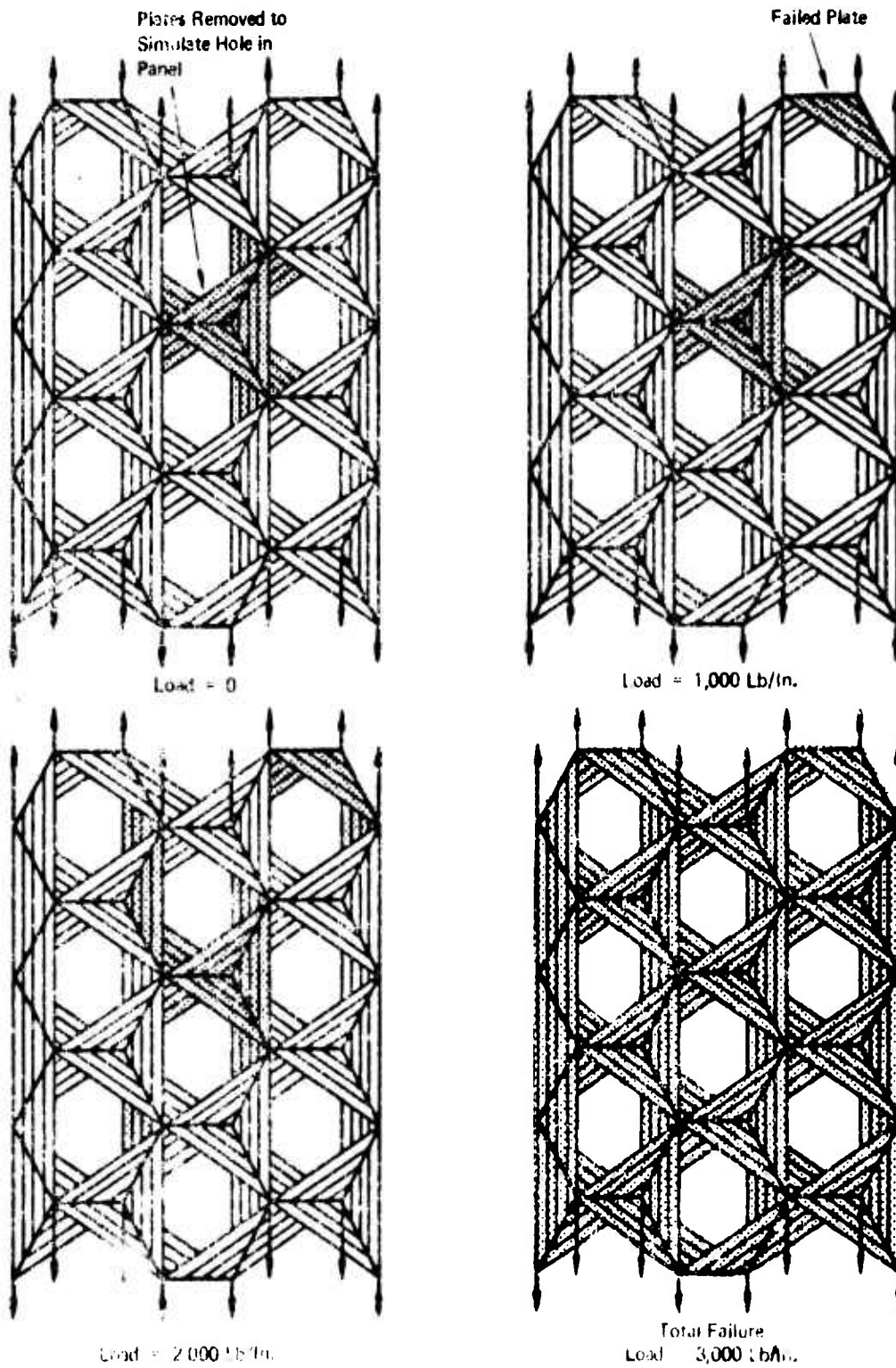


Figure 23. Example of Nonlinear Analysis.

BUCKLING ANALYSIS

A buckling analysis is performed on the plate elements in each leg making up the Tetra-core panels in the subroutine BUKL. No equation was available to calculate the compression buckling load of an orthotropic triangular plate with one edge free, loaded in compression. Therefore, it was decided to use the compression buckling equation (developed by Lackman and Ault⁶) for a rectangular orthotropic plate with one edge free and three edges simply supported. The length of the free edge on the triangle was used as the length of the rectangle, with the tetrahedron height used as a rectangle width. This simplification is expected to result in conservative values for the buckling load, due to the greater restraint imposed on the panel by the sides of the triangle compared with the sides of a rectangle. The equation used to calculate the buckling load is

$$\beta \left[\beta^2 + \left(\frac{\pi b}{a} \right)^2 \lambda \right] \left[\alpha^2 - \left(\frac{\pi b}{a} \right)^2 \mu_{xy} \right] \text{TANH} \alpha = \alpha \left[\alpha^2 - \left(\frac{\pi b}{a} \right)^2 \lambda \right] \left[\beta^2 + \left(\frac{\pi b}{a} \right)^2 \mu_{xy} \right] \text{TAN} \beta,$$

where

$$\alpha = \left(\frac{\pi b}{a} \right)^{1/2} \left[\left(\frac{D_3}{D_2} \frac{\pi b}{a} \right)^2 - \frac{D_1}{D_2} \left(\frac{\pi b}{a} \right)^2 + \frac{N_x b^2}{D_2} \right]^{1/2} + \frac{D_3}{D_2} \left(\frac{\pi b}{a} \right)^{1/2}$$

$$\beta = \left(\frac{\pi b}{a} \right)^{1/2} \left[\left(\frac{D_3}{D_2} \frac{\pi b}{a} \right)^2 - \frac{D_1}{D_2} \left(\frac{\pi b}{a} \right)^2 + \frac{N_x b^2}{D_2} \right]^{1/2} - \frac{D_3}{D_2} \left(\frac{\pi b}{a} \right)^{1/2}$$

$$D_1 = \frac{E_x t^3}{12(1 - \mu_{xy} \mu_{yx})}$$

$$D_2 = \frac{E_y t^3}{12(1 - \mu_{xy} \mu_{yx})}$$

$$D_3 = \frac{2 G_{xy} t^3}{12} + \frac{E_x \mu_{yx} t^3}{12(1 - \mu_{xy} \mu_{yx})}$$

$$\lambda = 4 G_{xy} (1 - \mu_{xy} \mu_{yx}) / E_y + \mu_{xy}$$

a = tetrahedron side length

b = tetrahedron height

t = leg thickness

An iteration method is used to find the lowest value of N_x which will satisfy the equality. This calculation is made for the Vertical, Skew A, and Skew B legs. If face sheets are applied to the Tetra-core panel, then all edges of the triangle will be simply supported and the equation of a simply supported rectangle is used⁷.

$$N_x = 2 \left(\frac{\pi}{b} \right)^2 [(D_1 D_2)^{1/2} + D_3]$$

This equation is used for vertical, Skew A and B legs, and upper and lower face sheets when face sheets are applied. If a more accurate buckling equation becomes available, it can easily be inserted into the buckling subroutine.

MARGIN OF SAFETY

The margin of safety is calculated for each plate element using the von Mises-Hill-Tsai equation⁸.

$$MS = 1 / \left[\frac{\sigma_1^2}{F_x^2} - \frac{F_y}{F_x} \frac{\sigma_1 \sigma_2}{F_x F_y} + \frac{\sigma_2^2}{F_y^2} + \frac{\tau_{12}^2}{F_{xy}^2} \right] - 1$$

σ_1 , σ_2 , and τ_{12} are the stresses as in the local plate element coordinate system, with σ_1 always in the fiber direction. F_x , F_y , and F_{xy} are the allowable stresses, with F_x being the allowable stress in the fiber direction, F_y the transverse allowable, and F_{xy} the shear allowable. Tension or compression allowable stresses (as input on card type 10) are used, depending on the sign of the stress. A margin of safety is calculated for the four locations in the plate for which stresses have been calculated, with the lowest margin being printed out for that plate. The margin of safety in buckling is also calculated, using the buckling stress calculated for the leg in which this plate is found.

$$MS_B = \frac{F}{\sigma_1} - 1$$

The buckling margin is calculated only if one of the σ_i 's is negative, indicating a compression load in the plate. The lowest margin of safety found in each leg for that load case is printed after printing of the plate element stresses.

CALCULATION OF ULTIMATE LOADS

After the margin of safety in each plate has been determined, the minimum margin of safety found in any plate is used to extrapolate the given applied load to determine the load at which the model would have failed.

$$X_{\text{allow}} = X_{\text{load}} / (1. - Y_{\text{MS}})^{.5}$$

$$Y_{\text{allow}} = Y_{\text{load}} / (1. - Y_{\text{MS}})^{.5}$$

$$XY_{\text{allow}} = XY_{\text{load}} / (1. - Y_{\text{MS}})^{.5}$$

$$XM_{\text{allow}} = X_{\text{mom}} / (1. - Y_{\text{MS}})^{.5}$$

$$YM_{\text{allow}} = Y_{\text{mom}} / (1. - Y_{\text{MS}})^{.5}$$

$$XYM_{\text{allow}} = XY_{\text{mom}} / (1. - Y_{\text{MS}})^{.5}$$

$$\text{TORQ}_{\text{allow}} = \text{TORQ} / (1. - Y_{\text{MS}})^{.5}$$

$$XQ_{\text{allow}} = XQ_{\text{SHR}} / (1. - Y_{\text{MS}})^{.5}$$

$$YQ_{\text{allow}} = YQ_{\text{SHR}} / (1. - Y_{\text{MS}})^{.5}$$

where Y_{MS} is the minimum margin of safety found in any plate. X_{load} , Y_{load} , etc., are the input loads from card types 12 and 13. The power of 0.5 may have to be changed to agree with tests when they become available.

CALCULATION OF STIFFNESSES

An equivalent modulus in the x direction is calculated whenever a uniform load is applied in that direction. This is done by taking two x deflections on opposite ends of the panel and adding them to determine the total change in length of the panel (ΔX). This ΔX is divided by the original panel length to get an equivalent strain ϵ_{xEQ} . The uniform load on the x face of the panel, divided by the height of the face, gives an equivalent stress σ_{xEQ} . The equivalent modulus in the x direction is then

$$E_{\text{xEQ}} = \frac{\sigma_{\text{xEQ}}}{\epsilon_{\text{xEQ}}}$$

An equivalent modulus in the y direction is calculated in the same way whenever a uniform load is applied to the y face.

OUTPUT DESCRIPTION

Primary output of the Tetra-core analysis program consists of the finite-element model generated by the data generator overlay and the deflections and stresses calculated by the stress analysis overlay. If the optimization option is used, additional printout is generated during each step of optimization.

The printed output from a standard stress analysis run starts with the input data (number of load cases, size of tetrahedron, leg thicknesses, etc.); the printout of calculated data starts with the number of node points and plate elements generated by the data generator. This node point number does not include the nodes at the midpoint of the triangle or quadrilateral which are added in the stress analysis overlay. The number of load cases is also printed. If a cylinder is generated, the radius to the outer surface is printed. If an airfoil is generated, the chord length is printed. The X, Y, Z coordinates of the generated nodal points are printed. Nodal loads are printed for each load case along with the original node number and the new node numbers that include mid-side nodes added by each linear strain triangle and quadrilateral. Each linear strain element used in the model adds a new node point at the midpoint of two sides of the element where it is attached to another element, which must be accounted for by increasing the size of the load vector and the stiffness matrix. The new node numbers are generated from the node numbers of the vertices on each side of the new node (Figure 24). The new node number is generated by multiplying the larger of the two nodes by 1,000 and adding to the smaller, resulting in a unique new node number. For example, if a new node is to be generated between nodes 5 and 3, as shown in the figure, the resulting number would be 5003. This node is then added into the stiffness matrix and load vector behind the location of the larger of the two vertex nodes. The added mid-side nodes for a typical case are shown in Figure 26.

Plate properties are printed. The node numbering sequence for each plate in the original node point order and as renumbered is printed. Nodes listed under the headings I, J, K, and L give the original plate connectivity. Nodes listed under LP, LQ, LR, LS, LT, and LU give the connectivity using the new nodal numbering sequence. LS and LT represent new mid-side nodes for the triangle. Plate moduli and plate thickness are printed along with the node numberings. EX, EY, and EXY have been divided by the factor $1-\mu_{xy}\mu_{yx}$ for use in the element stiffness generator. EX is the Young's modulus in the direction of nodes I and J, and EY is at an angle of 90 degrees to EX.

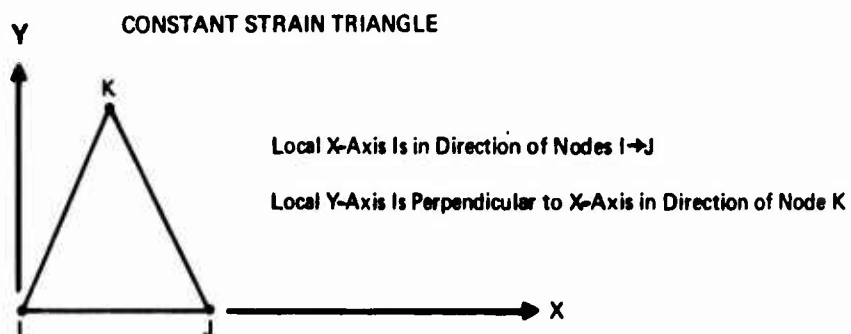
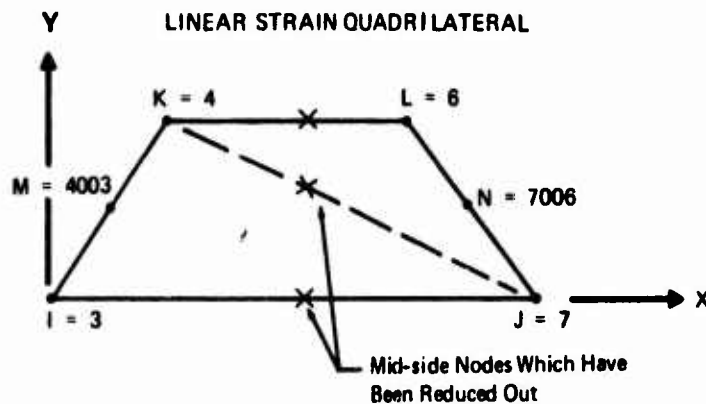
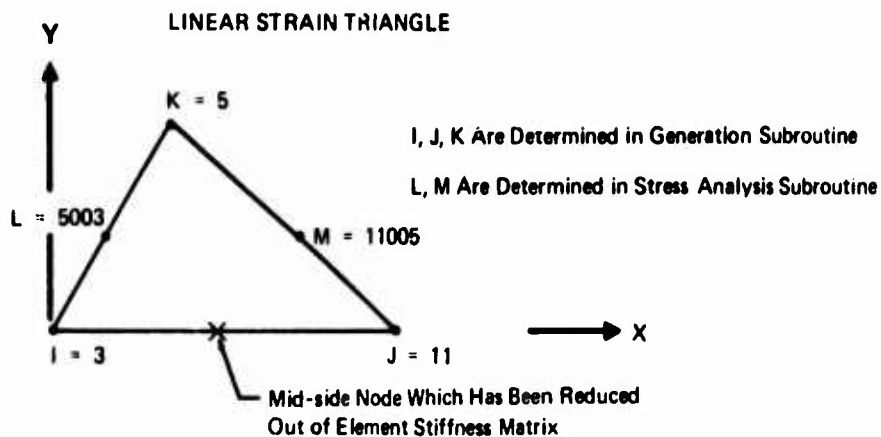


Figure 24. Elements Used in Tetra-Core Program .

Number of Vertex Nodes	= 72
Number of Mid-Side Nodes	= 96
Total Number of Nodes	= 168

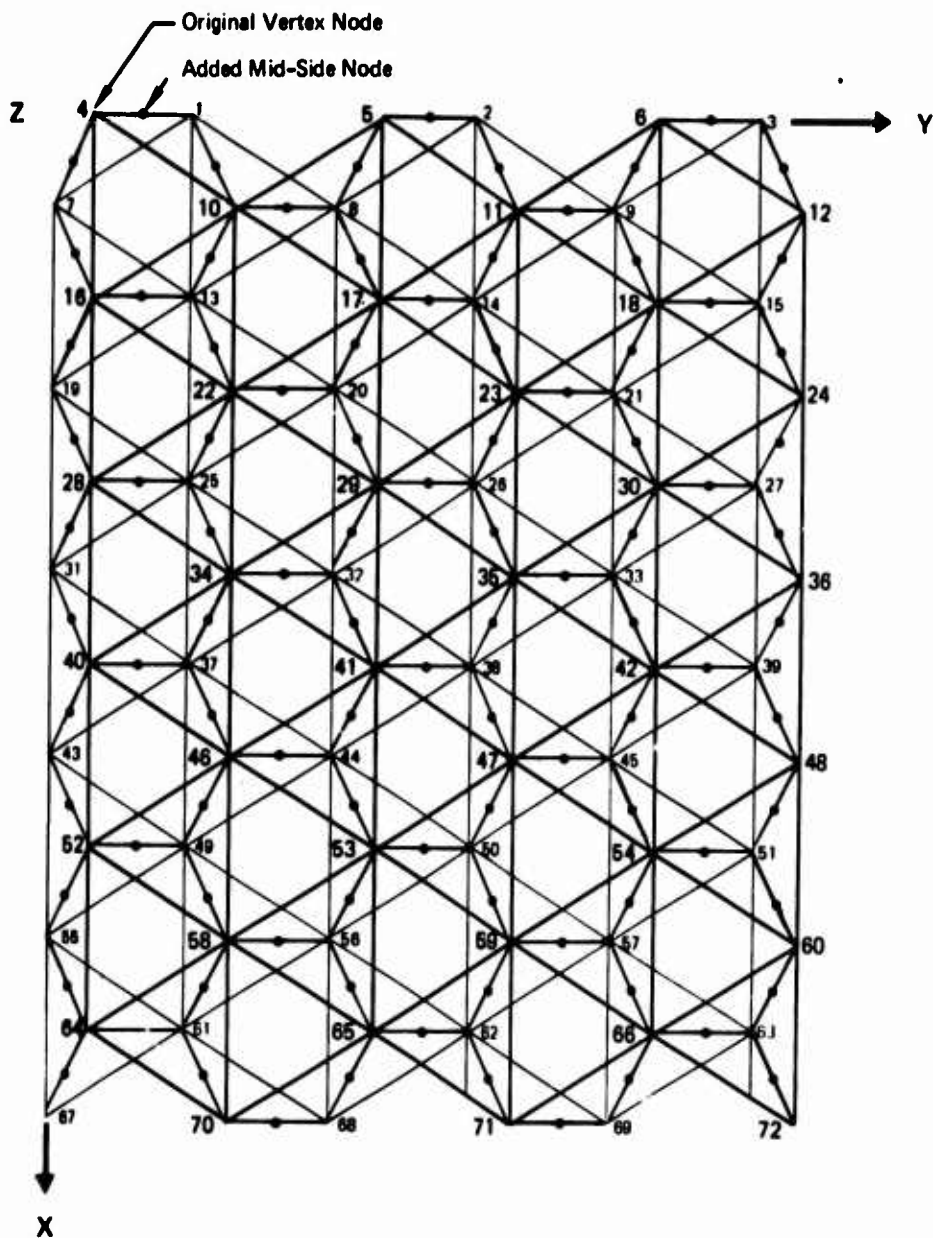


Figure 25. Added Mid-Side Nodes, True Tetra-Core Flat Plate
 $L_X = 6$, $L_Y = 6$.

The 1/2 bandwidth, in nodes, required to store the stiffness matrix is printed after the elements have been renumbered and printed. The storage space required for the overall stiffness matrix can be calculated by multiplying the number of nodes times 3 times the 1/2 bandwidth times 3. If this number added to the amount of space required to store the loads vectors is greater than the available disc space, then the problem will have to be reduced in size.

Nodal deflections and plate stresses are printed for each load case separately. Deflections are printed for each node with the node points numbered in the original sequence (1, 2, 3, 3001, 4, 4002, etc.). Stresses are printed for four locations in the linear strain element used in Vertical and Skew A and Skew B legs, and for the triangle center for the constant strain triangle used in face sheets. The connectivity of each triangle is printed in the original numbering sequence. For a linear strain triangle, the stresses are printed at node I, node J, node K, and at the center of the triangle, in that order. For a linear strain quadrilateral, which is used only in the truncated Tetra-core model, stresses are printed at nodes I, J, K and L. The margin of safety, calculated using the Hill-Tsai failure criterion, in each element and the leg which that element is in (Vertical leg = leg 1, Skew A = leg 2, Skew B = leg 3, lower face = leg 4, upper face = leg 5) are printed also. The buckling stress of a triangle in each leg is printed.

After all the plate stresses in a load case have been printed, the minimum margin of safety found in each leg is printed. Then the minimum margin of safety is used to extrapolate the original loads to predict a load case with the same ratio between each load that would have a margin of safety of zero. If the model has been loaded in the X or Y direction, the deflections of two node points on opposite ends of the model are used to calculate an equivalent modulus. Then printing of the deflections and stresses for the next load case starts.

If the nonlinear option is specified, a series of load increments will be applied until overall failure occurs. Nodal coordinates and load vectors will be printed out only for the first increment, since the coordinates do not change and the succeeding load vectors will be integer multiples of the first load vector. The load intensity (i.e., load in X direction in pounds/inch) is printed at the beginning of each load increment. Plate element moduli are printed at the start of each load increment to allow an inspection of the moduli in each plate to determine the extent of degradation in the panel. As each plate fails in a given direction (longitudinal, transverse, or shear), the modulus in that direction is reduced to near zero. Final failure occurs when all plates have failed. Deflections and stresses are printed out for each load increment.

If the hole option is specified, the moduli of each triangle that connects to a node in the center of the plate, cylinder, or airfoil are set to zero. This node is normally determined by the program but can be input if desired. The identity of the affected node is printed out in either case.

If the optimization option is specified, the coordinates, loads, elements, deflections, and stresses are printed out for the input configuration and for the final optimized configuration only. Intermediate output during each optimization step consists of the current design configuration and the margin of safety in each leg. When partial derivatives are being calculated, the variable that is being changed is printed out as "IK". Direction cosines of the constant weight surface, direction cosines of the constraint surface of each leg, margin of safety in each leg, and direction cosines of the composite constraint surface are printed at the start of each cycle of sidestepping. Weight of the design is checked at the end of each sidestepping cycle to determine if a new optimum has been reached. If so, the design variables are printed and optimization continues. When the input number of sidestep cycles "NSTP" have been completed, the optimization process is stopped and the deflections and stresses for the lowest weight design that was found are printed.

SAMPLE PROBLEMS

Examples of input for several types of analyses will be given for reference. For a nonlinear analysis, the set of cards shown in Figure 26 will be required. Three sets of cards of Types 8 and 9 are required to input stress-strain curves for each leg. The input cards required for an airfoil analysis are shown in Figure 27. An example of the input for an optimization run is shown in Figure 28.

The output from an optimization run is shown in Appendix II.

1	1	0	1	0	0	0	---	Card Type 1
12								
5								
4								
12								Card Type 2
12								
12								
12								
12								
12								
TETRA-CORE PLATE STRESSES - Card Type 3								
400000.	500000.	500000.	500000.	500000.	500000.	.25		Card Type 4
400000.	500000.	500000.	500000.	500000.	500000.	.25		
400000.	500000.	500000.	500000.	500000.	500000.	.25		
4	4	.75	.20	.030	.030	.030		Card Type 5
								Card Type 6 (Blank)
110000.	130000.	130000.	110000.	110000.	130000.	130000.		
110000.	130000.	130000.	110000.	110000.	130000.	130000.		Card Type 10
110000.	130000.	130000.	110000.	110000.	130000.	130000.		Card Type 11
.1	.1	.1	.1	.1	.1	.1		Card Type 12
1000.								Card Type 13 (Blank)

Figure 28. Input Data for Optimization Run.

COMPARISON WITH UNIVERSITY OF KANSAS TESTS

Computer runs have been made to compare to several of the tests run at the University of Kansas⁹. The results of this comparison are not very meaningful, however, because the basic properties of the material making up the legs were not specified in the test document. Information in the test document was limited to side length and tetrahedron height and the fiber used, which was S glass. The program requires elastic moduli and thickness for each leg. It was assumed that the material had the following moduli and thickness:

$$E_X = 4,000,000 \text{ psi}$$

$$E_Y = 500,000 \text{ psi}$$

$$G_{XY} = 500,000 \text{ psi}$$

$$\mu_{XY} = 0.25$$

$$t = 0.030 \text{ in.}$$

Using these as input with a side length of 0.6123 inch and a height of 0.4 inch (comparable to specimen 1 in Reference 9) produced an E_X of 281,000 psi and an E_Y of 284,000 psi. The predicted initial failure load, using a linear stress analysis, was 789 pounds. The cause of this failure was a high transverse tension stress in some Skew A and B plates which would not have resulted in final failure. If the criterion of failure had been fiber stress, the predicted ultimate load would have been 3,300 pounds for a fiber stress of 600,000 psi. This compares to an E_X of 363,053 psi, an E_Y of 303,198 psi, and an ultimate load of 5,800 pounds from the test. However, a comparison between the analysis and test would be risky, since material properties of the specimens used for the test were guessed at. The low predicted load of 789 pounds was caused by failure of the Skew A and B legs in the transverse direction.

An analysis run with a configuration comparable to that of Specimen 3 (HT = 0.5, side = 0.75) was run, resulting in an E_X of 293,400 psi that compares to the test result of E_Y = 288,150 psi. This appears to be good agreement, assuming that the input moduli and thickness were close.

MODEL CONSTRAINTS

The Tetra-core analysis program is dimensioned for 1100 plate elements, 300 vertex nodes, and 600 total nodes. A problem of this size cannot be run on the IBM 360/44 due to a lack of disc space. However, the program can be converted to a larger 360, such as a 360/65, very easily, should one become available. This can be done by removing the call load cards which are used to load a new overlay before referring to it. These cards are not necessary on the operating system used by the larger IBM machines. Also, job control cards must be changed. With a larger machine it will be possible to increase the size of problem that may be run to 600, or greater vertex nodes. This can be done by increasing the X, Y, Z and UX, UY, UZ dimensions in the generation and stress analysis subroutines. The dimensions of variables A and DIAG in subroutine STIFF must be increased. In subroutine SOLPAC, variables DIAG, AI, AJ, B, and X must be increased in size. In subroutine DEFL, variables IDO and B must be increased in size.

In converting to a CDC 6600 machine, it will be necessary to put in OVERLAY and CALL OVERLAY cards for each overlay. Direct access read/write cards used for the IBM 360 will not work on the CDC 6600. Direct-access read/write commands will have to be rewritten for the particular CDC 6600 on which the program is to be run, since they vary from installation to installation.

The truncated Tetra-core model can generate the full range of Tetra-core geometries, from a true Tetra-core to a truncated Tetra-core with vertical sides. However, both the true Tetra-core and the vertical-sided Tetra-core models will have several nodes which are connected to each other by plates generated for the same X, Y, Z coordinates. This will cause a linear dependence in the stiffness matrix, since these nodes will essentially have the same stiffnesses; that is, their rows and columns will be identical. This will result in the matrix being singular, and no solution can be found. This problem will also result when the connected nodes are within a small distance of each other.

The program does not compute nodal reaction force, since this would require more disc storage space than is available on the IBM 360/44 at Fort Eustis. If the program is converted to a larger machine, reactions can be calculated by storing a copy of the merged stiffness matrix and then multiplying it by the calculated deflections: $P = KA$.

COMPUTER PROGRAM

Input Formats for Tetra-Core Analysis Program

<u>CARD</u> <u>TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
1	NLOD	I (1-4)	Number of load cases to be run. If "IOPT" = 1 or "NLIN" = 1 then NLOD must = 1.
	IOPT	I (5-8)	If "IOPT" = 0, no optimization will be done. If "IOPT" = 1, the Tetra-core element will be optimized for the given load case.
	NLIN	I (9-12)	If "NLIN" = 0, standard linear stress analysis will be run. If "NLIN" = 1, a nonlinear analysis will be run using the input load case as the load increment and "NSTP" as the number of iterations.
	ITYP	I (13-16)	If "ITYP" = 1, flat plate analysis If "ITYP" = 2, cylinder analysis If "ITYP" = 3, airfoil analysis
	IFACE	I (17-20)	If "IFACE" = 0, no face sheets will be added to the Tetra-core. If "IFACE" = 1, face sheets will be added.
	IHOLE	I (21-24)	If "IHOLE" = 0, a standard analysis will be run. If "IHOLE" = 1, all plates connected to a grid point in the center of the panel will be defined with zero stiffnesses, giving the effect of a hole in the panel. If "IHOLE" > 1, all plates connected to grid point "IHOLE" will have zero stiffness, allowing a hole to be inserted where desired.
	NSTP	I (25-28)	"NSTP" determines the number of iterations to be made in a nonlinear analysis, or the number of optimization cycles to be made in an optimization run. The default option for "NSTP" is 3.

<u>CARD TYPE</u>	<u>VARIABLE FIELD</u>	<u>COMMENT</u>
ITET	I (29-32)	If "ITET" = 0, standard tetrahedrons will be generated. If "ITET" = 1, truncated tetrahedrons will be generated.
ITEST	I (33-36)	If "ITEST" = 1, the element and merged stiffness matrices will be printed out. If "ITEST" = 0, no printout of matrices.
ILOD	I (37-40)	If "ILOD" = 0, the program will automatically generate nodal loads and fixities. If "ILOD" > 0, nodal loads and/or fixities are input on Card Types 14 and 15, as shown below.

<u>ILOD</u>	<u>Comment</u>
1	Nodal loads are input on Card Type 14. Nodal fixities are automatically generated.
2	Nodal loads are input on Card Type 14. Nodal fixities are input on Card Type 15.
3	Nodal loads are automatically generated. Nodal fixities are input on Card Type 15.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
2	ISA	I (1-4)	Card Type 2 is input only if "OPT" = 1. "ISA", "ISB", "ISC" are used to set variables equal to each other during optimization. 11 cards of type 2 are input.
	ISB	I (1-5)	
	ISC	I (9-12)	During optimization the program will normally change each variable independently. ISA, ISB, ISC can be used to keep several variables, such as leg thicknesses, the same during optimization. ISA can also be used to hold variables constant during optimization by inputting ISA = 12 for that variable. One Card Type 2 is input for each variable, resulting in eleven cards of type 2. The first Card Type 2 is used to control the side length, the second controls height, the third controls vertical leg thickness, etc., as shown below. If ISA = 2 on the first card type 2 and ISA = 1 on the second card type 2 then the first and second variables (side length and height) will be kept the same during optimization. The program changes only the leg thicknesses during the first descent cycle when going from the input design to the initial design on the constraint curve. Therefore, at least one leg thickness must be allowed to vary during optimization or the program will not be able to find an initial feasible design point since it will not have anything to vary.

<u>CARD TYPE 2 NUMBER</u>	<u>VARIABLE CONTROLLED</u>	<u>ISA</u>	<u>ISB</u>	<u>ISC</u>
1	Side Length	12		
2	Height	12		
3	Vertical Leg Thickness	4	5	
4	Skew A Leg Thickness	3	5	

<u>CARD TYPE</u> <u>2 NUMBER</u>	<u>VARIABLE</u> <u>CONTROLLED</u>	<u>ISA</u>	<u>ISB</u>	<u>ISC</u>
5	Skew B Leg Thickness	3	4	
6	Upper Face Sheet Thickness	12		
7	Lower Face Sheet Thickness	12		
8	THETA	12		
9	Y1OFF	12		
10	Y2OFF	12		
11	THT	12		

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
3	Title	A(1-48)	Title Card

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
4	EX	F(1-12)	Elastic modulus in longitudinal direction of leg, psi.
	EY	F(13-24)	Elastic modulus in transverse direction of leg, psi.
	GXY	F(25-36)	Shear modulus of leg, psi.
	UXY	F(37-48)	Poisson's Ratio

One card type 4 must be input
for each leg (Vertical Skew A,
Skew B) and for the upper and
lower faces if "IFACE" = 1.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
5	LX	I(1-4)	Number of legs in X face of Tetra-core "LY" times the length of a tetrahedron times 0.866 gives the panel width.
	LY	I(5-8)	Number of legs in Y face of Tetra-core "LY" times the length of a tetrahedron side gives the panel length.
	SIDE	F(13-18)	Length of a tetrahedron side, in.
	HT	F(19-24)	Height of Tetra-Core Panel, in.
	THK(1)	F(25-30)	Thickness of Vertical Legs, in.
	THK(2)	F(31-36)	Thickness of Skew A Legs, in.
	THK(3)	F(37-42)	Thickness of Skew B Legs, in.
	THK(4)	F(43-48)	Thickness of Lower Face Sheet, in.
	THK(5)	F(49-54)	Thickness of Upper Face Sheet, in.
			"THK(4)" and "THK(5)" are input only if "IFACE" = 1 on card type 1.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
6	THETA	F(1-12)	The variables input on this card control the variation of . of the Tetra-core element from a true tetrahedron. See Figure 4 for a description.
	Y1OFF	F(13-24)	
	Y2OFF	F(25-36)	
	THT	F(37-48)	

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
7	TC	F(1-12)	Thickness/Chord Ratio
	XC	F(13-24)	X/Chord Ratio
Card Type 6 is input only if "ITYP" = 3 on card 1, that is, only if an airfoil shape is to be generated. Eleven cards of type 6 must be input, each with a thickness/chord for the specified X/Chord. The X/Chord at the nose of the airfoil is input first, with X/Chord for the tail of the airfoil last. The intermediate X/Chord cards must be in sequence from nose to tail.			

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
8	STT1	F(1-12)	Longitudinal tension strain increment, in./in.
	STT2	F(13-24)	Transverse tension strain increment, in./in.
	STC1	F(25-36)	Longitudinal compression strain increment, in./in.
	STC2	F(37-48)	Transverse compression strain increment, in./in.
	STSS	F(49-60)	Shear strain increment, in./in.
			"STT1", "STT2", "STC1", "STC2", "STSS" are the strain increments used in inputting tension and compression stress-strain curves on cards of Type 9. The stress-strain curves are broken down into increments of "STT1", "STT2," etc., for inputting on card 9.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
9	SIGT(I) I = 1,7	F(1-12) F(13-25)... F(61-72) F(73-80)	Stress-strain curves of the material used in each leg and each face sheet are input if a nonlinear analysis is performed ("NLIN" = 1, card 1). Longitudinal tension stresses are input for strains of 0, STT1, 2STT1, 3STT1,...6STT1, for example.

The first card following card 8 must contain the longitudinal tension stresses; the second card contains the transverse tension stresses, etc. The complete sequence is shown below.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>CONTENTS</u>
8		Strain increments - "STT1," "STT2," "STC1," "STC2," "STSS.", in./in.
9	SIGT1	Longitudinal tension stresses in increments of strain "STT1.", psi.
9	SIGT2	Transverse tension stresses in increments of strain "STT2.", psi.
9	SIGC1	Longitudinal compression stresses in increments of strain "STC1.", psi.
9	SIGC2	Transverse compression stresses in increments of strain "STC2.", psi.
9	PRXT	Poisson's ratio μ_{XY} in tension in increments of "STT1."
9	PRXC	Poisson's ratio μ_{XY} in compression increments of "STC1."

One set of Card 8 + six card 9's must be input for each leg of the Tetra-core. If "IFACE" = 0 on card 1, then three sets must be input- Vertical legs, Skew A legs, and Skew B legs. If "IFACE" = 1, then five sets must be input- three sets for the legs, plus one set for the lower face sheet, and one set for the upper face sheet.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
10	FXT	F(1-12)	Longitudinal tension allowable stress, psi.
	FYT	F(13-24)	Transverse tension allowable stress, psi.
	FXC	F(25-36)	Longitudinal compression allowable stresses, psi.
	FYC	F(37-48)	Transverse compression allowable stress, psi.
	SSS	F(49-60)	Shear Allowable Stress, psi.

One card type 9 must be input for each leg. If "IFACE" = 1, five card 9's are input, one each for the Vertical legs, Skew A legs, Skew B legs, lower face sheet, upper face sheet. If "IFACE" = 0 only the first three are input.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
11	RHO(1)	F(1-12)	Card 10 is input only if "IOPT" = 1 on card 1. Density of material used in vertical legs, lb/in. ³ .
	RHO(2)	F(13-24)	Density of material used in Skew A legs, lb/in. ³ .
	RHO(3)	F(25-36)	Density of material used in Skew B legs, lb/in. ³ .
	RHO(4)	F(37-48)	Density of material used in lower face sheet, lb/in. ³ .
	RHO(5)	F(49-60)	Density of material used in upper face sheet, lb/in. ³ .
			The density is used to calculate the weight of the Tetra-core for use in optimization.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
12	XLOAD	F(1-12)	Load on XFACE of flat plate Tetra-core, lb/in.; or total end load on cylinder or airfoil, lb.
	YLOAD	F(13-24)	Load on YFACE of flat plate Tetra-core, lb/in.
	XYLOD	F(25-36)	Shear load on flat plate Tetra- core, lb/in.
	XMOM	F(37-48)	Moment on XFACE of flat plate Tetra-core in.-lb/in.; or total moment on cylinder or airfoil, in.-lb.
	YMOM	F(49-60)	Moment on YFACE of flat plate Tetra-core in.-lb/in.
	XYMOM	F(61-72)	Twisting moment on flat or plate Tetra-core in.-lb/in.
	TORQ	F(73-80)	Torque on cylinder or airfoil of Tetra-core, in.-lb/in.

<u>CARD</u> <u>TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
13	XQSHR	F(1-12)	Shear load on XFACE of flat plate Tetra-core, lb/in.
	YQSHR	F(13-24)	Shear load on YFACE of flat plate Tetra-core, lb/in.

There is an upper limit of 10 on the number of load cases that may be run.

If a nonlinear analysis is to be made (NLIN" = 1) or an optimization is to be run ("IOPT" = 1), only one load case is allowed.

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
14	J	I (1 - 4)	Card type 14 is input only if "ILOD" = 1 or 2 on card 1. Node which loads input on this card will be applied to.
	UX (J)	F (12-24)	Load in X direction on node J
	UY (J)	F (25-36)	Load in Y direction on node J
	UZ (J)	F (37-48)	Load in Z direction on node J
A card Type 14 is input for each node being loaded. Input of Card Type 14 is terminated by a card with J = 300.			

<u>CARD TYPE</u>	<u>VARIABLE</u>	<u>FIELD</u>	<u>COMMENT</u>
15			Card Type 15 is input only if "ILOD" = 2 or 3 on Card 1.
	J	I (1 - 4)	Node for which fixity is read in.
	Code (J)	F (13-24)	Fixity of node J. Convention shown below is used.
	<u>Code (J)</u>	<u>Freedoms Fixed</u>	
	0.0	None	
	1.0	X	
	2.0	Y	
	3.0	Z	
	4.0	X, Y	
	5.0	X, Z	
	6.0	Y, Z	
	7.0	X, Y, Z	

Card Formats

Input to the Tetra-core analysis program must be in the form of data cards punched according to fixed formats.

Integer Fields: I (1-4), I (26-30), etc.

An integer must be right adjusted in the field. Unneeded field space may be left blank. Blanks are interpreted as zeros in the corresponding column. Decimal points are not used.

Decimal Fields: F(1-12), F(37-48), etc.

A decimal number, punched with a decimal point, may be located anywhere within the field. Blank fields are interpreted as zeros.

Alphanumeric Fields: A(1-27), A(21-30), etc.

Alphanumeric characters may be anywhere in the field. Legal characters for this field include all FORTRAN characters.

LIST OF SUBROUTINES

OVERLAY TETRA 1 - ROOT OVERLAY

- MAIN - Main program. Calls subroutines GEN, STRSS, SOLPAC, DEFL.
- OUTIN - Used to read and write stiffness matrix and load vectors on file 4. Called by STIFF, SOLPAC, DEFL.

OVERLAY TETRA 2

- GEN - Subroutine to read and print input data. Calls subroutines NODGEN, PLATEN, LODGEN, NODGAN, PLATGA, LODGAN, OPTM, NONLIN. Called by MAIN.
- CURV - Curve fitting subroutine, used in generation of airfoil shape. Called by NODGEN, NODGAN.
- OPTM - Optimization subroutine. Determines direction of travel to obtain minimum weight design. Calls WEIGHT, DAR. Called by GEN.
- WEIGHT - Subroutine to calculate weight of Tetra-core model being optimized. Calls NODGEN, NODGAN. Called by OPTM.
- DAR - Subroutine to calculate direction cosines of gradient given partial derivatives. Called by OPTM.
- NONLIN - Subroutine to increment loads before each step of nonlinear analysis. Called by GEN.

OVERLAY TETRA 3

- NODGEN - Subroutine to generate node points for true Tetra-core model. Calls CURV.

OVERLAY TETRA 4

- PLATGN - Subroutine to generate plates for true Tetra-core model. Called by GEN.

OVERLAY TETRA 5

- LODGEN - Subroutine to generate nodal loads and nodal fixities for true Tetra-core model. Called by GEN.

OVERLAY TETRA 6

NODGAN - Subroutine to generate node points for truncated Tetra-core model. Called by GEN, WEIGHT.

OVERLAY TETRA 7

PLATGA - Subroutine to generate plates for truncated Tetra-core model. Called by GEN.

OVERLAY TETRA 8

LODGAN - Subroutine to generate nodal loads and nodal fixities for truncated Tetra-core model. Called by GEN.

OVERLAY TETRA 9

STRSS - Overlay to perform finite-element stress analysis on model generated by previous overlays. Called by MAIN. Calls STRESS, STIFF.

OVERLAY TETRA 10

STRESS - Subroutine to renumber Tetra-core model to account for midside nodes. Called by STRSS.

NID - Function to generate new number for midpoint node.

OVERLAY TETRA 11

STIFF - Subroutine to merge element stiffness matrices into overall stiffness matrix. Called by STRSS. Calls TRPRD, TRIM6, OUTIN.

TRPRD - Subroutine to perform matrix multiplication $[A]^T[B][A]$. Called by TRIM6.

TRIM6 - Subroutine to generate linear strain triangle and quadrilateral and constant strain triangular stiffness and stress matrices. Called by STIFF. Calls TRPRD.

OVERLAY TETRA 12

SOLPAC - Subroutine for solution of stiffness matrix. Solves Eqn $P = KA$ using Choleski triangularization and back substitution. Called by MAIN.

OVERLAY TETRA 13

DEFL - Subroutine for printing of deflections, calculation of plate element stresses. Called by MAIN. Calls MARGIN, MODUL, BUKL, PLST.

MARGIN - Subroutine to calculate margin of safety in each plate element using von Mises-Tsai failure criterion. Called by DEFL.

MODUL - Subroutine to calculate plate secant moduli based on strains within the plate. Called by DEFL.

BUKL - Subroutine to calculate buckling stress for typical plate from each leg. Called by DEFL.

PLST - Function to calculate stress given the strain in a plate.

DESCRIPTION OF VARIABLES USED IN COMMON

COMMON/COMN/

NUMNP -- Number of nodal points in model

NUMEL - Number of plate elements in model

NLOD - Number of load cases

NLIN - Used to control nonlinear analysis

MBAND - 1/2 bandwidth of stiffness matrix, in freedoms

NBAND - 1/2 bandwidth of stiffness matrix, in nodes

COMMON/MATL/

E - Moduli of elasticity for each leg

FXT, FYT, FXC, FYC, SSS - Allowable stresses of material in each leg

CMS - Margin of safety in each leg

COMMON/LODA/

XLOAD, YLOAD, XYLOD, XMOM, YMOM, XYMOM, TORQ, XQHR, YQSHR - input applied loads

COMMON/ONE/

SIDE, HT - Tetrahedron side length and actual height
(see Figure 4)

THK - Thickness of each leg

THETA, Y1OFF, Y2OFF - Parameters to control variation
from theoretical equalateral tetra-
hedron (see Figure 4)

THT - Theoretical height at which tetrahedron sides would
meet (see Figure 4)

DUM - Not used

TC, XC - Thickness/chord and X/chord ratios used to
generate airfoil model

HED - Title

NX, NY - Number of legs on X and Y face. Called LX and LY
on input sheets (see Figure 5)

IOPT - Used to control optimization

ITYP - Type of model to be run--flat plate, cylinder,
airfoil

IFACE - Control of face sheet generation

IHOLE - Used to control if hole will be placed in model
or not

ITET - Used to control type of tetrahedron generated--
true or truncated

ICNT - Counter for number of steps that have been run
in nonlinear or optimization analysis

NNEW - Used to determine which leg a given plate is in

COMMON/OPTA/

NSTP - Limiting number of steps for nonlinear analysis.
limiting number of cycles for optimization

IPRT - Controls printing of intermediate printout in
optimization

IDS - Controls steepest descent mode in optimization

IDRV - Controls calculation of derivatives in optimization

ITOT - Counter for number of cycles in optimization

ICYC - Counter for number of sidesteps in optimization

LAST - Not used

IK - Used to identify variable being incremented when calculating derivatives in optimization

WTL - Base weight before start of calculation of derivatives in optimization

XMSL - Margin of safety at start of last sidestep in optimization

XMS - Present margin of safety

WTMN - Minimum weight found in optimization

TMLT - Step length used in sidestepping - optimization

WT - Present weight

HMAX - Value of variables associated with WTMN - optimization

HL - Value of variables at start of calculation of derivatives - optimization

CMSA - Value of margins of safety at start of calculation of derivatives - optimization

SLW - Partial derivatives of effect of variables on weight - optimization

SLOPE - Partial derivatives of effect of variables on margin of safety - optimization

RHO - Density of material in each leg

PWT - Gradient of constant weight surface - optimization

PMT - Composite constraint gradient - in optimization

ISA, ISB, ISC - Used to control variables during optimization

COMMON/MODU/

SIGT1, SIGT2, SIGC1, SIGC2, SIGSS - Stresses defining stress-strain curves used in nonlinear analysis

PRXT, PRXC - Poisson's ratios used in nonlinear analysis

STT1, STT2, STC2, STSS - Strain increments used to define stress-strain curves for nonlinear analysis

COMMON/LODB/

APPLD - XLOAD, YLOAD, etc. for each load case are stored in APPLD

COMMON/RAX/

NR4 - Pointer for next record to be read on File 4

NR4LOD - Gives location of first record in loads vector on File 4

NR4STF - Gives location of first record in stiffness matrix on File 4

NR4DIA - Gives location of first record in diagonal on File 4

NREC - Number of records required to store three rows of stiffness matrix on File 4

NRECST - Number of records in stiffness matrix

NRECLD - Number of records in loads vector

NR1 - Pointer for next record to be read on File 1

NR2 - Pointer for next record to be read on File 2

DESCRIPTION OF VARIABLES USED IN SUBROUTINES

SUBROUTINES - GEN, NODGEN, PLATGN, LODGEN, NODGAN, PLATGA, LODGAN, WEIGHT

X, Y, Z - Nodal coordinates

SLOPA - Slope of surface at each node--used for airfoil

NA, NB, NC, ND - Nodes at vertices of plate element

CODE - Fixity for each node

UX, UY, UZ - Loads on each node

SUBROUTINE - OPTM

SLP - Intermediate step in calculation of composite constraint gradient

PMS - Gradient to each margin of safety surface

H - Vector of variables to be optimized

SUBROUTINE - STRESS, STIFF, TRIM6

X,Y,Z,UX,UY,UZ,CODE - See subroutines NODGEN, PLATGEN, etc.

IDO - Node numbers as renumbered to include mid-side nodes

ID - Node numbers in original order, not including midside nodes

NI - Node number of new mid-side node

IN - Sequence of new mid-side node

ML - Added mid-side nodes in plate element

LM - Original vertex nodes in plate element - in original nodal numbering sequence

MM - Renumbered mid-side and vertex nodes in plate element

DUM1,DUM2,DUM3,DUM4 - Dummy variables used in renumbering of load vector and nodal fixities to include the effect of added midside nodes

S - Element stiffness matrix

ETR,ELGT,EPRM, G - Plate elasticity matrix - $ETR = EY/(1-\mu_{xy}\mu_{yx})$, $ELGT = EX/(1-\mu_{xy}\mu_{yx})$, $EPRM = EY\mu_{xy}/(1-\mu_{xy}\mu_{yx})$, $G = GXY$

T - Plate thickness

ST - Plate stress matrix

IMTL - Indicates leg that plate is in

A - Three rows of merged stiffness matrix
 DIAG - Diagonal of merged stiffness matrix
 PHYX, PHYX - Used in computing element stiffness matrix
 ALAM - Element transformation matrix. Transforms
 stiffnesses from local to global coordinates
 AL, AL2 - Direction cosines of element
 QMAT - Elasticity matrix
 AST, AK21, DUM, STA, STR, STRA - Dummy matrices used in
 calculating stiffness and
 stress matrices
 ZETA - Area coordinates of location in triangle for
 which stresses are calculated
 IK - Order in which triangle stiffnesses are added into
 quadrilateral stiffness matrix

SUBROUTINE SOLPAC

DIAG - Diagonal of stiffness matrix
 AI - Three rows of stiffness matrix
 AJ - Three columns of stiffness matrix
 B - Initial loads vector, deflections after solution
 is complete
 X - Loads vector after reduction

SUBROUTINES DEFL, BUKL, MODUL

P - Displacements for nodes connecting to an element
 LM, MM, ST, IDO - See subroutine STRESS
 STRESS - Element stresses
 FBKL - Buckling stress of an element in each leg
 STRAIN - Element strains
 B - Nodal deflections
 D - Plate stiffnesses
 XL - Width of plate

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
//ASSI 111806
// ALLUC ONE,091='SYSWK1',300,FMT
// LABEL 120
//
// 111832
//JC710280 JOB ,ST031100060C
//ASSI 111926
// ALLOC TWO,091='SYSWK1',300,FMT
// LABEL 60
//
// 111951
```

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
//ASSI 113800
// ALLOC FOUR,091='SYSWK1',300,FMT
// LABEL 2922
//
// 113844
```

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
//ASSI 113857
// EXEC UTILS
```

Nov. 17, 1971

```
MAP DVADM=091,VOL ID='SYSWK1'
MAP OF 'SYSWK1'
```

DATA SET NAME	SEQ	TYPE	LOW EXTENT	HIGH EXTENT	EXPIRED	LAST REC. WRITTEN
DVE	0	01	CC 000 MM 1	CC 019 MM 0		CC 019 MM 0 R 08
TWO	0	01	CC 019 MM 1	CC 038 MM 0		CC 038 MM 0 R 08
FOUR	0	01	CC 038 MM 1	CC 188 MM 0		CC 188 MM 0 R 01
VTOC	0	01	CC 199 MM 0	CC 199 MM 1		

```
//JC710280 JOB ,ST031100060C
//ASSI 141024
//STRESS EXEC FORTRAN(BCD,MAP)
```

```
//JC710280 JOB ,ST031100060C
//ASSI 141139
//TETNN EXEC FORTRAN(BCD,MAP)
```

```

FORTRAN IV      MODEL 44  PS      VERSION 3,  LEVEL 4   DATE 71321      PAGE 0001
C
0001 ***** ROOT OVERLAY *****
0002 ***** ROOT OVERLAY *****
0003 COMMON / COMN / NJMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
0004 COMMON / MATL / F14(5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
0005 COMMON / LODA / XLOAD,YLOAD,XVLOAD,XMOM,YMOM,XVMOM,YVMOM,XQSHR,YQSHR
0006 COMMON / DVE / SIDE,HT,THK(5),THETA,YIOFF,YZOFF,THT,DUM,IC(11)
0007 1,XC(11),HED(12),NX,NY,IOP1,IYPI,IFACE,INOLE,ITET,ICNT,NMEM(4)
0008 COMMON / OPTA / NSTP,IPRT,IDS,IDRV,ITOT,ICYC,LAST,IU, WTL,XMSL
0009 1,XMS,WTMN,TMLT,MT, HMAX(11),HL(11),CMSA(11)
0010 2,SLW(11),SLCPE(11,5),RHO(5), PNT(11),PMT(11)
0011 3,ISAI(11),ISBI(11),ISC(11)
0012 COMMON / MODU / SIGT( 5,9),SIGT2( 5,9),SIGC( 5,9),SIGC2( 5,9)
0013 1,SIGSSI( 5,9),PRXT( 5,9),PRXC( 5,9),STT1( 5),STT2( 5),STC1( 5)
0014 2,STC2( 5),STSS( 5)
0015 COMMON / LODB / APPLD(9,10)
0016 COMMON / RAX / NR4,NR5LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLO,NR1,NR2
0017 COMMON / DJMP / ITEST
0018 DEFINE FILE 1(300,120,U,NR1)
0019 DEFINE FILE 2(300,60,U,NR2)
0020 DEFINE FILE 4(300,2922,U,NR4)
0021 NREC=890
0022 50 IU = 0
0023 1CYC = 0
0024 1DXY = 0
0025 1CNT = 0
0026 1TOT = 0
0027 1OPT = 0
0028 1DS = 1
0029 TMLT = 1.0
0030 WTMN = 100.
0031 THK(4) = 0.0
0032 THK(5) = 0.0
0033 DUM = 0.0
0034 DO 100 I = 1,11
0035 1SA(1) = 0
0036 1SB(1) = 0
0037 1SC(1) = 0
0038 100 SLW(1) = 1.0
0039 10 CONTINUE
0040 CALL LOAD('TETRA1')
0041 CALL GEN
0042 CALL LOAD('TETRA9')
0043 CALL STRSS
0044 CALL LOAD('TETRA11')
0045 CALL SOLPAC
0046 CALL LOAD('TETRA12')
0047 CALL DEFL
0048 IF (1OPT.EQ.0) -AND. NLIN.EQ.0) GO TO 50
0049 IF (1OPT.EQ.5) GO TO 50
FORTRAN IV      MODEL 44  PS      VERSION 3,  LEVEL 4   DATE 71321      PAGE 0002
0042 GO TO 10
0043 END

```

SUBOL NUMP NUMBER	LOCATION 000000 000014	SUBOL NUMEL NBN	COMMON BLOCK / COMN / MAP SIZE LOCATION 000004 000018 SYMBOL MLSD NEL 000008 00001C	LOCATION 00000C	SUBOL MBAND	LOCATION 000010
SUBOL E SSS	LOCATION 000000 0000A0	SUBOL FXT CMS	COMMON BLOCK / MATL / MAP SIZE LOCATION 000050 0000B4 SYMBOL FYT 000064 FNC 0000DC	LOCATION 000078	SUBOL FVC	LOCATION 00008C
SUBOL XLCAO XMDM	LOCATION 000000 000014	SUBOL YLOAD TORQ	COMMON BLOCK / LOOA / MAP SIZE LOCATION 000004 000018 SYMBOL XLLOD XSHR 000008 00001C	LOCATION 00000C 000020	SUBOL YMDH	LOCATION 000010
SUBOL SIDE Y2OFF HED IFACE	LOCATION 000000 000024 000088 0000C8	SUBOL HT THT NX THOLE	COMMON BLOCK / ONE / MAP SIZE LOCATION 000004 000028 000088 0000CC SYMBOL THK DUM NY ITET 000008 00002C 00008C 000030 0000E8	LOCATION 00001C 000030 0000C0 0000D4	SUBOL YLOFF XC ITYP NNEH	LOCATION 000020 00005C 0000C4 0000D8
SUBOL NSTP ICVC AMS HL PMT	LOCATION 000000 000014 000028 000054 0001D8	SUBOL IPRT LAST WTN CMSA PMT	COMMON BLOCK / OPTA / MAP SIZE LOCATION 000004 000018 00002C 000090 000204 SYMBOL IDS IU TMLT SLW ISA 000008 00001C 000030 00008C 000230	LOCATION 00000C 000020 000034 0000E8 00025C	SUBOL ITOT XMSL HMAX RHO ISC	LOCATION 000010 000024 000039 0001C4 000288
SUBOL SIGT1 PRXT STCZ	LOCATION 000000 000084 000528	SUBOL SIGT2 PRXC STSS	COMMON BLOCK / MDDU / MAP SIZE LOCATION 000084 000438 00053C SYMBOL SIGC1 STT1 000168 0004EC	LOCATION 00021C 000500	SUBOL SIGSS STCL	LOCATION 000200 000514
SUBOL APLO	LOCATION 000000	SUBOL	COMMON BLOCK / LODR / MAP SIZE LOCATION 000000	LOCATION	SUBOL	LOCATION
SUBOL NR4 NRECS	LOCATION 000000 000014	SUBOL NR4LOD NRECLD	COMMON BLOCK / RAX / MAP SIZE LOCATION 000004 000018 SYMBOL NR4STF NR1 000008 00001C	LOCATION 00000C 000020	SUBOL NREC	LOCATION 000010
SUBOL	LOCATION	SUBOL	COMMON BLOCK / DUMP / MAP SIZE LOCATION 000000	LOCATION	SUBOL	LOCATION

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
I TEST 000000									
PAGE 0004									
SCALAR MAP									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
1	000108								
SUBPROGRAMS CALLED									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
DIOCS8	00010C	LOAD	000110	GEN	000114	STRSS	000118	SOLPAC	00011C
DEFL	000120	18COM8	000124						
LABEL MAP									
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
50	0001CC	100	000268	10	000282				
TOTAL MEMORY REQUIREMENTS 000358 BYTES									
COMPILER HIGHEST SEVERITY CODE WAS 0									
//OUTIN EXEC FORTRAN(BCD,MAP)									
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
PAGE 0001									
SUBROUTINE OUTIN(M,A,M)									
0001	COMMON/RAX/NR4, NR4LDD, NR4STF, NR4DIA, NREC, NRECLD, NR1, NR2								
0002	DIMENSION AIM)								
0003	IF(M.EQ.1) WRITE(4*NR4) A								
0004	IF(M.EQ.2) READ(4*NR4) A								
0005	RETURN								
0006									
0007	END								
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
PAGE 0002									
COMMON BLOCK / RAX / MAP SIZE 000024									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000000	NR4LDD	000004	NR4STF	000008	NR4DIA	00000C	NREC	000010
NRECLD	000014	NRECLD	000018	NR1	00001C	NR2	000020		
SCALAR MAP									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
M	00000C	N	0000E0						
ARRAY MAP									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	0000E4								
SUBPROGRAMS CALLED									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
18COM8	0000E8								
TOTAL MEMORY REQUIREMENTS 00026C BYTES									
COMPILER HIGHEST SEVERITY CODE WAS 0									
//GEN EXEC FORTRAN(BCD,MAP)									

C ***** OVERLAY A *****

```

0001 SUBROUTINE GEN
0002 COMMON / COMN / NUMP,MUMEL,NLOD,NLIN,MRAND,NUMBLK,NBY,NEL(20)
0003 COMMON / MATL / E(4,5),FXT(5),FYT(5),FZC(5),FZC(5),SSS(5),CMS(10)
0004 COMMON / LODA / XLOD,YLOD,XMOD,YMOD,XFOM,YFOM,XQSHR,YQSHR
0005 COMMON / ONE / SIDE,MT,THK(5),TMETA,YLOFF,YZOFF,TMT,DUM,TC(11)
0006 1 *XC(11),MED(12),NX,NY,IOP1,ITYP,IFACE,IMOLE,ITET,ICNT,MNEW(4)
0007 COMMON / OPTA / NSTP,IPRT,IDS,IDRV,ITOT,ICYC,LAST,IU, WTL,XMSL
0008 1 *XMS,MTPM,TALT,MT, HMAX(11),ML(11),CMSA(11)
0009 2 *SLW(11),SLOPE(11,5),RHO(5),PMT(11),PMT(11)
0010 3 *ISA(11),ISB(11),ISC(11)
0011 COMMON / MODU / SIGT( 5,9),SIGT2( 5,9),SIGC(1 5,9),SIGC2( 5,9)
0012 1 *SIGSS( 5,9),PRAT( 5,9),PRXCF( 5,9),STT( 5),STT2( 5),STC( 5)
0013 2 *STC2( 5),STSS( 5)
0014 COMMON /LODB / APPLD(9,10)
0015 DIMENSION X(300),Y(300),Z(300),SLOPA(300)
0016 COMMON /DUMP / ITET
0017 REMIND 3
0018 IF (ICNT .GT. 0) GO TO 210
0019 DO 50 I = 1,11
0020 TC(I) = 0.0
0021 50 XC(I) = 0.0
0022 READ (5,5023) NLOD,IOP1,NLIN,ITYP,IFACE,IMOLE,NSTP,ITET,ITET
0023 FORMAT (9I4)
0024 IF (NLOD .EQ. 0) CALL EXIT
0025 IF (NSTP .EQ. 0) NSTP = 3
0026 WRITE(6,6010) NLOD,IOP1,NLIN,ITYP,IFACE,IMOLE,NSTP,ITET
0027 FORMAT (1H1, //, 7X, 5HLOADS, 5X, 3HOPT, 5X, 7HPLASTIC, 5X,
0028 1 4HTYPE, 5X, 4HFACE, 5X, 4HHOLE, 5X, 5HSTEPS, 5X, 5HNETRA, /, 8110)
0029 IF (ITYP .EQ. 1) WRITE (6,6110)
0030 IF (ITYP .EQ. 2) WRITE (6,6120)
0031 IF (ITYP .EQ. 3) WRITE (6,6130)
0032 6110 FORMAT ( //, 5X, 3HNETRA-CORE FLAT PLATE ANALYSIS )
0033 6120 FORMAT ( //, 5X, 3HNETRA-CORE CYLINDER ANALYSIS )
0034 6130 FORMAT ( //, 5X, 3HNETRA-CORE AIRFOIL ANALYSIS )
0035 N = 3
0036 IF (IFACE .EQ. 1) N = 5
0037 JK = N + 2
0038 IF (IOP1 .GT. 0) READ (5,5025) (ISA(I),ISB(I),ISC(I)),I=1,11)
0039 5025 FORMAT ( 3I4 )
0040 IF (IOP1 .GT. 0) WRITE (6,6015) (ISA(I),ISB(I),ISC(I)),I=1,11)
0041 6015 FORMAT ( //, 5X, 20HOPTIMIZATION OPTIONS / 5X, 3HISA,
0042 1 5X, 3HISB, 5X, 3HISC, /, (316,/) )
0043 100 READ (5,5010) MED
0044 5010 FORMAT (12A4)
0045 WRITE(6,5010) MED
0046 READ (5,5030) (E(1,1),E(2,1),E(3,1),E(4,1),I=1,M)
0047 5030 FORMAT ( 4F12,5 )
0048 WRITE(6,6020) (E(1,1),E(2,1),E(3,1),E(4,1),I=1,M)
0049

```

TAPE 3

```

0041 6020 FORMAT ( //, 5X, 2HEX,10X, 2HEV, 10X, 3HGRY, 8X, 4HNUXY,
0042 1 / ( 3F12.0,F12.5) )
0043 READ (5,5040) NX,NY,SIDE,MT, (THK(I),I=1,M)
0044 5040 FORMAT (2I4,4X,8F6.0)
0045 IF (MT .LT. -COL) MT = -0149*SIDE
0046 WRITE(6,6030) NX,NY,SIDE,MT, (THK(I),I=1,M)
0047

```

```

0046      6030 FORMAT ( ///, 8X, 2MLX, 8X, 2HLY, 5X, 4HSIDE, 5X, 6HHEIGHT ,
0047      1 12X, 15HLEG THICKNESSES, /, 2110,8F12.5 )
0048      READ (5,5035) THETA,VLOFF,Y2OFF,THT
0049      IF (THT .LT. MT) THT = MT
0050      WRITE (6,6035) THETA,VLOFF,Y2OFF,THT
0051      6035 FORMAT ( ///, 5X, 5THETA, 7X, 5HLYOFF, 7X, 5H2OFF, 8X,
0052      1 3HMT, /, 4F12.4 )
0053      IF (ITYP .EQ. 3) READ (5,5045) (TC(I),XC(I),I=1,11)
0054      IF (ITYP .EQ. 3) WRITE(6,5045) (TC(I),XC(I),I=1,11)
0055      5045 FORMAT ( 2F12.5 )
0056      IF (NLIN .NE. 1) GO TO 120
0057      DO 110 IK = 1,N
0058      READ (5,5050) STYL(IK),STY2(IK),STC1(IK),STC2(IK),STSS(IK)
0059      READ (5,5050) (SIG1(IK,J),J=1,7)
0060      READ (5,5050) (SIG2(IK,J),J=1,7)
0061      READ (5,5050) (SIGC1(IK,J),J=1,7)
0062      READ (5,5050) (SIGC2(IK,J),J=1,7)
0063      READ (5,5050) (SIGSS(IK,J),J=1,7)
0064      READ (5,5050) (PRXT(IK,J),J=1,7)
0065      110 CONTINUE
0066      120 READ (5,5035) (FXT(I),FYT(I),FXC(I),FYC(I),SSS(I),I=1,N)
0067      5035 FORMAT ( 5F12.3 )
0068      WRITE(6,6070) (FXT(I),FYT(I),FXC(I),FYC(I),SSS(I),I=1,N)
0069      6070 FORMAT ( ///, 20X, 18HALLOWABLE STRESSES /, 9X, 3HFX, 9X, 3HFYT,
0070      1 9X,3HFXC, 9X, 3HFYC, 9X, 3HFX, / (5F12.0) )
0071      IF (11OPT .NE. 1) GO TO 145
0072      READ (5,5035) (RHO(I),I=1,N)
0073      WRITE(6,6075) (RHO(I),I=1,N)
0074      6075 FORMAT ( ///, 5X, 15HLEG DENSITY 0 . 5F12.5 )
0075      145 IF (ITET) 150,150,160
0076      CALL LOAD('TETRA2')
0077      CALL NODGEN (0,X,Y,Z,SLOPA)
0078      CALL PLATN
0079      GO TO 170
0080      160 CALL LOAD('TETRA5')
0081      CALL NODGEN (0,X,Y,Z,SLOPA)
0082      CALL PLATN
0083      170 DO 200 N = 1,NL0D
0084      READ (5,5050) XLOAD,YLOAD,ZLOAD,XMOM,YMOM,TORQ,XQSHR,YQSHR
0085      5050 FORMAT ( 6F12.0 , F8.0 )
0086      200 CONTINUE
0087      WRITE(6,6080) N,XLOAD,YLOAD,ZLOAD,XMOM,YMOM,TORQ,XQSHR,YQSHR
0088      6080 FORMAT ( ///, 20X, 9HLOAD CASE : 14./, 9X,2HMX, 10X, 2HMY, 9X,
0089      1 2HMX, 10X, 2HMY, 10X, 2HMY, 10X, 3HMY, 8X, 6HTORQUE, 8X,
0090      1 2HMX, 10X, 2HMY, /, 9F12.0 )
0091      APPLD(1,N) = XLOAD
0092      APPLD(2,N) = YLOAD
0093      APPLD(3,N) = ZLOAD
0094      APPLD(4,N) = XMOM
0095      APPLD(5,N) = YMOM
0096      APPLD(6,N) = TORQ
0097      APPLD(7,N) = XQSHR
0098      APPLD(8,N) = YQSHR
0099      IF (ITET) 180,180,190
0100      180 CALL LOAD('TETRA4')

```

PAGE 0003

DATE 71321

MODEL 44 PS VERSION 3, LEVEL 4

FORTRAM IV


```

0099 CALL LODGEN (X,Y,Z,SLOPA)
0100 GO TO 200
0101 CALL LOAD('TETRA7')
0102 CALL LODGAN (X,Y,Z,SLOPA)
0103 200 CONTINUE
0104 ICNT = 1
0105 GO TO 250
0106 210 CONTINUE
0107 IF (IOPT .GT. 0) CALL OPTM (X,Y,Z,SLOPA)
0108 IF (MLIN .EQ. 1) CALL MGLIN (ICNT,IOPT,MSTP)
0109 IF (ITET) 220,220,230
0110 CALL LOAD('TETRA2')
0111 CALL NOOGEN (O,X,Y,Z,SLOPA)
0112 CALL LOAD('TETRA3')
0113 CALL PLATGN
0114 CALL LODI('TETRA4')
0115 CALL LODGEN (X,Y,Z,SLOPA)
0116 RETURN
0117 CALL LOAD('TETRA5')
0118 CALL NOOGEN (O,X,Y,Z,SLOPA)
0119 CALL LOAD('TETRA6')
0120 CALL PLATGA
0121 CALL LODI('TETRA7')
0122 CALL LODGAN (X,Y,Z,SLOPA)
0123 250 RETURN
0124 END

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

PAGE 0004

SOURCE	SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK /	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	NUMEL	000000	NUMEL	000004	COMMON BLOCK /	NUMEL	000004	MAP SIZE	NUMEL	000004	NUMEL	000004
NUMBLK	NUMBLK	000014	NUMBLK	000018	COMMON BLOCK /	NUMBLK	000018	MAP SIZE	NUMBLK	000018	NUMBLK	000018
SYMBOL	SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK /	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
E	FXT	000000	FXT	000050	COMMON BLOCK /	FXT	000050	MAP SIZE	FXT	000050	FXT	000050
SSS	CMS	0000A0	CMS	0000B4	COMMON BLOCK /	CMS	0000B4	MAP SIZE	CMS	0000B4	CMS	0000B4
SYMBOL	SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK /	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
XLOAD	YLOAD	000000	YLOAD	000004	COMMON BLOCK /	YLOAD	000004	MAP SIZE	YLOAD	000004	YLOAD	000004
XMON	TORQ	000014	TORQ	000018	COMMON BLOCK /	TORQ	000018	MAP SIZE	TORQ	000018	TORQ	000018
SYMBOL	SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK /	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	HT	000000	HT	000004	COMMON BLOCK /	HT	000004	MAP SIZE	HT	000004	HT	000004
Y2OFF	THT	000024	THT	000028	COMMON BLOCK /	THT	000028	MAP SIZE	THT	000028	THT	000028
MEID	XX	000088	XX	000092	COMMON BLOCK /	XX	000092	MAP SIZE	XX	000092	XX	000092
IFACE	THOLE	0000C8	THOLE	0000CC	COMMON BLOCK /	THOLE	0000CC	MAP SIZE	THOLE	0000CC	THOLE	0000CC
SYMBOL	SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK /	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
NSTP	IPRT	000000	IPRT	000004	COMMON BLOCK /	IPRT	000004	MAP SIZE	IPRT	000004	IPRT	000004
ICYC	LAST	000014	LAST	000018	COMMON BLOCK /	LAST	000018	MAP SIZE	LAST	000018	LAST	000018
XMS	MTMN	000028	MTMN	000032	COMMON BLOCK /	MTMN	000032	MAP SIZE	MTMN	000032	MTMN	000032
ML	CMSA	000064	CMSA	000068	COMMON BLOCK /	CMSA	000068	MAP SIZE	CMSA	000068	CMSA	000068
PMT	PMT	000108	PMT	000112	COMMON BLOCK /	PMT	000112	MAP SIZE	PMT	000112	PMT	000112

SYMBOL SIGT1 PRXT STC2	LOCATION 000000 000384 000528	SYMBOL SIGT2 PRXC STSS	LOCATION 000004 000438 00053C	COMMON BLOCK / MODU / MAP SIZE SYMBOL LOCATION SIGC1 000168 STT1 0004EC	SYMBOL SIGC1 STC1	LOCATION 000200 000514
SYMBOL APPLD	LOCATION 000000	SYMBOL	LOCATION	COMMON BLOCK / LOOB / MAP SIZE SYMBOL LOCATION 000168	SYMBOL	LOCATION
SYMBOL ITEST	LOCATION 000000	SYMBOL	LOCATION	COMMON BLOCK / DUMP / MAP SIZE SYMBOL LOCATION 000004	SYMBOL	LOCATION
SYMBOL I	LOCATION 00020C	SYMBOL N	LOCATION 000210	SCALAR MAP SYMBOL LOCATION JK 000214	SYMBOL J	LOCATION 00021C
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321						
SYMBOL A	LOCATION 000220	SYMBOL Y	LOCATION 000600	ARRAY MAP SYMBOL LOCATION Z 000880	SYMBOL SLOPA	LOCATION 001330
SYMBOL JBCOM MODCAN MONLIN	LOCATION 0014E0 0014F4 001508	SYMBOL EXIT PLATGA	LOCATION 0014E4 0014F8	SUBPROGRAMS CALLED SYMBOL LOCATION LOAD 0014E8 LOOGEN 0014FC	SYMBOL NODGEN LODCAN OPTM	LOCATION 0014EC 001500 001504
SYMBOL 50 6130 5030 5045 6075 170 200	LOCATION 001622 001954 001BA8 001F44 002440 0024C8 002686	SYMBOL L 5020 5025 6020 110 145 5050 210	LOCATION 001760 001A28 001C20 002248 002462 00253C 0026E2	SYMBOL L 6010 6015 5040 120 150 6080 220	SYMBOL L 6120 5010 6035 6070 160 190 250	LOCATION 001926 001810 001E40 002348 0024A0 00269E 00278A
TOTAL MEMORY REQUIREMENTS 00281C BYTES						
COMPILER HIGHEST SEVERITY CODE WAS 0						
//CURV EXEC FORTRAN(BCD,MAP)						
0001 SUBROUTINE CURV (X,Y,A,B,C) 0002 DIMENSION A(11),B(11) 0003 Y = 1000000. 0004 C = 0.0 0005 DO 100 I = 1,11 0006 IF (X - A(I)) 110,100,100 0007 100 CONTINUE 0008 GO TO 120 0009 110 CONTINUE 0010 J = I - 1 0011 C = (B(I)-B(J)) / (A(I)-A(J)) 0012 Y = B(J) + (X-A(J)) * C 0013 C = ATAN2 (B(I)-B(J), (A(I)-A(J))) 0014 120 CONTINUE 0015 RETURN 0016 END						

SYMBOL Y	LOCATION 0000E0	SYMBOL C	LOCATION 0000E4	SCALAR MAP SYMBOL I	LOCATION 0000E8	SYMBOL X	LOCATION 0000EC	SYMBOL J	LOCATION 0000F0
SYMBOL A	LOCATION 0000F4	SYMBOL B	LOCATION 0000F8	ARRAY MAP SYMBOL F	LOCATION 0000FC	SYMBOL	LOCATION	SYMBOL	LOCATION
SYMBOL ATAN2	LOCATION 0000FC	SYMBOL	LOCATION	SUBPROGRAMS CALLED SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
LABEL 100	LOCATION 0001F6	LABEL 110	LOCATION 000212	LABEL MAP LABEL 120	LOCATION 000292	LABEL	LOCATION	LABEL	LOCATION

TOTAL MEMORY REQUIREMENTS 0002E0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
/OPTIM EXEC FORTRAM(BCD,MAP)

```

0001 SUBROUTINE OPTM(X,Y,Z,SLOPA)
0002 COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBY,VEL(20)
0003 COMMON / ONE / SIDE,MT,THK(5),THETA,YLOFF,YZOFF,THT,DUM,TC(11)
0004 1 ,XC(11),MED(12),NX,NY,IOPRT,ITYPE,IFACE,IMOLE,ITEI,ICNT,MNEW(4)
0005 COMMON / MATL / E(4,5),FAT(5),FYT(5),FVC(5),SSS(5),CNS(10)
0006 COMMON / OPTA / NSTP,IOPRT,IOS,IDRV,ITOT,ICYC,LAST,IK, NTL,NMSL
0007 1 ,XMS,MTMAT,MT,MT, MMAX(11),ML(11),CMSA(11)
0008 2 ,SLW(11),SLOPE(11,5),RHOD(5),PMT(11),PMT(11)
0009 3 ,ISA(11),FSB(11),ISC(11)
0010 DIMENSION SLP(11),PMS(11,5),H(12)
0011 DIMENSION X(1300),Y(1300),Z(1300),SLOPA(300)
0012 EQUIVALENCE (H,SIDE)
0013 IOPRT = 1
0014 6000 FORMAT ( 5E20.5 )
0015 6020 FORMAT ( 10F12.5 )
0016 N = 3
0017 IF (IFACE .EQ. 1) N = 5
0018 N = 11
0019 IF (ICNT .EQ. 1 .AND. IOPRT .EQ. 1) WRITE (6,6005)
0020 6005 FORMAT ( 1H1,/,5X,18-optimization steps /// )
0021 ICNT = ICNT + 1
0022 IF (ICNT .GT. 200) IOPRT = 5
0023 CALL WEIGHT (MT,RHO,NUMEL,X,Y,Z,SLOPA)
0024 XMS = 10.
0025 ON 120 I = 1,M
0026 120 IF (XMS .GT. CMS(11)) XMS = CMS(11)
0027 130 CONTINUE
0028 IF (IOPRT - 1) 150,140,150
0029 140 WRITE (6,6100) IK,ICYC,ICNT,ITOT
0030 6100 FORMAT ( /, 5X, 4H1R 0, 14, 5X, 11MSIDESTEPS 0, 14, 5X, 7HSTEPS 0,
0031 1 14, 5X, 8HCYCLES 0, 14 )
0032 WRITE (6,6110) MT,XMS,(H(11),I=1,11)

```

```

0028      6110 FORMAT ( 5X, 4HMT #, F12.5, 5X, 8HMARGIN #, F12.5, 5X
0029      1, 6HMEIC #, F12.5 / 7F12.5)
0030      WRITE (6,6120) (CMS(K),K=1,N)
0031      6120 FORMAT ( 5X, 11HMARGIN#, #, 5F12.5 )
0032      150 IF (IDS - 1) 160,550,160
0033      160 IF (IORT - 5) 170,590,170
0034      170 IF (IDRV - 1) 180,340,180
0035      C OPTIMIZATION
0036      180 ICYC = ICYC + 1
0037      IDRV = 0
0038      IF (ICYC - 10) 190,190,510
0039      190 IF (XMS-LT. XMSL-AND. ICYC -EQ.2) GO TO 320
0040      IF (XMS-LT. XMSL-AND. ICYC -GT. 2) GO TO 510
0041      200 IF (ICYC -GT. 1) XMSL = XMS
0042      DO 210 K = 1,N
0043      210 H(K) = H(K)
0044      IF (ICYC - 1) 260,220,260

```

PAGE 0002

DATE 71321

LEVEL 4

VERSION 3

PROGRAM IV

```

0043      C CALCULATE DIRECTION COSINES OF CONSTANT WEIGHT SURFACE
0044      220 CALL DAR (SLM,PMT,N,1,ISA)
0045      CALL DAR (SLOPE,PMS,N,M,ISA)
0046      DO 230 J = 1,N
0047      SLP(J) = 0.
0048      DO 230 K = 1,M
0049      IF (CMS(K) -GT. -9) CMS(K) = -9
0050      SLP(J) = SLP(J) + PMS(J,K)*1. - CMS(K)*.015
0051      C CALCULATE DIRECTION COSINES OF RESULTANT CONSTRAINT SURFACE
0052      CALL DAR(SLP,PMT,N,1,ISA)
0053      IF (IPRT - 1) 260,240,260
0054      240 WRITE (6,6030)
0055      6030 FORMAT ( //, 14H SIDE STEPPING : /, 5X, 39H DIRECTION COSINES OF C
0056      1ONST. WT. SURFACE : /, 5X, 21H, CONSTRAINT SURFACES : /, 5X,
0057      2 19H, MARGINS OF SAFETY : /, 5X, 46H, DIR. COSINES OF COMPOSITE CO
0058      NSTRAINT SURFACE )
0059      WRITE (6,6020) (PMT(K),K=1,N)
0060      DO 250 K = 1,N
0061      250 WRITE (6,6020) (PMS(K),J=1,M)
0062      WRITE (6,6020) (CMS(K),K=1,N)
0063      WRITE (6,6020) (PMT(K),K=1,N)
0064      DO 270 K = 1,N
0065      270 H(K) = H(K)*1. + PMT(K)*TMLT)
0066      280 WT = 0.0
0067      CALL WEIGHT (WT,RHO,MUMEL,X,Y,Z,SLOPA)
0068      DWT = WT / WTL - 1.
0069      IF (ABS(DWT) - .01) 640,300,300
0070      300 DO 310 K = 1,N
0071      IF (PMT(K)*DWT -GT. .8) DWT = -.8/PMT(K)
0072      310 H(K) = H(K)*1. - PMT(K)*DWT)
0073      GO TO 280
0074      C IF FIRST SIDESTEP DIDNT WORK REDUCE STEP LENGTH
0075      320 DO 330 K = 1,N
0076      330 H(K) = H(K)
0077      TMLT = TMLT/4.
0078      XMS = XMSL
0079      GO TO 200
0080      C CALCULATE DERIVATIVES
0081      340 IF (IK) 380,350,380
0082      350 IF (IPRT -EQ. 1) WRITE (6,6040)
0083      6040 FORMAT ( 31H CALCULATION OF DERIVATIVES )

```

```

0077      ITOT = ITOT + 1
0078      XMSL = XMS
0079      WTL = WT
0080      DO 360 K = 1,M
0081      CMSA(K) = CMS(K)
0082      TMLT = TMLT/2.
0083      DO 370 IJ = 1,M
0084      370 HL(IJ) = H(IJ)

FORTRAN IV      MODEL 44 PS      VERSION 3, LEVEL 4      DATE 71321      PAGE 0003

0085      380 CONTINUE
0086      IF (IK) 390,410,390
0087      390 SLW(IK) = (WT - WTL) / (H(IK) - HL(IK))
0088      DO 400 K = 1,M
0089      SLOPE(IK,K) = (CMS(K) - CMSA(K)) / (H(IK) - HL(IK))
0090      IF (IPRT .EQ. 1) WRITE(6,6050)(SLOPE(IK,K),K=1,M),SLW(IK)
0091      6050 FORMAT (7F15.5)
0092      410 CONTINUE
0093      IK = IK + 1
0094      IF (ISAT(IK)-12) 415,410,415
0095      415 DO 420 IJ = 1,M
0096      420 H(IJ) = HL(IJ)
0097      IF (IK - N) 430,430,180
0098      430 H(IK) = H(IK) + .001
0099      IF (ISAT(IK)) 440,450,440
0100      440 IA = ISAT(IK)
0101      H(IA) = H(IK)
0102      IF (ISB(IK)) 460,470,460
0103      460 IB = ISB(IK)
0104      H(IB) = H(IK)
0105      470 IF (ISCI(IK)) 480,490,480
0106      480 IC = ISCI(IK)
0107      H(IC) = H(IK)
0108      490 IF (IK - N) 500,500,180
0109      500 ICYC = 0
0110      GO TO 640

C      RETURN TO PREVIOUS VALUE BEFORE DESCENDING
0111      510 DO 520 K = 1,M
0112      520 H(K) = HL(K)
0113      IF (IPRT .EQ. 1) WRITE (6,6060)
0114      6060 FORMAT (11H DESCENDING )
0115      IF (ICYC .GT. 9) TMLT = 4.*TMLT
0116      IDS = 1
0117      GO TO 640

C      STORE NEW OPTIMUM
0118      530 WTM = WT
0119      IF (IPRT .EQ. 1) WRITE (6,6070)
0120      6070 FORMAT (17H NEW OPTIMUM )
0121      DO 540 K = 1, N
0122      IF (IPRT .EQ. 1) WRITE (6,6020) H(K)
0123      540 HMAX(K) = H(K)
0124      GO TO 160

C      DESCENDING
0125      550 IF (ABS(XMS) .LT. .050) IDS = 0
0126      IF (IDS) 570,560,570
0127      560 IK = 0
0128      IDRV = 1
0129      IF (ITOT .GT. NSTP) IDPT = 5
0130      IF (ICMT .GT. 100) IDPT = 5

```

```

0131 IF (WT - WMN) 530,160,160
0132 CALCULATE DIRECTION COSINES OF CONSTANT WEIGHT SURFACE
0133 C 570 CALL DAR (SLM,PMT,N,1,ISA)
0134 DO 580 K = 1,N
0135 IF (ITOT -GT- 0) GO TO 575
0136 IF (K -LT- 3) GO TO 580
0137 IF (K -GT- 7) GO TO 580
0138 IF (PMT(K)XMS -GT- .8) XMS = .8/PMT(K)
0139 H(K) = H(K)X(1.-PMT(K)XMS)
0140 580 CONTINUE
0141 GO TO 640
0142 590 CONTINUE
0143 LAST = 1
0144 IF ON FINAL CYCLE INSERT OPTIMUM VALUES
0145 WRITE (6,6080)
0146 6080 FORMAT(1H1,/,31H OPTIMIZED LAYER THICKNESSES , / )
0147 DO 600 K = 1,N
0148 600 WRITE (6,6090)(K,HMAX(K))
0149 6090 FORMAT (15,F15.5)
0150 610 DO 620 K = 1,N
0151 620 H(K) = HMAX(K)
0152 640 CONTINUE
0153 RETURN
0154 EMO

```

SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / COMN		MAP SIZE		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				LOCATION	SYMBOL	LOCATION	SYMBOL						
NUMAP	000000	NUMEL	000014	000004	NLOD	000008	NLTM	00000C	00000C	MBAND	000010		
				000019	NEL	00001C							
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / ONE		MAP SIZE		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				LOCATION	SYMBOL	LOCATION	SYMBOL						
STOE	000000	H	000000	000000	HT	000004	THR	000008	00000C	THETA	00001C		
YIDFF	000020	Y2OFF	000024	000024	THI	000028	DUM	000030	000030	TC	000030		
XC	00005C	HEU	000088	000088	NX	000088	NY	00008C	00008C	IOPT	0000C0		
ITYP	0000C4	IFACE	0000C8	0000C8	IHOLE	0000CC	ITET	0000D0	0000D0	ICNT	0000D4		
MNEW	0000D8												
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / MATL		MAP SIZE		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				LOCATION	SYMBOL	LOCATION	SYMBOL						
F	000000	FAT	000050	000050	FVT	000064	FAC	0000D0	0000D0	FIC	00008C		
SS	000040	CMS	0000B4	0000B4									
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / OPTA		MAP SIZE		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				LOCATION	SYMBOL	LOCATION	SYMBOL						
NSPT	000000	IPMT	000004	000004	IDS	000008	IDRV	00000C	00000C	ITOT	000010		
ISYC	000014	EAST	000018	000018	IK	00001C	WIL	000020	000020	AMSI	000024		
AMS	000028	WTMN	00002C	00002C	TMLT	000030	WT	000034	000034	HPAC	000038		
HL	000064	CMFA	000090	000090	SLM	00009C	SLOPE	0000A8	0000A8	RHO	0000C4		
PMT	0000D8	FMT	0000D4	0000D4	ISA	0000D0	ISB	0000D5	0000D5	ISL	0000D8		
SYMBOL	LOCATION	SYMBOL	LOCATION	SCALAR MAP				SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				LOCATION	SYMBOL	LOCATION	SYMBOL						
M	00025C	N	000260	000260	I	000264	K	000268	000268	J	00026C		
DNT	000270	IJ	000274	000274	IA	000278	IB	00027C	00027C	IC	000280		

ARRAY MAP												
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
SLP	000284			X	00038C			Y	000390	Z	000394	
SLOPA	000398											
SUBPROGRAMS CALLED												
SYMBOL	LOCATION	SYMBOL	LOCATION	DAR	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ISCOM#	00039C					0003A0		FRAP10	0003AB			
LABEL MAP												
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL
6000	000406	6020	0004E4	6005	030574	120	00060A	130	000646	150	0007E2	200
140	00065A	6100	00069C	6110	000740	6120	0007C4	200	0008C2	6030	000A38	300
160	0007FA	170	000812	180	00082A	190	000856	240	000A1E	300	000CE2	
210	0008EC	220	00092E	230	000990	240	000990	280	000C82			
250	000B3C	260	000C38	270	000C44							
FORTRAN IV MODEL 44 PS												
VERSION 3, LEVEL 4 DATE 71321												
310	00001C	320	000762	330	000D6E	340	000D8A	350	000DCA	360	000DF8	370
6040	000DF8	360	000F4E	370	000E90	380	000EC2	390	000ED2	400	000F10	410
400	000F10	6050	000F4E	410	000FF2	415	00101E	420	00102A	430	00105C	440
430	00105C	440	0010AB	450	00100C	460	0010F4	470	001130	480	001148	490
480	001148	490	00117C	500	001190	510	0011A6	520	0011B2	530	001256	540
6060	001208	530	001256	6070	001288	540	0012F4	550	00132C	560	00135C	570
560	00135C	570	0013C6	575	00140E	580	00146A	590	001486	600	00149C	610
6080	00149C	600	0014F6	6090	00153E	610	00154C	620	001554			

PAGE 0006

DATE 71321

VERSION 3, LEVEL 4

MODEL 44 PS

FORTRAN IV

TOTAL MEMORY REQUIREMENTS 001614 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//WEIGHT EXEC FORTRAN(BCD,MAP)

```

0001 SUBROUTINE WEIGHT (WT,RHO,MUMEL,X,Y,Z,SLOPA)
0002 DIMENSION X(300),Y(300),Z(300),SLOPA(300)
0003 COMMON / ONE / SIDE,MT,THK(SI),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
0004 1 ,XC(11),HED(12),NR,NY,IOPT,ITYP,IFACE,THOLE,ITET,ICNT,NMEM(4)
0005 COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECS,NRECLD,NR1,NR2
0006 DIMENSION AL(3),AL2(3),LM(6),RH(15)
0007 IF (ITET) 50,50,60
0008 50 CALL NODGEN (1,X,Y,Z,SLOPA)
0009 GO TO 70
0010 60 CALL NODGAN (1,X,Y,Z,SLOPA)
0011 DO 200 I = 1,MUMEL
0012 DO 200 J = 1,MUMEL
0013 NR2=1
0014 READ(2,NR2) IDP,IP,IQ,IR,IS,ETR,ELGT,EPH,G,T,INTL
0015 IJK = 1
0016 IF (IS -GT. 0) IJK = 2
0017 DO 200 K = 1,IJK
0018 IF (K.EQ. 1) GO TO 71
0019 IP = IR
0020 IR = IS
0021 71 XQP = X(IQ)-X(IP)
0022 YQP = Y(IQ)-Y(IP)
0023 ZQP = Z(IQ)-Z(IP)
0024 XRP = X(IR)-X(IP)
0025 YRP = Y(IR)-Y(IP)
0026 ZRP = Z(IR)-Z(IP)
0027 XRQ = X(IR)-X(IQ)
0028 YRQ = Y(IR)-Y(IQ)
0029 ZRQ = Z(IR)-Z(IQ)

```

TAPE

```

0029      D1= SQRT(XQP*YQP+YQP*YQP+ZQP*ZQP)
0030      AL(1)= XQP/D1
0031      AL(2)= YQP/D1
0032      AL(3)= ZQP/D1
0033      I02 RR= AL(1)*XRP +AL(2)*YRP +AL(3)*ZRP
0034      X2 = XRP -AL(1)*RR
0035      Y2 = YRP -AL(2)*RR
0036      Z2 = ZRP -AL(3)*RR
0037      D2 =SQRT(X2*X2 +Y2*Y2 +Z2*Z2)
0038      AREA = D1*D2/2.
0039      WT = WT + AREA*RHQ(INTL1*THK(INTL))
0040      CONTINUE
0041      YILGT = -28867 + YIOFF
0042      YZLGT = -86657 + Y2OFF
0043      XLGT = (MX-1)*YZLGT + YILGT 10SIDE
0044      YLGT = (FLOAT(MY))-510SIDE
0045      WT = WT/(YLGT*XLGT)
0046      WRITE (6,6100) WT
0047      6100 FORMAT (6F 15.5)
0048      RETURN

```

FORTRAN IV MODEL 44 PS VERSION 3. LEVEL 4 DATE 71321 PAGE 0002

0049 END

FORTRAN IV MODEL 44 PS VERSION 3. LEVEL 4 DATE 71321 PAGE 0003

SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / ONE	SYMBOL	LOCATION	MAP SIZE	0000E8	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	WT	000004	THK	000008	THETA	00001C	000020	YIOFF	000030	XC	00005C
YZOFF	000024	THY	000028	DUM	00002C	TC	000030	00005C	ITYP	0000C4	NHEW	0000D8
MED	000088	MX	00008C	MY	000090	ICNT	0000D4	0000D8				
IFACE	0000C8	IMOLE	0000CC	ITET	0000D0							
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / RAX	SYMBOL	LOCATION	MAP SIZE <td>000024</td> <td>SYMBOL</td> <td>LOCATION</td> <td>SYMBOL</td> <td>LOCATION</td>	000024	SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000000	NR4LOD	000004	NR4STF	000008	NR4DIA	00001C	000020	NREC	000010		
NRECST	000014	NRECLO	000018	NR1	00001C							
SYMBOL	LOCATION	SYMBOL	LOCATION	SCALAR MAP	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
WT	0000FC	I	000100	MJUEL	000104	TOP	000108	00010C	IP	00010C		
IQ	000110	IR	000114	IS	000118	ETR	00011C	000120	ELGT	000120		
EPHM	000124	C	000128	T	00012C	IMTL	000130	000134	1JK	000134		
K	000138	XQP	00013C	YQP	000140	ZQP	000144	000148	XRP	000148		
YRP	00014C	ZRP	000150	XRP	000154	YRP	000158	00015C	ZRP	00015C		
D1	000160	RR	000164	X2	000168	Y2	00016C	000170	Z2	000170		
D2	000174	AREA	000178	YLGT	00017C	YLGT	000180	000184	KLGT	000184		
YLGT	000188											
SYMBOL	LOCATION	SYMBOL	LOCATION	ARRAY MAP	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	00018C	Y	000190	Z	000194	SLOPA	000198	00019C	AL	00019C		
AL2	0001A8	LM	0001B4	RHO	0001CC							
SYMBOL	LOCATION	SYMBOL	LOCATION	SUBPROGRAMS CALLED	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NOODGM	0001D0	NOODGM	0001D4	ISCDMG	0001D8	ISCDMG	0001DC	0001DC				

LABEL	LOCATION	LABEL	LOCATION	LABEL MAP	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
50	0002E2	60	000316	70	000344	71	000424	100	000506		
102	00052A	200	0005E6	6100	00064C						

TOTAL MEMORY REQUIREMENTS 000710 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//DAR EXEC FORTRAN(BCD,MAP)

```

0001 C SUBROUTINE OARIA,B,M,M,ISA)
0002 CALCULATE DIRECTION COSINES
0003 DIMENSION A(11,5),B(11,5),ISA(11)
0004 DO 30 J = 1,M
0005 SUM = 0.
0006 DO 10 K = 1,M
0007 IF (ISA(K)-12) 5,10,5
0008 5 IF (A(K,J) -LT. 1.0 E-10) A(K,J) = 1.0 E-10
0009 SUM = SUM + A(K,J) *A(K,J)
0010 10 CONTINUE
0011 SUM = SUM*.5
0012 DO 20 K = 1,M
0013 B(K,J) = 0.0
0014 IF (ISA(K)-12) 15,20,15
0015 15 B(K,J) = A(K,J)/SUM
0016 20 CONTINUE
0017 30 RETURN
0018 END

```

PAGE 0002

FORTTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

SCALAR MAP				ARRAY MAP				SUBPROGRAMS CALLED			
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
J	0000F0	M	0000F4	SUM	0000F8	K	00010C	M	000100		
A	000104	B	000108	ISA	00010C						
PRPR0	000110										
5	000206	10	000268	15	0002DA	20	000308	30	00031E		

TOTAL MEMORY REQUIREMENTS 000378 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//NONLIN EXEC FORTRAN(BCD,MAP)

PAGE 0001

FORTTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0001 SUBROUTINE NONLIN (ICNT,IOPT,NSTP)
0002 COMMON / LODA / XLOAD,YLOAD,XVLOAD,XVMOV,XVMOV,TORQ,XQSHR,YQSHR
0003 ICNT = ICNT + 1
0004 IF (ICNT .GT. NSTP) IOPT = 5

```

0005

N = ICNT

FACT = FLOAT(ICNT)/FLOAT(ICNT-1)

XLOAD = XLOAD+FACT

YLOAD = YLOAD+FACT

XYLOD = XYLOD+FACT

XMON = XMON+FACT

YMON = YMON+FACT

XYMON = XYMON+FACT

TORO = TORO+FACT

XQSHR = XQSHR+FACT

YQSHR = YQSHR+FACT

WRITE(6,6080) N,XLOAD,YLOAD,XMON,YMON,TORO,XQSHR,YQSHR

6080 FORMAT (141,/,20X, 'PHLOAD INC. : 14,/, 9X,2HNY, 10X, 24V, 9X,

1 3HNY, 10X, 2HNY, 10X, 2HNY, 10X, 3HNY, 6X, 6HTORQUE, 8X,

1 2HNY, 10X, 2HNY, /, 9F12.0)

RETURN

END

FORTAN IV MODEL 44 PS

VERSION 3, LEVEL 4 DATE 71321

PAGE 0002

SYMBOL

LOCATION

SYMBOL

COMMON BLOCK / LODA / MAP SIZE

000024

LOCATION

SYMBOL

LOCATION

SYMBOL

LOCATION

SYMBOL

SCALAR MAP

000008

LOCATION

SYMBOL

LOCATION

SYMBOL

LOCATION

SYMBOL

SUBPROGRAMS CALLED

000008

LOCATION

SYMBOL

LOCATION

LABEL

LOCATION

LABEL

MAP

000024

LOCATION

LABEL

LOCATION

TOTAL MEMORY REQUIREMENTS 000380 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0

//NOGEN EXEC FORTAN(BCD,MAP)

FORTAN IV MODEL 44 PS

VERSION 3, LEVEL 4 DATE 71321

PAGE 0001

C*****OVERLAY AA*****C

SUBROUTINE NOGEN (IHT,X,Y,Z,SLOPA)

COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,VEL(20)

COMMON / ONE / SIDE,HT,THK(5),THETA,VIOFF,Y2OFF,IHT,DUM,TC(11)

1,XC(11),HED(12),NX,NY,IOP,ITP,IFACE,IMOLE,ITET,ICNT,NYEN(4)

DIMENSION X(300),Y(300),Z(300),SLOPA(300)

C

C

GENERATE FLAT PLATE MODAL COORDINATES

XC(11) = XC(11) + .00001

NUMNP = 2*NX*NY

NYA = 2*NX

Y1LGT = .28867 + Y1OFF

Y2LGT = .06657 + Y2OFF

DO 300 I = 1,NUMNP

IROW = (I-1)/NX + 1

ICOL = I - I/NYA*NYA

```

0013 IF (ICOL) 80,70,80
0014 70 ICOL = NX
0015 IF (FLOAT(1/NX)/2.-FLOAT(1/(NX*2))) .LT. .01) ICOL = 2*NX
0016 80 CONTINUE
0017 IF (ICOL - NX/2) 100,100,85
0018 85 IF (ICOL - NX) 150,150,90
0019 90 IF (ICOL - 3*NX/2) 200,200,95
0020 95 Y(1) = SIDE*(Y2LGT + Y1LGT*2.-Y2LGT * ((ICOL-3*NX/2-1)))
0021 Z(1) = HT
0022 GO TO 250
0023 100 Y(1) = Y2LGT * SIDE*(2+ICOL-1)
0024 Z(1) = 0.0
0025 GO TO 250
0026 150 Y(1) = SIDE*(Y1LGT + Y2LGT*(ICOL-1-NX/2)*2.)
0027 Z(1) = HT
0028 GO TO 250
0029 200 Y(1) = Y2LGT*SIDE*(ICOL-1-NX)*2.
0030 Z(1) = 0.0
0031 250 X(1) = SIDE/2.*(IROW-1) + Y(1)*SIN(THETA/57.2958)
0032 300 CONTINUE
0033 IF (IOPT .NE. 1 .AND. ITPP .EQ. 1) WRITE (6,6140) Y(NUMPP)
0034 6140 FORMAT (///, 5X, 18HFLAT PANEL WIDTH 0 , F15.5 )
0035 IF (IOPT .NE. 1) WRITE (6,6150) X(NUMPP-NX*1)
0036 6150 FORMAT (///, 14X, 8HLENGTH 0 , F15.5 )
0037 IF (ITPP - 2) 800,350,550
C
C GENERATE CYLINDER COORDINATES
0038 350 CIRC = Y2LGT*SIDE*NX
0039 RAD = Y2LGT*SIDE*NX/6.28318
0040 IF (IOPT .NE. 1) WRITE (6,6200) RAD
0041 6200 FORMAT (///, 10X, 17HCYLINDER RADIUS 0 , F15.5 )
0042 6000 FORMAT (6F15.5 )
0043 DO 400 I = 1,NUMPP
C
C GENERATE AIRFOIL COORDINATES
0050 550 CONTINUE
0051 CHORD = Y2LGT*SIDE*NX/2.
0052 IF (ILOPT .NE. 1) WRITE (6,6250) CHORD
0053 6250 FORMAT (///, 10X, 15HAIRFOIL CHORD 0 , F15.5 )
C
DO 640 I = 1,NUMPP
0054 FACT = 1.0
0055 IF (Y(1) - CHORD) 610,610,605
0056 605 Y(1) = 2.*CHORD - Y(1)
0057 FACT = -1.0
0058 610 XOC = Y(1)/CHORD
0059 CALL CURV (XOC,TOC,IC,IC,SLOP)
0060 SLOPA(1) = SLOP*FACT
0061 ZOLD = Z(1)
0062 Z(1) = TOC*CHORD*FACT
0063 IF (ZOLD - .001) 640,620,620
0064 620 Z(1) = Z(1) - HT*FACT
0065

```

PAGE 0002

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0066 IF (FACT) 625,622,622
0067 622 IF (Z(1)) 625,640,640
0068 624 IF (Z(1)) 640,640,625
0069 625 Z(1) = 0.0
0070 IF (Y(1)) - CHORD/2.) 630,630,635
0071 630 Y(1) = ABS(HT/SLOPA(1))
0072 GO TO 640
0073 635 Y(1) = CHORD - ABS(HT/SLOPA(1))
0074 640 CONTINUE
0075 800 CONTINUE
0076 IF (IMRT -EQ. 1) RETURN
0077 500 CONTINUE
0078 WRITE (3) (1,X(1),Y(1),Z(1),I=1,MUMNP)
0079 RETURN
0080 END

```

TAPE 3

PAGE 0003

DATE 71321

LEVEL 5

MODEL 44 PS

FORTRAN IV

SYMBOL MUMNP NUMBER	LOCATION	SYMBOL NUMBER	COMMON BLOCK /		LOCATION	/ MAP SIZE		SYMBOL MLIN	LOCATION	SYMBOL MBAND	LOCATION
			SYMBOL	LOCATION		SYMBOL	LOCATION				
	000000		MLIN	000004		MLIN	000008		00000C		000010
	000014		NEL	000018		NEL	00001C				
COMMON BLOCK / ONE / MAP SIZE 000008											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THX	000008	THETA	00001C	YLOFF	000020	YLOFF	000020
YZOFF	000024	THY	000028	DUM	00002C	TC	000030	XC	00003C	XC	00003C
MED	000038	NX	000038	MY	00003C	ICPT	000040	ITYP	00004C	ITYP	00004C
IFACE	00004C	THOLE	00004C	ITET	000050	ICNT	000054	MNEW	000058	MNEW	000058
SCALAR MAP											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
MYA	000168	YILSY	00016C	YZLGY	000170	I	000174	IRON	000178	IRON	000178
ICOL	00017C	CIRC	000180	RAD	000184	ABC	000188	CHORD	00018C	CHORD	00018C
FACT	000190	XDC	000194	TDC	000198	SLOP	00019C	ZOLD	0001A0	ZOLD	0001A0
IMRT	0001A4										
ARRAY MAP											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	0001A8	Y	0001AC	Z	0001B0	SLOPA	0001B4				
SUBPROGRAMS CALLED											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ISCON9	0001B8	CURV	0001BC	SIN	0001C0	COS	0001C4				
LABEL MAP											
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
70	000328	80	000344	85	0003C6	90	0003DA	95	000402	95	000402
100	00047C	150	0004D8	200	000546	250	0005A0	300	00060C	300	00060C
6140	00058C	6150	000704	350	00073C	6200	0007C8	6000	0007EE	6000	0007EE
400	000862	450	000896	550	00089C	6250	000900	605	00095A	605	00095A
610	000986	620	0009F0	622	000A26	624	000A4C	625	000A6C	625	000A6C
630	000AA2	635	000AD2	640	000B02	800	000B1C	500	000B36	500	000B36

TOTAL MEMORY REQUIREMENTS 000C1C BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//PLATON EXEC FORTRAN(BCO.NAP)

***** SUBROUTINE PLATAN ***** OVERLAY AB *****

```

0001 COMMON / CONN / NUMNP, NUMEL, NLOD, NLIN, MBAND, NUMBLK, MBN, VEL (20)
0002 COMMON / ONE / SIDE, HT, THK(S), THETA, YIOFF, YZOFF, THT, OUM, TC(11)
0003 1, XC(11), HED(12), NK, NY, IOPT, ITYP, IFACE, INHOLE, ITET, ICNT, NMENI(4)
0004 COMMON / PLAT / E(4,5), FXT(S), FXT(S), FXC(S), FYC(S), SSC(S), CHS(10)
0005 COMMON / RAX / NR4, NR6LDD, NR4STF, NR4DIA, NR4REC, NR4ECST, NR4ECLD, NR1, NR2
0006 IF (ICNT .GT. 0) GO TO 1000
0007

```

C
C GENERATE VERTICAL PLATES

```

0008 NAODD = NX/2 + 1
0009 NBODD = 5*NX/2 + 1
0010 NCDD = NX + 1
0011 NAEVN = 1
0012 NBEVN = 2*NX + 1
0013 NCEVN = 3*NX/2 + 1
0014 NP = 0
0015 NK = 0
0016 DO 300 NV = 1, NX
0017 NP = NP + 1
0018 NK = NK + 1
0019 IF (NK - 1) 90, 100, 90
0020 90 MAINP = NAEVN + NV/2 - 1
0021 NBINP = NBEVN + NV/2 - 1
0022 NCINP = NCEVN + NV/2 - 1
0023 NK = 0
0024 GO TO 150
0025 100 MAINP = NAODD + NV/2
0026 NBINP = NBODD + NV/2
0027 NCINP = NCDD + NV/2
0028 150 CONTINUE
0029 NPIR = 2*(NY-1)
0030 DO 300 I = 2, MPIR
0031 NP = NP + 1
0032 NA(NP) = NCINP-1
0033 NCINP = NBINP-1
0034 300 NBINP = MAINP + 2*NX

```

C
C GENERATE SKEW A PLATES

```

0035 NNEW(1) = NP + 1
0036 NAA = 1
0037 NAB = NX + 1
0038 NCA = NX/2 + 1
0039 NAB = 2*NX
0040 NBB = 3*NX
0041 NCB = 5*NX/2
0042 NP = NP + 1
0043 NA(NP) = NAA
0044 NBINP = NAB
0045 NCINP = NCA
0046 NOSA = NX/2 + NY - 1
0047 NPIR = 5
0048 DO 500 I = 2, NOSA
0049 NP = NP + 1
0050 IF (I - NX/2) 310, 310, 350
0051 310 MAINP = NAA + I - 1
0052 NBINP = NAB + I - 1
0053 NCINP = NCA + I - 1
0054 NK = 3

```

```

0055      GO TO 400
0056      350 NA(NP) = MAB + 2*NX(I-1-NX/2-1)
0057      NB(NP) = MBB + 2*NX(I-1-NX/2-1)
0058      NC(NP) = MCB + 2*NX(I-1-NX/2-1)
0059      NK = 2
0060      6000 FORMAT ( 515 )
0061      400 DO 450 J = 2,MP1R
0062      NP = NP + 1
0063      NA(NP) = NC(NP-1)
0064      NC(NP) = NB(NP-1)
0065      NK = NK + 1
0066      NB(NP) = NC(NP) + NX/2
0067      IF (NK.EQ.4) NB(NP) = NB(NP) - 1
0068      IF (11.LT. NX/2) MP1R = MP1R + 4
0069      IF (11.EQ. NX/2) MP1R = MP1R + 4
0070      IF (11.GE. NX/2) AND. (.GE. NY) .DR. (11.GE. NY)) MP1R = MP1R - 4
0071      IF (11.EQ. NY) MP1R = MP1R + 1
0072      500 IF (11.EQ. NX/2) MP1R = MP1R + 1
C
C      ZIP UP SEAM OF CYLINDER
C      IF (11.NE. 2) 540,505,505
0073      505 DO 530 I = 1,NY
0074      NP = NP + 1
0075      NA(NP) = MCA + (1-1)*NX/2
0076      NC(NP) = MCB + (1-1)*NX/2
0077      NB(NP) = NC(NP) + NX - 1
0078      IF (11.NE. NY) 510,530,510
0079      510 NP = NP + 1
0080      NA(NP) = NC(NP-1)
0081      NC(NP) = NB(NP-1)
0082      NB(NP) = NC(NP) + NX/2
0083      530 CONTINUE
C
C      GENERATE SEEN 0 PLATES
0085      540 MNCN(12) = NP + 1
0086      MAA = NK
0087      MCB = 2*NK
0088      MCA = NX/2
C
C      MAB = MAA + 1
0089      MBS = MSA + 1
0090      MCB = 5*NX/2 + 1
0091      NP = NP + 1
0092      NA(NP) = MAA
0093      NB(NP) = MSA
0094      NC(NP) = MCB
0095      MD5B = NX/2 + NY - 1
0096      MP1R = 5
0097      DO 700 I = 2,MD5B
0098      NP = NP + 1
0099      IF (11.NE. NX/2) 545,545,550
0100      545 NA(NP) = MAA - 1 + 1
0101      NB(NP) = MSA - 1 + 1
0102      NC(NP) = MCB - 1 + 1
0103      NK = 1
0104      GO TO 600
0105      550 NA(NP) = MAB + 2*NX(I-1-NX/2-1)
0106      NB(NP) = MBB + 2*NX(I-1-NX/2-1)
0107      NC(NP) = MCB + 2*NX(I-1-NX/2-1)
0108      NK = 3
0109

```

FORTRAN IV MODEL 44 PS VERSION 3. LEVEL 4 DATE 7/3/21 PAGE 0003


```

0163 IF (ITYP - 2) 800,780,780
0164 780 NP = NP + 1
0165 NA(NP) = NX/2
0166 NB(NP) = 5*NX/2
0167 NC(NP) = NX + 1
0168 DO 790 I = 2,NX
0169 NP = NP + 1
0170 NA(NP) = NC(NP-1)
0171 NC(NP) = NB(NP-1)
0172 790 NB(NP) = NA(NP) + 2*NX

C
0173 GENERATE UPPER FACE SHEET
0174 800 NHEM(4) = NP + 1
0175 NA(000) = NX/2 + 1
0176 NB(000) = 5*NX/2 + 1
0177 MC(000) = 3*NX/2 + 1
0178 NK = 1
      K = 0

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0005

```

0179 DO 810 I = 1,NY1
0180 K = K + 1
0181 IF (K.EQ. 3) K = 1
0182 NP = NP + 1
0183 NA(NP) = NA(000) + (I-1)/2
0184 NB(NP) = NB(000) + (I-1)/2
0185 NC(NP) = MC(000) + (I-1)/2
0186 IF (K-2) 805,802,802
0187 802 NA(NP) = NA(NP) + 1
0188 NB(NP) = NB(NP) + 1
0189 805 NMPT = 2*(NY-1)
0190 DO 810 J = 2,NMPT
0191 NP = NP + 1
0192 NA(NP) = NC(NP-1)
0193 NC(NP) = NB(NP-1)
0194 810 NB(NP) = NA(NP) + 2*NX

```

```

C
0195 ZIP UP SIDE OF CYLINDER
0196 IF (ITYP - 2) 850,820,820
0197 820 NP = NP + 1
0198 NA(NP) = NX/2 + 1
0199 NB(NP) = 5*NX/2 + 1
0200 NC(NP) = 2*NX
0201 DO 830 I = 2,NX
0202 NP = NP + 1
0203 NA(NP) = NC(NP-1)
0204 NC(NP) = NB(NP-1)
0205 830 NB(NP) = NA(NP) + 2*NX
0206 850 NMUEL = NP
0207 WRITE (6,200) NAEVN,NHEM
      FORMAT (1//, 5X, 22HFIRST VERTICAL PLATE 0,14, 5X,8-SKEW A 0,
1 14, 5X, 8SKEW 0 0, 14, 5X,10HWR FACE 0,14, 5X,12HUPPER FACE 0,
2 14, )
      J = 1
0208 NM0H = NMUEL/2 - NX + 1 + NX/4
0209 IF (NM0H.GT. 1) NM0H = NM0H
0210 DO 900 I = 1,NMUEL
0211 MD(I) = 0
0212 IF (I.EQ. NHEM(J)) J = J + 1
0213 XNU = 1. - E(4, J)*E(12,J)/E(11,J) *E(14,J)
0214 E11 = E(1,J)/XNU
0215

```



```

0216      E22 = E12,J)/XNU
0217      E12 = E12,J)*E(4,J)/XNU
0218      E33 = E13,J)
0219      IF (IHOLE-1) 890,860,860
0220      860 IF (MA(1)-NODM) 865,885,865
0221      865 IF (MB(1)-NODM) 870,885,870
0222      870 IF (MC(1)-NODM) 890,885,890
0223      885 E11 = 0.0

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0006

```

0224      E22 = 0.0
0225      E12 = 0.0
0226      E33 = 0.0
0227      WRITE (6,6150) NODM
0228      6150 FORMAT (///, 5X, 25HPLATES CONNECTING TO NODE, 19, 10H HAVE BEEN
1 40H GIVEN ZERO STIFFNESS TO SIMULATE A HOLE )
0229      IHOLE = NODM
0230      THICK = THK(1,J)
0231      900 WRITE (3) 1,MA(1),MB(1),MC(1),E22,E11,E12,E33,THICK,J
0232      GO TO 1020
0233      1000 CONTINUE
0234      K = 1
0235      DO 1010 J = 1,NUMEL
NR2=J
0236      READ(2,NR2)1,MA(1),MB(1),MC(1),MD(1),E22,E11,E12,E33,THICK,INTL
0237      IF (J.EQ. NNEWIR) K = K + 1
0238      THICK = THK(K)
0239      1010 WRITE (3) 1,MA(1),MB(1),MC(1),MD(1),E22,E11,E12,E33,THICK,INTL
0240      WRITE (6,6100) (THK(I),I=1,INTL)
0241      6100 FORMAT (6F15.5)
0242      1020 CONTINUE
0243      RETURN
0244      END
0245

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0007

SYMBOL		LOCATION	SYMBOL	LOCATION	COMMON BLOCK / CORR	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMBLK	000000	000014	NUMEL	000018	NUMEL	000018	00001C	00000B	00001C	00000C	00000C	00000C	00000C	00000C	00000C	000010
SIZE	000000	000000	HT	000004	HT	000004	000000	000000	000000	000000	000000	000000	000000	000000	000000	000020
Y2OFF	000024	000024	THY	000028	THY	000028	00002C	00002C	00002C	00002C	00002C	00002C	00002C	00002C	00002C	00005C
MEQ	000088	000088	NX	000088	NX	000088	00008C	00008C	00008C	00008C	00008C	00008C	00008C	00008C	00008C	00006C
IFACE	0000C8	0000C8	IHOLE	0000CC	IHOLE	0000CC	0000D0	0000D0	0000D0	0000D0	0000D0	0000D0	0000D0	0000D0	0000D0	0000D8
SYMBOL	000000	000000	SYMBOL	000050	SYMBOL	000050	000064	000064	000064	000064	000064	000064	000064	000064	000064	00008C
E	0000A0	0000A0	FXT	000050	FXT	000050	000064	000064	000064	000064	000064	000064	000064	000064	000064	00008C
SSS	0000A0	0000A0	CMS	000084	CMS	000084	000084	000084	000084	000084	000084	000084	000084	000084	000084	00008C
SYMBOL	000000	000000	SYMBOL	001130	SYMBOL	001130	002260	002260	002260	002260	002260	002260	002260	002260	002260	003390
MA	000000	000000	MB	001130	MB	001130	002260	002260	002260	002260	002260	002260	002260	002260	002260	003398
ND	003380	003380	ND	001130	ND	001130	002260	002260	002260	002260	002260	002260	002260	002260	002260	003398

COMMON BLOCK / RAX / MAP SIZE							
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000000	NR4LDD	000004	NR4STF	000008	NR4DIA	000010
NRECSY	000014	NRECLD	000018	NR1	00001C	NR2	000020
SCALAR MAP							
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NADD	000218	NBDD	00021C	NCOOD	000220	NAEVN	000224
NCEVN	00022C	NP	000230	NK	000234	NV	000238
I	000240	NAA	000244	NBA	000248	NCA	00024C
NBB	000254	NAB	000258	NDA	00025C	J	000260
NY1	000268	NCB	00026C	NMT	000270	NDDH	000274
E11	00027C	E22	000280	E12	000284	E33	000288
INTL	000290					THICK	00028C
SUBPROGRAMS CALLED							
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ISCOM#	000294						
LABEL MAP							
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
90	0003A0	100	000416	150	00046E	300	0004C8
350	00065E	400	000704	400	000714	450	0007A6
505	000806	510	000964	530	0009C6	540	0009E0
550	000842	600	0008EA	605	000C78	610	000C94
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4, DATE 71321							
630	000CFC	650	000D2E	700	000DFO	710	000E44
730	000F3A	740	000F54	750	000F78	770	0010E2
790	0011D0	800	0011FA	802	001318	805	001348
820	0013EE	830	001498	850	0014C2	850	0014FC
865	001694	870	0016DA	885	0016DE	890	001720
900	0017A2	1000	00184C	1010	001934	6100	001A24
				PAGE 0008			
TOTAL MEMORY REQUIREMENTS 001A88 BYTES							
COMPILER HIGHEST SEVERITY CODE WAS 0							
//LODGEV EXEC FORTRAN(BCD,MAP)							
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4, DATE 71321							
PAGE 0001							
C*****OVERLAY AC*****							
0001	SUBROUTINE LODGEN (X,Y,Z,SLOPA)						
0002	COMMON / COMN / NUMP,NUMEL,NLOD,NLIN,MAND,NUMBLK,MEN,MEL(20)						
0003	COMMON / ONE / SIDE,MT,THK(5),THETA,VLOFF,V2OFF,THT,DUM,TC(11)						
	1 - XC(11),HED(12),NX,NY,IOTP,ITYP,IFACE,IMOLE,ITET,ICNT,MEN(4)						
0004	COMMON / LODA / XLOD,YLOD,XROM,YROM,XTROM,YTROM,TORQ,XZ SHM,YZ SHM						
0005	DIMENSION XI(300),YI(300),ZI(300),SLOPA(300)						
0006	COMMON / LODS /						
0007	NM3 = 3*NUMP						
0008	DO 50 I = 1,NUMP						
0009	UX(I) = 0.0						
0010	UY(I) = 0.0						
0011	50 UZ(I) = 0.0						
0012	N1 = NX/2 + 1						
0013	N2 = NX + 1						
0014	N3 = NX/2						
0015	N4 = 2*NX						
0016	KONST = 2*NX*(NY-1)						
0017	NMND = 2*NX + (2*NX-1)*4						

```

0018 NYND = 2*NY + (2*NY-1)*4
0019 YLGT = .2867 + VLDF
0020 Y2LGT = .8667 + Y2OFF
0021 XLGT = (1*Y-1)*Y2LGT + YLGT)*SIDE
0022 YLGT = (FLOA(NY)-.5)*SIDE
0023 IUNIT = 0
0024 IF (ABS(XLOAD) - .001) 60,102,102
0025 60 IF (ITYP - 2) 62,75,75
0026 62 IF (ABS(YLOAD) - .001) 65,110,110
0027 65 IF (ABS(XYLOAD) - .001) 70,170,170
0028 70 IF (ABS(YFOM) - .001) 75,270,270
0029 75 IF (ABS(XFOM) - .001) 80,320,320
0030 80 IF (ITYP - 2) 82,85,85
0031 82 IF (ABS(XYFOM) - .001) 83,360,360
0032 83 IF (ABS(XQSHR) - .001) 84,500,500
0033 84 IF (ABS(YQSHR) - .001) 85,600,600
0034 85 IF (ABS(TORQ) - .001) 800,650,650
C
C FLAT PLATE END LOADS
C
0035 NX
0036 102 PX = XLOAD*XLGT/NXND
0037 IF (ITYP - 2) 105,109,109
0038 103 PX = XLOAD/NXND
0039 105 DO 100 I = 1,N4
0040 UX(I) = UX(I) - PX
0041 100 UX(I+KONST) = UX(I+KONST) + PX
C
C
0042 150 PY = YLOAD*YLGT/NYND
C
C
0043 DO 150 I = 1,NY
0044 IAB = 2*NX*(I-1)
0045 UY(N1+IAB) = UY(N1+IAB) - PY
0046 UY(N2+IAB) = UY(N2+IAB) - PY
0047 UY(N3+IAB) = UY(N3+IAB) + PY
0048 UY(N4+IAB) = UY(N4+IAB) + PY
0049 IF (IUNIT) 65,65,800
C
C
0050 170 PXA = XYLOAD*XLGT/NXND
0051 PYB = XYLOAD*YLGT/NYND
0052 DO 200 I = 1,N4
0053 UY(I) = UY(I) - PXA
0054 UY(I+KONST) = UY(I+KONST) + PXA
0055 DO 250 I = 1,NY
0056 IAS = 2*NX*(I-1)
0057 UX(N1+IAS) = UX(N1+IAS) - PYB
0058 UX(N2+IAS) = UX(N2+IAS) - PYB
0059 UX(N3+IAS) = UX(N3+IAS) + PYB
0060 UX(N4+IAS) = UX(N4+IAS) + PYB
0061 IF (IUNIT) 70,70,800
C
C
0062 270 PNY = YND*YLGT/(NY*NY)
0063 DO 300 I = 1,NY
0064 IAB = 2*NX*(I-1)
0065 UY(N1+IAB) = UY(N1+IAB) + PNY
0066 UY(N2+IAB) = UY(N2+IAB) - PNY
0067 UY(N3+IAB) = UY(N3+IAB) + PNY
0068 300 UY(N4+IAB) = UY(N4+IAB) - PNY

```

PAGE 0002

DATE 71321

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4


```

0118      UZ(I+3*NX/2) = UZ(I+3*NX/2) + PZ
0119      UX(I+KONST) = UX(I+KONST) + PMY
0120      UZ(I+KONST) = UZ(I+KONST) - PZ
0121      UX(I+KONST+NX/2) = UX(I+KONST+NX/2) - PMY
0122      UZ(I+KONST+NX/2) = UZ(I+KONST+NX/2) - PZ
0123      UX(I+KONST+NX) = UX(I+KONST+NX) + PMY
0124      UZ(I+KONST+NX) = UZ(I+KONST+NX) - PZ
0125      UX(I+KONST+3*NX/2) = UX(I+KONST+3*NX/2) - PMY
0126      UZ(I+KONST+3*NX/2) = UZ(I+KONST+3*NX/2) - PZ
0127      IF (IUNIT) 84,84,800

520      IF (IUNIT) 84,84,800

C
0128      QY
0129      PZ = QSHR*VLGT/MYND
0130      PMX = QSHR*SIDE*(1.86657*Y2OFF)*VLGT/(12.0*HT*MY)
0131      DO 620 I = 1, NY
0132      IAB = 2*NX*(I-1)
0133      UY(I+IAB) = UY(I+IAB) - PMX
0134      UZ(I+IAB) = UZ(I+IAB) + PZ
0135      UY(I+2*IAB) = UY(I+2*IAB) + PMX
0136      UZ(I+2*IAB) = UZ(I+2*IAB) - PZ
0137      UY(I+3*IAB) = UY(I+3*IAB) + PMX
0138      UZ(I+3*IAB) = UZ(I+3*IAB) - PZ
0139      UY(I+4*IAB) = UY(I+4*IAB) - PMX
0140      UZ(I+4*IAB) = UZ(I+4*IAB) - PZ
0141      IF (IUNIT) 85,85,800

C
0141      C
0142      GENERATE TORSION LOADS ON CYLINDER
0143      650 IF (ITYP - 2) 800,660,720
0144      660 RAD1 = SORT(V11)*2+Z(11)*2
0145      K = NX/2 + 1
0146      RAD2 = SORT(V11)*2+Z(K)*2
0147      PA = TORQ/(NX*RAD1+RAD2*2/RAD1)*3.1
0148      PB = RAD2/RAD1*PA
0149      DO 700 I = 1, M
0150      THETA = ATAN2(Z(I),Y(I))
0151      SINTH = SIN(THETA+1.57079)
0152      COSH = COS(THETA+1.57079)
0153      PX = PA
0154      IF (I - GT. NX/2 .AND. I .LE. NX) PX = PB
0155      IF (I - GT. 3*NX/2) PX = PB
0156      UY(I) = UY(I) + PX*COSH
0157      UZ(I+KONST) = UY(I+KONST) - PX*COSH
0158      GO TO 800

700      UZ(I+KONST) = UY(I+KONST) - PX*SINTH
0159      GO TO 800

C
0159      C
0160      GENERATE TORSION LOADS ON AIRFOIL
0161      720 NEVD = 5*NX/4 + 1
0162      ARM = 0.0
0163      DO 750 I = 1, M
0164      ANG = SLOPA(I)
0165      PA = TORQ/ARM/3.
0166      DO 760 I = 1, M
0167      ANG = SLOPA(I)
0168      IF (Z(I) - LT. 0.0) ANG = ANG + 3.14159
0169      UY(I) = UY(I) + PA*COS(ANG)
0170      UZ(I) = UZ(I) + PA*SIN(ANG)
0171      UY(I+KONST) = UY(I+KONST) - PA*COS(ANG)
0172      UZ(I+KONST) = UZ(I+KONST) - PA*SIN(ANG)

```

PAGE 0004

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

TAPE 3

```

0172      800 CONTINUE
0173      850 WRITE (3) (UX(I),UY(I),UZ(I),I=1,MUMNP)
0174      DO 900 I = 1,MUMNP
0175      900 CODE(I) = 0.0
0176      IF (IMOLE -GT. 0) CODE(IMOLE) = 7.
0177      CODE(I) = 7.
0178      CODE(I+1) = 3.
0179      CODE(I+7) = 6.
0180      WRITE (3) (CODE(I),I=1,MUMNP)
0181      RETURN
0182      END
    
```

TAPE 3

SYMBOL MUMNP NUMBLK	LOCATION 000000 000014	SYMBOL NUMBL MBN	COMMON BLOCK / COHN / MAP SIZE 00006C			SYMBOL MBAND	LOCATION 00000C 000010
			LOCATION 000004 000018	SYMBOL NLOD NEL	LOCATION 000008 00001C		
SYMBOL SIDE Y2OFF MED IFACE	LOCATION 000000 000024 00003C 00004C 000058	SYMBOL MT THT HK IMOLE	LOCATION 000004 000028 000038 00004C	SYMBOL THK DJM NY ITET	LOCATION 000008 00002C 00003C 00004C	SYMBOL Y2OFF XC ITYP MNEW	LOCATION 000020 00003C 00004C 000058
SYMBOL XLOAD XZMOM	LOCATION 000000 000014	SYMBOL YLOAD TORQ	LOCATION 000004 000018	SYMBOL XYLOD XQSHR	LOCATION 000008 00001C	SYMBOL Y2MOM	LOCATION 000020 000030
SYMBOL CODE	LOCATION 000000	SYMBOL UX	LOCATION 000048	SYMBOL UY	LOCATION 000060	SYMBOL	LOCATION 000010
SYMBOL M3 M4 Y2LGT PY PX RAD1 SINTH	LOCATION 0001A8 0001BC 0001D0 0001E4 0001F8 00020C 000220	SYMBOL I KONST HLGT TAB SUM K COSTH	LOCATION 0001AC 0001C0 0001D4 0001E8 0001FC 000210 000224	SYMBOL M1 NAND YVLT PHYA RAD2 MEMO	LOCATION 0001B0 0001C4 0001D8 0001EC 000200 000214 000228	SYMBOL M3 VILGT PX PHY PZ PB ANG	LOCATION 0001B8 0001CC 0001E0 0001F4 000208 00021C 000230
SYMBOL K	LOCATION 000234	SYMBOL Y	LOCATION 000238	SYMBOL Z	LOCATION 00023C	SYMBOL	LOCATION 000240
SYMBOL IBCON8	LOCATION 000244	SYMBOL SORT	LOCATION 000248	SUBPROGRAMS CALLED SYMBOL ATANZ LOCATION 00024C		SYMBOL COS	LOCATION 000254

LABEL		LOCATION		LABEL		LOCATION		LABEL		LOCATION		LABEL		LOCATION		LABEL		LOCATION	
50	00036A	60	00048A	62	0004CC	65	0004E6	70	000500										
75	00051A	80	00053A	82	00054C	83	000566	84	000580										
85	00059A	102	0005BA	103	000600	105	00062E	100	000642										
110	0006AA	150	00079E	170	0007F2	200	00086E	250	000976										
270	0009CA	300	000AC6	320	00081A	325	000832	330	000804										
335	000D74	340	000D88	345	000E04	350	000E4C	360	000E5E										

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0007

TOTAL MEMORY REQUIREMENTS 001E90 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//MODCAN EXEC FORTRAN(BCD,MAP)

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0001

***** OVERLAY AD *****

0001 SUBROUTINE MODCAN (MRT,X,Y,Z,SLOPA)
0002 COMMON / COMN / MUMNP,MUMEL,NLDD,NLIN,MHAND,NUMBLK,NBN,NEL(20)
0003 COMMON / ONE / SIDE,MT,THK(5),THETA,YIOFF,Y2OFF,TMT,DUR,IC(11)
0004 1, XC(11),HED(12),MX,MV,IOPT,ITYP,IFACE,INOLE,ITET,ICNT,MWEN(4)
0005 DIMENSION X(300),Y(300),Z(300),SLOPA(300)

C GENERATE FLAT PLATE MODAL COORDINATES

0005 XC(11) = XC(11) + .00001
0006 MODLG = 4*MY - 1
0007 SINTH = SIN(THETA/57.2958)
0008 TANTH = TAN(THETA/57.2958)
0009 XOFF = (-28867 + YIOFF)*SIDE*TANTH
0010 MUMNP = MX*MY + MY*(3*MY-1)
0011 X1 = (TMT - MT)/TMT*SIDE
0012 X2 = (SIDE/2. + XOFF)*MT/TMT
0013 Y1 = (-86603*Y2OFF)*SIDE - (-57734*YIOFF)*SIDE*MT/TMT
0014 Y2 = (-28867 + YIOFF)*SIDE*MT/TMT
0015 YINC = SIDE*(-86603*Y2OFF)
0016 Y2LGT = .86603 + Y2OFF
0017 IK = 0

0018 DC 300 I = 1,MX
0019 IF (IK) 110,110,120
0020 110 I1 = (I-1)*MODLG + 1
0021 V(11) = Y1 + YINC*(I-1)
0022 V(11+1) = Y2 + YINC*(I-1)
0023 V(11+2) = YINC*(I-1)
0024 V(11+3) = Y(11+1)
0025 X(11) = V(11)*SINTH
0026 X(11+1) = X1/2. + V(11+1)*SINTH
0027 X(11+2) = SIDE/2. + V(11+2)*SINTH
0028 X(11+3) = SIDE - X1/2. + V(11+3)*SINTH
0029 Z(11) = MT
0030 Z(11+1) = MT
0031 Z(11+2) = 0.0
0032 Z(11+3) = MT
0033 IK = 1
0034 I1 = I1 + 3
0035 GO TO 150

FORTRAN IV	MODEL 44	PS	VERSION 3.	LEVEL 4	DATE 71321	PAGE 0002
0036	120	11 = (11-1)*MODLG + 1				
0037		Y(11) = YINC*(11-1)				
0038		Y(11+1) = Y(11) + Y2				
0039		Y(11+2) = Y(11) + Y1				
0040		Y(11+3) = Y(11+1)				
0041		X(11) = Y(11)*SINTH				
0042		X(11+1) = X2 + Y(11+1)*SINTH				
0043		X(11+2) = SIDE/2. + Y(11+2)*SINTH				
0044		X(11+3) = X(11+1) + X1 + Y(11+3)*SINTH				
0045		Z(11) = 0.0				
0046		Z(11+1) = HT				
0047		Z(11+2) = HT				
0048		Z(11+3) = HT				
0049		IK = 0				
0050		11 = 11 + 3				
0051	150	DO 300 J = 5,MODLG				
0052		11 = 11 + 1				
0053		X(11) = X(11-4) + SIDE				
0054		Y(11) = Y(11-4)				
0055	300	Z(11) = Z(11-4)				
0056		IF (IOPT .NE. 1) WRITE (6,6150) X(MODLG)				
0057	6150	FORMAT (// ,14X, BLENGTH 0 , F15.5)				
0058		IF (ITYP - 2) 800,350,550				
0059		C GENERATE CYLINDER COORDINATES				
0060	350	CIRC = Y2LGT*SIDE*NX				
0061		RAD = Y2LGT*SIDE*NX/6.28318				
0062		IF (IOPT .NE. 1) WRITE (6,6200) RAD				
0063	6200	FORMAT (// ,10X, 17HCYLINDER RADIUS 0 , F15.5)				
0064	6000	FORMAT (6F15.5)				
0065		DO 400 I = 1,NMNP				
0066		THETA = (Y(11)/CIRC)*6.28318				
0067		ABC = HT - Z(11)				
0068		Y(11) = (RAD-ABC)*COS(THETA)				
0069	400	Z(11) = (RAD-ABC)*SIN(THETA)				
0070	450	CONTINUE				
		GO TO 800				
0071		C GENERATE AIRCIL COORDINATES				
0072	550	CONTINUE				
0073		CHORD = Y2LGT*SIDE*NX/2.				
0074		IF (IOPT .NE. 1) WRITE (6,6250) CHORD				
	6250	FORMAT (// ,10X, 15HAIRFOIL CHORD 0 , F15.5)				
0075		DO 640 I = 1,NMNP				
0076		FACT = 1.0				
0077		IF (Y(11) - CHORD) 610,610,605				
0078	605	Y(11) = 2.*CHORD - Y(11)				
0079		FACT = -1.0				
0080	610	XOC = Y(11)/CHORD				
0081		CALL CURV (XOC,YOC,XC,TC,SLOP)				
0082		SLOPA(11) = SLOP*FACT				
0083		ZOLD = Z(11)				
0084		Z(11) = TOC*CHORD*FACT				
0085		IF (ZOLD - .001) 640,620,620				
0086	620	Z(11) = Z(11) - HT*FACT				
0087		IF (FACT) 625,622,622				
0088	622	IF (Z(11)) 625,640,640				
0089	624	IF (Z(11)) 640,640,625				


```

0090      625 Z(1) = 0.0
0091      IF (Y(1) - CHORD/2.) 630,630,635
0092      630 Y(1) = ABS(HT/SLOPA(I))
0093      GO TO 640
0094      635 Y(1) = CHORD - ABS(HT/SLOPA(I))
0095      640 CONTINUE

```

C

```

0096      800 CONTINUE
0097      6020 FORMAT (10X, 10HMODAL DATA )
0098      DO 500 I = 1,NUMMP
0099      6010 FORMAT (14,20X,4F12.5 )
0100      500 CONTINUE

```

C

```

0101      IF (IMRY -EQ. 1) RETURN
0102      WRITE (3) (I,X(I),Y(I),Z(I),I=1,NUMMP)
0103      RETURN
0104      END

```

TAPE 3

SYMBOL NUMMP NUMBER	LOCATION	SYMBOL NUMEL NUMBER	COMMON BLOCK / COMN / MAP SIZE			LOCATION	SYMBOL MBAND	LOCATION
			LOCATION	SYMBOL	SIZE			
	000000		000004	NLOD	000008	00000C		000010
	000014	NBN	000010	NEL	00001C			

SYMBOL SIDE VZOFF WED IFACE	LOCATION	SYMBOL	COMMON BLOCK / ONE / MAP SIZE			LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL	SIZE			
	000000	HT	000004	THK	000008	00001C		000020
	000024	THT	000028	DUM	00002C	000030	XC	00003C
	000088	HX	000088	MY	00008C	0000C0	ITYP	0000C4
	0000C8	THOLE	0000CC	ITET	0000D0	0000D4	NNEW	0000D8

SYMBOL VZDLC X2 IK RAD TDC	LOCATION	SYMBOL	SCALAR MAP			LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL	SIZE			
	000154	SINTH	000158	TANTH	00015C	000160	XI	000164
	000168	YI	00016C	Y2	000170	000174	Y2LGT	000178
	00017C	I	000180	II	000184	000188	CIRC	00018C
	000190	ABC	000194	CHORD	000198	00019C	XDC	0001A0
	0001A4	SLOP	0001A8	ZOLD	0001AC	0001B0		

SYMBOL X	LOCATION	SYMBOL	ARRAY MAP			LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL	SIZE			
	0001B4	Y	0001B8	Z	0001BC	0001C0		

SYMBOL 18C3H8	LOCATION	SYMBOL	SUBPROGRAMS CALLED			LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL	SIZE			
	0001C4	CURV	0001C8	SIN	0001CC	0001D0	COS	0001D4

LABEL	LOCATION	LABEL	LABEL MAP			LOCATION	LABEL	LOCATION
			LOCATION	SYMBOL	SIZE			
110	0003D0	120	00052E	150	00083C	000884	6150	0008FC
350	000734	6200	0007C0	6000	0007E6	00085E	450	000892
550	000898	6250	0008FC	605	000956	000982	620	0009EC
622	000A22	624	000A48	625	000A68	000A9E	635	000ACE
640	000AFE	800	000B18	6020	000B18	000B3A	500	000B4C

TOTAL MEMORY REQUIREMENTS 000C6C BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//PLATCA EXEC FORTRAN18CO.MAP)

***** OVERLAY AE *****

0001 SURROUTINE PLATGA
0002 COMMON / COMN / NUMMP, NUMEL, NLOD, NLIN, MBAND, NUMBLK, NBN, MEL (20)
0003 COMMON / ONE / SIDE, HT, THK (5), THETA, Y1OFF, Y2OFF, TMT, DUM, TC (11)
0004 1 .XC (11), HED (12), NX, NY, IOT, ITP, IFACE, IHDLE, ITET, ICNT, MNEW (4)
0005 COMMON / MATL / E14, S1, FXT (5), FYT (5), FXC (5), FYC (5), SSS (5), CHS (10)
0006 COMMON / PLAT / NALL100, NALL100, NC (1100), NST (2), INT (6), NDI100)
0007 COMMON / RAN / NR4, NRALOD, NR4STP, NR4DIA, NRECL, NMECST, NRECLD, NR1, NR2
0008 IF (ICNT .GT. 0) GO TO 1000
MAEVN = 1

C GENERATE VERTICAL LEGS

0009 NST (1) = 2
0010 NST (2) = 4*NY
0011 NPTR = 2*NY - 2
0012 INT (1) = 2
0013 INT (2) = -1
0014 INT (3) = 4
0015 INT (4) = -3
0016 INT (5) = 2
0017 NLEG = 8*NY - 2
0018 K = 0
0019 NP = 0
0020 DO 300 I = 1, NX

0021 NP = NP + 1
0022 IF (K) 210, 210, 220
0023 210 NA (NP) = NST (1) + NLEG * (I - 1) / 2
0024 NK = 1
0025 K = 1

0026 GO TO 230
0027 220 NA (NP) = NST (2) + NLEG * (I - 2) / 2
0028 NK = 3
0029 K = 0

0030 230 NB (NP) = NA (NP) + INT (NK)
0031 NC (NP) = NR (NP) + INT (NK + 1)
0032 ND (NP) = 0
0033 IF (NK .EQ. 3) ND (NP) = NC (NP) + INT (NK + 2)

0034 DO 300 J = 2, NPTR
0035 NP = NP + 1
0036 NK = NK + 2
0037 IF (NK .EQ. 5) NK = 1

0038 250 NA (NP) = NC (NP - 1)
0039 IF (NK .EQ. 1) NA (NP) = ND (NP - 1)
0040 260 NB (NP) = NA (NP) + INT (NK)
0041 NC (NP) = NB (NP) + INT (NK + 1)
0042 ND (NP) = 0

0043 IF (NK .EQ. 1) GO TO 300
0044 ND (NP) = NC (NP) + INT (NK + 2)
0045 300 CONTINUE

C GENERATE SKEW A LEGS

0046 MNEW (1) = NP + 1
0047 NLEG = 8*NY - 2
0048 NST (1) = 4*NY
0049 NST (2) = 4*NY + NK / 2 - 1 + NLEG + 3
0050 INT (1) = 4*NY + 2
0051 INT (2) = 4*NY - 1
0052 INT (3) = INT (1) + 1
0053 INT (4) = -2
0054 INT (5) = 1

```

0055 NP = NP + 1
0056 NA(NP) = MST(1)
0057 NB(NP) = NA(NP) + INT(3)
0058 NC(NP) = NB(NP) + INT(4)
0059 ND(NP) = NC(NP) + INT(5)
0060 NPIR = 5
0061 NDSA = NX/2 + NY - 1
0062 DO 500 I = 2, NDSA
0063 NP = NP + 1
0064 IF (I-NX/2) 350, 350, 360
0065 350 NA(NP) = MST(1) / (I-1) * NLEG
0066 NK = 3
0067 GO TO 370
0068 360 NA(NP) = MST(2) + (I-NX/2-1) * 4
0069 NK = 1
0070 370 NB(NP) = NA(NP) + INT(NK)
0071 NC(NP) = NB(NP) + INT(NK+1)
0072 ND(NP) = 0
0073 IF (NK .EQ. 3) ND(NP) = NC(NP) + INT(NK+2)
0074 DO 450 J = 2, NPIR
0075 NP = NP + 1
0076 NK = NK + 2
0077 IF (NK .EQ. 5) NK = 1
0078 410 NA(NP) = NC(NP-1)
0079 IF (NK .EQ. 1) NA(NP) = ND(NP-1)
0080 420 NB(NP) = NA(NP) + INT(NK)
0081 NC(NP) = NB(NP) + INT(NK+1)
0082 ND(NP) = 0
0083 IF (NK .EQ. 1) GO TO 450
0084 ND(NP) = NC(NP) + INT(NK+2)
0085 450 CONTINUE
0086 IF (I .LT. NX/2) NPIR = NPIR + 4
0087 IF (I .GE. NX/2 .AND. I .GE. NY) .OR. (I .GE. NY) NPIR = NPIR-4
0088 IF (I .EQ. NY) NPIR = NPIR + 1
0089 IF (I .EQ. NX/2) NPIR = NPIR + 1
0090 500 CONTINUE
0091 IF (I .EQ. 1) GO TO 540
0092 C 219 UP SIDE OF CYLINDER
0093 NDA = 2
0094 MOB = (4*NY-1) * (NX-1) + 3
0095
0094 DO 530 I = 1, NY
0095 NP = NP + 1
0096 NA(NP) = NDA + (I-1) * 4
0097 NB(NP) = MOB + (I-1) * 4
0098 NC(NP) = NA(NP) + 1
0099 ND(NP) = 0
0100 IF (I .EQ. NY) GO TO 530
0101 NP = NP + 1
0102 NA(NP) = NC(NP-1)
0103 NB(NP) = NB(NP-1) + 2
0104 NC(NP) = NB(NP-1)
0105 ND(NP) = NC(NP) + 1
0106 530 CONTINUE
0107 C GENERATE SKEW B LEGS
0108 540 MNEW(2) = NP + 1
0109 MST(1) = NLEG * (NX/2-1) + 1
0110 NST(2) = 3
0111 INT(1) = 4*NY
0112 INT(2) = -1
0113 INT(3) = 4*NY + 1

```

PAGE 0003

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0113 INT(4) = -4*NY
0114 INT(5) = 1
0115 NOSB = NX/2 + NY - 1
0116 NP = NP + 1
0117 NA(NP) = NST(1)
0118 NB(NP) = NA(NP) + INT(1)
0119 NC(NP) = NB(NP) + INT(2)
0120 ND(NP) = 0
0121 NPIR = 5
0122 DO 700 I = 2, NOSB
0123 NP = NP + 1
0124 IF (1-NX/2) 550, 550, 560
0125 550 MAINP = NST(1) - (1-1)*NLEG
0126 NK = 1
0127 GO TO 570
0128 560 MAINP = NST(2) + (1-NX/2-1)*4
0129 NK = 3
0130 NB(NP) = MAINP + INT(NK)
0131 NC(NP) = NB(NP) + INT(NK+1)
0132 ND(NP) = 0
0133 IF (NK .EQ. 3) ND(NP) = NC(NP) + INT(NK+2)
0134 DO 650 J = 2, NPIR
0135 NP = NP + 1
0136 NK = NK + 2
0137 IF (NK .EQ. 5) NK = 1
0138 NA(NP) = NC(NP-1)
0139 IF (NK .EQ. 1) NA(NP) = ND(NP-1)
0140 NB(NP) = NA(NP) + INT(NK)
0141 NC(NP) = NB(NP) + INT(NK+1)
0142 ND(NP) = 0
0143 IF (NK .EQ. 1) GO TO 650
0144 ND(NP) = NC(NP) + INT(NK+2)
0145 650 CONTINUE
0146 IF (1 .LT. NX/2) NPIR = NPIR + 4
0147 IF ((1 .GE. NX/2 .AND. 1 .GE. NY) .OR. (1 .GE. NY) .AND. (1 .GE. NX/2)) NPIR = NPIR + 4
0148 IF (1 .EQ. NY) NPIR = NPIR + 1
0149 IF (1 .EQ. NX/2) NPIR = NPIR + 1
0150 700 CONTINUE
0151 IF (11*P .EQ. 1) GO TO 740
0152 ZIP UP SIDE OF CYLINDER
0153 NOA = 3
0154 DO 730 I = 1, NY
0155 NP = NP + 1
0156 NA(NP) = NOB + (1-1)*4
0157 NB(NP) = NOA + (1-1)*4
0158 NC(NP) = NA(NP) + 1
0159 ND(NP) = NC(NP) + 1
0160 IF (1 .EQ. NY) GO TO 730
0161 NP = NP + 1
0162 NA(NP) = ND(NP-1)
0163 NB(NP) = NB(NP-1) + 1
0164 NC(NP) = NB(NP-1)
0165 ND(NP) = 0
0166 730 CONTINUE
0167 740 NUNEL = NP
0168 WRITE (6, 6200) NAEVN, NNEW(1), NNEW(2)

```

PAGE 0004

DATE 71321

LEVEL 4

PS

MODEL 44

VERSION 3

FORTRAN IV

6200 FORMAT 1, //, 5X, 22HFIRST VERTICAL PLATE 0, 14, 5X, 8MSKEW A 0,
1 14, 5X, 8MSKEW B 0, 14)

J = 1
NODM = (4*NY-1)*NX/2 + 2*NY - 1
IF (IHOLE -GT- 1) NODM = IHOLE
DO 900 I = 1, NUMEL
IF (I.EQ. MNEM(J)) J = J + 1
XNU = 1. - E(4, J)*E(2, J)/E(1, J) *E(4, J)
E11 = E(1, J)/XNU
E22 = E(2, J)/XNU
E12 = E(1, J)*E(4, J)/XNU
E33 = E(3, J)
IF (IHOLE-1) 890, 860, 860
860 IF (NA(1) -NODM) 865, 885, 865
865 IF (NA(1) -NODM) 870, 885, 870
870 IF (ND(1) -NODM) 880, 885, 880
880 IF (ND(1) -NODM) 890, 885, 890
885 E11 = 0.0
E22 = 0.0
E12 = 0.0
E33 = 0.0

PAGE 0005

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

0189 IHOLE = NODM
0190 WRITE (6, 6150) NODM
0191 6150 FORMAT (//, 5X, 25HPLATES CONNECTING TO NODE. 15, 10H HAVE BEEN
1 40H GIVEN ZERO STIFFNESS TO SIMULATE A HOLE)
0192 890 THICK = THK(J)
0193 6000 FORMAT (615)
0194 900 WRITE (3) I, NA(1), NB(1), NC(1), ND(1), E22, E11, E12, E33, THICK, J
0195 GO TO 1020
0196 1000 CONTINUE
0197 K = 1
0198 DO 1010 J = 1, NUMEL
0199 NR2=J
0200 READ(2, NR2) I, NA(1), NB(1), NC(1), ND(1), E22, E11, E12, E33, THICK, INTL
0201 IF (J.EQ. MNEM(K)) K = K + 1
0202 THICK = THK(K)
0203 1010 WRITE (3) I, NA(1), NB(1), NC(1), ND(1), E22, E11, E12, E33, THICK, INTL
0204 6100 WRITE (6, 6100) (THK(I), I=1, INTL)
0205 6100 FORMAT (6F15.5)
0206 1020 CONTINUE
0207 RETURN
0208 END

TAPE 3

TAPE 2

TAPE 3

PAGE 0006

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

SYMBOL NUMP NUMBLK	LOCATION 000000 000014	SYMBOL NUMEL MBN	COMMON BLOCK / COMN		LOCATION 000004 000018	SYMBOL NL0D NEL	MAP SIZE 000008 00001C	SYMBOL NLIN	LOCATION 00000C	SYMBOL MBAND	LOCATION 000010
			SYMBOL	LOCATION							
SYMBOL	LOCATION	SYMBOL	SYMBOL	LOCATION	SYMBOL	LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	MT	000004	000004	THK	000008	000008	SYMBOL	00001C	YIOFF	00002C
Y2OFF	000024	THY	000028	000028	DUM	00002C	TC	000030	XC	00005C	
MED	000085	MX	000088	000088	NY	00008C	IDPT	0000C0	ITYP	0000C4	
IFACE	0000C8	IHOLE	0000CC	0000CC	ITEP	0000D0	ICNT	0000D4	MNEW	0000D8	


```

0012      UX(1) = 0.0
0013      UY(1) = 0.0
0014      UZ(1) = 0.0
0015      M1 = 4*(NY-1)
0016      M2 = 4*(NY-1)*(MX-1) - 1
0017      M3 = 4*NY-2
0018      V2 = (-28867*VLOFF)*SIDEENT/THT
0019      NXND = 2*MX + 5*ZMX-1 - 1
0020      NYND = 2*NY + 5*ZNY-1 - 1
0021      XLGT = (MX-1)*SIDE*(-86057*Y2OFF) + V2
0022      YLGT = (FLOAT(MY) - .5)*SIDE
0023      M4 = 2*NY
0024      KONST = 2*NY*(MX-1)
0025      LUNIT = 0
0026      IF (ITYP -GT- 1) GO TO 37
0027      XFIX = 3.0
0028      YFIX = 3.0
0029      IF (ABS(XQSHR) .LT. .001) GO TO 20
0030      XFIX = 2.0
0031      YFIX = 2.0
0032      20 IF (ABS(YQSHR) .LT. .001) GO TO 25
0033      XFIX = 1.0
0034      YFIX = 1.0
0035      25 CONTINUE
0036      DO 30 I = 1,MX
0037      J = (I-1)*(4*NY-1)
0038      DO 30 K = 1,3
0039      CODE(J+K) = XFIX
0040      30 CODE(J+K*N1) = XFIX
0041      L = 0
0042      K = 1
0043      DO 35 I = 1,MY
0044      K = K + 1
0045      L = L + 1
0046      IF (L -EQ- 4) GO TO 32
0047      CODE(K) = YFIX

```

PAGE 0002

DATE 71321

LEVEL 4

MODEL 44 PS

FORTRAN IV

```

0048      32 IF (L -EQ- 3) GO TO 35
0049      CODE(K*N2) = YFIX
0050      35 IF (L -EQ- 4) L = 0
0051      37 CONTINUE
0052      55 IF (ABS(XLOAD) - .001) 60,102,102
0053      60 IF (ITYP - 2) 62,75,75
0054      62 IF (ABS(YLOAD) - .001) 65,110,110
0055      65 IF (ABS(XYLOD) - .001) 70,170,170
0056      70 IF (ABS(YMOD) - .001) 75,270,270
0057      75 IF (ABS(XMOD) - .001) 80,320,320
0058      80 IF (ITYP - 2) 82,85,85
0059      82 IF (ABS(XYMOD) - .001) 83,360,360
0060      83 IF (ABS(XQSHR) - .001) 84,500,500
0061      84 IF (ABS(YQSHR) - .001) 85,600,600
0062      85 IF (ABS(XORQ) - .001) 800,650,650

```

```

C      FLAT PLATE END LOADS
C      MX
C      102 PX = XLOAD*XLGT/MXND
C      L = 1
C      IF (ITYP-2) 105,103,103
C      103 PX = XLOAD/MXND

```

```

0067 105 DO 109 I = 1,NX
0068 J = (I-1)*(4*NY-1)
0069 GO TO (106,107),L
0070 106 M = 3
0071 L = 2
0072 GO TO 108
0073 107 M = 1
0074 L = 1
0075 108 CONTINUE
0076 DO 109 K = 1,3
0077 FACT = 1.0
0078 IF (K - M) 112,111,112
0079 111 FACT = 2.0
0080 IF (I .EQ. 1) FACT = 1.0
0081 112 UX(J+K) = UX(J+K) - PX*FACT
0082 109 UX(J+K+M) = UX(J+K+M) + PX*FACT
0083 IF (IUNIT) 60,60,800
C
0084 NY
0085 110 PY = YLOAD*VLGT/MYND
0086 K = 1
0087 L = 0
0088 DO 150 I = 1,NA
0089 K = K + 1
0090 L = L + 1
0091 IF (L-4) 120,130,120
120 FACT = 1.0
C
0092 IF (L .EQ. 2) FACT = 2.0
0093 IF (I .EQ. NA) FACT = 1.0
0094 UX(K) = UX(K) - PY*FACT
0095 130 IF (L-3) 140,150,140
0096 140 FACT = 1.0
0097 IF (L .EQ. 1) FACT = 2.0
0098 IF (I .EQ. 1) FACT = 1.0
0099 UX(K+2) = UX(K+2) + PY*FACT
0100 150 IF (L .EQ. 4) L = 0
0101 IF (IUNIT) 65,65,800
C
0102 NXY
0103 LOAD ON X FACE
0104 170 PXVA = XYLOAD*XLGT/MYND
0105 L = 1
0106 DO 200 I = 1,NX
0107 J = (I-1)*(4*NY-1)
0108 GO TO (180,185),L
0109 180 M = 3
0110 L = 2
0111 GO TO 190
0112 185 M = 1
0113 L = 1
0114 190 CONTINUE
0115 DO 200 K = 1,3
0116 FACT = 1.0
0117 IF (K - M) 194,192,194
0118 192 FACT = 2.0
0119 IF (I .EQ. 1) FACT = 1.0
0120 194 UX(J+K) = UX(J+K) - PXVA*FACT
200 UX(J+K+M) = UX(J+K+M) + PXVA*FACT
C
0120 LOAD ON Y FACE
PXVB = XYLOAD*VLGT/MYND

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

PAGE 0003


```

0121 K = 1
0122 L = 0
0123 DO 250 I = 1,NA
0124 K = K + 1
0125 L = L + 1
0126 IF (L - 4) 210,220,210
0127 210 FACT = 1.0
0128 IF (L .EQ. 2) FACT = 2.0
0129 IF (L .EQ. NA) FACT = 1.0
0130 UX(K) = UX(K) - PXY*FACT
0131 220 IF (L-3) 230,250,230
0132 230 FACT = 1.0
0133 IF (L .EQ. 1) FACT = 2.0
0134 IF (L .EQ. 1) FACT = 1.0
0135 UX(K+2) = UX(K+2) + PXY*FACT
0136 250 IF (L .EQ. 4) L = 0

```

PAGE 0004

DATE 71321

LEVEL 4

VERSION 3

MODEL 44 PS

FORTRAN IV

```

0137 IF (IUNIT) 70,70,800
C
C
0138 MY
0139 270 PMV = YPOM*VLGT/(HT*(2*MY-1))
0140 K = 1
0141 L = 0
0142 DO 300 I = 1,NA
0143 K = K + 1
0144 L = L + 1
0145 IF (L-4) 275,280,275
0146 275 FACT = -1.0
0147 IF (L .EQ. 2) FACT = 2.0
0148 IF (L .EQ. NA) FACT = 1.0
0149 UX(K) = UX(K) - PMV*FACT
0150 280 IF (L-3) 290,300,290
0151 290 FACT = -1.0
0152 IF (L .EQ. 1) FACT = 2.0
0153 IF (L .EQ. 1) FACT = 1.0
0154 UX(K+2) = UX(K+2) + PMV*FACT
0155 300 IF (L .EQ. 4) L = 0
IF (IUNIT) 75,75,800

```

```

C
C
0156 MX
0157 320 CONTINUE
0158 IF (ITYP - 2) 325,325,325
0159 325 PMK = XMON*VLGT/(HT*(2*MX-1))
0160 L = 1
0161 DO 330 I = 1,MX
0162 J = (I-1)*(4*MY-1)
0163 GO TO (326,327),L
0164 326 M = 3
0165 L = 2
0166 GO TO 328
0167 327 M = 1
0168 L = 1
0169 328 CONTINUE
0170 DO 330 K = 1,3
0171 IF (K - M) 331,329,331
0172 FACT = -1.0
0173 329 FACT = 2.0
0174 IF (I .EQ. 1) FACT = 1.0
0175 331 UX(J+K) = UX(J+K) - PMK*FACT
330 UX(J+K+1) = UX(J+K+1) + PMK*FACT

```

```

0176      GO TO 350
0177      C      BENDING MOMENT ON CYLINDER AND AIRFOIL
0178      335 SUM = 0.0
0179      DO 341 I = 1,NX
0180      J = (I-1)*(4*NY-1)
      GO TO (336,337),L

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0805

```

0181      336 M = 3
0182      L = 2
0183      GO TO 338
0184      337 M = 1
0185      L = 1
0186      338 CONTINUE
0187      DO 341 K = 1,3
0188      FACT = 1.0
0189      IF (K - M) 340,339,340
0190      339 FACT = 2.0
0191      340 CONTINUE
0192      341 SUM = SUM + Z(J,K)*FACT
0193      PHX = X*OM/SUM/3.0
0194      DO 348 I = 1,NX
0195      J = (I-1)*(4*NY-1)
0196      GO TO (343,344),L
0197      343 M = 3
0198      L = 2
0199      GO TO 345
0200      344 M = 1
0201      L = 1
0202      345 CONTINUE
0203      DO 348 K = 1,3
0204      FACT = 1.0
0205      IF (K - M) 347,346,347
0206      346 FACT = 2.0
0207      347 UX(J,K) = UX(J,K) + PHX*FACT*(I)
0208      348 UX(J,K+M1) = UX(J,K+M1) - PHX*FACT*(I)
0209      C      PKV
0210      360 PHXVB = X*OM*VLC/(INT(2*NY-1))
0211      C      PHXVA = X*OM*VLC/(INT(2*NY-1))
      LOAD ON X FACES
      L = 1
      DO 400 I = 1,NX
      J = (I-1)*(4*NY-1)
      GO TO (370,375),L
      370 M = 3
      L = 2
      GO TO 380
      375 M = 1
      L = 1
      380 CONTINUE
      DO 400 K = 1,3
      FACT = 1.0
      IF (K - M) 390,389,390
      389 FACT = 2.0
      IF (I - EQ, 1) FACT = -1.0
      390 UX(J,K) = UX(J,K) - PHXVA*FACT

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

400 UY(J+K-N1) = UY(J+K-N1) + PNYVDFACT

C LOAD ON Y FACES

K = 1

L = 0

DO 450 I = 1,NA

K = K + 1

L = L + 1

410 FACT = 1.0

IF (L - 4) 410,415,410

IF (L - EQ. 2) FACT = -2.0

IF (L - EQ. NA) FACT = -1.0

UY(K) = UY(K) - PNYVDFACT

415 IF (L-3) 420,450,420

420 FACT = 1.0

IF (L - EQ. 1) FACT = -2.0

IF (L - EQ. 1) FACT = -1.0

UY(K+M2) = UY(K+M2) + PNYVDFACT

450 IF (L - EQ. 4) L = 0

IF (LUNIT) 83,83,800

C

500 PZ = QSHRDELCT/MIND

PNV = -QSHR * S1DE*(NY-1)*DELCT/(2.0*MT*(20*MX-1))

L = 1

DO 520 I = 1,MX

J = (I-1)*(QNT-1)

GO TO (505,507),L

505 M = 3

L = 2

GO TO 510

507 M = 1

L = 1

510 CONTINUE

DO 520 K = 1,3

FACT = - 1.0

IF (K-M) 517,515,517

515 FACT = 2.0

IF (L - EQ. 1) FACT = 1.0

517 UY(J+K) = UY(J+K) - PNYVDFACT

UY(J+K-N1) = UY(J+K-N1) - PNYVDFACT

UY(J+K) = UY(J+K) + P2ABSIFACT

520 UZ(J+K-N1) = UZ(J+K-N1) - P2ABSIFACT

IF (LUNIT) 84,84,800

C

600 PZ = QSHRDELCT/MIND

PNV = -QSHR * S1DE*(NY-1)*DELCT/(2.0*MT*(20*MX-1))

K = 1

L = 0

DO 620 I = 1,NA

K = K + 1

L = L + 1

605 FACT = -1.0

IF (L - EQ. 2) FACT = 2.0

IF (L - EQ. NA) FACT = 1.0

UY(K) = UY(K) - PNYVDFACT

UZ(K) = UZ(K) + P2ABSIFACT

610 IF (L-3) 615,620,615

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

0274

0275 IF (L-4) 605,610,605

0276 FACT = -1.0

0277 IF (L - EQ. 2) FACT = 2.0

0278 IF (L - EQ. NA) FACT = 1.0

0279 UY(K) = UY(K) - PNYVDFACT

0280 UZ(K) = UZ(K) + P2ABSIFACT

0281 610 IF (L-3) 615,620,615

```

0282      615 FACT = -1.0
0283      IF (L - EQ. 1) FACT = 2.0
0284      IF (L - EQ. 1) FACT = 1.0
0285      UY(K+M2) = UY(K+M2) - PM*FACT
0286      UZ(K+M2) = UZ(K+M2) - PZ*ABS(FACT)
0287      620 IF (L - EQ. 4) L = 0
0288      IF (UNIT) 65.85,800

C
0289      C GENERATE TORSION LOADS ON CYLINDER
0290      650 IF (ITYP - 2) 800,660,720
0291      660 RAD1 = SORT(Y(1)*2+Z(1)*2)
0292      RAD2 = RAD1 + MT
0293      PA = TORQ/(2.*RAD1*MX+2.*RAD2*MX)*0.0(RAD1+RAD2)/2.0MX*4.1
0294      PB = RAD2/RAD1*PA
0295      DJ 690 I = 1,MX
0296      J = (I-1)*(4*MY-1)
0297      THETA = ATAN2(Z(J*K),Y(J*K))
0298      SINTH = SIN(THETA+1.57079)
0299      COSTH = COS(THETA+1.57079)
0300      GO TO 1685,686),L
0301      685 M = 3
0302      L = 2
0303      GO TO 687
0304      686 M = 1
0305      L = 1
0306      687 CONTINUE
0307      DO 690 K = 1,3
0308      FACT = PA
0309      IF (K - M) 689,688,689
0310      688 FACT = 2.0*PB
0311      UY(J*K) = UY(J*K) + COSTH*FACT
0312      UZ(J*K) = UZ(J*K) + SINTH*FACT
0313      UY(J*K+M1) = UY(J*K+M1) - COSTH*FACT
0314      UZ(J*K+M1) = UZ(J*K+M1) - SINTH*FACT
0315      GO TO 800

C
0315      C GENERATE TORSION LOADS ON AIRFOIL
0316      720 MEND = 5*MY/4 + 1
0317      ARM = 0.0
0318      DO 750 I = 1,MX
0319      J = (I-1)*(4*MY-1)
0320      GO TO (745,746),L
0321      745 M = 3
0322      L = 2
0323      GO TO 747
0324      746 M = 1
0325      L = 1
0326      747 CONTINUE
0327      DO 750 K = 1,3
0328      FACT = 1.0
0329      IF (K - M) 749,748,749
0330      748 FACT = 2.0
0331      749 ANG = SLOPA(J*K)
0332      IF (Z(J*K) - LT. 0.0) ANG = ANG - 3.14159
0333      750 ARM = TORQ + (Z(J*K)*COS(ANG) - Y(J*K)*SIN(ANG))*FACT
0334      PA = TORQ/ARM/3.0
0335      TORQ = 0.0
0336      DO 760 I = 1,MX
0337      J = (I-1)*(4*MY-1)
0338      GO TO (755,756),L

```

PAGE 0008

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0338 755 M = 3
0339 L = 2
0340 GO TO 757
0341 756 M = 1
0342 L = 1
0343 757 CONTINUE
0344 DO 760 K = 1,3
0345 FACT = 1.0
0346 IF (K - M) 759,758,759
0347 758 FACT = 2.0
0348 759 ANG = SLOPA(J+K)
0349 IF (Z(J+K) - L, 0.0) ANG = ANG - 3.14159
0350 UY(J+K) = UY(J+K) + PACOS(ANG)*FACT
0351 UZ(J+K) = UZ(J+K) + PASIN(ANG)*FACT
0352 TORK = TORK + (Z(J+K)*UY(J+K) - Y(J+K)*UZ(J+K))*FACT
0353 UY(J+K+1) = UY(J+K+1) - PACOS(ANG)*FACT
0354 UZ(J+K+1) = UZ(J+K+1) - PASIN(ANG)*FACT
0355 760 WRITE (5,5020) I, TORK, ARM, PA
0356 800 CONTINUE
0357 IF (ITYP-2) 810,830,830
0358 810 NFIX = N2
0359 DO 820 I = 1,MY
0360 NFIX = NFIX + 4
0361 54 CODE(NFIX) = 7.0
0362 UY(NFIX) = 0.0
0363 UY(NFIX) = 0.0
0364 820 UZ(NFIX) = 0.0
0365 830 CONTINUE
0366 6020 FORMAT (110,6F15.5)

```

TAPE 3

PAGE 0009

DATE 71321

LEVEL 4

MODEL 44

FORTRAN IV

```

0367 850 WRITE (3) (UX(I),UY(I),UZ(I),I=1,MUMNP)
0368 IF (IHOLE -EQ- 0) GO TO 860
0369 CODE(IHOLE-1) = 7.0
0370 CODE(IHOLE) = 7.0
0371 CODE(IHOLE+1) = 7.0
0372 860 CONTINUE
0373 NUM = 4
0374 IF (NY -LT- 4) NUM = 0
0375 MOD1 = 4*NY + NUM
0376 CODE(MOD1) = 7.0
0377 CODE(MOD1+4) = 6.0
0378 CODE(MOD1+4*NY+1) = 3.0
0379 WRITE (3) (CODE(I),I=1,MUMNP)
0380 IF (ITEST -NE- 0)
0381 XWRITE(6,30007) ((I,CODE(I)),I=1,MUMNP)
0382 30007 FORMAT(18,F7.1)
0383 RETURN
0384 END

```

PAGE 0010

DATE 71321

LEVEL 4

MODEL 44

FORTRAN IV

SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / COMM	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
MUMNP	000000	NUMEL	000004	LOCATION	000004	SYMBOL	000008	SYMBOL	000010
NUMBLK	000014	NBN	000018	LOCATION	000018	NEL	00001C	MBAND	00001C

COMMON BLOCK / ONE / MAP SIZE 00008														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
HT	000004	THK	000008	THETA	00001C	YLOFF	000020	XC	000030	YLOFF	000040	YLOFF	000050	YLOFF
THY	000028	DUM	00002C	TC	000030	YLOFF	000040	YLOFF	000050	YLOFF	000060	YLOFF	000070	YLOFF
MX	000088	NY	00008C	ICNT	000090	YLOFF	0000A0	YLOFF	0000B0	YLOFF	0000C0	YLOFF	0000D0	YLOFF
IMOLE	0000CC	ITET	0000D0	ICNT	0000E0	YLOFF	0000F0	YLOFF	000100	YLOFF	000110	YLOFF	000120	YLOFF
COMMON BLOCK / LODA / MAP SIZE 00024														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
XLOAD	000004	XYLOD	000008	XQSHR	00001C	YQSHR	000020	YLOD	000030	YLOD	000040	YLOD	000050	YLOD
TORG	000014	XQSHR	000018	YQSHR	00002C	YLOD	000030	YLOD	000040	YLOD	000050	YLOD	000060	YLOD
COMMON BLOCK / LODS / MAP SIZE 0012C														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
UX	000080	UY	000090	UZ	0000A0	YLOD	0000B0	YLOD	0000C0	YLOD	0000D0	YLOD	0000E0	YLOD
COMMON BLOCK / DUMP / MAP SIZE 00004														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
ITEST	000000													
SCALAR MAP														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
N1	0003C0	N2	0003C4	NA	0003C8	Y2	0003CC	N4	0003E0	J	0003F4	PHK	000408	PHK
NYND	0003D0	XLGT	0003D8	YFIX	0003E0	M	000404	PRY	000418	PZ	00042C	SINTH	000440	SINTH
LUNIT	0003E4	PK	0003F0	PRYB	000414	PZ	00042C	SINTH	000440	TORK	000454			
L	0003F8	PRYA	000410	PRYB	000414	PZ	00042C	SINTH	000440	TORK	000454			
PRYA	00040C	PRYB	000410	PRYV	000420	PZ	00042C	SINTH	000440	TORK	000454			
PRYV	000420	PRYV	000424	PB	000438									
PA	000434	ARM	000448	ANG	000450									
ARM	000448	MOD1	00045C											
ARRAY MAP														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
Y	000464	Z	000468	SLOPA	00046C									
SUBPROGRAMS CALLED														
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
SQRT	000474	ATAN2	000478	SIN	00047C									
LABEL MAP														
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL
VERSION 3. LEVEL 4 DATE 71321														
40	0005A8	50	000602	20	00079E	25	0007C6	30	00081E	30	00081E	30	00081E	30
32	0008BE	35	0008EE	37	00091E	55	00091E	60	000938	60	000938	60	000938	60
62	000950	65	00096A	70	000984	75	000984	80	000988	80	000988	80	000988	80
82	0009C0	85	0009EA	84	000A04	85	000A1E	102	000A3E	102	000A3E	102	000A3E	102
103	000A8C	105	000A9A	106	000B14	107	000B2E	108	000B42	108	000B42	108	000B42	108
111	000B6A	112	000B8C	109	000B8A	110	000C40	120	000C86	120	000C86	120	000C86	120
130	000D10	140	000D24	150	000D9C	170	000DDE	180	000E70	180	000E70	180	000E70	180
185	000E8A	190	000E9E	192	000EC6	194	000EE8	200	000F26	200	000F26	200	000F26	200
210	001000	220	00105A	230	00106E	250	0010E6	270	001128	270	001128	270	001128	270
275	0011AE	280	001208	290	00121C	300	001294	320	0012D4	320	0012D4	320	0012D4	320
325	0012EE	326	001390	327	0013A4	328	0013BE	329	0013E4	329	0013E4	329	0013E4	329
331	001408	330	001446	335	001480	336	001514	337	00152E	337	00152E	337	00152E	337
338	001542	339	00156A	340	001576	341	001576	343	001638	343	001638	343	001638	343
344	001652	345	00168E	346	00168E	347	00169A	348	0016EA	348	0016EA	348	0016EA	348
350	001754	360	001766	370	00183C	375	001856	390	00186A	390	00186A	390	00186A	390
388	001892	390	001884	400	0018F2	410	00199A	415	0019F4	415	0019F4	415	0019F4	415

420	001A08	450	001A80	500	001AC2	505	001B8C	507	001B06
510	001BFA	515	001C12	517	001C34	520	001C55	600	001D60
605	001E20	610	001E8C	615	001EAO	620	001F52	650	001F94
640	001F82	645	002170	646	00218A	687	00219E	688	0021C2
689	0021D2	690	002280	720	0022EC	745	00236C	746	002386
747	00239A	748	0023C2	749	0023CE	750	002420	755	002520
756	00253A	757	00254E	758	002576	759	002582	760	002730
800	0027D8	810	0027F0	84	002814	820	00283C	830	00285A
6020	00285A	850	00286C	860	00291A	30007	002A58		

TOTAL MEMORY REQUIREMENTS 002804 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//STRESS EXEC FORTRAN(BCD,MAP)

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0001

0001	C	000000	000000	000000	OVERLAY 0	000000	000000	000000	000000
0002					SUBROUTINE STRESS				
0003					CALL LOAD('TETRA9')				
0004	C	000000	000000	000000	FORM STIFFNESS MATRIX				PSM1020
0005					CALL LOAD('TETRA10')				
0006					CALL STIFF				
0007					RETURN				
					END				

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0002

SUBPROGRAMS CALLED	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
STRESS	000004	000004	STIFF	000008		

TOTAL MEMORY REQUIREMENTS 000190 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//STRESS EXEC FORTRAN(BCD,MAP)

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0001

0001	C	000000	000000	000000	OVERLAY BA	000000	000000	000000	000000
0002					SUBROUTINE STRESS				PSM0010
0003					COMMON / CONN / NUMNP, NUMEL, NLOD, NLIN, MBAND, NUMBLK, NBN, NEL(20)				
0004					COMMON / ONE / SIDE, MT, THK(5), THETA, YLOFF, Y2OFF, TH1, DUM, TC(11)				
0005					1, XC(11), MED(12), NR, NY, IDPT, ITYP, IFACE, IMOLE, ITET, ICMT, NME(14)				
0006					COMMON / XYZ / X(300), Y(300), Z(300), UX(600), UY(600), UZ(600)				
0007					1, CODE(600), IDO(600), IELM				
0008					COMMON / ELM / S(18, 18), LM(6), T, ETR, EPRM, ELGT, G, ST(3, 18, 4), INTL				
0009					COMMON/RAX/NR4, NR4LCO, NR4STF, NR4DIA, NR4ECST, NR4ECLO, NR41, NR42				
0010					DIMENSION I0(300), N(1300), IM(300), ML(13), MM(6)				
0011					DIMENSION DUM1(600), DUM2(600), DUM3(600), DUM4(600), B(11)				
					EQUIVALENCE (LM(4), IS), (LM(13), IR), (LM(12), IQ), (LM(11), IP)				
					S, (MM(16), LU), (PMIS), LT), (MM(4), IS), (MM(13), LR), (MM(12), LG)				
					S, (MM(11), LP)				
					EQUIVALENCE (ML(13), JR), (ML(12), JQ), (ML(11), JP), (B(11), UX(11))				
					REWIND 3				TAPE 3


```

0064 260 CONTINUE
0065 ML(I) = KAS
0066 270 CONTINUE
0067 280 CONTINUE
0068 290 CONTINUE
0069 WR2=I
0070 WRITE(2,WR2)
      1 IDP,I,P,IQ,IR,IS,JP,JQ,ETR,ELGT,EPAM,G,T,INTL
0071 6000 FORMAT(1216)
0072 300 CONTINUE
0073 NODS = NUMP + NADD
0074 NRS=3*NODS
0075 DO 415 K = 1,NLOD
0076 MPRINT=0
0077 READ (3) (UX(I),UY(I),UZ(I),I=1,NUNMP)
0078 READ (3) (CODE(I),I=1,NUNMP)
      C RE-ORDER LOAD MATRIX TO ACCOUNT FOR ADDITIONAL NODES USING
      C RE-ORDER LOAD MATRIX TO INCLUDE ADDITIONAL NODES
0079 DO 350 I = 1,NODES
0080 DUM1(I) = 0.0
0081 DUM2(I) = 0.0
      TAPE 3
      TAPE 3
PORTNAM IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0003
0082 DUM3(I) = 0.0
0083 350 DUM4(I) = 0.0
0084 1*O(I)=ID(I)
0085 DUM1(I) = UX(I)
0086 DUM2(I) = UY(I)
0087 DUM3(I) = UZ(I)
0088 DUM4(I) = CODE(I)
0089 KAT = 1
0090 DO 400 I = 1,NUNMP
0091 DO 370 J = 1,NADD
0092 IF (ID(I)-IN(J)) 370,360,370
0093 360 KAT = KAT + 1
0094 100(KAT) = M(I,J)
0095 101 = 100(KAT)/1000
0096 102 = 100(KAT) - 101*1000
0097 IF (UX(ID1)) 362,365,362
0098 362 IF (UX(ID2)) 364,365,364
0099 364 IF (UX(ID1)/UX(ID2) -LE- 0.0) GO TO 365
0100 AL00 = ABS(UX(ID1))
0101 BLOO = ABS(UX(ID2))
0102 CLOO = AMIN1(AL00,BLOO)
0103 DUM1(KAT) = 4.*CLOO*UX(ID1)/AL00
0104 365 IF (UY(ID1)) 366,368,366
0105 366 IF (UY(ID2)) 367,368,367
0106 367 IF (UY(ID1)/UY(ID2) -LE- 0.0) GO TO 368
0107 AL00 = ABS(UY(ID1))
0108 BLOO = ABS(UY(ID2))
0109 CLOO = AMIN1(AL00,BLOO)
0110 DUM2(KAT) = 4.*CLOO*UY(ID1)/AL00
0111 368 IF (UZ(ID1)) 371,374,371
0112 371 IF (UZ(ID2)) 373,374,373
0113 373 IF (UZ(ID1)/UZ(ID2) -LE- 0.0) GO TO 374
0114 AL00 = ABS(UZ(ID1))
0115 BLOO = ABS(UZ(ID2))
0116 CLOO = AMIN1(AL00,BLOO)
0117 DUM3(KAT) = 4.*CLOO*UZ(ID1)/AL00
0118 374 CONTINUE
0119 370 CONTINUE

```

```

0120      IF (1-MUMNP) 380,390,380
0121      380 KAT = KAT + 1
0122      DO 100 KAT
0123      DUM1(KAT) = DUM1(1)
0124      DUM2(KAT) = DUM2(1)
0125      DUM3(KAT) = DUM3(1)
0126      DUM4(KAT) = DUM4(1)
0127      390 CONTINUE
0128      400 CONTINUE
0129      DO 410 I = 1, NNODES
0130      J=301

FORTRAN IV      MODEL 44 PS      VERSION 3, LEVEL 4      DATE 71321      PAGE 0004

0131      B1J-2J= DUM1(I)
0132      B1J-1J= DUM2(I)
0133      B1J= DUM3(I)
0134      IF (ICHT-GT-1) GO TO 410
0135      IF (PRINT-NE-0) GO TO 406
0136      PRINT=50
0137      WRITE (6,6045) K
0138      6045 FORMAT (1H1, 20X, 9HLOAD CASE, 14,/, 5X, 3HMEM, 2X, 3HOLD,/,
1 4X, 4HNODE, 2X, 4HMODE, 10X, 2HUX, 11X, 2HUT, 11X, 2HUX,/)
0139      406 CONTINUE
0140      PRINT=PRINT-1
0141      WRITE(6,6070) I, IDO(I), B1J-2J, B1J-1J, B1J
0142      6070 FORMAT(2I6, 3F15.4)
0143      410 CONTINUE
0144      DO 412 I = 1, NUNMP
0145      DUM1(I) = CODE(I)
0146      DO 413 I = 1, NNODES
0147      B1J CODE(I) = DUM4(I)
0148      415 CONTINUE
0149      DO 480 I = 1, NNODES
0150      417 IF (IDO(I) - 1000) 480, 480, 420
0151      420 KAL = IDO(I)/1000
0152      K2=IDO(I)- KAL*1000
0153      K7=0
0154      DO 470 J = 1, NUNMP
0155      IF (ID (J)-KAL) 440, 430, 440
0156      430 M1 = J
0157      GO TO 460
0158      440 IF (ID (J) - KAL) 460, 450, 460
0159      450 M2 = J
0160      460 CONTINUE
0161      470 CONTINUE
0162      COO1 = DUM1(M1)
0163      COO2 = DUM1(M2)
0164      CODE(I) = ANIM1(COO1, COO2)
0165      480 CONTINUE
0166      REWIND 3
0167      KK = 0
0168      PRINT=0
0169      DO 720 I = 1, NUNEL
0170      NR2=I
0171      READ(2, NR2)
1 10P, KP, KQ, KR, KS, JP, JB, ETR, ELST, EPRA, G, T, INFL
0172      IS = 0
0173      LU = 0
0174      TELM = 1
0175      DO 540 J = 1, NUNMP
0176      IF (EP-ID(J)) 510, 500, 510
0177      500 IP = J

```

TAPE 3

TAPE 2

```

0178      GO TO 550
0179      510 IF (KQ-ID(J)) 530,520,530
0180      520 IQ = J
0181
0182      530 IF (KQ-ID(J)) 545,540,545
0183      540 IR = J
0184
0185      GO TO 550
0186      545 IF (KS-ID(J)) 550,547,550
0187      547 IS = J
0188      IELA = 2
0189      550 CONTINUE
0190      560 CONTINUE
0191      DO 670 J = 1,MODES
0192      IF (KP-ID(J)) 580,570,580
0193      570 LP = J
0194      GO TO 660
0195      580 IF (KQ-ID(J)) 600,590,600
0196      590 LQ = J
0197      GO TO 660
0198      600 IF (KR-ID(J)) 620,610,620
0199      610 LR = J
0200      GO TO 660
0201      620 IF (JP-ID(J)) 640,630,640
0202      630 LS = J
0203      GO TO 660
0204      640 IF (JQ-ID(J)) 652,650,652
0205      650 LY = J
0206      GO TO 660
0207      652 IF (KS-ID(J)) 660,655,660
0208      655 LA = J
0209      660 CONTINUE
0210      LU = LT
0211      IF (IELM.EQ. 1) GO TO 675
0212      LT = LS
0213      LS = LA
0214      675 IF (IMTL.LE. 3) GO TO 690
0215      680 LS = LQ
0216      LT = LQ
0217      LU = LQ
0218      690 IF (MLIN-1) 692,695,695
0219      692 IF (ICVT-1) 695,695,697
0220      695 IF (IMPRINT) 170,165,170
0221      165 WRITE(6,6040)
0222      6040 FORMAT(10I10,
     1 2X,INJ,3X,IME,3X,IML,6X, 2MEI, 9 X, 2MEY, 10X, 3MEY,12X
     1 , 1MC, 9X, 1MT, 3X,23M LP LQ LR LS LY LU , / )
0223      170 NPRINT=NPRINT-1
0224

```

```

0225      WRITE (6,6050) IDP,IP,IQ,IR,IS,ELGT,ETR,EPHM,G,T,LP,LQ,LR,LS,LY,LU
0226      6050 FORMAT (16A14, 4F12.0,F10.4,6I4 )
0227      697 CONTINUE
0228      C
0229      NRI=1
0230      WRITE(11,NRI) IDP,KP,KQ,KR,KS,LM,MM,ETR,ELGT,EPHM,G,T,IMTL,IELM TAPE 1
0231      COMPUTE BAND WIDTH
0232      IF (LU.EQ. 0) LU = LY
0233      NMAX = MAX(ILP,LQ,LR,LS,LY,LU)
0234      NMIN = MIN(ILP,LQ,LR,LS,LY,LU)

```


COMMON BLOCK / RAX / MAP SIZE 000024			
SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000000	NR4L0D	000004
NRECST	000014	NRECLO	000018
		NR1	00001C
		NR2	000020
EQUIVALENCE DATA MAP			
SYMBOL	LOCATION	SYMBOL	LOCATION
NR	0002C0	LQ	0002C4
LT	0002D0	ML	0002D8
JR	0002E0		
SCALAR MAP			
SYMBOL	LOCATION	SYMBOL	LOCATION
NR4PE	0002E4	I	0002E8
IDP	0002F8	NADD	0002FC
KAS	00030C	KAT	000310
K	000320	J	000324
BLOD	000334	CLOD	000338
M2	000348	COD1	00034C
KQ	00035C	KR	000360
NHIM	000370	MOR1	000374
ARRAY MAP			
SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000380	NR1	00038C
DUM3	000450	DUM4	00045C
SUBPROGRAMS CALLED			
SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000370	NR1	000374
NR10	000372	NR10	000376
TOTAL MEMORY REQUIREMENTS 004E60 BYTES			
COMPILER HIGHEST SEVERITY CODE WAS 0			
//STIFF EXEC FORTRAN(BCO,MAP)			


```

0113      0      CONTINUE
0114      DO 9 K=NS,ME,NJ
0115      9      DIAG(3+1-3+K)=DIAG(3+1-3+K)+1.0E+12 +1.0 E+12
0116      2001 CONTINUE
0117      2000 CONTINUE
0118      NR4=NR4+DIA
0119      CALL OUTIN (1,DIAG ,MRS )
0120      IF (INTEST-EO-0) GO TO 1020
0121      WRITE(6,1019)
0122      1019 FORMAT(
0123      1020 CONTINUE
0124      WRITE (3) (IBOX(I),I=1,MUMNP)
0125      RETURN
0126      END

```

DIAG(I),I=1,MRS)
IN ,11E12.4)

TAPE 3

PAGE 0004

VERSION 3, LEVEL 4, DATE 71321

FORTRAN IV MODEL 44 PS

SYMBOL MUMNP MUMBLK	LOCATION 000000 000014	SYMBOL MUMBL MUM	COMMON BLOCK / COMM / MAP SIZE 00006C			LOCATION 00000C	SYMBOL MBAND	LOCATION 000010
			LOCATION 000004	SYMBOL MLOC	LOCATION 000008			
			000010	NEL	00001C			
COMMON BLOCK / XY2 / MAP SIZE 003CF4								
SYMBOL X UZ	LOCATION 000000 000000	SYMBOL Y CODE	LOCATION 000480 002A30	SYMBOL Z ID0	LOCATION 000960 003390	SYMBOL UX IELN	SYMBOL UY	LOCATION 001770
COMMON BLOCK / ELM / MAP SIZE 0008A0								
SYMBOL S ELGT	LOCATION 000000 000534	SYMBOL LM G	LOCATION 000510 000530	SYMBOL T ST	LOCATION 000528 00053C	SYMBOL ETR INIL	SYMBOL EPM	LOCATION 000530
COMMON BLOCK / MAX / MAP SIZE 000024								
SYMBOL MR4 MRECST	LOCATION 000000 000014	SYMBOL MRAIOD MRECLD	LOCATION 000004 000010	SYMBOL MRASTF MR1	LOCATION 000008 00001C	SYMBOL MRADIA MR2	SYMBOL MREC	LOCATION 000010
COMMON BLOCK / DUMP / MAP SIZE 000004								
SYMBOL MTEST	LOCATION 000000	SYMBOL	LOCATION 000000	SYMBOL	LOCATION 000000	SYMBOL	SYMBOL	LOCATION
SCALAR MAP								
SYMBOL MRS N RS LR4 L	LOCATION 000180 0001C4 0001D8 0001EC 000200	SYMBOL M IDP I1 K JJ	LOCATION 0001B4 0001C8 0001DC 0001F0 000200	SYMBOL ME KP I2 NS LL	LOCATION 0001B0 0001CC 0001E0 0001FA 000208	SYMBOL I KQ MMD II NJ	SYMBOL J KR MDO3 KK NC	LOCATION 0001C0 0001D4 0001E8 0001FC 000210
ARRAY MAP								
SYMBOL A	LOCATION 000214	SYMBOL DIAG	LOCATION 002C40	SYMBOL MM	LOCATION 004864	SYMBOL	SYMBOL	LOCATION
SUBPROGRAMS CALLED								
SYMBOL IBCOM8	LOCATION 00487C	SYMBOL OUTIN	LOCATION 004880	SYMBOL TRIN6	LOCATION 004884	SYMBOL	SYMBOL	LOCATION

LABEL	LOCATION	LABEL	LOCATION	LABEL MAP		LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
				51	62							
49	00495C	50	00498A	51	0049CE	62	004AFA	64	004B0C	64	004B0C	64
45	004B34	1007	004B3E	556	004BAC	557	004C26	6000	004C26	6000	004C26	6000
30009	004C34	400	004CEC	803	004DC8	805	004E2A	804	004E44	804	004E44	804
800	004E66	700	004E82	820	004ED2	1000	004EE8	1005	004F2C	1005	004F2C	1005
1008	004FF0	1010	005016	1011	005028	2002	005058	1	0050E0	1	0050E0	0050E0

FORTRAM IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0005

2	0050FA	3	005114	4	00512E	5	005148	6	00516A	6	00516A	6
7	005184	8	005198	9	0051A0	2001	005206	2000	005206	2000	005206	2000
1019	0052A4	1020	005286									

TO MEMORY REQUIREMENTS 005364 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//NID EXEC FORTRAM(BCO,MAP)

FORTRAM IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0001

0001 FUNCTION NID(KP,KQ)
0002 NID = MAX(KP,KQ)*1000 + MIN(KP,KQ)
0003 RETURN
0004 END

FORTRAM IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0002

SYMBOL NID	LOCATION	SYMBOL	LOCATION	EQUIVALENCE DATA MAP		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
				SCALAR MAP	SYMBOL						
SYMBOL	LOCATION	SYMBOL	LOCATION	0000DC	0000DC	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
KP	000008	KQ	000008								
SYMBOL	LOCATION	SYMBOL	LOCATION	0000E4	0000E4	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
MAX	0000E0	MINO	0000E0								

TOTAL MEMORY REQUIREMENTS 0001E0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//TRIM EXEC FORTRAM(BCO,MAP)

FORTRAM IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0001

0001 SUBROUTINE TRIM6
0002 COMMON / XYZ / X(300),Y(300),Z(300),UX(600),UY(600),UZ(600)
1 * CODE(600),IDU(600),IELM
COMMON / ELM / S(10,10),LM(6),T,ETR,EPRM,ELGT,6,ST(3,10,4),INTL
0003 DIMENSION PHY(3,6),PHY(3,6),ALAM(12,10),AL(3),ALZ(3),
0004 1 QMA(6,6),AST(10,10),ZETA(3),AKZ(13,10),DUM(6,6)
0005 DIMENSION STAI(14,14),STRI(3,10,4),STRA(3,12,4),IM(6)
C INITIALIZE STIFFNESS MATRIX
DO 450 I = 1,14
DO 450 J = 1,14
450 STAI(I,J) = 0.0
DO 475 I = 1,3
DO 475 J = 1,14

```

0011 DO 475 K = 1,4
0012 475 STR(I,J,K) = 0.0
0013 IJK = 1
0014 IF (IELM -EQ. 2) IJK = 2
0015 DO 500 JKL = 1,IJK
0016 DO 50 I=1,18
0017 DO 50 J=1,18
0018 50 S(I,J)=0.0
0019 IP = LM(1)
0020 I = LM(2)
0021 IR = LM(3)
0022 IF (JKL -EQ. 1) GO TO 80
0023 IP = LM(4)
0024 IO = LM(5)
0025 IR = LM(2)
0026 80 CONTINUE
0027 6000 FORMAT ( 615 )
C SET UP TRANSFORMATION INTO LOCAL COORDINATES
0028 XQP = X(IQ)-X(IP)
0029 YQP = Y(IQ)-Y(IP)
0030 ZQP = Z(IQ)-Z(IP)
0031 XRP = X(IR)-X(IP)
0032 YRP = Y(IR)-Y(IP)
0033 ZRP = Z(IR)-Z(IP)
0034 XQZ = X(IQ)-X(IQ)
0035 YQZ = Y(IR)-Y(IQ)
0036 ZQZ = Z(IR)-Z(IQ)
0037 IF (JKL -EQ. 2) GO TO 170
0038 D1= SORT(XQP,XRP,YQP,YRP,ZQP,ZRP)
0039 6100 FORMAT (6F 15.5)
0040 IF(D1)101,101,100
0041 100 AL(1)= XQP/D1
0042 AL(2)= YQP/D1
0043 AL(3)= ZQP/D1
0044 GO TO 102
0045 101 AL(1)=0.0
C
0046 AL(2)=0.0
0047 AL(3)=0.0
0048 102 RR= AL(1)*XRP +AL(2)*YRP +AL(3)*ZRP
0049 X2 = XRP -AL(1)*RR
0050 Y2 = YRP -AL(2)*RR
0051 Z2 = ZRP -AL(3)*RR
0052 D2 = SORT(X2,X2+Y2*Y2 +Z2*Z2)
0053 IF(D2)104,104,103
0054 103 AL2(1)=X2/D2
0055 AL2(2)=Y2/D2
0056 AL2(3)=Z2/D2
0057 GO TO 105
0058 104 AL2(1)=0.0
0059 AL2(2)=0.0
0060 AL2(3)=0.0
0061 105 CONTINUE
C SET UP TRANSFORMATION MATRIX
0062 DO 221 I=1,12
0063 DO 221 J=1,18
0064 221 ALAM(I,J)=0.0
0065 MT=0
0066 MOD3 = 10
0067 IF (IELM -EQ. 2) MOD3 = 12
0068 IF (IMTL .GT. 3) MOD3 = 9

```

PAGE 0002

DATE 71321

LEVEL 4

VERSION 3.

FORTRAM IV

```

0069 DO 222 J=1,MOD3,2
0070 DO 222 I=1,3
0071 KT=KT+1
0072 ALAM(J,KT)=AL2(I)
0073 C 222 ALAM(J+1,KT)=AL(I)
      TRANSFORM FROM GLOBAL TO LOCAL
0074 170 A1=(XQ*AL2(1)+YQ*AL2(2)+ZQ*AL2(3))*(-1,-1)
0075 B1=XQ*AL(1)+YQ*AL(2)+ZQ*AL(3)
0076 A2=(XQ*AL2(1)+YQ*AL2(2)+ZQ*AL2(3))*(-1,-1)
0077 B2=XQ*AL(1)+YQ*AL(2)+ZQ*AL(3)
0078 A3=XQ*AL2(1)+YQ*AL2(2)+ZQ*AL2(3)
0079 B3=XQ*AL(1)+YQ*AL(2)+ZQ*AL(3)
0080 AREA2=A2*B1-A1*B2
0081 A1=1.0/AREA2
0082 IF (IMTL-GT. 3) GO TO 500
      STIFFNESS AND STRESS MATRICES FOR 6 NODE TRIANGLE
      SET UP PHVX
0083 C 180 CONTINUE
0084 PHVX(2,1)=-81*AI
0085 PHVX(1,1)=-3.0*PHVX(2,1)
0086 PHVX(3,1)=PHVX(2,1)
0087 PHVX(1,2)=-82*AI
0088 PHVX(2,2)=-3.0*PHVX(1,2)
0089 PHVX(3,2)=PHVX(1,2)
0090 PHVX(1,3)=-83*AI
      PHVX(2,3)=PHVX(1,3)
      PHVX(3,3)=-3.0*PHVX(1,3)
      PHVX(1,4)=-4.0*PHVX(1,2)
0093 PHVX(2,4)=-4.0*PHVX(2,1)
0094 PHVX(3,4)=0.0
0095 PHVX(1,5)=0.0
0096 PHVX(2,5)=-4.0*PHVX(1,3)
0097 PHVX(3,5)=PHVX(1,4)
0098 PHVX(1,6)=PHVX(2,5)
0099 PHVX(2,6)=0.0
0100 PHVX(3,6)=PHVX(2,4)
      SET UP PHVY
0101 C
0102 PHVY(2,1)=-A1*AI
0103 PHVY(1,1)=-3.0*PHVY(2,1)
0104 PHVY(3,1)=PHVY(2,1)
0105 PHVY(1,2)=-A2*AI
0106 PHVY(2,2)=-3.0*PHVY(1,2)
0107 PHVY(3,2)=PHVY(1,2)
0108 PHVY(1,3)=-A3*AI
0109 PHVY(2,3)=PHVY(1,3)
0110 PHVY(3,3)=-3.0*PHVY(1,3)
0111 PHVY(1,4)=-4.0*PHVY(1,2)
0112 PHVY(2,4)=-4.0*PHVY(2,1)
0113 PHVY(3,4)=0.0
0114 PHVY(1,5)=0.0
0115 PHVY(2,5)=-4.0*PHVY(1,3)
0116 PHVY(3,5)=PHVY(1,4)
0117 PHVY(1,6)=PHVY(2,5)
0118 PHVY(2,6)=0.0
0119 PHVY(3,6)=PHVY(2,4)
      GENERATE STIFFNESS MATRIX IN LOCAL CO-ORDS.
0120 C
0121 DO 200 I=1,3
0122 DO 200 J=1,3
0123 200 QMAT(I,J)=T*AREA2/24.

```

PAGE 0003

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0124      201 QMAT(I,1)= 2.0*QMAT(I,1)
0125      CALL TRPRD (PHYX,QMAT,PHYX,DUM,ETR ,3,6)
0126      DO 202 I=1,6
0127      DO 202 J=1,6
0128      202 S(I,J) = S(I,J) + DUM(I,J)
0129      CALL TRPRD (PHYX,QMAT,PHYX,DUM,G ,3,6)
0130      DO 205 I=1,6
0131      DO 205 J=1,6
0132      205 S(I,J) = S(I,J) + DUM(I,J)
0133      CALL TRPRD (PHYX,QMAT,PHYX,DUM,EPAN,3,6)
0134      DO 208 I=1,6
0135      DO 208 J=1,6
0136      208 S(I,J+6) = S(I,J+6) + DUM(I,J)
0137      CALL TRPRD (PHYX,QMAT,PHYX,DUM,G ,3,6)

FORTRAN IV      MODEL 44 PS      VERSION 3, LEVEL 4      DATE 71321      PAGE 0004

0138      DO 211 I=1,6
0139      DO 211 J=1,6
0140      211 S(I,J+6) = S(I,J+6) + DUM(I,J)
0141      CALL TRPRD (PHYX,QMAT,PHYX,DUM,ELGT,3,6)
0142      DO 214 I=1,6
0143      DO 214 J=1,6
0144      214 S(I+6,J+6) = S(I+6,J+6) + DUM(I,J)
0145      CALL TRPRD (PHYX,QMAT,PHYX,DUM,G ,3,6)
0146      DO 217 I=1,6
0147      DO 217 J=1,6
0148      217 S(I+6,J+6)=S(I+6,J+6) + DUM(I,J)
0149      C      COMPLETE LOCAL MATRIX USING SYMMETRY AND RE-ARRANGE ROWS & COLS
0150      KT= 0
0151      DO 218 I=2,12
0152      KT=KT+1
0153      DO 218 J=1,KT
0154      218 S(I,J) = S(I,J)
0155      DO 219 I=1,12
0156      AST(I,1)= S(I,1)
0157      AST(2,1)= S(7,1)
0158      AST(3,1)= S(2,1)
0159      AST(4,1)= S(8,1)
0160      AST(5,1)= S(3,1)
0161      AST(6,1)= S(9,1)
0162      AST(7,1)= S(4,1)
0163      AST(8,1)= S(10,1)
0164      AST(9,1)= S(5,1)
0165      AST(11,1)= S(6,1)
0166      AST(12,1)= S(12,1)
0167      219 AST(10,1)= S(11,1)
0168      DO 220 I=1,12
0169      S(I,1)= AST(I,1)
0170      S(I,2)= AST(I,7)
0171      S(I,3)= AST(I,2)
0172      S(I,4)= AST(I,8)
0173      S(I,5)= AST(I,3)
0174      S(I,6)= AST(I,9)
0175      S(I,7)= AST(I,4)
0176      S(I,8)= AST(I,10)
0177      S(I,9)= AST(I,5)
0178      S(I,11)= AST(I,6)
0179      S(I,12)= AST(I,12)
0180      220 S(I,10)= AST(I,11)
0181      C      ELIMINATE ROWS AND COLS CORRESPONDING TO MODE 6
0182      6010 FORMAT ( 12F10.0)
0183      DEF = S(1,11)*S(12,12) - S(11,12)*S(12,11)

```

```

0182 IF (DET) 230,240,230
0183 230 AST(1,1) = S(12,12)/DET
0184 AST(2,2) = S(11,11)/DET

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0005

0185 AST(1,2) = S(11,12)/DET*(-1.)
0186 AST(2,1) = AST(1,2)
0187 DO 501 I = 1,2
0188 DO 501 J = 1,10
0189 AK21(I,J) = 0.0
0190 DO 501 K = 1,2
0191 501 AK21(I,J) = AK21(I,J) + AST(I,K)*S(K*10,J)
0192 DO 503 I=1,10
0193 DO 503 J=1,10
0194 AST(I,J)=0.0
0195 DO 503 K=1,2
0196 503 AST(I,J)=AST(I,J)+ S(I,K*10)*AK21(K,J)
0197 DO 504 I=1,10
0198 DO 504 J=1,10
0199 504 S(I,J) = S(I,J) - AST(I,J)
0200 6500 FORMAT ( 12F10.0 )
C SET UP APPROPRIATE STRESS MATRICES
0201 240 ILK = 1
0202 IKL = 4
0203 IF ( IELM.EQ.1 ) GO TO 250
0204 IF ( JKL.EQ.1 ) IKL= 3
0205 IF ( IJKL.EQ.2 ) IKL= 2
0206 250 DO 300 L = IKL,IKL
0207 DO 301 I=1,3
0208 301 ZETA(I)=0.0
0209 IF ( JKL.EQ.2 ) GO TO 302
0210 GO TO ( 303,304,307,308 ), L
0211 302 GO TO ( 308,307,304,303 ), L
0212 303 ZETA(I)=1.0
0213 GO TO 305
0214 304 ZETA(I)=1.0
0215 GO TO 305
0216 307 ZETA(I)=1.0
0217 GO TO 305
0218 308 DO 309 I=1,3
0219 309 ZETA(I)=1.0/3.0
0220 305 CONTINUE
0221 DO 310 I = 1,3
0222 DO 310 J = 1,18
0223 310 ST(I,J,L) = 0.0
0224 ST(1,1,L) = (4.*ZETA(1)-1.)*B1*AI
0225 ST(1,2,L) = (4.*ZETA(2)-1.)*B2*AI
0226 ST(1,3,L) = (4.*ZETA(3)-1.)*B3*AI
0227 ST(1,4,L) = 4.*ZETA(2)*B1+ZETA(1)*B2*AI
0228 ST(1,5,L) = 4.*ZETA(3)*B2+ZETA(2)*B3*AI
0229 ST(1,6,L) = 4.*ZETA(1)*B3+ZETA(3)*B1*AI
0230 ST(2,7,L) = (4.*ZETA(1)-1.)*A1*AI
0231 ST(2,8,L) = (4.*ZETA(2)-1.)*A2*AI
0232 ST(2,9,L) = (4.*ZETA(3)-1.)*A3*AI
0233 ST(2,10,L) = 4.*ZETA(2)*A1+ZETA(1)*A2*AI
0234 ST(2,11,L) = 4.*ZETA(3)*A2+ZETA(2)*A3*AI
0235 ST(2,12,L) = 4.*ZETA(1)*A3+ZETA(3)*A1*AI
0236 DO 410 I = 1,6
0237 ST(3,1,L) = ST(2,1+6,L)
0238 410 ST(3,1+6,L) = ST(1,1,L)
C REARRANGE STRESS MATRIX
DO 416 I=1,3
0239

```

```

0240      AST(I,1) = ST(I,1,L)
0241      AST(I,2) = ST(I,7,L)
0242      AST(I,3) = ST(I,2,L)
0243      AST(I,4) = ST(I,8,L)
0244      AST(I,5) = ST(I,3,L)
0245      AST(I,6) = ST(I,9,L)
0246      AST(I,7) = ST(I,4,L)
0247      AST(I,8) = ST(I,10,L)
0248      AST(I,9) = ST(I,5,L)
0249      AST(I,10) = ST(I,11,L)
0250      AST(I,11) = ST(I,6,L)
0251      AST(I,12) = ST(I,12,L)
0252      C      ELIMINATE NODE 6 FROM STRESS MATRIX
0253      DO 601 I=1,3
0254      DO 601 J=1,10
0255      ST(I,J,L) = 0.0
0256      DO 601 K=1,2
0257      601 ST(I,J,L) = ST(I,J,L) + AST(I,K+10)*AK2(I,K,J)
0258      DO 602 I=1,3
0259      602 AST(I,J) = AST(I,J) - ST(I,J,L)
0260      DO 605 I = 1,3
0261      DO 605 J = 1,10
0262      605 ST(I,J,L) = AST(I,J)
0263      300 CONTINUE
0264      IF (IELM.EQ. 1) GO TO 500
0265      C      ADD TWO TRIANGLES TO GET A SEVEN NODE QUADRILATERAL
0266      IF (JKL.EQ. 2) GO TO 480
0267      IK(1) = 1
0268      IK(2) = 3
0269      IK(3) = 2
0270      IK(4) = 5
0271      IK(5) = 7
0272      GO TO 470
0273      460 IK(1) = 4
0274      IK(2) = 2
0275      IK(3) = 3
0276      IK(4) = 6
0277      IK(5) = 7
0278      470 DO 480 K = 1,K,IKL
0279      DO 480 I = 1,5
0279      K3 = IK(I)*2 - 2
0280      DO 480 L = 1,2
0281      DO 430 J = 1,3
0282      480 STR(J,K3+L,K) = STR(J,K3+L,K) + ST(J,201-2*L,K)
0283      DO 490 I = 1,5
0284      DO 490 J = 1,5
0285      K3 = IK(I)*2 - 2
0286      K4 = IK(J)*2 - 2
0287      DO 490 L = 1,2
0288      DO 490 M = 1,2
0289      490 STAIK3+L,K4+M) = STAIK3+L,K4+M) + S(201-2*L,20J-2+M)
0290      500 CONTINUE
0291      IF (IMTL.EQ. 3) GO TO 700
0292      IF (IELM.EQ. 1) GO TO 560
0293      DO 495 I = 1,14
0294      495 WRITE (6,5000) (STAI(I,J),J=1,14)
0295      C      REMOVE NODE SEVEN FROM QUAD
0296      DET = STAI(13,13)*STAI(4,14) - STAI(13,14)*STAI(4,13)
0297      IF (ABS(DET).LT. 1.0E-10) WRITE (6,6200)
0298      6200 FORMAT (/// '5X, 26REDUCED MATRIX IS SINGULAR' )

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0007

```

0298 IF (ABS(DET) - LT. 1.0 E-10) DET = 1.0
0299 QMAT(1,1) = STA(1,14)/DET
0300 QMAT(2,2) = STA(13,13)/DET
0301 QMAT(1,2) = STA(13,14)/DET*(-1.)
0302 QMAT(2,1) = QMAT(1,2)
0303 DO 510 I = 1,2
0304 DO 510 J = 1,12
0305 AK21(I,J) = 0.0
0306 DO 510 IJ = 1,2
0307 510 AK21(I,J) = AK21(I,J) + QMAT(I,IJ)*STA(IJ+12,J)
0308 DO 520 I = 1,12
0309 DO 520 J = 1,12
0310 AST(I,J) = 0.0
0311 DO 520 IJ = 1,2
0312 520 AST(I,J) = AST(I,J) + STA(I,IJ+12)*AK21(IJ,J)
0313 DO 530 I = 1,12
0314 DO 530 J = 1,12
0315 530 SII(J) = STA(I,J) - AST(I,J)
0316 DO 550 K = 1,4
0317 DO 540 I = 1,3
0318 DO 540 J = 1,12
0319 STRA(I,J,K) = 0.0
0320 DO 540 IJ = 1,2
0321 540 STRA(I,J,K) = STRA(I,J,K) + STA(I,IJ+12,K)*AK21(IJ,J)
0322 DO 550 I = 1,3
0323 DO 550 J = 1,12
0324 550 ST(I,J,K) = STRA(I,J,K) - STRA(I,J,K)
0325 DO 555 K = 2,3
0326 DO 555 I = 1,3

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0000

0327 DO 555 J = 1,12
0328 555 ST(I,J,K) = ST(I,J,K)/2.0
0329 DO 556 I = 1,12
0330 556 WRITE (6,6500) (SII(J),J=1,12)
C GENERATE STIFFNESS MATRIX IN GLOBAL CO-ORDINATES
0331 560 CALL TRPRD (ALAM,S,ALAM,S,1.0,12,18)
C TRANSFORM STRESS MATRIX INTO GLOBAL CO-ORDINATES
0332 DO 417 L = 1,4
0333 DO 415 I = 1,3
0334 DO 415 J = 1,12
0335 415 AST(I,J) = ST(I,J,L)
0336 DO 417 I=1,3
0337 DO 417 J=1,18
0338 ST(I,J,L) = 0.0
0339 DO 417 K=1,12
0340 417 ST(I,J,L) = ST(I,J,L) + AST(I,K)*ALAM(K,J)
0341 RETURN
C STIFFNESS AND STRESS MATRICES FOR 3 NODE TRIANGLE
0342 700 CONTINUE
0343 DO 750 I = 1,3
0344 DO 750 J = 1,6
0345 QMAT(I,J) = 0.0
0346 750 PHX(I,J) = 0.0
0347 PHX(1,1) = B1
0348 PHX(1,3) = B3
0349 PHX(1,5) = B2
0350 PHX(2,2) = A1
0351 PHX(2,4) = A3
0352 PHX(2,6) = A2
0353 PHX(3,1) = A1
0354 PHX(3,2) = B1

```

```

0355 PHX(3,3) = A3
0356 PHX(3,4) = B3
0357 PHX(3,5) = A2
0358 PHX(3,6) = B2
0359 DO 760 I = 1,3
0360 DO 760 J = 1,6
0361 QMAT(1,1) = 0.0
0362 PHX(1,1) = PHX(1,1)/AREA2
0363
0364 760 CONTINUE
0365 DO 765 I = 1,3
0366 DO 765 J = 1,9
0367 ST(1,1) = 0.0
0368 DO 765 K = 1,6
0369 ST(1,1) = ST(1,1) + PHX(1,1)*ALAN(K,J)
0370 QMAT(1,1) = ETR
0371 QMAT(2,2) = ELGT
0372 QMAT(1,2) = EPRM
0373 QMAT(2,1) = EPRM

```

PAGE 0009

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321

```

0373 QMAT(3,3) = G
0374 W = 1/AREA2/2.
0375 CALL TAPROD (PHX,QMAT,PHX,DUM,N,3,6)
0376 DO 780 I = 1,9
0377 DO 780 J = 1,6
0378 AST(1,1) = 0.0
0379 DO 780 K = 1,6
0380 AST(1,1) = AST(1,1) + ALAN(K,1)*DUM(K,J)
0381 DO 790 I = 1,9
0382 DO 790 J = 1,9
0383 ST(1,1) = 0.0
0384 DO 790 K = 1,6
0385 ST(1,1) = ST(1,1) + AST(1,1)*ALAN(K,J)
0386 RETURN
0387 END

```

SYMBOL	LOCATION	SYMBOL	COMMON BLOCK / XYZ		LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL			SYMBOL					
X	000000	Y	000480	Z	000960		UK	000E10	UY	001770		
UZ	002000	CODE	002A30	IDO	003390		IELM	003CF0				
SYMBOL	LOCATION	SYMBOL	COMMON BLOCK / ELM		LOCATION	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL			SYMBOL					
S	000000	LM	000510	T	000528		ETR	00052C	EPRM	000530		
ELGT	000534	G	000538	ST	00053C		IMTL	00089C				
SYMBOL	LOCATION	SYMBOL	SCALAR MAP		LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
			LOCATION	SYMBOL								
I	000328	J	00032C	K	000330		ILK	000334	JKL	000338		
IP	00033C	IQ	000340	IR	000344		XQP	000348	YQP	00034C		
ZQP	000350	XRP	000354	YRP	000358		ZRP	000360	XRO	000364		
VRQ	000364	ZRQ	000368	D1	00036C		RR	000370	X2	000374		
VZ	000378	Z2	00037C	D2	000380		KT	000384	NCD3	000388		
AI	00038C	B1	000390	A2	000394		B2	000398	A3	00039C		
B3	0003A0	AREAZ	0003A4	A1	0003A8		DET	0003AC	ILK	0003B0		
IKL	0003B4	L	0003B8	K3	0003BC		K4	0003C0	M	0003C4		
IJ	0003C8	M	0003CC									
SYMBOL	LOCATION	SYMBOL	ARRAY MAP		LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL
			LOCATION	SYMBOL								
PHX	000300	PHYV	000418	ALAM	000460		AL	0007C0	AL2	0007CC		
OMAT	0007D8	AST	000868	ZETA	000878		AK21	000884	DUM	000ESC		
STA	000EEC	STRA	0011FC		00149C		IK	00160C				

SYMBOL	LOCATION	SYMBOL	SUBPROGRAMS CALLED		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL						
TRPRD	0016F4	IBCONB	0016F8	SORT	0016FC					
LABEL MAP										
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LOCATION
450	00184A	475	0018AC	50	001958	80	0019E0	6000	0019EC	
6100	001AC6	100	001AED	101	00180A	102	001826	103	0018BE	
104	0018E8	105	001C04	221	001C1C	222	001CFO	170	001D1C	
180	001E28	200	001FC8	201	00201E	202	002074	205	0020F6	
208	002178	211	0021F6	214	002274	217	0022F2	218	002376	
219	00244C	220	0024DA	6010	0024F4	230	002526	501	00259A	
503	00265C	504	0026F4	6500	002742	240	002750	250	0027A6	
301	0027BA	302	002824	303	002858	304	00286A	307	00287C	
308	00288E	309	00289A	305	0028C4	310	0028DC	410	002AA2	
416	00283E	601	0028A4	602	002C46	605	002CC0	300	002D22	
460	002D86	470	002D86	480	002DDE	490	002F4A	500	003032	
495	003074	4200	003134	510	0031F4	520	0032B0	530	003348	
540	0033FC	550	00344C	555	003572	556	0035E0	560	00365A	

PAGE 0011

DATE 71321

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4

0011

003814

750

0037DE

700

003736

417

0036BC

765

003964

780

003A68

TOTAL MEMORY REQUIREMENTS 003C78 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0

//TRPRD EXEC FORTRAN(BCD,MAP)

```

0001 SUBROUTINE TRPRD (A,B,C,E,FACT,M1,M2)
0002 DIMENSION A(M1,M2),B(M2,M2),C(M1,M2),D(18,18),E(M2,M2)
0003 C
0004 A SUBROUTINE TO PERFORM A-TRANS * B * C MULTIPLICATION
0005 RETURN SOLUTION IN E
0006 DO 100 I=1,M2
0007 DO 100 J=1,M1
0008 D(I,J)=0.0
0009 DO 100 K=1,M1
0010 D(I,J)=D(I,J)+A(K,I)*B(K,J)
0011 DO 200 K=1,M1
0012 D(I,J)=E(I,J)+D(I,K)*C(K,J)*FACT
0013 RETURN
0014 END

```

PAGE 0002

DATE 71321

VERSION 3, LEVEL 4

PS

MODEL 44

FORTRAN IV

SYMBOL	LOCATION	SYMBOL	SCALAR MAP		SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL						
M1	0000E8	N2	0000EC	I	0000F0	J	0000FA	K	0000FB	
FACT	0000FC									
ARRAY MAP										
A	000100	B	000104	C	000108	D	00010C	E	00011C	

LABEL	LOCATION	LABEL	LOCATION	LABEL MAP	LABEL	LOCATION	LABEL	LOCATION	PAGE
100	000798	200	00087A						0001
TOTAL MEMORY REQUIREMENTS 0009A4 BYTES									
COMPILER HIGHEST SEVERITY CODE WAS 0									
//SOLPAC EXEC FORTMAN(BCD,MAP)									
FORTMAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
C ***** OVERLAY C *****									
0001				SUBROUTINE SOLPAC					
0002				COMMON / COMN / NUMP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NON,MEL(20)					
0003				COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2					
0004				COMMON /DUMP / ITEST					
0005				DIMENSION DIAG(1800),AI(2700),AJ(2700),BI(800),XI(800)					
0006				EQUIVALENCE (X(1),B(1),AJ(1))					
0007				REAL*8 DSUM,DA,DB					
0008				NRS=3*NUMP					
0009				NR4=NR4DIA					
0010				NRD=1					
0011				CALL OUTIN (2,DIAG ,NRS)					
0012				DIAG(NRD)= SQRT(DIAG(NRD))					
0013				NRD=1					
0014				M=MBAND-1					
0015				NE=3*M					
0016				DO 1010 II=1,NUMP					
0017				NR4=III-1)*NRECST*NR4STF					
0018				LR4=NR4					
0019				IFRED=0					
0020				IL=3*(II-1)+1					
0021				IF (IL.EQ.1) IFRED=1					
0022				IF (IL.EQ.1) IL=2					
0023				IR=3*II					
0024				CALL OUTIN (2,AI ,NE)					
0025				DO 100 I=IL,IR					
0026				IFRED=IFRED+1					
0027				IMM=I-M					
0028				IF (IMM)101,101,102					
0029				101 IMM=1					
0030				102 IM1=I-1					
0031				M1=MBAND-I+(IFRED-1)*M					
0032				INITAL=0					
0033				DO 200 J=IMM,IM1					
0034				JFRED=MOD(J-1,3)+1					
0035				M2=MBAND-J* (JFRED-1)*M					
0036				NRD=J					
0037				DSUM=0.0					
0038				JM1=J-1					
0039				IF (INITIAL .GT. 0) GO TO 150					
0040				JJ= (J+2) /3					
0041				NR4= (JJ-1) *NRECST*NR4STF					
0042				CALL OUTIN (2,AJ ,NE)					
0043				INITAL=1					
0044				CONTINUE					
0045				IF (JFRED .EQ. 3) INITIAL = 0					
0046				IF (II.NE.JJ.OR.JFRED.GE.IFRED) GO TO 170					
0047				JL = (JFRED-1)*M + 1					
0048				JR = JL + M-1					

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0002

```

0049      DO 160 L = JL,JR
0050      A(JL) = A(L)
0051      160 CONTINUE
0052      170 CONTINUE
0053      IF(IJM-JM1)201,201,202
0054      DO 300 K=IMM,JM1
0055      K1=K+NI
0056      K2=K+NZ
0057      DA=DBLE(AI(K1))
0058      DB=DBLE(AJ(K2))
0059      DSUM=DSUM+DA*DB
0060      300 CONTINUE
0061      202 CONTINUE
0062      JL=J+NI
0063      200 AI(J1)=(AI(J1)-SNGL(DSUM))/DIAG(NRD)
0064      NRD=I
0065      DSUM=0.0
0066      DO 400 K=IMM,IMI
0067      K1=NI+K
0068      DA=DBLE(AI(K1))
0069      DSUM=DSUM+DA*DA
0070      NRD=I
0071      C*****TEST POS. DEF. AND SINGULARITY*****
0072      IF(DIAG(NRD)-SNGL(DSUM))204,205,206
0073      204 NERR=1
0074      GO TO 207
0075      205 NERR=0
0076      WRITE(6,203) NRD,NERR,DIAG(NRD),DSUM
0077      FORMAT(1H0,4NRD8,15,5X,5NERR8,15,2E15.5)
0078      DIAG(NRD)= DABS(DSUM) +1.E+12
0079      206 NERR=1
0080      DIAG(NRD)=SQRT(DIAG(NRD))-SNGL(DSUM)
0081      100 CONTINUE
0082      NRD=LR4
0083      CALL OUTIN (1,AI ,NE )
0084      1010 CONTINUE
0085      C*****BEGIN FORWARD SUBSTITUTION*****
0086      DO 500 K=1,NLOD
0087      NR4= (K-1)*NRECLD+NR4LOD
0088      LR4= NR4
0089      NRD=1
0090      CALL OUTIN (2,B ,NR4 )
0091      X(1)=B(1)/DIAG(NRD)
0092      DO 6010 JJ=1,NUNP
0093      IL = 3*(JJ-1)+1
0094      JFRED=0
0095      IF ( IL-EQ.1 ) JFRED=1
0096      IF ( IL-EQ.1 ) IL = 2
0097      IF ( IL-EQ.1 ) IL = 3+JJ
0098      NR4=(JJ-1)*NRECLD+1
0099      CALL OUTIN (2,AI ,NE )
0100      DO 600 J=IL,IR
0101      JFRED=JFRED+1
0102      NI=MBAND-J+ (JFRED-1)*M
0103      NRD=J
0104      JNM=J-M
0105      IF(IJM)601,601,602
0106      601 JNM=1
0107      602 JNM=J-1
0108      DSUM=0.0
0109      DO 603 I=JNM,JM1

```

PORTAN IV	MODEL	MODEL	VERSION	LEVEL	DATE	PAGE
0108	DA=DBLE(IAT(I+NI))	0144	NI = 1 + (JFRED-1)*M	71921	0004	
0109	DB=DBLE(X(I))	0145	IF (I-(K-MBAND)) 802,802,805			
0110	DSUM=DSUM+DA*DB	0146	CONTINUE			
0111	603 CONTINUE	0147	DB=DBLE(B(K))			
0112	X(J)=X(J)-SGL(DSUM)	0148	DA=DBLE(IAT(MI+MBAND-K))			
0113	X(J)=X(J)/DIAG(NRD)	0149	DSUM=DSUM+DA*DB			
0114	600 CONTINUE	0150	801 CONTINUE			
0115	6010 CONTINUE	0151	802 CONTINUE			
0116	NR4=LR4	0152	B(I)=X(I)-SGL(DSUM)			
0117	CALL OUTIN (1,X ,NRS)	0153	B(I)=B(I)/DIAG(NRD)			
0118	500 CONTINUE	0154	NR4=LR4			
	*****BEGIN BACK SUBSTITUTION*****	0155	CALL OUTIN (1,B ,NRS)			
0119	DO 700 J=1,NLDD	0156	700 CONTINUE			
0120	NR4 = (J-1)*NRECLD+NR4LDD	0157	NEAR=1			
0121	LR4= NR4	0158	RETURN			
0122	NRD=NRS	0159	END			
0123	CALL OUTIN (2,X ,NRS)					
0124	B(NRS)=X(NRS)/DIAG(NRD)					
0125	NI=NRS-1					
0126	DO 800 L=1,NI					
0127	I=NRS-L					
0128	NRD=I					
0129	IPI=I+1					
0130	DSUM=0.0					
0131	IF (NRS-IPI) 802,803,803					
0132	CONTINUE					
0133	NUNP1=IPI/3					
0134	IF (3*NUNP1-LT-IPI) NUNP1=NUNP1+1					
0135	DO 801 LL=NUNP1,NUNP1					
0136	IL = 3*(LL-1)+1					
0137	IR = 3*LL					
0138	IF (LL.EQ.NUNP1) IL=IPI					
0139	JFRED=MOD(IL-1,3)					
0140	NR4=(LL-1)*NRECLD+1					
0141	CALL OUTIN (2,AI ,ME)					
0142	DO 804 K=IL,IR					
0143	JFRED=JFRED + 1					

SYMBOL NUMP NUMBLK	LOCATION 000000 000014	SYMBOL NUMEL MBN	LOCATION 000004 000018	COMMON BLOCK / COMN SYMBOL HLOD MEL	MAP SIZE LOCATION 000008 00001C	SYMBOL NLIV	LOCATION 00000C 000020	SYMBOL MBAND	LOCATION 000010
SYMBOL NR4 MRECS7	LOCATION 000000 000014	SYMBOL NR4LOD MRECLD	LOCATION 000004 000018	COMMON BLOCK / RAX SYMBOL NR4STF NR1	MAP SIZE LOCATION 000008 00001C	SYMBOL NR4O1A NR2	LOCATION 00000C 000020	SYMBOL MREC	LOCATION 000010
SYMBOL ITEST	LOCATION 000000	SYMBOL	LOCATION	COMMON BLOCK / DUMP SYMBOL	MAP SIZE LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SYMBOL X	LOCATION 000184	SYMBOL B	LOCATION 000184	EQUIVALENCE DATA MAP SYMBOL AJ	LOCATION 000184	SYMBOL	LOCATION	SYMBOL	LOCATION
SYMBOL NR5 LR4 IM4 JFRED JR JI NUMNP1	LOCATION 002884 0028C8 0028DC 0028F0 002C04 002C18 002C2C	SYMBOL NRD IFRED IN1 M2 L MERR LL	LOCATION 002888 0028CC 0028E0 0028F4 002C08 002C1C 002C30	SCALAR MAP SYMBOL H IL NL JM1 K JMM DSUM	LOCATION 00288C 0028D0 0028E4 0028F8 002C0C 002C20 002C38	SYMBOL NE IR INITAL JJ K1 NM1 DA	LOCATION 0028C0 0028D4 0028E8 0028FC 002C10 002C24 002C40	SYMBOL II J JL K2 IP1 DB	LOCATION 0028C4 0028D8 0028EC 002C00 002C14 002C28 002C48
SYMBOL DIAG	LOCATION 002C50	SYMBOL AI	LOCATION 004870	ARRAY MAP SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SYMBOL DUTIN	LOCATION 0072A0	SYMBOL IBCOM#	LOCATION 0072A4	SUBPROGRAMS CALLED SYMBOL SORT	LOCATION 0072A8	SYMBOL	LOCATION	SYMBOL	LOCATION
LABEL 101 201 204 100 600 804	LOCATION 007510 007700 007856 007932 007892 007E28	LABEL 102 300 205 1010 6010 801	LOCATION 007520 007768 00786C 00795A 0078A4 007E4A	LABEL MAP LABEL 150 202 207 601 500 802	LOCATION 007634 00777A 00787C 007ADA 0078D8 007E60	LABEL 160 200 203 602 803 800	LOCATION 0076DA 00778A 0078C4 007AEA 007C8A 007E8C	LABEL 170 400 206 603 805 700	LOCATION 0076EC 00780E 0078FC 007B58 007DE2 007E8C

TOTAL MEMORY REQUIREMENTS 007F60 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//DEFL EXEC FORTRAN(BCD,MAP)

165

```

0050 STRESS(1,2) = STRAIN(1,1)*E12      + STRAIN(1,2)*E11
0051 STRESS(1,3) = STRAIN(1,3)*E33
0052 122 CONTINUE
0053 CALL MARGIN (INTL,STRESS,XMS,FBKL)
0054 IF (INLN -GT. 0) CALL MODUL (INTL,10P,STRAIN,E11 ,E22
      1 ,E12 ,E33 )
0055 IF (ICNT -GT. 1 .AND. . 10PT .EQ. 1) GO TO 160
0056 IF (IMPRINT) 150,140,150
0057 140 WRITE (6,6120)
0058 6120 FORMAT (1H1, 9H ELEMENT , /, 21H NO 1 J K L, 12H
      1 , 7H SIGMA 1, 10X, 7H SIGMA 2, 12X, 6H TAU 12, 10X, 6H MARGIN
      1 , 10X, 8H LEG // )
0059 MPRINT = 10
0060 150 MPRINT-MPRINT-1
0061 150 WRITE(6,6130) 10P,KPO ,KQO ,KRO,KSD ,STRESS(1,2)
      1 ,STRESS(1,1),STRESS(1,3),XMS,INTL
0062 6130 FORMAT (1 / ,1X, 15,4F19.3,115)
0063 IF (INMOD-3) 160,160,152
0064 152 DO 155 1-2,4
0065 155 WRITE(6,6140) STRESS(1,2),STRESS(1,1),STRESS(1,3)
0066 6140 FORMAT(122X,3F19.3)
0067 160 CONTINUE
0068 300 CONTINUE
0069 MR2=1J
0070 WRITE(12,MR2)
      9 ,1,KPO ,KQO ,KRO,KSD,E22 ,E11 ,E12 ,E33      TAPE 2
0071 1 ,TO ,INTL
0072 310 CONTINUE
0073 IF (ICNT -GT. 1 .AND. . 10PT .EQ. 1) GO TO 320
0074 N = 3
0075 IF (IFACE .EQ. 1) N = 5
0076 WRITE (6,6150) (CMS(I),I=1,N)
0077 6150 FORMAT (1H1, 5X, 30H MARGIN OF SAFETY IN EACH LEG 0 , 5F10.5 )
0078 XLOAD = APPLD(1,12)
0079 XLOAD = APPLD(2,12)
0080 XLOAD = APPLD(3,12)
0081 XMOH = APPLD(4,12)
0082 XMOH = APPLD(5,12)
      XMOH = APPLD(6,12)
0083 TORQ = APPLD(7,12)
0084 XQSHR = APPLD(8,12)
0085 YQSHR = APPLD(9,12)
0086 YMS = XMS
0087 IF (YMS -GT. .95) YMS = .95
0088 XALOM = XLOAD/(1.0-YMS)**.5
0089 VALOM = XLOAD/(1.0-YMS)**.5
0090 XVALM = XVALD/(1.0-YMS)**.5
0091 XHALM = XMOH/(1.0-YMS)**.5
0092 XHALM = XMOH/(1.0-YMS)**.5
0093 XHALM = XMOH/(1.0-YMS)**.5
0094 XHALM = XMOH/(1.0-YMS)**.5
0095 XQSHR = XQSHR/(1.0-YMS)**.5
0096 YQSHR = YQSHR/(1.0-YMS)**.5
0097 WRITE (6,6160) XALOM,VALOM,XVALM,XHALM,YHALM,YVALM,TORQAL
      1 ,XQSHR,YQSHR
0098 6160 FORMAT (1 // , 20X,14MULTIMATE LOADS,/, 9X,2HMX, 10X, 2HMY, 9X,
      1 3HNY, 10X, 2HMX, 10X, 2HMY, 10X, 3HMY, 6X, 6HTORQUE, 6X,
      1 2HMX, 10X, 2HMY, / , 9F12.0 )
0099 Y1LGT = .28867 + Y1OFF
0100 Y2LGT = .86657 + Y2OFF

```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0003

```

0101 XLGT = ((NX-1)*2XLGT + YLGT)*SIDE
0102 YLGT = (FLOAT(NY)-5)*SIDE
      CALCULATE STIFFNESSES
0103 IF (ABS(XLOAD) .LT. .001) GO TO 340
0104 STRS = XLOAD/(INT*XLGT)
0105 IF (ITET .EQ. 1) GO TO 330
0106 J1 = (NX-1)
0107 J2 = (NX*NY*2 - NX - 1)
0108 YLT = YLGT
0109 GO TO 335
0110 J1 = (NY/2*(4*NY-1)+3)
0111 J2 = (NY/2*(4*NY-1))
0112 YLT = SIDE*(NY-1)
0113 DO 336 K = 1, NUMNP
0114 IF (IDOK) .EQ. J1) N1 = K*3 - 2
0115 IF (IDOK) .EQ. J2) N2 = K*3 - 2
0116 STRN = (-B(N1)*8(N2))/YLT
0117 XMOD = STRS/STRN
0118 WRITE (6,200) XMOD
0119 6200 FORMAT ( / / / , 5X, 26H MODULUS IN X DIRECTION 0, F15.5, 3HPS1 )
0120 340 IF (ABS(YLOAD) .LT. .001) GO TO 360
0121 STRS = YLOAD/(INT*YLGT)
0122 IF (ITET .EQ. 1) GO TO 350
0123 J1 = (NX*(NY-1)+1)
0124 J2 = NX*NY
0125 GO TO 355
0126 J1 = (2*NY-1)
0127 J2 = ((4*NY-1)*(NY-1)+2*NY-2)
0128 DO 356 K = 1, NUMNP
0129 IF (IDOK) .EQ. J1) N1 = K*3 - 1
0130 IF (IDOK) .EQ. J2) N2 = K*3 - 1
0131 STRN = (-B(N1)*8(N2))/XLGT
0132 XMOD = STRS/STRN
0133 WRITE (6,250) XMOD
0134 6250 FORMAT ( / / / , 5X, 26H MODULUS IN Y DIRECTION 0, F15.5, 3HPS1 )
0135 360 CONTINUE
0136 320 CONTINUE
0137 RETURN
0138 END

```

PAGE 0005

VERSION 3, LEVEL 4 DATE 71321

FORTRAN IV MODEL 44 PS

SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / COMN	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	SYMBOL	000008	MLOD	000008	MLIN	00000C
NUMBLK	000014	NBN	000018	NEL	00001C				
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / MATL	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
E	000000	FXT	000050	SYMBOL	000064	FYC	000078	FYC	00008C
SSS	0000A0	CMS	000084						
SYMBOL	LOCATION	SYMBOL	LOCATION	COMMON BLOCK / ONE	MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION
STDE	000000	MT	000004	SYMBOL	000008	THETA	00001C	YLOFF	000020
YZOFF	000024	TMT	000028	TMK	00002C	TC	000030	XC	00003C
NEO	000088	MX	000088	DUM	00008C	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IMOLE	0000CC	ITET	0000D0	ICNT	0000DA	NNEW	0000D8

COMMON BLOCK / LODA / MAP SIZE 000024									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
XLOAD	000004	XVLOD	000008	XHOM	00001C	YQSHR	000020	YHOM	000010
XTHOM	000014	TORQ	000018						
COMMON BLOCK / LODB / MAP SIZE 000160									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
APPLD	000000								
COMMON BLOCK / RAX / MAP SIZE 000024									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
MRA	000004	NR4STF	000008	NR4DIA	00001C	NR2	000020	NREC	000010
MRECST	000014								
SCALAR MAP									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
MODES	000100	MRS	0001C0	I2	0001C4	KPD	0001D8	MPRINT	0001C8
M	00010C	IJ	0001D4	LDP	0001E8	E22	0001EC	E11	0001DC
KQD	0001F0	K50	0001E4	TO	0001FC	NMOD	000200	0001F0	0001F0
E12	0001F4	E33	0001F8	KJ	000210	MOD3	000214	J	000218
I1	000208	JJ	00020C	KVALM	000220	YALOW	000228	KVALM	00022C
K	00021C	YMS	000220	XNAL	000234	TRQAL	00023C	XQSAL	000240
XMALM	000230	YMALM	000234	YLGCT	00024C	YLT	000250	YLGST	000254
YQSAL	000244	YLGST	000248	J2	000260	YMOD	000278	M1	000268
STRS	000258	J1	00025C	XMOD	000274				
M2	00026C	STRAN	000270						
ARRAY MAP									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
P	00027C	LM	0002C4	STRESS	0002DC	FBKL	00030C	ST	000320
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
NR	000680	STRAIN	000698	100	0006C8	8	000878		
SUBPROGRAMS CALLED									
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOMB	001998	BUKL	00199C	OUTIN	001990	MARGIN	001994	MODUL	001998
FRRPRB	00199C								
LABEL MAP									
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
80	00186E	6100	001894	90	0018F6	100	001C06	6110	001C6A
102	001C7C	110	001CA0	120	001CC0	123	001E46	122	001EF4
140	001F76	6120	001F94	150	002006	6130	002080	152	0020AA
155	002086	6140	002106	160	002118	300	002118	310	002194
6150	002254	6160	002490	330	0025CC	335	002640	336	002676
6200	002708	340	00273A	350	00279E	356	0027E6	356	00281C
6250	002850	360	0028E2	320	0028E2				
TOTAL MEMORY REQUIREMENTS 002974 BYTES									
COMPILER HIGHEST SEVERITY CODE WAS 0									
//MARGIN EXEC FORTRAN(BCO,MAP)									
FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321									
PAGE 0001									
SUBROUTINE MARGIN (INTL,STRESS,XMS,FBKL)									
COMMON / MATL / E(4,5),EXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CHS(10)									
DIMENSION LM(6),FBKL(5),STRESS(4,3)									
JK = 4									
IF (IML .GT. 3) JK = 1									
DO 190 J = 1,JK									

```

0007 IF (STRESS(J,2)) 110,120,120
0008 110 RX = -STRESS(J,2)/FXC(IMTL)
0009 F1 = FXC(IMTL)
0010 GO TO 130
0011 120 RX = STRESS(J,2)/FXT(IMTL)
0012 F1 = FXT(IMTL)
0013 130 CONTINUE
0014 IF (STRESS(J,1)) 140,150,150
0015 140 RY = -STRESS(J,1)/FYC(IMTL)
0016 F2 = FYC(IMTL)
0017 GO TO 160
0018 150 RY = STRESS(J,1)/FYT(IMTL)
0019 F2 = FYT(IMTL)
0020 160 CONTINUE
0021 RS = ABS(STRESS(J,3))/SSS(IMTL)
0022 R = RXRX - F2/FLOXRY + RYRY + RSRS
0023 IF (R .LT. 1.0 E-6) R = 1.0 E-6
0024 AMS = 1./SORT(R) - 1.
0025 IF (STRESS(J,2)) 170,180,170
0026 170 BMS = -FBKL(IMTL)/STRESS(J,2) - 1.0
0027 IF (STRESS(J,2) .GE. 0.0) BMS = 10.
0028 IF (BMS .LT. AMS) AMS = BMS
0029 180 IF (AMS .LT. CMS(IMTL)) CMS(IMTL) = AMS
0030 IF (AMS .LT. XMS) XMS = AMS
0031 190 CONTINUE
0032 200 CONTINUE
0033 RETURN
0034 END

```

PAGE 0002

DATE 71321

VERSION 3, LEVEL 4

MODEL 44 PS

FORTRAN IV

SYMBOL	LOCATION	SYMBOL	COMMON BLOCK / MATL		MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			FXC	FYT		FXC	000064	FXC	000078	FYC	00008C
SSS	000000 0000A0	FXC CMS									
SCALAR MAP											
JK	000110	SYMBOL	000114	000118	00012C	00013C	00011C	000130	000120	000134	000134
RY	000124	INTL	J	RS							
BMS	000138	F2									
ARRAY MAP											
LM	000140	SYMBOL	000158	00015C							
		FBKL	STRESS								
SUBPROGRAMS CALLED											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SYMBOL	000160	SYMBOL	000160	SYMBOL	000160	SYMBOL	000160	SYMBOL	000160	SYMBOL	000160
LABEL MAP											
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
110	0002AC	120	000280	130	0002AC	140	0002CC	150	000308	160	000308
160	00033C	170	0003E2	180	000440	190	000480	200	000496	210	000496

TOTAL MEMORY REQUIREMENTS 0004E4 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//MODUL EXEC FORTRAN(BCD,MAP)

0049
0049 A4 = E29MU12/MU
0050 210 CONTINUE
0051 RETURN
END

COMMON BLOCK / MODU / MAP SIZE 000550			
SYMBOL	LOCATION	SYMBOL	LOCATION
SIGT1	000000	SIGT2	000084
PRXT	000384	PRXC	000438
STC2	000528	STSS	00053C

SCALAR MAP			
SYMBOL	LOCATION	SYMBOL	LOCATION
A1	000118	STNMK	00011C
INTL	00012C	E1	000130
A3	000140	J	000144
MU	000154	A4	000158

ARRAY MAP			
SYMBOL	LOCATION	SYMBOL	LOCATION
STN	000160	STRAIN	00016C

SUBPROGRAMS CALLED

LABEL MAP			
LABEL	LOCATION	LABEL	LOCATION
100	000326	105	000368
135	000548	140	00057A
170	00068A	180	0006C6

TOTAL MEMORY REQUIREMENTS 000700 BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//PLST EXEC FORTRAN(BCD,MAP)

```

0001 FUNCTION PLST (SIG1,SIG2,STRN,DEL,J)
0002 PLST = 1.0
0003 IF (J-7) 100,120,120
0004 100 IF (SIG2) 120,120,110
0005 110 PLST = SIG1 + (SIG2-SIG1) * (ABS(STRN)-DEL*(J-1)) / DEL
0006 120 CONTINUE
0007 RETURN
0008 END
    
```

EQUIVALENCE DATA MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PLST	0000E0				

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
J	0000E4	SIG2	0000E8	STRN	0000F0

LABEL MAP

LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
100	0001E6	110	0001F2	120	00023A

TOTAL MEMORY REQUIREMENTS 00027C BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//BURL EXEC FORTRAN(BCD,MAP)

```

0001 SUBROUTINE BUKL (FBKL)
0002 COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
0003 COMMON / ONE / SIDE,HT,THK(5),TMEY,YLOFF,Y2OFF,TMT,DUM,TC(11)
0004 1 .XC(11),HED(12),NX,NY,IOP,T,ITYP,IFACE,IMOLE,ITET,ICAT,MNEW(4)
0005 DIMENSION FBKL(5),D(3),XL(5)
0006 N = 3
0007 IF (IFACE .EQ. 1) N = 5
0008 DO 500 I = 1,N
0009 XL(I) = HT
0010 XNU = 1. - E(4,1)*E(2,1)/E(1,1)*E(4,1)
0011 E12 = E(2,1)*E(4,1)/XNU
0012 TCUB = THK(1)*E3/12.
0013 D(1) = E(1,1)*TCUB/XNU
0014 D(2) = E(2,1)*TCUB/XNU
0015 D(3) = IE12 + E(3,1)*E2.1 *TCUB
0016 IF (IFACE .EQ. 1) GO TO 277
0017 IF (I-3) 280,280,277
0018 277 XNK = 2.*E(3,1)*E(5)/XL(1)*E2*(D(1)*D(2))*E5 + D(3)
0019 GO TO 345
0020 280 PHA = 3.14159*HT/SIDE
0021 GAM = 4.*E(3,1)*XNU/E(2,1) + E(4,1)
0022 DPHA = D(3)/D(2)*PHA
0023 IMO = 0
0024 ICOT = 0
0025 Z = 1.
0026 XNK = D(1)*E(PHA/XL(1))*E2 + 1.
0027 XFT = XNK/50.
0028 290 DPHB = (-D(1)*PHA*E2 + XNK*XL(1)*E2) / D(2)
0029 BETA = PHA*E5*(DPHA*E2+DPHB)*E5-DPHA)*E5
0030 ALPHA = PHA*E5*(DPHA*E2+DPHB)*E5*DPHA)*E5
0031 XA = BETA*(BETA*E2+PHA*E2+GAM)*ALPHA*E2-PHA*E2*PHA*E2*3)*TAN(1/ALPHA)
0032 XB = ALPHA*(ALPHA*E2-PHA*E2+GAM)*BETA*E2+PHA*E2*3)*TAN(1/BETA)
0033 ZL = Z
0034 Z = XA - XB
0035 SCN = Z/ZL
0036 IF (SCN) 292,292,300
0037 292 IF (IMO) 293,300,293
0038 293 IF (ICOT - 3) 295,295,345
0039 295 XFT = XFT/10.
0040 XNK = XNK
0041 Z = ZL
0042 ICOT = ICOT + 1
0043 300 XNKL = XNK
0044 XNK = XNK + XFT
0045 IMO = IMO + 1
0046 GO TO 290
0047 345 FBKL(1) = XNK/THK(1)
0048 500 CONTINUE
0049 IF (ICAT .GT. 1 .AND. IOP .EQ. 1) RETURN
0050 WRITE (6,6000) (FBKL(I),I=1,N)
0051 6000 FORMAT (1M1,21M LEG BUCKLING STRESS ,/, 5F10.0 )
0052 RETURN
0053 END

```

SYMBOL	LOCATION	SYMBOL	COMMON BLOCK / MATL		MAP SIZE	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
			LOCATION	SYMBOL		FAC					
E	000000	FXT	000050	FYT	000064		000078				
SSS	0000A0	CMS	0000B4								00008C
COMMON BLOCK / ONE											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SLIDE	000000	MT	000004	T4K	000008	THETA	00001C	VI OFF	000020	XC	00005C
Y2OFF	000024	TMT	000028	DUM	00002C	TC	000030	ITYP	0000C4	NNEW	0000D8
MED	000038	MX	000088	NY	0000BC	ICPT	0000C0				
IFACE	0000C8	IMOLE	0000CC	ITET	0000D0	ICNT	0000D4				
SCALAR MAP											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
N	000110	I	000114	XNU	000118	E12	00011C	TCUB	000120	IMO	000134
PNX	000124	PHA	000128	GAM	00012C	DPHA	000130	BETA	000148	SEN	00015C
TCCT	000138	Z	00013C	XFT	000140	DPHB	000144				
ALPA	00014C	XA	000150	XB	000154	ZL	000158				
MXL	000160										
ARRAY MAP											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PBL	000154	D	000168	XL	000174						
SUBPROGRAMS CALLED											
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCONB	000188	FXPRAB	00018C	TAMH	000190	TAM	000194				
LABEL MAP											
LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
277	000322	280	000368	290	0003FE	292	000558	293	000564	6000	000694
295	000578	300	0005A4	345	0005CE	500	0005F2				

TOTAL MEMORY REQUIREMENTS 000734 BYTES

COMPILER HIGH-EST SEVERITY CODE WAS 0
 //SYSAB1 ACCESS SDSABS,090-,"SYSRE7"
 // DELETE SDSABS1TETRA1,TETRA1,TETRA2,TETRA3,TETRA4,TETRA5,TETRA6)
 // DELETE SDSABS1TETRA7,TETRA8,TETRA9,TETRA10,TETRA11,TETRA12)
 // CONDENSE SDSABS
 // EXEC LINKUT(KEEP)

JC710280 LINKAGE EDIT

71/321

LIST PHASE TETRA,ROOT
 LIST INCLUDE TETRA,L
 LIST INCLUDE OUTIN,L
 LIST INCLUDE CURV,L
 LIST AUTOLINK DIOCS0
 LIST AUTOLINK LOAD
 LIST AUTOLINK IBCOMB
 LIST AUTOLINK ATAN2
 LIST AUTOLINK VDI0CS0
 LIST AUTOLINK UNITAB0
 LIST AUTOLINK USEROPT
 LIST PHASE TETRA1,0
 LIST INCLUDE GEN,L

```

LIST INCLUDE OPTIM.L
LIST INCLUDE WEIGHT.L
LIST INCLUDE DAR.L
LIST INCLUDE NONLIN.L
LIST AUTOLINK EXIT
LIST AUTOLINK FRAP0
LIST AUTOLINK SORT
LIST AUTOLINK FRAP0
LIST AUTOLINK ALOC
LIST AUTOLINK EXP
LIST PHASE TETRA2.0
LIST INCLUDE MODGEN.L
LIST AUTOLINK SIN
LIST PHASE TETRA3.TETRA2
LIST INCLUDE PLATGN.L
LIST PHASE TETRA4.TETRA2
LIST INCLUDE LODGEN.L
LIST AUTOLINK SORT
LIST AUTOLINK SIN
LIST PHASE TETRA5.TETRA2
LIST INCLUDE MODGAN.L
LIST AUTOLINK SIN
LIST AUTOLINK TAN
LIST PHASE TETRA6.TETRA2
LIST INCLUDE PLATGA.L
LIST PHASE TETRA7.TETRA2
LIST INCLUDE LODGAN.L
LIST AUTOLINK SORT
LIST AUTOLINK SIN
LIST PHASE TETRA8.TETRA1
LIST INCLUDE STRESS.L
LIST PHASE TETRA9.0
LIST INCLUDE STRESS.L
LIST INCLUDE MID.L
LIST AUTOLINK MAX0
LIST AUTOLINK ARIM1
LIST PHASE TETRA10.TETRA9
LIST INCLUDE STIFF.L
LIST INCLUDE TAIN6.L
LIST INCLUDE TAPRO.1

```

71/321

JC710200 LINKAGE EDIT

```

LIST AUTOLINK SORT
LIST PHASE TETRA11.TETRA1
LIST INCLUDE SOLPAC.L
LIST AUTOLINK SORT
LIST PHASE TETRA12.TETRA1
LIST INCLUDE DEFL.L
LIST INCLUDE MARGIN.L
LIST INCLUDE MODUL.L
LIST INCLUDE BURL.L
LIST INCLUDE PLST.L
LIST AUTOLINK FRAP0
LIST AUTOLINK SORT
LIST AUTOLINK TANH
LIST AUTOLINK TAN
LIST AUTOLINK ALOC
LIST AUTOLINK EXP
LIST ENTRY

```

CREATED

71/321	PHASE	TRANSFER ADDR.	LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
COMMON						COMMON	COMN	004100	00006C
COMMON						COMMON	MATL	004240	0000DC
COMMON						COMMON	LODA	004320	000024
COMMON						COMMON	DNE	004348	0000E8
COMMON						COMMON	OPTA	004430	000284
COMMON						COMMON	MOOU	0046E8	000350
COMMON						COMMON	LODB	004C38	000168
COMMON						COMMON	RAX	004DA0	000024
COMMON						COMMON	DUMP	004DC8	000004
COMMON						COMMON	PLAT	004DD0	0044E0
COMMON						COMMON	LODS	009280	0012C0
COMMON						COMMON	XYZ	00A570	003CF4
COMMON						COMMON	ELM	00E268	0008A0
ROOT	YETRA	00EB08	00EB08	012A5F	686	CSECT • ENTRY	MAIN446 MAIN446	00E808 00E808	00E808
						CSECT ENTRY	OUTING OUTIN	00EE60 00EE60	00EE60
						CSECT ENTRY	CURVE CURV	00F0D0 00F0D0	00F0D0
						CSECT ENTRY	BOAD10CS D10CS8	00F380 00F380	00F380
						CSECT ENTRY • ENTRY	BOADVLY LOAD LINK	00F670 00F688 00F678	00F670
						CSECT ENTRY ENTRY ENTRY • ENTRY	BOA18COM 18COM8 FIRSTIM FIXPOINT ADCOM8	00F788 00F788 0108E8 010880 00F844	00F788
						CSECT ENTRY • ENTRY	BOASATN2 ATAMZ ATAM	012110 012116 012130	012110
						CSECT	BOAF10CS	012288	012288
						ENTRY	VD10CS8	01299C	
						ENTRY	BUFORC8	012994	
						ENTRY	ACBORC8	012998	
						ENTRY	F10C08	0122C2	
						ENTRY	F10CS8	012288	

TETRA1	012A60	012A60	01707F	709	CSECT ENTRY	BOAUNITB UNITAB0	012A08 012A08	012A08	REL-FACTOR
					CSECT ENTRY	BOAUOPT USEROPT	012A58 012A58	012A58	
					CSECT ENTRY	GENS GEN	012A60 012A60	012A60	
					CSECT ENTRY	OPTMS OPTM	015280 015280	015280	
					CSECT ENTRY	WEIGHTS WEIGHT	016898 016898	016898	
					CSECT ENTRY	DARS DAR	016FA8 016FA8	016FA8	
					CSECT ENTRY	NONLINB NONLIN	017320 017320	017320	
					CSECT ENTRY	BOAFEXIT EXIT	0176A0 0176A6	0176A0	
					CSECT ENTRY	BOAFRIPI FRXPI0	0176C0 0176C8	0176C0	
					CSECT ENTRY	BOASSORT SORT	017778 01777E	017778	
					CSECT ENTRY	BOAFRIPI FRXPI0	017828 017830	017828	
					CSECT ENTRY	BOASLOG ALOG	017920 017942	017920	
					CSECT ENTRY	BOASEXP EXP	017A30 017A34	017A30	
					CSECT ENTRY	MODGENS MODGEN	017880 017880	017880	
TETRA2	017880	017880	018007	738	CSECT ENTRY	BOASSCH SIN	0187A0 0187C0	0187A0	REL-FACTOR
					CSECT ENTRY	BOASSCH COS	0187A0 0187A4	0187A0	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	LODCENS LODCEN	017880 017880	017880	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
					CSECT ENTRY	BOASSORT SORT	019A10 019A16	019A10	
TETRA3	017880	017880	019607	743	CSECT ENTRY	PLATMS LABEL	017880	017880	REL-FACTOR
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
TETRA4	017880	017880	019807	753	CSECT ENTRY	PLATMS LABEL	017880	017880	REL-FACTOR
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	
					CSECT ENTRY	PLATMS LABEL	017880	017880	

TETRA5	017800	018A37	765	CSECT ENTRY COS	BOASSCH SIN COS	019AC0 019AE0 019AC4	019AC0
				CSECT ENTRY	MODGANC MODGANC	017880 017880	017880
				CSECT ENTRY COS	BOASSCH SIN COS	0187F0 018810 0187F4	0187F0
				CSECT ENTRY COS	BOASTNCT TAN COTAN GTAN	018908 018926 01890E 018A34	018908
TETRA6	017800	01911F	771	CSECT ENTRY	PLATGAC PLATGA	017880 017880	017880
				CSECT ENTRY	LODGANC LODGAN	017880 017880	017880
TETRA7	017800	01A84F	779	CSECT ENTRY	BOASSORT SQRT	01A688 01A68E	01A688
				CSECT ENTRY COS	BOASSCH SIN COS	01A738 01A758 01A73C	01A738
TETRA8	012A60	012B8F	795	CSECT ENTRY	STRASSE STRASS	012A60 012A60	012A60
				CSECT ENTRY	STRASSE STRESS	012B8F 012B8F	012B8F
TETRA9	012B8F	017E4F	796	CSECT ENTRY	NID6 NID	017A50 017A50	017A50
				CSECT ENTRY COS	BOAFMAXI MAXO MINO AMINO AMAXO	017C30 017C36 017C3C 017CAB 017C82	017C30
TETRA10	012B8F	01C627	826	CSECT ENTRY	BOAFMAXI STIFFE STIFF	017D50 017D50 012B8F	017D50
				CSECT ENTRY	AMINI AMAXI MINI MAXI	017D56 017D56 017D56 017D56	017D56
TETRA10	012B8F	01C627	826	CSECT ENTRY	TRIM6E TRIM6	017F58 017F58	017F58
				CSECT ENTRY	TRIM6E TRIM6	017F58 017F58	017F58

TETRA11	012A60	012A60	01A66F	881	CSECT ENTRY	TRPRDS TRPAD	01B8D0 01B8D0	01B8D0	01B8D0
					CSECT ENTRY	BOASSORT SORT	01C578 01C57E	01C578	
TETRA12	012A60	012A60	01709F	927	CSECT ENTRY	SOLPACS SOLPAC	012A60 012A60	012A60	012A60
					CSECT ENTRY	BOASSORT SORT	01A9C0 01A9C6	01A9C0	
					CSECT ENTRY	DEFL DEFL	012A60 012A60	012A60	012A60
					CSECT ENTRY	MARGIN MARGIN	015308 015308	015308	
					CSECT ENTRY	MODULE MODUL	0158C0 0158C0	0158C0	0158C0
					CSECT ENTRY	BULK BULK	0160A0 0160A0	0160A0	
					CSECT ENTRY	PLST PLST	0167D8 0167D8	0167D8	0167D8
					CSECT ENTRY	BOAEXPR FRXPRB	016A58 016A60	016A58	
					CSECT ENTRY	BOASSORT SORT	016B50 016B56	016B50	016B50
					CSECT ENTRY	BOASTAM TAMH	016C00 016C06	016C00	
					CSECT ENTRY	BOASTMCT TAN	016D10 016D2E	016D10	016D10
					• ENTRY	COTAN	016D16	016D16	
					• ENTRY	OTAN	016E3C	016E3C	
					CSECT	BOA2LOG	016E40	016E40	
71/321	PHASE	TRANSFER ADDR.	LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
						ENTRY	ALOG	016E42	
					• ENTRY	ALOG10	016E48	016E48	
					CSECT	BOA2EXP	016F50	016F50	
					ENTRY	EXP	016F56	016F56	

LINKAGE EDITOR HIGHEST SEVERITY WAS 0
/E 142803

LITERATURE CITED

1. Tocher, J. L., AREA COORDINATES AND THEIR USE IN FINITE ELEMENT DERIVATIONS, Boeing Co. Internal Report MA-93. May 1967, pp 3-5, 17-19, 26-29.
2. Felippa, C. A., REFINED FINITE ELEMENT ANALYSIS OF LINEAR AND NONLINEAR STRUCTURES, Structures and Materials Report SESM 66-22. Dept. Civil Engineering, University of California at Berkeley, October 1966.
3. Przemieniecki, J. S., THEORY OF MATRIX STRUCTURAL ANALYSIS, McGraw-Hill, 1968, pp 62-64, 83-89, 433-437
4. Gellatly, R. A., Gallagher, R. H., and Lubieracki, W. A., DEVELOPMENT OF A PROCEDURE FOR AUTOMATED SYNTHESIS OF MINIMUM-WEIGHT STRUCTURES, AFFDL-TDR-64-141.
5. Petit, P. H., and Waddoups, M. E., A METHOD OF PREDICTING THE NONLINEAR BEHAVIOR OF LAMINATED COMPOSITES, J. Composite Materials, Vol. 3, Jan 1969, pp 2-19.
6. Lackmann, L. M., and Ault, R. M., MINIMUM-WEIGHT ANALYSIS OF FILAMENTARY COMPOSITE WIDE COLUMNS, J. Aircraft, Vol. 5, No. 2, March-April 1960, pp 184-190.
7. Timoshenko, S., and Gere, J. M., THEORY OF ELASTIC STABILITY, New York, McGraw-Hill 1961, pp 403-404.
8. Tsai, S. W., Adams, D. F., and Doner, D. R., ANALYSIS OF COMPOSITE STRUCTURES, NASA Report CR-620, November 1966, pp 3-5.

GLOSSARY

A	- Merged stiffness matrix
$\bar{A}$	- Triangularized stiffness matrix
B	- Matrix relating plate strains to plate nodal deflections
C_{11}, C_{22} , etc	- Plate elasticity matrix
D_1, D_2, D_3	- Plate bending stiffnesses
E_x	- Plate elastic modulus in fiber direction
E_y	- Plate elastic modulus transverse to fiber direction
F_{XT}	- Plate tension allowable stress in fiber direction
F_{XC}	- Plate compression allowable stress in fiber direction
F_{YT}	- Plate tension allowable stress transverse to fiber direction
F_{YC}	- Plate compression allowable stress transverse to fiber direction
G_{XY}	- Plate shear modulus
L_X	- Number of Vertical legs on X face
L_Y	- Number of Skew B legs on Y face
N_X	- Plate buckling load in X direction (lb/in.)
P_X	- Vertex nodal load for uniform load in X direction
P_Y	- Vertex nodal load for uniform load in Y direction
P_{XY}	- Vertex nodal load for uniform load in XY direction
P_{MX}	- Vertex nodal load for uniform moment in X direction

- P_{MY} - Vertex nodal load for uniform moment in Y direction
- P_{MXY} - Vertex nodal load for uniform moment in XY direction
- P_Z - Vertex nodal load in Z direction

APPENDIX I
ADDITIONAL GENERATION EXAMPLES

Additional examples of flat plate model generation are included in this appendix to clarify the examples given in the body of the report. The nodal numbering systems used for a 6 by 6 and a 10 x 6 true Tetra-core plate are shown in Figures 29 and 30, respectively. The nodal numbering systems used for a 4 x 6 and an 8 x 4 truncated Tetra-core plate are shown in Figures 31 and 32. Plate numbering systems used for the Vertical, Skew A, and Skew B legs in the truncated Tetra-core flat plate are shown in Figures 33, 34, and 35. The addition plates added to Skew A and Skew B legs to complete the cylinder model are shown in Figures 36 and 37. The nodal loads generated for a true Tetra-core flat plate model for each load type are shown in Figures 38, 39, 40, 41, 42, 43, 44, and 45. The nodal loads generated for a truncated Tetra-core flat plate model are shown in Figures 46, 47, 48, 49, and 50.

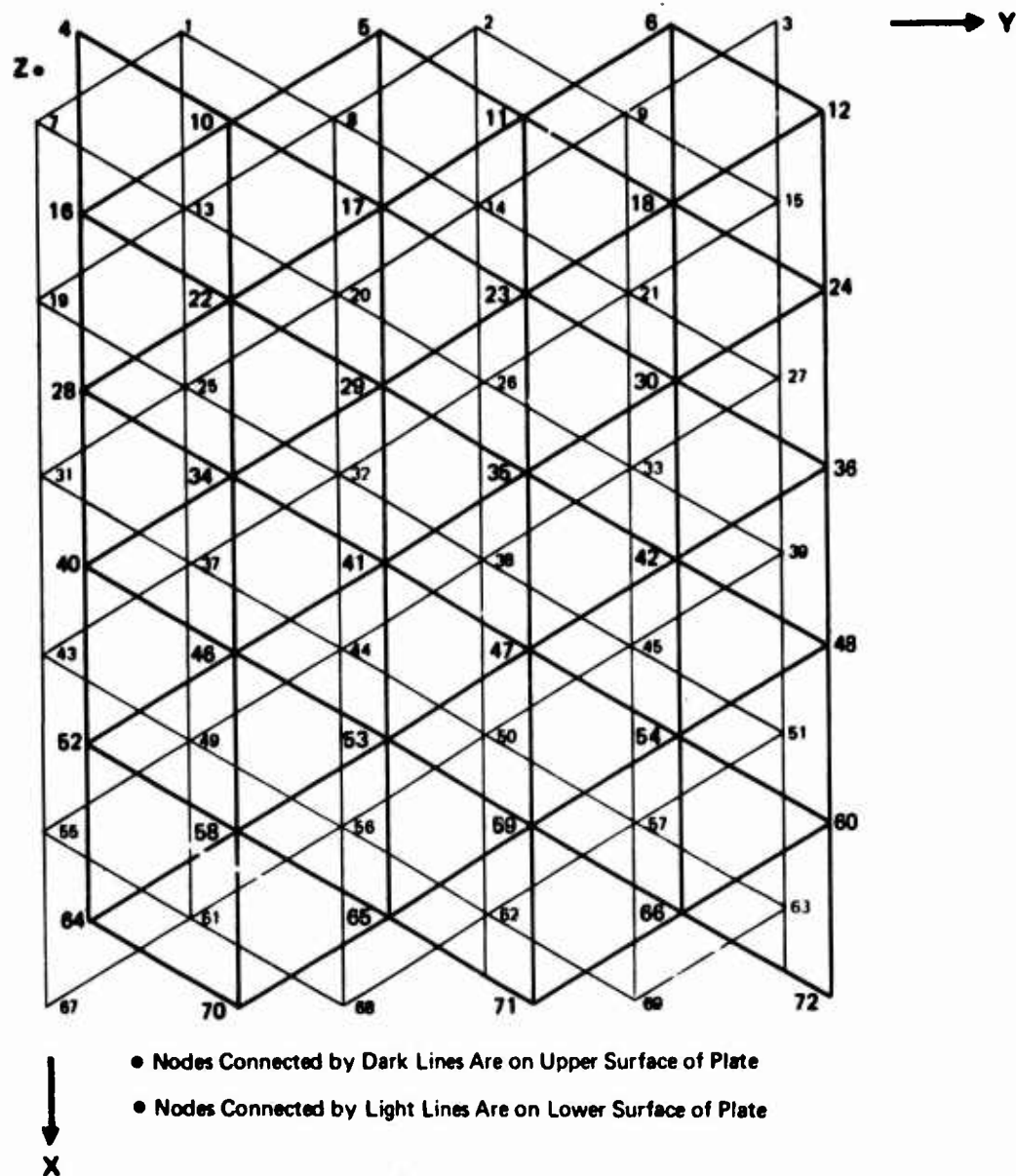


Figure 29. Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 6, L_Y = 6$.

Nodes Connected by Dark Lines Are on Upper Surface of Plate
Nodes Connected by Light Lines Are on Lower Surface of Plate

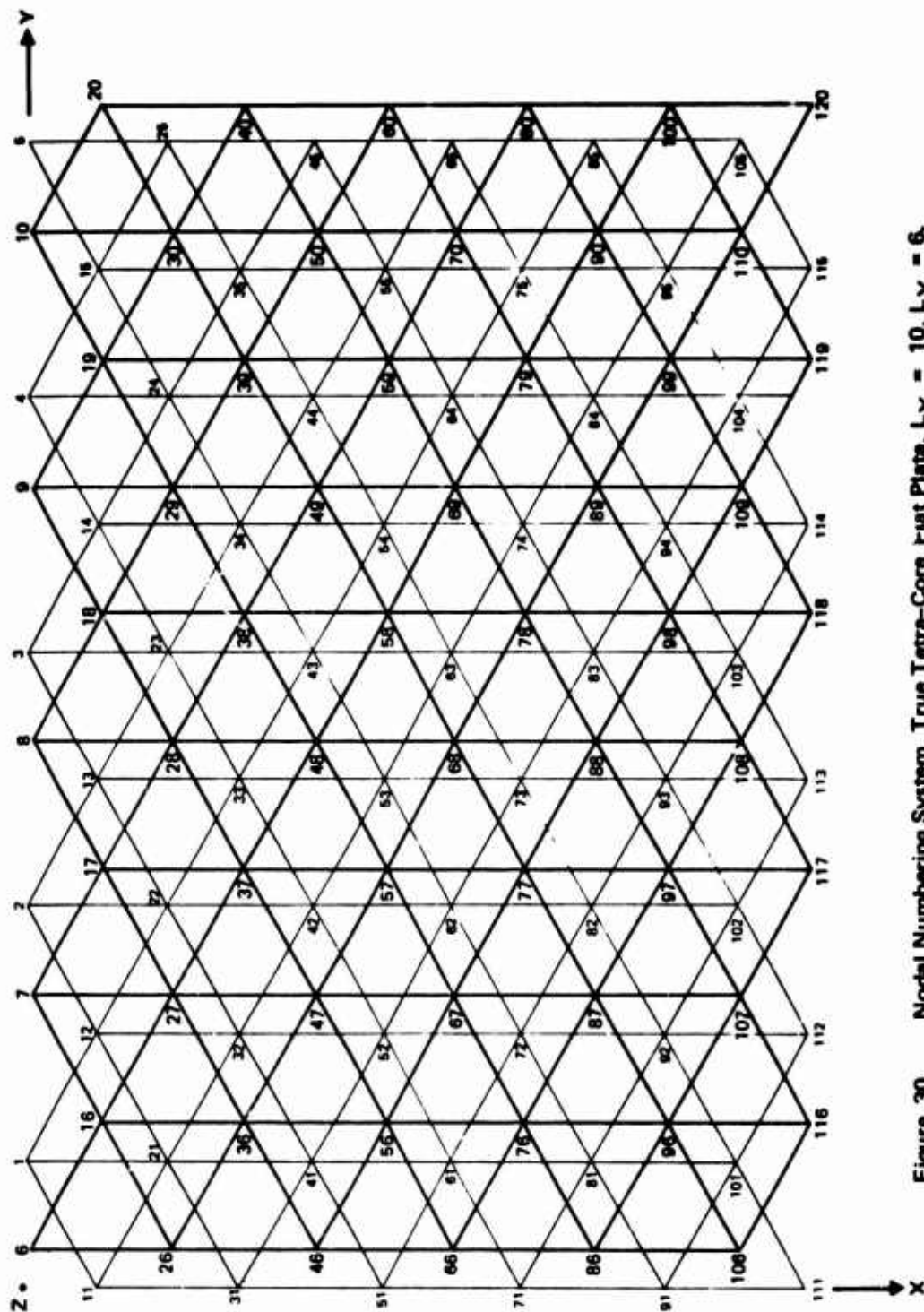


Figure 30. Nodal Numbering System, True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$.

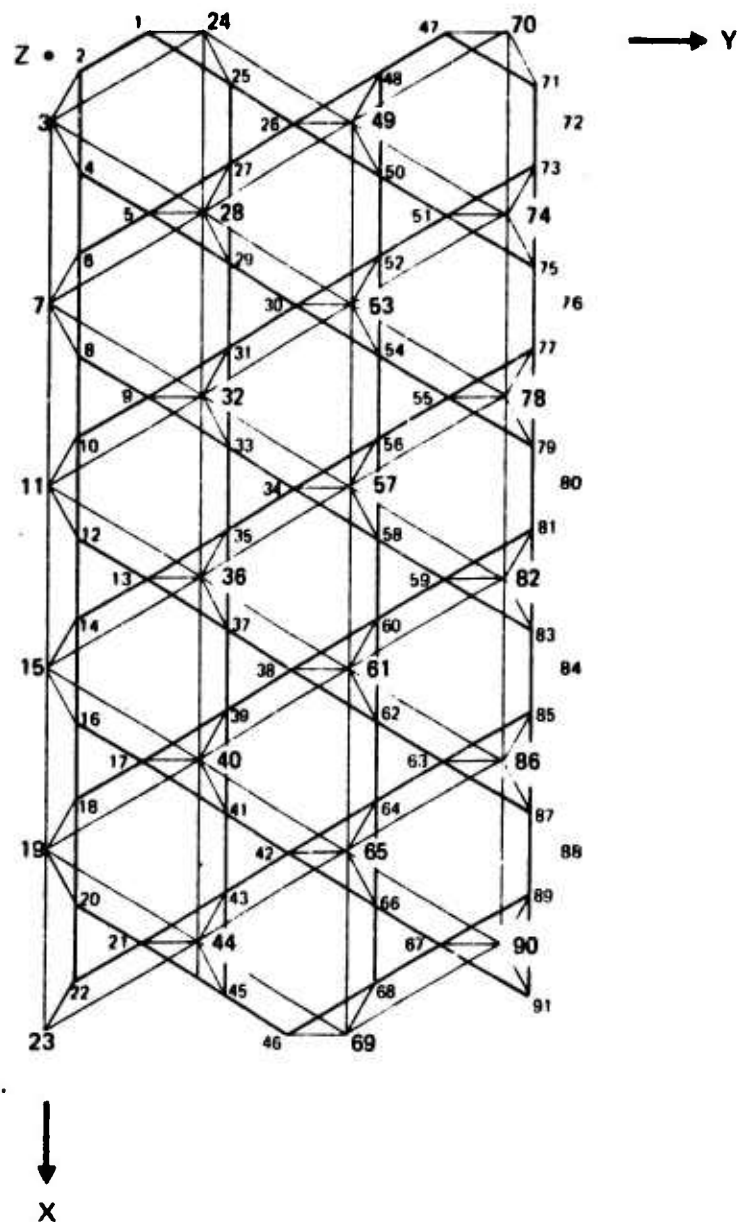


Figure 31. Nodal Numbering System, Truncated Tetra-Core Flat Plate,
 $L_X = 4$, $L_Y = 6$.

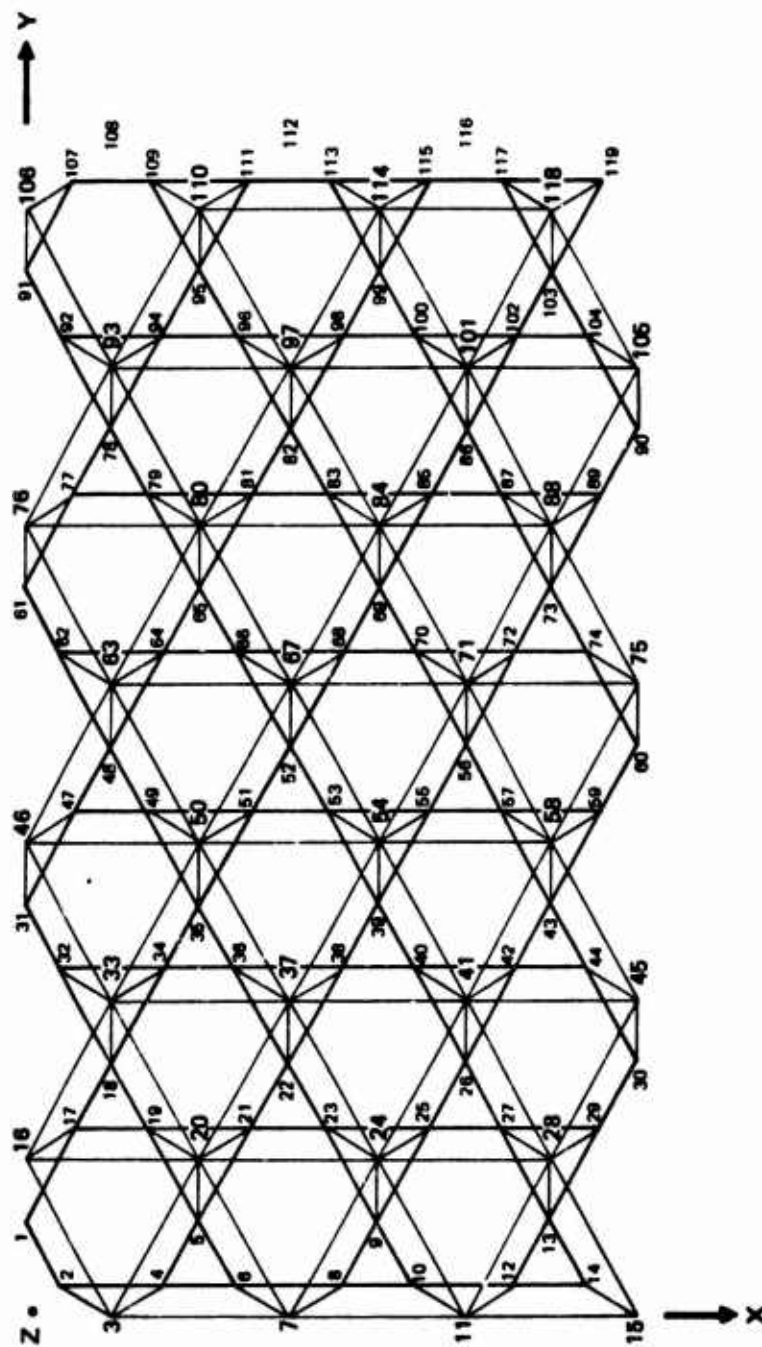


Figure 32. Nodal Numbering System, Truncated Tetra-Core Flat Plate,
 $L_X = 8, L_Y = 4$.

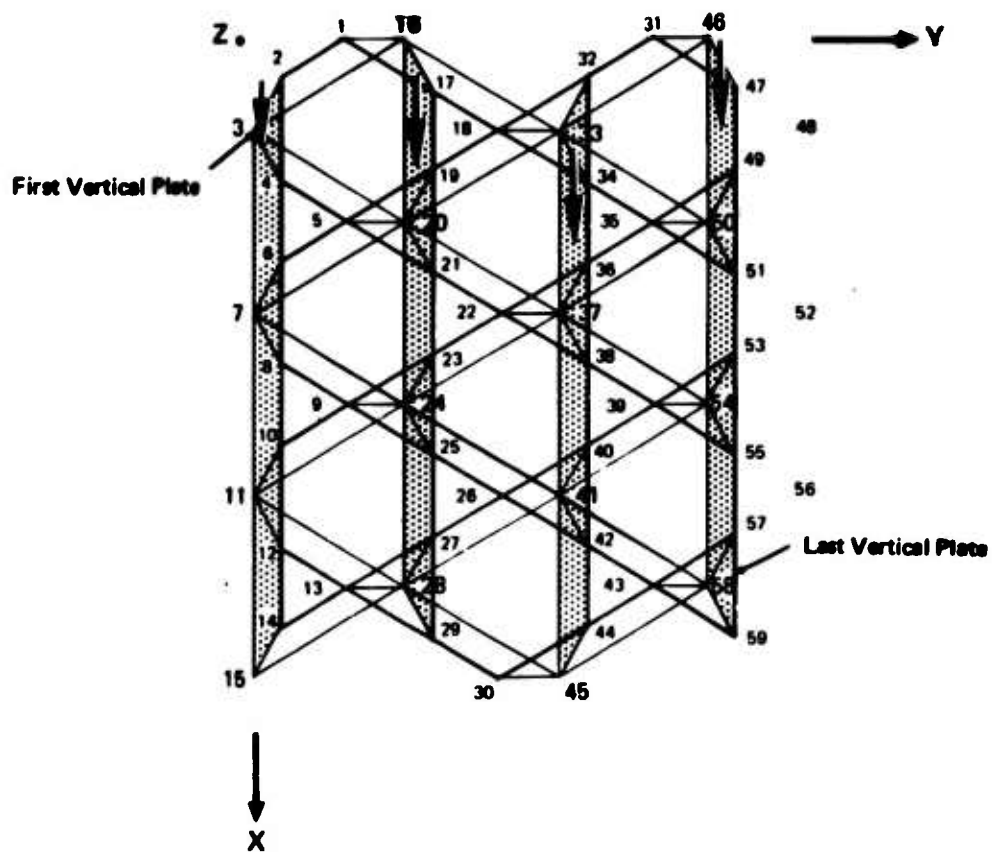


Figure 33. Vertical Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$.

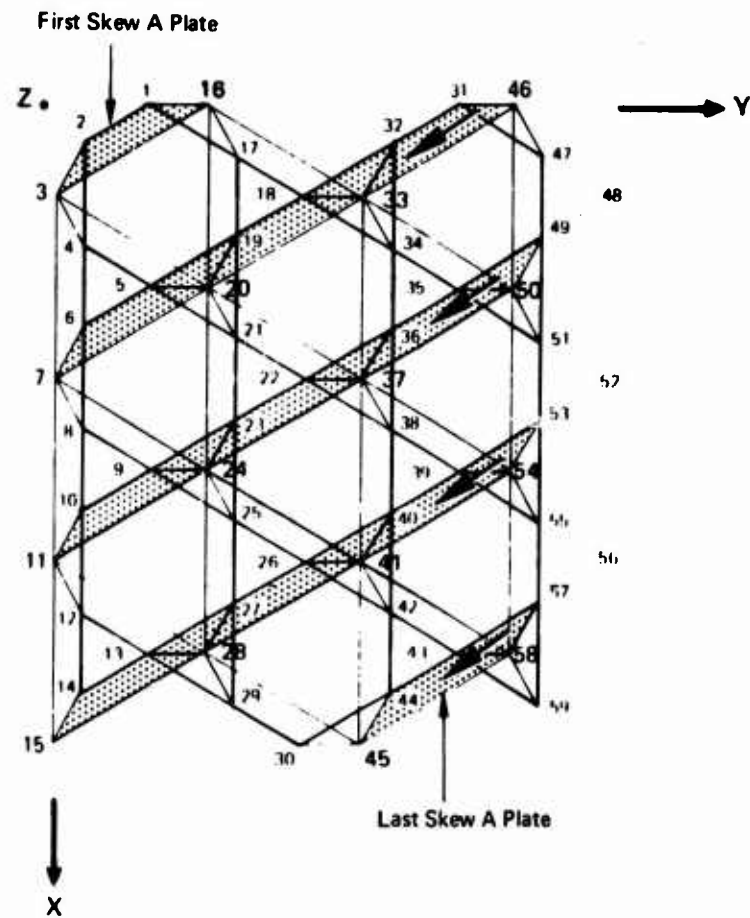


Figure 34. Skew A Leg Plate Numbering System Truncated Tetra-Core,
 $L_X = 4$, $L_Y = 4$.

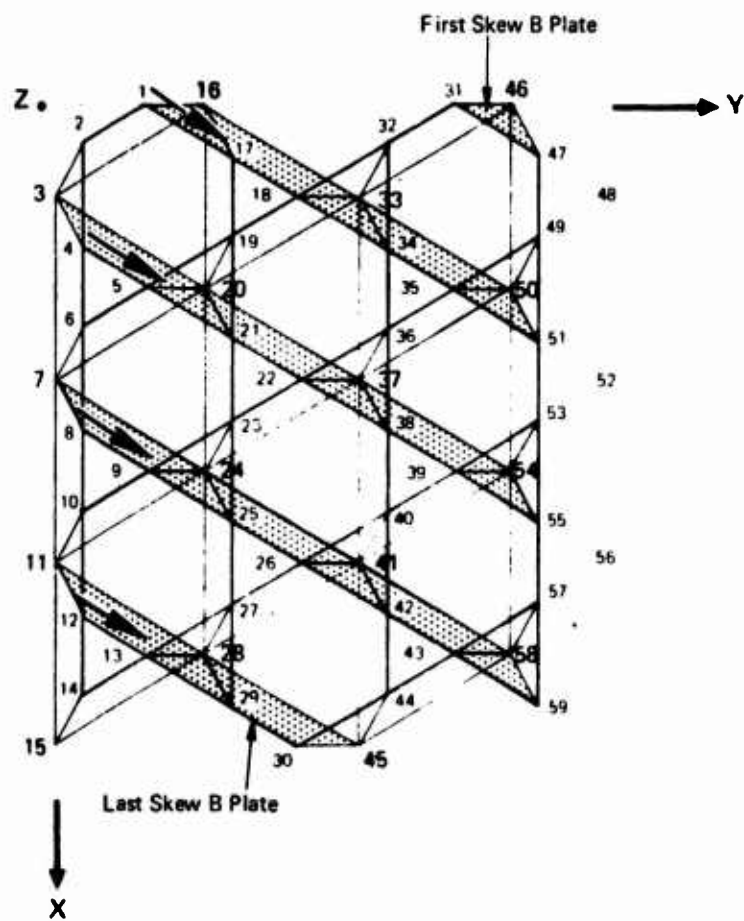


Figure 35. Skew B Leg Plate Numbering System Truncated Tetra-Core,
 $L_X = 4$, $L_Y = 4$.

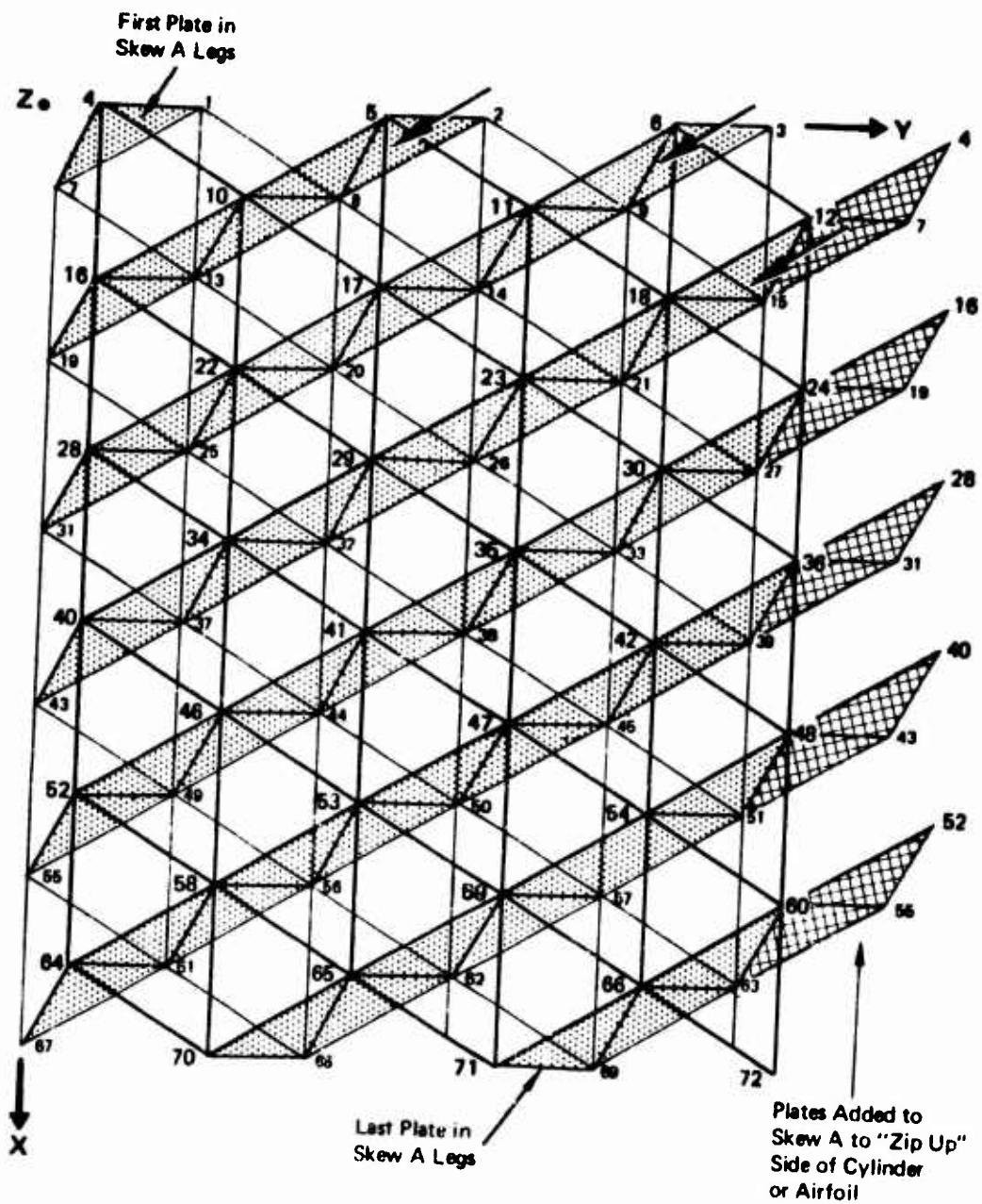


Figure 36. Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$.

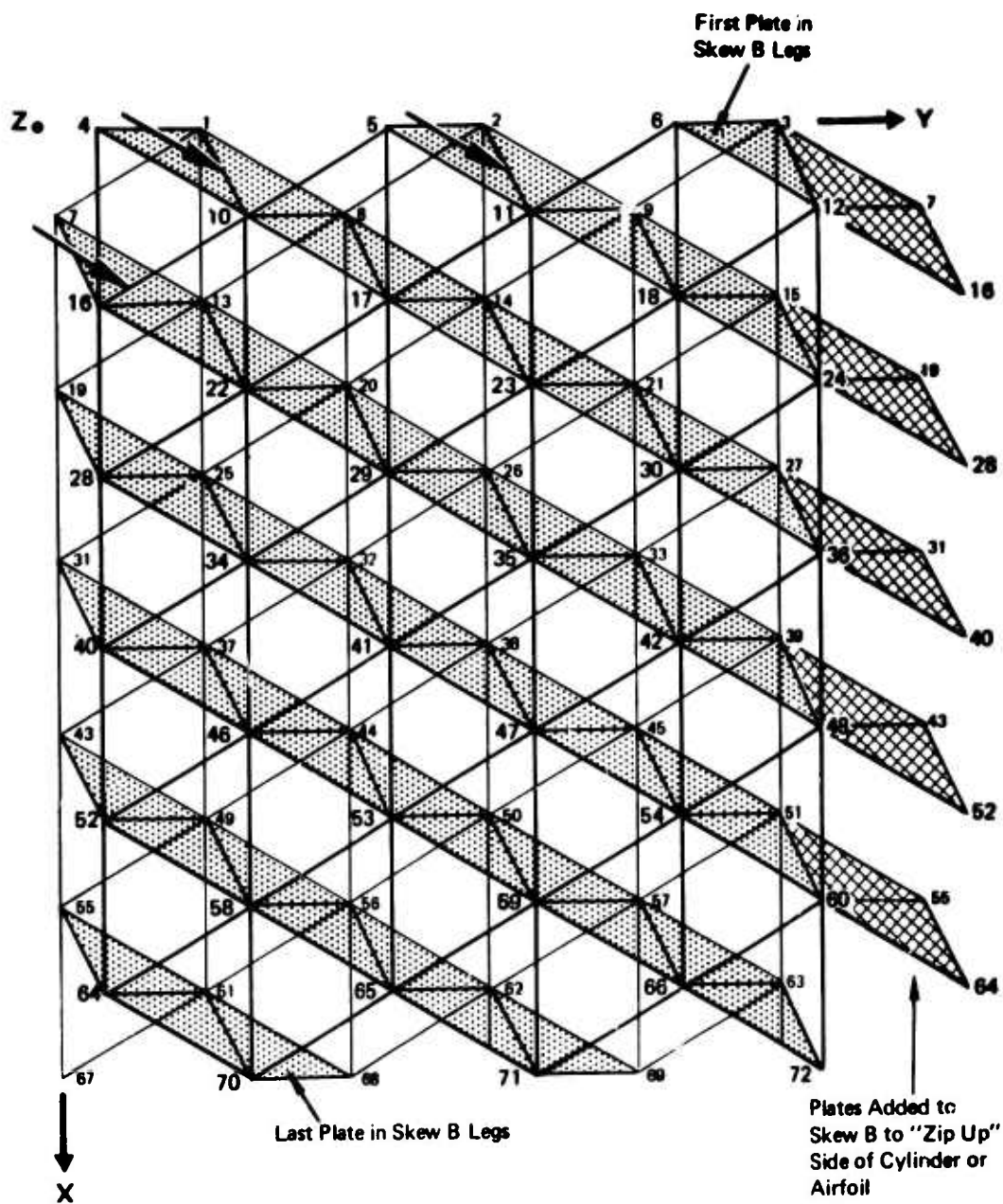


Figure 37. Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$.

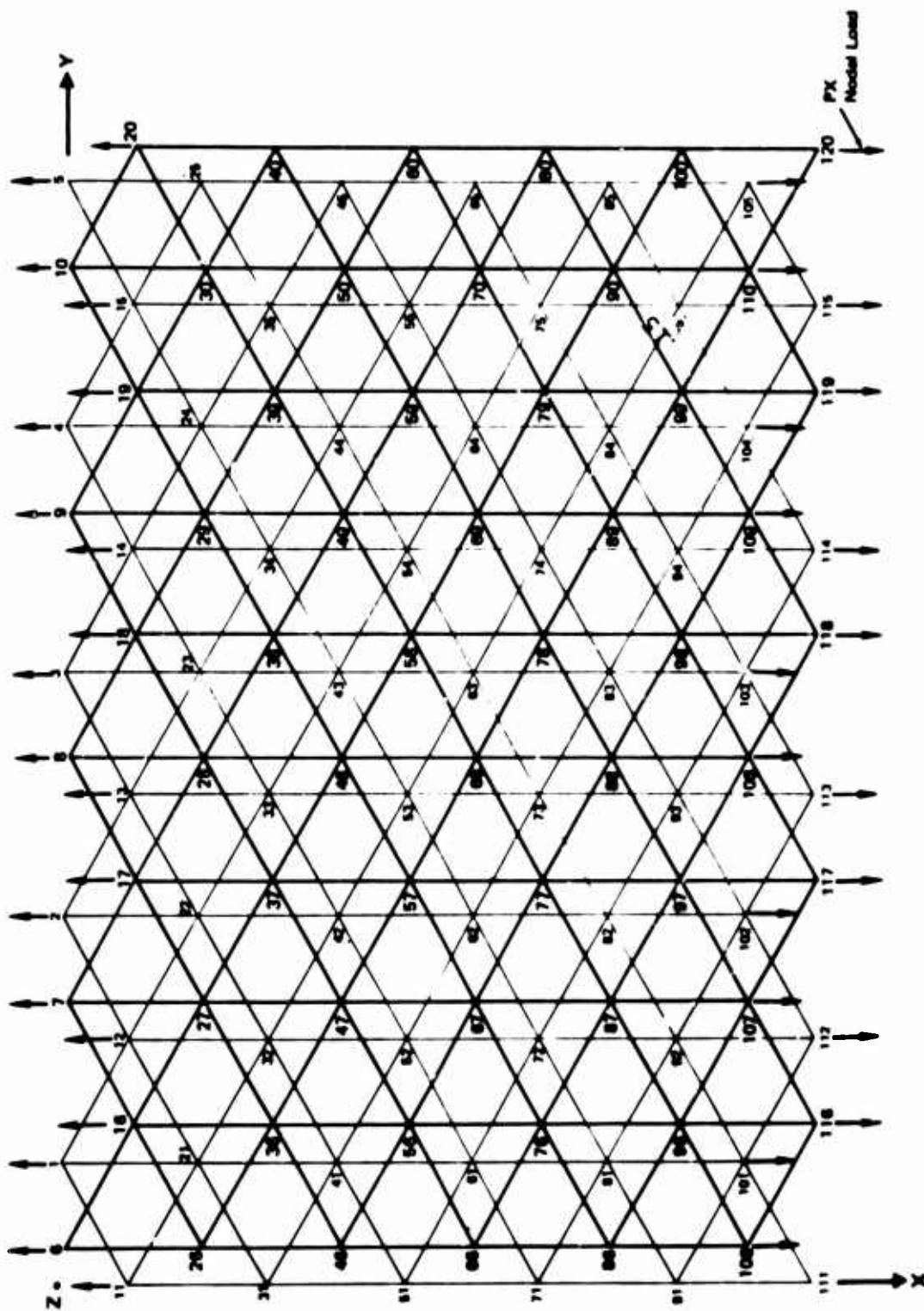


Figure 38. Load N_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

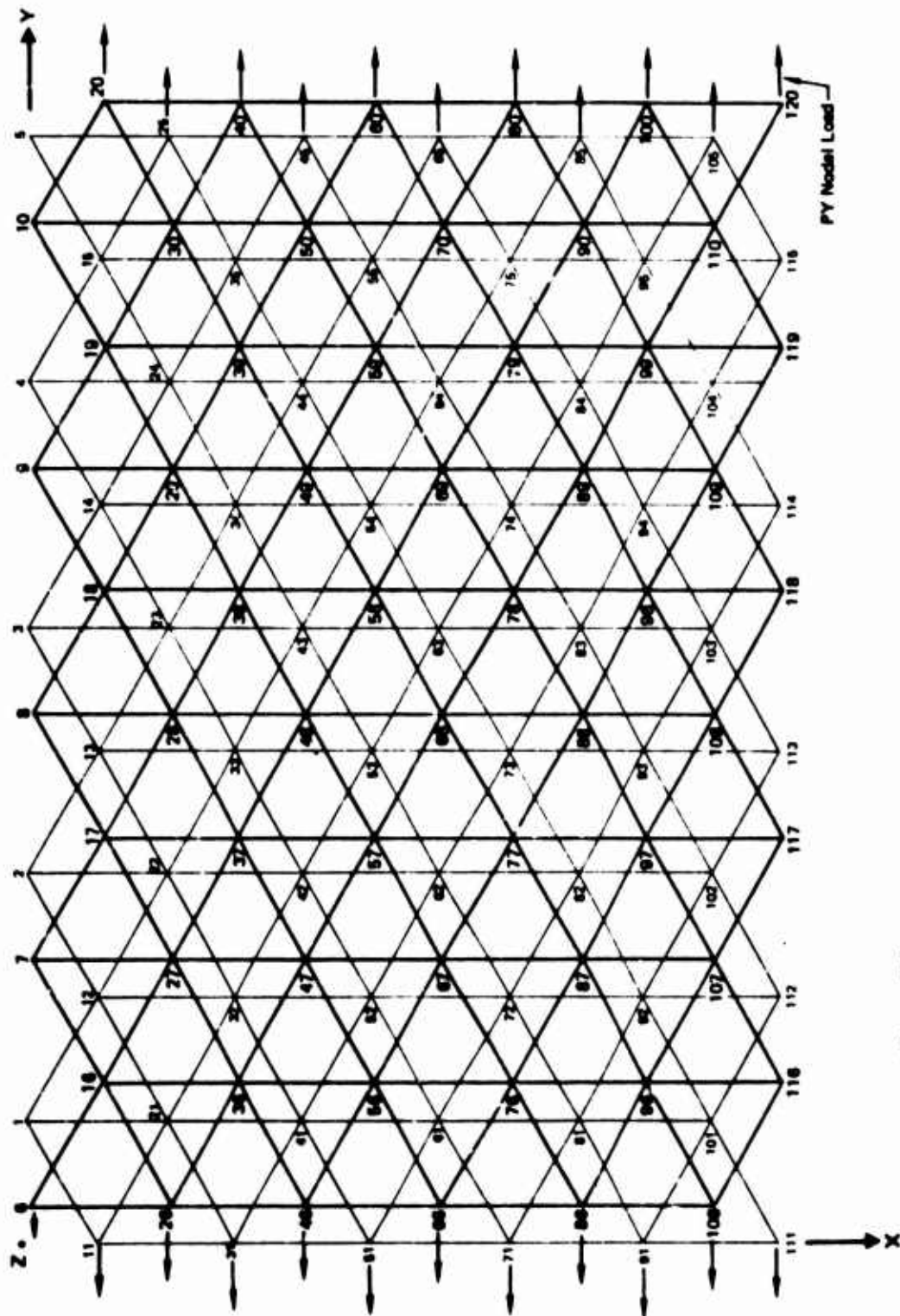


Figure 38. Load N_y Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$.

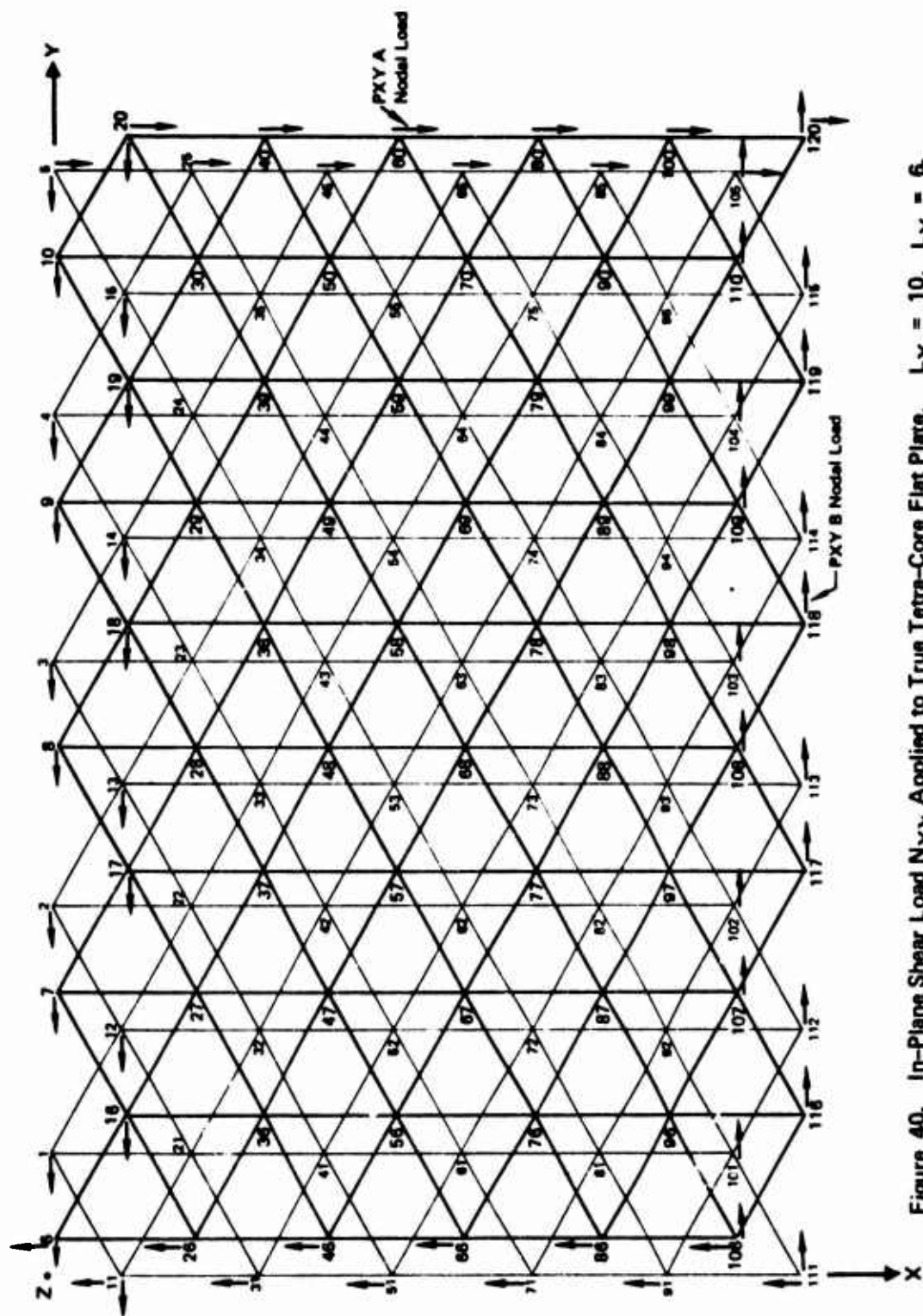


Figure 40. In-Plane Shear Load N_{xy} Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$.

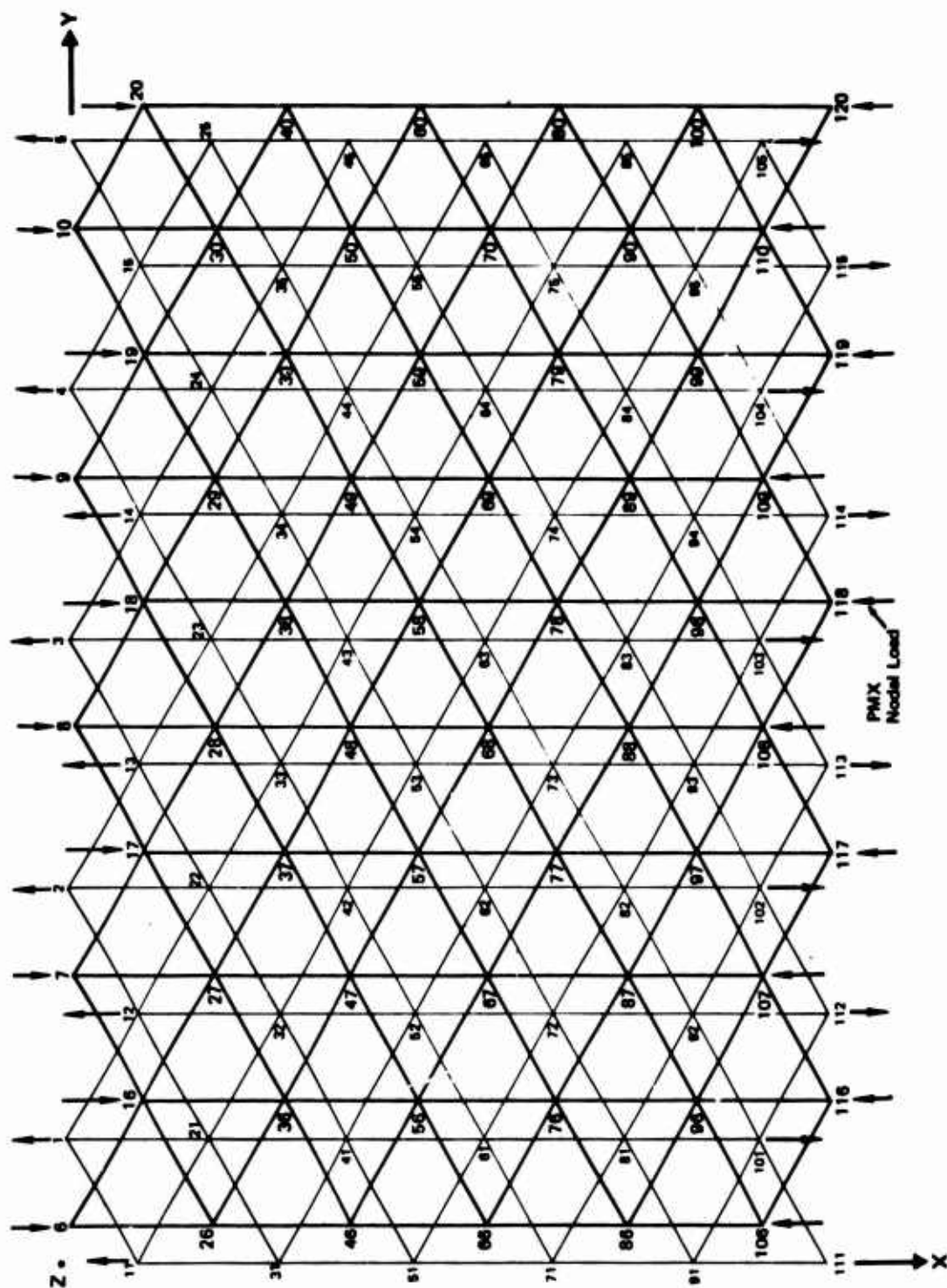


Figure 41. Moment M_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

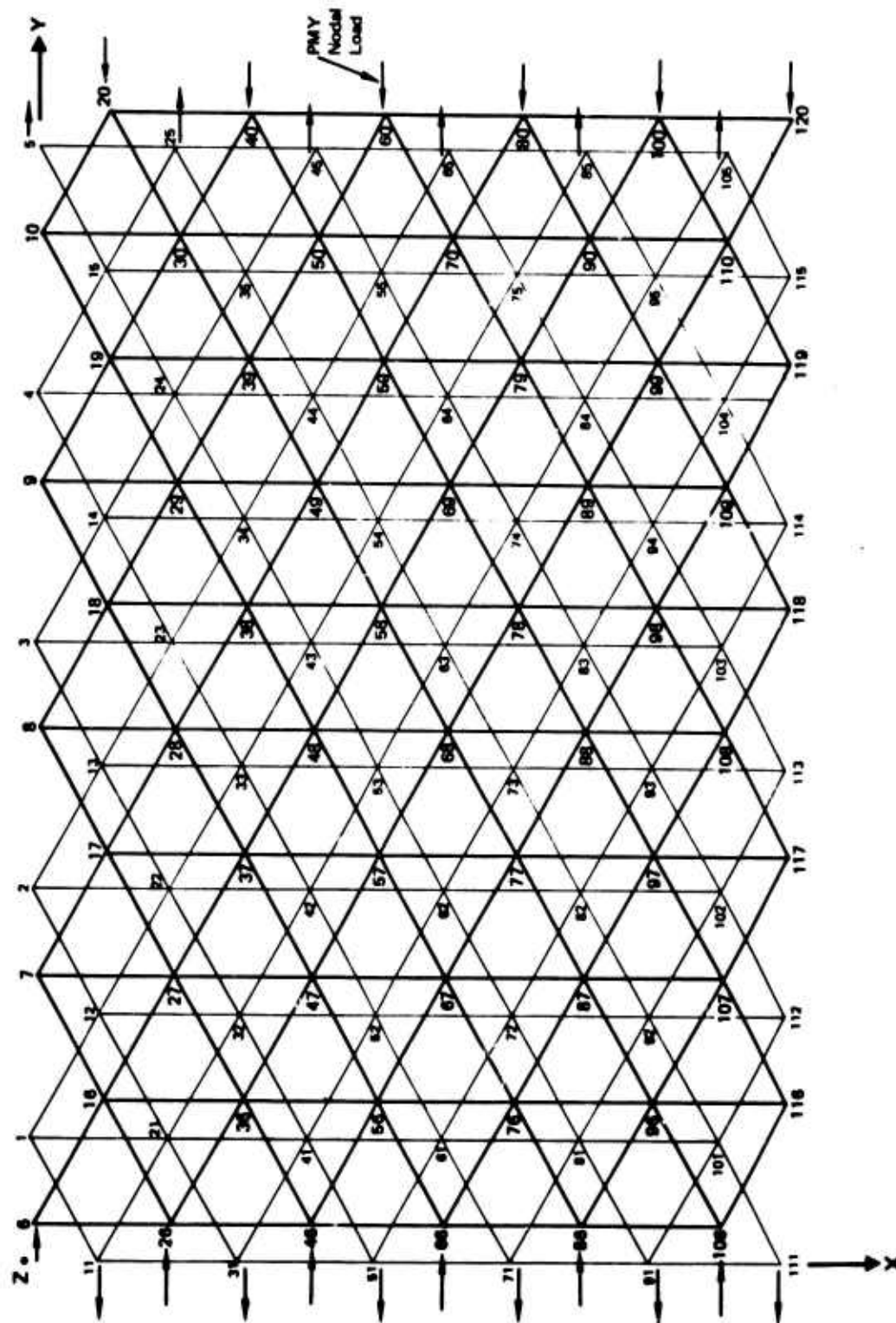
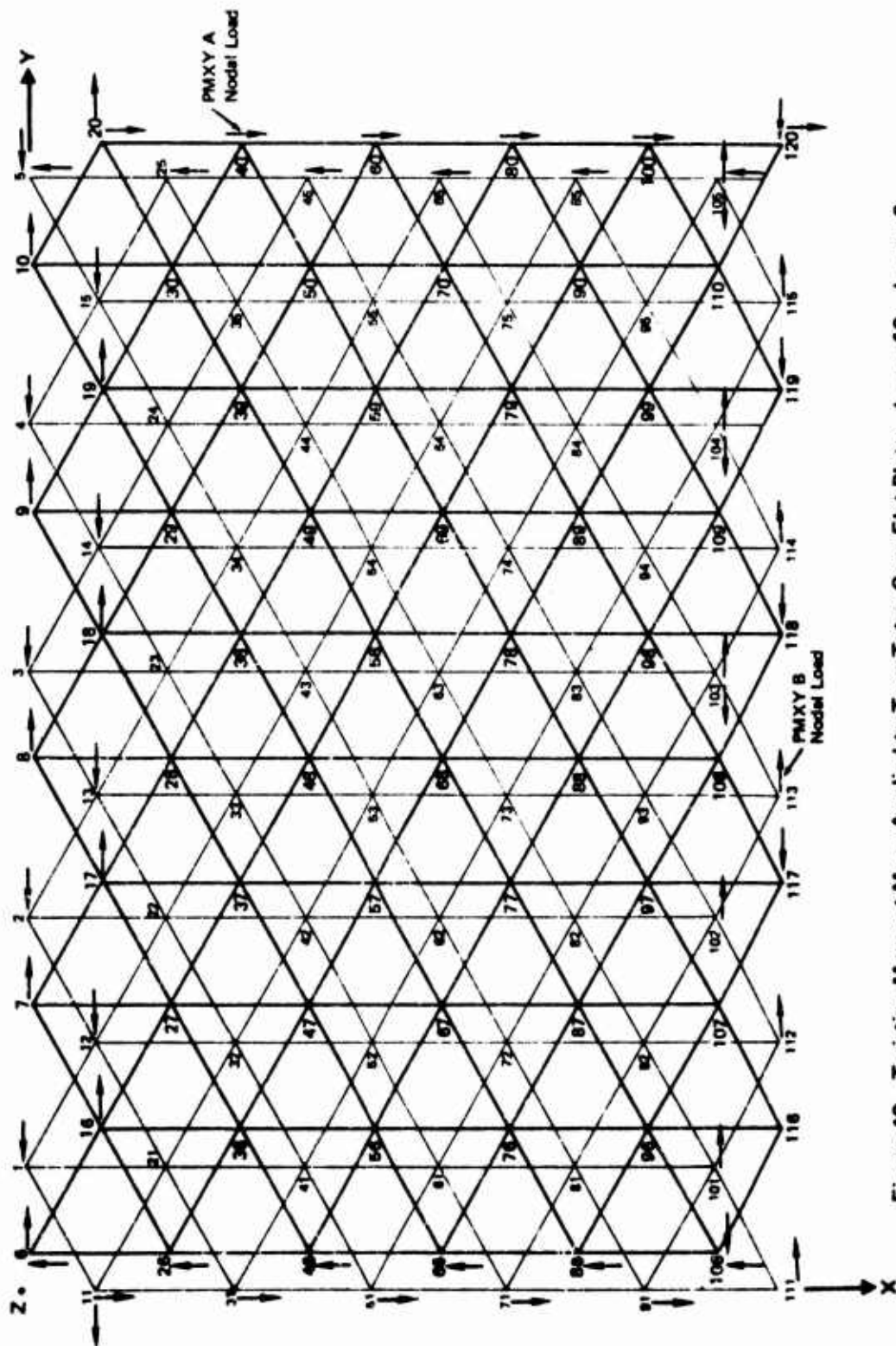


Figure 4.2. Moment M_y Applied to True Tetra-Core Flat Plate, $L_x = 10$, $L_y = 6$.



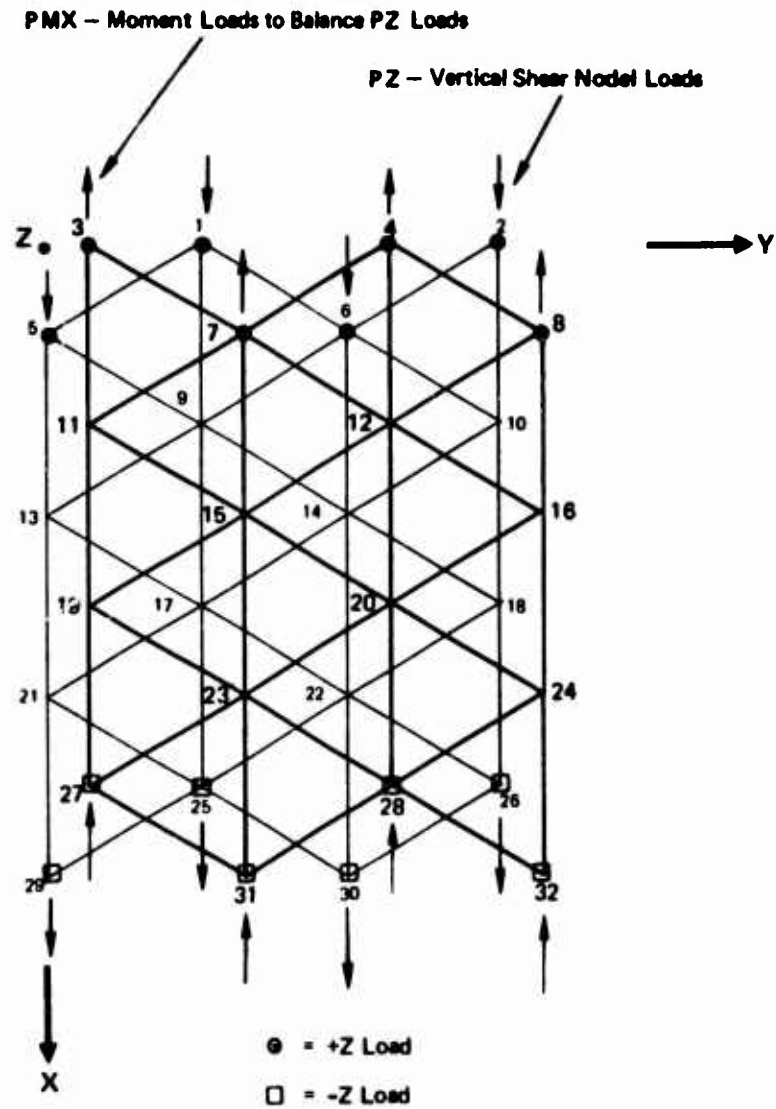


Figure 44. Vertical Shear Load XOSHR Applied to Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$.

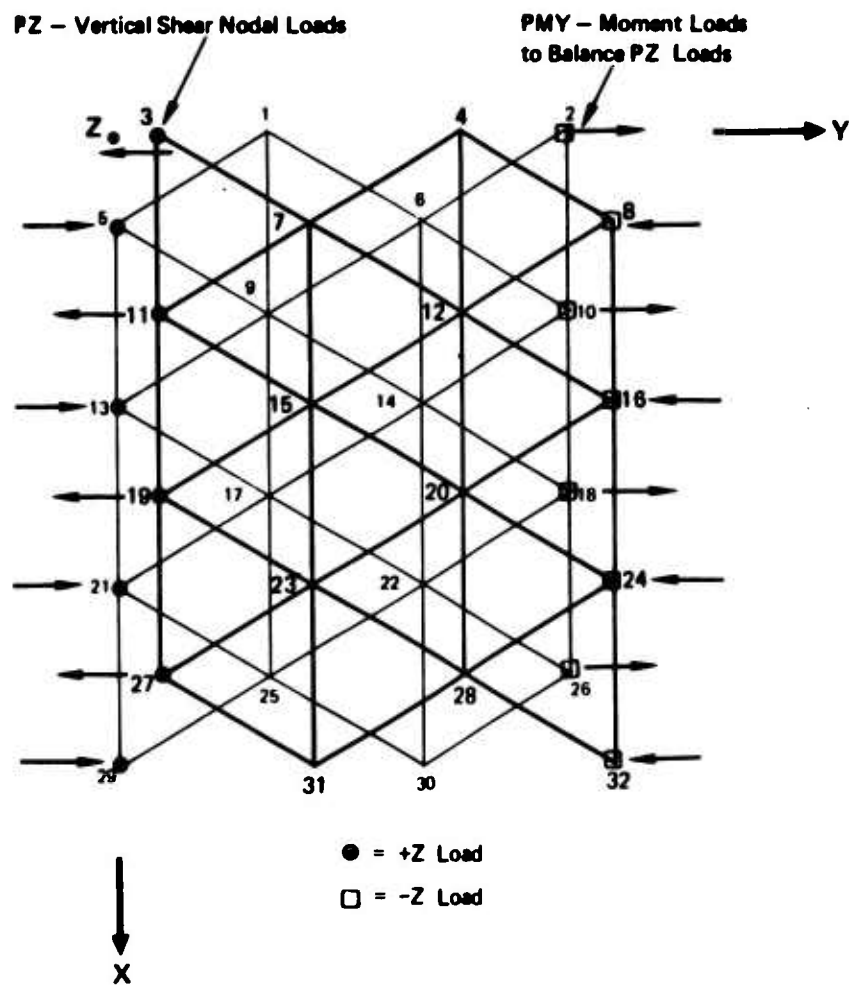


Figure 45. Vertical Shear Load YOSHR Applied to Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

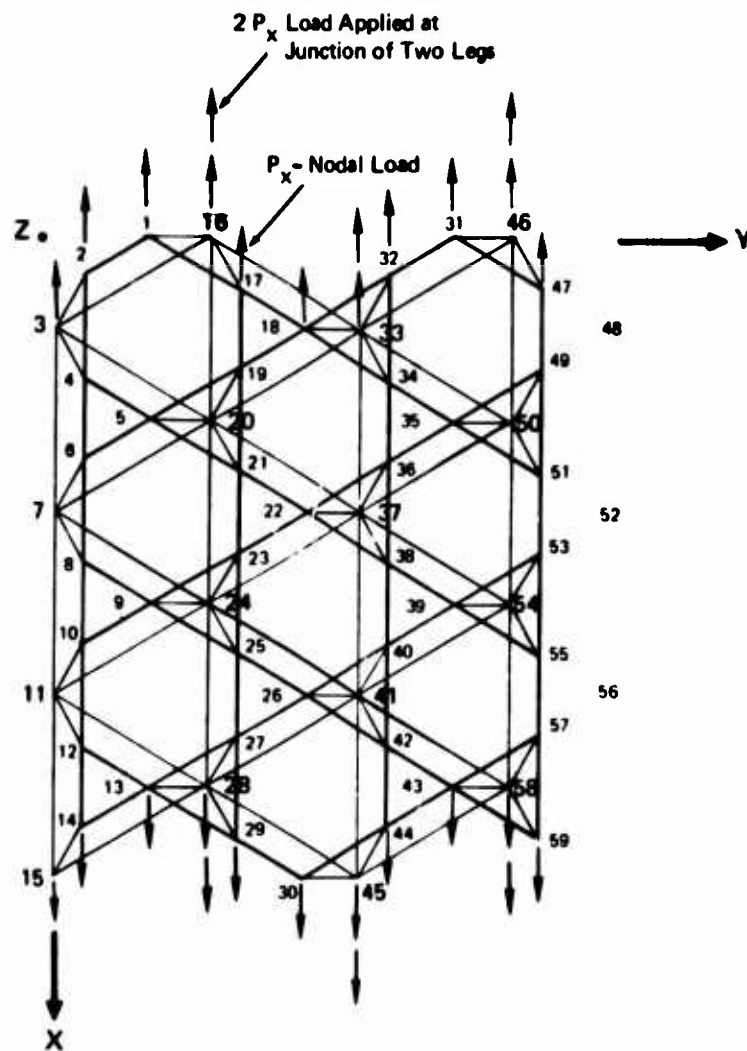


Figure 46. Load N_x Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$.

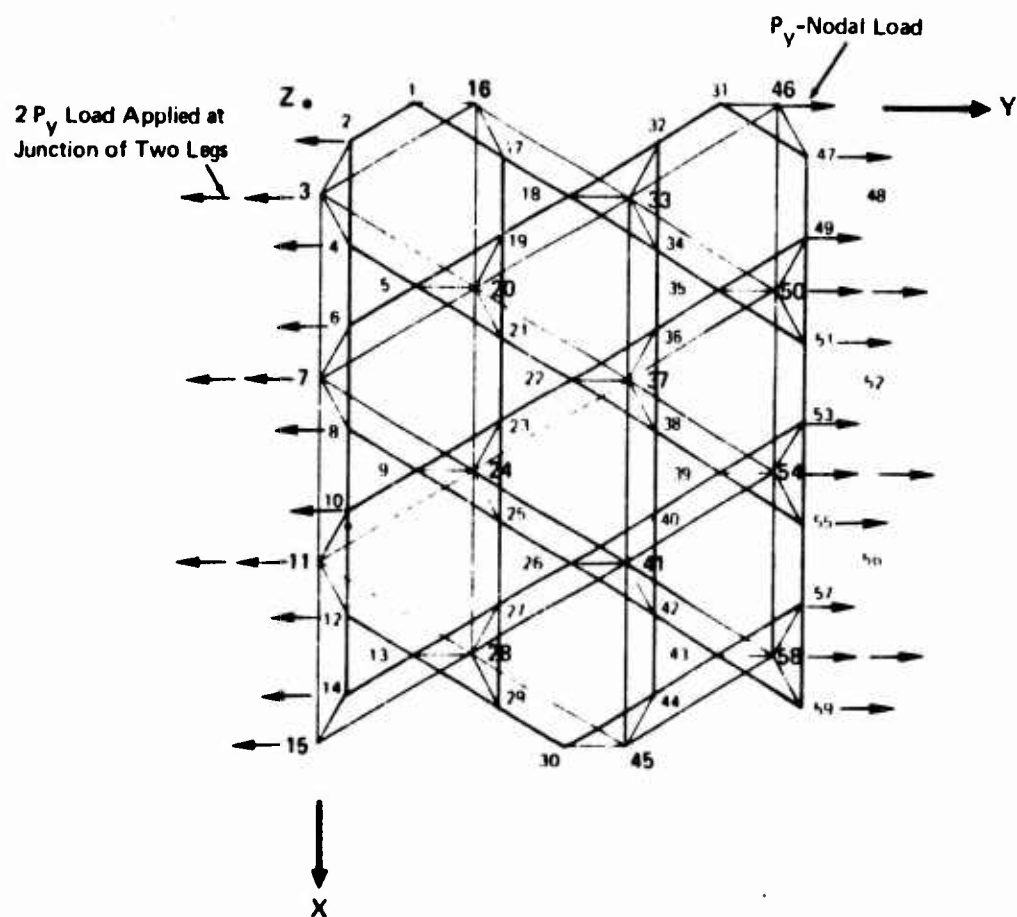


Figure 47. Load N_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

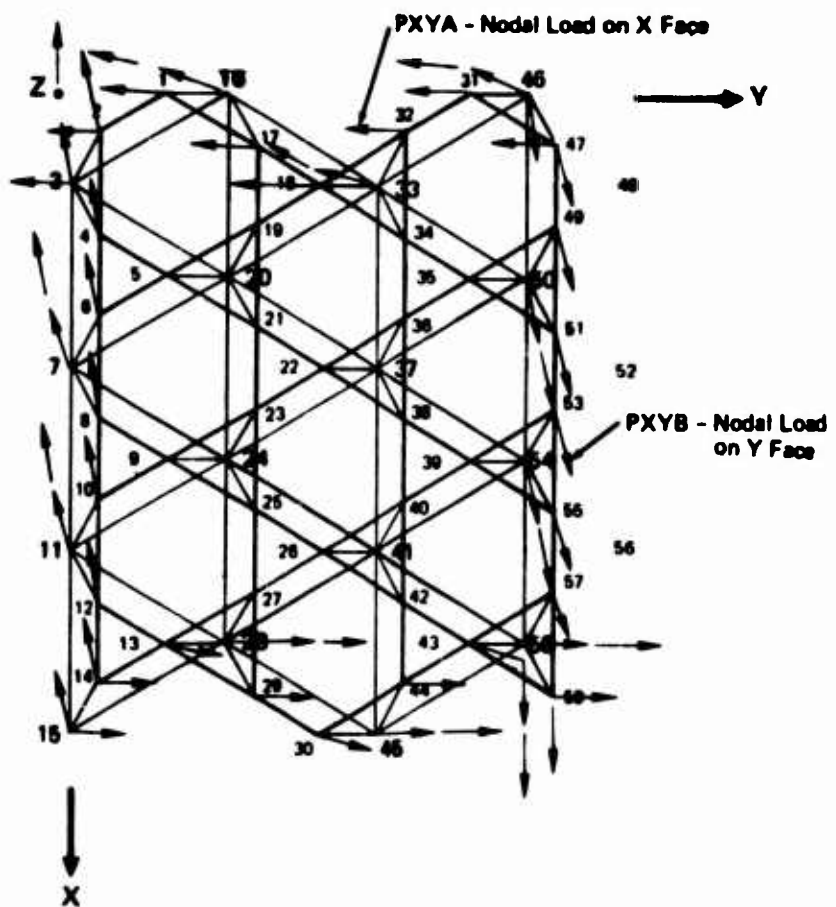


Figure 48. In-Plane Shear Load N_{XY} Applied to Truncated Tetra-Core Flat Plate, $L_X = 4, L_Y = 4$.

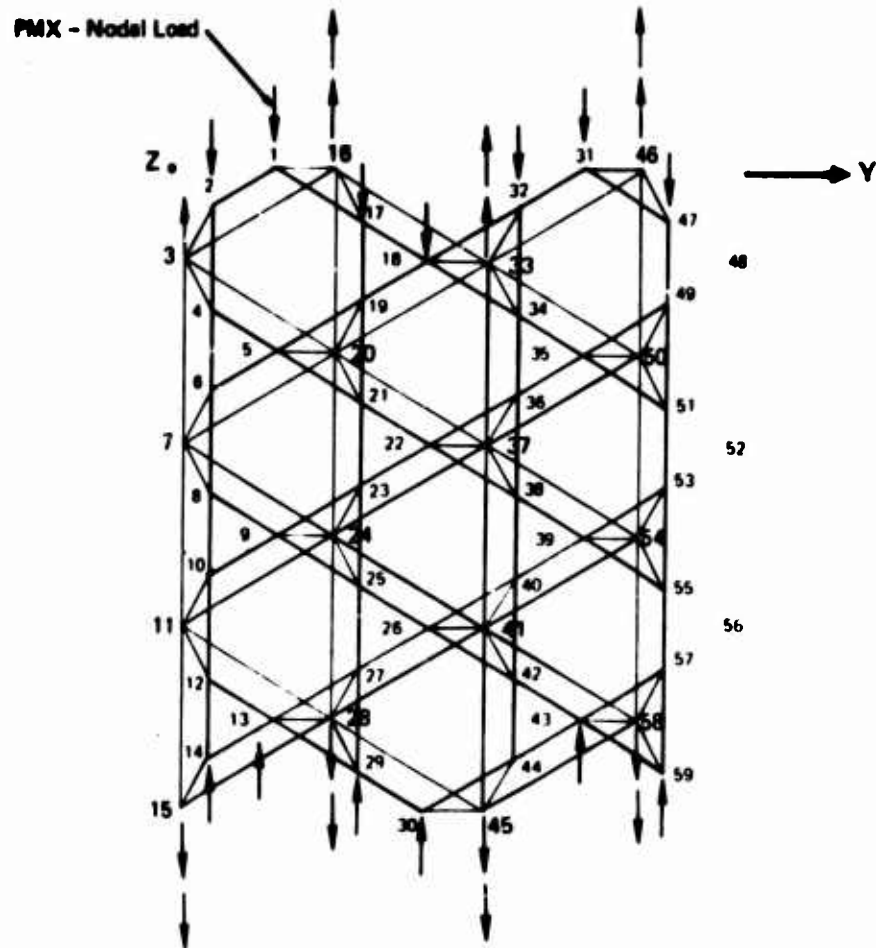


Figure 49. Moment M_x Applied to Truncated Tetra-Core Flat Plate, $L_x = 4$, $L_y = 4$.

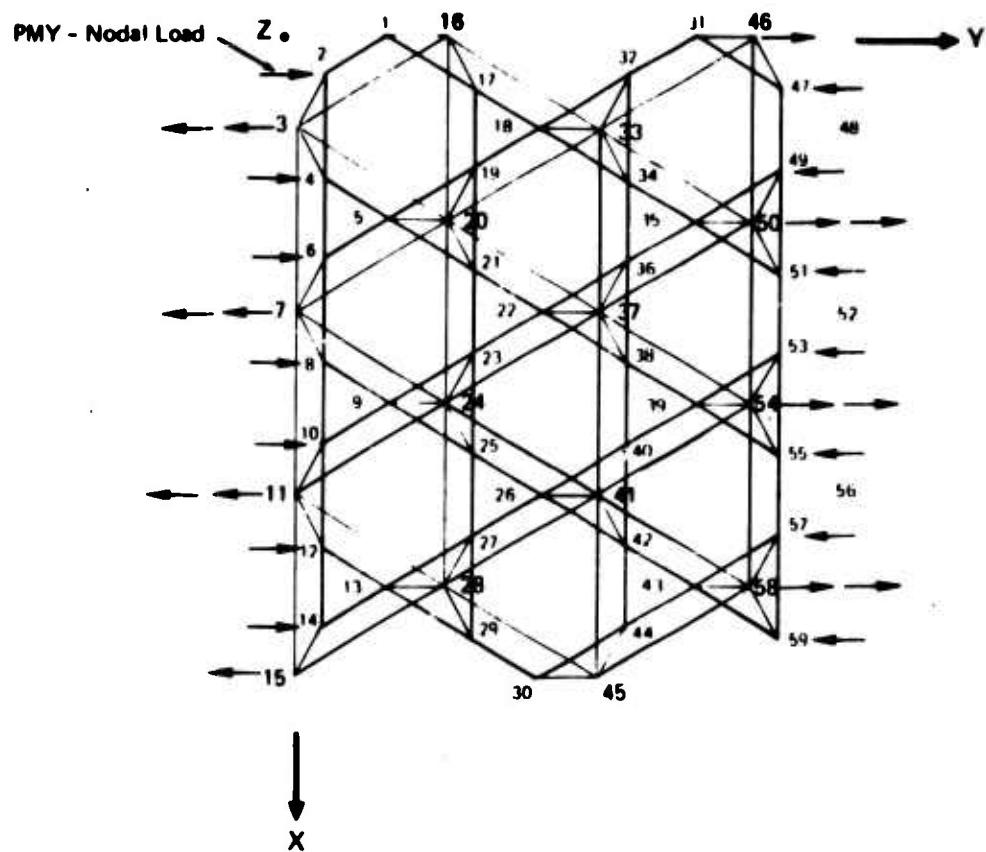


Figure 50. Moment M_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

APPENDIX II
EXAMPLE OF COMPUTER PROGRAM OUTPUTS

An example of the output from an optimization run is given (Figure 51). Stresses and deflections for the input configuration are printed. Then the optimization steps are printed. Finally, stresses and deflections for the optimized section are printed. Element stresses are printed out of sequence in this example, although they are in sequence in the latest version of the program.

FAT	FIT	FAC	FVC	FAY
11000	13000	11000	13000	11000
11000	13000	11000	13000	11000
11000	13000	11000	13000	11000

LEG DENSITY = .10000 .10000 .10000

FIRST VERTICAL PLATE = 1 SKEW A = 25 SKEW B = 45 LWC FACE = 0 UPPER FACE = 0

NA	LOAD CASE	1	MAX	MY	TORQUE	CX	CY
1000	MT	-J	-0	-0	-0	-0	-0

NUMBER OF NOJAL POINTS----- JK
NUMBER OF ELEMENTS----- 00
NUMBER OF LOAL CASES --- 1

[illegible]

NEW CLO		LGAW CASE 1		UX		UY		LZ	
NR	NCDE	NR	NCDE	NR	NCDE	NR	NCDE	NR	NCDE
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	3001	4	3001	4	3001	4	3001	4	3001
5	4	5	4	5	4	5	4	5	4
6	4002	6	4002	6	4002	6	4002	6	4002
7	5	7	5	7	5	7	5	7	5
8	5003	8	5003	8	5003	8	5003	8	5003
9	6	9	6	9	6	9	6	9	6
10	6004	10	6004	10	6004	10	6004	10	6004
11	7	11	7	11	7	11	7	11	7
12	7005	12	7005	12	7005	12	7005	12	7005
13	7006	13	7006	13	7006	13	7006	13	7006
14	8	14	8	14	8	14	8	14	8
15	8002	15	8002	15	8002	15	8002	15	8002
16	9	16	9	16	9	16	9	16	9
17	9007	17	9007	17	9007	17	9007	17	9007
18	10	18	10	18	10	18	10	18	10
19	10008	19	10008	19	10008	19	10008	19	10008
20	11	20	11	20	11	20	11	20	11
21	11009	21	11009	21	11009	21	11009	21	11009
22	12	22	12	22	12	22	12	22	12
23	12009	23	12009	23	12009	23	12009	23	12009
24	13	24	13	24	13	24	13	24	13
25	13010	25	13010	25	13010	25	13010	25	13010
26	14	26	14	26	14	26	14	26	14
27	14011	27	14011	27	14011	27	14011	27	14011
28	15	28	15	28	15	28	15	28	15
29	15012	29	15012	29	15012	29	15012	29	15012
30	16	30	16	30	16	30	16	30	16
31	16013	31	16013	31	16013	31	16013	31	16013
32	17	32	17	32	17	32	17	32	17
33	17014	33	17014	33	17014	33	17014	33	17014
34	18	34	18	34	18	34	18	34	18
35	18015	35	18015	35	18015	35	18015	35	18015
36	19	36	19	36	19	36	19	36	19
37	19016	37	19016	37	19016	37	19016	37	19016
38	20	38	20	38	20	38	20	38	20
39	20017	39	20017	39	20017	39	20017	39	20017
40	21	40	21	40	21	40	21	40	21
41	21018	41	21018	41	21018	41	21018	41	21018
42	22	42	22	42	22	42	22	42	22
43	22019	43	22019	43	22019	43	22019	43	22019
44	23	44	23	44	23	44	23	44	23
45	23020	45	23020	45	23020	45	23020	45	23020
46	24	46	24	46	24	46	24	46	24
47	24021	47	24021	47	24021	47	24021	47	24021
48	25	48	25	48	25	48	25	48	25
49	25022	49	25022	49	25022	49	25022	49	25022
50	26	50	26	50	26	50	26	50	26
51	26023	51	26023	51	26023	51	26023	51	26023
52	27	52	27	52	27	52	27	52	27
53	27024	53	27024	53	27024	53	27024	53	27024

Reproduced from
Best available copy.

23 24014
24 25025
25 25023
26 26
27 26024
28 26027
29 27021
30 27025
31 28
32 28322
33 28026
34 29
35 29027
36 30030
37 30028
38 31
39 31025
40 31030
41 32
42 32026

23.1607
24.1740
25.0000
26.0002
27.1607
28.1621
29.0000
30.11.5317
31.1740
32.0000
33.0000
34.0000
35.0000
36.0000
37.0000
38.0000
39.0000
40.0000
41.0000
42.0000

23.0000
24.0000
25.0000
26.0000
27.0000
28.0000
29.0000
30.0000
31.0000
32.0000
33.0000
34.0000
35.0000
36.0000
37.0000
38.0000
39.0000
40.0000
41.0000
42.0000

23.0000
24.0000
25.0000
26.0000
27.0000
28.0000
29.0000
30.0000
31.0000
32.0000
33.0000
34.0000
35.0000
36.0000
37.0000
38.0000
39.0000
40.0000
41.0000
42.0000

Reproduced from
best available copy.

ELEMENT
NUMBER

I	J	K	EA	EX	LAY	G	T	CF	LU	LR	LS	LI
1	3	11	12	4031496	203937	200000	00000	3	20	7	21	0
2	3	13	11	4031496	203937	200000	00000	3	20	7	21	0
3	13	19	13	4031496	203937	200000	00000	30	39	35	40	21
4	13	21	19	4031496	203937	200000	00000	30	39	35	40	21
5	21	27	21	4031496	203937	200000	00000	34	20	42	29	40
6	21	29	27	4031496	203937	200000	00000	34	20	42	29	40
7	1	9	7	4031496	203937	200000	00000	1	10	11	11	12
8	7	15	7	4031496	203937	200000	00000	1	10	11	11	12
9	9	17	17	4031496	203937	200000	00000	10	32	33	30	31
10	15	23	17	4031496	203937	200000	00000	30	49	33	30	30
11	17	25	25	4031496	203937	200000	00000	35	24	42	52	50
12	23	31	25	4031496	203937	200000	00000	49	20	54	63	55
13	4	12	0	4031496	203937	200000	00000	3	23	5	24	10
14	0	12	0	4031496	203937	200000	00000	3	23	5	24	10
15	12	20	14	4031496	203937	200000	00000	23	42	42	43	25
16	14	22	20	4031496	203937	200000	00000	28	47	42	48	43
17	20	20	22	4031496	203937	200000	00000	42	51	47	62	40
18	22	30	20	4031496	203937	200000	00000	47	50	61	67	62
19	2	10	0	4031496	203937	200000	00000	2	13	14	15	15
20	0	10	0	4031496	203937	200000	00000	2	13	14	15	15
21	10	10	10	4031496	203937	200000	00000	13	37	37	39	34
22	10	14	10	4031496	203937	200000	00000	13	32	37	39	34
23	10	20	24	4031496	203937	200000	00000	37	20	22	22	23
24	24	32	20	4031496	203937	200000	00000	37	21	22	22	23
25	1	5	3	4031496	203937	200000	00000	1	7	5	10	6
26	2	0	4	4031496	203937	200000	00000	2	9	5	10	6
27	7	7	5	4031496	203937	200000	00000	3	11	5	17	13
28	0	9	7	4031496	203937	200000	00000	3	10	11	17	13
29	7	11	9	4031496	203937	200000	00000	11	20	16	22	17
30	9	13	11	4031496	203937	200000	00000	10	20	16	22	17
31	0	12	12	4031496	203937	200000	00000	14	23	16	25	15
32	10	14	12	4031496	203937	200000	00000	18	28	19	25	15
33	12	15	14	4031496	203937	200000	00000	23	30	21	32	25
34	14	17	15	4031496	203937	200000	00000	28	32	23	32	25
35	15	19	17	4031496	203937	200000	00000	39	39	22	41	30
36	17	21	19	4031496	203937	200000	00000	35	42	22	46	41
37	10	20	10	4031496	203937	200000	00000	33	42	27	44	30
38	10	22	20	4031496	203937	200000	00000	37	47	32	48	44
39	20	23	25	4031496	203937	200000	00000	42	43	47	51	40
40	22	25	23	4031496	203937	200000	00000	47	34	42	50	31
41	25	27	27	4031496	203937	200000	00000	44	20	14	00	25
42	25	29	27	4031496	203937	200000	00000	44	20	14	00	25
43	20	30	20	4031496	203937	200000	00000	22	21	22	02	57
44	20	30	20	4031496	203937	200000	00000	22	21	22	02	57
45	20	31	30	4031496	203937	200000	00000	21	20	22	02	57
46	0	0	0	4031496	203937	200000	00000	14	14	14	14	0
47	3	7	7	4031496	203937	200000	00000	3	11	1	14	4
48	1	0	7	4031496	203937	200000	00000	1	9	11	13	12
49	7	12	5	4031496	203937	200000	00000	11	23	5	24	13
50	6	10	21	4031496	203937	200000	00000	5	10	23	25	14

ELEMENT NUMBER	I	J	K	EX	EV	2AV	U	1	1F	1G	1H	1J	1K	1L
51	12	10	16	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
52	13	11	17	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
53	14	12	18	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
54	15	13	19	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
55	16	14	20	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
56	17	15	21	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
57	18	16	22	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
58	19	17	23	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
59	20	18	24	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
60	21	19	25	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
61	22	20	26	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
62	23	21	27	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
63	24	22	28	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
64	25	23	29	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
65	26	24	30	4031496	203937	125984	200000	0.300	27	33	10	34	22	25
66	27	25	31	4031496	203937	125984	200000	0.300	27	33	10	34	22	25

Reproduced from
best available copy.

1/2 HANDWIDTH = 21

DEFLECTIONS			LOAD CASE		1	
MODAL POINT	UX	UY	UX	UY	UX	UY
1	-1.1062902E-02	1.2413046E-02	1.2413046E-02		-1.0444619E-02	
2	4.7463019E-03	4.7231095E-03	4.7231095E-03		-1.7285240E-02	
3	-2.3341566E-02	1.1763335E-02	1.1763335E-02		-7.3456332E-03	
3001	-3.3944329E-02	1.3387172E-02	1.3387172E-02		-1.2020755E-02	
4	-5.3550875E-03	3.5473333E-04	3.5473333E-04		-1.1754336E-02	
4002	-1.4746332E-02	4.4915973E-04	4.4915973E-04		-1.5476175E-02	
5	-2.3164284E-02	1.1414373E-02	1.1414373E-02		1.7729734E-04	
5003	-2.5235011E-02	1.0909123E-02	1.0909123E-02		-3.3925754E-03	
6	-1.5310636E-03	3.4940712E-03	3.4940712E-03		-3.0271049E-03	
6004	-4.3065695E-03	5.0543306E-03	5.0543306E-03		-1.0132737E-02	
7	-1.4053242E-02	1.4449315E-03	1.4449315E-03		-1.1508348E-02	
7001	-9.0026975E-03	1.5094453E-03	1.5094453E-03		-1.0941043E-02	
7006	-1.0440842E-02	2.0276010E-03	2.0276010E-03		-1.0012035E-02	
8	-5.3715116E-04	5.5499002E-03	5.5499002E-03		-1.5338681E-02	
8002	4.8900670E-03	-4.2643045E-03	-4.2643045E-03		-1.2555408E-02	
9	-3.1923305E-15	4.0723257E-15	4.0723257E-15		-1.5481304E-15	
9007	-7.0528317E-03	3.3842375E-03	3.3842375E-03		-0.8466256E-03	
10	1.1442845E-02	-3.1050342E-03	-3.1050342E-03		-0.5118781E-03	
10008	7.0440530E-03	-1.047849E-03	-1.047849E-03		-1.0413720E-02	
11	-1.9079627E-02	3.1254333E-03	3.1254333E-03		0.0231570E-04	
11005	-2.2041034E-02	3.0450042E-03	3.0450042E-03		1.4444402E-04	
11009	-1.0042124E-02	1.447745E-03	1.447745E-03		0.0253312E-07	
12	-2.4442270E-03	-3.4565179E-03	-3.4565179E-03		-5.4746308E-03	
12006	-1.5947310E-07	-1.1208104E-03	-1.1208104E-03		-0.0289755E-03	
12010	3.3723237E-03	-3.1526341E-03	-3.1526341E-03		-3.0640379E-03	
13	-1.3170995E-02	2.3306306E-04	2.3306306E-04		6.0174545E-02	
13011	-1.6718359E-02	-2.7227334E-03	-2.7227334E-03		2.9803395E-02	
14	5.2224635E-03	-2.4443032E-03	-2.4443032E-03		0.1228755E-27	
14012	1.7170295E-03	-2.8041480E-03	-2.8041480E-03		-0.5835345E-03	
15	-4.7023240E-03	-2.0094145E-03	-2.0094145E-03		1.4038527E-04	
15009	-3.0194293E-03	-3.0474471E-04	-3.0474471E-04		1.425216E-04	
15014	-1.5522870E-04	-2.5760343E-03	-2.5760343E-03		6.2332411E-04	
16	7.7012324E-03	-0.2006306E-03	-0.2006306E-03		-7.0387350E-04	
16018	9.4471733E-03	-0.7108335E-03	-0.7108335E-03		-3.2057755E-03	
17	1.1053369E-03	-1.2729227E-15	-1.2729227E-15		1.5481334E-15	
17015	-9.0858034E-04	-1.4338790E-03	-1.4338790E-03		1.0675527E-02	
18	1.0773460E-02	-0.0134011E-03	-0.0134011E-03		4.3450301E-03	
18016	1.2952500E-02	3.1003342E-03	3.1003342E-03		1.0431127E-03	
19	-1.2570513E-02	-3.7706352E-03	-3.7706352E-03		7.3416707E-04	
19013	-1.4987469E-02	2.7610879E-03	2.7610879E-03		3.3906245E-04	
19017	-5.4967213E-03	-2.4255833E-03	-2.4255833E-03		5.7171534E-04	
20	5.1660640E-03	-3.5553572E-03	-3.5553572E-03		4.2140122E-03	
20014	9.6143110E-03	-4.4168635E-03	-4.4168635E-03		1.4046193E-03	
20018	9.6143110E-03	-7.0473233E-03	-7.0473233E-03		4.0943135E-03	
21	-8.7750070E-03	-3.4815035E-04	-3.4815035E-04		1.3870734E-03	
21019	-1.1540645E-02	-3.132706E-03	-3.132706E-03		2.372672E-03	
22	1.1920084E-02	-0.016768E-03	-0.016768E-03		7.372733E-03	
22020	0.0260243E-03	-7.3066331E-03	-7.3066331E-03		2.528440E-03	
23	1.5741611E-02	-1.0124793E-02	-1.0124793E-02		2.132571E-03	
23017	1.7563289E-03	-5.3626741E-03	-5.3626741E-03		3.3815556E-03	

MODAL POINT	DEFLECTIONS			LOAD CASE 1		
	UX	UY	UZ	UX	UY	UZ
23022	6.8456785E-03	-8.358810E-03	8.243331E-03			
24	1.4950785E-02	-1.738888E-02	1.320138E-02			
24018	1.6023437E-02	-1.1530498E-02	7.2378431E-03			
25	9.3847077E-03	-9.4083597E-03	9.015504E-03			
25023	4.7572811E-03	-3.1146975E-03	8.856003E-03			
26	2.3811748E-02	-1.2835115E-02	1.247652E-02			
26024	1.8947976E-02	-1.733574E-02	1.312241E-02			
27	-3.7743278E-03	-1.144423E-02	3.746276E-03			
27021	-8.5317552E-03	-1.4849798E-03	2.1452881E-03			
27025	5.7343423E-03	-8.7324325E-03	8.171701E-03			
28	1.4248730E-02	-1.7891484E-02	1.0303388E-02			
28022	1.2188580E-02	-1.1755332E-02	9.9512777E-03			
28026	2.2758118E-02	-1.3489312E-02	1.2888881E-02			
29	-6.3238010E-03	-1.0714148E-02	5.1736521E-03			
29027	-8.2481130E-03	-1.447140E-02	4.728843E-03			
30	1.3060498E-02	-1.3225970E-02	1.2487038E-02			
30028	1.4888791E-02	-1.3792861E-02	1.387911E-02			
31	8.3256107E-03	-1.0478274E-02	1.5773045E-02			
31025	7.2088386E-03	-3.1741880E-03	1.2727312E-02			
31030	2.3137753E-02	-1.8533128E-02	1.3982038E-02			
32	1.9550315E-02	-1.342508E-02	2.3274049E-02			
32026	4.1351489E-02	-1.472888E-02	1.8905025E-02			

ELEMENT				TAU 12		PARQUIN		LEG	
NO	I	J	K	SIGMA 1	SIGMA 2				
1	3	11	5	+8462.052 31750.415 377.257 3332.741	-3515.262 351.831 -500.648 -1231.643	-2220.102 1806.235 2727.222 -223.112	.435	1	
2	5	13	11	35061.031 54754.322 45347.761 43450.334	349.136 -1339.030 -109.294 -243.332	-5512.023 -3303.312 28.735 -3334.117	.468	1	
3	11	19	15	38756.302 39128.773 40495.441 35443.703	1070.242 -2109.409 -439.674 -324.206	-2315.314 -623.87 7101.700 307.236	.510	1	
4	13	21	13	25003.104 +2427.240 35781.015 35223.307	1204.131 -2340.430 -1356.405 -822.322	-5470.072 -1834.134 -339.134 -2702.347	.575	1	
7	1	9	7	09392.092 47503.031 -4400.701 3753.714	-7300.132 3313.259 704.77 -590.132	2415.041 0048.442 1303.325 3077.330	.285	1	
8	7	15	5	5650.744 40711.370 18071.335 39114.402	191.111 -661.558 -808.502 -1432.333	-3412.703 2825.113 4705.27 1393.117	.755	1	
9	9	17	15	-5055.274 18130.184 54151.727 2211.542	-1042.100 1154.329 -1334.922 -554.274	-8022.004 -3427.448 -2164.202 -4301.030	.231	1	
10	15	23	17	45512.103 20016.053 2592.050 30750.173	-1344.430 570.221 243.207 410.430	3431.744 2003.17 1324.095 2455.177	1.009	1	
13	4	12	5	45524.430 30313.408 17551.403 30223.125	-3434.141 2359.114 122.736 -324.077	2770.147 4508.73 2300.131 3401.430	.724	1	
14	6	14	12	37243.774 34224.931 343.2042 35370.217	-2308.002 330.914 -1074.707 -713.033	-1140.06 2515.133 -75.112 450.005	1.575	1	

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	L26
19	12	20	14	3744.0000 3501.0017 3547.9664 37400.222	-594.134 87.461 -911.328 -473.362	-331.23 1473.97 -400.937 77.075	1.042	1
10	14	22	20	3804.4440 38072.067 38070.332 38782.263	-300.030 160.201 -1039.210 -413.037	-473.92 1411.004 350.035 359.036	1.000	1
14	2	10	0	42473.272 28010.573 9612.592 28860.140	-0372.022 4702.075 -1920.092 -1035.759	-370.173 373.111 -1272.001 2272.513	.000	1
20	0	10	10	47300.243 48010.145 17225.126 33141.161	-2001.443 254.927 -339.034 -403.305	797.000 3550.023 -1319.022 1012.034	1.016	1
21	10	16	16	30654.533 29750.554 35053.501 30766.313	-751.409 002.414 -201.201 -36.039	1443.320 1232.950 -472.020 701.175	1.566	1
22	1	5	3	74477.740 -24291.442 56671.913 -632.421	13207.214 -13100.555 784.401 1750.539	-0170.049 -16541.230 12191.095 -3521.015	-043	2
20	2	0	4	-23517.340 17.063 3932.123 5297.345	11173.719 -0093.437 1953.551 2150.444	-1259.244 -18271.742 4635.722 -3240.302	-0704	2
27	4	7	0	-2410.073 -4752.501 12204.671 -6621.363	3443.020 -3906.570 467.473 -21.303	-4205.74 -3331.369 2503.020 -1037.245	-092	2
28	0	3	7	7607.344 36014.375 -5079.000 11541.024	3601.022 -4140.054 -304.407 -451.123	-7509.031 -4047.044 -015.123 -4091.070	.226	2
29	7	11	3	-26200.747 -17637.365 16051.000 -9775.349	2700.071 -1345.042 -2447.535 -362.493	-3745.016 -3106.416 2403.77 -1357.049	-0754	2

ELEMENT NO	J	K	Sigma 1	Sigma 2	TAU 12	MARGIN	LEG
30	3	13	11	-36145.002 -44420.003 35512.075 -15720.102	-4071.025 227.208 -252.072 214.304	20045	2
31	8	12	10	-24045.021 -4314.724 3232.114 -8021.011	3721.341 -2255.411 30.722 100.304	-0.721	2
32	10	14	10	-19224.070 -19444.060 -3303.907 -14400.114	1133.215 -2711.524 040.777 403.350	-0.044	2
33	12	13	14	-10000.370 -7331.522 -1633.120 -11354.752	378.103 -350.773 034.230 225.331	-0.005	2
34	14	17	13	-25331.077 -14710.003 -7414.767 -17007.309	153.044 1704.302 1103.350 1350.209	-0.725	2
35	13	13	17	-14735.229 -10035.433 -12075.003 -12054.904	-315.000 411.103 225.331 1350.209	-0.022	2
40	4	0	2	-21052.003 1727.004 23271.953 0435.003	15127.002 -13054.000 225.331 043.300	-0.070	2
47	3	7	1	-10123.757 5100.157 4303.003 10340.003	14117.333 -3311.400 2132.400 2247.004	-0.010	2
48	1	0	7	-20437.011 -3041.477 24440.053 -2443.203	4212.740 -4950.200 -400.003 -410.000	-0.700	2
49	7	12	0	40.103 16074.010 22725.003 13140.003	2130.004 -230.000 -1325.733 -0030.007	0.000	2

ELEMENT NO	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
20	0	10	10	10	2	1.070	2
			-2011.000	100.000	-3000.000		
			1100.000	-400.000	-2000.000		
			1100.000	170.000	-300.000		
			0000.000	-20.000	-2000.000		
21	12	10	10	10	1	0.000	1
			1012.000	100.000	-700.000		
			0000.000	-200.000	-2000.000		
			1100.000	110.000	300.000		
			1100.000	370.000	-1000.000		
22	0	11	10	10	2	1.000	2
			1000.000	-200.000	-700.000		
			1100.000	-000.000	400.000		
			-1000.000	300.000	300.000		
			0000.000	-700.000	1000.000		
23	11	10	10	10	3	1.000	3
			1000.000	000.000	-1000.000		
			1100.000	700.000	300.000		
			-1000.000	-700.000	1000.000		
			0000.000	-1000.000	1000.000		
24	0	10	10	10	2	0.000	2
			-2000.000	-200.000	-0100.000		
			1100.000	-20.000	-200.000		
			1100.000	-700.000	-1100.000		
			0000.000	-000.000	-2000.000		
25	10	10	10	10	3	1.000	3
			-2000.000	-200.000	0100.000		
			1100.000	200.000	2100.000		
			1100.000	100.000	200.000		
			0000.000	100.000	270.000		
26	10	10	10	10	2	-0.000	2
			0000.000	400.000	-2000.000		
			1100.000	-700.000	-400.000		
			-7000.000	700.000	-400.000		
			0000.000	100.000	-000.000		
27	10	10	10	10	2	-0.000	2
			1000.000	-200.000	0000.000		
			1100.000	200.000	300.000		
			-1000.000	-200.000	-1000.000		
			0000.000	100.000	000.000		
28	10	10	10	10	1	0.000	1
			1000.000	000.000	-4000.000		
			1100.000	-200.000	-1000.000		
			1100.000	-200.000	-200.000		
			0000.000	-200.000	-1000.000		
29	10	10	10	10	1	1.000	1
			0000.000	000.000	-3000.000		
			1100.000	-200.000	-200.000		
			1100.000	-200.000	-1000.000		
			0000.000	-200.000	-2000.000		

ELEMENT			SIGMA 1			SIGMA 2			SIGMA 3		
NO	I	J	K	L	M	N	O	P	Q	R	S
11	17	25	23			42980.800			42740.415		-305.202
						44244.243			-1427.312		4.113
						20130.174			-555.025		-884.012
						35891.072			-116.374		-323.720
12	23	31	25			27012.022			2490.323		-470.000
						44220.473			-4757.033		-2264.037
						18510.204			1317.023		-1796.004
						29349.083			-247.086		-3023.037
17	20	28	26			47107.010			413.072		-2223.005
						21314.070			-1465.076		432.330
						24054.030			-845.054		104.039
						40052.008			-553.021		-845.032
18	22	30	28			27001.030			3214.074		-433.074
						48020.014			-4933.002		-3900.002
						20970.040			330.002		-1713.077
						32211.053			-442.000		-3333.077
23	16	24	12			42303.042			-554.009		1315.024
						34509.044			230.002		2307.030
						28225.007			-88.033		-107.023
						34079.477			-103.007		117.030
23	18	26	24			30384.003			577.030		-1345.037
						38473.060			-1406.044		302.040
						24738.023			-499.040		-927.030
						33200.023			-420.055		-441.000
24	24	32	20			15335.010			1107.053		-1033.074
						23.01475			-201.050		-1301.023
						31758.183			-477.024		-237.023
						23491.051			5.031		-1497.005
36	17	21	15			-10640.000			-204.043		1435.034
						-37055.074			3410.004		310.077
						10730.071			331.030		723.026
						-12058.050			1053.073		4443.002
37	10	20	18			-7573.000			213.000		-770.000
						-225.045			-1320.050		-319.000
						-212.043			425.020		175.037
						-2043.030			461.024		-1230.030
38	18	22	20			-14113.030			233.070		230.004
						-13430.021			-128.077		-307.030
						4057.007			200.042		242.034
						-7027.014			223.014		-423.030

ELEMENT NO	J	K	SIGMA 1	SIGMA 2	140 12	MARGIN	LEG
39	20	23	22	-3350.224 -3023.372 -6506.575 -4439.530	-3344.477 473.934 507.238 242.232	507.72 -255.41 -122.536 -94.339	2
40	22	25	23	-2036.467 -9205.011 -5565.453 -5165.137	-221.430 545.454 518.237 257.373	1434.577 77.525 225.244 532.102	2
41	23	27	25	-4112.574 -2255.975 17182.773 -3152.151	-2410.344 3753.302 -1404.326 -43.307	423.226 323.459 1784.521 3172.235	2
42	25	29	27	11872.014 -1050.001 -16322.730 -2253.567	-4311.275 3150.740 1634.954 -45.950	2502.222 323.235 238.432 2300.73	2
43	24	28	26	11245.051 -10350.304 11500.215 4129.394	-3556.739 3041.157 -258.443 -211.359	5138.269 633.242 -1372.314 3300.730	2
44	26	30	28	5070.700 -15715.335 490.451 -3182.004	-3750.613 3323.567 307.172 -38.027	2453.447 8314.595 -2506.431 3130.73	2
45	28	31	30	-803 -21607.350 39438.126 5942.052	-501.034 3209.008 1154.819 1520.751	11345.055 2713.204 -5390.132 2800.72	2
57	20	24	18	-12760.274 -12056.001 13187.043 -4145.812	706.271 -354.820 173.525 183.009	845.73 -304.344 -103.714 -25.258	2
59	19	23	17	9730.500 6018.215 4578.323 6572.000	113.710 -1506.330 380.361 809.254	1114.705 -1242.775 2451.43 774.33	2
60	17	22	23	-3708.721 178.234 11025.740 2765.087	1589.040 -603.500 -256.435 307.533	504.777 -1051.258 -332.40 -720.226	2

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
01	23	20	26	2052.003 -663.797 1350.112 3650.774	-731.034 1051.017 -153.462 55.241	2046.040 1013.022 5350.004 1309.045	.122	3
62	22	26	20	12176.200 -0149.015 23560.772 9005.250	-3351.090 3752.020 -1022.023 -203.795	3300.001 4407.020 870.751 3054.713	.132	3
03	20	32	20	14203.093 3533.520 -13040.550 2025.405	-3326.403 2412.091 1403.003 150.397	2404.014 2333.043 -1301.072 1007.043	-.034	3
04	21	25	27	35735.272 3709.074 -1083.020 12520.541	-707.077 4050.321 -319.575 -495.047	7020.041 9327.024 -1494.004 5123.040	.037	3
05	27	31	25	13763.001 -7419.550 5500.173 3570.305	-3700.000 3235.953 57.905 -1433.003	3045.027 0050.039 -3553.030 3323.042	-.002	3
00	25	30	31	050.152 -22879.121 43753.000 7243.500	-7727.970 3994.050 1237.130 1534.005	12300.700 3422.014 -0024.005 3101.036	-.050	3

OPTIMIZATION STEPS

IK =	0	SUBSTEPS =	0	STEPS =	2	CYCLES =	0		
WNT =	.00712	MARGIN =		-.64475		M(I) =	.75000		.50000
Q.00000		Q.00000	0.00000		Q.00000		Q.00000		.50000
MARGIN(I) =	.07935		-.64475		-.01063				
	.04267		.04267		.04267				

IK =	0	SIDESTEPS =	0	STEPS =	3	CYCLES =	0
WT =	0.3361	MARGIN =	-0.5115	M(1) =	0.5000	0.7115	0.04207
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.04207
MARGIN(1) =	0.3375	-0.50115	-0.0349				
0.05336	0.05336	0.05336	0.05336				

IK = 0	SUBSTEPS = 0	STEPS = 4	CYCLES = 3	
WT = .02127	MARGIN =	.34903	HLI =	.75000
0.00000	0.00000	0.00000	0.00000	.50000
MARGIN(1) =	1.50910	.07359	.04903	
NEW OPTIMUM				

.88939
 .05336
 .05336
 .05336
 0.00000
 0.00000

0.00000	
0.00000	
0.00000	
.50000	
CALCULATION OF DERIVATIVES	
.05316	.05316
.05316	.05316
.05316	.05316

```

IK = 2 SIDESTEPS = 0 STEPS = 5 CYCLES = 1
WT = .02124 MARGIN = .04971 M(1) = .05949
3.00000 3.00000 0.00000 .00000
MARGIN(1) = 1.26049 .07130 .04971 .02227
-31303 .07383
-68604

```

```

      IK = 3      STVE STEPS = 0      STEPS = 0      CYCLES = 1
      WT = .0141  MARGIN = .0525      M(1) = .7000      .05330
      J = .00000      0.00000      0.00000      0.00000      .05430
      MARGIN(1) = 1.00000      .0715      .0525      .00000      .05330

```

```
IK = 4 SIDESTEPS = 0 STEPS = 7 CIRCLES = 1  
MT = -.0122 MARGIN = .00000 MI() = .75000  
.00000 0.00000 6.00000 .50000  
99.07829 .00000 .00000 .00000  
99.07829 .00000 .00000 .00000
```

9.00997	10.11690	19.04437	0.25000	20002
---------	----------	----------	---------	-------

[illegible]

SIDE SLEEPING
DIRECTION COSINES OF CONST. WT. SURFACE

, DIR. COSINES OF COMPOSITE CONSTRAINT SURFACE

IK = 12	SIDESTEPS = 1	STEPS = 23	CYCLES = 2
WT =	.01030	MARGIN = -.00157	M(1) = .05000
0.00000	0.00000	0.00000	0.00000
MARGIN(1) =	-.00157	.00731	.00317
.02230	.00829	.00323	

IK = 12	SIDESTEPS = 2	STEPS = 24	CYCLES = 2
WT =	.01032	MARGIN = -.00244	M(1) = .05000
0.00000	0.00000	0.00000	0.00000
MARGIN(1) =	-.00244	.01153	.01742
DESCENDING			
.02231	.00843	.00343	

IK = 12	SIDESTEPS = 1	STEPS = 31	CYCLES = 3		
MT =	.01033	MARGIN =	-.00009	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	-.00000		.00000		.00000
	.00240		.00000		.00000
				.75132	.00239
					.00924
					.05924
IK = 12	SIDESTEPS = 2	STEPS = 31	CYCLES = 3		
MT =	.01034	MARGIN =	.00007	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00307		.00000		.00000
	.00235		.00000		.00000
				.75423	.00240
					.00900
					.05900
IK = 12	SIDESTEPS = 3	STEPS = 32	CYCLES = 3		
MT =	.01031	MARGIN =	-.00012	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	-.00012		.00000		.00000
	.00240		.00000		.00000
				.75043	.00239
					.00934
					.05934
DESCENDING					
IK = 12	SIDESTEPS = 4	STEPS = 33	CYCLES = 3		
MT =	.01034	MARGIN =	.00007	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00307		.00000		.00000
	.00240		.00000		.00000
				.75429	.00240
					.00900
					.05900
CALCULATION OF DERIVATIVES					
IK = 2	SIDESTEPS = 0	STEPS = 34	CYCLES = 4		
MT =	.01030	MARGIN =	.00000	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00000		.00000		.00000
	.00240		.00000		.00000
				.75029	.00240
					.00900
					.05900
IK = 3	SIDESTEPS = 0	STEPS = 35	CYCLES = 4		
MT =	.01047	MARGIN =	.00000	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00000		.00000		.00000
	.00240		.00000		.00000
				.75429	.00234
					.00900
					.05900
IK = 4	SIDESTEPS = 0	STEPS = 36	CYCLES = 4		
MT =	.01030	MARGIN =	.00000	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00000		.00000		.00000
	.00240		.00000		.00000
				.75429	.00240
					.00900
					.05900
IK = 5	SIDESTEPS = 0	STEPS = 37	CYCLES = 4		
MT =	.01030	MARGIN =	.00000	M(1) =	.75000
	.00000		.00000		.00000
MARGIN(1) =	.00000		.00000		.00000
	.00240		.00000		.00000
				.75429	.00240
					.00900
					.05900

Reproduced from
best available copy.

SIDE STEPPING
DIRECTION COSINES OF CONST. MT. SURFACE
+ CONSTRAINT SURFACES


```

MT = .01045 MARGIN = .03441 M(1) = .70422 .J2301 .05901
J.00000 MARGIN(1) = .09341 .10757 .13053 .00000 .00000
.02310 .05903

IK = 12 SIDE STEPS = 8 STEPS = 40 CYCLES = 4
MT = .01045 MARGIN = .11090 M(1) = .70422 .J2310 .05903
J.00000 MARGIN(1) = .00000 .00000 .13709 .00000 .00000
.02313 .05905

IK = 12 SIDE STEPS = 9 STEPS = 40 CYCLES = 4
MT = .01045 MARGIN = .12000 M(1) = .70422 .J2315 .05905
J.00000 MARGIN(1) = .00000 .00000 .13044 .00000 .00000
.02320 .05907

IK = 12 SIDE STEPS = 10 STEPS = 47 CYCLES = 4
MT = .01045 MARGIN = .13440 M(1) = .70422 .J2320 .05907
J.00000 MARGIN(1) = .00000 .00000 .13115 .00000 .00000
DESCENDING .02313 .05905

IK = 12 SIDE STEPS = 11 STEPS = 43 CYCLES = 4
MT = .01045 MARGIN = .12000 M(1) = .70422 .J2315 .05905
J.00000 MARGIN(1) = .00000 .00000 .13944 .00000 .00000
.02214 .05913

IK = 12 SIDE STEPS = 11 STEPS = 43 CYCLES = 4
MT = .01045 MARGIN = .10175 M(1) = .70422 .J2214 .05913
J.00000 MARGIN(1) = .00000 .00000 .00220 .00000 .00000
NEW OPTIMUM .70000
.72000
.02214
.05913
J.00000
J.00000
J.00000
J.00000
.00000
.00000

```


OPTIMIZED LAYER THICKNESSES

1	.75000
2	.77850
3	.02214
4	.05513
5	.05513
6	0.00000
7	0.00000
8	0.00000
9	0.00000
10	0.00000
11	.50000

FLAT PANEL WIDTH = 2.10020

LENGTH = 2.02500

FIRST VERTICAL PLATE = 1 SKEN A = 25 SKEN B = 40 SKEN C = 0 UPPER FACE = 0

NUMBER OF NODAL POINTS-----	32
NUMBER OF ELEMENTS-----	60
NUMBER OF LOAD CASES	1

NGUAL POINT	X	Y	Z
1	0.0000	0.0000	0.0000
2	0.0000	1.0000	0.0000
3	0.0000	0.0000	0.7790
4	0.0000	0.0000	0.7790
5	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000
24	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000
28	0.0000	0.0000	0.0000
29	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000
32	0.0000	0.0000	0.0000

NEW OLD			LCAD CASE 1		
NOUE	MODE		UX	UY	UZ
1	1	-60.1740	-0.0173	0.0000	0.0000
2	2	-22.0029	-0.0173	0.0000	0.0000
3	3	-07.4002	-0.0173	0.0000	0.0000
4	3001	-09.0029	-0.0173	0.0000	0.0000
5	4	-60.1740	-0.0173	0.0000	0.0000
6	4002	-29.0029	-0.0173	0.0000	0.0000
7	5	-07.4002	-0.0173	0.0000	0.0000
8	5003	-29.0029	-0.0173	0.0000	0.0000
9	6	-60.1740	-0.0173	0.0000	0.0000
10	6004	-29.0029	-0.0173	0.0000	0.0000
11	7	-60.1740	-0.0173	0.0000	0.0000
12	7001	-29.0029	-0.0173	0.0000	0.0000
13	7000	-29.0029	-0.0173	0.0000	0.0000
14	8	-22.0029	-0.0173	0.0000	0.0000
15	8002	-22.0029	-0.0173	0.0000	0.0000
16	9	0.0000	0.0000	0.0000	0.0000
17	9007	0.0000	0.0000	0.0000	0.0000
18	10	7.2917	0.0000	0.0000	0.0000
19	10009	0.0000	0.0000	0.0000	0.0000
20	11	-7.2917	0.0000	0.0000	0.0000
21	11005	-29.0029	0.0000	0.0000	0.0000
22	11009	0.0000	0.0000	0.0000	0.0000
23	12	0.0000	0.0000	0.0000	0.0000
24	12000	0.0000	0.0000	0.0000	0.0000
25	12010	0.0000	0.0000	0.0000	0.0000
26	13	-7.2917	0.0000	0.0000	0.0000
27	13011	-29.0029	0.0000	0.0000	0.0000
28	14	0.0000	0.0000	0.0000	0.0000
29	14012	0.0000	0.0000	0.0000	0.0000
30	15	0.0000	0.0000	0.0000	0.0000
31	15009	0.0000	0.0000	0.0000	0.0000
32	15014	0.0000	0.0000	0.0000	0.0000
33	16	7.2917	0.0000	0.0000	0.0000
34	16010	29.0029	0.0000	0.0000	0.0000
35	17	0.0000	0.0000	0.0000	0.0000
36	17012	0.0000	0.0000	0.0000	0.0000
37	18	7.2917	0.0000	0.0000	0.0000
38	18016	29.0029	0.0000	0.0000	0.0000
39	19	-7.2917	0.0000	0.0000	0.0000
40	19013	-29.0029	0.0000	0.0000	0.0000
41	19017	0.0000	0.0000	0.0000	0.0000
42	20	0.0000	0.0000	0.0000	0.0000
43	20014	0.0000	0.0000	0.0000	0.0000
44	20010	0.0000	0.0000	0.0000	0.0000
45	21	-7.2917	0.0000	0.0000	0.0000
46	21019	-29.0029	0.0000	0.0000	0.0000
47	22	0.0000	0.0000	0.0000	0.0000
48	22020	0.0000	0.0000	0.0000	0.0000
49	23	0.0000	0.0000	0.0000	0.0000
50	23017	0.0000	0.0000	0.0000	0.0000
51	23022	0.0000	0.0000	0.0000	0.0000
52	24	7.2917	0.0000	0.0000	0.0000

23	24018	23.1567	0.0001	0.0000	0.0000
24	25	60.1746	6.0175	0.0000	0.0000
25	25023	0.0000	0.0000	0.0000	0.0000
26	26	67.4662	0.0175	0.0000	0.0000
27	26024	23.1567	0.0000	0.0000	0.0000
28	27	52.4629	0.0175	0.0000	0.0000
29	27021	0.0000	0.0000	0.0000	0.0000
30	27025	211.5317	24.0698	0.0000	0.0000
31	28	60.1746	0.0175	0.0000	0.0000
32	28022	0.0000	0.0000	0.0000	0.0000
33	28026	240.6983	24.0698	0.0000	0.0000
34	29	52.4629	0.0175	0.0000	0.0000
35	29027	211.5317	24.0698	0.0000	0.0000
36	30	60.1746	0.0175	0.0000	0.0000
37	30024	240.6983	24.0698	0.0000	0.0000
38	31	60.1746	0.0175	0.0000	0.0000
39	31025	240.6983	24.0698	0.0000	0.0000
40	31030	240.6983	24.0698	0.0000	0.0000
41	32	67.4662	0.0175	0.0000	0.0000
42	32026	269.8650	24.0698	0.0000	0.0000

ELEMENT NUMBER	I	J	K	EX	EY	EXY	G	F	CF	LU	LR	LS	LT
1	3	11	5	4031496	203937	125984	200000	0.221	3	20	7	21	9
2	5	13	11	4031496	203937	125984	200000	0.221	7	20	20	27	21
3	11	19	13	4031496	203937	125984	200000	0.221	20	35	40	40	27
4	13	21	19	4031496	203937	125984	200000	0.221	20	35	39	46	40
5	19	27	21	4031496	203937	125984	200000	0.221	39	50	35	39	40
6	21	29	27	4031496	203937	125984	200000	0.221	45	50	50	65	50
7	1	9	7	4031496	203937	125984	200000	0.221	1	10	11	17	12
8	7	15	9	4031496	203937	125984	200000	0.221	11	30	10	31	17
9	9	17	15	4031496	203937	125984	200000	0.221	10	39	39	30	31
10	15	23	17	4031496	203937	125984	200000	0.221	10	49	35	20	30
11	17	25	23	4031496	203937	125984	200000	0.221	45	54	45	25	50
12	23	31	25	4031496	203937	125984	200000	0.221	49	60	54	05	50
13	4	12	6	4031496	203937	125984	200000	0.221	3	20	23	25	24
14	6	14	12	4031496	203937	125984	200000	0.221	23	47	20	43	29
15	12	20	14	4031496	203937	125984	200000	0.221	23	47	42	48	43
16	14	22	23	4031496	203937	125984	200000	0.221	42	61	61	67	62
17	20	28	22	4031496	203937	125984	200000	0.221	47	60	61	67	62
18	22	30	28	4031496	203937	125984	200000	0.221	2	10	14	15	15
19	2	10	6	4031496	203937	125984	200000	0.221	2	10	14	15	15
20	6	16	10	4031496	203937	125984	200000	0.221	14	33	18	34	15
21	10	18	16	4031496	203937	125984	200000	0.221	19	37	32	30	34
22	16	24	18	4031496	203937	125984	200000	0.221	33	52	37	53	38
23	18	26	24	4031496	203937	125984	200000	0.221	30	50	36	27	23
24	24	32	26	4031496	203937	125984	200000	0.221	71	56	72	57	
25	1	5	3	4031496	203937	125984	200000	0.221	1	7	3	8	4
26	5	6	4	4031496	203937	125984	200000	0.221	2	9	5	10	5
27	4	7	6	4031496	203937	125984	200000	0.221	3	14	5	15	10
28	6	9	7	4031496	203937	125984	200000	0.221	3	10	11	17	13
29	7	11	9	4031496	203937	125984	200000	0.221	14	20	16	22	17
30	9	13	11	4031496	203937	125984	200000	0.221	10	20	20	27	22
31	8	12	10	4031496	203937	125984	200000	0.221	14	20	19	25	15
32	10	14	12	4031496	203937	125984	200000	0.221	14	20	23	25	20
33	12	15	14	4031496	203937	125984	200000	0.221	23	30	26	32	24
34	14	17	15	4031496	203937	125984	200000	0.221	28	35	35	36	32
35	15	19	17	4031496	203937	125984	200000	0.221	30	39	35	41	38
36	17	21	19	4031496	203937	125984	200000	0.221	45	45	35	40	41
37	16	20	18	4031496	203937	125984	200000	0.221	33	42	37	44	30
38	18	22	20	4031496	203937	125984	200000	0.221	37	47	42	43	44
39	20	23	22	4031496	203937	125984	200000	0.221	42	49	45	51	48
40	22	25	23	4031496	203937	125984	200000	0.221	47	54	45	55	51
41	23	27	25	4031496	203937	125984	200000	0.221	49	50	54	60	50
42	25	29	27	4031496	203937	125984	200000	0.221	54	64	50	65	60
43	24	28	26	4031496	203937	125984	200000	0.221	22	61	50	63	57
44	26	30	28	4031496	203937	125984	200000	0.221	50	60	51	67	63
45	28	31	30	4031496	203937	125984	200000	0.221	61	60	55	70	67
46	4	8	2	4031496	203937	125984	200000	0.221	3	14	2	15	6
47	3	7	1	4031496	203937	125984	200000	0.221	3	11	1	12	4
48	1	6	7	4031496	203937	125984	200000	0.221	1	9	11	13	12
49	7	12	6	4031496	203937	125984	200000	0.221	11	23	9	24	13
50	6	10	12	4031496	203937	125984	200000	0.221	9	10	15	25	24

ELEMENT NUMBER	I	J	K	EA	CF	LAY	G	T	LF	LV	LR	LS	LI
51	12	19	19	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
52	13	19	11	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
53	11	19	15	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
54	13	14	15	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
55	19	20	14	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
56	14	10	20	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
57	21	24	19	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
58	13	17	15	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
59	17	22	17	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
60	17	22	23	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
61	23	26	26	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
62	22	26	26	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
63	28	32	26	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
64	21	29	27	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
65	27	31	27	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0
66	25	30	31	+0.11436	0.00000	12.0000	0.00000	0.00000	0.0	0.0	0.0	0.0	0.0

LEU DUCKLING STICKS
/225

44224

DEFLECTIONS LOAD CASE 1

MODAL POINT	U1	U2
1	-1.31891005-04	0.21113335-03
2	-2.33636035-03	-2.37723005-04
3	-2.68730865-02	3.68719335-03
30J1	-2.70478835-02	0.01100135-03
4	-1.20454605-02	-4.44390914-03
40J2	-1.08976405-02	-4.44390914-03
5	-1.33785305-02	5.58124005-03
50J3	-1.0053295-02	1.12643145-02
6	-3.21758635-03	-1.16003385-04
60J4	-7.24332155-03	1.42717345-03
7	-1.32846045-02	-1.02447315-03
70J1	-0.91635485-03	-5.7471465-03
70J6	-1.10134455-02	-1.13362445-03
8	-1.02730755-02	-7.44332305-03
80J2	-2.28681075-02	-1.16113305-02
9	-2.71519325-03	3.62603315-03
90J7	-3.08015515-03	1.62290455-03
10	3.04166525-03	-3.74920845-03
100J8	-3.25614335-04	-3.74920845-03
11	-1.43006735-02	6.07050455-04
110J5	-1.59310005-02	5.25212005-03
110J9	-7.34522815-03	0.02753335-04
12	-7.33640525-03	-0.27633005-03
120J6	-4.03767005-03	-0.18033225-03
120J10	-3.62334405-03	-4.03390335-03
13	-5.30423725-03	7.43036755-04
130J11	-1.21621305-02	-2.1143395-03
14	-0.45603425-03	-2.01023935-03
140J12	-1.88611935-03	-3.33222935-03
15	-0.43340455-03	-0.00477115-03
150J9	-3.31330655-03	-1.22443075-03
150J14	-1.42476035-03	-2.54432235-03
16	-1.77113025-03	-3.14327335-03
160J10	2.25612425-04	-7.0773635-03
17	2.53187245-03	-5.25033315-03
170J15	-1.10260315-03	-1.74303305-03
18	5.03456755-03	-0.71829115-03
180J16	4.39603335-03	-0.06404455-03
19	-7.70075035-03	-2.43632405-03
190J13	-8.74306755-03	-4.41003155-03
190J17	-1.43306295-03	-1.12297795-03
20	-3.03221225-04	-0.72010215-03
200J14	1.14449875-03	-4.07103715-03
200J18	3.42384315-03	-0.04100325-03
21	7.36129315-04	1.21303215-03
210J19	-5.3302235-03	-3.34390255-03
22	6.36567745-03	-3.26762555-03
220J20	4.10846725-03	-2.26567105-03
23	-1.49740335-04	-0.02347055-03
230J17	1.30116015-03	-3.26144455-03
240J18	1.34502425-03	1.34502425-03
250J19	4.03293805-03	4.03293805-03
260J20	1.10164225-03	1.10164225-03
270J21	4.33973955-03	4.33973955-03
280J22	1.16264755-03	1.16264755-03
290J23	5.72401155-03	5.72401155-03
300J24	4.43843655-03	4.43843655-03
310J25	4.33973955-03	4.33973955-03
320J26	4.93163015-03	4.93163015-03
330J27	4.14157405-03	4.14157405-03
340J28	4.43843655-03	4.43843655-03
350J29	6.32371955-03	6.32371955-03

Reproduced from
Best available copy.

DEFLECTIONS

LOAD CASE 1

NODAL POINT	UJ	UJ	UJ
23022	4.2242-03	-1.7122342-02	5.2263.3/2-02
24018	7.7249474-03	-7.7249474-03	4.2263.3/2-02
25023	4.2249474-03	-2.3042222-03	4.2263.3/2-02
26024	3.2249474-03	-2.3042222-03	4.2263.3/2-02
27025	1.2249474-03	-2.3042222-03	4.2263.3/2-02
28026	1.2249474-03	-2.3042222-03	4.2263.3/2-02
29027	1.2249474-03	-2.3042222-03	4.2263.3/2-02
30028	1.2249474-03	-2.3042222-03	4.2263.3/2-02
31029	1.2249474-03	-2.3042222-03	4.2263.3/2-02
32030	1.2249474-03	-2.3042222-03	4.2263.3/2-02
33031	1.2249474-03	-2.3042222-03	4.2263.3/2-02
34032	1.2249474-03	-2.3042222-03	4.2263.3/2-02
35033	1.2249474-03	-2.3042222-03	4.2263.3/2-02
36034	1.2249474-03	-2.3042222-03	4.2263.3/2-02
37035	1.2249474-03	-2.3042222-03	4.2263.3/2-02
38036	1.2249474-03	-2.3042222-03	4.2263.3/2-02
39037	1.2249474-03	-2.3042222-03	4.2263.3/2-02
40038	1.2249474-03	-2.3042222-03	4.2263.3/2-02
41039	1.2249474-03	-2.3042222-03	4.2263.3/2-02
42040	1.2249474-03	-2.3042222-03	4.2263.3/2-02
43041	1.2249474-03	-2.3042222-03	4.2263.3/2-02
44042	1.2249474-03	-2.3042222-03	4.2263.3/2-02
45043	1.2249474-03	-2.3042222-03	4.2263.3/2-02
46044	1.2249474-03	-2.3042222-03	4.2263.3/2-02
47045	1.2249474-03	-2.3042222-03	4.2263.3/2-02
48046	1.2249474-03	-2.3042222-03	4.2263.3/2-02
49047	1.2249474-03	-2.3042222-03	4.2263.3/2-02
50048	1.2249474-03	-2.3042222-03	4.2263.3/2-02
51049	1.2249474-03	-2.3042222-03	4.2263.3/2-02
52050	1.2249474-03	-2.3042222-03	4.2263.3/2-02
53051	1.2249474-03	-2.3042222-03	4.2263.3/2-02
54052	1.2249474-03	-2.3042222-03	4.2263.3/2-02
55053	1.2249474-03	-2.3042222-03	4.2263.3/2-02
56054	1.2249474-03	-2.3042222-03	4.2263.3/2-02
57055	1.2249474-03	-2.3042222-03	4.2263.3/2-02
58056	1.2249474-03	-2.3042222-03	4.2263.3/2-02
59057	1.2249474-03	-2.3042222-03	4.2263.3/2-02
60058	1.2249474-03	-2.3042222-03	4.2263.3/2-02
61059	1.2249474-03	-2.3042222-03	4.2263.3/2-02
62060	1.2249474-03	-2.3042222-03	4.2263.3/2-02
63061	1.2249474-03	-2.3042222-03	4.2263.3/2-02
64062	1.2249474-03	-2.3042222-03	4.2263.3/2-02
65063	1.2249474-03	-2.3042222-03	4.2263.3/2-02
66064	1.2249474-03	-2.3042222-03	4.2263.3/2-02
67065	1.2249474-03	-2.3042222-03	4.2263.3/2-02
68066	1.2249474-03	-2.3042222-03	4.2263.3/2-02
69067	1.2249474-03	-2.3042222-03	4.2263.3/2-02
70068	1.2249474-03	-2.3042222-03	4.2263.3/2-02
71069	1.2249474-03	-2.3042222-03	4.2263.3/2-02
72070	1.2249474-03	-2.3042222-03	4.2263.3/2-02
73071	1.2249474-03	-2.3042222-03	4.2263.3/2-02
74072	1.2249474-03	-2.3042222-03	4.2263.3/2-02
75073	1.2249474-03	-2.3042222-03	4.2263.3/2-02
76074	1.2249474-03	-2.3042222-03	4.2263.3/2-02
77075	1.2249474-03	-2.3042222-03	4.2263.3/2-02
78076	1.2249474-03	-2.3042222-03	4.2263.3/2-02
79077	1.2249474-03	-2.3042222-03	4.2263.3/2-02
80078	1.2249474-03	-2.3042222-03	4.2263.3/2-02
81079	1.2249474-03	-2.3042222-03	4.2263.3/2-02
82080	1.2249474-03	-2.3042222-03	4.2263.3/2-02
83081	1.2249474-03	-2.3042222-03	4.2263.3/2-02
84082	1.2249474-03	-2.3042222-03	4.2263.3/2-02
85083	1.2249474-03	-2.3042222-03	4.2263.3/2-02
86084	1.2249474-03	-2.3042222-03	4.2263.3/2-02
87085	1.2249474-03	-2.3042222-03	4.2263.3/2-02
88086	1.2249474-03	-2.3042222-03	4.2263.3/2-02
89087	1.2249474-03	-2.3042222-03	4.2263.3/2-02
90088	1.2249474-03	-2.3042222-03	4.2263.3/2-02
91089	1.2249474-03	-2.3042222-03	4.2263.3/2-02
92090	1.2249474-03	-2.3042222-03	4.2263.3/2-02
93091	1.2249474-03	-2.3042222-03	4.2263.3/2-02
94092	1.2249474-03	-2.3042222-03	4.2263.3/2-02
95093	1.2249474-03	-2.3042222-03	4.2263.3/2-02
96094	1.2249474-03	-2.3042222-03	4.2263.3/2-02
97095	1.2249474-03	-2.3042222-03	4.2263.3/2-02
98096	1.2249474-03	-2.3042222-03	4.2263.3/2-02
99097	1.2249474-03	-2.3042222-03	4.2263.3/2-02
100098	1.2249474-03	-2.3042222-03	4.2263.3/2-02

Reproduced from
best available copy.

RECEIVED
NO 1 J K

SLOPA 1

SLOPA 2

TAB 12

MARGIN

156

1 5 11 5

34874004
40111117
20000047
30000044

-11500007
-00000000
-10000007
-10000001

-01430000
-00000000
-00000000
-00000000

0.544

1

2 5 13 11

32277007
31047000
37000000
40000000

20000000
-00000000
-00000000
-00000000

-07230037
-00000000
-00000000
-00000000

0.000

1

3 11 19 13

34133400
37000000
37000000
37000000

00000000
-00000000
-00000000
-00000000

-00000000
-00000000
-00000000
-00000000

0.000

1

4 13 21 15

24474000
40000000
35000000
35000000

00000000
-00000000
-00000000
-00000000

-07134071
-00000000
-00000000
-00000000

0.434

1

5 1 5 7

05127000
05000000
05000000
05000000

-00000000
-00000000
-00000000
-00000000

00000000
00000000
00000000
00000000

-0.011

1

6 7 15 7

40127000
40000000
40000000
40000000

00000000
-00000000
-00000000
-00000000

-00000000
-00000000
-00000000
-00000000

1.040

1

7 9 17 15

04910007
20000000
20000000
20000000

-00000000
-00000000
-00000000
-00000000

-00000000
-00000000
-00000000
-00000000

1.175

1

8 15 23 17

30150000
20000000
20000000
20000000

-00000000
-00000000
-00000000
-00000000

24000000
00000000
00000000
00000000

1.000

1

9 4 12 0

40000000
20000000
30000000
30000000

-00000000
-00000000
-00000000
-00000000

01000000
00000000
00000000
00000000

0.400

1

10 0 14 12

30000000
30000000
30000000
30000000

-00000000
-00000000
-00000000
-00000000

-00000000
-00000000
-00000000
-00000000

0.130

1

Reproduced from
best available copy.

DOCUMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
10	12	20	14	1117.142	-245.693	017.442	1.043	1
				245.693	017.442	775.74		
				245.693	-37.646	3.333		
				345.693	-0.077	400.224		
10	14	22	20	1117.142	145.340	-500.725	1.151	1
				332.740	-42.934	-003.737		
				310.340	-11.047	-200.236		
				324.340	1.332	-340.204		
10	2	10	0	37011.734	-200.241	200.241	1.171	1
				2233.734	310.437	0015. 03		
				1005.313	-205.074	-779.241		
				2355.304	-1700.053	3274. 34		
20	5	10	10	45231.700	-1275.515	2101.072	1.740	1
				41300.745	1770.132	4000.736		
				9200.745	-05.107	-1200.234		
				3322.745	-050.473	1310.433		
21	10	10	10	3720.314	-14.441	2075.496	1.511	1
				2540.052	173.040	2347.496		
				2005.052	151.934	-1257.522		
				2222.745	100.324	1250. 73		
22	1	5	3	1722.707	041.270	270.496	1.531	2
				111.111	-4577.707	210.515		
				217.717	17.171	010.000		
				217.717	113.043	-170.171		
22	2	0	4	1722.707	340.000	1331. 71	1.551	2
				112.643	-130.446	-007.707		
				111.111	230.000	2330.002		
				245.693	103.309	-100. 30		
22	4	7	0	110.001	244.147	-120.147	1.554	2
				010.001	-200.147	-2345.129		
				100.001	37.147	170.126		
				-300.001	245.337	-770.336		
22	0	6	7	111.107	131.107	-537.033	1.752	2
				040.240	-230.033	000.000		
				1300.204	-300.011	000.000		
				0077.344	-037.320	-130.004		
22	7	11	6	1420.000	-15.240	-310.240	1.144	2
				-500.000	-000.000	-120.240		
				1411.011	-245.400	130.000		
				-3145.000	-050.034	-130.000		

ELEMENT NO	J K			SIGMA 1	SIGMA 2	FAJ 12	MARGIN	LEO
	1	2	3					
30	9	13	11	-10024.930 -17259.343 18237.429 -6052.090	-2411.404 2177.004 -327.714 -103.030	-1240.204 -021.220 1400.030 -103.031	1.300	2
31	8	12	10	-9437.190 297.071 -558.077 -3232.521	-014.006 -040.940 -371.000 -010.309	-432.025 -1775.140 -1507.024 -2507.030	1.000	2
32	10	14	12	-10015.022 -10270.180 447.375 -6020.011	207.319 -223.053 -57.402 2.020	37.141 51.020 91.031 700.64	0.002	2
33	12	15	14	-4324.030 -3480.409 -6127.300 -4040.090	-43.021 -13.417 -00.570 -14.007	-190.700 -010.30 -273.037 -201.070	0.010	2
34	14	17	15	-11544.031 -0362.272 -1049.030 -7180.010	00.933 1791.212 240.094 00.013	144.009 -2392.032 -457.103 -012.009	2.700	2
35	15	19	17	-4210.040 -7135.123 -5404.022 -5613.735	-535.004 -70.191 3025.107 000.731	200.001 140.134 -225.034 074.017	1.000	2
40	4	0	0	-0554.100 0000.354 0551.934 2108.715	7397.430 -234.272 -574.740 14.402	-3294.037 -3050.049 3397.042 -2041.030	0.000	2
47	3	7	1	-6953.002 082.143 1802.060 4003.017	4300.774 -2510.040 2055.117 1707.002	-1127.430 -7300.031 3374.077 -1337.037	0.500	2
48	1	0	7	-14422.401 -5554.400 17203.119 -1027.017	2354.000 -2253.100 -20.031 34.043	-3500.047 -3711.030 1791.041 -1020.014	1.000	2
49	7	12	0	-512.400 4535.320 14013.742 6012.213	007.007 -550.337 -1103.204 -400.133	-3391.037 -1051.022 1065.071 -1220.000	2.000	2



Reproduced from
best available copy.

ELEMENT NO		J K		SIGMA 1		SIGMA 2		140 12		MARGIN		120	
50	6	10	12	-3059.999	-194.728	-3066.294				1.621		3	
				6006.094	-703.714	-2227.693							
				7023.743	-331.517	-146.670							
				3323.681		-2107.736							
51	12	16	10	398.211	710.419	-1373.509				5.571		3	
				3810.583	-1118.423	-1226.741							
				4.064	225.190	344.391							
				1337.954	-26.950	-350.305							
52	5	9	11	3740.187	-2418.413	-2311.000				2.487		3	
				3015.412	-1453.093	2307.114							
				-3305.199	194.588	-228.333							
				1152.147	-1451.506	145.440							
53	11	15	4	6751.021	310.340	-29.727				2.520		3	
				2153.901	226.647	2458.323							
				-1923.274	-345.732	1253.083							
				2308.649	-984.246	1253.026							
54	9	16	13	-6169.887	-2591.633	-4305.745				1.364		3	
				2096.780	-80.909	-85.434							
				3243.920	-434.053	-1332.120							
				-240.724	-871.158	-2004.255							
55	15	20	14	391.171	-287.425	891.737				13.000		3	
				-2416.722	82.559	356.534							
				160.310	-28.722	231.103							
				-619.747	-57.554	731.332							
56	14	18	20	3054.051	57.104	-552.223				0.600		3	
				5110.092	-80.320	-443.724							
				-4531.036	-48.579	-733.714							
				1211.049	-25.932	-574.337							
58	13	17	19	1745.952	-2750.403	-203.177				1.541		3	
				18266.059	3043.321	-491.048							
				-17234.910	-75.035	237.775							
				6022.034	46.008	-172.532							
5	19	27	21	27082.771	2119.202	-422.522				0.895		1	
				37703.424	-3153.336	-3303.336							
				26032.073	-506.960	368.445							
				30494.759	-515.498	-2519.148							
6	21	29	27	3676.151	1800.157	-524.719				0.923		1	
				18512.955	-1742.330	-432.706							
				34207.239	-1270.459	-1356.431							
				16756.763	-463.704	-3671.519							

Reproduced from
best available copy.

LEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	PARCIN	LEG
11	17	25	23	59406.773 4749.709 4400.753 31545.737	2170.000 -002.777 -274.931 435.041	318.236 -107.652 -1229.231 -430.208	1.502	1
12	23	31	25	29300.370 43000.361 5090.013 27432.073	2320.337 -2259.790 555.349 231.017	-4193.335 -4003.724 -4395.32 -3007.117	.035	1
17	20	20	22	1179.001 4242.349 40752.323 34224.772	731.102 -372.220 55.400 -57.530	-1449.101 -1174.021 303.733 -772.106	1.178	1
18	22	30	20	29470.270 42544.219 13740.703 29005.340	2300.470 -2200.552 323.120 113.079	-4307.123 -2330.340 -1395.002 -4371.138	.031	1
22	16	24	18	42000.452 39204.242 10027.021 33752.102	-2720.322 57.337 401.007 20.003	2321.117 2322.030 -403.331 1333.014	1.000	1
23	18	20	24	34023.050 30337.233 20011.463 30503.331	002.012 -1420.373 -400.002 -337.341	-274.936 643.737 -1412.34 -447.704	1.517	1
24	24	32	20	13077.004 21394.000 25000.010 21453.224	1310.731 -240.022 -002.010 35.313	-1032.022 -2147.733 -2342.512 -2001.332	1.035	1
30	17	21	15	2533.174 -12320.425 3091.730 -2153.350	347.109 004.175 -131.103 404.327	4503.006 2577.730 201.293 2007.041	1.307	2
37	10	20	10	3073.114 -1044.700 1191.009 -1308.737	330.407 -240.397 -1.000 143.153	-303.020 -1293.720 -135.035 -044.040	7.352	2
38	18	22	20	2445.470 -3433.004 0202.024 -2702.200	103.222 -21.219 100.402 00.373	-544.042 -200.737 471.040 45.335	7.220	2

ELEMENT NO	J	K	SIGMA 1	SIGMA 2	1MO 12	MARGIN	LEG
37	20	23	22	-1209.102 -1032.034 -572.052 -1377.050	-100.015 250.077 -20.233 2.420	10.000	2
40	24	25	23	-2010.253 -3073.034 1723.015 -1029.442	-307.751 -00.249 240.003 -00.010	10.000	2
41	25	27	22	-2030.000 -4450.010 0010.040 -1047.027	-000.273 040.103 -700.001 -170.070	2.000	2
42	25	24	27	4100.002 -077.013 -0400.710 -1000.010	-1000.000 1100.000 -070.000 -300.771	0.000	2
43	24	24	26	0074.002 -3400.000 0000.001 2412.011	-1400.000 1100.000 -000.000 -300.000	1.000	2
44	20	30	20	1000.000 -000.000 0000.000 -000.000	-070.000 1000.000 0000.000 -000.000	10.000	2
45	20	31	30	-1000.000 -000.000 0000.000 1000.000	-1700.000 0000.000 0000.000 1000.000	0.700	2
46	20	24	10	-0000.000 -0000.000 0000.000 -0000.000	-000.000 -000.000 0000.000 -000.000	0.000	2
47	19	23	17	0000.000 -0000.000 0000.000 -0000.000	-000.000 -000.000 0000.000 -000.000	0.000	2
48	17	26	00	1000.000 0000.000 0000.000 1000.000	-000.000 -000.000 0000.000 -000.000	0.000	2

ELEMENT		J K		SIGMA 1		SIGMA 2		TAD 12		P1401A		L20	
NO	I	J	K										
61	23	28	22	280.010		-191.073		313.142		7.024		3	
				-2770.337		-7.170		1337.016					
				8.07.047		-153.952		-0.0014					
				1040.900		-119.004		720.112					
62	22	26	25	2254.771		-187.350		1733.173		3.473		3	
				-3764.754		1128.054		2250.006					
				12250.331		-773.324		420.172					
				3265.343		-26.0.150		1492.142					
63	28	32	26	4763.800		-1003.074		1247.044		4.254		3	
				1226.373		1340.224		1444.070					
				-3400.274		101.160		-225.777					
				949.092		-20.741		922.002					
64	21	25	27	14053.250		-003.154		4217.073		4.570		3	
				1027.160		1416.140		0422.031					
				-1005.008		-1000.302		-1000.017					
				5.10.343		-003.151		2335.036					
65	27	31	25	5168.723		-211.000		4211.174		1.000		3	
				-4123.714		1050.350		4400.031					
				3423.413		177.003		-2402.711					
				1400.011		-07.302		1403.001					
66	25	30	31	-000.393		-1074.014		0324.027		0.973		3	
				-5030.273		3903.544		2042.010					
				17422.064		1000.001		-2043.070					
				2065.144		1332.003		1070.000					

MARGIN OF SAFETY IN EACH LEG = $\frac{10000}{1541} = 6.49$

ULTIMATE LOADS		MAX		MY		MAX		TORQUE		LX		LY	
NA	NY	MAX	NY	MAX	NY	MAX	NY	MAX	NY	MAX	NY	MAX	NY
1541	0	124	0	0	0	0	0	0	0	0	0	0	0

LEG BUCKLING STRESS
6962 6962 6962