ASACS - ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM
A PROBLEM ORIENTED LANGUAGE
FOR A LIMITED COMPUTER ENVIRONMENT

by

James K. Nelson, Jr.

A Thesis
Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Civil Engineering
The University of Houston

14 August 1976

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

Academic Structural Analysis Computer System, ASACS, is a general purpose computer software system for the analysis of plane trusses, plane frames, and plane grids. The geometry of the structure, the member properties, and the loads are provided by the user in a problem oriented language. Member forces, joint displacements, and reactions are produced in an easily understood tabular format. A linear, elastic, static, small-deflection stiffness analysis is used to calculate these values.

ASACS is ideally suited to the analysis of small to medium-sized structures in a time-sharing or limited computer environment. Minimal storage requirement and computational time are features of ASACS that enable both student users and practicing engineers to obtain rapid turnaround and reliability in structural problem solution.

14 August 1976

CHAPTER 1

INTRODUCTION

Structural engineering, especially structural engineering education, has advanced to the point that it is no longer practical or feasible to analyze all structures by hand in the process of design. Several large scale computer software systems have been developed - among them: STRUDL, STRESS, and NASTRAN - to perform analysis. In order to operate, these programs require sophisticated computers with large storage capacity and long execution times. As such they are not well-suited to a time-sharing system, for use as a preliminary design tool, or for the analysis of small structures using limited computing facilities.

A computer software system has been developed to fill this gap, not filled by these other systems. The program (ASACS - Academic Structural Analysis Computer System) will analyze plane trusses, plane frames, and plane grids using the stiffness method of analysis. Input is in a problem oriented language easily understood by structural engineers. Output provides the user with information needed for design: member forces, joint reactions, and displacements presented in a tabular form which is easily interpreted. All external similarities between STRUDL and ASACS are intentional. These similarities make the transition for the user from one system to the other as simple as possible.

Computational procedures for ASACS utilize a matrix stiffness analysis. Extensive use was made of matrix overlays in order to hold memory requirements at an absolute minimum for operation in a time-sharing system or a small stand-alone computer. Coding was done in FORTRAN IV so that except for minor modifications, the programs are machine independent. This should enhance user acceptance. Prior to this, 1620 STRESS was the only system really comparable to ASACS, but the approach and basic philosophy were much different.

CHAPTER 2

THEORETICAL CONSIDERATIONS

Academic Structural Analysis Computer System employs a linear, elastic, static, small-deflection stiffness analysis. The classes of structures that can be handled are pin-connected plane trusses, plane frames, and plane grids. Formulation of the solution for all three classifications is essentially the same, however the complexity of solution increases in the order named. All three programs could have been incorporated into a single, large program system. In order to keep the system small enough to provide rapid turnaround in a batch environment, reasonable partition size in time-sharing, and compatibility with small computers, each class was programmed as a separate entity.

2.1 General Matrix Formulation

The analysis of a structure by the stiffness method requires two general equations.

$$P = K * X \qquad (1)$$

$$F = k * B * X \qquad (2)$$

Where: P = joint load vector
K = global stiffness matrix
X = global displacement vector
k = local stiffness matrix
B = transformation-rotation matrix
F = member force vector
* denotes matrix multiplication

Equations (1) and (2) are the force - deflection relations in global and local coordinates respectively.

Each degree of freedom at each joint in the structure is assigned a coordinate number regardless of the type of structure. The constrained coordinates are numbered last. This places the free-joint information in leading partitions of each matrix. The restriction that no constrained coordinate is allowed to displace permits the matrices of Equation (1) to be partitioned as shown in Equation (1a).

$$\left[\begin{array}{c|c} K_{11} & K_{12} \\ \hline K_{21} & K_{22} \end{array}\right] \left\{\begin{array}{c} X_1 \\ \hline 0 \end{array}\right\} = \left\{\begin{array}{c} P_1 \\ \hline P_2 \end{array}\right\} \quad (1a)$$

The partitioning separates the free and constrained coordinates. P_1 contains the joint loads and P_2 contains the resultant support reactions. The elements of X_1 are the resultant global joint displacements. Solution requires inversion of K_{11} only. This is performed using Gaussian elimination.

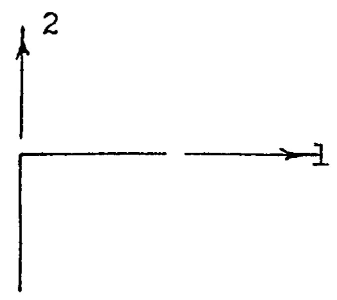
The development of the local stiffness matrix, k, and the transformation-rotation matrix, B, is presented for each type of structure in the sections that follow. The global stiffness matrix is calculated from these elements as

$$K = \sum_{m=1}^{n} B_m^T * k_m * B_m \quad (3)$$

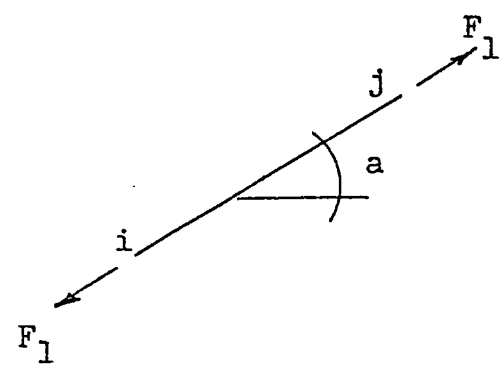
where: m = member number

2.2 Plane Truss Formulation

In a plane truss, each joint may translate vertically and horizontally and the individual members deform axially. A typical joint with coordinates labelled is shown in Figure 2.1. A member is described in Figure 2.2 showing the local force coordinate.



Typical Truss Joint
Figure 2.1



Typical Truss Member
Figure 2.2

The transpose of the transformation-rotation vector, B, is obtained by satisfying joint equilibrium for each end of the member.

$$i_1 = -\cos(a) \cdot F_1$$

$$i_2 = -\sin(a) \cdot F_1 \qquad (4)$$

$$j_1 = \cos(a) \cdot F_1$$

$$j_2 = \sin(a) \cdot F_1$$

Representing this in matrix form, one obtains

$$B^T = \begin{Bmatrix} -\cos(a) \\ -\sin(a) \\ \cos(a) \\ \sin(a) \end{Bmatrix} \qquad (5)$$

For the coordinate considered the local stiffness matrix is (the single element)

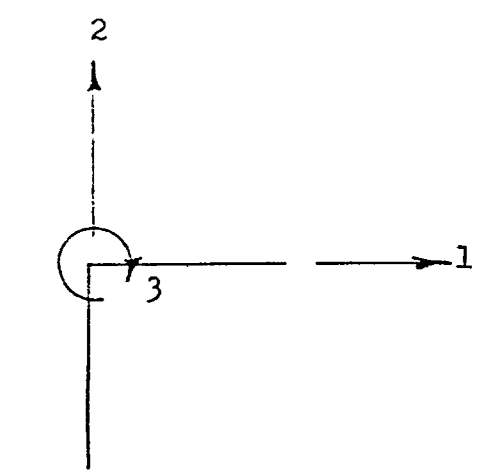
$$k = \left[\frac{A \cdot E}{L}\right] \qquad (6)$$

Where: A = cross-sectional area
E = Young's modulus
L = member length

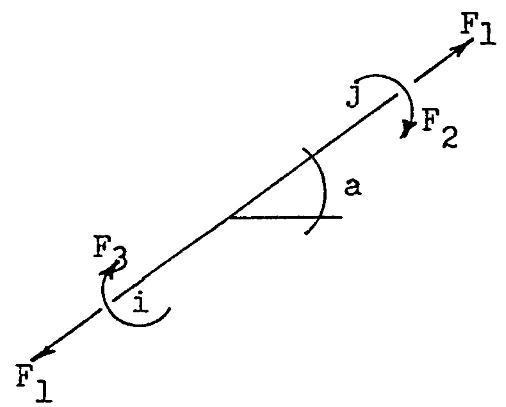
Equations (5) and (6) are substituted into Equations (1) thru (3) for the solution of member forces, joint reactions, and joint displacements.

2.3 Plane Frame Formulation

Each joint in the plane frame has three degrees of freedom: horizontal and vertical translation, and in-plane rotation. The members of the frame also have three force components: in-plane rotation at each end of the member and an axial force. Typical joints and members are illustrated in Figures 2.3 and 2.4 respectively, showing the force and displacement components. The shears on the member ends are calculated from static equilibrium.



Typical Frame Joint
Figure 2.3



Typical Frame Member
Figure 2.4

As with the plane truss, the transformation-rotation matrix is obtained from joint equilibrium at each end of the member.

$$
\begin{aligned}
i_1 &= -F_1\cos(a) + (F_2+F_3)/L \cdot \sin(a) \\
i_2 &= -F_1\sin(a) - (F_2+F_3)/L \cdot \cos(a) \\
i_3 &= F_3 \\
j_1 &= F_1\cos(a) - (F_2+F_3)/L \cdot \sin(a) \\
j_2 &= F_1\sin(a) + (F_2+F_3)/L \cdot \cos(a) \\
j_3 &= F_2
\end{aligned}
\qquad (7)
$$

Representing this in matrix form

$$B^T = \begin{bmatrix} -\cos(a) & \frac{\sin(a)}{L} & \frac{\sin(a)}{L} \\ -\sin(a) & \frac{-\cos(a)}{L} & \frac{-\cos(a)}{L} \\ 0 & 0 & 1 \\ \cos(a) & \frac{-\sin(a)}{L} & \frac{-\sin(a)}{L} \\ \sin(a) & \frac{\cos(a)}{L} & \frac{\cos(a)}{L} \\ 0 & 1 & 0 \end{bmatrix} \qquad (8)$$

The member stiffness matrix for the plane frame is

$$k = \begin{bmatrix} \frac{A \cdot E}{L} & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \qquad (9)$$

Where: A = cross-sectional area
E = Young's modulus
I = moment of inertia about z-axis
L = member length

With the plane frame it is possible to release one or both ends of the individual members with respect to moment. When this occurs the member stiffness matrix becomes one of the following three types.

If the initial end is released member stiffness is given by

$$k = \begin{bmatrix} \frac{A \cdot E}{L} & 0 & 0 \\ 0 & \frac{3EI}{L} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (10)$$

If the terminal end is released member stiffness is given by

$$k = \begin{bmatrix} \frac{A \cdot E}{L} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{3EI}{L} \end{bmatrix} \qquad (11)$$

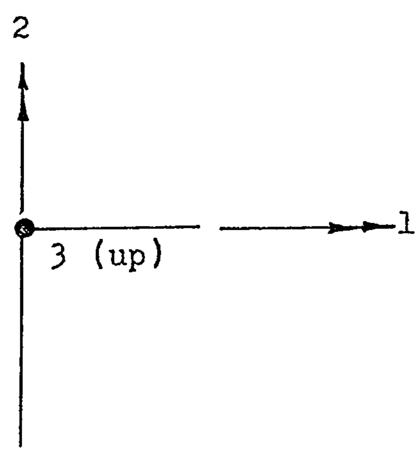
If both ends are released member stiffness is given by

$$k = \begin{bmatrix} \frac{A \cdot E}{L} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (12)$$

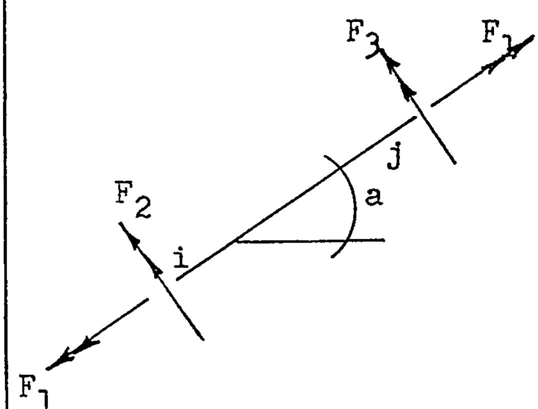
Equations (8) thru (12) are substituted into Equations (1) thru (3). The resulting matrix equations are solved for member forces, joint reactions, and joint displacements.

2.4 Plane Grid Formulation

Each joint in the plane grid has three degrees of freedom: out-of-plane rotation about x and y axes in the plane of the grid and translation normal to the plane. The members of the grid have three force components: torsion and out-of-plane bending at each end of the member. End shears are calculated from static equilibrium. Figures 2.5 and 2.6 show the force and displacement components on typical joints and members.



Typical Grid Joint
Figure 2.5



Typical Grid Member
Figure 2.6

As with both of the preceding structures, the transpose of the transformation-rotation matrix is obtained from joint equilibrium at each end of the member.

$$\begin{aligned}
i_1 &= -F_1\cos(a) - F_2\sin(a) \\
i_2 &= -F_1\sin(a) + F_2\cos(a) \\
i_3 &= -(F_2 + F_3)/L \\
j_1 &= F_1\cos(a) - F_3\sin(a) \\
j_2 &= F_1\sin(a) + F_3\cos(a) \\
j_3 &= (F_2 + F_3)/L
\end{aligned} \tag{13}$$

Representing this in matrix form, one obtains

$$B^T = \begin{bmatrix} -\cos(a) & -\sin(a) & 0 \\ -\sin(a) & \cos(a) & 0 \\ 0 & -1/L & -1/L \\ \cos(a) & 0 & -\sin(a) \\ \sin(a) & 0 & \cos(a) \\ 0 & 1/L & 1/L \end{bmatrix} \tag{14}$$

The stiffness matrix for a plane grid member is

$$k = \begin{bmatrix} \frac{J \cdot G}{L} & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \qquad (15)$$

Where: J = torsional constant
I = moment of inertia about z-axis
E = Young's modulus
G = shear modulus
L = length of the member

If the torsional component is released, the member stiffness matrix is

$$k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \qquad (16)$$

If both end moments are released, the stiffness matrix is

$$k = \begin{bmatrix} \frac{J \cdot G}{L} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (17)$$

Equations (14) thru (17) can now be entered into Equations (1) thru (3) to obtain member forces, joint reactions, and joint displacements.

CHAPTER 3

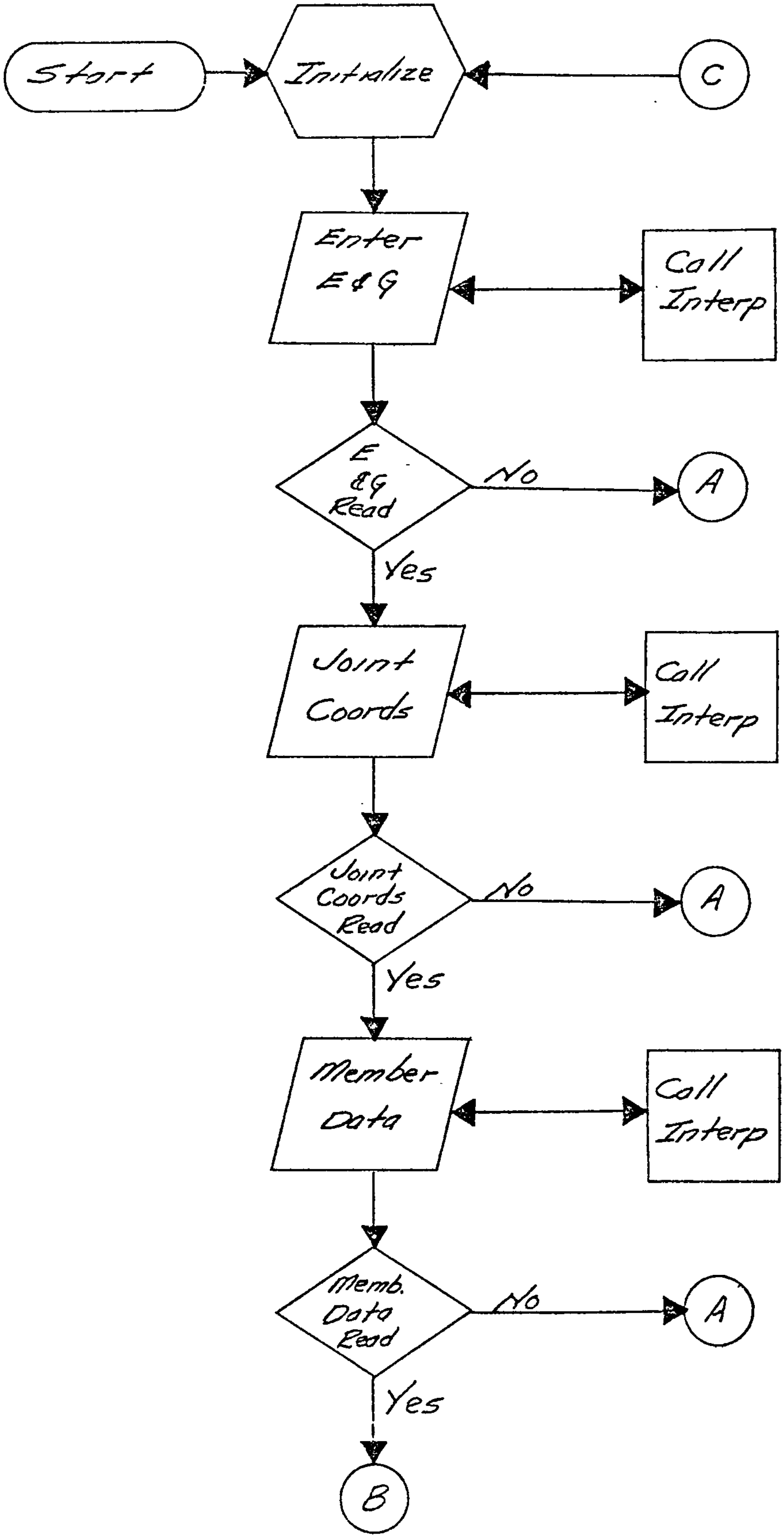
ASACS PROGRAM COMPONENTS

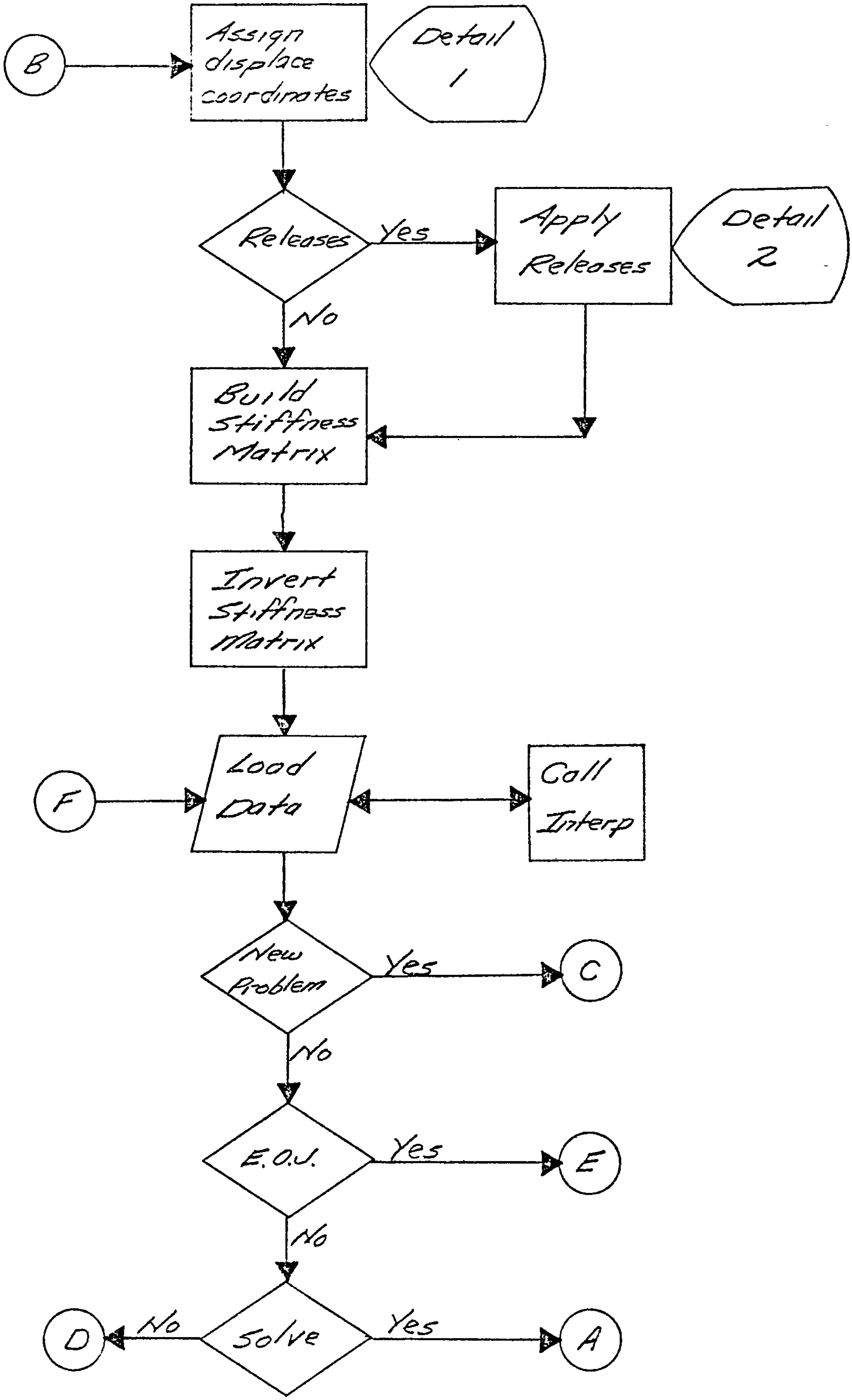
This section contains the flowcharts and a brief description of the main program and all of the functions and subprograms that are used in ASACS. The entire system was written in the FORTRAN programming language. With the exception of the unit numbers for the card reader and line printer, the program is machine independent. In addition, the program is set up in such a way that it is not necessary to make any changes between batch processing and time-sharing. A complete source code listing from the UNIVAC 1108 computer for the plane truss, plane frame, and plane grid appears in appendices B, C, and D respectively.

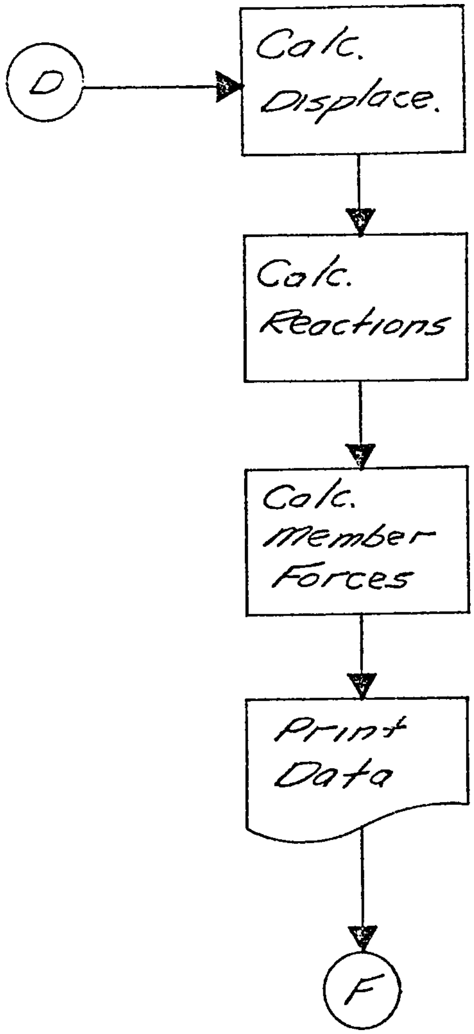
The matrix inverter used for this program encompasses the subroutines INVERT, PIVOT, PIVOTR, and BACSUB. These constitute a general-purpose matrix inversion routine using the Gaussian elimination algorithm with pivoting for size.

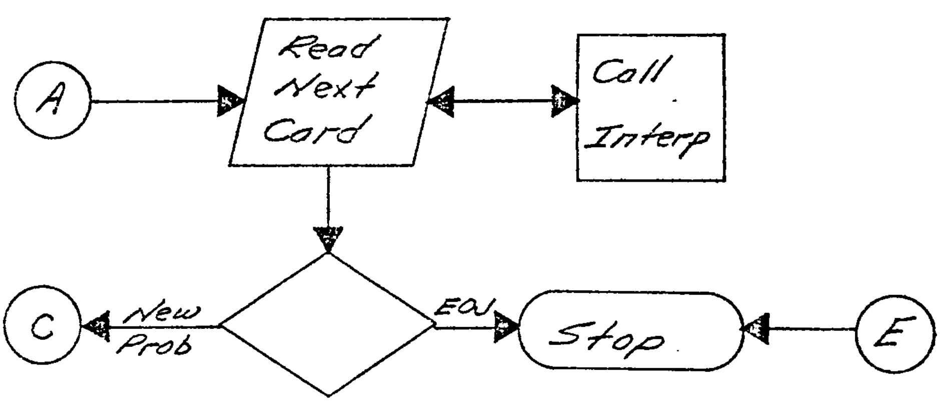
3.1 Main Program

The primary function of the main program is to determine that there is sufficient data for execution, build and invert the global stiffness matrix, calculate the joint reactions, joint displacements, and member forces, and print these data out in an orderly fashion. The subroutines are called as needed by the main program to obtain the data necessary to perform the calculations.

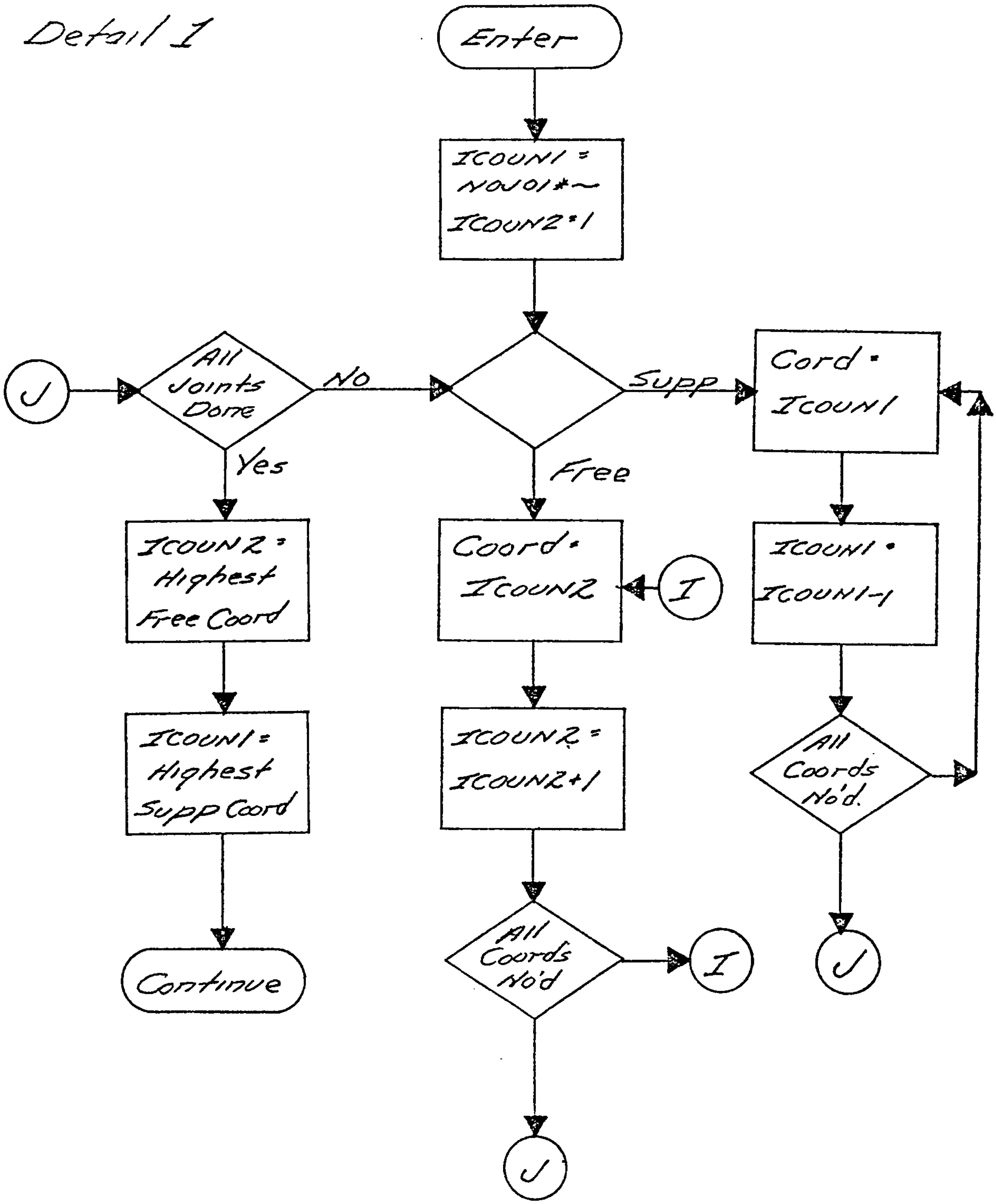

Start
Initialize
C
Enter
E & G
Call
Interp
E
& G
Read
No
A
Yes
Joint
Coords
Call
Interp
Joint
Coords
Read
No
A
Yes
Member
Data
Call
Interp
Memb.
Data
Read
No
A
Yes
B

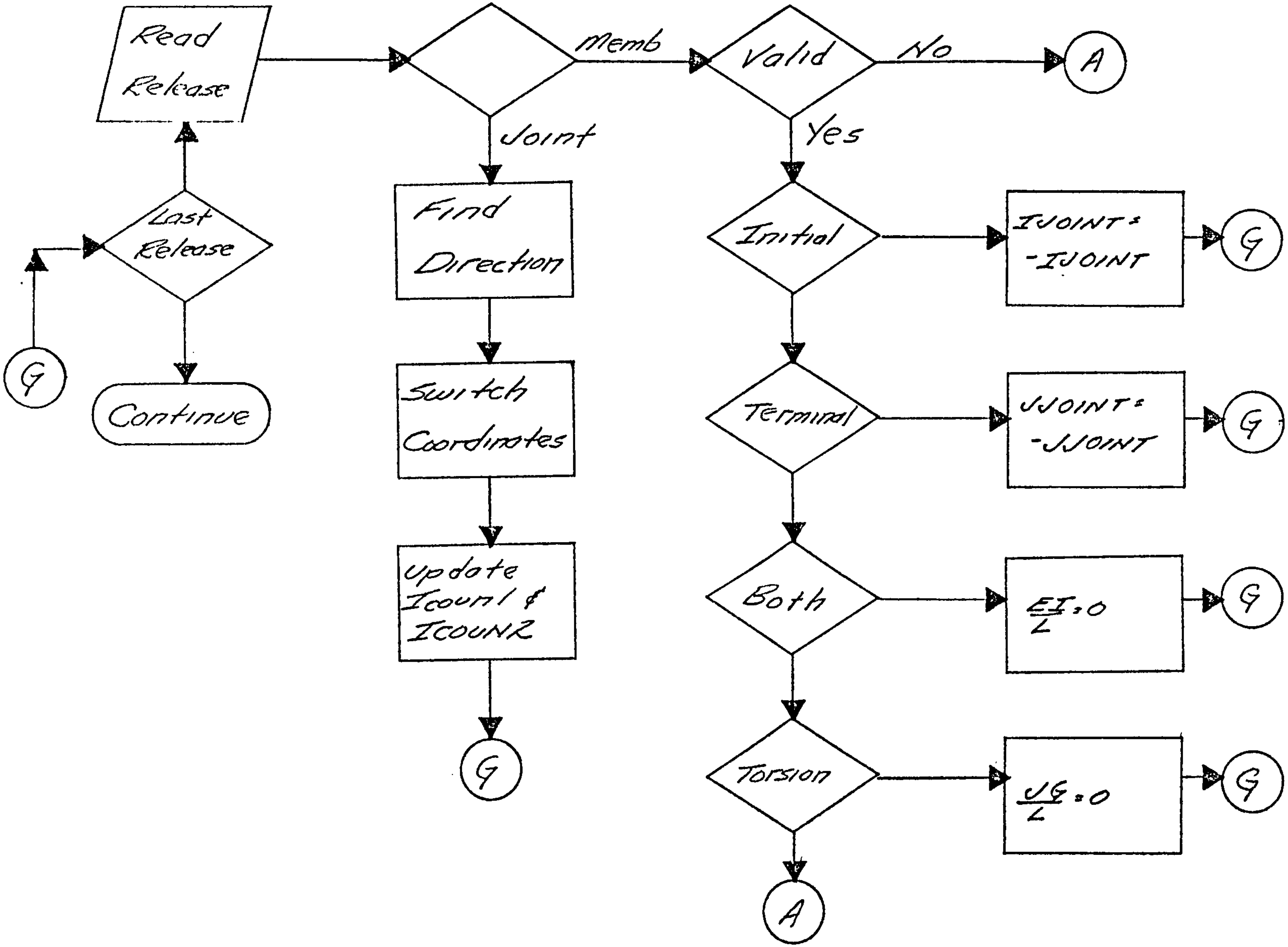

B
Assign displace coordinates
Detail 1
Releases
Yes
Apply Releases
Detail 2
No
Build Stiffness Matrix
Invert Stiffness Matrix
F
Load Data
Call Interp
New Problem
Yes
C
No
E.O.J.
Yes
E
No
D
No
Solve
Yes
A


D
Calc. Displace.
Calc. Reactions
Calc. Member Forces
Print Data
F


A
Read Next Card
Call Interp
C
New Prob
EOJ
Stop
E

Detail 1

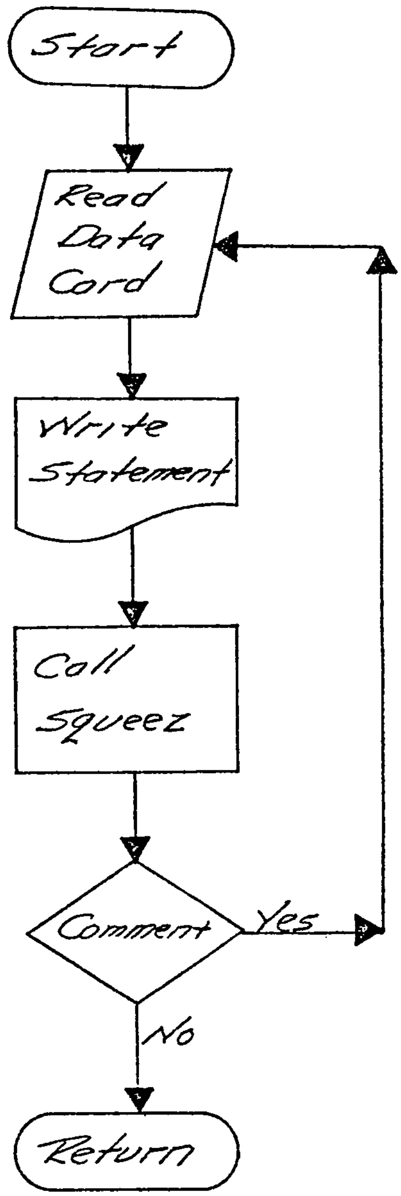

Enter
ICOUN1 = NOJ01*~
ICOUN2=1
All Joints Done
No
Supp
Free
Yes
Cord = ICOUN1
ICOUN1 = ICOUN1-1
All Coords No'd.
ICOUN2 = Highest Free Coord
ICOUN1 = Highest Supp Coord
Continue
Coord = ICOUN2
ICOUN2 = ICOUN2+1
All Coords No'd
I
J


Read Release
Last Release
G
Continue
Memb
Joint
Find Direction
Switch Coordinates
Update Icoun1 & ICOUN2
G
Valid
No
A
Yes
Initial
IJOINT= -IJOINT
G
Terminal
JJOINT= -JJOINT
G
Both
EI/L=0
G
Torsion
JG/L=0
G
A

Detail 2

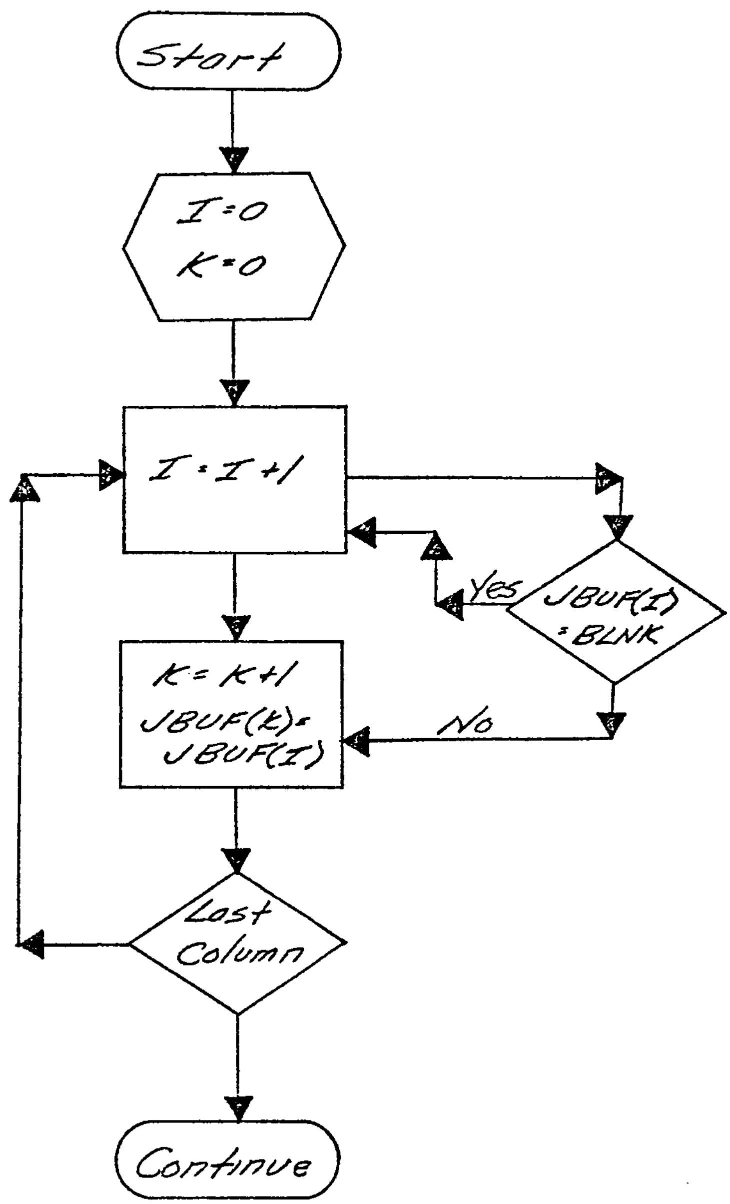
3.2 Subroutine Reader

This subroutine reads each of the input cards and echo prints it. Comment cards are also searched for and printed. The data on the card is read into a buffer array.



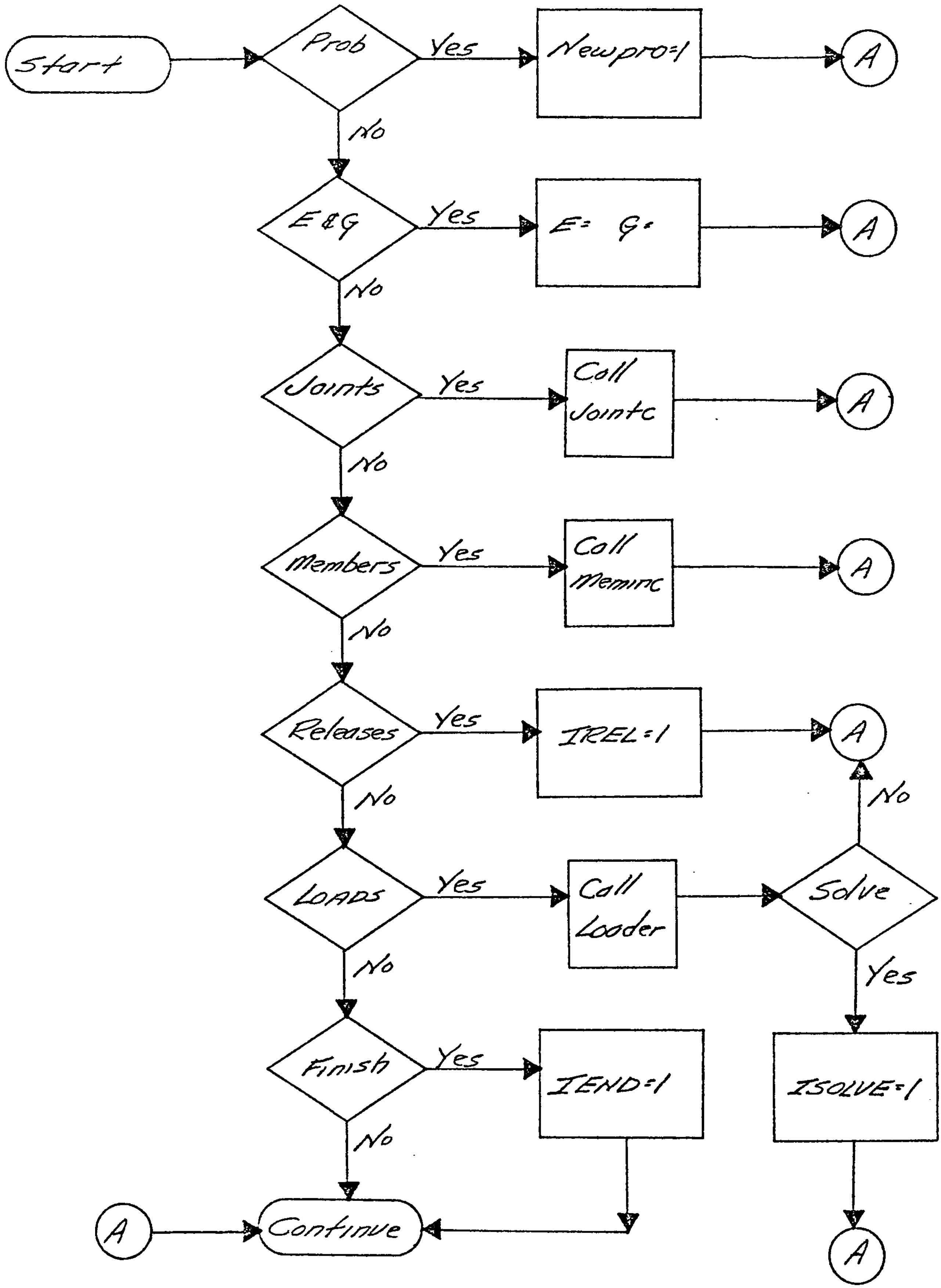
3.3 Subroutine Squeez

This subroutine removes blanks from the input card.



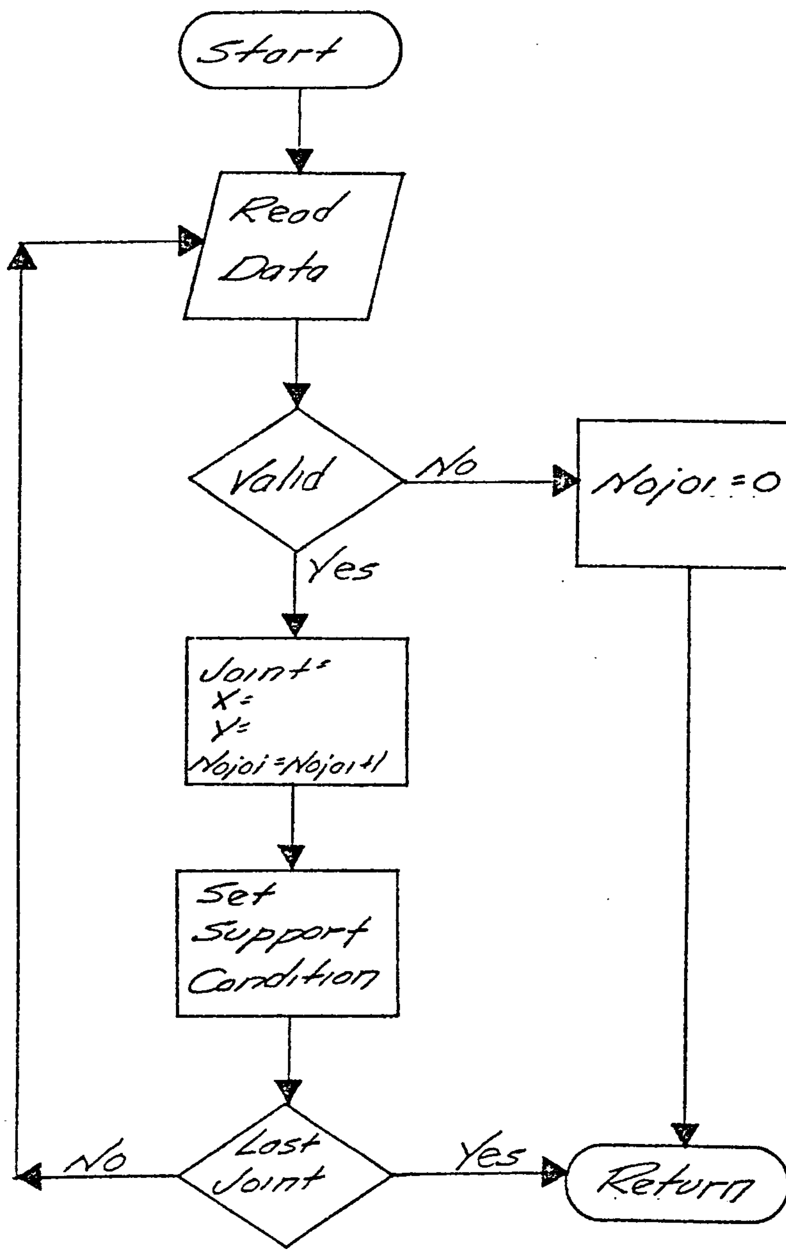
3.4 Subroutine Interp

This subroutine interprets the input cards to determine what type of statment the card is. After this is determined the data is either handled in this routine or an appropiate subroutine is called to handle the data.


Start
Prob
Yes
Newpro=1
A
No
E&G
Yes
E= G=
A
No
Joints
Yes
Call Jointc
A
No
Members
Yes
Call Meminc
A
No
Releases
Yes
IREL=1
A
No
LOADS
Yes
Call Loader
Solve
No
Yes
ISOLVE=1
A
No
Finish
Yes
IEND=1
No
A
Continue

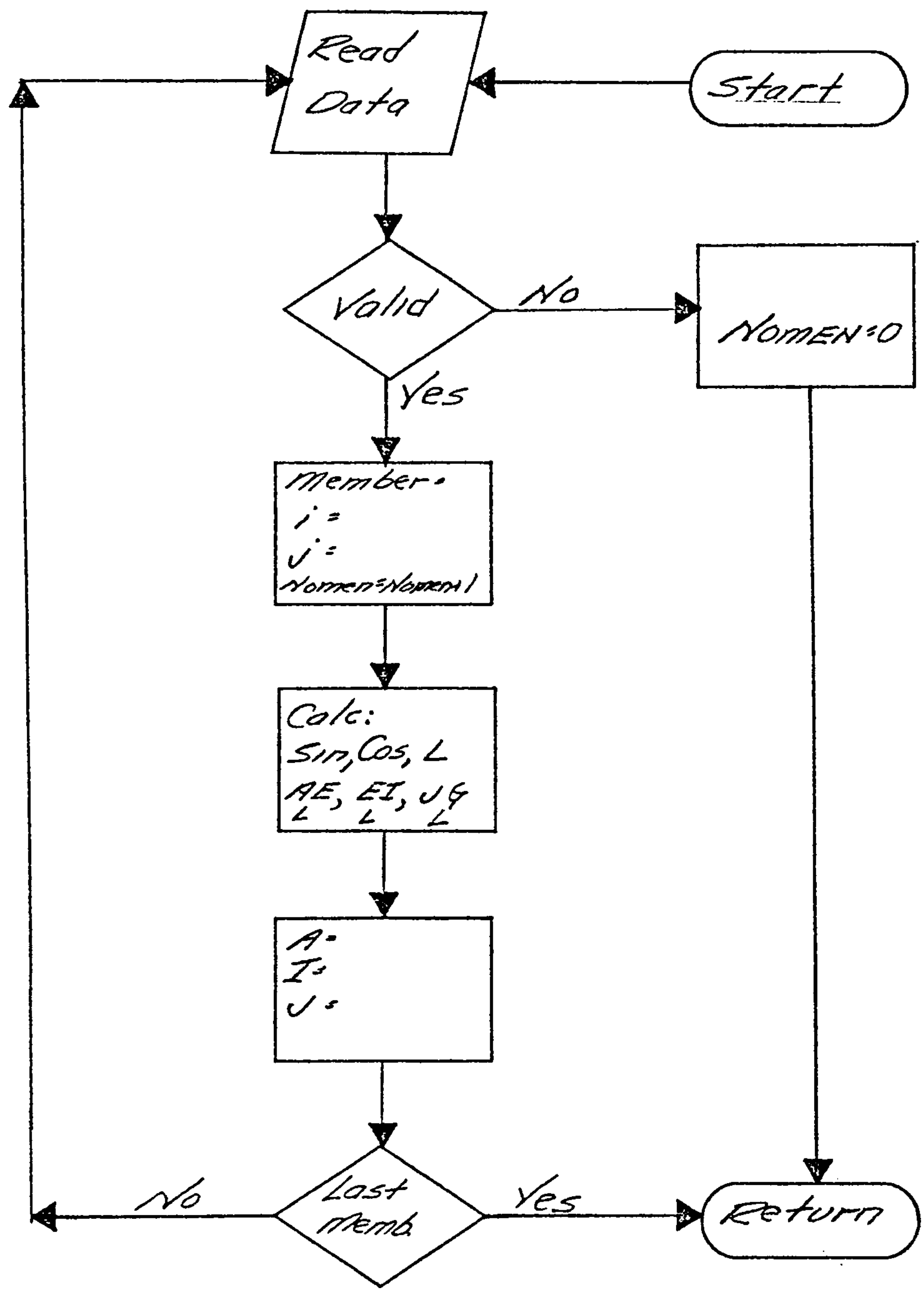
3.4 Subroutine Jointc

This subroutine handles the data involving description of joints. It is called from subroutine INTERP. The x and y coordinates are decoded and placed into a real array. The condition of the joint is placed into an integer array.



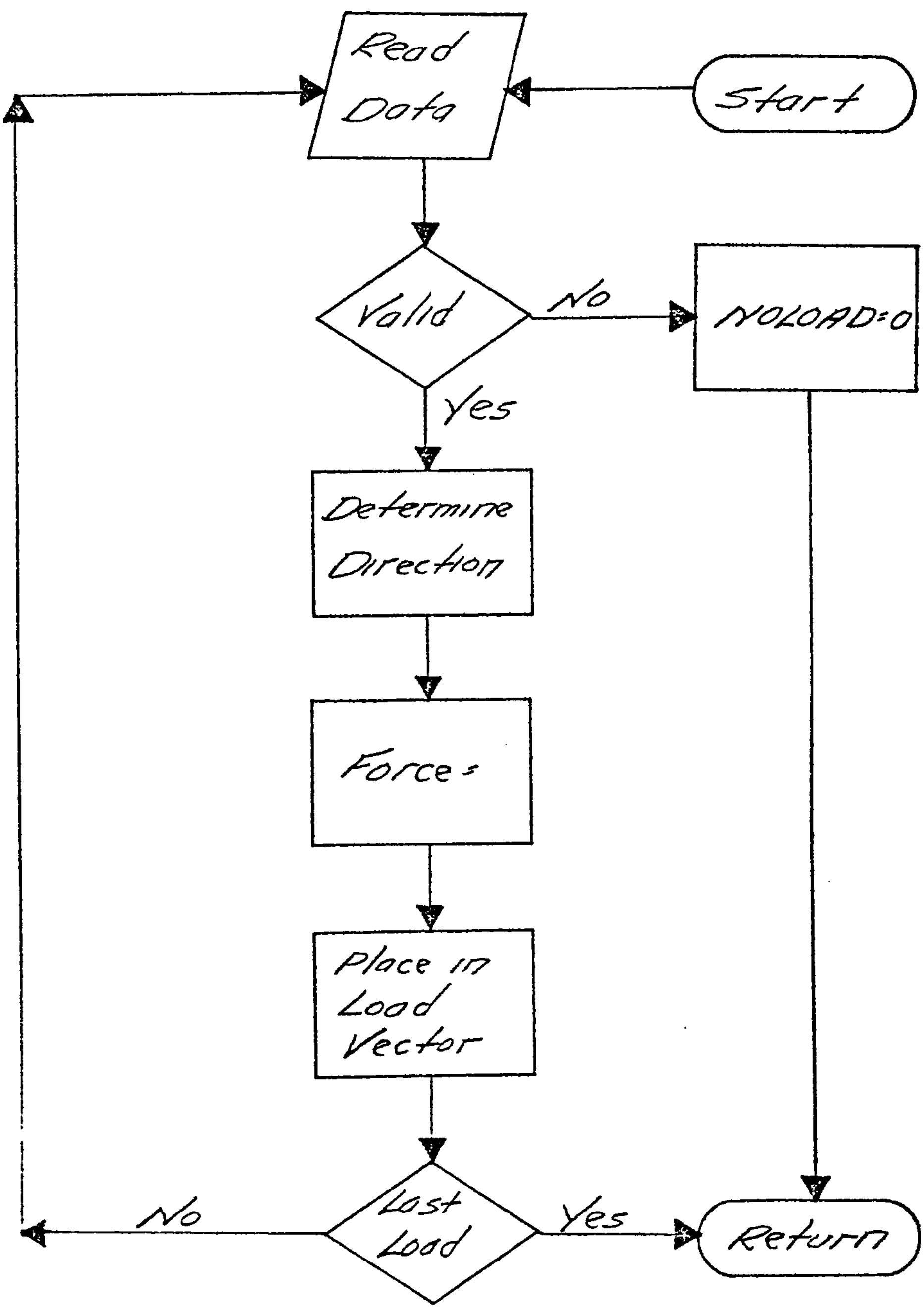
3.6 Subroutine Meminc

This subroutine handles the data involving description of members. The member incidences are entered into an integer array, the same array as the joint condition. The various stiffnesses of the mamber are calculated and entered into the real array.



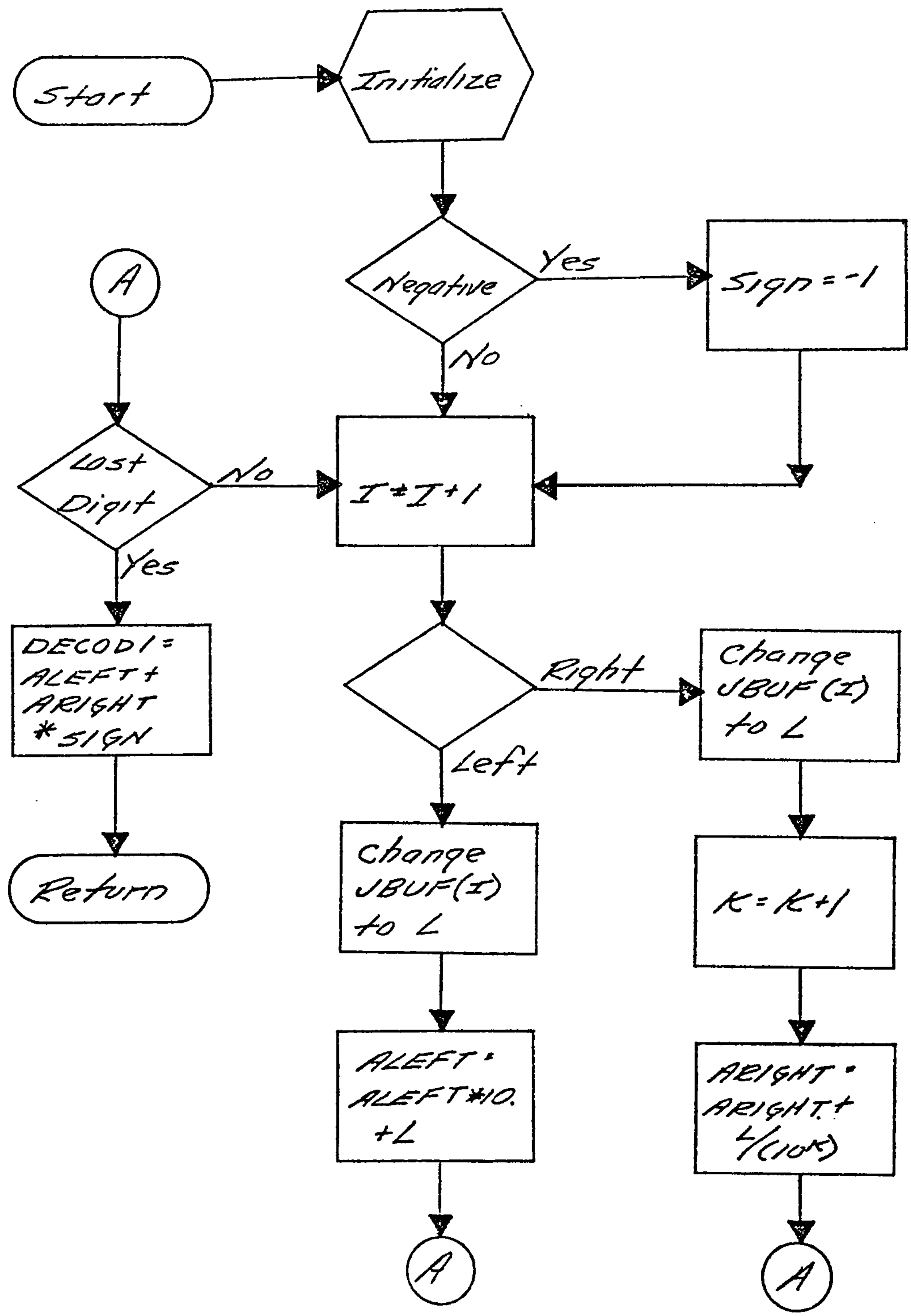
3.7 Subroutine Loader

This subroutine reads the joint loads, determines which component is loaded, and enters the load into the proper location in the load vector.



3.8 Function Decodl

This function is used to convert the numerics found on the input cards into real or integer numbers.



CHAPTER 4

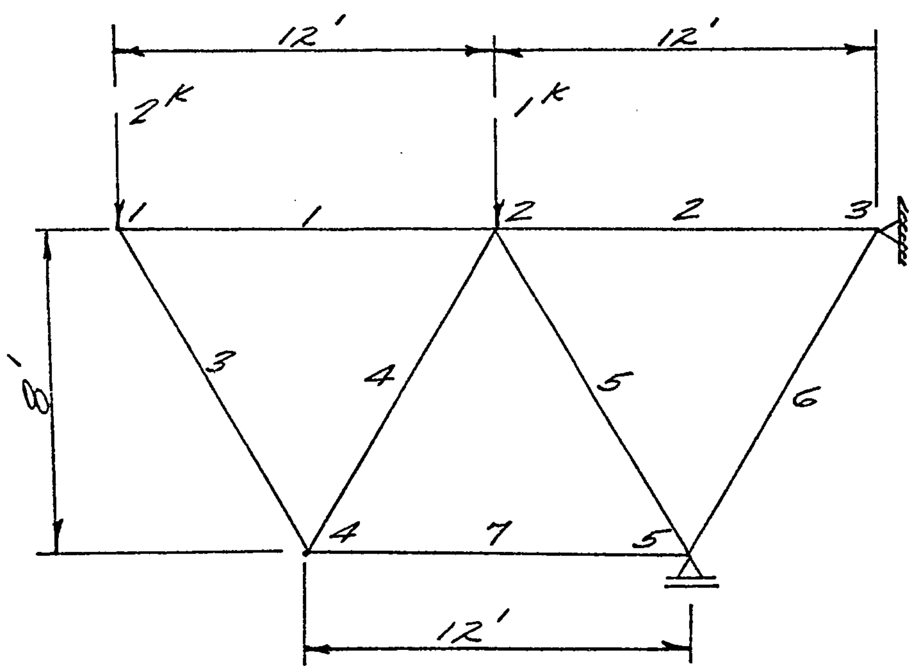
EXAMPLES - TEST PROBLEMS

This chapter contains the test problems that were used to test the accuracy of ASACS. Test problems used were selected because the solution is well documented in textbooks. For each problem, a description of the structure, the ASACS solution, and a comparison of the results with the other solutions is presented.

4.1 Plane Truss Test Problem

This problem tests the accuracy of the plane truss program. The truss, shown in Figure 4.1, is composed of seven members and five joints. Two of the joints are supports and one of these requires a release for horizontal translation. The structure is subjected to two joint loads. The theoretical solution can be found in

<u>Mechanics for Engineers - Statics</u>
Beer and Johnson
McGraw-Hill Book Company
Example 6.1



Planr Truss Test Problem
Figure 4.1

```
PROBLEM   ***** PLANE TRUSS TEST PROBLEM  *****
$ REFERENCE: MECHANICS FOR ENGINEERS
$            BEER AND JOHNSON
$            EXAMPLE 6.1
E=30000.
JOINT COORDINATES
1,0.,96.
2,144.,96.
3,288.,96.,S
4,72.,0.
5,216.,0.,S
MEMBER INCIDENCES
1,1,2,10.
2,2,3,10.
3,1,4,10.
5,2,5,10.
4,4,2,10.
6,5,3,10.
7,4,5,10.
RELEASE JOINT 5, FORCE X
LOADING LIVE
JOINT 2, FORCE Y -1.0
JOINT 1, FORCE Y -2.0
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION X	Y	Z
1	-.00324	-.02587	
2	-.00252	-.00814	
3	.00000	.00000	
4	.00727	-.01673	
5	.00583	.00000	

SUPPORT REACTIONS

JOINT	FORCE X	Y	Z
3	.00000	-7.00000	
5	.00000	10.00000	

MEMBER FORCES

MEMBER	AXIAL	FORCE SHEAR Y	SHEAR Z
1	1.50000		
2	5.25000		
3	-2.50000		
4	2.50000		
5	-3.75000		
6	-8.75000		
7	-3.00000		

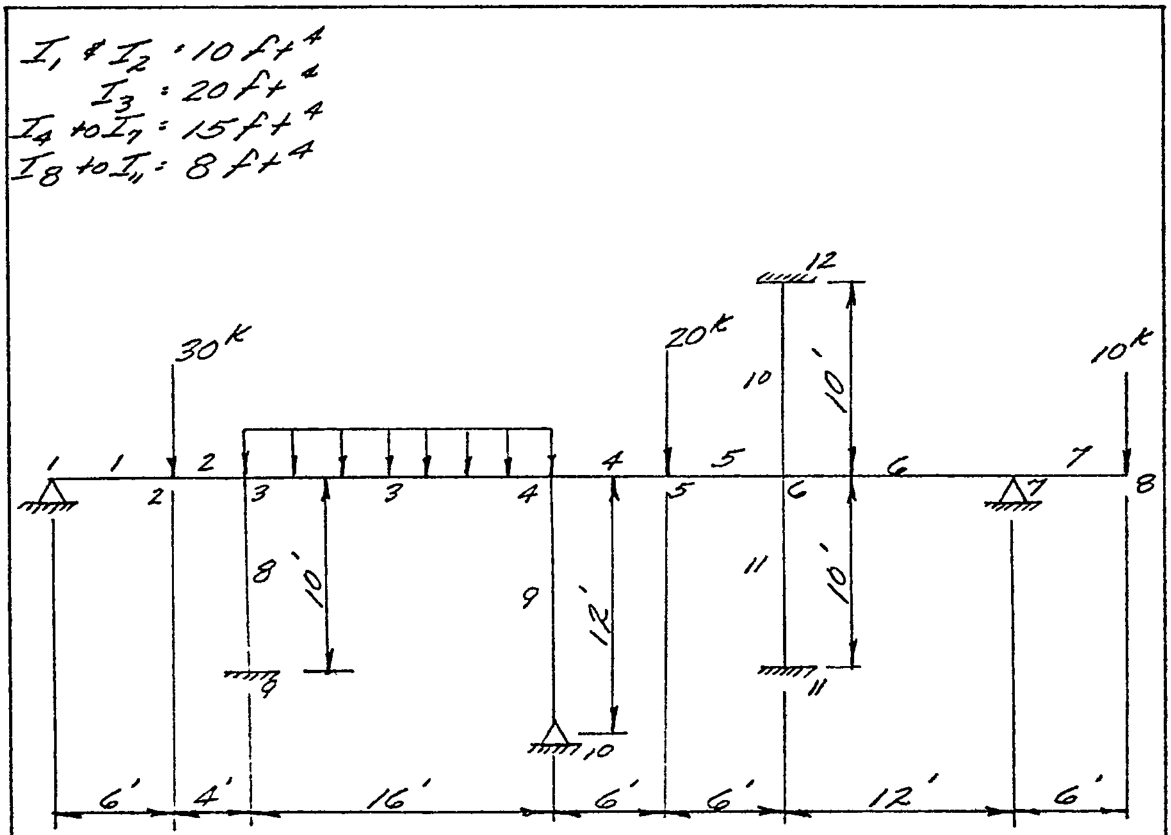
Member	Text	ASACS
1	1.5	1.5
2	5.25	5.25
3	-2.5	-2.5
4	2.5	2.5
5	-3.75	-3.75
6	-8.75	-8.75
7	-3.0	-3.0

Comparison of Results
Plane Truss
Table 4.1

4.2 Plane Frame Test Problem

This problem tests the accuracy of the plane frame program. The frame, shown in Figure 4.2, is composed of eleven members and twelve joints. Six of the joints are support joints and three of these have moment releases. The theoretical solution can be found in

<u>Elementary Structural Analysis</u>
Norris and Wilbur
Page 459



Plane Frame Test Problem
Figure 4.2

```
PROBLEM  PLANE FRAME TEST PROBLEM
$   REFERENCE  ELEMENTARY STRUCTURAL ANALYSIS
$              NORRIS AND WILBUR
$              PAGE 459
E=1.0
JOINT COORDINATES
1,0.,0.,S
2,5.,0.
3,10.,0.
4,26.,0.
5,32.,0.
6,38.,0.
7,50.,0.,S
8,56.,0.
9,10.,-10.,S
10,26.,-12.,S
11,38.,-10.,S
12,38.,10.,S
MEMBER INCIDENCES
1,1,2,99999.,10.
2,2,3,99999.,10.
3,3,4,99999.,20.
4,4,5,99999.,15.
5,5,6,99999.,15.
6,6,7,99999.,15.
7,7,8,99999.,15.
8,9,3,99999.,6.
9,10,4,99999.,8.
10,6,12,99999.,3.
11,11,6,99999.,8.
```

```
RELEASE JOINT 1, MOMENT Z
RELEASE MEMBER 9, INITIAL END
RELEASE JOINT 7, MOMENT Z
LOADING TEST
JOINT 2, FORCE Y -30.
JOINT 5, FORCE Y -20.
JOINT 8, FORCE Y -10.
JOINT 3, FORCE Y -24.
JOINT 3, MOMENT Z 64.
JOINT 4, FORCE Y -24.
JOINT 4, MOMENT Z -64.
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION X	TRANSLATION Y	ROTATION Z
1	.00000	.00000	6.65662
2	.00001	-18.65045	-3.98800
3	.00002	-.00487	1.08823
4	-.00002	-.00445	-2.31475
5	-.00002	-13.39301	1.47278
6	-.00001	.00000	-3.57861
7	.00000	.00000	13.78931
8	.00000	-130.73584	25.78931
9	.00000	.00000	.00000
10	.00000	.00000	.00000
11	.00000	.00000	.00000
12	.00000	.00000	.00000

SUPPORT REACTIONS

JOINT	FORCE X	FORCE Y	MOMENT Z
1	-.23270	5.91369	.00000
7	.09614	16.38169	.00000
9	.52235	48.66123	1.74115
10	-.38579	37.10798	.00000
11	-1.71773	-.03229	-5.72577
12	1.71773	-.03229	-5.72578

MEMBER FORCES

MEMBER	JOINT	FORCE AXIAL	FORCE SHEAR Y	MOMENT BENDING Z
1	1	.23270	-5.31368	.00000
1	2	.23270	-5.31368	-35.48206
2	2	.23270	24.08832	35.48206
2	3	.23270	24.08832	60.36322
3	3	-.28965	-.57491	-.34555
3	4	-.28965	-.57491	-8.85300
4	4	.09614	-13.68289	-50.51750
4	5	.09614	-13.68289	-31.57982
5	5	.09614	6.31711	31.57982
5	6	.09614	6.31711	6.32286
6	6	.09614	6.38169	16.58022
6	7	.09614	6.38169	60.00001
7	7	.00000	-10.00000	-60.00001
7	8	.00000	-10.00000	.00000
8	9	-48.66123	.52235	1.74115
8	3	-48.66123	.52235	3.48232
9	-10	-37.10798	-.38579	.00000
9	4	-37.10798	-.38579	-4.62950
10	6	-.03229	-1.71773	-11.45155
10	12	-.03229	-1.71773	-5.72578
11	11	.03229	-1.71773	-5.72577
11	6	.03229	-1.71773	-11.45154

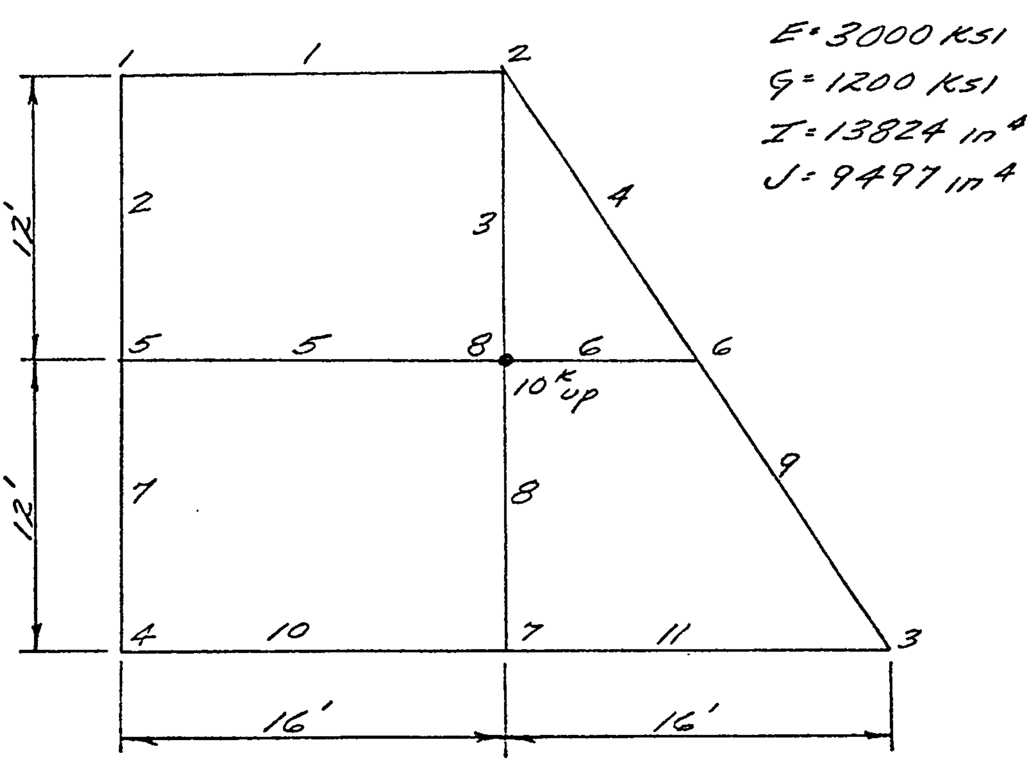
Member	Text	ASACS
1	0.0	0.0
1		-35.48
2		35.48
2	60.86	60.86
3	-64.34	-64.34
3	55.15	55.15
4	-50.52	-50.52
4	-31.58	-31.58
5	31.58	31.58
5	-6.32	-6.32
6	16.58	16.58
6	60.0	60.0
7	-60.00	-60.00
7	0.0	0.0
8	1.74	1.74
8	3.48	3.48
9	0.0	0.0
9	-4.63	-4.63
10	-11.45	-11.45
10	-5.72	-5.72
11	-5.72	-5.72
11	-11.45	-11.45

Comparison of Results
Plane Frame
Figure 4.2

4.3 Plane Grid Test Problem

This problem tests the accuracy of the plane grid program. The grid, as shown in Figure 4.3, is composed of eleven members and eight joints. It is simply supported at the four corner joints only. The grid has a ten kip load at the center joint. The theoretical solution can be found in

Matrix Methods of Structural Analysis
C. K. Wang
Page 241



Plane Grid Test Problem
Figure 4.3

```
PROBLEM  PLANE GRID TEST
S   REFENENCE  MATRIX METHODS OF STRUCTURAL ANALYSIS
S              C. K. WANG
S              PAGE 241
E=3000.
G=1200.
JOINT COORDINATES
1,0.,0.,S
2,192.,0.,S
3,384.,288.,S
4,0.,288.,S
5,0.,144.
6,288.,144.
7,192.,288.,
8,192.,144.
MEMBER INCIDENCES
1,1,2,     1.,13824.,9497.
2,1,5,     1.,13824.,9497.
3,2,8,     1.,13824.,9497.
4,2,6,     1.,13824.,9497.
5,5,8,     1.,13824.,9497.
6,8,6,     1.,13824.,9497.
7,5,4,     1.,13824.,9497.
8,8,7,     1.,13824.,9497.
9,6,3,     1.,13824.,9497.
10,4,7,    1.,13824.,9497.
11,7,3,    1.,13824.,9497.
RELEASE JOINT 1, MOMENT X
RELEASE JOINT 2, MOMENT X
RELEASE JOINT 3, MOMENT X
```

```
RELEASE JOINT 4, MOMENT X
RELEASE JOINT 1, MOMENT Y
RELEASE JOINT 2, MOMENT Y
RELEASE JOINT 3, MOMENT Y
RELEASE JOINT 4, MOMENT Y
LOAD TEST
JOINT 8, FORCE Z 10.
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION Z	ROTATION X	ROTATION Y
1	.00000	.00027	-.00007
2	.00000	.00076	.00005
3	.00000	-.00031	.00057
4	.00000	-.00025	-.00055
5	.02333	.00000	-.00052
6	.06726	.00023	.00037
7	.07138	-.00034	-.00000
8	.09231	.00025	.00005

SUPPORT REACTIONS

JOINT	FORCE Z	MOMENT X	MOMENT Y
1	-.56283	.00000	.00000
2	-4.43717	.00000	.00000
3	-2.78142	.00000	.00000
4	-2.21659	.00000	.00000

MEMBER FORCES

MEMBER	JOINT	FORCE SHEAR	MOMENT TORSIONAL	MOMENT BENDING Z
1	1	.10685	30.26437	-35.82026
1	2	-.10685	30.26437	15.30485
2	1	-.66968	-35.82027	-30.26436
2	5	.66965	-35.82027	126.69873
3	2	-2.94605	-.41188	59.33490
3	8	2.94605	-.41188	365.18390
4	2	-1.38227	-3.04833	-32.90622
4	6	1.38227	-3.04833	272.13078
5	5	-1.61602	14.92182	33.12831
5	8	1.61602	14.92182	277.53245
6	8	2.97430	-2.88098	-280.89377
6	6	-2.97430	-2.88098	-4.63928
7	5	.94634	-2.69195	-141.62054
7	4	-.94634	-2.69195	5.05904
8	8	2.45962	-3.77314	-347.38104
8	7	-2.45962	-3.77314	-6.80486
9	6	1.59203	-8.50648	-271.95453
9	3	-1.59203	-8.50648	-3.57279
10	4	-1.27024	-5.05904	2.69197
10	7	1.27024	-5.05904	241.19437
11	7	1.18939	1.74583	-237.42123
11	3	-1.18938	1.74583	9.05963

Member	Text	ASACS
1	-35.82	-35.82
1	15.30	15.30
2	-30.26	-30.26
2	126.70	126.70
3	59.33	59.33
3	365.18	365.18
4	-32.91	-32.91
4	272.12	272.13
5	33.13	33.13
5	277.53	277.53
6	388.88	388.88
6	-4.64	-4.64
7	-141.62	-141.62
7	5.06	5.06
8	-347.38	-347.38
8	-6.80	-6.80
9	-271.95	-271.95
9	-3.57	-3.57
10	2.69	2.69
10	241.19	241.19
11	-237.42	-237.42
11	9.06	9.06

Comparison of Results
Plane Grid
Table 4.3

CHAPTER 5

CONCLUSIONS AND RECOMENDATIONS

Academic Structural Analysis Computer System is functioning properly as shown by the test problems in Chapter 4. All objectives of the program system have been acheived. These objectives were to obtain a system that could be used in either batch or time-sharing environment with minimal memory requirement and computational time.

There are three obvious area in which further development of ASACS is needed. At the present time all loading is handled in terms of joint loads. Algorithms need to be developed to handle member loads. This will make the program more versatile and reduce the amount of manual work needed for solution.

Secondly, more sophisticated error analysis schemes need to be employed. Only gross input errors are now detected prior to execution. A more refined analysis of the errors and potentially hazardous conditions is needed and print more specific diagnostic messages. When operating in a time-sharing environment this routine should return to the location at which the error was made and allow the user to reenter the data.

Last, the matrix inversion routine itself needs to be improved. ASACS uses a Gaussian elimination algorithm which requires a second array of equal size as the matrix being inverted. A more efficient routine both in terms of computational time and memory requirement needs to be incorporated.

REFERENCES

Fenves, Steven J., et al. STRESS: A Reference Manual. M.I.T. Press; Cambridge Mass.; 1965.

Logcher, Robert D., et al. ICES STRUDL II The Structural Design Language Engineering User's Manual. M.I.T. Press; Cambridge Mass.; 1968.

Przemieniecki, J. S. Theory of Matrix Structural Analysis. McGraw-Hill Book Co.; New York; 1968.

Rubinstein, Moshe F. Matrix Computer Analysis of Structures. Prentice-Hall; Englewood Cliffs, N.J.;1966.

Timoshenko, S. Young, D.H., Theory of Structures. McGraw-Hill Book Co.; New York; 1965.

Wang, C.K. Matrix Methods of Structural Analysis. International Textbook Co.; Scranton, Pa.; 1970.

Appendix A

User's Manual

1.0 Introduction

Academic Structural Analysis Computer System, ASACS, is a general purpose computer software system for the analysis of plane trusses, plane frames, and plane grids. The geometry of the structure, the member properties, and the loads are provided by the user in a problem oriented language. Member forces, joint displacements, and reactions are listed in an easily understood tabular form. A linear, elastic, static, small-deflection stiffness analysis is used to calculate these values.

ASACS is ideally suited to the analysis of small to medium-sized structures in a time-sharing or limited computer environment. Minimal stroage requirement and computational time are the features of ASACS that enable both student users and practicing engineers to obtain rapid turnaround and reliability in structural problem solution.

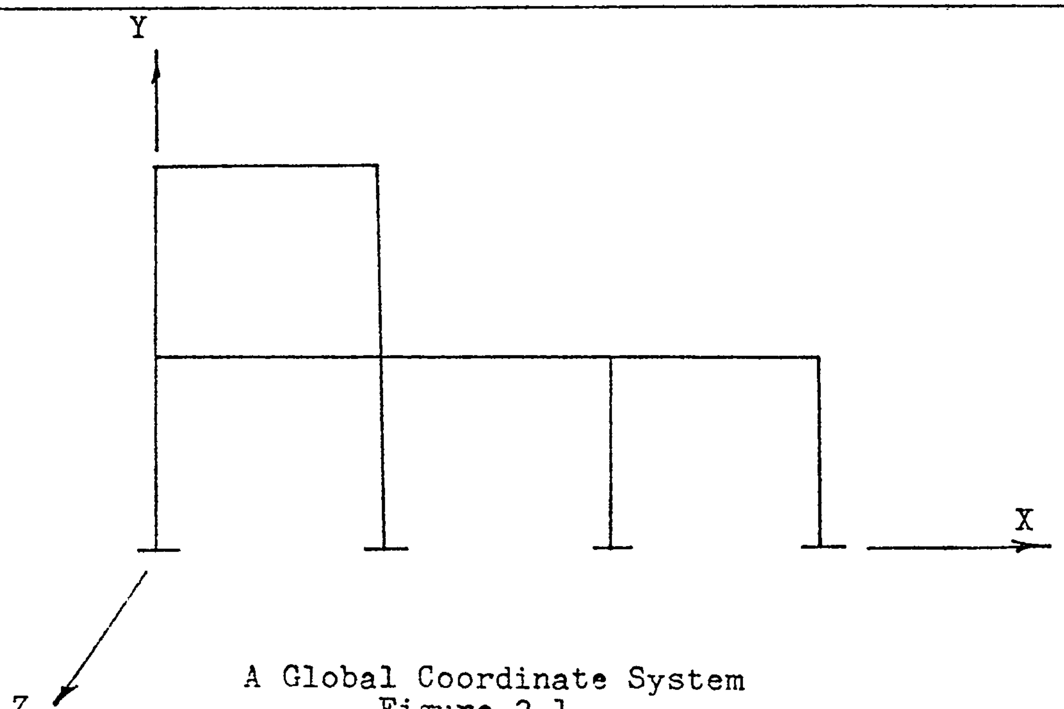
A complete, concise set of operating instructions, description of the output, and example problems are contained in the sections that follow.

2.0 Coordinate Reference Systems

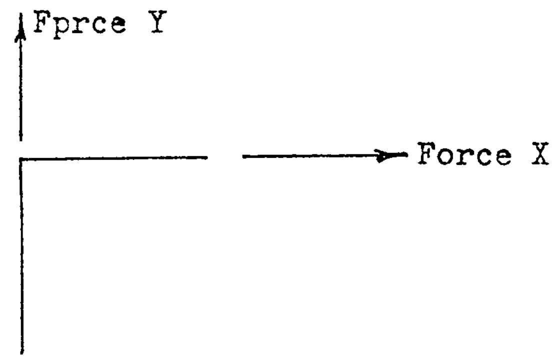
Coordinate reference systems must be selected to describe the geometry of a structure and allow interpretation of the resulting output. The basis of the coordinate systems selected is the right-hand, orthogonal, Cartesian system. ASACS requires the use of two coordinate systems: a global system for the overall structure and local coordinate systems for the individual members.

2.1 Global Coordinate System

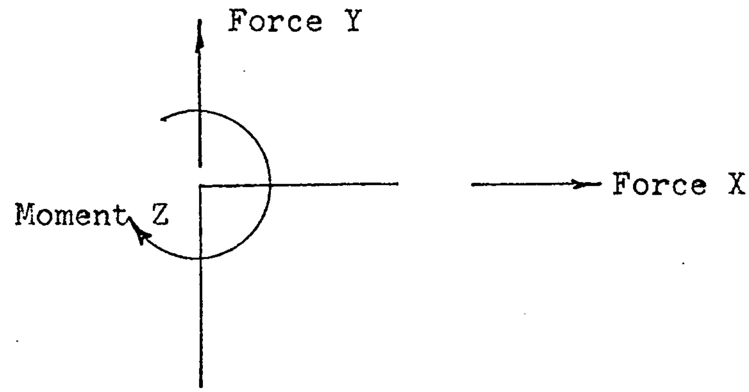
The global coordinate system is generally chosen such that the coordinate axes coincide with the major dimensions of the structure. An example is given in Figure 2.1. Figure 2.2 shows the positive global force and displacement components for each of the structural types.



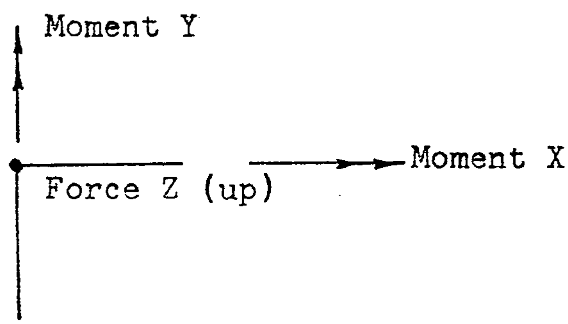
A Global Coordinate System
Figure 2.1



Positive Truss Forces
Figure 2.2a



Positive Frame Forces
Figure 2.2b

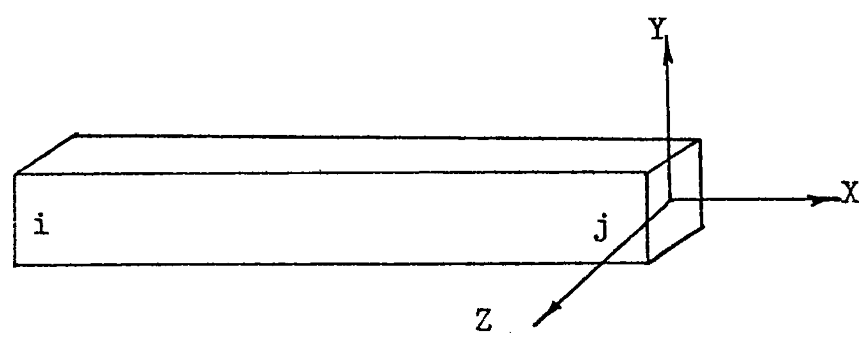


Positive Grid Forces
Figure 2.2c

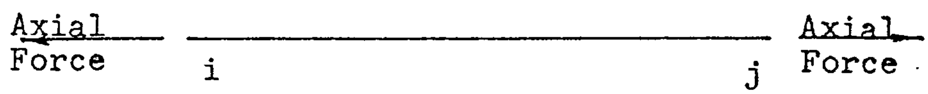
The global coordinate system is used when specifying the location of the joints in the structure and when results pertaining to joint displacements and reactions are given.

2.2 Local Coordinate System

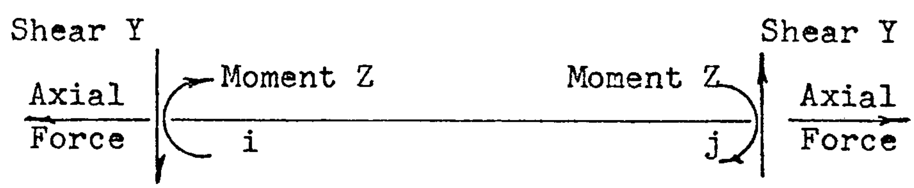
A local coordinate system is associated with each member. Information pertaining to the individual members is given in this local coordinate system. The local x-axis for a member coincides with the centroidal axis for that member. The positive direction is taken along the member from the initial joint, i, to the terminal joint, j. The y and z axes coincide with the principal axes of the member cross-section as shown in Figure 2.3. The positive force components for the members in each type of structure are shown in Figure 2.4.



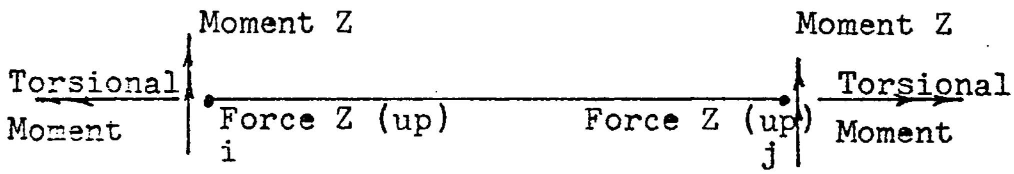
A Local Coordinate System
Figure 2.3



Positive Truss Forces
Figure 2.4a



Positive Frame Forces
Figure 2.4b



Positive Grid Forces
Figure 2.4c

3.0 Data Entry to ASACS

The user supplies the data specifying the geometry, supports, and loads on the structure. All input is free-field, with commas separating items of data. Imbedded blanks are permissible as long as the entire statement can be fit on one card. Continuation cards are not permitted. A dollar sign ($) in column one denotes a comment which will be printed in the input listing, but otherwise ignored by the system. The general organization of a data deck is shown in Figure 3.1.

```
PROBLEM
E=
G=
JOINT COORDINATES
            joint coordinate cards
MEMBER INCIDENCES
            member incidence and property
RELEASES
LOAD 1
            load data
SOLVE
FINISH
```

Data Deck Organization
Figure 3.1

In the description of each of the input statements the portion which <u>must</u> be entered is underlined. Other parts of the statement are optional and left to the discretion of the user. When reading this manual and using ASACS it should be kept in mind that the entire command is not always required; an abbreviated form can be used.

3.1 Problem Card

The first card in each set of data is the problem card. This card identifies the beginning of the data and has the general form

<u>PROB</u>LEM title

Example:

PROBLEM COLUMN LINE A

PROB ROOF TRUSS OVER EAST WING

3.2 Material Property Cards

Young's modulus and the shear modulus must be specified for the material to be used. It is assumed that all members are of the same material. The shear modulus is to be entered only for grids. Young's modulus is required for all structures. These constant definitions take the general form

<u>E=xxxxx</u> (Young's modulus)

<u>G=xxxxx</u> (shear modulus)

Example:

E=29000.

G=12000.

3.3 Joint Coordinates

The start of the joint coordinates is signified by the card

JOINT COORDINATES

The cards that follow this statement specify the joint number, the x and y coordinates, and whether the joint is free to displace or a support. If the joint is not specified free or support, a free joint is assumed. All joint coordinates must be entered at the same time. The general form of these statements is

joint, x, y, SUPPORT / FREE ←

Example:

JOINT COORDINATES

1,54.8, 37.01, S

2, 0.0, 43.67

When numbering the joints, no numbers can be excluded. If any of the degrees of freedom at a joint are constrained the joint must be specified as a support. The non-restrained degrees of freedom are later released using RELEASE cards.

3.4 Member Incidence and Property Cards

The start of the member incidence and property data is signified by the statement

MEMBER INCIDENCES

On the statements that follow are the member number, initial and terminal joints, cross-sectional area, plane moment of inertia, and the torsional constant. Only the area is required for trusses and the torsional constant is required only for grids. The general form of this statement is

member, i, j, A, I, J

Example:

1, 4, 7, 14.2, 250., 16.8

This is member 1 connecting joint 4 to joint 7. The area is 14.2, I is 250., and J is 16.8. All member incidences must be entered at the same time and no numbers excluded.

3.5 Releases

3.5.1 Joint Releases

On the joint coordinate cards some of the joints were specified as support joints. When this was done all the degrees of freedom were constrained. This is not necessarily the case for a given structure. One or several of the components may be free to displace. These are specified using release cards. One card is required for each degree of freedom that is released. The releases are in the global coordinate system and the general form is

RELEASE JOINT no., FORCE X / FORCE Y / FORCE Z / MOMENT X / MOMENT Y / MOMENT Z

Example:

RELEASE JOINT 5, FORCE Y

For any of the structures, the only degrees of freedom which can be released are the components shown in Figure 2.2.

3.5.2 Member Releases

On the member incidence cards when a member was specified it was assumed to be continuous with the joint. In the actual structure there may be rotational hinges at the ends of the member. These are also specified using release cards. The general form of the member release statement is

RELEASE MEMBER no., { INITIAL END / TERMINAL END / BOTH ENDS } M_{zz}
TORSION

Example:

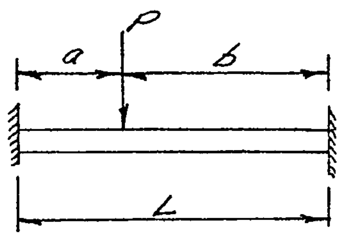
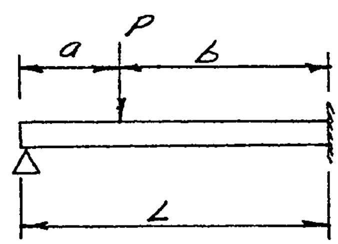
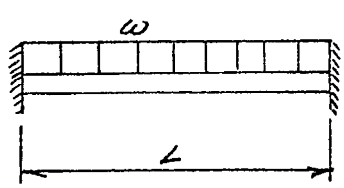
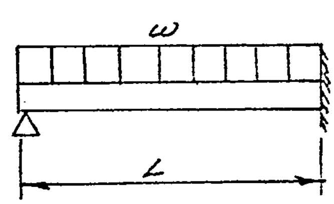
RELEASE MEMBER 6, TORSION

Only frames can have the initial and terminal end moments released and only grids have the torsional moment release.

3.6 Structural Loading

All loads acting on the structure are entered to ASACS as joint loads. In the actual structure if a member is loaded, the load must be changed into an equivalent joint load. When interpreting the output these joint loads must be added to the results for the respective member in order to obtain the true forces for that member. Figure 3.2 is a chart of fixed-end moments the user will commonly encounter.

The first statement for load data identifies the

Shear	Moment	Beam & Load	Moment	Shear
$\frac{Pb^2(3a+b)}{L^3}$	$\frac{Pab^2}{L^2}$		$\frac{Pa^2b}{L^2}$	$\frac{Pa^2(a+3b)}{L^3}$
$\frac{Pb^2(a+2L)}{2L^3}$	0		$\frac{Pab(a+L)}{2L^2}$	$\frac{Pa(3L^2-a^2)}{2L^3}$
$\frac{wL}{2}$	$\frac{wL^2}{12}$		$\frac{wL^2}{12}$	$\frac{wL}{2}$
$\frac{3wL}{8}$	0		$\frac{wL^2}{8}$	$\frac{5wL}{8}$

Fixed-End Moments
Figure 3.2

load set. It has the general form

LOAD title

The load acting on a joint is specified as

```
                   FORCE X xxxxx
                   FORCE Y xxxxx
                   FORCE Z xxxxx
JOINT no.,         MOMENT X xxxx
                   MOMENT Y xxxx
                   MOMENT Z xxxx
```

There is to be one statement for each force or moment component acting at the joint. If there is more than one card for a particular component, only the last card is used. For a given structure, the components which can be loaded are those indicated in Figure 2.2. An invalid component will be ignored by the system. The last card in a particular load condition is

SOLVE

A typical load condition might appear as follows

```
LOAD WIND FROM RIGHT
JOINT 5, FORCE X -5.6
JOINT 5, MOMENT Z 23.7
JOINT 2, FORCE Y 10.0
SOLVE
```

Multiple load conditions can be entered by placing complete data decks back-to-back. Each set must begin with LOAD and end with SOLVE.

4.0 Output

The output from ASACS includes joint displacements, joint reactions, and member forces. Output pertaining to joints is in the global coordinate system. Output pertaining to members is in the local coordinate system.

5.0 Additional Features

More than one problem can be run at a time by placing the complete data decks one after the other. The FINISH statement is to follow the last data deck. If a fatal error is made in one problem, an error message is printed and the system searches for the next problem or the FINISH statement.

6.0 Sample Problems

On the pages that follow are complete listings of sample problems demonstrating the various features of the ASACS system.

```
**************************************************************
*                                                            *
*    ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM            *
*                                                            *
*                     PLANE TRUSS                            *
*                     RELEASE 1.0                            *
*                                                            *
*                UNIVERSITY OF HOUSTON                       *
*                                                            *
*           DEPARTMENT OF CIVIL ENGINEERING                  *
*                                                            *
**************************************************************
```

```
PROBLEM   ***** PLANE TRUSS TEST PROBLEM  *****
$ REFERENCE: MECHANICS FOR ENGINEERS
$            BEER AND JOHNSON
$            EXAMPLE 6.1
E=30000.
JOINT COORDINATES
1,0.,96.
2,144.,96.
3,288.,96.,S
4,72.,0.
5,216.,0.,S
MEMBER INCIDENCES
1,1,2,10.
2,2,3,10.
3,1,4,10.
5,2,5,10.
4,4,2,10.
6,5,3,10.
7,4,5,10.
RELEASE JOINT 5, FORCE X
LOADING LIVE
JOINT 2, FORCE Y -1.0
JOINT 1, FORCE Y -2.0
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION X	TRANSLATION Y	TRANSLATION Z
1	-.00324	-.02587	
2	-.00252	-.00814	
3	.00000	.00000	
4	.00727	-.01673	
5	.00583	.00000	

SUPPORT REACTIONS

JOINT	FORCE X	FORCE Y	FORCE Z
3	.00000	-7.00000	
5	.00000	10.00000	

MEMBER FORCES

MEMBER	FORCE AXIAL	SHEAR Y	SHEAR Z
1	1.50000		
2	5.25000		
3	-2.50000		
4	2.50000		
5	-3.75000		
6	-8.75000		
7	-3.00000		

```
*****************************************************
*                                                   *
*    ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM   *
*                                                   *
*                    PLANE FRAME                    *
*                    RELEASE 1.0                    *
*                                                   *
*               UNIVERSITY OF HOUSTON               *
*                                                   *
*           DEPARTMENT OF CIVIL ENGINEERING         *
*                                                   *
*****************************************************
```

```
PROBLEM  PLANE FRAME TEST PROBLEM
$   REFERENCE  ELEMENTARY STRUCTURAL ANALYSIS
$              NORRIS AND WILBUR
$              PAGE 459
E=1.0
JOINT COORDINATES
1,0.,0.,S
2,6.,0.
3,10.,0.
4,26.,0.
5,32.,0.
6,38.,0.
7,50.,0.,S
8,56.,0.
9,10.,-10.,S
10,26.,-12.,S
11,38.,-10.,S
12,38.,10.,S
MEMBER INCIDENCES
1,1,2,99999.,10.
2,2,3,99999.,10.
3,3,4,99999.,20.
4,4,5,99999.,15.
5,5,6,99999.,15.
6,6,7,99999.,15.
7,7,8,99999.,15.
8,9,3,99999.,8.
9,10,4,99999.,8.
10,6,12,99999.,8.
11,11,6,99999.,8.
```

```
RELEASE JOINT 1, MOMENT Z
RELEASE MEMBER 9, INITIAL END
RELEASE JOINT 7, MOMENT Z
LOADING TEST
JOINT 2, FORCE Y -30.
JOINT 5, FORCE Y -20.
JOINT 8, FORCE Y -10.
JOINT 3, FORCE Y -24.
JOINT 3, MOMENT Z 64.
JOINT 4, FORCE Y -24.
JOINT 4, MOMENT Z -64.
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION X	TRANSLATION Y	ROTATION Z
1	.00000	.00000	6.65662
2	.00001	-18.65045	-3.98800
3	.00002	-.00487	1.08823
4	-.00002	-.00445	-2.31475
5	-.00002	-13.89801	1.47278
6	-.00001	.00000	-3.57861
7	.00000	.00000	13.78931
8	.00000	-130.73584	25.78931
9	.00000	.00000	.00000
10	.00000	.00000	.00000
11	.00000	.00000	.00000
12	.00000	.00000	.00000

SUPPORT REACTIONS

JOINT	FORCE X	FORCE Y	MOMENT Z
1	-.23270	5.91368	.00000
7	.09614	16.38169	.00000
9	.52235	48.66123	1.74115
10	-.38579	37.10798	.00000
11	-1.71773	-.03229	-5.72577
12	1.71773	-.03229	-5.72578

MEMBER FORCES

MEMBER	JOINT	FORCE AXIAL	FORCE SHEAR Y	MOMENT BENDING Z
1	1	.23270	-5.91368	.00000
1	2	.23270	-5.91368	-35.48206
2	2	.23270	24.08632	35.48206
2	3	.23270	24.08632	60.86322
3	3	-.28965	-.57491	-.34555
3	4	-.28965	-.57491	-8.85300
4	4	.09614	-13.68289	-50.51750
4	5	.09614	-13.68289	-31.57982
5	5	.09614	6.31711	31.57982
5	6	.09614	6.31711	6.32286
6	6	.09614	6.38169	16.58022
6	7	.09614	6.38169	60.00001
7	7	.00000	-10.00000	-60.00001
7	8	.00000	-10.00000	.00000
8	9	-48.66123	.52235	1.74115
8	3	-48.66123	.52235	3.48232
9	-10	-37.10798	-.38579	.00000
9	4	-37.10798	-.38579	-4.62950
10	6	-.03229	-1.71773	-11.45155
10	12	-.03229	-1.71773	-5.72578
11	11	.03229	-1.71773	-5.72577
11	6	.03229	-1.71773	-11.45154

```
*****************************************************
*                                                   *
*    ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM   *
*                                                   *
*                   PLANE  GRID                     *
*                   RELEASE 1.0                     *
*                                                   *
*               UNIVERSITY OF HOUSTON               *
*                                                   *
*          DEPARTMENT OF CIVIL ENGINEERING          *
*                                                   *
*****************************************************
```

```
PROBLEM  PLANE GRID TEST
$   REFENENCE  MATRIX METHODS OF STRUCTURAL ANALYSIS
$              C. K. WANG
$              PAGE 241
E=3000.
G=1200.
JOINT COORDINATES
1,0.,0.,S
2,192.,0.,S
3,384.,288.,S
4,0.,288.,S
5,0.,144.
6,288.,144.
7,192.,288.,
8,192.,144.
MEMBER INCIDENCES
1,1,2,    1.,13824.,9497.
2,1,5,    1.,13824.,9497.
3,2,8,    1.,13824.,9497.
4,2,6,    1.,13824.,9497.
5,5,8,    1.,13824.,9497.
6,8,6,    1.,13824.,9497.
7,5,4,    1.,13824.,9497.
8,8,7,    1.,13824.,9497.
9,6,3,    1.,13824.,9497.
10,4,7,   1.,13824.,9497.
11,7,3,   1.,13824.,9497.
RELEASE JOINT 1, MOMENT X
RELEASE JOINT 2, MOMENT X
RELEASE JOINT 3, MOMENT X
```

```
RELEASE JOINT 4, MOMENT X
RELEASE JOINT 1, MOMENT Y
RELEASE JOINT 2, MOMENT Y
RELEASE JOINT 3, MOMENT Y
RELEASE JOINT 4, MOMENT Y
LOAD TEST
JOINT 8, FORCE Z 10.
SOLVE
```

JOINT DISPLACEMENTS

JOINT	TRANSLATION	ROTATION	
	Z	X	Y
1	.00000	.00027	-.00007
2	.00000	.00078	.00005
3	.00000	-.00031	.00057
4	.00000	-.00025	-.00055
5	.02383	.00000	-.00052
6	.06726	.00023	.00037
7	.07138	-.00034	-.00000
8	.09231	.00025	.00005

SUPPORT REACTIONS

JOINT	FORCE Z	MOMENT X	MOMENT Y
1	-.56283	.00000	.00000
2	-4.43717	.00000	.00000
3	-2.78142	.00000	.00000
4	-2.21859	.00000	.00000

MEMBER FORCES

MEMBER	JOINT	FORCE SHEAR	MOMENT TORSIONAL	MOMENT BENDING Z
1	1	.10685	30.26437	-35.82026
1	2	-.10685	30.26437	15.30485
2	1	-.66968	-35.82027	-30.26436
2	5	.66968	-35.82027	126.69373
3	2	-2.94805	-.41188	59.33490
3	8	2.94805	-.41188	365.18390
4	2	-1.38227	-3.04833	-32.90622
4	6	1.38227	-3.04833	272.13078
5	5	-1.61802	14.92182	33.12831
5	8	1.61802	14.92182	277.53245
6	8	2.97430	-2.88098	-280.89377
6	6	-2.97430	-2.88098	-4.63928
7	5	.94834	-2.69195	-141.62054
7	4	-.94834	-2.69195	5.05904
8	8	2.45962	-3.77314	-347.38104
8	7	-2.45962	-3.77314	-6.80486
9	6	1.59203	-3.50648	-271.95453
9	3	-1.59203	-3.50648	-3.57279
10	4	-1.27024	-5.05904	2.69197
10	7	1.27024	-5.05904	241.19437
11	7	1.18938	1.74583	-237.42123
11	3	-1.18938	1.74583	9.05963

Appendix B

Plane Truss Program Listing

```
      DIMENSIO  S(50,50),D(50,50),PP(50)
      DIMENSION DISP(50)
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      DATA JAST /1H*/, JSEM/1H;/, JCMA /1H,/, JPER/1H./, JDOL/1H$/,
     1   JLB/1H#/, JCOL /1H:/, JBLK/1H /, JRP/1H)/, JLP/1H(/, JEQ /1H=/
      DATA JAC/1HA/, JBC/1HB/, JCC/1HC/, JDC/1HD/, JEC/1HE/, JFC/1HF/,
     1     JGC/1HG/, JHC/1HH/, JIC/1HI/, JJC/1HJ/, JKC/1HK/, JLC/1HL/,
     1     JMC/1HM/, JNC/1HN/, JOC/1HO/, JPC/1HP/, JQC/1HQ/, JRC/1HR/,
     1     JSC/1HS/, JTC/1HT/, JUC/1HU/, JVC/1HV/, JWC/1HW/, JXC/1HX/,
     1     JYC/1HY/, JZC/1HZ/
      DATA JPLS/1H+/, JMIN/1H-/, JDIV/1H//, JGT/1H>/, JLT/1H</
      DATA NUM(1) /1H1/, NUM(2) /1H2/, NUM(3)  /1H3/, NUM(4)/1H4/,
     1     NUM(5) /1H5/, NUM(6) /1H6/, NUM(7)  /1H7/, NUM(8)/1H8/,
     1     NUM(9) /1H9/, NUM(10) /1H0/
      INTEGER PP
      INTEGER R,W
      MJ=50
      W=6
      R=5
 26   WRITE(W,130)
      WRITE(W,131)
      WRITE(W,132)
      WRITE(W,133)
      WRITE(W,132)
      WRITE(W,138)
      WRITE(W,134)
      WRITE(W,132)
      WRITE(W,135)
      WRITE(W,132)
      WRITE(W,136)
      WRITE(W,132)
      WRITE(W,131)
      WRITE(W,139)
      CALL READER
      CALL INTERP
 55   CONTINUE
      NEWPRO=0
      E=0.0
      CALL READER
      CALL INTERP
      IF(E.EQ.0.0)GO TO 29
      NOJOI=0
      CALL READER
      CALL INTERP
      IF(NOJOI.EQ.0)GO TO 30
      NOMEN=0
      NOLOAD=1
      CALL INTERP
      IF(NOMEN.EQ.0)GO TO 31
```

```
      ICOUN1=NOJOI*2
      ICOUN2=1
      DO 3 I=1,NOJOI,1
      IF(J(I,1).NE.JSC)GO TO 8
5     J(I,1)=ICOUN1
      ICOUN1=ICOUN1-1
      J(I,2)=ICOUN1
      ICOUN1=ICOUN1-1
      GO TO 3
8     J(I,1)=ICOUN2
      ICOUN2=ICOUN2+1
      J(I,2)=ICOUN2
      ICOUN2=ICOUN2+1
3     CONTINUE
      ICOUN2=ICOUN2-1
      ICOUN1=ICOUN1+1
      NOREL=0
51    CALL INTERP
      IF(NOREL.EQ.0)GO TO 35
      LOC=0
      IF(JBUF(8).NE.JJC)GO TO 32
      JOINT=DFCODI(13,L)
      L=L+1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JYC)LOC=2
      IF(LOC.EQ.0)GO TO 32
      ICOUN1=ICOUN1+1
      ICOUN2=ICOUN2+1
      IHOLD=J(JOINT,LOC)
       DO 37 LOK=1,NOJOI,1
      IF(J(LOK,1).NE.ICOUN2.AND.J(LOK,2).NE.ICOUN2)GO TO 37
      IF(J(LOK,1).EQ.ICOUN2)LOK1=1
      IF(J(LOK,2).EQ.ICOUN2)LOK1=2
      LOK3=LOK
      GO TO 38
37    CONTINUE
38    CONTINUE
      J(JOINT,LOC)=ICOUN2
      J(LOK3,LOK1)=IHOLD
36    CALL READER
      NOREL=0
      GO TO 51
35    NOREL=1
      N1=NOJOI*2
      DO 10 I=1,N1,1
      DO 10 K=1,N1,1
10    B(I,K)=0.0
      DO 12 I=1,NOMEN,1
      A(I,1)=-A(I,3)*A(I,4)
      A(I,2)=-A(I,3)*A(I,5)
       LOCCJJ=J(I,4)
      LOCCJK=J(I,5)
      DO 12 K=4,5,1
      N=K-3
      K1=J(LOCCJJ,N)
      K2=J(LOCCJK,N)
      DO 12 L=1,2,1
      K3=J(LOCCJJ,L)
      K4=J(LOCCJK,L)
      B(K1,K3)=B(K1,K3)+(-A(I,   K)*A(I,L))
      B(K1,K4)=B(K1,K4)+(A(I,   K)*A(I,L))
      B(K2,K3)=B(K2,K3)+(A(I,   K)*A(I,L))
```

```
      B(K2,K4)=B(K2,K4)-(A(I,  K)*A(I,L))
12    CONTINUE
      CALL INVERT(ICOUN2,IFLAG,W,B,D,PP,MJ)
      IF(IFLAG.EQ.1)GO TO 29
24    NOLOAD=0
      ISOLV=0
      NOP=2*NOJOI
      DO 17 I=1,NOP,1
17    P(I)=0.0
      CALL INTERP
      IF(IEND.EQ.1)GO TO 27
      IF(NEWPRO.EQ.1)GO TO 55
      IF(NOLOAD.EQ.0)GO TO 33
      IF(JFOUND.EQ.0)GO TO 25
      IF(ISOLV.NE.1)GO TO 29
      DO 18 I=1,ICOUN2,1
      DISP(I)=0.0
      DO 18 K=1,ICOUN2,1
18    DISP(I)=DISP(I)+B(I,K)*P(K)
      DO 19 I=ICOUN1,NOP,1
      P(I)=-P(I)
      DISP(I)=0.0
      DO 19 K=1,ICOUN2,1
19    P(I)=P(I)+B(I,K)*DISP(K)
      DO 20 L=1,NOMEN,1
      MFL1=J(L,4)
      MFL2=J(L,5)
      MFL3=J(MFL1,1)
      MFL4=J(MFL1,2)
      MFL5=J(MFL2,1)
      MFL6=J(MFL2,2)
      A(L,3)=A(L,1)*DISP(MFL3)+A(L,2)*DISP(MFL4)-A(L,1)*DISP(MFL5)
      A(L,3)=A(L,3)-A(L,2)*DISP(MFL6)
20    CONTINUE
      WRITE(W,126)
      WRITE(W,120)
      WRITE(W,121)
      DO 21 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
21    WRITE(W,122)L,          DISP(MFL1),DISP(MFL2)
      DO 39 L=1,ICOUN2,1
39    P(L)=0.0
      WRITE(W,127)
      WRITE(W,123)
      WRITE(W,121)
      DO 22 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
      IF(MFL1.LT.ICOUN1.AND.MFL2.LT.ICOUN1)GO TO 22
      WRITE(W,122)L,          P(MFL1),P(MFL2)
22    CONTINUE
      WRITE(W,128)
      WRITE(W,124)
      WRITE(W,125)
      DO 23 L=1,NOMEN,1
23    WRITE(W,122)L,          A(L,3)
      WRITE(W,139)
      CALL READER
      GO TO 24
29    WRITE(W,142)
      GO TO 27
```

```
30      WRITE(6,143)
        GO TO 25
31      WRITE(6,144)
        GO TO 25
32      WRITE(6,145)
        GO TO 25
33      WRITE(6,146)
25      IF(IEND.EQ.1)GO TO 27
        CALL READER
        CALL INTERP
        IF(NEWPRO.EQ.1)GO TO 55
        GO TO 25
120     FORMAT(' ',2X,'JOINT            ',15X,'TRANSLATION')
121     FORMAT(' ',20X,'X',13X,'Y',13X,'Z'/)
122     FORMAT(' ',I6,7X, 2F14.5)
123     FORMAT(' ',2X,'JOINT            ',18X,'FORCE')
124     FORMAT(' ',2X,'MEMBER           ',18X,'FORCE')
125     FORMAT(' ',18X,'AXIAL',8X,'SHEAR Y',7X,'SHEAR Z'/)
126     FORMAT(///'1','JOINT DISPLACEMENTS'//)
127     FORMAT(///'1','SUPPORT REACTIONS'//)
128     FORMAT(///'1','MEMBER FORCES'//)
130     FORMAT('1',////////////////)
131     FORMAT(' ',10X,'*************************************************
       6****')
132     FORMAT(' ',10X,'*',52X,'*')
133     FORMAT(' ',10X,'*     ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM
       6    *')
134     FORMAT(' ',10X,'*',20X,'RELEASE 1.0',21X,'*')
135     FORMAT(' ',10X,'*',15X,'UNIVERSITY OF HOUSTON',16X,'*')
136     FORMAT(' ',10X,'*',10X,'DEPARTMENT OF CIVIL ENGINEERING',11X,'*')
138     FORMAT(' ',10X,'*',20X,'PLANE TRUSS',21X,'*')
139     FORMAT(1H1)
142     FORMAT(' ','INSUFFICIENT DATA OR INPUT ERROR')
143     FORMAT(' ','ERROR IN JOINT COORDINATES')
144     FORMAT(' ','ERROR IN MEMBER INCIDENCES')
145     FORMAT(' ','ERROR IN JOINT RELEASES')
146     FORMAT(' ','ERROR IN LOADING')
27      STOP
        END

F COMPILATION:          NO  DIAGNOSTICS.
```

/76-13:33:22 (,0)

INVERT ENTRY POINT 003334

D: CODE(1) 000436; DATA(0) 000047; BLANK COMMON(2) 000000

FERENCES (BLOCK, NAME)

OT
OTR
SUB
US
2$
R3$

IGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

007 106F	0001	000027 110G	0001	000030 113G	0001	000140 133G
200 155G	0001	000220 170G	0001	000314 20L	0001	000301 201G
072 34L	0001	000040 40L	0001	000213 91L	0000 R	000006 A
005 I	0003	000020 INJP$	0000 I	000003 J	0000 I	000001 K
000 Z						

```
      SUBROUTINE INVERT(N,IFLAG,W,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      INTEGER Z,W
      IFLAG=0
      DO 5 K=1,N,1
      DO 5 L=1,N,1
5     D(K,L)=0.0
      J=1
40    Z=J+1
      CALL PIVOT(J,N,C,D,P,MJ)
      IF(C(J,J))34,30,34
30    WRITE(W,106)
      IFLAG=1
      GO TO 20
34    DO 60 K=Z,N,1
      FACT=C(K,J)/C(J,J)
      C(K,J)=FACT
      DO 10 I=Z,N,1
      A=FACT*C(J,I)
      C(K,I)=A-C(K,I)
10    CONTINUE
60    CONTINUE
      J=J+1
      IF(J-N)40,50,50
50    DO 91 I=1,N,1
      IF(C(I,I).NE.0.0)GO TO 91
      WRITE(W,106)
      IFLAG=1
      GO TO 20
91    CONTINUE
```

```
      DO 601 I=1,N,1
      D(I,I)=1.0
      CALL PIVOTR(N,I,C,D,P,MJ)
      CALL BAKSUB(I,N,C,D,P,MJ)
 601  CONTINUE
 106  FORMAT('0','THE SYSTEM OF EQUATIONS IS SINGULAR')
      DO 603 I=1,N,1
      DO 603 K=1,N,1
 603  C(I,K)=D(I,
 20   RETURN
      END

COMPILATION:         NO DIAGNOSTICS.
```

```
PIVOT      ENTRY POI-T 000123

D: CODE(1) 000137; DATA 0) 000031; BLANK COMMON(2) 000000

FERENCES (BLOCK, NAME)

R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

043 112G        0001    000076 130G        0001    000054 20L        0001    000104 50L
004 INJP$       0000 I 000002 L            0000 R 000000 LARGE       0000 I 000001 M


          SUBROUTINE PIVOT(J,N,C,D,P,MJ)
          DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
          INTEGER P
          REAL LARGE
          LARGE=ABS(C(J,J))
          M=0
          P(J)=J
          DO 20 L=J,N,1
          IF(LARGE-ABS(C(L,J)))10,20,20
    10    LARGE=ABS(C(L,J))
          M=L
    20    CONTINUE
          IF(M)40,50,40
    40    P(J)=M
          DO 30 L=J,N,1
          HOLD=C(J,L)
          C(J,L)=C(M,L)
          C(M,L)=HOLD
    30    CONTINUE
    50    RETURN
          END

OMPILATION:          NO  DIAGNOSTICS.
```

BAKSUB ENTRY POINT 000137

): CODE(1) 000161; DATA 0) 000043; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

R3$

IGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
161 111G        0001   000102 115G       0000 R 000002 COUP       0000 I 000003 I
001 K           0000 I 000000 Y
```

```
      SUBROUTINE BAKSUB(J,N,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      INTEGER Y
      D(N,J)=D(N,J)/C(N,N)
      Y=N-1
      DO 70 K=1,Y,1
      COUP=0.0
      DO 80 I=1,K,1
      COUP=COUP+(D(N+1-I,J)*C(N-K,N+1-I))
   80 CONTINUE
      D(N-K,J)=(D(N-K,J)-COUP)/C(N-K,N-K)
   70 CONTINUE
      RETURN
      END
```

OMPILATION: NO DIAGNOSTICS.

PIVOTR ENTRY POI T 000115

D: CODE(1) 000137; -T-(0) 000037; BLANK COMMON(2) 000000

FERENCES (BLOCK, NAME)

R3$

IGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
026 107G      0001    000062 117G      0000 I 000005 I        0000   000007 INJP$
000 JP        0000 I 000004 JP1        0000 I 000003 M        0000 R 000002 RHOLD
```

```
      SUBROUTINE PIVOTR(N,K,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      JP=N-1
      DO 1 J=1,JP,1
      RHOLD=D(J,K)
      M=P(J)
      D(J,K)=D(M,K)
      D(M,K)=RHOLD
      JP1=J+1
      DO 2 I=JP1,N,1
      D(I,K)=D(J,K)*C(I,J)-D(I,K)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

OMPILATION: NO DIAGNOSTICS.

```
EADER     ENTRY POINT 000040

: CODE(1) 000042; DATA(0) 000011; BLANK COMMON(2) 000000

KS:

K1  000121
K2  001212
K3  000003
    000032
    000020
RS  000012


ERENCES (BLOCK, NAME)

EZ
S
S
S
S
3S


GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

02 201F      0000    000003 301F      0001    000004 5L       0004    000000 A
05 INJP$     0004    000536 J         0006    000000 JAC      0007    000000 JAST
07 JBLK      0003 I  000000 JBUF      0006    000002 JCC      0007    000002 JCMA
03 JDC       0007    000014 JDIV      0007 I  000004 JDOL     0006    000004 JEC
05 JFC       0003    000120 JFOUND    0006    000006 JGC      0007    000015 JGT
10 JIC       0006    000011 JJC       0006    000012 JKC      0007    000005 JLB
11 JLP       0007    000016 JLT       0006    000014 JMC      0007    000013 JMIN
16 JOC       0006    000017 JPC       0007    000003 JPER     0007    000012 JPLS
21 JRC       0007    000010 JRP       0006    000022 JSC      0007    000001 JSEM
24 JUC       0006    000025 JVC       0006    000026 JWC      0006    000027 JXC
31 JZC       0005    000001 NOJOI     0005    000002 NOMEN    0010    000000 NUM
00 R         0000 I  000001 W



          SUBROUTINE READER
          COMMON /BLOCK1/ JBUF(80),JFOUND
          COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
          COMMON /BLOCK3/ E,NOJOI,NOMEN
          COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
         1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
          COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
         1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
          COMMON /NUMERS/ NUM(10)
          INTEGER R,W
          R=5
          W=6
    C REAR AND PRINT THE INPUT STATEMENT
    5     READ(R,201)JBUF
          WRITE(W,301)JBUF
```

```
C  PEMOVE BLANKS FROM THE INPUT DATA
      CALL SQUEEZ
      IF(JBUF(1).EQ.JDOL)GO TO 5
201   FORMAT(80A1)
301   FORMAT('L',80A1)
      RETURN
      END
```

OMPILATION: NO DIAGNOSTICS.

```
76-13:33:51 (,0)

NTERP     ENTRY POINT 000424

: CODE(1) 000426; DATA(0) 000006; BLANK COMMON(2) 000000

KS:

K1  000121
K2  001212
K3  000003
K4  000005
    000032
    000020
RS  000012

ERENCES (BLOCK, NAME)

D1
TC
NC
ER
3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

15 9999L      0004     000000 A          0012 R 000000 DECOD1     0005 R 000000 E
02 INJP$      0006 I 000002 ISOLV        0004   000536 J          0007 I 000000 JAC
01 JBC        0010     000007 JBLK       0003 I 000000 JBUF       0007 I 000002 JCC
06 JCOL       0007 I 000003 JDC          0010   000014 JDIV       0010   000004 JDOL
17 JEQ        0007 I 000005 JFC          0003 I 000120 JFOUND     0007   000006 JGC
07 JHC        0007 I 000010 JIC          0007 I 000011 JJC        0007   000012 JKC
13 JLC        0010     000011 JLP        0010   000016 JLT        0007 I 000014 JMC
15 JNC        0007 I 000016 JOC          0007 I 000017 JPC        0010   000003 JPER
20 JQC        0007 I 000021 JRC          0010   000010 JRP        0007 I 000022 JSC
23 JTC        0007     000024 JUC        0007   000025 JVC        0007   000026 JWC
30 JYC        0007     000031 JZC        0000 I 000000 L          0006 I 000004 NEWPR
01 NOLOAD     0005     000002 NOMEN      0006 I 000000 NOREL      0011   000000 NUM


          SUBROUTINE INTERP
          COMMON /BLOCK1/ JBUF(80),JFOUND
          COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
          COMMON /BLOCK3/ E,NOJOI,NOMEN
          COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
          COMMON /LETP/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
         1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
          COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
         1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
          COMMON /NUMBRS/ NUM(10)
          JFOUND=0
    C PROBLEM STATEMENT
          IF(JBUF(1).EQ.JPC.AND.
         &   JBUF(2).EQ.JRC.AND.
         &   JBUF(3).EQ.JOC.AND.
```

```
     &    JBUF(4).EQ.JEC)JFOUND=1
      IF(JFOUND.EQ.1)NEWPRO=1
      IF(JFOUND.EQ.1)GO TO 9999
C MODULUS OF ELASTICITY
      IF(JBUF(1).EQ.JEC)JFOUND=1
      IF(JFOUND.EQ.1.AND.JBUF(2).EQ.JEQ)E=DECODI(3,L)
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT COORDINATES
      IF(JBUF(1).EQ.JJC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JIC.AND.
     &    JBUF(4).EQ.JNC.AND.
     &    JBUF(5).EQ.JTC.AND.
     &    JBUF(6).EQ.JCC)CALL JOINTC
      IF(JFOUND.EQ.1)GO TO 9999
C MEMBER INCIDENCES
      IF(JBUF(1).EQ.JMC.AND.
     &    JBUF(2).EQ.JFC.AND.
     &    JBUF(3).EQ.JMC.AND.
     &    JBUF(4).EQ.JPC.AND.
     &    JBUF(5).EQ.JEC.AND.
     &    JBUF(6).EQ.JRC.AND.
     &    JBUF(7).EQ.JIC)CALL MEMINC
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT AND MEMBER RELEACES
      IF(JBUF(1).EQ.JRC.AND.
     &    JBUF(2).EQ.JEC.AND.
     &    JBUF(3).EQ.JLC)JFOUND=1
      IF(JFOUND.EQ.1)NOREL=1
      IF(JFOUND.EQ.1)GO TO 9999
C LOADING CARDS
      IF(JBUF(1).EQ.JLC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JAC.AND.
     &    JBUF(4).EQ.JDC)JFOUND=1
      IF(JFOUND.EQ.1.AND.NOLOAD.EQ.1)GO TO 9999
      IF(JFOUND.EQ.1)CALL LOADER
      IF(JBUF(1).EQ.JSC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JLC)ISOLV=1
      IF(JFOUND.EQ.1)GO TO 9999
C END STATEMENT
      IF(JBUF(1).EQ.JFC.AND.
     &    JBUF(2).EQ.JIC.AND.
     &    JBUF(3).EQ.JNC)IEND=1
9999  RETURN
      END

OMPILATION:          NO  DIAGNOSTICS.
```

```
76-13:33:56 (,2)
```



```
QUEEZ     ENTRY POINT 000036

: CODE(1) 000043; DATA (0) 000011; BLANK COMMON(2) 000000

KS:

K1 000121
K2 001212
K3 000003
   000032
   000020
RS 000012

ERENCES (BLOCK, NAME)

3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

14 ICIL      0001   000003 113G     0004  000000 A      0005   000000 E
02 INJP$     0004   000536 J        0006  000000 JAC    0007 I 000000 JAST
07 JBLK      0003 I 000000 JBUF     0006  000002 JCC    0007   000002 JCMA
03 JDC       0007   000014 JDIV     0007  000004 JDOL   0006   000004 JEC
05 JFC       0003   000120 JFOUND   0006  000006 JGC    0007   000015 JGT
10 JIC       0006   000011 JJC      0006  000012 JKC    0007   000005 JLB
11 JLP       0007   000016 JLT      0006  000014 JMC    0007   000013 JMIN
16 JOC       0006   000017 JPC      0007  000003 JPER   0007   000012 JPLS
21 JRC       0007   000010 JRP      0006  000022 JSC    0007   000001 JSEM
24 JUC       0306   000025 JVC      0006  000026 JWC    0006   000027 JXC
31 JZC       0000 I 000000 K        0005  000001 NOJOI  0005   000002 NOMEN
30 P


      SUBROUTINE SQUEEZ
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBERS/ NUM(10)
      K=0
      DO 101 I=1,80,1
      IF(JBUF(I).EQ.JBLK)GO TO 101
      K=K+1
      JBUF(K)=JBUF(I)
  101 CONTINUE
      K=K+1
      JBUF(K)=JAST
      RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

JOINTC ENTRY POINT 000113

D: CODE(1) 000121; DATA(0) 000017; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000003
R 000032
R 000020
RS 000012

FERENCES (BLOCK, NAME)

DER
OL1
R35

IGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

002	10L	0001		000007	115G	0001		000077	20L	0001		000100	99L
000	DECOD1	0005		000000	E	0000	I	000001	I	0000	I	000000	IGOOF
536	J	0006		000000	JAC	0007	I	000000	JAST	0006		000001	JBC
000	JBUF	0006		000002	JCC	0007		000002	JCMA	0007		000006	JCOL
014	JDIV	0007		000004	JDOL	0006		000004	JEC	0007		000017	JEQ
120	JFOUND	0006		000006	JGC	0007		000015	JGT	0006		000007	JHC
011	JJC	0006		000012	JKC	0007		000005	JLB	0006		000013	JLC
016	JLT	0006		000014	JMC	0007		000013	JMIN	0006		000015	JNC
002	JOINT	0006		000017	JPC	0007		000003	JPER	0007		000012	JPLS
021	JRC	0007		000010	JRP	0006		000022	JSC	0007		000001	JSEM
024	JUC	0006		000025	JVC	0006		000026	JWC	0006		000027	JXC
031	JZC	0000	I	000004	J1	0000	I	000003	L	0005	I	000001	NOJOI
000	NUM	0004		001130	P								

```
      SUBROUTINE JOINTC
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      JFOUND=1
10    IGOOF=0
      CALL READER
      DO 5 I=1,10
5     IF(JBUF(1).EQ.NUM(I))IGOOF=1
      IF(IGOOF.NE.1)GO TO 99
      JOINT=DECOD1(1,L)
      IF(JBUF(L).EQ.JAST)GO TO 20
```

```
      JI=L+1
      A(JOINT,6)=DECODI(JI,L)
      IF(JBUF(L).EQ.JAST)GO TO 20
      JI=L+1
      A(JOINT,7)=DECODI(JI,L)
      L=L+1
      J(JOINT,1 =JBUF(L)
      NOJOI=NOJOI+1
      GO TO 10
   20 NOJOI=0
   99 RETURN
      END

OMPILATION:          NO  DIAGNOSTICS.
```

```
ODI        ENTRY POI.T 003157

: CODE(1) 000176; D-T-(3) 000025; BLANK COMMON(2) 000000

KS:

   000020
RS 000012
KI 000121


EPENCES (BLOCK, NAME)


3$


GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

43 121G       0001    000056 130G     0001    000075 20L       0001    000136 30L
06 ARIGHT     0000 R  000001 DECODI   0000 I  000007 I         0000    000013 INJP$
07 JBLK       0005 I  000000 JBUF     0003    000002 JCMA      0003    000006 JCOL
04 JDOL       0003    000017 JEQ      0005    000120 JFOUND    0003    000015 JGT
05 JLB        0003    000011 JLP      0003    000016 JLT       0003 I  000013 JMIN
12 JPLS       0003    000010 JRP      0003    000001 JSEM      0000 I  000004 JSIDE
0C NUM        0003 R  000000 SIGN


      FUNCTION DECODI(J,L)
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1             JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      COMMON /BLOCKI/ JBUF(80),JFOUND
      K=1
      JJ=J
      SIGN=1.0
      IF(JBUF(J).EQ.JMIN)SIGN=-1.0
      IF(JBUF(J).EQ.JMIN.OR.JBUF(J).EQ.JPLS)JJ=J+1
      JSIDE=0
      ALEFT=0.0
      ARIGHT=0.0
      DO 30 I=JJ,80,1
      IF(JBUF(I).EQ.JPER)JSIDE=1
      IF(JBUF(I).EQ.JPER)GO TO 30
      DO 15 L=1,10,1
      IF(JBUF(I).EQ.NUM(L))GO TO 20
   15 CONTINUE
      DECODI=SIGN*(ALEFT+ARIGHT)
      L=I
      RETURN
   20 IF(L.EQ.10)L=0
      IF(JSIDE.EQ.0)ALEFT=ALEFT*10.+L
      IF(JSIDE.EQ.1)ARIGHT=ARIGHT+FLOAT(L)/FLOAT(10 **K)
      IF(JSIDE.EQ.1)K=K+1
   30 CONTINUE
```

```
      END

OMPILATION:        '0  DIAGNOSTICS.
```

```
MEMINC     ENTRY POINT 000203

): CODE(1) 000214; DATA(0) 000035; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000003
R   000032
R   000020
BRS 000012

FERENCES (BLOCK, NAME)

DER
OD1
R
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

002 IOL       0001    000007 115G     0001   000165 98L      0001   000164 99L
007 AREA      0012 R  000000 DECOD1   0005 R 000000 E        0000 R 000012 FLENG
000 IGOOF     0000 I  000005 IJOINT   0000   000016 INJP$    0004 I 000536 J
000 JAST      0006    000001 JBC      0007   000007 JBLK     0003 I 000000 JBUF
002 JCMA      0007    000006 JCOL     0006   000003 JDC      0007   000014 JDIV
004 JEC       0007    000017 JEQ      0006   000005 JFC      0003 I 000120 JFOUN
015 JGT       0006    000007 JHC      0006   000010 JIC      0006   000011 JJC
012 JKC       0007    000005 JLB      0006   000013 JLC      0007   000011 JLP
014 JMC       0007    000013 JMIN     0006   000015 JNC      0006   000016 JOC
003 JPER      0007    000012 JPLS     0006   000020 JQC      0006   000021 JRC
022 JSC       0007    000001 JSEM     0006   000023 JTC      0006   000024 JUC
026 JWC       0006    000027 JXC      0006   000030 JYC      0006   000031 JZC
003 L         0000 I  000002 MEMBER   0005   000001 NOJOI    0000 I 000013 NOMEM
000 NUM       0004    001130 P        0000 R 000010 X        0000 R 000011 Y


      SUBROUTINE MEMINC
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      JFOUND=1
10    IGOOF=0
      CALL READER
      DO 5 I=1,10,1
5     IF(JBUF(1).EQ.NUM(I))IGOOF=1
      IF(IGOOF.NE.1)GO TO 98
```

```
      MEMBER=DECODI(1,L)
      JI=L+1
      IJOINT=DECODI(JI,L)
      JI=L+1
      JJOINT=DECODI(JI,L)
      JI=L+1
      AREA=DECO  (JI,L)
      NUMEM=NUMEM+1
      IF(IJOINT.EQ.0.OR.JJOINT.EQ.0.OR.MEMBER.EQ.0)GO TO 99
      IF(AREA.EQ.0.0)GO TO 99
      J(MEMBER,4)=IJOINT
      J(MEMBER,5)=JJOINT
      X=A(JJOINT,6)-A(IJOINT,6)
      Y=A(JJOINT,7)-A(IJOINT,7)
      FLENGT=((X**2.)+(Y**2.))**0.5
      IF(FLENGT.EQ.0.0)GO TO 99
      A(MEMBER,4)=X/FLENGT
      A(MEMBER,5)=Y/FLENGT
 2    A(MEMBER,3)=AREA*E/FLENGT
      GO TO 10
 99   NUMEM=0
 98   RETURN
      END

MPILATION:          NO  DIAGNOSTICS.
```

```
OADER     ENTRY POINT 000146

): CODE(1) 000152; DATA 0) 000020; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000003
CK4 000005
R   000032
R   000026
RRS 000012

FERENCES (BLOCK, NAME)

ER
DI
R3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

13 13L         0001    000003 15L      0001   000134 98L      0001    000135 99L
00 DECOD1      0005    000000 E        0006 R 000005 FORCE    0006    000003 IEND
L2 ISOLV       0004 I  000536 J        0007   000000 JAC      0010    000000 JAST
07 JBLK        0003 I  000000 JBUF     0007   000002 JCC      0010    000002 JCMA
03 JDC         0010    000014 JDIV     0010   000004 JDOL     0007    000004 JEC
05 JFC         0003    000120 JFOUND   0007   000006 JGC      0010    000015 JGT
10 JIC         0007 I  000011 JJC      0007   000012 JKC      0010    000005 JLB
11 JLP         0010    000016 JLT      0007   000014 JMC      0010    000013 JMIN
16 JOC         0000 I  000002 JOINT    0007   000017 JPC      0010    000003 JPER
20 JQC         0007    000021 JRC      0010   000010 JRP      0007    000022 JSC
23 JTC         0007    000024 JUC      0007   000025 JVC      0007    000026 JWC
30 JYC         0007    000031 JZC      0000 I 000001 J1       0000 I  000003 L
04 LOCJP1      0000 I  000006 LOCJP2   0006   000004 NEWPRO   0005    000001 NOJOI
02 NOMEN       0006    000000 NOREL    0011   000000 NUM      0004 R  001130 P
```

```
      SUBROUTINE LOADER
C THIS SUBROUTINE HANDLES LOADING CONDITIONS
C THE ROUTINE IS CALLED FROM INTERP AND TRTURNS THERE.
C THE LOADS ARE ENTERED INTO THE P VECTOR
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      NOLOAD=0
      J1=0
```

```
      LOC=0
15    CONTINUE
      CALL READER
      IF(JBUF(1).NE.JJC)GO TO 99
      NOLOAD=1
      JOINT=DECOD1(6,L)
      L=L+1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JYC)LOC=2
      IF(LOC.NE.1)GO TO 13
14    LOCJP1=J(JOINT,1)
      J1=L+6
      FORCE=DECOD1(J1,L)
      P(LOCJP1     )=FORCE
      GO TO 15
13    IF(LOC.NE.2)GO TO 98
16    LOCJP2=J(JOINT,2)
      J1=L+6
      FORCE=DECOD1(J1,L)
      P(LOCJP2     )=FORCE
      GO TO 15
98      NOLOAD=0
99    RETURN
      END

OMPILATION:          NO  DIAGNOSTICS.



13:34
ROG SIZE(I/D)=6512/8658


EL 7L-1
ON - TIME 1.688 SECONDS
```

Appendix C

Plane Frame Program Listing

```
774 K          0000 I 012004 K1         0000 I 012005 K2         0000 I 012006 K3
010 K5         0005 I 012011 K6         0000 I 011766 L          0000 I 000000 LOC
776 LOCCJK     0000 I 011770 LOK        0000 I 011771 LOK1       0000 I 011772 LOK3
030 MFL5       0000 I 012025 MFL1       0000 I 012026 MFL2       0000 I 012027 MFL3
033 MFL5       0000 I 012034 MFL6       0000 I 011761 MJ         0006 I 000004 NEWPR
001 NOLOAD     0005 I 000002 NOMEN      0000 I 012024 NOP        0005 I 000000 NOREL
773 N1         0004 R 001130 F          0000 I 011613 PP         0000 I 011757 R
052 SHEAR      0000 R 000001 SIN        0000 R 012036 SINA       0000 R 012040 SINE
003 TT         0000 I 011760 W
```

```
      DIMENSION S(50,50),D(50,50),PP(50)
      DIMENSION DISP(50)
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      DATA JAST /1H*/, JSEM/1H;/, JCMA /1H,/, JPER/1H./, JDOL/1H$/,
     1   JLB/1H#/, JCOL /1H:/, JBLK/1H /, JRP/1H)/, JLP/1H(/, JEQ /1H=/
      DATA JAC/1HA/, JBC/1HB/, JCC/1HC/, JDC/1HD/, JEC/1HE/, JFC/1HF/,
     1     JGC/1HG/, JHC/1HH/, JIC/1HI/, JJC/1HJ/, JKC/1HK/, JLC/1HL/,
     1     JMC/1HM/, JNC/1HN/, JOC/1HO/, JPC/1HP/, JQC/1HQ/, JRC/1HR/,
     1     JSC/1HS/, JTC/1HT/, JUC/1HU/, JVC/1HV/, JWC/1HW/, JXC/1HX/,
     1     JYC/1HY/, JZC/1HZ/
      DATA JPLS/1H+/, JMIN/1H-/, JDIV/1H//, JGT/1H>/, JLT/1H</
      DATA NUM(1) /1H1/, NUM(2) /1H2/, NUM(3)  /1H3/, NUM(4)/1H4/,
     1     NUM(5) /1H5/, NUM(6) /1H6/, NUM(7)  /1H7/, NUM(8)/1H8/,
     1     NUM(9) /1H9/, NUM(10) /1H0/
      INTEGER PP
      INTEGER R,W
      MJ=50
      W=6
      R=5
26    WRITE(W,130)
      WRITE(W,131)
      WRITE(W,132)
      WRITE(W,133)
      WRITE(W,132)
      WRITE(W,138)
      WRITE(W,134)
      WRITE(W,132)
      WRITE(W,135)
      WRITE(W,132)
      WRITE(W,136)
      WRITE(W,132)
      WRITE(W,131)
      WRITE(W,139)
      CALL READER
      CALL INTERP
55    CONTINUE
      NEWPRO=0
      E=0.0
      CALL READER
      CALL INTERP
      IF(E.EQ.0.0)GO TO 29
```

```
      NOJOI=0
      CALL READER
      CALL INTERP
      IF(NOJOI.EQ.0)GO TO 30
      NOMEN=0
      NOLOAD=1
      CALL INTERP
      IF(NOMEN.EQ.0)GO TO 31
      ICOUN1=NOJOI*3
      ICOUN2=1
      DO 3 I=1,NOJOI,1
      IF(J(I,1).NE.JSC)GO TO 8
5     J(I,1)=ICOUN1
      ICOUN1=ICOUN1-1
      J(I,2)=ICOUN1
      ICOUN1=ICOUN1-1
      J(I,3)=ICOUN1
      ICOUN1=ICOUN1-1
      GO TO 3
8     J(I,1)=ICOUN2
      ICOUN2=ICOUN2+1
      J(I,2)=ICOUN2
      ICOUN2=ICOUN2+1
      J(I,3)=ICOUN2
      ICOUN2=ICOUN2+1
3     CONTINUE
      ICOUN2=ICOUN2-1
      ICOUN1=ICOUN1+1
      NOREL=0
51    CALL INTERP
      IF(NOREL.EQ.0)GO TO 35
      LOC=0
      IF(JBUF(8).NE.JJC)GO TO 47
      JOINT=DECOD1(13,L)
      L=L+1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JYC)LOC=2
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JZC)LOC=3
      IF(LOC.EQ.0)GO TO 32
      ICOUN1=ICOUN1+1
      ICOUN2=ICOUN2+1
      IHOLD=J(JOINT,LOC)
       DO 37 LOK=1,NOJOI,1
      IF(J(LOK,1).NE.ICOUN2.AND.J(LOK,2).NE.ICOUN2
     6.AND.J(LOK,3).NE.ICOUN2)GO TO 37
      IF(J(LOK,1).EQ.ICOUN2)LOK1=1
      IF(J(LOK,2).EQ.ICOUN2)LOK1=2
      IF(J(LOK,3).EQ.ICOUN2)LOK1=3
      LOK3=LOK
      GO TO 38
37    CONTINUE
38    CONTINUE
      J(JOINT,LOC)=ICOUN2
      J(LOK3,LOK1)=IHOLD
      GO TO 36
47    IF(JBUF(8).NE.JMC)GO TO 32
      JOINT=DECOD1(14,L)
      L=L+1
      IF(JBUF(L).EQ.JIC)J(JOINT,4)=-J(JOINT,4)
      IF(JBUF(L).EQ.JTC)J(JOINT,5)=-J(JOINT,5)
      IF(JBUF(L).EQ.JBC)A(JOINT,2)=0.
36    CALL READER
```

```
      NOREL=0
      GO TO 51
35    NOREL=1
      N1=NOJOI*3
      DO 10 I=1,N1,1
      DO 10 K=1,N1,1
10    B(I,K)=0.0
      DO 12 I=1,NOMEN,1
      LOCCJJ=J(I,4)
      LOCCJK=J(I,5)
      HH=4.
      GG=12.
      RR=6.
      SS=1.
      TT=1.
      IF(LOCCJJ.GT.0)GO TO 45
      IF(LOCCJK.LE.0)GO TO 32
      HH=3.
      GG=3.
      RR=3.
      SS=0.
      LOCCJJ=ABS(LOCCJJ)
45    IF(LOCCJK.GT.0)GO TO 46
      HH=3.
      GG=3.
      RR=3.
      TT=0.
      LOCCJK=ABS(LOCCJK)
46    CONTINUE
      K1=J(LOCCJJ,1)
      K2=J(LOCCJJ,2)
      K3=J(LOCCJJ,3)
      K4=J(LOCCJK,1)
      K5=J(LOCCJK,2)
      K6=J(LOCCJK,3)
      SIN=A(I,5)
      COS=A(I,4)
      AEL=A(I,3)
      EIL=A(I,2)
      FLENGT=A(I,1)
      AA=COS*COS*AEL+SIN*SIN*GG *EIL/FLENGT/FLENGT
      BB=SIN*SIN*AEL+COS*COS*GG *EIL/FLENGT/FLENGT
      CC=HH*EIL
      DD=SIN*COS*AEL-SIN*COS*GG *EIL/FLENGT/FLENGT
      EE=SIN*RR*EIL/FLENGT
      FF=-COS*RR*EIL/FLENGT
      B(K1,K1)=B(K1,K1)+AA
      B(K1,K2)=B(K1,K2)+DD
      B(K1,K3)=B(K1,K3)+EE*SS
      B(K1,K4)=B(K1,K4)-AA
      B(K1,K5)=B(K1,K5)-DD
      B(K1,K6)=B(K1,K6)+EE*TT
      B(K2,K1)=B(K2,K1)+DD
      B(K2,K2)=B(K2,K2)+BB
      B(K2,K3)=B(K2,K3)+FF*SS
      B(K2,K4)=B(K2,K4)-DD
      B(K2,K5)=B(K2,K5)-BB
      B(K2,K6)=B(K2,K6)+FF*TT
      B(K3,K1)=B(K3,K1)+EE*SS
      B(K3,K2)=B(K3,K2)+FF*SS
      B(K3,K3)=B(K3,K3)+CC*SS
      B(K3,K4)=B(K3,K4)-EE*SS
```

```
      B(K3,K5)=B(K3,K5)-FF*SS
      B(K3,K6)=B(K3,K6)+CC/2.*SS*TT
      B(K4,K1)=B(K4,K1)-AA
      B(K4,K2)=B(K4,K2)-DD
      B(K4,K3)=B(K4,K3)-EE*SS
      B(K4,K4)=B(K4,K4)+AA
      B(K4,K5)=B(K4,K5)+DD
      B(K4,K6)=B(K4,K6)-EE*TT
      B(K5,K1)=B(K5,K1)-DD
      B(K5,K2)=B(K5,K2)-BB
      B(K5,K3)=B(K5,K3)-FF*SS
      B(K5,K4)=B(K5,K4)+DD
      B(K5,K5)=B(K5,K5)+BB
      B(K5,K6)=B(K5,K6)-FF*TT
      B(K6,K1)=B(K6,K1)+EE*TT
      B(K6,K2)=B(K6,K2)+FF*TT
      B(K6,K3)=B(K6,K3)+CC/2.*TT*SS
      B(K6,K4)=B(K6,K4)-EE*TT
      B(K6,K5)=B(K6,K5)-FF*TT
      B(K6,K6)=B(K6,K6)+CC*TT
12    CONTINUE
      CALL  INVERT(ICOUN2,IFLAG,W,B,D,PP,MJ)
      IF(IFLAG.EQ.1)GO TO 29
24    NOLOAD=0
      ISOLV=0
      NOP=3*NOJOI
      DO 17 I=1,NOP,1
17    P(I)=0.0
      CALL  INTERP
      IF(IEND.EQ.1)GO TO 27
      IF(NEWPRO.EQ.1)GO TO 55
      IF(NOLOAD.EQ.0)GO TO 33
      IF(JFOUND.EQ.0)GO TO 25
      IF(ISOLV.NE.1)GO TO 29
      DO 18 I=1,ICOUN2,1
      DISP(I)=0.0
      DO 18 K=1,ICOUN2,1
18    DISP(I)=DISP(I)+B(I,K)*P(K)
      DO 19 I=ICOUN1,NOP,1
      P(I)=-P(I)
      DISP(I)=0.0
      DO 19 K=1,ICOUN2,1
19    P(I)=P(I)+B(I,K)*DISP(K)
      WRITE(W,126)
      WRITE(W,120)
      WRITE(W,121)
      DO 21 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
      MFL3=J(L,3)
21    WRITE(W,129)L,        DISP(MFL1),DISP(MFL2),DISP(MFL3)
      DO 39 L=1,ICOUN2,1
39    P(L)=0.0
      WRITE(W,127)
      WRITE(W,123)
      WRITE(W,121)
      DO 22 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
      MFL3=J(L,3)
      IF(MFL1.LT.ICOUN1.AND.MFL2.LT.ICOUN1.AND.MFL3.LT.ICOUN1)GO TO 22
      WRITE(W,129)L,        P(MFL1),P(MFL2),P(MFL3)
```

```
22    CONTINUE
      WRITE(W,123)
      WRITE(W,124)
      WRITE(W,125)
      DO 23 L=1,NOMEM,1
      MFLS=ABS(J(L,4))
      MFLE=ABS(J(L,5))
      MFL1=J(MFLS,1)
      MFL2=J(MFLS,2)
      MFL3=J(MFLS,3)
      MFL4=J(MFLE,1)
      MFL5=J(MFLE,2)
      MFL6=J(MFLE,3)
      SIN=A(L,5)
      COS=A(L,4)
      AEL=A(L,3)
      EIL=A(L,2)
      FLENGT=A(L,1)
      COSA= COS*AEL
      SINA=SIN*AEL
      COSE=COS*6.*EIL/FLENGT
      SINE=SIN*6.*EIL/FLENGT
      DISP1=DISP(MFL1)
      DISP2=DISP(MFL2)
      DISP3=DISP(MFL3)
      DISP4=DISP(MFL4)
      DISP5=DISP(MFL5)
      DISP6=DISP(MFL6)
      AXIAL=-COSA*DISP1-SINA*DISP2+COSA*DISP4+SINA*DISP5
      BENDE=SINE*DISP1-COSE*DISP2-SINE*DISP4+COSE*DISP5
      IF(J(L,4).LT.0)GO TO 48
      IF(J(L,5).LT.0)GO TO 49
      BENDS=BENDE+4.*EIL*DISP3+2.*EIL*DISP6
      BENDE=BENDE+2.*EIL*DISP3+4.*EIL*DISP6
      GO TO 50
48    BENDE=BENDE/2.+3*EIL*DISP6
      BENDS=0.
      GO TO 50
49    BENDS=BENDE/2.+3.*EIL*DISP3
         BENDE=0.
50    SHEAR=(BENDE+BENDS)/FLENGT
      WRITE(W,122)L,J(L,4),AXIAL,SHEAR,BENDS
23    WRITE(W,122)L,J(L,5),AXIAL,SHEAR,BENDE
      WRITE(W,139)
      CALL READER
      GO TO 24
29    WRITE(W,142)
      GO TO 25
30    WRITE(W,143)
      GO TO 25
31    WRITE(W,144)
      GO TO 25
32    WRITE(W,145)
      GO TO 25
33    WRITE(W,146)
25    IF(IEND.EQ.1)GO TO 27
      CALL READER
      CALL INTERP
      IF(NEWPRO.EQ.1)GO TO 55
      GO TO 25
120   FORMAT(' ',2X,'JOINT         ', 8X,'TRANSLATION',12X,'ROTATION')
121   FORMAT(' ',20X,'X',13X,'Y',13X,'Z'/)
```

```
122   FORMAT(' ',I6,I7,3F14.5)
123   FORMAT(' ',2X,'JOINT        ',11X,'FORCE',15X,'MOMENT')
124   FORMAT(' ',2X,'MEMBER JOINT',11X,'FORCE',15X,'MOMENT')
125   FORMAT(' ',13X,'AXIAL',9X,'SHEAR Y',8X,'BENDING Z'/)
126   FORMAT(///'1','JOINT DISPLACEMENTS'//)
127   FORMAT(///'1','SUPPORT REACTIONS'//)
128   FORMAT(///'1','MEMBER FORCES'//)
129   FORMAT(' ',I6,F21.5,2F14.5)
130   FORMAT('1',////////////////)
131   FORMAT(' ',10X,'************************************************************
     &*****')
132   FORMAT(' ',10X,'*',52X,'*')
133   FORMAT(' ',10X,'*    ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM
     &   *')
134   FORMAT(' ',10X,'*',20X,'RELEASE 1.0',21X,'*')
135   FORMAT(' ',10X,'*',15X,'UNIVERSITY OF HOUSTON',16X,'*')
136   FORMAT(' ',10X,'*',10X,'DEPARTMENT OF CIVIL ENGINEERING',11X,'*')
137   FORMAT(' ',10X,'*',6X,40A1,6X,'*')
138   FORMAT(' ',10X,'*',20X,'PLANE FRAME',21X,'*')
139   FORMAT('1')
141   FORMAT(' ','ERROR IN HEADER CARD')
142   FORMAT(' ','INSUFFICIENT DATA OR INPUT ERROR')
143   FORMAT(' ','ERROR IN JOINT COORDINATES')
144   FORMAT(' ','ERROR IN MEMBER INCIDENCES')
145   FORMAT(' ','ERROR IN JOINT RELEASES')
146   FORMAT(' ','ERROR IN LOADING')
27    STOP
      END

OMPILATION:        NO  DIAGNOSTICS.
```

```
75-14:04:24 (,0)

NVERT     ENTRY POINT 000334

: CODE(1) 000436; DATA(0) 000047; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

T
TR
UB
$
$
3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

07 106F      0001    000027 110G      0001    000030 113G      0001    000140 133G
00 155G      0001    000220 170G      0001    000314 20L       0001    000301 201G
72 34L       0001    000040 40L       0001    000213 91L       0000 R 000006 A
05 I         0000    000020 INJP$     0000 I 000003 J          0000 I 000001 K
00 Z


       SUBROUTINE INVERT(N,IFLAG,W,C,D,P,MJ)
       DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
       INTEGER P
       INTEGER Z,W
       IFLAG=0
       DO 5 K=1,N,1
       DO 5 L=1,N,1
 5     D(K,L)=0.0
       J=1
 40    Z=J+1
       CALL PIVOT(J,N,C,D,P,MJ)
       IF(C(J,J))34,30,34
 30    WRITE(W,105)
       IFLAG=1
       GO TO 20
 34    DO 60 K=Z,N,1
       FACT=C(K,J)/C(J,J)
       C(K,J)=FACT
       DO 10 I=Z,N,1
       A=FACT*C(J,I)
       C(K,I)=A-C(K,I)
 10    CONTINUE
 60    CONTINUE
       J=J+1
       IF(J-N)40,50,50
 50    DO 91 I=1,N,1
       IF(C(I,I).NE.0.0)GO TO 91
       WRITE(W,106)
       IFLAG=1
       GO TO 20
 91    CONTINUE
```

```
      DO 601 I=1,N,1
      D(I,I)=1.0
      CALL PIVCTR(N,I,C,D,P,MJ)
      CALL BAKSUB(I,N,C,D,P,MJ)
 601  CONTINUE
 106  FORMAT('0','THE SYSTEM OF EQUATIONS IS SINGULAR')
      DO 603 I=1,N,1
      DO 603 K=1,N,1
 603  C(I,K)=D(I,K)
 20   RETURN
      END

MPILATION:          NO  DIAGNOSTICS.
```

```
IVOT       ENTRY POINT 000123

: CODE(1) 000137; DATA(0) 000031; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

43 112G        0001   000076 130G       0001   000054 20L        0001   000104 50L
04 INJP$       0000 I 000002 L          0000 R 000000 LARGE      0000 I 000001 M


         SUBROUTINE PIVOT(J,N,C,D,P,MJ)
         DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
         INTEGER P
         REAL LARGE
         LARGE=ABS(C(J,J))
         M=0
         P(J)=J
         DO 20 L=J,N,1
         IF(LARGE-ABS(C(L,J)))10,20,20
   10    LARGE=ABS(C(L,J))
         M=L
   20    CONTINUE
         IF(M)40,50,40
   40    P(J)=M
         DO 30 L=J,N,1
         HOLD=C(J,L)
         C(J,L)=C(M,L)
         C(M,L)=HOLD
   30    CONTINUE
   50    RETURN
         END

MPILATION:         NO  DIAGNOSTICS.
```

```
AKSUB     ENTRY POINT 000137

: CODE(1) 000161; DATA(0) 000043; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

3S

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

51 111G       0001   000102 115G       0000 R 000002 COUP       0000 I 000003 I
01 K          0000 I 000000 Y


        SUBROUTINE BAKSUB(J,N,C,D,P,MJ)
        DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
        INTEGER P
        INTEGER Y
        D(N,J)=D(N,J)/C(N,N)
        Y=N-1
        DO 70 K=1,Y,1
        COUP=0.0
        DO 80 I=1,K,1
        COUP=COUP+(D(N+1-I,J)*C(N-K,N+1-I))
  80    CONTINUE
        D(N-K,J)=(D(N-K,J)-COUP)/C(N-K,N-K)
  70    CONTINUE
        RETURN
        END

OMPILATION:          NO  DIAGNOSTICS.
```

```
IVOTR     ENTRY POINT 000115

: CODE(1) 000137; DATA(0) 000037; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

26 107G       0001   000062 117G       0000 I 000005 I          0000   000007 INJP$
00 JP         0000 I 000004 JP1        0000 I 000003 M          0000 R 000002 RHOLD


      SUBROUTINE PIVOTR(N,K,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      JP=N-1
      DO 1 J=1,JP,1
      RHOLD=D(J,K)
      M=P(J)
      D(J,K)=D(M,K)
      D(M,K)=RHOLD
      JP1=J+1
      DO 2 I=JP1,N,1
      D(I,K)=D(J,K)*C(I,J)-D(I,K)
 2    CONTINUE
 1    CONTINUE
      RETURN
      END

MPILATION:        NO  DIAGNOSTICS.
```

```
IADER     ENTRY POINT 000040

: CODE(1) 000042; DATA(0) 000011; BLANK COMMON(2) 000000

(S:

(1 000121
(2 001212
(3 000003
   000032
   000020
RS 000012

RENCES (BLOCK, NAME)

EZ
5
5
5
5
35

NMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

02 201F       0000   000003 301F     0001   000004 5L      0004   000000 A
05 INJP$      0004   000536 J        0005   000000 JAC     0007   000000 JAST
07 JBLK       0003 I 000000 JBUF     0006   000002 JCC     0007   000002 JCMA
03 JDC        0007   000014 JDIV     0007 I 000004 JDOL    0006   000004 JEC
05 JFC        0003   000120 JFOUND   0006   000006 JGC     0007   000015 JGT
10 JIC        0006   000011 JJC      0006   000012 JKC     0007   000005 JLB
11 JLP        0007   000016 JLT      0006   000014 JMC     0007   000013 JMIN
16 JOC        0006   000017 JPC      0007   000003 JPER    0007   000012 JPLS
21 JRC        0007   000010 JRP      0005   000022 JSC     0007   000001 JSEM
24 JUC        0006   000025 JVC      0006   000026 JWC     0006   000027 JXC
31 JZC        0005   000001 NOJOI    0005   000002 NOMEN   0010   000000 NUM
0C R          0000 I 000001 W

      SUBROUTINE READER
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /LETR/  JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/  JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      INTEGER R,W
      R=5
      W=6
C REAR AND PRINT THE INPUT STATEMENT
5     READ(R,201)JBUF
      WRITE(W,301)JBUF
```

```
C  REMOVE BLANKS FROM THE INPUT DATA
      CALL SQUEEZ
      IF(JBUF(1).EQ.JDCL)GO TO 5
201   FORMAT(80A1)
301   FORMAT('C',80A1)
      RETURN
      END
```

OMPILATION: NO DIAGNOSTICS.

```
NTERP     ENTRY POINT 000427

: CODE(1) 000431; DATA(0) 000006; BLANK COMMON(2) 000000

KS:

K1 000121
K2 001212
K3 000003
K4 000005
   000032
   000020
RS 000012

ERENCES (BLOCK, NAME)

D1
TC
NC
ER
3S

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

74 5L        0001   000420 9993L      0004   000000 A         0012 R 000000 DECOD
03 IEND      0000   000002 INJP$      000E I 000002 ISOLV     0004   000530 J
00 JAST      0007 I 000001 JBC        0010   000007 JBLK      0003 I 000000 JBUF
02 JCMA      0010   000006 JCOL       0007 I 000003 JDC       0010   000014 JDIV
04 JEC       0010 I 000017 JEQ        0007 I 000005 JFC       0003 I 000120 JFOUN
15 JGT       0007   000007 JHC        0007 I 000010 JIC       0007 I 000011 JJC
05 JLB       0007 I 000013 JLC        0010   000011 JLP       0010   000016 JLT
13 JMIN      0007 I 000015 JNC        0007 I 000016 JOC       0007 I 000017 JPC
12 JPLS      0007   000020 JQC        0007 I 000021 JRC       0010   000010 JRP
01 JSEM      0007 I 000023 JTC        0007   000024 JUC       0007   000025 JVC
27 JXC       0007   000030 JYC        0007   000031 JZC       0000 I 000000 L
01 NOJOI     0006 I 000001 NOLOAD     0005   000002 NOMEN     0006 I 000000 NOREL
30 P


      SUBROUTINE INTERP
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),F(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/  JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1               JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/  JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1               JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      JFOUND=0
C PROBLEM STATEMENT
      IF(JBUF(1).EQ.JPC.AND.
     8   JBUF(2).EQ.JRC.AND.
```

```
     &    JBUF(3).EQ.JCC.AND.
     &    JBUF(4).EQ.JBC)JFOUND=1
      IF(JFOUND.EQ.1)NEWPRO=1
      IF(JFOUND.EQ.1)GO TO 9999
C MODULUS OF ELASTICITY
      IF(JBUF(1).EQ.JEC)JFOUND=1
      IF(JFOUND.EQ.1.AND.JBUF(2).EQ.JEG)E=DECOD1(3,L)
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT COORDINATES
      IF(JBUF(1).EQ.JJC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JIC.AND.
     &    JBUF(4).EQ.JNC.AND.
     &    JBUF(5).EQ.JTC.AND.
     &    JBUF(6).EQ.JCC)CALL JOINTC
      IF(JFOUND.EQ.1)GO TO 9999
C MEMBER INCIDENCES
      IF(JBUF(1).EQ.JMC.AND.
     &    JBUF(2).EQ.JEC.AND.
     &    JBUF(3).EQ.JMC.AND.
     &    JBUF(4).EQ.JBC.AND.
     &    JBUF(5).EQ.JEC.AND.
     &    JBUF(6).EQ.JRC.AND.
     &    JBUF(7).EQ.JIC)CALL MEMINC
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT AND MEMBER RELEACES
      IF(JBUF(1).EQ.JRC.AND.
     &    JBUF(2).EQ.JEC.AND.
     &    JBUF(3).EQ.JLC)JFOUND=1
      IF(JFOUND.EQ.1)NOREL=1
      IF(JFOUND.EQ.1)GO TO 9999
C LOADING CARDS
      IF(NOREL.EQ.0)GO TO 5
      IF(JBUF(1).EQ.JLC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JAC.AND.
     &    JBUF(4).EQ.JDC)JFOUND=1
      IF(JFOUND.EQ.1.AND.NOLOAD.EQ.1)GO TO 9999
      IF(JFOUND.EQ.1)CALL LOADER
      IF(JBUF(1).EQ.JSC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JLC)ISOLV=1
      IF(JFOUND.EQ.1)GO TO 9999
C END STATEMENT
5     IF(JBUF(1).EQ.JEC.AND.
     &    JBUF(2).EQ.JIC.AND.
     &    JBUF(3).EQ.JNC)IEND=1
9999  RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
SQUEEZ     ENTRY POINT 000036

): CODE(1) 000043; DATA(0) 000011; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000003
R   000032
R   000020
BRS 000012

FERENCES (BLOCK, NAME)

R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

014 101L       0001    000003 113G      0004    000000 A        0005    000000 E
002 INJP$      0004    000536 J         0006    000000 JAC      0007 I  000000 JAST
007 JBLK       0003 I  000000 JBUF      0006    000002 JCC      0007    000002 JCMA
003 JDC        0007    000014 JDIV      0007    000004 JDOL     0006    000004 JEC
005 JFC        0003    000120 JFOUND    0006    000006 JGC      0007    000015 JGT
010 JIC        0006    000011 JJC       0006    000012 JKC      0007    000005 JLB
011 JLP        0007    000016 JLT       0006    000014 JMC      0007    000013 JMIN
016 JOC        0006    000017 JPC       0007    000003 JPER     0007    000012 JPLS
021 JRC        0007    000010 JRP       0006    000022 JSC      0007    000001 JSEM
024 JUC        0006    000025 JVC       0006    000026 JWC      0006    000027 JXC
031 JZC        0000 I  000000 K         0005    000001 NOJOI    0005    000002 NOMEN
130 P


      SUBROUTINE SQUEEZ
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JPC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      K=0
      DO 101 I=1,80,1
      IF(JBUF(I).EQ.JBLK)GO TO 101
      K=K+1
      JBUF(K)=JBUF(I)
  101 CONTINUE
      K=K+1
      JBUF(K)=JAST
      RETURN
      END
```

OMPILATION: NO DIAGNOSTICS.

```
OINTC    ENTRY POINT 000113

: CODE(1) 000121; DATA(0) 000017; BLANK COMMON(2) 000000

KS:

K1 000121
K2 001212
K3 000003
   000032
   000020
RS 000012

ERENCES (BLOCK, NAME)

ER
01
35

ONMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

02 10L        0001     000007 115G      0001     000077 20L      0001     000100 99L
00 DECO01     0005     000000 E         0000 I 000001 I          0000 I 000000 IGOOF
36 J          0006     000000 JAC       0007 I 000000 JAST       0006     000001 JBC
00 JBUF       0006     000002 JCC       0007     000002 JCMA     0007     000006 JCOL
14 JDIV       0007     000004 JDOL      0006     000004 JEC      0007     000017 JEQ
20 JFOUND     0006     000005 JGC       0007     000015 JGT      0006     000007 JHC
11 JJC        0006     000012 JKC       0007     000005 JLB      0006     000013 JLC
16 JLT        0006     000014 JMC       0007     000013 JMIN     0006     000015 JNC
02 JOINT      0006     000017 JPC       0007     000003 JPER     0007     000012 JPLS
21 JRC        0007     000010 JRP       0006     000022 JSC      0007     000001 JSEM
24 JUC        0006     000025 JVC       0006     000026 JWC      0006     000027 JXC
31 JZC        0000 I 000004 J1          0000 I 000003 L          0005 I 000001 NOJOI
00 NUM        0004     001130 F


        SUBROUTINE JOINTC
        COMMON /BLOCK1/ JBUF(80),JFOUND
        COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
        COMMON /BLOCK3/ E,NOJOI,NOMEN
        COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
       1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
        COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
       1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
        COMMON /NUMBRS/ NUM(10)
        JFOUND=1
   10   IGOOF=0
        CALL READER
        DO 5 I=1,10
    5   IF(JBUF(1).EQ.NUM(I))IGOOF=1
        IF(IGOOF.NE.1)GO TO 99
        JOINT=DECO01(1,L)
        IF(JBUF(L).EQ.JAST)GO TO 20
```

```
      J1=L+1
      A(JOINT,6)=DECOD1(J1,L)
      IF(JBUF(L).EQ.JAST)GO TO 20
      J1=L+1
      A(JOINT,7)=DECOD1(J1,L)
      L=L+1
      J(JOINT,1)=JBUF(L)
      NOJOI=NOJOI+1
      GO TO 10
 20   NOJOI=0
 99   RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
COD1        ENTRY POINT 000157

: CODE(1) 000176; DATA(0) 000025; BLANK COMMON(2) 000000

CKS:

    000020
RS 000012
K1 000121

ERENCES (BLOCK, NAME)

3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

43 121G        0001    000056 130G       0001    000075 20L        0001    000136 30L
06 ARIGHT      0000 R  000001 DECOD1     0000 I  000007 I          0000    000013 INJP$
07 JBLK        0005 I  000000 JBUF       0003    000002 JCMA       0003    000006 JCOL
04 JDOL        0003    000017 JEQ        0005    000120 JFOUND     0003    000015 JGT
05 JLB         0003    000011 JLP        0003    000016 JLT        0003 I  000013 JMIN
12 JPLS        0003    000010 JRP        0003    000001 JSEM       0000 I  000004 JSIDE
00 NUM         0000 R  000000 SIGN

      FUNCTION DECOD1(J,L)
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      COMMON /BLOCK1/ JBUF(80),JFOUND
      K=1
      JJ=J
      SIGN=1.0
      IF(JBUF(J).EQ.JMIN)SIGN=-1.0
      IF(JBUF(J).EQ.JMIN.OR.JBUF(J).EQ.JPLS)JJ=J+1
      JSIDE=0
      ALEFT=0.0
      ARIGHT=0.0
      DO 30 I=JJ,80,1
      IF(JBUF(I).EQ.JPER)JSIDE=1
      IF(JBUF(I).EQ.JPER)GO TO 30
      DO 15 L=1,10,1
      IF(JBUF(I).EQ.NUM(L))GO TO 20
   15 CONTINUE
      DECOD1=SIGN*(ALEFT+ARIGHT)
      L=I
      RETURN
   20 IF(L.EQ.10)L=0
      IF(JSIDE.EQ.0)ALEFT=ALEFT*10.+L
      IF(JSIDE.EQ.1)ARIGHT=ARIGHT+FLOAT(L)/FLOAT(10 **K)
      IF(JSIDE.EQ.1)K=K+1
   30 CONTINUE
```

```
      END

COMPILATION:          NO  DIAGNOSTICS.
```

```
MEMINC     ENTRY POINT 000226

D: CODE(1) 000237; DATA(0) 000036; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000003
R    000032
AR   000020
BRS  000012

EFERENCES (BLOCK, NAME)

DER
COD1
RR
RR3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

002 10L        0001   000007 116G      0001   000210 98L      0001   000207 99L
010 AREA       0012 R 000000 DECOD1    0005 R 000000 E        0000 R 000013 FLENG
001 IGOOF      0000 I 000006 IJOINT    0000 R 000000 INERT    0000   000017 INJP$
000 JAC        0007   000000 JAST      0006   000001 JBC      0007   000007 JBLK
002 JCC        0007   000002 JCMA      0007   000006 JCOL     0006   000003 JDC
004 JDOL       0006   000004 JEC       0007   000017 JEQ      0006   000005 JFC
006 JGC        0007   000015 JGT       0006   000007 JHC      0006   000010 JIC
007 JJOINT     0006   000012 JKC       0007   000005 JLB      0006   000013 JLC
016 JLT        0006   000014 JMC       0007   000013 JMIN     0006   000015 JNC
017 JPC        0007   000003 JPER      0007   000012 JPLS     0006   000020 JQC
010 JRP        0006   000022 JSC       0007   000001 JSEM     0006   000023 JTC
025 JVC        0006   000026 JWC       0006   000027 JXC      0006   000030 JYC
005 J1         0000 I 000004 L         0000 I 000003 MEMBER   0005   000001 NOJOI
002 NOMEN      0010 I 000000 NUM       0004   001130 P        0000 R 000011 X



       SUBROUTINE MEMINC
       COMMON /BLOCK1/ JBUF(80),JFOUND
       COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
       COMMON /BLOCK3/ E,NOJOI,NOMEN
       COMMON /LETR/  JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
      1               JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
       COMMON /CHAR/  JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
      1               JPLS,JMIN,JDIV,JGT,JLT,JEQ
       COMMON /NUMBRS/ NUM(10)
       REAL INERT
       JFOUND=1
 10    IGOOF=0
       CALL READER
       DO 5 I=1,10,1
 5     IF(JBUF(1).EQ.NUM(I))IGOOF=1
```

```
      IF(ISOOF.NE.1)GO TO 98
      MEMBER=DECODI(1,L)
      J1=L+1
      IJOINT=DECODI(J1,L)
      J1=L+1
      JJOINT=DECODI(J1,L)
      J1=L+1
      AREA=DECODI(J1,L)
      J1=L+1
      INERT=DECODI(J1,L)
      NOMEN=NOMEN+1
      IF(IJOINT.EQ.0.OR.JJOINT.EQ.0.OR.MEMBER.EQ.0)GO TO 99
      IF(AREA.EQ.0.0.OR.INERT.EQ.0.0)GO TO 99
      J(MEMBER,4)=IJOINT
      J(MEMBER,5)=JJOINT
      X=A(JJOINT,6)-A(IJOINT,6)
      Y=A(JJOINT,7)-A(IJOINT,7)
      FLENGT=((X**2.)+(Y**2.))**0.5
      IF(FLENGT.EQ.0.0)GO TO 99
      A(MEMBER,4)=X/FLENGT
      A(MEMBER,5)=Y/FLENGT
      A(MEMBER,1)= FLENGT
      A(MEMBER,2)=INERT*E/FLENGT
 2    A(MEMBER,3)=AREA*E/FLENGT
      GO TO 10
 99   NOMEM=0
 98   RETURN
      END

COMPILATION:        NO  DIAGNOSTICS.
```

```
LOADER     ENTRY POINT CCCCC6

D: CODE(1) 000212; DATA(C) CCCC21; BLANK COMMON(2) 000000

CKS:

CK1 CC0121
CK2 001212
CK3 000003
CK4 000005
R   000032
R   000020
BRS 000012

FERENCES (BLOCK, NAME)

DER
DD1
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

132 13L       0001    000003 15L       0001    000153 40L       0001    000174 98L
000 A         0013 R 000000 DECOD1     0005    000000 E         0000 R 000005 FORCE
011 INJP$     0006    000002 ISCLV     0004 I 000538 J          0007    00000C JAC
001 JBC       0010    000007 JBLK      0003 I 000000 JBUF       0007    000002 JCC
005 JCOL      0007    000003 JDC       0010    000014 JDIV      0010    000004 JDOL
017 JEQ       0007 I 000005 JFC        0003    000120 JFOUND    0007    000006 JGC
007 JHC       0007    000010 JIC       0007 I 000011 JJC        0007    000012 JKC
013 JLC       0010    000011 JLP       0010    000016 JLT       0007 I 000014 JMC
015 JNC       0007    000016 JOC       0000 I 000002 JOINT      0007    000017 JPC
012 JPLS      0007    000020 JQC       0007    000021 JRC       0010    000010 JRP
001 JSEM      0007    000023 JTC       0007    000024 JUC       0007    000025 JVC
027 JXC       0007 I 000030 JYC        0007 I 000031 JZC        0000 I 000001 J1
000 LOC       0000 I 000004 LOCJP1     0000 I 000006 LOCJP2     0000 I 000007 LOCJP
001 NOJOI     0006 I 000001 NOLOAD     0005    000002 NOMEN     0006    000000 NOREL
130 P


      SUBROUTINE LOADER
C THIS SUBROUTINE HANDLES LOADING CONDITIONS
C THE ROUTINE IS CALLED FROM INTERP AND TRTURNS THERE.
C THE LOADS ARE ENTERED INTO THE P VECTOR
      COMMON /BLOCK1/  JBUF(80),JFOUND
      COMMON /BLOCK2/  A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/  E,NOJOI,NOMEN
      COMMON /BLOCK4/  NOREL,NOLOAD,ISCLV,IEND,NEWPRO
      COMMON /LETR/  JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1               JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/  JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1               JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      NOLOAD=0
```

```
      J1=0
      LOC=0
15    CONTINUE
      CALL READER
      IF(JBUF(I).NE.JJC)GO TO 99
      NOLOAD=I
      JOINT=DECOD1(6,L)
      L=L+1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JYC)LOC=2
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JZC)LOC=3
      IF(LOC.NE.1)GO TO 13
14    LOCJP1=J(JOINT,1)
      J1=L+6
      FORCE=DECOD1(J1,L)
      P(LOCJP1     )=FORCE
      GO TO 15
13    IF(LOC.NE.2)GO TO 40
16    LOCJP2=J(JOINT,2)
      J1=L+6
      FORCE=DECOD1(J1,L)
      P(LOCJP2     )=FORCE
      GO TO 15
40    IF(LOC.NE.3)GO TO 98
41    LOCJP3=J(JOINT,3)
      J1=L+7
      FORCE=DECOD1(J1,L)
      P(LOCJP3)=FORCE
      GO TO 15
98      NOLOAD=0
99    RETURN
      END

COMPILATION:          NO  DIAGNOSTICS.



14:04
PROG SIZE(I/D)=7046/8745


EL 70-1
ON - TIME 1.700 SECONDS
```

Appendix D

Plane Grid Program Listing

```
2002 K4        0000 I 012003 K5        0000 I 012004 K6       0000 I 011766 L
1775 LOCCJJ    0000 I 011776 LOCCJK    0000 I 011770 LOK      0000 I 011771 LOK1
2027 MFLE      0000 I 012026 MFLS      0000 I 012023 MFL1     0000 I 012024 MFL2
2030 MFL4      0000 I 012031 MFL5      0000 I 012032 MFL6     0000 I 011761 MJ
0001 NOJOI     0006 I 000001 NOLOAD    0005 I 000002 NOMEN    0000 I 012022 NOP
0000 NUM       0000 I 011773 N1        0004 R 001130 P        0000 I 011613 PP
2046 SHEAR     0000 R 000001 SIN       0000 R 012034 SINA     0000 I 011760 W
```

```
      DIMENSION R(50,50),D(50,50),PP(50)
      DIMENSION DISP(50)
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN,G
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      DATA JAST /1H*/, JSEM/1H;/, JCMA /1H,/, JPER/1H./, JDOL/1H$/,
     1   JLB/1H#/, JCOL /1H:/, JBLK/1H /, JRP/1H)/, JLP/1H(/, JEQ /1H=/
      DATA JAC/1HA/, JBC/1HB/, JCC/1HC/, JDC/1HD/, JEC/1HE/, JFC/1HF/,
     1     JGC/1HG/, JHC/1HH/, JIC/1HI/, JJC/1HJ/, JKC/1HK/, JLC/1HL/,
     1     JMC/1HM/, JNC/1HN/, JOC/1HO/, JPC/1HP/, JQC/1HQ/, JRC/1HR/,
     1     JSC/1HS/, JTC/1HT/, JUC/1HU/, JVC/1HV/, JWC/1HW/, JXC/1HX/,
     1     JYC/1HY/, JZC/1HZ/
      DATA JPLS/1H+/, JMIN/1H-/, JDIV/1H//, JGT/1H>/, JLT/1H</
      DATA NUM(1) /1H1/, NUM(2) /1H2/, NUM(3)  /1H3/, NUM(4)/1H4/,
     1     NUM(5) /1H5/, NUM(6) /1H6/, NUM(7)  /1H7/, NUM(8)/1H8/,
     1     NUM(9) /1H9/, NUM(10) /1H0/
      INTEGER PP
      INTEGER R,W
      MJ=50
      W=6
      R=5
 26   WRITE(W,130)
      WRITE(W,131)
      WRITE(W,132)
      WRITE(W,133)
      WRITE(W,132)
      WRITE(W,138)
      WRITE(W,134)
      WRITE(W,132)
      WRITE(W,135)
      WRITE(W,132)
      WRITE(W,136)
      WRITE(W,132)
      WRITE(W,131)
      WRITE(W,139)
      CALL READER
      CALL INTERP
 55   CONTINUE
      NEWPRO=3
      E=0.0
      G=0.0
      CALL READER
      CALL INTERP
      CALL READER
      CALL INTERP
```

```
      IF(E.EQ.0.0)GO TO 29
      IF(G.EQ.0.0)GO TO 29
      NOJOI=0
      CALL READER
      CALL INTERP
      IF(NOJOI.EQ.0)GO TO 30
      NOMEN=0
      NULOAD=1
      CALL INTERP
      IF(NOMEN.EQ.0)GO TO 31
      ICOUN1=NOJOI*3
      ICOUN2=1
      DO 3 I=1,NOJOI,1
      IF(J(I,1).NE.JSC)GO TO 8
5     J(I,1)=ICOUN1
      ICOUN1=ICOUN1-1
      J(I,2)=ICOUN1
      ICOUN1=ICOUN1-1
      J(I,3)=ICOUN1
      ICOUN1=ICOUN1-1
      GO TO 3
8     J(I,1)=ICOUN2
      ICOUN2=ICOUN2+1
      J(I,2)=ICOUN2
      ICOUN2=ICOUN2+1
      J(I,3)=ICOUN2
      ICOUN2=ICOUN2+1
3     CONTINUE
      ICOUN2=ICOUN2-1
      ICOUN1=ICOUN1+1
      NOREL=0
51    CALL INTERP
      IF(NOREL.EQ.0)GO TO 35
      LOC=0
      IF(JBUF(8).NE.JJC)GO TO 47
      JOINT=DECOD1(13,L)
      L=L+1
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JYC)LOC=2
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JZC)LOC=3
      IF(LOC.EQ.0)GO TO 32
      ICOUN1=ICOUN1+1
      ICOUN2=ICOUN2+1
      IHOLD=J(JOINT,LOC)
       DO 37 LOK=1,NOJOI,1
      IF(J(LOK,1).NE.ICOUN2.AND.J(LOK,2).NE.ICOUN2
     6.AND.J(LOK,3).NE.ICOUN2)GO TO 37
      IF(J(LOK,1).EQ.ICOUN2)LOK1=1
      IF(J(LOK,2).EQ.ICOUN2)LOK1=2
      IF(J(LOK,3).EQ.ICOUN2)LOK1=3
      LOK3=LOK
      GO TO 38
37    CONTINUE
38    CONTINUE
      J(JOINT,LOC)=ICOUN2
      J(LOK3,LOK1)=IHOLD
      GO TO 36
47    IF(JBUF(8).NE.JMC)GO TO 32
      JOINT=DECOD1(14,L)
      L=L+1
      IF(JBUF(L).EQ.JTC.AND.JBUF(L+1).EQ.JOC)A(JOINT,3)=0.
36    CALL READER
```

```
      NOREL=1
      GO TO 51
35    NOREL=1
      NI=NOJOI*3
      DO 10 I=1,NI,1
      DO 10 K=1,NI,1
10    B(I,K)=0.0
      DO 12 I=1,NOMEM,1
      LOCCJJ=J(I,4)
      LOCCJK=J(I,5)
      K1=J(LOCCJJ,1)
      K2=J(LOCCJJ,2)
      K3=J(LOCCJJ,3)
      K4=J(LOCCJK,1)
      K5=J(LOCCJK,2)
      K6=J(LOCCJK,3)
      SIN=A(I,5)
      COS=A(I,4)
      AEL=A(I,3)
      EIL=A(I,2)
      FLENGT=A(I,1)
      A1=COS*COS*AEL+SIN*SIN*4.*EIL
       A2=SIN*COS*AEL-SIN*COS*4*EIL
      A3=SIN*6.*EIL/FLENGT
      A4=SIN*SIN*AEL+COS*COS*4.*EIL
      A5=COS*6.*EIL/FLENGT
      A6=12.*EIL/FLENGT/FLENGT
      A7=-COS*COS*AEL+SIN*SIN*2.*EIL
      A8=-SIN*COS*AEL-SIN*COS*2.*EIL
      A9=-SIN*SIN*AEL+COS*COS*2.*EIL
      B(K1,K1)=B(K1,K1)+A1
      B(K1,K2)=B(K1,K2)+A2
      B(K1,K3)=B(K1,K3)+A3
      B(K1,K4)=B(K1,K4)+A7
      B(K1,K5)=B(K1,K5)+A8
      B(K1,K6)=B(K1,K6)-A3
      B(K2,K1)=B(K2,K1)+A2
      B(K2,K2)=B(K2,K2)+A4
      B(K2,K3)=B(K2,K3)-A5
      B(K2,K4)=B(K2,K4)+A8
      B(K2,K5)=B(K2,K5)+A9
      B(K2,K6)=B(K2,K6)+A5
      B(K3,K1)=B(K3,K1)+A3
      B(K3,K2)=B(K3,K2)-A5
      B(K3,K3)=B(K3,K3)+A6
      B(K3,K4)=B(K3,K4)+A3
      B(K3,K5)=B(K3,K5)-A5
      B(K3,K6)=B(K3,K6)-A6
      B(K4,K1)=B(K4,K1)+A7
      B(K4,K2)=B(K4,K2)+A8
      B(K4,K3)=B(K4,K3)+A3
      B(K4,K4)=B(K4,K4)+A1
      B(K4,K5)=B(K4,K5)+A2
      B(K4,K6)=B(K4,K6)-A3
      B(K5,K1)=B(K5,K1)+A8
      B(K5,K2)=B(K5,K2)+A9
      B(K5,K3)=B(K5,K3)-A5
      B(K5,K4)=B(K5,K4)+A2
      B(K5,K5)=B(K5,K5)+A4
      B(K5,K6)=B(K5,K6)+A5
      B(K6,K1)=B(K6,K1)-A3
      B(K6,K2)=B(K6,K2)+A5
```

```
      B(K6,K3)=B(K6,K3)-A6
      B(K6,K4)=B(K6,K4)-A3
      B(K6,K5)=B(K6,K5)+A5
      B(K6,K6)=B(K6,K6)+A6
12    CONTINUE
      CALL INVERT(ICOUN2,IFLAG,W,B,D,PP,MJ)
      IF(IFLAG.EQ.1)GO TO 29
24    NOLOAD=0
      ISOLV=0
      NOP=3*NOJOI
      DO 17 I=1,NOP,1
17    P(I)=0.0
      CALL INTERP
      IF(NEWPRO.EQ.1)GO TO 55
      IF(IEND.EQ.1)GO TO 27
      IF(NOLOAD.EQ.0)GO TO 33
      IF(JFOUND.EQ.0)GO TO 25
      IF(ISOLV.NE.1)GO TO 29
      DO 18 I=1,ICOUN2,1
      DISP(I)=0.0
      DO 18 K=1,ICOUN2,1
18    DISP(I)=DISP(I)+B(I,K)*P(K)
      DO 19 I=ICOUN1,NOP,1
      P(I)=-P(I)
      DISP(I)=0.0
      DO 19 K=1,ICOUN2,1
19    P(I)=P(I)+B(I,K)*DISP(K)
      WRITE(W,126)
      WRITE(W,120)
      WRITE(W,121)
      DO 21 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
      MFL3=J(L,3)
21    WRITE(W,129)L,          DISP(MFL3),DISP(MFL1),DISP(MFL2)
      DO 39 L=1,ICOUN2,1
39    P(L)=0.0
      WRITE(W,127)
      WRITE(W,123)
      WRITE(W,121)
      DO 22 L=1,NOJOI,1
      MFL1=J(L,1)
      MFL2=J(L,2)
      MFL3=J(L,3)
      IF(MFL1.LT.ICOUN1.AND.MFL2.LT.ICOUN1.AND.MFL3.LT.ICOUN1)GO TO 22
      WRITE(W,129)L,          P(MFL3),P(MFL1),P(MFL2)
22    CONTINUE
      WRITE(W,128)
      WRITE(W,124)
      WRITE(W,125)
      DO 23 L=1,NOMEN,1
      MFLS=ABS(J(L,4))
      MFLE=ABS(J(L,5))
      MFL1=J(MFLS,1)
      MFL2=J(MFLS,2)
      MFL3=J(MFLS,3)
      MFL4=J(MFLE,1)
      MFL5=J(MFLE,2)
      MFL6=J(MFLE,3)
      SIN=A(L,5)
      COS=A(L,4)
      AEL=A(L,3)
```

```
      EIL=A(L,2)
      FLENGT=A(L,1)
      COSA= COS*AEL
      SINA=SIN*AEL
      DISP1=DISP(MFL1)
      DISP2=DISP(MFL2)
      DISP3=DISP(MFL3)
      DISP4=DISP(MFL4)
      DISP5=DISP(MFL5)
      DISP6=DISP(MFL6)
      AXIAL=-COSA*DISP1-SINA*DISP2+COSA*DISP4+SINA*DISP5
      BENDS=-SIN*EIL*4.*DISP1+COS*EIL*4.*DISP2-6.*EIL*DISP3/FLENGT
     &-SIN*2.*EIL*DISP4+COS*2.*EIL*DISP5+6.*EIL*DISP6/FLENGT
      BENDE=-SIN*EIL*2.*DISP1+COS*EIL*2.*DISP2-6.*EIL*DISP3/FLENGT
     &-SIN*4.*EIL*DISP4+COS*4.*EIL*DISP5+6.*EIL*DISP6/FLENGT
50    SHEAR=(BENDE+BENDS)/FLENGT
      SHEAR=-SHEAR
      WRITE(W,122)L,J(L,4), SHEAR,AXIAL,BENDS
      SHEAR=-SHEAR
23    WRITE(W,122)L,J(L,5),SHEAR,AXIAL,BENDE
      WRITE(W,139)
      CALL READER
      GO TO 24
29    WRITE(W,142)
      GO TO 25
30    WRITE(W,143)
      GO TO 25
31    WRITE(W,144)
      GO TO 25
32    WRITE(W,145)
      GO TO 25
33    WRITE(W,146)
25    IF(IEND.EQ.1)GO TO 27
      CALL READER
      CALL INTERP
      IF(NEWPRO.EQ.1)GO TO 55
      GO TO 25
120   FORMAT(' ',2X,'JOINT         ',2X,'TRANSLATION',11X,'ROTATION')
121   FORMAT(' ',20X,'Z',13X,'X',13X,'Y'/)
122   FORMAT(' ',I6,I7,3F14.5)
123   FORMAT(' ',2X,'JOINT         ',4X,'FORCE',17X,'MOMENT')
124   FORMAT(' ',2X,'MEMBER JOINT',4X,'FORCE',17X,'MOMENT')
125   FORMAT(' ',18X,'SHEAR',8X,'TORSIONAL',5X,'BENDING Z'/)
126   FORMAT(///'1','JOINT DISPLACEMENTS'//)
127   FORMAT(///'1','SUPPORT REACTIONS'//)
128   FORMAT(///'1','MEMBER FORCES'//)
129   FORMAT(' ',I6,F21.5,2F14.5)
130   FORMAT('1',///////////////)
131   FORMAT(' ',10X,'**************************************************
     o****')
132   FORMAT(' ',10X,'*',52X,'*')
133   FORMAT(' ',10X,'*    ACADEMIC STRUCTURAL ANALYSIS COMPUTER SYSTEM
     6    *')
134   FORMAT(' ',10X,'*',20X,'RELEASE 1.0',21X,'*')
135   FORMAT(' ',10X,'*',15X,'UNIVERSITY OF HOUSTON',16X,'*')
136   FORMAT(' ',10X,'*',10X,'DEPARTMENT OF CIVIL ENGINEERING',11X,'*')
137   FORMAT(' ',10X,'*',6X,40A1,6X,'*')
138   FORMAT(' ',10X,'*',20X,'PLANE  GRID',21X,'*')
139   FORMAT('1')
141   FORMAT(' ','ERROR IN HEADER CARD')
142   FORMAT(' ','INSUFFICIENT DATA OR INPUT ERROR')
143   FORMAT(' ','ERROR IN JOINT COORDINATES')
```

```
  144     FORMAT(' ','ERROR IN MEMBER INCIDENCES')
  145     FORMAT(' ','ERROR IN JOINT RELEASES')
  146     FORMAT(' ','ERROR IN LOADING')
  27      STOP
          END
```

COMPILATION: NO DIAGNOSTICS.

```
INVERT     ENTRY POINT 000334

D: CODE(1) 000436; DATA(0) 000047; BLANK COMMON(2) 000000

FERENCES (BLOCK, NAME)

OT
OTR
SUB
US
2$
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

007 106F       0001    000027 110G       0001    000030 113G       0001    000140 133G
200 155G       0001    000220 170G       0001    000314 20L        0001    000301 201G
072 34L        0001    000040 40L        0001    000213 91L        0000 R 000006 A
005 I          0000    000020 INJP$      0000 I 000003 J           0000 I 000001 K
000 Z
```

```
      SUBROUTINE INVERT(N,IFLAG,W,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      INTEGER Z,W
      IFLAG=0
      DO 5 K=1,N,1
      DO 5 L=1,N,1
5     D(K,L)=0.0
      J=1
40    Z=J+1
      CALL PIVOT(J,N,C,D,P,MJ)
      IF(C(J,J))34,30,34
30    WRITE(W,106)
      IFLAG=1
      GO TO 20
34    DO 60 K=Z,N,1
      FACT=C(K,J)/C(J,J)
      C(K,J)=FACT
      DO 10 I=Z,N,1
      A=FACT*C(J,I)
      C(K,I)=A-C(K,I)
10    CONTINUE
60    CONTINUE
      J=J+1
      IF(J-N)40,50,50
50    DO 91 I=1,N,1
      IF(C(I,I).NE.0.0)GO TO 91
      WRITE(W,106)
      IFLAG=1
      GO TO 20
91    CONTINUE
```

```
      DO 601 I=1,N,1
      D(I,I)=1.0
      CALL PIVOTR(N,I,C,D,P,MJ)
      CALL BAKSUB(I,N,C,D,P,MJ)
601   CONTINUE
106   FORMAT('0','THE SYSTEM OF EQUATIONS IS SINGULAR')
      DO 603 I=1,N,1
      DO 603 K=1,N,1
603   C(I,K)=D(I,K)
20    RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
IVOT      ENTRY POINT 000123

: CODE(1) 000137; DATA(0) 000031; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

13$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

043 112G       0001   000076 130G       0001   000054 20L        0001   000104 50L
004 INJP$      0000 I 000002 L          0000 R 000000 LARGE      0000 I 000001 M


        SUBROUTINE PIVOT(J,N,C,D,P,MJ)
        DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
        INTEGER P
        REAL LARGE
        LARGE=ABS(C(J,J))
        M=0
        P(J)=J
        DO 20 L=J,N,1
        IF(LARGE-ABS(C(L,J)))10,20,20
   10   LARGE=ABS(C(L,J))
        M=L
   20   CONTINUE
        IF(M)40,50,40
   40   P(J)=M
        DO 30 L=J,N,1
        HOLD=C(J,L)
        C(J,L)=C(M,L)
        C(M,L)=HOLD
   30   CONTINUE
   50   RETURN
        END

COMPILATION:         NO  DIAGNOSTICS.
```

BAKSUB ENTRY POINT 000137

ED: CODE(1) 000161; DATA(0) 000043; BLANK COMMON(2) 000000

EFERENCES (BLOCK, NAME)

RR3S

SIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0061 111G        0001   000102 115G        0000 R 000002 COUP        0000 I 000003 I
0001 K           0000 I 000000 Y
```

```
      SUBROUTINE BAKSUB(J,N,C,D,P,MJ)
      DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
      INTEGER P
      INTEGER Y
      D(N,J)=D(N,J)/C(N,N)
      Y=N-1
      DO 70 K=1,Y,1
      COUP=0.0
      DO 80 I=1,K,1
      COUP=COUP+(D(N+1-I,J)*C(N-K,N+1-I))
80    CONTINUE
      D(N-K,J)=(D(N-K,J)-COUP)/C(N-K,N-K)
70    CONTINUE
      RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
IVOTR     ENTRY POINT 000115

: CODE(1) 000137; DATA(0) 000037; BLANK COMMON(2) 000000

ERENCES (BLOCK, NAME)

3$

GNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

26 107G        0001    000062 117G       0000 I 000005 I          0000   000007 INJP
00 JP          0000 I 000004 JP1         0000 I 000003 M          0000 R 000002 RHOL


         SUBROUTINE PIVOTR(N,K,C,D,P,MJ)
         DIMENSION C(MJ,MJ),D(MJ,MJ),P(MJ)
         INTEGER P
         JP=N-1
         DO 1 J=1,JP,1
         RHOLD=D(J,K)
         M=P(J)
         D(J,K)=D(M,K)
         D(M,K)=RHOLD
         JP1=J+1
         DO 2 I=JP1,N,1
         D(I,K)=D(J,K)*C(I,J)-D(I,K)
    2    CONTINUE
    1    CONTINUE
         RETURN
         END

COMPILATION:          NO  DIAGNOSTICS.
```

```
READER     ENTRY POINT 000040

D: CODE(1) 000042; DATA(0) 000011; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000004
R   000032
R   000020
BRS 000012

FERENCES (BLOCK, NAME)

EEZ
U$
3$
2$
U$
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

002 2U1F      0000   000003 3U1F     0001   000004 5L       0004   000000 A
003 G         0000   000005 INJP$    0004   000536 J        0006   000000 JAC
001 JBC       0007   000007 JBLK     0003 I 000000 JBUF     0006   000002 JCC
006 JCOL      0006   000003 JDC      0007   000014 JDIV     0007 I 000004 JDOL
017 JEQ       0006   000005 JFC      0003   000120 JFOUND   0006   000006 JGC
007 JHC       0006   000010 JIC      0006   000011 JJC      0006   000012 JKC
013 JLC       0007   000011 JLP      0007   000016 JLT      0006   000014 JMC
015 JNC       0006   000016 JOC      0006   000017 JPC      0007   000003 JPER
020 JQC       0006   000021 JRC      0007   000010 JRP      0006   000022 JSC
023 JTC       0006   000024 JUC      0006   000025 JVC      0006   000026 JWC
030 JYC       0006   000031 JZC      0005   000001 NOJOI    0005   000002 NOMEN
130 P         0000 I 000000 R        0000 I 000001 W
```

```
      SUBROUTINE READER
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN,G
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      INTEGER R,W
      R=5
      W=6
C REAR AND PRINT THE INPUT STATEMENT
5     READ(R,201)JBUF
      WRITE(W,301)JBUF
```

```
C  REMOVE BLANKS FROM THE INPUT DATA
      CALL SQUEEZ
      IF(JBUF(1).EQ.JDOL)GO TO 5
201   FORMAT(80A1)
301   FORMAT('0',80A1)
      RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
 INTERP     ENTRY POINT 000460

D: CODE(1) 000462; DATA(0) 000006; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000004
CK4 000005
R    000032
R    000020
BRS 000012

FERENCES (BLOCK, NAME)

OD1
NTC
INC
DER
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0425 5L        0001    000451 9999L     0004    000000 A        0012 R 000000 DECOD
0003 G         0006 I  000003 IEND      0000    000002 INJP$    0006 I 000002 ISOLV
0000 JAC       0010    000000 JAST      0007 I  000001 JBC      0010   000007 JBLK
0002 JCC       0010    000002 JCMA      0010    000006 JCOL     0007 I 000003 JDC
0004 JDOL      0007 I  000004 JEC       0010 I  000017 JEQ      0007 I 000005 JFC
0006 JGC       0010    000015 JGT       0007    000007 JHC      0007 I 000010 JIC
0012 JKC       0010    000005 JLB       0007 I  000013 JLC      0010   000011 JLP
0014 JMC       0010    000013 JMIN      0007 I  000015 JNC      0007 I 000016 JOC
0003 JPER      0010    000012 JPLS      0007    000020 JQC      0007 I 000021 JRC
0022 JSC       0010    000001 JSEM      0007 I  000023 JTC      0007   000024 JUC
0026 JWC       0007    000027 JXC       0007    000030 JYC      0007   000031 JZC
0004 NEWPRO    0005    000001 NOJOI     0006 I  000001 NOLOAD   0005   000002 NOMEN
0000 NUM       0004    001130 P


      SUBROUTINE INTERP
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN,G
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      JFOUND=0
C PROBLEM STATEMENT
      IF(JBUF(1).EQ.JPC.AND.
     &   JBUF(2).EQ.JRC.AND.
```

```
     &    JBUF(3).EQ.JOC.AND.
     &    JBUF(4).EQ.JBC)JFOUND=1
      IF(JFOUND.EQ.1)NEWPRO=1
      IF(JFOUND.EQ.1)GO TO 9999
C MODULUS OF ELASTICITY
      IF(JBUF(1).EQ.JEC)JFOUND=1
      IF(JFOUND.EQ.1.AND.JBUF(2).EQ.JEQ)E=DECOD1(3,L)
      IF(JFOUND.EQ.1)GO TO 9999
C SHEAR MODULUS
      IF(JBUF(1).EQ.JGC)JFOUND=1
      IF(JBUF(1).EQ.JGC.AND.JBUF(2).EQ.JEQ)G=DECOD1(3,L)
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT COORDINATES
      IF(JBUF(1).EQ.JJC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JIC.AND.
     &    JBUF(4).EQ.JNC.AND.
     &    JBUF(5).EQ.JTC.AND.
     &    JBUF(6).EQ.JCC)CALL JOINTC
      IF(JFOUND.EQ.1)GO TO 9999
C MEMBER INCIDENCES
      IF(JBUF(1).EQ.JMC.AND.
     &    JBUF(2).EQ.JEC.AND.
     &    JBUF(3).EQ.JMC.AND.
     &    JBUF(4).EQ.JBC.AND.
     &    JBUF(5).EQ.JEC.AND.
     &    JBUF(6).EQ.JRC.AND.
     &    JBUF(7).EQ.JIC)CALL MEMINC
      IF(JFOUND.EQ.1)GO TO 9999
C JOINT AND MEMBER RELEACES
      IF(JBUF(1).EQ.JRC.AND.
     &    JBUF(2).EQ.JEC.AND.
     &    JBUF(3).EQ.JLC)JFOUND=1
      IF(JFOUND.EQ.1)NOREL=1
      IF(JFOUND.EQ.1)GO TO 9999
C LOADING CARDS
      IF(NOREL.EQ.0)GO TO 5
      IF(JBUF(1).EQ.JLC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JAC.AND.
     &    JBUF(4).EQ.JDC)JFOUND=1
      IF(JFOUND.EQ.1.AND.NOLOAD.EQ.1)GO TO 9999
      IF(JFOUND.EQ.1)CALL LOADER
      IF(JBUF(1).EQ.JSC.AND.
     &    JBUF(2).EQ.JOC.AND.
     &    JBUF(3).EQ.JLC)ISOLV=1
      IF(JFOUND.EQ.1)GO TO 9999
C END STATEMENT
5     IF(JBUF(1).EQ.JFC.AND.
     &    JBUF(2).EQ.JIC.AND.
     &    JBUF(3).EQ.JNC)IEND=1
9999  RETURN
      END

COMPILATION:          NO  DIAGNOSTICS.
```

```
SQUEEZ     ENTRY POINT 000036

D: CODE(1) 000043; DATA(0) 000011; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000004
R   000032
R   000020
BRS 000012


FERENCES (BLOCK, NAME)

R3$


IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0014 101L     0001    000003 113G     0004    000000 A         0005   000000 E
0001 I        0000    000002 INJP$    0004    000536 J         0006   000000 JAC
0001 JBC      0007 I  000007 JBLK     0003 I  000000 JBUF      0006   000002 JCC
0006 JCOL     0006    000003 JDC      0007    000014 JDIV      0007   000004 JDOL
0017 JEQ      0006    000005 JFC      0003    000120 JFOUND    0006   000006 JGC
0007 JHC      0006    000010 JIC      0006    000011 JJC       0006   000012 JKC
0013 JLC      0007    000011 JLP      0007    000016 JLT       0006   000014 JMC
0015 JNC      0006    000016 JOC      0006    000017 JPC       0007   000003 JPER
0020 JQC      0006    000021 JRC      0007    000010 JRP       0006   000022 JSC
0023 JTC      0006    000024 JUC      0006    000025 JVC       0006   000026 JWC
0030 JYC      0006    000031 JZC      0000 I  000000 K         0005   000001 NOJOI
0000 NUM      0004    001130 P
```

```
      SUBROUTINE SQUEEZ
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN,G
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      K=0
      DO 101 I=1,80,1
      IF(JBUF(I).EQ.JBLK)GO TO 101
      K=K+1
      JBUF(K)=JBUF(I)
  101 CONTINUE
      K=K+1
      JBUF(K)=JAST
      RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
 JOINTC      ENTRY POINT 000113

ID: CODE(1) 000121; DATA(0) 000017; BLANK COMMON(2) 000000

OCKS:

OCK1 000121
OCK2 001212
OCK3 000004
TR   000032
AR   000020
MBRS 000012

EFERENCES (BLOCK, NAME)

ADER
COD1
RR3$

SIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0002 10L        0001    000007 115G      0001    000077 20L       0001    000100 99L
0000 DECOD1     0005    000000 E         0005    000003 G         0000 I  000001 I
0006 INJP$      0004 I  000536 J         0006    000000 JAC       0007 I  000000 JAST
0007 JBLK       0003 I  000000 JBUF      0006    000002 JCC       0007    000002 JCMA
0003 JDC        0007    000014 JDIV      0007    000004 JDOL      0006    000004 JEC
0005 JFC        0003 I  000120 JFOUND    0006    000006 JGC       0007    000015 JGT
0010 JIC        0006    000011 JJC       0006    000012 JKC       0007    000005 JLB
0011 JLP        0007    000016 JLT       0006    000014 JMC       0007    000013 JMIN
0016 JOC        0000 I  000002 JOINT     0006    000017 JPC       0007    000003 JPER
0020 JQC        0006    000021 JRC       0007    000010 JRP       0006    000022 JSC
0023 JTC        0006    000024 JUC       0006    000025 JVC       0006    000026 JWC
0030 JYC        0006    000031 JZC       0000 I  000004 J1        0000 I  000003 L
0002 NOMEN      0010 I  000000 NUM       0004    001130 P

..

      SUBROUTINE JOINTC
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NOMEN,G
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      JFOUND=1
 10   IGOOF=0
      CALL READER
      DO 5 I=1,10
 5    IF(JBUF(1).EQ.NUM(I))IGOOF=1
      IF(IGOOF.NE.1)GO TO 99
      JOINT=DECOD1(1,L)
      IF(JBUF(L).EQ.JAST)GO TO 20
```

```
      J1=L+1
      A(JOINT,6)=DECOD1(J1,L)
      IF(JBUF(L).EQ.JAST)GO TO 20
      J1=L+1
      A(JOINT,7)=DECOD1(J1,L)
      L=L+1
      J(JOINT,1)=JBUF(L)
      NOJOI=NOJOI+1
      GO TO 10
 20   NOJOI=0
 99   RETURN
      END
```

COMPILATION: NO DIAGNOSTICS.

```
ECODI       ENTRY POINT 000157

ED: CODE(1) 000176; DATA(0) 000025; BLANK COMMON(2) 000000

OCKS:

AR    000020
MBRS 000012
OCK1 000121

EFERENCES (BLOCK, NAME)

II
RR3S

SIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0043 121G       0001    000056 130G       0001    000075 20L       0001    000136 30L
0006 ARIGHT     0000 R 000001 DECODI     0000 I 000007 I         0000    000013 INJP
0007 JBLK       0005 I 000000 JBUF       0003    000002 JCMA      0003    000006 JCOL
0004 JDOL       0003    000017 JEQ        0005    000120 JFOUND    0003    000015 JGT
0005 JLB        0003    000011 JLP        0003    000016 JLT       0003 I 000013 JMIN
0012 JPLS       0003    000010 JRP        0003    000001 JSEM      0000 I 000004 JSID
0000 NUM        0000 R 000000 SIGN


      FUNCTION DECODI(J,L)
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      COMMON /BLOCK1/ JBUF(80),JFOUND
      K=1
      JJ=J
      SIGN=1.0
      IF(JBUF(J).EQ.JMIN)SIGN=-1.0
      IF(JBUF(J).EQ.JMIN.OR.JBUF(J).EQ.JPLS)JJ=J+1
      JSIDE=0
      ALEFT=0.0
      ARIGHT=0.0
      DO 30 I=JJ,80,1
      IF(JBUF(I).EQ.JPER)JSIDE=1
      IF(JBUF(I).EQ.JPER)GO TO 30
      DO 15 L=1,10,1
      IF(JBUF(I).EQ.NUM(L))GO TO 20
 15   CONTINUE
      DECODI=SIGN*(ALEFT+ARIGHT)
      L=I
      RETURN
 20   IF(L.EQ.10)L=0
      IF(JSIDE.EQ.0)ALEFT=ALEFT*10.+L
      IF(JSIDE.EQ.1)ARIGHT=ARIGHT+FLOAT(L)/FLOAT(10 **K)
      IF(JSIDE.EQ.1)K=K+1
 30   CONTINUE
```

```
      END

COMPILATION:          NO  DIAGNOSTICS.
```

```
 MEMINC     ENTRY POINT 000236


ED: CODE(1) 000247; DATA(0) 000037; BLANK COMMON(2) 000000

OCKS:

OCK1 000121
OCK2 001212
OCK3 000004
TR   000032
AR   000020
MBRS 000012


EFERENCES (BLOCK, NAME)

ADER
COD1
RR
RR3$


SIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0002 10L        0001     000007 116G      0001     000220 98L       0001     000217 99L
0010 AREA       0012 R   000000 DECOD1    0005 R   000000 E         0000 R   000014 FLEN
0002 I          0000 I   000001 IGOOF     0000 I   000006 IJOINT    0000 R   000000 INER
0536 J          0006     000000 JAC       0007     000000 JAST      0006     000001 JBC
0000 JBUF       0006     000002 JCC       0007     000002 JCMA      0007     000006 JCOL
0014 JDIV       0007     000004 JDOL      0006     000004 JEC       0007     000017 JEQ
0120 JFOUND     0006     000006 JGC       0007     000015 JGT       0006     000007 JHC
0011 JJC        0000 I   000007 JJOINT    0006     000012 JKC       0007     000005 JLB
0011 JLP        0007     000016 JLT       0006     000014 JMC       0007     000013 JMIN
0016 JOC        0006     000017 JPC       0007     000003 JPER      0007     000012 JPLS
0021 JRC        0007     000010 JRP       0006     000022 JSC       0007     000001 JSEM
0024 JUC        0006     000025 JVC       0006     000026 JWC       0006     000027 JXC
0031 JZC        0000 I   000005 JI        0000 I   000004 L         0000 I   000003 MEMB
0015 NOMEM      0005 I   000002 NOMEN     0010 I   000000 NUM       0004     001130 P
0012 X          0000 R   000013 Y


          SUBROUTINE MEMINC
          COMMON /BLOCK1/ JBUF(80),JFOUND
          COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
          COMMON /BLOCK3/ E,NOJOI,NUMEN,G
          COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
         1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
          COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
         1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
          COMMON /NUMBRS/ NUM(10)
          REAL INERT
          JFOUND=1
   10     IGOOF=0
          CALL READER
          DO 5 I=1,10,1
```

```
5       IF(JBUF(1).EQ.NUM(I))IGOOF=1
        IF(IGOOF.NE.1)GO TO 98
        MEMBER=DECOD1(1,L)
        J1=L+1
        IJOINT=DECOD1(J1,L)
        J1=L+1
        JJOINT=DECOD1(J1,L)
        J1=L+1
        AREA=DECOD1(J1,L)
        J1=L+1
        INERT=DECOD1(J1,L)
        J1=L+1
        POLI=DECOD1(J1,L)
        NOMEN=NOMEN+1
        IF(IJOINT.EQ.0.OR.JJOINT.EQ.0.OR.MEMBER.EQ.0)GO TO 99
        IF(POLI.EQ.0.0.OR.INERT.EQ.0.0)GO TO 99
        J(MEMBER,4)=IJOINT
        J(MEMBER,5)=JJOINT
        X=A(JJOINT,6)-A(IJOINT,6)
        Y=A(JJOINT,7)-A(IJOINT,7)
        FLENGT=((X**2.)+(Y**2.))**0.5
        IF(FLENGT.EQ.0.0)GO TO 99
        A(MEMBER,4)=X/FLENGT
        A(MEMBER,5)=Y/FLENGT
        A(MEMBER,1)= FLENGT
        A(MEMBER,2)=INERT*E/FLENGT
2       A(MEMBER,3)=POLI*G/FLENGT
        GO TO 10
99      NOMEM=0
98      RETURN
        END
```

OMPILATION: NO DIAGNOSTICS.

```
LOADER     ENTRY POINT 000206

D: CODE(1) 000212; DATA(0) 000021; BLANK COMMON(2) 000000

CKS:

CK1 000121
CK2 001212
CK3 000004
CK4 000005
R   000032
R   000020
BRS 000012

FERENCES (BLOCK, NAME)

DER
OD1
R3$

IGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

132 13L        0001    000003 15L        0001    000153 40L       0001    000174 98L
000 A          0013 R  000000 DECOD1     0005    000000 E         0000 R  000005 FORCE
003 IEND       000C    000011 INJP$      0006    000002 ISOLV     0004 I  000536 J
000 JAST       0007    000001 JBC        0010    000007 JBLK      0003 I  000000 JBUF
002 JCMA       0010    000006 JCOL       0007    000003 JDC       0010    000014 JDIV
004 JEC        0010    000017 JEQ        0007 I  000005 JFC       0003    000120 JFOUN
015 JGT        0007    000007 JHC        0007    000010 JIC       0007 I  000011 JJC
005 JLB        0007    000013 JLC        0010    000011 JLP       0010    000016 JLT
013 JMIN       0007    000015 JNC        0007    000016 JOC       0000 I  000002 JOINT
003 JPER       0010    000012 JPLS       0007    000020 JQC       0007    000021 JRC
022 JSC        0010    000001 JSEM       0007    000023 JTC       0007    000024 JUC
026 JWC        0007 I  000027 JXC        0007 I  000030 JYC       0007 I  000031 JZC
003 L          0000 I  000000 LOC        0000 I  000004 LOCJP1    0000 I  000006 LOCJP
004 NEWPRO     0005    000001 NOJOI      0006 I  000001 NOLOAD    0005    000002 NOMEN
000 NUM        0004 R  001130 P
```

```
      SUBROUTINE LOADER
C THIS SUBROUTINE HANDLES LOADING CONDITIONS
C THE ROUTINE IS CALLED FROM INTERP AND TRTURNS THERE.
C THE LOADS ARE ENTERED INTO THE P VECTOR
      COMMON /BLOCK1/ JBUF(80),JFOUND
      COMMON /BLOCK2/ A(50,7),J(50,5),P(50)
      COMMON /BLOCK3/ E,NOJOI,NUMEN,G
      COMMON /BLOCK4/ NOREL,NOLOAD,ISOLV,IEND,NEWPRO
      COMMON /LETR/ JAC,JBC,JCC,JDC,JEC,JFC,JGC,JHC,JIC,JJC,JKC,JLC,JMC,
     1              JNC,JOC,JPC,JQC,JRC,JSC,JTC,JUC,JVC,JWC,JXC,JYC,JZC
      COMMON /CHAR/ JAST,JSEM,JCMA,JPER,JDOL,JLB,JCOL,JBLK,JRP,JLP,
     1              JPLS,JMIN,JDIV,JGT,JLT,JEQ
      COMMON /NUMBRS/ NUM(10)
      NOLOAD=0
```

```
      JI=0
      LOC=0
15    CONTINUE
      CALL READER
      IF(JBUF(1).NE.JJC)GO TO 99
      NOLOAD=1
      JOINT=DECOD1(6,L)
      L=L+1
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JXC)LOC=1
      IF(JBUF(L).EQ.JMC.AND.JBUF(L+6).EQ.JYC)LOC=2
      IF(JBUF(L).EQ.JFC.AND.JBUF(L+5).EQ.JZC)LOC=3
      IF(LOC.NE.1)GO TO 13
14    LOCJP1=J(JOINT,1)
      J1=L+7
      FORCE=DECOD1(J1,L)
      P(LOCJP1     )=FORCE
      GO TO 15
13    IF(LOC.NE.2)GO TO 40
16    LOCJP2=J(JOINT,2)
      J1=L+7
      FORCE=DECOD1(J1,L)
      P(LOCJP2     )=FORCE
      GO TO 15
40    IF(LOC.NE.3)GO TO 98
41    LOCJP3=J(JOINT,3)
      J1=L+6
      FORCE=DECOD1(J1,L)
      P(LOCJP3)=FORCE
      GO TO 15
98      NOLOAD=0
99    RETURN
      END

COMPILATION:          NO  DIAGNOSTICS.



-13:34
PROG SIZE(I/D)=7043/8741


VEL 7C-1
ION - TIME 1.849 SECONDS
```