

AD-A016 778

FEASIBILITY STUDY ON DESIGN OF SELF-ERECTING AIRCRAFT  
SHELTER

Roy L. Johnson

New Mexico University

Prepared for:

Air Force Weapons Laboratories

October 1975

DISTRIBUTED BY:

**NTIS**

National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE

ADA01678



# FEASIBILITY STUDY ON DESIGN OF SELF-ERECTING AIRCRAFT SHELTER

Roy L. Johnson

CERF, University of New Mexico  
Albuquerque, New Mexico 87131

October 1975

Final Report



Approved for public release; distribution unlimited.

Reproduced by  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
US Department of Commerce  
Springfield, VA 22151

**AIR FORCE WEAPONS LABORATORY**  
Air Force Systems Command  
Kirtland Air Force Base, NM 87117

This final report was prepared by CERF, University of New Mexico, Albuquerque, New Mexico, under Contract F29601-74-C-0008, Job Order ILIR7425 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Lt Dennis Morrison (OL-AA, AFCEC/DEZ) was the Laboratory Project Officer-in-Charge. Captain Galen C. Bessert was the former project officer.

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

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

*Dennis Morrison*  
DENNIS MORRISON  
Lt, USAF  
Project Officer

SEARCHED	INDEXED
SERIALIZED	FILED
DEC 1975	
AFCEC/DEZ	
BY	
CONTINUOUS AVAILABILITY	
The	
A	

*Loren M. Womack*  
LOREN M. WOMACK  
Chief, Aerospace Facilities Division

*Kenneth R. Porter*  
KENNETH R. PORTER, Major, USAF  
Commander, AFCEC/OL-AA

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-75-139	2. GOVT ACCESSION NO	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FEASIBILITY STUDY ON DESIGN OF SELF-ERECTING AIRCRAFT SHELTER		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Roy L. Johnson		8. CONTRACT OR GRANT NUMBER(s) F29601-74-C-0030
9. PERFORMING ORGANIZATION NAME AND ADDRESS CERF, University of New Mexico Albuquerque, New Mexico 87131		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101E, ILIR, 74, 25
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory Kirtland Air Force Base, New Mexico 87117		12. REPORT DATE October 1975
		13. NUMBER OF PAGES <del>68</del> 70
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Shelters Aircraft Shelters                      Structures Air Mobility                                Structural Analysis Computer Code Erection                                    Bare Base		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This research examined the feasibility of designing and constructing an air-transportable, self-erecting aircraft shelter. Behavior of the shelter during erection and diserection was studied by field testing a 1/4-scale, segmented arch model of semicircular geometry, and by analyzing the behavior of a circular segmental prototype structure. Analyses of the structural behavior of the model and prototype were performed using the matrix-dispacement method. Large displacements of the structure were accommodated analytically by (OVER)		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ABSTRACT (Cont'd)

use of a procedure of incremental displacement followed by an updating of structure geometry. A computer program (BOOTSTRAP) was developed to perform these analyses. Analyses showed that stable configurations of the 1/4-scale semicircular arch were achieved only when a single internal hinge was allowed in the structure. A satisfactory diserection procedure for the full-scale prototype was not achieved.

ia

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION	1
2	MODEL TESTS	2
3	ANALYTICAL FORMULATION	8
4	MODEL STUDIES	11
5	PROTOTYPE STUDIES	14
6	CONCLUSIONS AND RECOMMENDATIONS	17
	APPENDIX: PROGRAM BOOTSTRAP	19
	PROGRAM STRUCTURE	20
	INPUT INSTRUCTIONS	28
	PROGRAM LISTING	35

## ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Model of Self-Erecting Aircraft Shelter	3
2	Joint-Locking Device	4
3	Hinge-Rotation Limiting Device	5
4	Partially Erected Model Configurations	6
5	Computed Configurations of Model	13
6	Prototype Configurations	15
7	Distribution of Bending Moments in Low-Profile Prototype	16

## SECTION 1 INTRODUCTION

### BACKGROUND

With funding from the AFWL, faculty members of the USAF Academy successfully demonstrated the feasibility of constructing a model of a self-erecting aircraft shelter. Thus, a model consisting of sixteen 2-ft-long prefabricated modules connected to one another by a hinge mechanism was developed. Controlled buckling of the structure at the hinges is obtained by applying a moment to opposite ends of the horizontal beam column by a cable attached to the ends. Since the structure resists bending in one direction and not the other, it is gradually erected into a semicircular arch. This relatively rapid and simple construction technique could make practical immediate full-scale prototype application in support of highly mobile tactical air forces. Therefore, feasibility studies on developing a prototype and design of such a structure were undertaken at the Civil Engineering Research Facility (CERF).

### OBJECTIVE

The overall objective of this research effort was to design an air-transportable aircraft shelter which can be erected in the field with a minimum amount of equipment other than that provided with the shelter. Therefore, the feasibility of such a structural system with respect to the self-erection technique was examined.

### SCOPE

Field tests and analytical studies of a 1/4-scale model of a semicircular arch and analytical studies of a full-scale prototype were conducted. An existing computer program was modified and used to model the large displacement behavior of the arch.

## SECTION 2 MODEL TESTS

The 1/4-scale model in its fully erected position is shown in figure 1. Details of the joint-locking and hinging mechanisms are shown in figures 2 and 3.

When received at CERF, all joints of the model were free hinged with the exception of the first two or three joints near the ends; these were equipped with gravity-actuated latches to lock the joints in their fully closed positions (fig. 2b). Each joint was equipped with a bolt mechanism (fig. 3) which limited the relative rotations of adjacent members when the joint was opened. Positive or negative relative rotations of the members may be obtained by adjusting this bolt. (A positive relative rotation is measured counterclockwise between longitudinal axes of adjacent members.) However, the bolts provided with the model (as received from the USAF Academy) permitted only negative relative rotations of the members (fig. 3a). Therefore, the bolt mechanisms were modified at CERF to permit positive relative rotations as well (fig. 3b).

Several tests were performed on the 1/4-scale model to observe overall behavior during erection and diserection. From these tests and from films of tests at the USAF Academy, it was evident that when no positive rotation of adjacent members is permitted (model as received), the maximum loading on the structure occurs when the first panel is raised (fig. 4a). To minimize the bending moment in the arch, the joints were modified to permit positive relative rotations of the members so as to minimize the span over which the weight of the structure is carried (fig. 4b).

Observation of the erection and diserection of the arch with the modified joints revealed the following facts:

- (1) Cable tensions during erection were more constant in the early stages although somewhat higher than those when no positive rotations were permitted. As expected, a number of panels remained flat on the pad, thus reducing the load which the semierected arch carried.

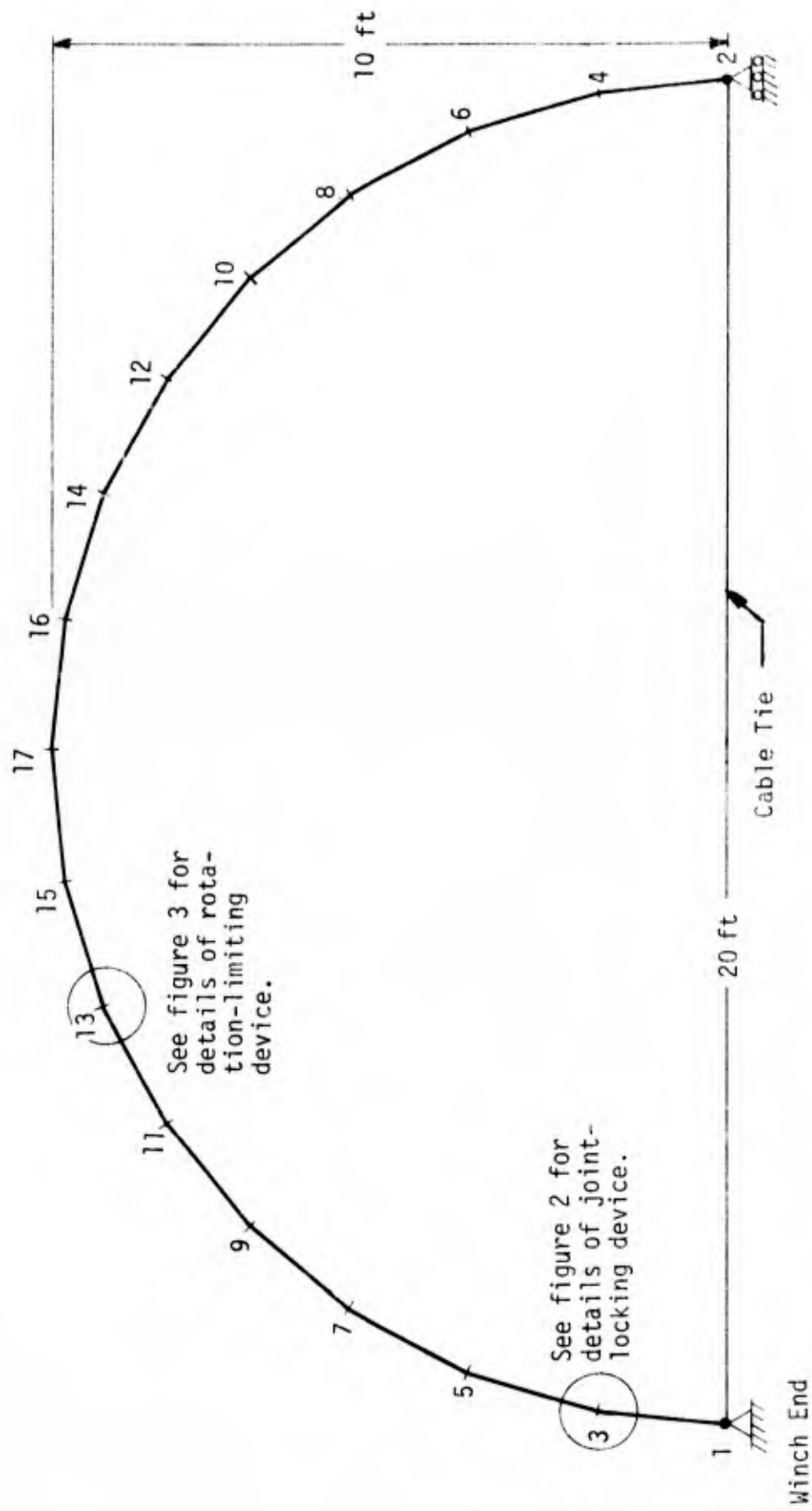
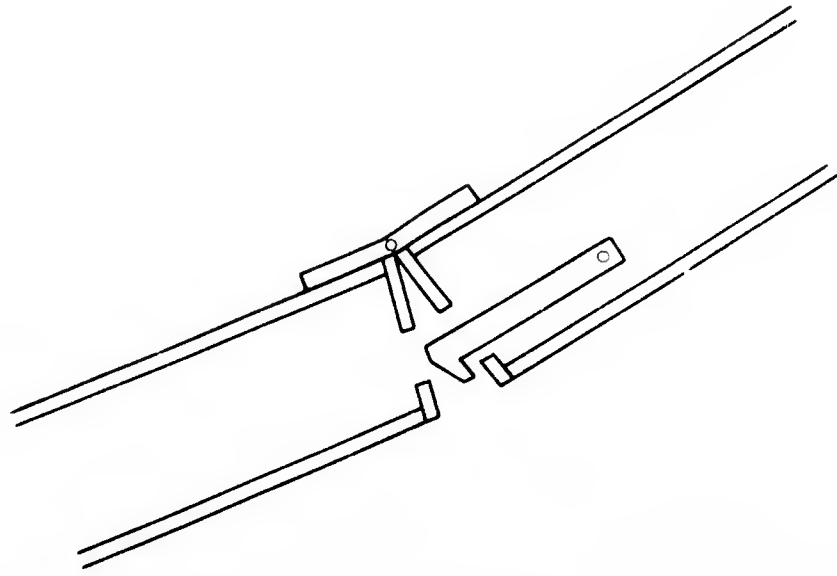
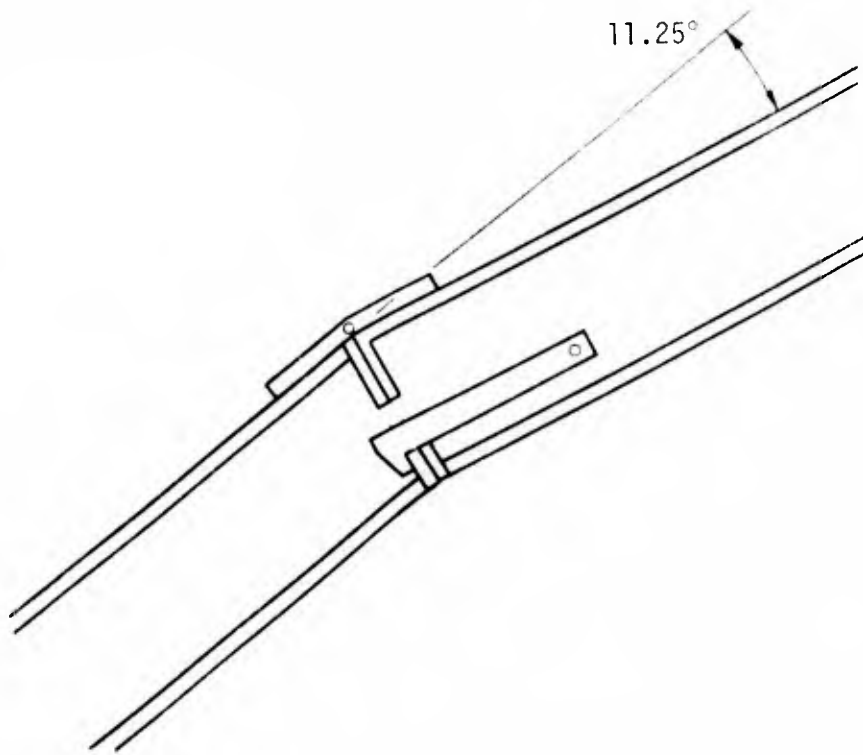


Figure 1. Model of Self-Erecting Aircraft Shelter

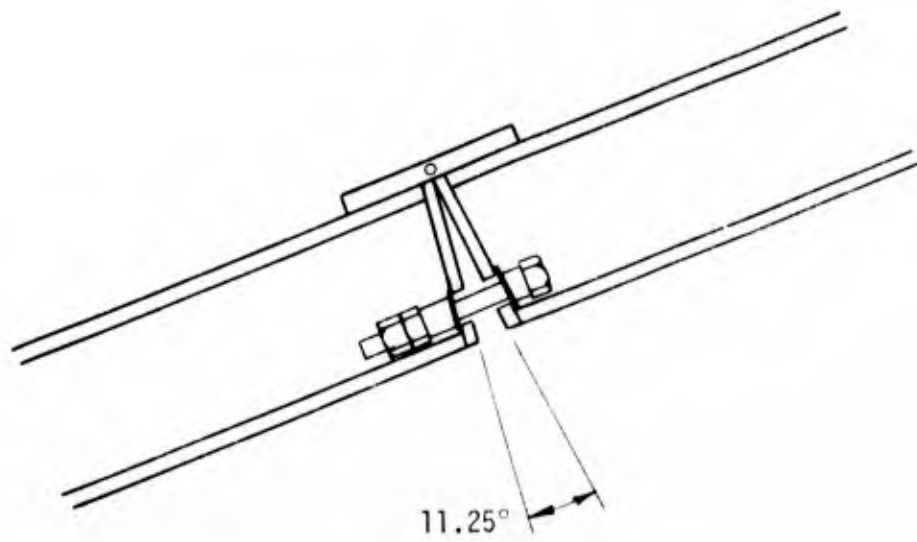


(a) Opened Joint

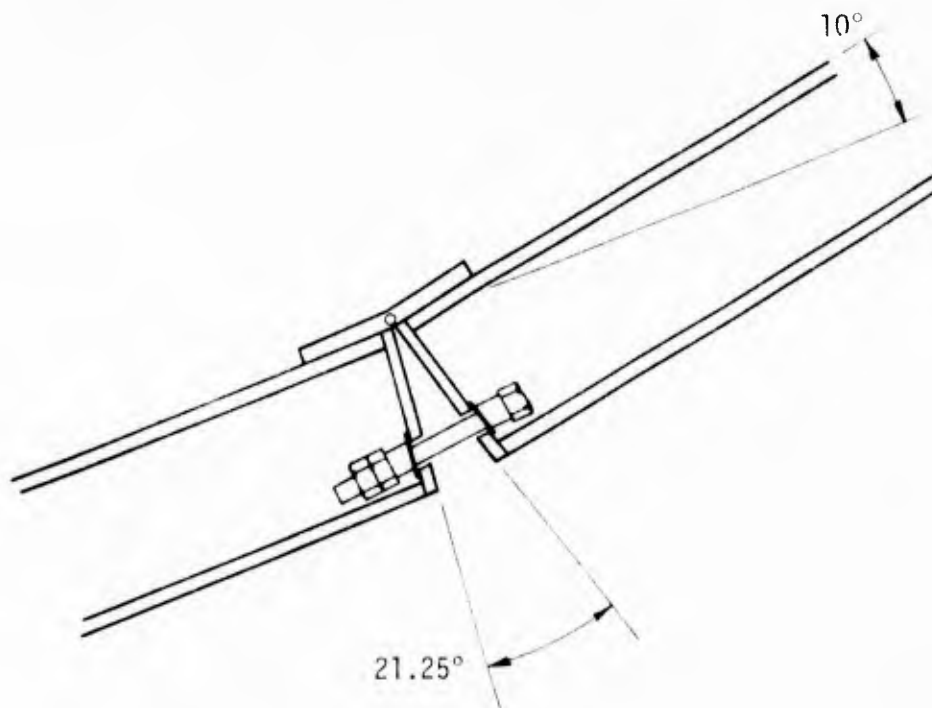


(b) Locked Joint

Figure 2. Joint-Locking Device

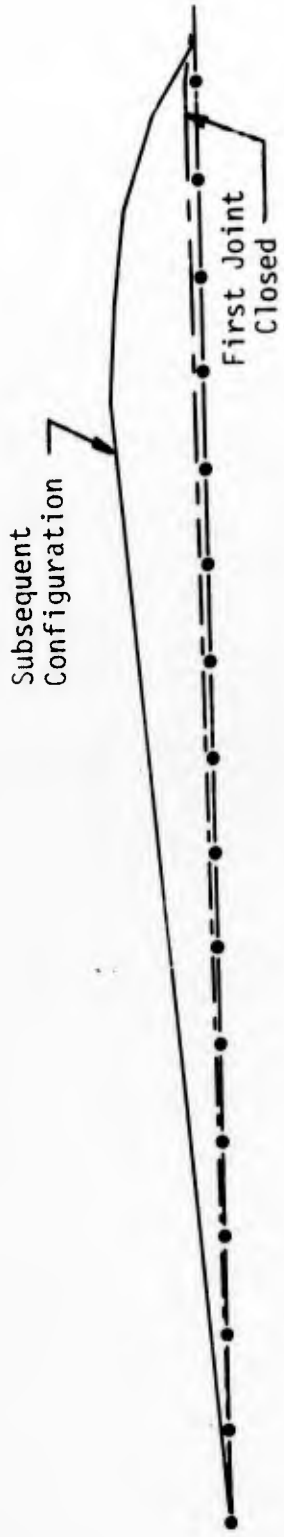


(a) Before Modification



(b) After Modification

Figure 3. Hinge-Rotation Limiting Device



(a) Before Hinge Modification



(b) After Hinge Modification

Figure 4. Partially Erected Model Configurations

(2) The model assumed a symmetric mode during diserection; i.e., the center joint opened and the structure symmetrically lowered. In the later stages of diserection, the structure *snapped-through* into an asymmetric mode and continued in this mode until diserection was completed. The snap-through was violent and several joint pins failed. This snap-through was also observed before joint modification; however, it was less violent because of the restricted rotation of the joints.

To avoid snap-through, a modification was made to the cable arrangement to allow the cable to be passed through a bracket at the second joint above the winch. With this arrangement, the active cable tension was utilized to load the structure asymmetrically, thus forcing it into an asymmetric mode for diserection. Subsequent testing of the model with this arrangement showed that adequate control of the configuration could be maintained during diserection and snap-through of the arch could be avoided.

### SECTION 3 ANALYTICAL FORMULATION

The structure was modeled analytically as a circular arch composed of straight-line segments. Only one internal hinge was permitted at a time. Thrusts of the arch were equilibrated by a cable tie capable of transmitting axial force only. Horizontal displacement of one of the supports was induced by lengthening the cable tie and allowing the structure to displace under its own weight. Displacements were applied in small increments to permit the use of a linear stiffness matrix formulation. The geometry of the structure was updated at the end of each displacement increment by adding algebraically the elastic displacements of the structure to the previous structure geometry. This permitted the structure to undergo very large displacements without the use of a nonlinear geometric formulation.

#### MATRIX FORMULATION

The static equations of equilibrium for a fixed configuration of the structure were

$$\{F\} = [K]\{u\} \quad (1)$$

where  $\{F\}$  and  $\{u\}$  are vectors of the generalized nodal forces and displacements, respectively, and  $[K]$  is the assembled stiffness matrix of the structure formed by appropriately summing the stiffness matrices for the individual elements of the structure. An element is defined as that portion of the structure which connects two nodes (joints). The formulation of  $[K]$  may be found in any standard text on matrix structural analysis (ref. 1).

An equilibrium configuration is obtained from eq. (1).

$$\{u\} = [K]^{-1}\{F\} \quad (2)$$

---

1. Gere, J. M., and Weaver, W., Jr., *Analysis of Framed Structures*, Van Nostrand, New York, 1965.

Here, the inversion of  $[K]$  is implied; but, in the actual computation a solution was obtained by solving the set of simultaneous linear equations.

The vector  $\{F\}$  is a set of generalized nodal forces which consist of those derived from the gravity load, i.e., the weight of the structure, and those induced by initial support displacement. Accordingly, the vector  $\{F\}$  may be written as the sum of the two parts; thus

$$\{F\} = \{F\}_g + \{F\}_{u_i} \quad (3)$$

where the subscripts  $g$  and  $u_i$  indicate gravity forces and forces due to initial displacements, respectively.  $\{F\}_g$  is obtained by appropriately summing generalized nodal forces which are statically equivalent to the gravity loads for each element of the structure.

Initial support displacements are transformed into equivalent generalized forces by

$$\{F\}_{u_i} = [K]\{u\}_i \quad (4)$$

where  $\{u\}_i$  is a set of initial displacements specified in small increments.

The stiffness matrix is reformulated at the end of each increment by algebraically adding the spatial coordinates of each node at the beginning of the increment to the corresponding nodal displacements occurring during that increment. The updated geometry is then

$$\begin{aligned} x_{i+1} &= x_i + u_i \\ y_{i+1} &= y_i + v_i \\ z_{i+1} &= z_i + w_i \end{aligned} \quad (5)$$

where  $x$ ,  $y$ , and  $z$  are spatial coordinates of a given node and  $u$ ,  $v$ , and  $w$  are nodal displacements in the corresponding directions. The subscript  $i$  indicates the geometry at the beginning of the  $i$ th increment. The procedure is repeated until the structure has been put through the full range of displacements desired.

The logic necessary to perform the computer analysis of the structure is presented in the appendix.

#### SECTION 4 MODEL STUDIES

The behavior of the model during testing indicated that an analytical procedure capable of predicting equilibrium configurations and stresses in the arch at various stages of erection and diserection was necessary. Also, other than semicircular configurations should be investigated in order to minimize structural height while maintaining necessary work space. A computer solution seemed most appropriate. Therefore, an existing University of New Mexico computer program for displacement analysis of elastic structures was extensively modified for this purpose. (See appendix.)

The fully hinged 1/4-scale model was not successfully analyzed; however, modifications to permit analysis as well as to control configurations during diserection were investigated. With more than one joint opened in the arch, the structure is a mechanism and only conditionally stable. The stiffness matrix is singular and a solution was not possible. Attempts to maintain small residual bending and translational stiffnesses at the joints by using elements with semirigid connections, such as those described by Gere and Weaver (ref. 1), also resulted in solutions which were unstable unless substantial residual stiffness was maintained, but this resulted in structural models which were too stiff when compared with the physical model.

In consultation with Captain Bessert and Lieutenant Morrison, AFWL/DEZ Task Officers, it was decided to investigate the possible locking of certain joints of the structure so as to maintain only one free hinge and permit unlocking of subsequent joints when the previously designated joint had fully opened and regained bending stiffness. This locking and unlocking sequence could be determined in such a way that the structure could be forced to diserect in an

---

1. Gere, J. M., and Weaver, W., Jr., *Analysis of Framed Structures*, Van Nostrand, New York, 1965.

asymmetric mode. Subsequent analytical studies of the 1/4-scale model indicated that such a procedure was feasible.

Figure 5 shows the computed configurations of the model at two different stages of diserection. Joint 3 was initially designated as a hinge and all other joints were locked. The first configuration shows the geometry of the displaced structure with joint 3 fully opened and with continuity restored at that joint. Subsequent hinges were designated at joints 5, 7, 9, and 11 and formed sequentially as the right end of the arch was further displaced. The second configuration shows joint 11 partially opened with all other joints locked. No joint was allowed to open until the previously designated hinged joint had fully opened. If this procedure is followed, the arch immediately goes into an asymmetric, stable mode for diserection.

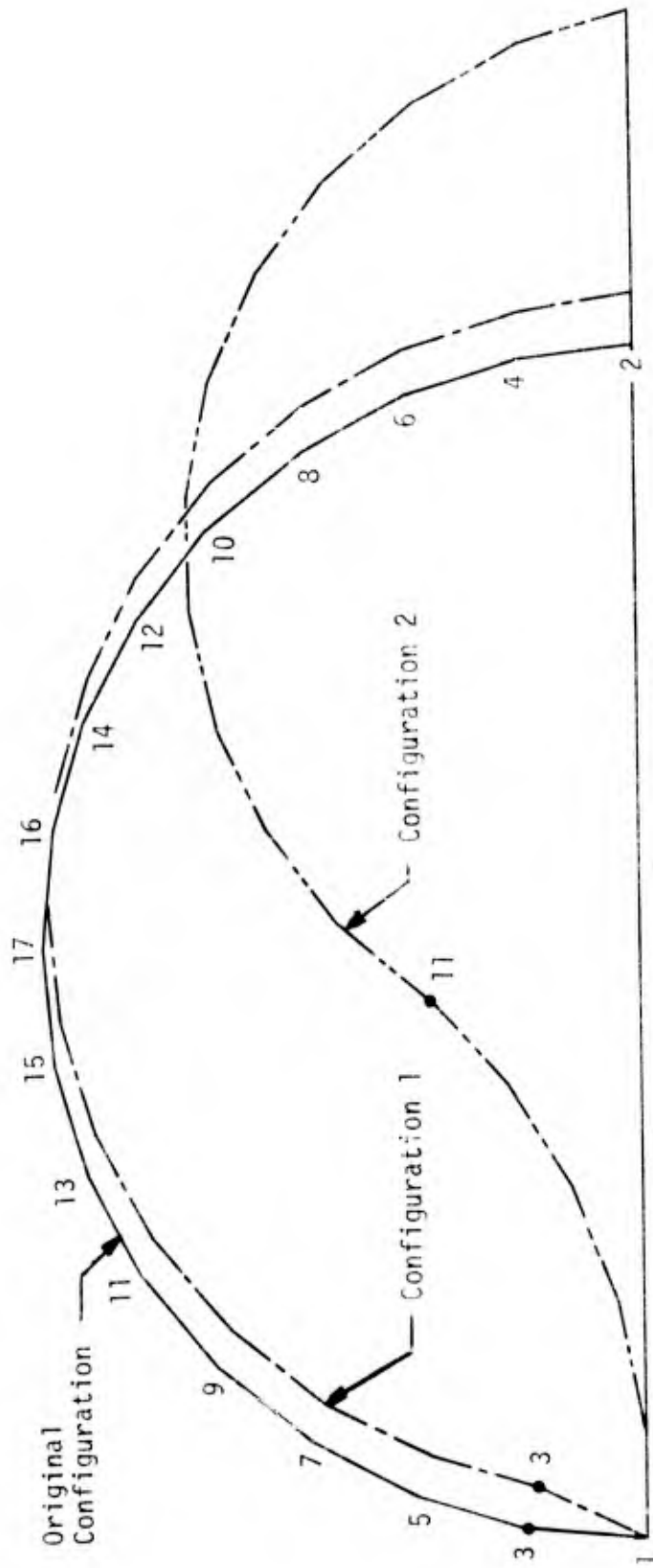


Figure 5. Computed Configurations of Model

## SECTION 5 PROTOTYPE STUDIES

The low-profile configuration of the fully erected prototype structure is shown in figure 6 along with a semicircular configuration. The dimensions of the prototype structure were established by AFWL/DEZ officers to accommodate the latest tactical aircraft. The arch consists of fourteen 2-ft-long panels hinged at their ends in a manner similar to that used for the 1/4-scale model. The arch is erected by a cable attached to the ends of the structure. When the cable is retracted by a winch, controlled buckling of the structure at the hinges results. Rotations at the hinges are limited so as to produce the final low-profile configuration when the arch is fully erected.

Several joint-hinging sequences were investigated; none of these, however, resulted in a stable, asymmetric configuration during diserection.

A hinge located on the bottom third of the arch closed and remained closed during subsequent incremental displacements. The analysis showed that the crown of the arch tended to snap through in the later stages of diserection, resulting in large, downward displacements. Locating a hinge at the crown resulted in a nearly symmetric displacement configuration. However, the analysis showed that this configuration goes unstable in the later stages of diserection.

The low-profile configuration substantially changes the behavior of the prototype from that of the semicircular model. There is appreciably higher thrust at the supports of an arch subtending an angle less than  $180^\circ$  than at the supports of a semicircular arch of comparable span. Figure 7 shows the bending moments resulting from the elastic deformation of the low-profile arch. The points of contraflexure are approximately located at joints 11 and 12, below which moments are negative and tend to close any hinge located in these regions. Subsequent displacements of the right end of the arch do not substantially change this until large displacements of the crown occur. At this time the structure becomes unstable.

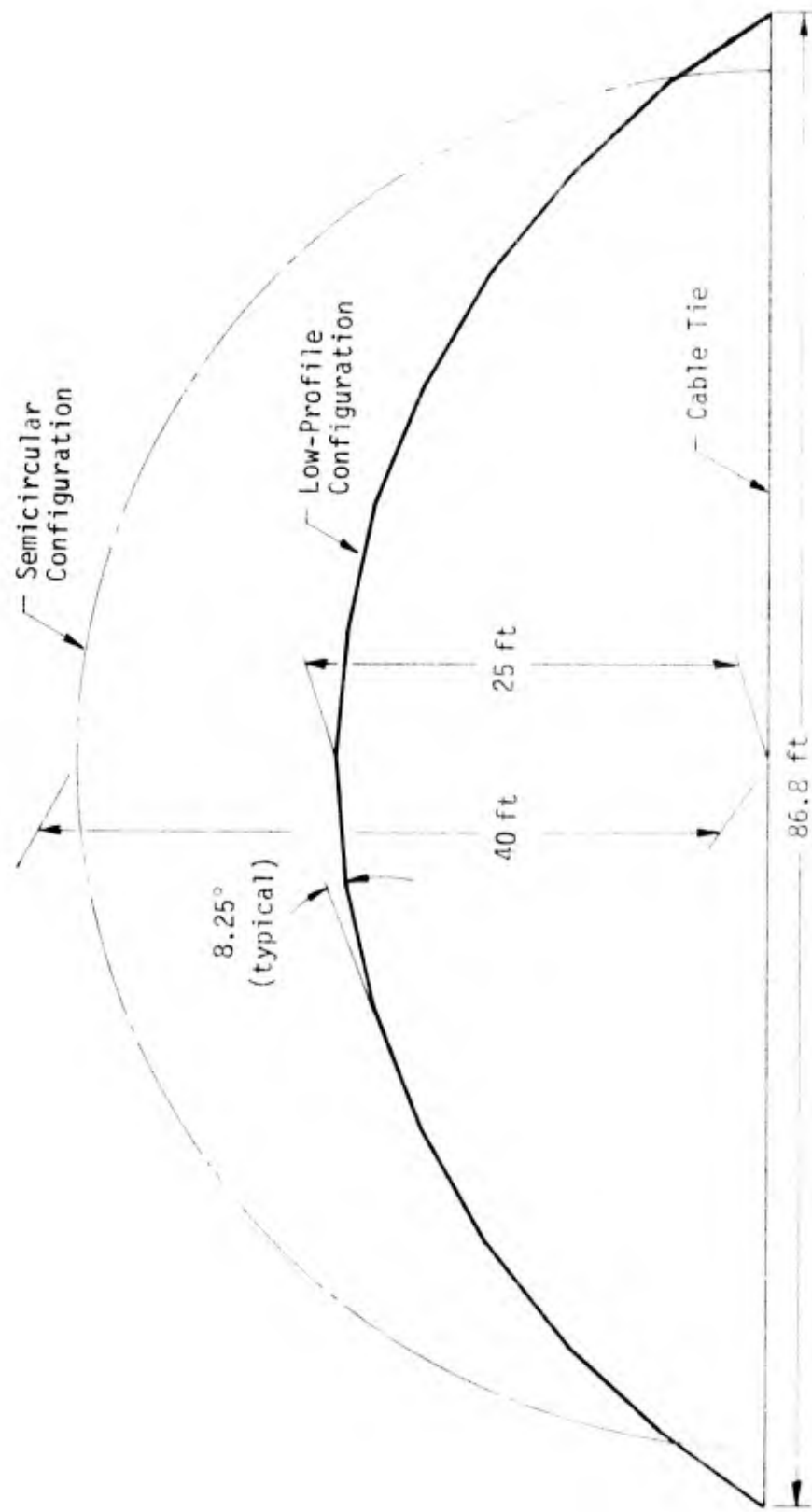


Figure 6. Prototype Configurations

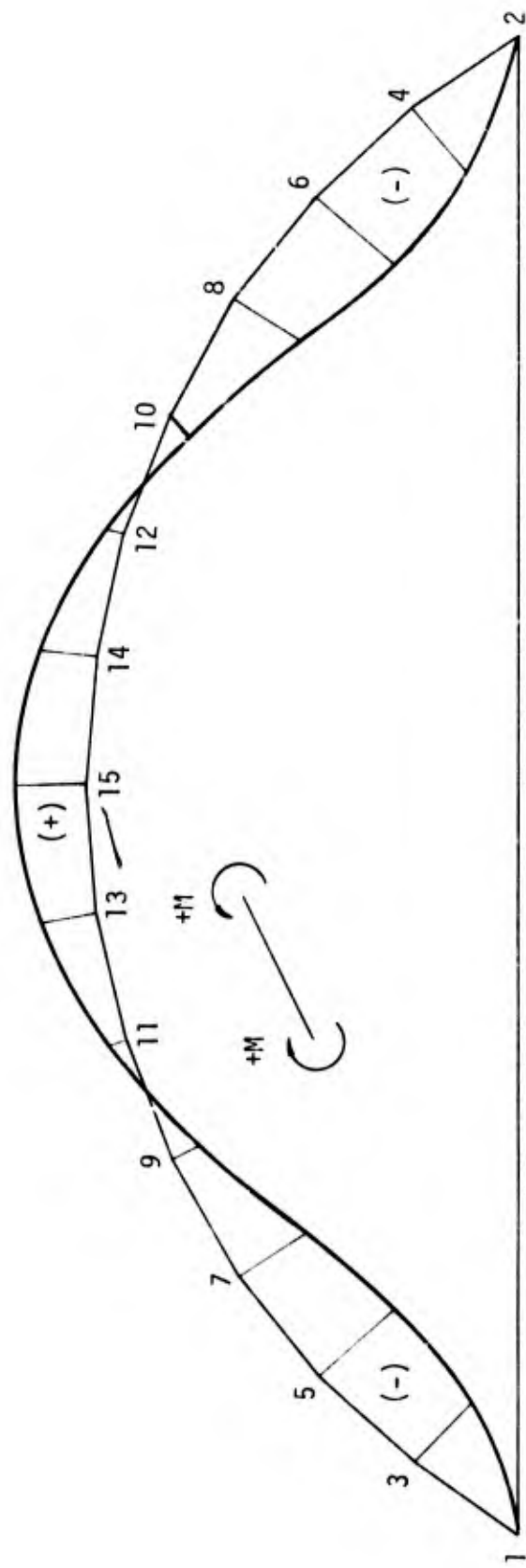


Figure 7. Distribution of Bending Moments in Low-Profile Prototype

SECTION 6  
CONCLUSIONS AND RECOMMENDATIONS

The analytical studies of the 1/4-scale model showed that, for a semicircular geometry, the single-hinge concept was successful in producing a stable configuration during diserection. This was supported by observations of the behavior of the physical model during the tests. From this, it is also concluded that the analysis procedure incorporated into the computer program used in the analytical studies is adequate to predict the large displacement behavior of the structure.

The inability to identify joint-hinging sequences which result in a satisfactory diserection procedure for the prototype structure suggests that the self-erecting technique is not feasible with the low-profile configuration for the following reasons:

- (1) There is a possibility of instabilities of the arch during diserection.
- (2) The size of the cable necessary for erection and diserection of the prototype appears to be excessive (possibly 1-inch diameter) with subsequent requirements of large size and high cost as well as weight of the cable winching apparatus.

Analytical studies of other configurations should be made in order to fully assess the feasibility of constructing a self-erecting arch.

APPENDIX  
PROGRAM BOOTSTRAP

A Special Purpose Program for the Large Displacement  
Analysis of Circular, Segmental, Tied Arches

## PROGRAM STRUCTURE

### BASIC PROGRAM LOGIC

A solution is obtained by a two-step procedure as follows:

- (1) A displacement increment is applied to one of the end supports of the arch and an elastic analysis is performed using the displacement method to determine an equilibrium configuration.
- (2) The configuration of the structure is updated and a new elastic solution is obtained using the new configuration. This step *stress relieves* the structure of the forces induced by the incremental displacement. The force in the cable tie is thus a result of the elastic deformation of the structure due to gravity load only, using the configuration computed in step 1 as a reference.

Possible events occurring during step 1 are as follows:

- (1) The designated hinge joint may close. Action: Restore continuity and resolve before proceeding to step 2.
- (2) The designated hinge joint may partially open. Action: Proceed to step 2.
- (3) The designated hinge joint may open beyond the maximum value specified. Action: Iterate to find the incremental displacement which just causes the joint to open to its maximum specified value before proceeding to step 2.

Possible events occurring during step 2 are as follows:

- (1) The designated hinge joint may close. Action: Restore continuity and resolve.
- (2) The designated hinge joint may partially open. Action: Accept the solution and print results.

- (3) The designated hinge joint may open beyond the maximum value specified. Action: Restore continuity and resolve.

The program consists of a main calling routine (BOOTSTRAP) and the following subroutines:

<u>Name</u>	<u>Principal Function</u>
(1) PLOTXY	Plots deformed structure.
(2) INPUT	Reads input data.
(3) STIFF	Calls element stiffness routine.
(4) BEAM	Forms element stiffness matrix.
(5) LOAD	Computes element load vector.
(6) DLD	Computes member forces due to gravity loads.
(7) MERGE	Assembles structure stiffness matrix and load vector.
(8) BOUND	Applies boundary constraints.
(9) SOLVER	Solves structure stiffness equations.
(10) FORCE	Computes member forces.
(11) MODIFY	Calls modified element stiffness routine.
(12) GEOM	Updates structure geometry.
(13) BM	Forms modified element stiffness matrix.
(14) ROTATE	Performs coordinate transformations.
(15) ANGCHK	Checks relative member joint rotations.
(16) SUPCHK	Checks joints for negative coordinates.

#### MAIN CALLING ROUTINE

BOOTSTRAP is the main routine which calls INPUT, STIFF, MODIFY, BOUND, SOLVER, FORCE, GEOM, and PLOTXY. The principal parameters which control branching are NONLIN, IUPDTE, IMOD, ITER, LASTIT, and ICONV. NONLIN is an input parameter with a value of 0 or 1 depending on whether a linear or nonlinear problem, respectively, is to be solved. (See section 2.) IUPDTE, IMOD, LASTIT, and ICONV

are defined in subroutine ANGCHK or SUPCHK and may have values of 0 or 1. ITER is an integer set to 1 or 2. When ITER = 1, an incremental displacement is imposed on the structure. ITER is set to 1 at three stages in the solution:

- (1) When an incremental displacement is imposed initially;
- (2) During iteration to find the incremental displacement which just causes a joint to open fully; and
- (3) At the completion of a solution cycle.

When ITER = 2, the structure geometry is updated and a new solution is obtained.

## SUBROUTINES

### PLOTXY

PLOTXY calls a standard IBM/360 system subroutine called PLOT which plots updated joint coordinates on the line printer. Any other plot routine may be used in lieu of PLOT for operation on another system. The y-displacements of the structure may be amplified by using the factor AMPLF. (See section 2.)

### INPUT

INPUT reads and prints control parameters and basic structure data such as material data, nodal point coordinates, member data, boundary constraint data, member release data, joint release sequences, and structure loading codes. The half-bandwidth of the structure stiffness matrix is computed and printed in this subroutine.

### STIFF

STIFF forms the structure stiffness matrix. Only structures without member releases are handled in this subroutine. Member stiffness matrices are generated in BEAM and transformed to global coordinates in a call to ROTATE before being assembled in MERGE.

## BEAM

BEAM forms the element stiffness matrices for only those elements without member releases.

## LOAD

LOAD reads and prints concentrated joint forces, computes member gravity loads in a call to DLD, reads and prints member fixed-end forces from distributed loads, and computes and prints member forces due to initial joint displacements. Member forces are transformed to global coordinates in a call to ROTATE before being assembled in MERGE to form the structure load vector.

## DLD

DLD computes the equilibrating end forces for each member for the g-load specified. (See section 2.) Forces are computed for the following conditions of end fixity: fixed-fixed, hinged-hinged, hinged left, and hinged right.

## MERGE

MERGE forms the structure stiffness matrix and the load vector by merging the element stiffness matrices and the load vectors. The structure stiffness matrix is stored in array A(NUMEQ,MBAND), where NUMEQ is the number of simultaneous linear equations ( $\text{NUMEQ} = 2 * \text{NUMNP}$ ) and MBAND is the half-bandwidth of the stiffness matrix. The structure load vector is stored in array B(NUMEQ).

## BOUND

BOUND imposes boundary constraints on the structure stiffness matrix and on the structure load vector.

## SOLVER

SOLVER uses Cholesky's Method (ref. 2) for symmetrical matrices (Square Root

2. Ketter, Robert L., and Prawel, Sherwood P., Jr., *Modern Methods of Engineering Computation*, McGraw-Hill Book Co., New York, 1969.

Method or Banachiewicz Method) to solve the system of simultaneous linear equations. The stiffness matrix is reduced in a forward sweep only when modifications to the structure occur, i.e., when the geometry is updated or additional hinges are formed. The partially reduced stiffness matrix is used to obtain solutions for each new load vector during iteration to find the incremental displacement which just causes the designated joint to open fully.

#### FORCE

FORCE computes member forces in global coordinates and transforms them to member coordinates in a call to ROTATE. Member forces are printed in the member coordinate system. Element stiffness matrices are regenerated for use in FORCE rather than written on and read from an external input/output unit.

#### MODIFY

MODIFY forms the structure stiffness matrix including members with releases. Member stiffness matrices are generated in BM or in BEAM depending on whether or not the member contains a release and transformed to global coordinates in ROTATE before being assembled in MERGE.

#### GEOM

GEOM updates structure geometry by summing nodal spatial coordinates and corresponding nodal displacements. Relative rotations of adjacent members at each joint are computed using the updated geometry. A check on maximum and minimum values of joint angles and minimum values of joint coordinates are made in ANGCHK and SUPCHK which are called from this subroutine.

#### BM

BM generates stiffness matrices for members containing releases. Three types of stiffness matrices are generated depending on the type of member release specified: hinged-hinged, hinged left, and hinged right.

## ROTATE

ROTATE transforms member stiffness matrices and load vectors into global coordinates. Member force vectors are transformed from global coordinates to member coordinates.

## ANGCHK

ANGCHK checks relative member rotations to detect joint openings and closings. The check is made on the angle  $\phi$  (defined in section 2) for the interior joint which has been designated as a hinge. Only three hinges, including supports, are permitted in the structure. A hinge is modeled by specifying releases in members joined at the designated hinge joint. A boundary constraint at the joint which prohibits rotation of the joint about the z-axis is necessary to reduce the singularity in the stiffness matrix resulting from zero bending stiffness at the joint. The current and previous member releases are stored in two arrays, IREL(NUMEM,2) and JREL(NUMEM,2); the first column contains the previous releases and the second column contains the current releases. The values in IREL and JREL are 1 or 0 depending on whether the member is or is not released, respectively. These values change as joints open and close.

The principal parameters assigned in this subroutine and used to control branching are ICLOSE, JCLOSE, IUPDTE, IMOD, LASTIT, and ICONV. These parameters are assigned values of 0 or 1 as indicated in the following table:

<u>Event</u>	<u>ITER</u>	<u>Set Flag</u>					
		<u>ICLOSE</u>	<u>JCLOSE</u>	<u>LASTIT</u>	<u>IUPDTE</u>	<u>IMOD</u>	<u>ICONV</u>
Designated hinge joint closes on first iteration (JITER = 1). $\phi \leq \phi_{\min}$ .	1	0	1	1	0	1	0
Designated hinge joint opens partially. $\phi_{\min} < \phi < \phi_{\max}$ .	1	0	0	1	1	0	1
Designated hinge joint opens fully (JITER = 1). $\phi > \phi_{\max}$ . (Iterate to find new incremental displacement.)	1	0	0	1	0	0	0

table (Cont'd)

<u>Event</u>	<u>ITER</u>	<u>Set Flag</u>					
		<u>ICLOSE</u>	<u>JCLOSE</u>	<u>LASTIT</u>	<u>IUPDTE</u>	<u>IMOD</u>	<u>ICONV</u>
Solution has converged. $\phi = \phi_{\max}$ . Next joint in sequence is opened.	1	0	0	1	1	1	1
Designated hinge joint closes. $\phi \leq \phi_{\min}$ .	2	0	0	1	0	1	0
Designated hinge joint opens partially. $\phi_{\min} < \phi < \phi_{\max}$ .	2	0	0	0	0	0	1
Designated hinge joint opens fully. $\phi > \phi_{\max}$ .	2	1	0	1	0	1	0

JITER is a counter incremented at the end of each solution cycle. LASTIT = 1 signals that one or more additional iterations are necessary. ICONV = 1 signals convergence of the iteration process when finding the incremental displacement which just causes the designated hinge joint to open fully; the Regula Falsi Method (ref. 3) is used to find this displacement increment. IUPDTE = 1 signals that the structure geometry is to be updated for the next solution cycle. IMOD = 1 signals opening or closing of a joint not previously opened or closed. When IUPDTE or IMOD is 1, the structure stiffness matrix is reformulated for the next solution cycle.

#### SUPCHK

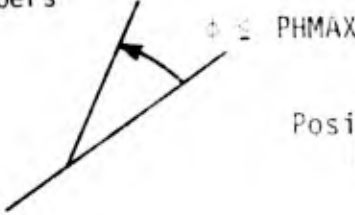
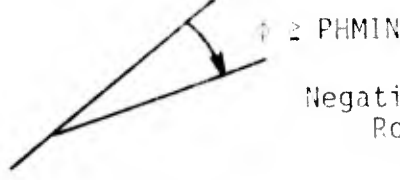
SUPCHK checks the updated y-coordinates of each interior joint for negative values. This may occur during an initial displacement increment, whereupon an iteration procedure is used to find the incremental displacement which just causes the y-coordinate of the most critical joint to have a value of zero. The most critical joint is taken to be the joint having the least value of the y-coordinate algebraically less than zero. The Regula Falsi Method is used in this iteration.

- 
3. Carnahan, Brice, Luther, H. A., and Wilkes, James O., *Applied Numerical Methods*, John Wiley & Sons, Inc., New York, 1969.

Both ANGCHK and SUPCHK are executed at each solution cycle. The smaller of the two incremental displacements computed in ANGCHK and SUPCHK is taken as the next incremental displacement during iteration.

## INPUT INSTRUCTIONS

### STRUCTURE TYPE IDENTIFICATION (15)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	MTYPE	Structure type identifier 1 = plane truss. 2 = plane frame. 3 = space truss. 4 = space frame.
6-10	NONLIN	Nonlinear solution control parameter Any positive integer causes the program to branch to the nonlinear solution.
11-15	ITMAX	Maximum number of iteration cycles in a given displacement step
16-20	MAXIT	Maximum number of displacement steps
21-30	PHMAX	Limit of relative counterclockwise rotation in degrees of two adjacent members 
31-40	PHMIN	Limit of relative clockwise rotation in degrees of two adjacent members 
41-50	DX	Size of displacement increment in inches
51-60	AMPLF	Amplification factor to be applied to vertical displacements if required for plotting

TITLE (20A4)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-80	TITLE	Any alphanumeric identifier

CONTROL PARAMETERS (6I5)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	NUMNP	Total number of nodes including supports
6-10	NBEAM	Total number of beam-type members
11-15	NSTRUT	Total number of two-force members
16-20	NUMAT	Number of different materials
21-25	NRNP	Number of constrained (supported) nodes
26-30	NMREL	Number of member releases

MATERIAL DATA (3E10.4)

Repeat NUMAT times (one card for each different material).

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-10	E(IM)	Young's modulus of elasticity
11-20	PR(IM)	Poisson's ratio
21-30	RHO(IM)	Material density (wt/unit vol.)

NODAL POINT COORDINATES (I5, 3E10.4)

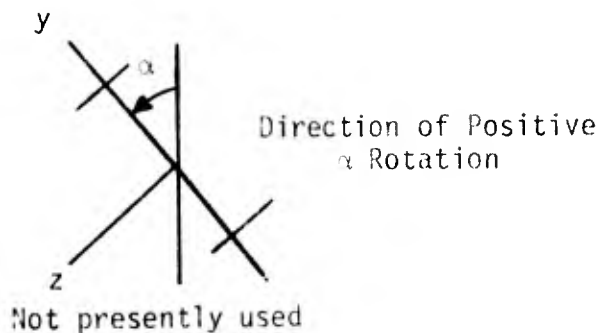
Repeat NUMNP times (one card for each nodal point including supports).

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	NN	Number of nodal point (must be less than NUMNP)
6-15	X(NN)	x-coordinate
16-25	Y(NN)	y-coordinate
26-35	Z(NN)	z-coordinate

MEMBER DESCRIPTIONS (4I5, 5E10.4, I5)

Repeat NUMEM times (one card for each member). NUMEM = NBEAM + NSTRUT.

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	ID	Member identification number (must be less than NBEAM + NSTRUT)
6-10	ND(ID,1)	Nodal point number - node i
11-15	ND(ID,2)	Nodal point number - node j
16-20	MD(ID)	Material identification number
21-30	XA(ID)	Cross sectional area ( $L^2$ )
31-40	ZI(ID)	Second moment of area about the local z-axis of the cross section ( $L^4$ )
41-50	YI(ID)	Second moment of area about the local y-axis of the cross section ( $L^4$ )
51-60	XJ(ID)	Second moment of area about the local x-axis of the cross section ( $L^4$ )
61-70	ALFA(ID)	Angle of rotation in degrees of the principal y-axis of the cross section with respect to a vertical plane



71-80      MLC(ID)

BOUNDARY CONDITIONS (I3, I1, 6I1, 6E10.4)

Repeat NRNP times (one card for each restrained node).

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-3	N	Number of restrained node
4	NTYPE(N)	Boundary condition type 0 or blank node is constrained against motion - no initial displacement specified. 1 is initial displacement specified.
5-10	IR(N,J)	Type of constraint identifier This is a six-digit integer, each digit of which is either 0 or 1 depending on whether motion in the corresponding direction is constrained or unconstrained. <u>Examples</u> For a completely fixed node, IR(N,J) = 111111. This specifies that translations and rotations are restrained in all possible directions. For a node restrained from translation in the global y-direction with all other motions free, IR(N,J) = 010000. For a node restrained from rotation in the global x-direction with all other motions free, IR(N,J) = 000100.
11-20	UI(N,1)	Specified initial translation in global x-direction (L)

21-30	UI(N,2)	Specified initial translation in global y-direction (L)
31-40	UI(N,3)	Specified initial translation in global z-direction (L)
41-50	UI(N,4)	Specified initial rotation about global x-axis (RAD)
51-60	UI(N,5)	Specified initial rotation about global y-axis (RAD)
61-70	UI(N,6)	Specified initial rotation about global z-axis (RAD)

If initial displacements are specified, a 1 must appear in the columns corresponding to the direction of the specified displacements.

#### MEMBER RELEASES (I3, 2I1)

These data are required for members with initially specified releases only. Repeat NMREL times (one card for each released member).

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-3	IM	Member number
4	MREL(IM,1)	Release code for node i of member as specified in member data
5	MREL(IM,2)	Release code for node j of member The release code is specified by a two-digit integer; either 1 or 0 is used depending on whether the i or j nodes of the member are released or not released, respectively.

#### LOADING PARAMETERS (3I5,3F10.4)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	NLND	Number of nodes with concentrated loads

6-10	NLMEM	Number of members having distributed loads along their length between nodes in addition to the member dead load
11-15	LGRAV	Member gravity load parameter If 1, member dead load generalized forces are computed for a g-force as specified below.
16-25	AX	Specified constant acceleration in the global x-direction (Not currently used)
26-35	AY	Specified constant g-acceleration in the global y-direction
36-45	AZ	Specified constant acceleration in the global z-direction (Not currently used)

#### CONCENTRATED NODAL LOADS (I5,6F10.4)

One card per loaded node for each loading condition is required. Cards are omitted if NLND is zero.

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	NL	Node number
6-15	P(1)	Component of concentrated force in global x-direction
16-25	P(2)	Force in global y-direction
26-35	P(3)	Force in global z-direction
36-45	P(4)	Component of concentrated couple about global x-axis
46-55	P(5)	Couple about global y-axis
56-65	P(6)	Couple about global z-axis

#### MEMBER LOADS

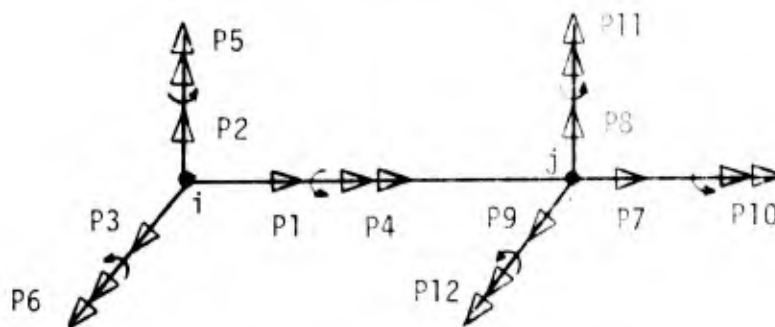
Two cards per loaded member (NLMEM members) are required for each loading condition involving members with intermediate loads. Cards are omitted if NLMEM is zero.

FIRST CARD (2I5,6F10.4)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5	MN	Member number
6-10	I	Node number
11-20	P(1)	Fixed-end axial force at node i
21-30	P(2)	Fixed-end y-shear at node i
31-40	P(3)	Fixed-end z-shear at node i
41-50	P(4)	Fixed-end x-moment (torque) at node i
51-60	P(5)	Fixed-end y-moment at node i
61-70	P(6)	Fixed-end z-moment at node i

SECOND CARD (5X, I5, 6F10.4)

<u>Columns</u>	<u>Variable Name</u>	<u>Entry</u>
1-5		Blank
6-10	J	Node number
11-20	P(7)	Fixed-end axial force at node j
21-30	P(8)	Fixed-end y-shear at node j
31-40	P(9)	Fixed-end z-shear at node j
41-50	P(10)	Fixed-end x-moment (torque) at node j
51-60	P(11)	Fixed-end y-moment at node j
61-70	P(12)	Fixed-end z-moment at node j



Directions of Positive  
Fixed-End Forces

## PROGRAM LISTING

	<u>Page</u>
BOOTSTRAP	36
PLOTXY	41
INPUT	41
STIFF	44
BEAM	45
LOAD	46
DLD	49
MERGE	50
BOUND	51
SOLVER	52
FORCE	54
MODIFY	55
GEOM	56
BM	58
ROTATE	59
ANGCHK	62
SUPCHK	64

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000091
00000092
00000093
00000094
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000195
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480

PROGRAM BOOTSTRAP (INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT)
*****
P R O G R A M   B O O T S T R A P

BOOTSTRAP IS A SPECIAL PURPOSE COMPUTER PROGRAM FOR THE
ANALYSIS OF THE NONLINEAR BEHAVIOR OF A TIED ARCH OF
ARBITRARY GEOMETRY BY THE DIRECT STIFFNESS METHOD.

R. L. JOHNSON
31 OCTOBER 1974

*****
COMMON/PAR/ MTYPE, NONLIN, NUMNP, NBEAM, NSTRUT, NUMEM, NUMAT, NRNP,
1 MBAND, NEQ, NQ, NUMEQ, PHMAX, PHMIN, LGRAV, ITER, LASTIT, KREL
COMMON/SLV/ A(110,30), B(110)
COMMON/MEM/ ND(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
1 XJ(200), ALFA(200), MREL(200,2), XL(200), FI(100,12)
COMMON/NCD/ X(100), Y(100), Z(100), NTYPE(100), IR(100,6), UI(100,6),
1 NR(100), JDISP, KITER, IUPDTE, IMOD
COMMON/MOD/ IREL(200,2), JREL(200,2), PHI(100), THETA(200),
1 XX(100), YY(100), ZZ(100), DX1, PHI1, JITER, ICONV, DX, IMEM,
2 PHIMX, PHIMN, JCLOSE, ICLOSE, LITER, YMIN, INOD, YY1, DX2

DOUBLE PRECISION A,B,FI,FID,PHI,THETA,XX,YY,ZZ.

INITIALIZE CONTROL PARAMETERS

NUMNP=1
NUMEM=1
NUMAT=1
NRNP=1
MBAND=1
READ (5,1000) MTYPE, NONLIN, ITMAX, MAXIT, PHMAX, PHMIN, DX, AMPLF
1000 FORMAT (4I5,5E10.4)
PHMIN=PHMIN+.05
IF (MTYPE.GT.1) GO TO 60
50 NEQ=6
NQ=3
GO TO 61
60 NEQ=12
NQ=6
61 CONTINUE

CALL INPUT
WRITE BAND WIDTH
WRITE(6,1050) MBAND
1060 FORMAT(/,13H BAND WIDTH #,15//)

```

```

00000490
00000500
00000510
00000520
00000530
00000535
00000536
00000540
00000541
00000542
00000543
00000544
00000545
00000546
00000547
00000548
00000549
00000550
00000551
00000552
00000553
00000554
00000555
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000675
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840

```

```

C
C
C
INITIALIZE ARRAYS
IPLOT=1
ITER=1
PHIMX=PHMAX+.05
PHIMN=PHMIN-.05
NITER=1
JITER=1
ICONV=0
PHI1=PHMIN
DX1=DX
KREL=NR(1)
KITER=1
IUPDTE=1
IMOD=1
JCLOSE=1
ICLOSE=1
LITER=1
DX2=DX
YY1=0.0
YMIN=0.0
DO 100 I=1,NUMEM
DO 100 J=1.2
IREL(IM,J)=MREL(IM,1)
JREL(IM,J)=MREL(IM,2)
NUMEQ=NUMNP*NG
DO 101 I=1,NUMEQ
B(I)=0.0
100 CONTINUE
DO 102 I=1,NUMNP
XX(IN)=X(IN)
YY(IN)=Y(IN)
ZZ(IN)=Z(IN)
CONTINUE
102 IF (IMOD.EQ.0.AND.IUPDTE.EQ.0) GO TO 71
DO 5 I=1,NUMEQ
DO 5 J=1,MHAND
A(I,J)=0.0
CONTINUE
5 CONTINUE
71 IF (MTYPE.NE.1) GO TO 6
DO 8 I=1,NUMNP
KENG*I
A(K,1)=1.0
8 CONTINUE
6 CONTINUE
IF (NGNLIN.GT.0) GO TO 80
75 CONTINUE
C
C
C
GENERATE STIFFNESS MAIRIX %AC
IF (IUPDTE.EQ.1.OR.IMOD.EQ.1) CALL STIFF

```

```

0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070

```

```

0071 GO TO 85
0072 IF (IUPDTE.EQ.1.OR.IMOD.EQ.1) CALL MODIFY
C
C
C
0073 IN=JDISP
0074 IF (ITER.EQ.1) GO TO 84
0075 NTYPE(IN)=0
0076 IR(IN,1)=0
0077 UI(IN,1)=0.0
0078 GO TO 81
0079 NTYPE(IN)=1
0080 IR(IN,1)=1
0081 IF (JITER.EQ.1) GO TO 83
0082 UI(IN,1)=DX1
0083 GO TO 81
0084 UI(IN,1)=DX
0085 CONTINUE
C
0086 IF (IUPDTE.EQ.1.OR.IMOD.EQ.1) CALL BOUND (1)
C
C
C
0087 REDUCE SYSTEM STIFFNESS MATRIX %K
C
C
C
0088 INITIALIZE LOAD VECTOR %K
0089 DO 45 I=1,NUMEQ
0090 B(I)=0.0
45 CONTINUE
C
C
C
0091 READ STRUCTURE LOADS
C
C
C
0092 COMPUTE NODAL DISPLACEMENTS
C
C
C
0093 CALL SOLVER(2)
C
C
C
0094 UPDATE TEMPORARY STRUCTURE GEOMETRY
C
C
C
0095 CALL GEOM (2)
C
C
C
0096 IF (ITER.EQ.1) GO TO 115
0097 IF (ITER.GT.1) GO TO 140
0098 WRITE (6,1075) JITER
0099 FORMAT (I10,37X,SHAFTER,I3,1X,25HITERATIONS STRUCTURE HAS STABILIZE
1075 /34X,5PHINCRMENT SUPP-JRT DISPLACEMENT AND CONTINUE SOLUTION//)
0099 WRITE (6,1035)
0100 FORMAT (I10,37X,44HS T R U C T U R E D I S P L A C E M E N T S//)
0101 WRITE(4,1070)

```

```

00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000971
00000972
00000973
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001220
00001230
00001240
00001250
00001260
00001345
00001350
00001360
00001365
00001370
00001380
00001390
00001400
00001410
00001420

```

```

0102 1070 FORMAT(2X,4HNODE,6X,6HU %INC,13X,6HV %INC,13X,6HW %INC,13X,
0103 12HTHETAX %PAD<,7X,12HTHETAY %RAD<,7X,12HTHETAZ %RAD</)
0104 DO 30 I=1,NUMNP
0105 N=NO*I-(NO-1)
0106 NN=NO*I
0107 WRITE (6,1025) I,(B(K),K=N,NN)
30 CONTINUE
C
C WRITE CURRENT STRUCTURE GEOMETRY
0108 WRITE (6,1110)
0109 1110 FORMAT (1H1,35X,49HU P D A T E D J O I N T C O O R D I N A T E
1 S//)
0110 DO 36 IN=1,NUMNP
0111 WRITE (6,1115) IN,XX(IN),YY(IN),ZZ(IN),PHI(IN)
0112 36 CONTINUE
0113 1115 FORMAT (25X,15,1X,4E16.6)
0114 37 CONTINUE
C
C WRITE MEMBER RELEASES
0115
0116 WRITE (6,1100)
0117 1100 FORMAT (1H0,52X,15HMEMBER RELEASES//)
0118 1105 FORMAT (6,1105) (IM,IREL(IM,2),JREL(IM,2),IM=1,NUMEM)
C
C COMPUTE MEMBER FORCES
0119 CALL FORCE
C
C WRITE MEMBER FORCES
0120 WRITE (6,1030)
0121 1030 FORMAT (1H1,47X,25HM E M B E R F O R C E S//)
0122 WRITE (6,2000)
0123 2000 FORMAT(7H MEMBER,3X,4HNODE,5X,8HFORCEXX<,9X,8HFORCEXY<,9X,
18HFORCEZK<,9X,9HMOMENTXX<,8X,9HMOMENTXY<,8X,9HMOMENTXZ</)
DO 120 ID=1,NUMEM
0124 I=ND(ID,1)
0125 J=ND(ID,2)
0126 WRITE (6,1080) ID,I,(FI(ID,LK),LK=1,NO)
0127 1080 FORMAT (/15,3X,15,6E17.7)
0128 NO1=NO+1
0129 WRITE (6,1085) J,(FI(ID,LK),LK=NO1,NEQ)
0130 1085 FORMAT (8X,15,6E17.7/)
0131 120 CONTINUE
C
C PLOT DEFORMED STRUCTURE
0132 CALL PLOTXY(IPL0T,AMPLF)
C
C INCREMENT LOADS AND CONTINUE SOLUTION
0133

```

```

00001960
00001970
00001980
00001990
00001995
00002000
00002010
00002020
00002025
00002026
00002027
00002028
00002029
00002030
00002031
00002032
00002033
00002034
00002035
00002040
00002045
00002046
00002047
00002050
00002060
00002070
00002075
00002080
00002085
00002086
00002087
00002088
00002089
00002090
00002091
00002092
00002093
00002094
00002095
00002096
00002097
00002098
00002100
00002101
00002102
00002103
00002110
00002112
00002120
00002130
00002135
00002140
00002150

```

```

IPL0T=IPLOT+1
IF (NITER*GE.MAXIT) GO TO 130
NITER=NITER+1
ITER=1
JITER=1
C
C      UPDATE STRUCTURE GEOMETRY
C
CALL GEOM (3)
C
DO 114 IM=1,NUMEM
  I=ND(IM,1)
  J=ND(IM,2)
  XI=X(J)-X(I)
  YI=Y(J)-Y(I)
  ZI=Z(J)-Z(I)
  XL(IM)=SQRT(XI**2+YI**2+ZI**2)
114 CONTINUE
IMOD=1
IUPDTE=1
ICONV=0
JCLOSE=0
ICLOSE=0
GO TO 70
C
115 CONTINUE
IF (ICONV*EQ.0.OR.IUPDTE*EQ.0) GO TO 116
CALL GEOM (1)
ITER=ITER+1
JITER=JITER+1
C
DO 117 IM=1,NUMEM
  I=ND(IM,1)
  J=ND(IM,2)
  XI=X(J)-X(I)
  YI=Y(J)-Y(I)
  ZI=Z(J)-Z(I)
  XL(IM)=SQRT(XI**2+YI**2+ZI**2)
117 CONTINUE
LITER=1
IUPDTE=1
GO TO 70
C
116 IF (JITER*GE.ITMAX) GO TO 125
JITER=JITER+1
GO TO 70
C
114 CONTINUE
JITER=JITER+1
LITER=LITER+1
GO TO 70
C

```

```

0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177

```

```

0178 1050 CONTINUE
0179 125 WRITE (6,1090)
0180 1090 FORMAT (46HMAXIMUM NUMBER OF ITERATIONS HAS BEEN EXCEEDED/
19HSOLUTION TERMINATED)
GO TO 135
0181 1025 FORMAT (15,6E12.7)
0182 130 WRITE (6,1120)
0183 1120 FORMAT (1H0.51X.17HSOLUTION COMPLETE//)
0184 135 CONTINUE
0185 CALL EXIT
0186 END
0187

0001 SUBROUTINE PLOTXY (IPLOT,AMPLF)
0002 COMMON/PAR/ MTYPE,NONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
1 MBAND,NEQ,NO,NUMEG,PHMAX,PHMIN,LGRAV,ITER,ITER,ITER,ITER,KREL
0003 COMMON /SLV/ A(110,30),B(110)
0004 COMMON/NOD/ X(100),Y(100),Z(100),NTYPE(100),IK(100,6),UI(100,6),
NR(100),JDISP,KITER,IUPDTE,IMOD
0005 1 COMMON/MOD/ IREL(200,2),JREL(200,2),PHI(100),THETA(200),
XK(100),YK(100),ZK(100),DXI,PHI1,JITER,ICONV,DX,IMEM,
PHIMX,PHIMN,JCLOSE,ICLOSE,LITER,YMIN,INOD,YI,DX2
2 DOUBLE PRECISION A,B,PHI,THETA,XX,YY,ZZ
DIMENSION XY(17,2)
DO 5 IN=1,NUMNP
K1=NO*(IN-1)+1
K2=NO*(IN-1)+2
XY(IN,1)= X(IN)+B(K1)
XY(IN,2)= Y(IN)+AMPLF*B(K2)
5 CONTINUE

C IPT=NUMNP
CALL PLOT (IPLOT,XY,IPT,2,120,1)

C RETURN
C FND

0001 SUBROUTINE INPUT
0002 COMMON/PAR/ MTYPE,NONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
1 MBAND,NEQ,NO,NUMEG,PHMAX,PHMIN,LGRAV,ITER,ITER,ITER,ITER,KREL
0003 COMMON /NOD/ X(100),Y(100),Z(100),NTYPE(100),IR(100,6),UI(100,6),
NR(100),JDISP,KITER,IUPDTE,IMOD
0004 1 COMMON/LOD/ NLND,NLMEM,AX,AY,AZ
0005 COMMON/MEW/ NO(200,2),MID(200),XA(200),ZI(200),YI(200),FID(100,12)
1 COMMON/MAT/ XJ(200),ALFA(200),MREL(200,2),XL(200),FI(100,12)
DIMENSION TITLE(20)

C DOUBLE PRECISION FI,FID
C READ AND WRITE CONTROL PARAMETERS

```

```

00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260

```

```

00002270
00002280
00002290
00002295
00002296
00002297
00002300
00002310
00002311
00002315
00002320
00002330
00002331
00002332
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420

```

```

00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570

```

```

0009 READ(5,950) TITLE
0010 FORMAT(20A4)
0011 WRITE(6,995) TITLE
0012 FORMAT(1H1,19X,20A4//)
0013 READ(5,1000) NUMNP,NBEAM,NSTRUT,NUMAT,NRNP,NMREL
0014 NUMEM=NBEAM*NSTRUT
0015 FORMAT(12I5)
0016 WRITE(6,1005)
0017 FORMAT(43X,34HC GNTRGLPARAME T E R S//)
0018 WRITE(6,1010) NUMNP,NUMEM,NUMAT,NRNP,NMREL
0019 FORMAT(42X,32HNUMBER OF NODAL POINTS #.15//
0020 42X,32HNUMBER OF MEMBERS #.15//
0021 42X,32HNUMBER OF MATERIALS #.15//
0022 42X,32HNUMBER OF BOUNDARY CONSTRAINTS #.15//
0023 42X,32HNUMBER OF MEMBER RELEASES #.15//)
0024
0025
0026
0027 READ AND WRITE OF MATERIAL CONSTANTS
C
C
C
1015 WRITE(6,1015)
1016 FORMAT(1H1,42X,34H M A T E R I A L C O N S T A N T S//)
1017 DO 10 IM=1,NUMAT
1018 READ(5,1020) E(IM),PH(IM),RHO(IM)
1019 G(IM)=.5*E(IM)/(1.+PH(IM))
1020 WRITE(6,1025) IM,E(IM),PH(IM),G(IM),RHO(IM)
1021 CONTINUE
1022 FORMAT(6E10.4)
1023 FORMAT(33X,15,4X,21HYOUNGS MODULUS #.E16.7//
1024 42X,21HPOISSONS RATIO #.E16.7//
1025 42X,21HSHEAR MODULUS #.E16.7//
1026 42X,21HMASS DENSITY #.E16.7//)
1027
1028
1029
1030 WRITE(6,1030)
1031 FORMAT(1H1,43X,32H J O I N T C O O R D I N A T E S//)
1032 WRITE(6,1031)
1033 FORMAT(35X,4HNODE,11X,1HX,15X,1HY,15X,1HIZ/
1034 48X,4H%INC,12X,4H%INC,12X,4H%INC//)
1035 DO 15 N=1,NUMNP
1036 READ(5,1120) NN,X(NN),Y(NN),Z(NN)
1037 FORMAT(15,3E10.4)
1038 WRITE(6,1035) NN,X(NN),Y(NN),Z(NN)
1039 FORMAT(33X,15,1X,3F10.4)
1040 CONTINUE
1041
1042 READ AND WRITE OF MEMBER PROPERTIES
C
C
C
1043 WRITE(6,1043)
1044 FORMAT(1H1,41X,30H M E M B E R D E F I N I T I O N S//)
1045 NHW=0
1046 WRITE(6,1101)
1047 FORMAT(2X,6HMEMBER,3X,4HLEFT,3X,5HRIGHT,5X,6HLENGTH,7X,4HAREA,3X,
1048 2HZ1,9X,2HY1,9X,2HX1,8X,4HJ1A,4X,8HJ1R1AL,4X,7HJ1D1N1J)

```

```

0002580
0002590
0002600
0002610
0002620
0002630
0002640
0002650
0002660
0002670
0002680
0002690
0002700
0002710
0002720
0002730
0002740
0002750
0002760
0002770
0002780
0002790
0002800
0002810
0002820
0002830
0002840
0002850
0002860
0002870
0002880
0002890
0002900
0002910
0002920
0002930
0002940
0002950
0002960
0002970
0002980
0002990
0003000
0003010
0003020
0003030
0003040
0003050
0003060
0003070
0003080
0003090
0003100

```

```

00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630

0043      2X,6HNUMBER,3X,4HNODE,4X,4HNODE,6X,4HIN<,6X,7HIN*2<,4X,
0044      7HIN**4<,4X,7HIN**4<,4X,7HIN**4<,6X,5HXRAD<,6X,4HTYPE,
          8X,4HCODE/)
          DO 20 II=1,NUMEM
          READ(5,1C40) ID,(ND(ID,K),K=1,2),MID(ID),XA(ID),ZI(ID),YI(ID),
1 1040  FORMAT(4I5,5F10.4,15)
          I=ND(ID,1)
          J=ND(ID,2)
          X1=X(I)
          X2=X(J)
          Y1=Y(I)
          Y2=Y(J)
          Z1=Z(I)
          Z2=Z(J)
          XL(ID)=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
          WRITE(6,1100) ID,(ND(ID,K),K=1,2),XL(ID),XA(ID),ZI(ID),YI(ID),
          XJ(ID),ALFA(ID),MID(ID)
1 1100  FORMAT(1X,15.2I8,F13.4,5F11.2,I8.3X,18)
          C
          C      COMPUTE HALF BAND WIDTH MBAND
          C
          C      IF(J-I) 35,40,45
          40  WRITE(6,1055) ID
          1055 FORMAT(42H IDENTICAL END NODAL POINTS FOR MEMBER NO.,I4)
          C      CALL EXIT
          35  IJ=NQ*J-NQ+1
          IJ=NQ*I
          GO TO 50
          45  IJ=NQ*I-NQ+1
          IJ=NQ*J
          50  NBD=IABS(JI-IJ)
          IF(NBD-NHW) 20,20,25
          25  NHW=IABS(JI-IJ)
          20  CONTINUE
          MBAND=NHW+1
          C
          C      READ BOUNDARY RESTRAINT CODES
          C
          C      DO 30 NN=1,NUMNP
          NTYPE(NN)=0
          DO 30 J=1,NQ
          UI(NN,J)=0.0
          30  IR(NN,J)=0
          WRITE(6,1135)
          1135  FORMAT(1H1,45X,24HBOUNDARY CONDITION CODES/)
          DO 60 NB=1,NRNP
          READ(5,1125) N,NTYPE(N),(IR(N,J),J=1,6),(UI(N,J),J=1,6)
          WRITE(6,1130) N,NTYPE(N),(IR(N,J),J=1,6),(UI(N,J),J=1,6)
          IF(NTYPE(N).EQ.1) JDISP=N
          60  CONTINUE
          1125  FORMAT(13,11,6I1,6E10.4)
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056

0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070

0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083

```

```

0084      1130 FORMAT (5X,I3,5X,I1,2X,6I1,5X,6E15,6)
C
C      READ AND WRITE MEMBER RELEASES
C
0085      DO 31 IM=1,NUMEM
0086      DO 32 J=1,2
0087      MREL(IM,J)=0
0088      31 CONTINUE
0089      WRITE (6,1060)
0090      1060 FORMAT (1H0,49X,20HMEMBER RELEASE CODES/)
0091      DO 33 MR=1,NMREL
0092      READ (5,1065) IM,(MREL(IM,J),J=1,2)
0093      WRITE(6,1070) IM,(MREL(IM,J),J=1,2)
0094      33 CONTINUE
0095      1065 FORMAT (I3,2I1)
0096      1070 FORMAT (53X,I5,5X,I1,2X,I1)
C
C      READ AND WRITE JOINT RELEASE SEQUENCE
C
0097      READ (5,1075) (NR(IN),IN=1,NUMNP)
0098      FORMAT (16I5)
0099      WRITE (6,1080)
0100      WRITE (6,1085) (NR(IN),IN=1,NUMNP)
0101      1080 FORMAT (1H0,39X,41HN O D E R E L E A S E S E Q U E N C E/)
0102      1085 FORMAT (20X,16I5)
C
C      READ LOADING CODES
C
0103      READ (5,1140) NLND,NLMEM,LGRAV,AX,AY,AZ
0104      1140 FORMAT (3I5,3E10,4)
C
0105      RETURN
0106      END
C
0001      SUBROUTINE STIFF
0002      COMMON/PAR/ MTYPE, NONLIN, NUMNP, NBEAM, NSTRUT, NUMEM, NUMAT, NRNP,
1      MBAND, NEQ, NQ, NUMEG, PHMAX, PHMIN, LGRAV, ITER, LASTIT, KREL
0003      COMMON/SLV/ A(110,30), B(110)
0004      COMMON/MEM/ IJ(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
1      COMMON/STIF/ S(12,12), R(3,3), T(12,12), ST(12,12), TF(12), P(12)
C
C      DOUBLE PRECISION A,B,S,R,T,ST,TF,FI,P,FID
0006      DO 5 ID=1,NUMEM
0007      I=IJ(ID,1)
0008      J=IJ(ID,2)
0009      N=ID
0010      CALL BEAM (N,I,J)
C
C      MERGE ELEMENT STIFFNESS MATRIX
C
0011      CALL MERGE (I,J,1)
0012      5 CONTINUE
0013      RETURN
0014      END
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003791
00003792
00003793
00003794
00003795
00003796
00003797
00003798
00003799
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
0004000
0004010
0004020
0004030
0004040
0004050
0004060
0004070
0004080

```

```

0001 SUBROUTINE BEAM (ID,I,J)
0002 COMMON/PAR/ MTYPE, NONLIN, NUMNP, NBEAM, NSTRUT, NUMEM, NUMAT, NRNP,
0003 MBAND, NEQ, NO, NUMEQ, PHMAX, PHMIN, LGRAV, ITER, LASTIT, KREL
0004 NR(100), X(100), Y(100), Z(100), NTYPE(100), IR(100,6), UI(100,6),
0005 IJ(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
0006 /MAT/ E(5), PR(5), G(5), RHO(5)
0007 /STIF/ S(12,12), R(3,3), T(12,12), SI(12,12), TF(12), P(12)
0008 DOUBLE PRECISION S,S1,S2,S3,S4,FI,P,R,T,SI,TF,FID
0009
0010 GENERATE ELEMENT STIFFNESS MATRIX %S<
0011
0012 M=MID(ID)
0013 YM=E(M)
0014 SM=G(M)
0015 S1=XA(ID)*YM/XL(ID)
0016 S2=SM*XJ(ID)/XL(ID)
0017 S3=YM*ZI(ID)/XL(ID)
0018 S4=YM*YI(ID)/XL(ID)
0019
0020 DU 10 II=1,NEQ
0021 DO 10 JJ=1,NEQ
0022 S(II,JJ)=0.0
0023 IF (NEQ-12) 11,12,12
0024
0025 11 S(1,1)=S1
0026 S(1,4)=-S1
0027 S(4,1)=-S1
0028 S(4,4)=S1
0029 GO TO 14
0030
0031 12 S(1,1)=S1
0032 S(1,7)=-S1
0033 S(2,2)=12.*S3/(XL(ID)**2)
0034 S(2,6)=6.*S3/XL(ID)
0035 S(2,9)=-S(2,2)
0036 S(2,12)=S(2,6)
0037 S(3,3)=12.*S4/(XL(ID)**2)
0038 S(3,5)=-6.*S4/XL(ID)
0039 S(3,9)=-S(3,3)
0040 S(3,11)=S(3,5)
0041 S(4,4)=S2
0042 S(4,10)=-S2
0043 S(5,5)=4.*S4
0044 S(5,9)=6.*S4/XL(ID)
0045 S(5,11)=2.*S4
0046 S(6,6)=4.*S3
0047 S(6,8)=-6.*S3/XL(ID)
0048 S(6,12)=2.*S3
0049 S(7,7)=S1
0050 S(8,8)=12.*S3/(XL(ID)**2)
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500

```

00004620  
 00004630  
 00004640  
 00004650  
 00004660  
 00004670  
 00004680  
 00004690  
 00004700  
 00004710  
 00004720  
 00004730  
 00004740  
 00004750  
 00004760  
 00004770  
 00004780  
 00004790  
 00004800  
 00004810  
 00004820  
 00004830

S(8,12)=-6.\*S3/XL(ID)  
 S(9,9)=12.\*S4/(XL(ID)\*\*2)  
 S(9,11)=6.\*S4/XL(ID)  
 S(10,10)=S2  
 S(11,11)=4.\*S4  
 S(12,12)=4.\*S3

SYMMETRIZE %SK

00 15 II=I,NEQ  
 00 15 JJ=II,NEQ  
 15 S(JJ,II)=S(II,JJ)

TRANSFORM %SK TO STRUCTURE COORDINATES

N=ID  
 14 CALL ROTATE (1,I,J,N)

80 CONTINUE

RETURN  
 END

00004840  
 00004850  
 00004860  
 00004870  
 00004880  
 00004890  
 00004900  
 00004910  
 00004920  
 00004930  
 00004940  
 00004950  
 00004960  
 00004965  
 00004970  
 00004980  
 00004990  
 00005000  
 00005010  
 00005020  
 00005030  
 00005040  
 00005050  
 00005060  
 00005070  
 00005080  
 00005090  
 00005100  
 00005110  
 00005120

SUBROUTINE LOAD  
 COMMON/PAR/ MTYPE,NONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,  
 MBAND,NEQ,NQ,NUMEG,PHMAX,PHMIN,LGRAY,ITER,LASTIT,KREL  
 1 COMMON/SLV/ A(110,30),B(110)  
 COMMON/MEM/ IJ(200,2),MID(200),XA(200),ZI(200),YI(200),FID(100,12)  
 COMMON/MEM/ XJ(200),ALFA(200),MREL(200,2),XL(200),FI(100,12)  
 1 COMMON/MAT/ E(5),PR(5),G(5),RHC(5)  
 COMMON /NDD/ X(100),Y(100),Z(100),NTYPE(100),IR(100,6),UI(100,6),  
 NR(100),JDISP,KITER,IUPDTE,IMGD  
 1 COMMON/LOD/ NLND,NLME,AX,AY,AZ  
 COMMON/STIF/ S(12,12),R(3,3),I(12,12),SI(12,12),IF(12),P(12)  
 COMMON/MOD/ IREL(200,2),JREL(200,2),PHI(100),THETA(200),  
 COMMON/MOD/ XX(100),YY(100),ZZ(100),DXI,PHI1,JITER,ICONV,DX,IMEM,  
 PHIMX,PHIMN,JCLOSE,LITER,YMIN,INDY,Y1,DX2  
 1 2 DIMENSION U(12)

DOUBLE PRECISION A,B,S,R,I,SI,IF,U,FI,P,FID,PHI,THETA,XX,YY,ZZ

INITIALIZE

NDI=NDI+1  
 U(1) S K=I,NUMEM  
 00 5 LE=I,NEQ  
 P(L)=0.  
 U(L)=0.  
 FI(K,L)=0.  
 FID(K,L)=0.  
 CONTINUE

0043  
 0044  
 0045  
 0046  
 0047  
 0048

0049  
 0050  
 0051

0052  
 0053

0054  
 0055  
 0056

0001  
 0002

0003  
 0004

0005  
 0006

0007  
 0008  
 0009

0010

0011  
 0012  
 0013  
 0014  
 0015  
 0016  
 0017  
 0018

```

0019      IF(NLND) 420,420,360
C
C      READ AND WRITE CONCENTRATED NODAL LOADS
C
0020      360 WRITE(6,361)
0021      361 FURMAT(1H1,37X,45HC 0 N C E N T R A T E D J O I N T L O A D S///
          116X,5HJOINT,6X,8HFORCEXX<,6X,8HFORCEXY<,6X,8HFORCEXZ<,7X,
          29HMOMENTXX<,5X,9HMOMENTXY<,5X,9HMOMENTXZ</)
C
0022      DO 405 L=1,NLND
0023      READ(5,1120) NL,(P(LK),LK=1,NQ)
0024      FORMAT(15,6F10.4)
0025      WRITE(6,362) NL,(P(LK),LK=1,NQ)
0026      362 FURMAT(14X,15.2X,3F14.4,3F14.2)
C
C      COMPUTE LOAD VECTOR %BK
C
0027      DO 405 KK=1,NQ
0028      K=NQ*NL-(NQ-KK)
0029      B(K)=P(KK)
0030      405 CONTINUE
C
C      READ AND/OR COMPUTE AND WRITE MEMBER LOADS
C
0031      420 IF(LGRAV.EQ.0.AND.NLMEM.EQ.0) GO TO 490
0032      IF (LGRAV.EQ.0) GO TO 425
C
C      MEMBER FORCES DUE TO GRAVITY
C
0033      DO 300 IM=1,NBEAM
0034      M=MID(IM)
0035      W=AY*XA(IM)*RHO(M)
0036      I=IJ(IM,1)
0037      J=IJ(IM,2)
0038      IF (ITER.EQ.1) GO TO 6
0039      XP=XX(J)-XX(I)
0040      GO TO 7
0041      XP=X(J)-X(I)
0042      SP=XL(IM)
0043      DO 10 I1=1,NEG
0044      10 P(I1)=0.0
C
0045      IF (IREL(IM,2).EQ.0.AND.JREL(IM,2).EQ.0) GO TO 100
0046      IF (IREL(IM,2).GT.0.AND.JREL(IM,2).GT.0) GO TO 105
0047      IF (IREL(IM,2).GT.0.AND.JREL(IM,2).EQ.0) GO TO 110
0048      IF (IREL(IM,2).EQ.0.AND.JREL(IM,2).GT.0) GO TO 115
C
0049      100 CALL DLD (1,W,XP,SP)
0050      GO TO 120
0051      105 CALL DLD (2,W,XP,SP)
0052      GO TO 120
0053      110 CALL DLD (3,W,XP,SP)
0054      GO TO 120
0055      115 CALL DLD (4,W,XP,SP)

```

```

0005130
0005140
0005150
0005160
0005170
0005180
0005190
0005200
0005210
0005220
0005230
0005240
0005250
0005260
0005270
0005280
0005290
0005300
0005310
0005320
0005330
0005340
0005350
0005360
0005370
0005380
0005390
0005400
0005410
0005420
0005430
0005440
0005450
0005460
0005470
0005480
0005490
0005500
0005510
0005520
0005530
0005540
0005550
0005560
0005570
0005580
0005590
0005600
0005610
0005620
0005630
0005640
0005650
0005660

```

00005670  
 00005680  
 00005690  
 00005700  
 00005710  
 00005720  
 00005730  
 00005740  
 00005750  
 00005760  
 00005770  
 00005780  
 00005790  
 00005800  
 00005810  
 00005820  
 00005830  
 00005840  
 00005850  
 00005860  
 00005870  
 00005880  
 00005890  
 00005900  
 00005910  
 00005920  
 00005930  
 00005940  
 00005950  
 00005960  
 00005970  
 00005980  
 00005990  
 00006000  
 00006010  
 00006020  
 00006030  
 00006040  
 00006050  
 00006060  
 00006070  
 00006080  
 00006090  
 00006100  
 00006110  
 00006120  
 00006130  
 00006140  
 00006150  
 00006160  
 00006170  
 00006180  
 00006190  
 00006200

```

0056      120 CONTINUE
0057      DO 305 JJ=1,NEQ
0058      FI(IM,JJ)=P(JJ)
0059      305 TF(JJ)=-P(JJ)

0060      C
0061      C      MERGE GRAVITY LOADS
0062      C
0063      C      CALL MERGE (I,J,2)
0064      C
0065      300 CONTINUE
0066      IF (NLMEM.EQ.0) GO TO 490
0067      DO 423 L=1,NLMEM
0068      READ (5,1121) MN,I,(P(LM),LM=1,NQ)
0069      1121 FORMAT (2I5,6F10.4)
0070      READ (5,1122) J,(P(LM),LM=NQ1,NEQ)
0071      1122 FORMAT (5X,15,6E10.4)
0072      GO TO (410,411,410,411), MTYPE
0073      410 P(4)=P(7)
0074      P(5)=P(8)
0075      P(6)=P(9)
0076      411 WRITE(6,601) MN,I,(P(LK),LK=1,NQ)
0077      601 FORMAT(10X,15,3X,15,3F14.4,3F14.2)
0078      WRITE(6,602) J,(P(LK),LK=NQ1,NEQ)
0079      602 FORMAT(18X,15,3F14.4,3F14.2)
0080      C
0081      C      TRANSFORM LOADS TO STRUCTURE AXES
0082      C
0083      C      CALL ROTATE (2,I,J,MN)
0084      C
0085      DO 20 II=1,NEQ
0086      FI(MN,II)=TF(II)
0087      TF(II)=-TF(II)
0088      20 CONTINUE
0089      C
0090      C      MERGE FORCES DUE TO MEMBER LOADS
0091      C
0092      C      CALL MERGE (I,J,2)
0093      C
0094      423 CONTINUE
0095      C
0096      C      IMPOSE INITIAL DISPLACEMENT BOUNDARY CONDITIONS
0097      C
0098      490 IF (ITER.GT.1) GO TO 500
0099      DO 424 L=1,NBEAM
0100      I=J(L,1)
0101      J=J(L,2)
0102      IF (NTYPE(I).LT.1) GO TO 30
0103      DO 35 JJE=1,NQ
0104      U(JJ)=UI(I,JJ)
0105      35 IF(NTYPE(I).LT.1.AND.NTYPE(J).LT.1) GO TO 25
0106      DO 40 JJE=1,NQ
0107      KK=JJ+NQ
0108      U(KK)=UI(J,JJ)
0109      40 U(KK)=UI(J,JJ)

```

```

0094      N=L
0095      IF ( IREL(L,2).EQ.0.AND.JREL(L,2).EQ.0) GO TO 41
0096      CALL BM(N,I,J)
0097      GO TO 42
0098      41 CALL BEAM (N,I,J)
0099      42 DO 50 JJ=1,NEQ
0100          FID(L,JJ)=0.0
0101          DO 45 KK=1,NEQ
0102              45 FID(L,JJ)=FID(L,JJ)+S(JJ,KK)*U(KK)
0103              TF(JJ)=-FID(L,JJ)
0104          50 CONTINUE
0105      C          COMPUTE LOAD VECTOR %BK
0106      C          CALL MERGE (I,J,2)
0107      C          25 CONTINUE
0108      C          424 CONTINUE
0109      C          500 CONTINUE
0110      C          CALL BOUND(2)
0111      C          RETURN
0112      C          END
0001      SUBROUTINE OLD (ILD,W,XP,SP)
0002      COMMON/STIF/ S(12,12),R(3,3),T(12,12),ST(12,12),TF(12),P(12)
0003      DOUBLE PRECISION S,R,T,ST,TF,P
0004      GO TO (5,10,15,20), ILD
0005      C          FIXED - FIXED ENDS
0006      C          5 P(2)=W*SP/2.
0007      C          P(6)=W*SP*XP/12.
0008      C          P(8)=P(2)
0009      C          P(12)=-P(6)
0010      C          GO TO 25
0011      C          HINGE - HINGE ENDS
0012      C          10 P(2)=0.5*W*SP
0013      C          P(8)=P(2)
0014      C          GO TO 25
0015      C          HINGE LEFT END
0016      C          15 P(12)=-.125*W*SP*XP
0017      C          P(2)=0.375*W*SP
0018      C          P(8)=0.625*W*SP

```

```

00006210
00006220
00006230
00006240
00006250
00006260
00006270
00006280
00006290
00006300
00006310
00006320
00006330
00006340
00006350
00006360
00006370
00006380
00006390
00006400
00006410
00006420
00006430
00006440
00006450
00006460
00006470
00006480
00006490
00006500
00006510
00006520
00006530
00006540
00006550
00006560
00006570
00006580
00006590
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730

```

00006740  
 00006750  
 00006760  
 00006770  
 00006780  
 00006790  
 00006800  
 00006810  
 00006820  
 00006830

GO TO 25

HINGE RIGHT END

C  
 C  
 C

20 P(6)=0.125\*\*SP\*\*XP  
 P(2)=0.625\*\*SP\*\*SP  
 P(8)=0.375\*\*SP\*\*SP  
 25 CONTINUE  
 RETURN  
 END

0015  
 0016  
 0017  
 0018  
 0019  
 0020  
 0021

00006840  
 00006850  
 00006860  
 00006870  
 00006880  
 00006890  
 00006900  
 00006910  
 00006920  
 00006930  
 00006940  
 00006950  
 00006960  
 00006970  
 00006980  
 00006990  
 00007000  
 00007010  
 00007020  
 00007030  
 00007040  
 00007050  
 00007060  
 00007070  
 00007080  
 00007090  
 00007100  
 00007110  
 00007120  
 00007130  
 00007140  
 00007150  
 00007160  
 00007170  
 00007180  
 00007190  
 00007200  
 00007210  
 00007220  
 00007230

SUBROUTINE MERGE (I,J,IMERGE)  
 COMMON/PAR/ MTYPE,NDNLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,  
 MBAND,NEQ,NQ,NUMEQ,PHMAX,PHMIN,LGRAV,ITER,LASTIT,KREL  
 1 COMMON/SLV/ A(110,30),B(110)  
 COMMON/STIF/ S(12,12),R(3,3),T(12,12),ST(12,12),TF(12),P(12)  
 DIMENSION NDF(12)

DOUBLE PRECISION A,B,S,P,T,ST,TF,P  
 GO TO (5,16), IMERGE

FORM STRUCTURE STIFFNESS MATRIX

C  
 C  
 C  
 C  
 C

5 DO 2 K=1,N2  
 KK=K+NQ  
 NDF(K)=NQ\*I-(NQ-K)  
 NDF(KK)=NO\*J-(NO-K)  
 2 CONTINUE

0007  
 0008  
 0009  
 0010  
 0011

DO 15 II=1,NEQ  
 K1=NDF(II)  
 DO 15 JJ=1,NEQ  
 K2=NDF(JJ)  
 IF(K2-K1) 15,10,10  
 10 K3=K2-K1+1  
 A(K1,K3)=A(K1,K3)+S(II,JJ)  
 15 CONTINUE  
 GO TO 25

C  
 C  
 C  
 C  
 C

FORM STRUCTURE LOAD VECTOR

C  
 C  
 C

16 DO 20 II=1,NQ  
 K=NO\*I-(NO-II)  
 KK=NO\*J-(NO-II)  
 INQ=II+NO  
 B(K)=B(K)+TF(II)  
 B(KK)=B(KK)+TF(INQ)  
 20 CONTINUE  
 25 CONTINUE  
 RETURN  
 END

0021  
 0022  
 0023  
 0024  
 0025  
 0026  
 0027  
 0028  
 0029  
 0030

```

0001 SUBROUTINE BOUND(IBND)
0002 COMMON/PAR/ MTYPE,NONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
0003 MBAND,NEQ,NQ,NUMEG,PHMAX,PHMIN,LGRAY,ITER,LASTIT,KREL
0004 COMMON/SLV/ A(110,30),B(110)
1 COMMON /NOD/ X(100),Y(100),Z(100),NTYPE(100),IR(100,6),UI(100,6),
1 NR(100),JDISP,KITER,IUPUTE,IMOD
C
C DOUBLE PRECISION A,B
C GO TO (20,25), IBND
20 NHW=MBAND-1
DO 60 N=1,NUMNP
C
C COMPUTE CONTROL COUNTERS
C
C
DO 5 KK=1,NQ
K=NQ*N-(NQ-KK)
JI=K-NHW
IF(JI) 10,10,15
10 II=1
15 II=JI
C
C REDUCE ROW OF %K
C
30 IF (IR(N,KK).EQ.0) GO TO 5
35 A(K,1)=1.0
DO 40 J=1,NHW
L=J+1
40 A(K,L)=0.0
C
C REDUCE COLUMN OF %K
C
IF(K-1) 5,5,50
50 JJ=K-1
DO 55 J=1, JJ
KI=K-J+1
55 A(J,KI)=0.0
5 CONTINUE
60 CONTINUE
GO TO 85
C
C REDUCE %K
C
25 DO 80 N=1,NUMNP
65 DO 70 KK=1,NQ
K=NQ*N-(NQ-KK)
C
IF (IR(N,KK).EQ.0) GO TO 70
75 B(K)=UI(N,KK)
70 CONTINUE
80 CONTINUE
85 CONTINUE

```

```

00007240
00007250
00007260
00007270
00007280
00007290
00007300
00007310
00007320
00007330
00007340
00007350
00007360
00007370
00007380
00007390
00007400
00007410
00007420
00007430
00007440
00007450
00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610
00007620
00007630
00007640
00007650
00007660
00007670
00007680
00007690
00007700
00007710
00007720
00007730
00007740
00007750
00007760

```



```

0028      500 IF(IHPIG) 600,610,600
C
C      SINGULAR MATRIX
C
0029      600 WRITE(6,602) IHPIG
0030      602 FORMAT(1X,3HSINGULAR MATRIX AT REDUCTION NRED #,I4)
0031      CALL EXIT
0032      CONTINUE
0033      GO TO 700

C
C      REDUCE THE RIGHT HAND SIDES
C
0034      300 CONTINUE
0035      NRED=0
0036      301 IF(NRED+1-N) 302,401,401
0037      302 NRED=NRED+1

C
C      DIVIDE ROW BY SQUARE ROOT OF DIAGONAL ELEMENT
C
0038      B(NRED)=B(NRED)/A(NRED,1)

C
C      REDUCE REMAINING BLOCK OF NUMBERS
C
0039      DO 351 I=1,NHW
0040      L=NRED+1
0041      IF(L-N) 311,311,351
0042      B(L)=B(L)-A(NRED,1+1)*B(NRED)
0043      CONTINUE
0044      GO TO 301

C
C      BACK SUBSTITUTION
C
0045      401 B(N)=B(N)/A(N,1)
0046      N1=N-1
0047      DO 451 I=1,N1
0048      I=N-I
0049      SUM=0.0
0050      DO 421 JJ=1,NHW
0051      M=JJ+1
0052      IF(N-M) 451,421,421
0053      SUM=SUM+A(I,JJ+1)*B(M)
0054      451 B(I)=(B(I)-SUM)/A(I,1)
0055      25 CONTINUE
0056      700 CONTINUE

C
C      RETURN
C      END
0057
0058

```

```

00008270
00008280
00008290
00008300
00008310
00008320
00008330
00008340
00008350
00008360
00008370
00008380
00008390
00008400
00008410
00008420
00008430
00008440
00008450
00008460
00008470
00008480
00008490
00008500
00008510
00008520
00008530
00008540
00008550
00008560
00008570
00008580
00008590
00008600
00008610
00008620
00008630
00008640
00008650
00008660
00008670
00008680
00008690
00008700
00008710
00008720
00008730

```

```

0001 SUBROUTINE FORCE
0002 COMMON/PAR/ MTYPE, NONLN, NUMNP, NBEAM, NSTRUT, NUMEM, NUMAT, NRNP,
0003 MBAND, NEG, NO, NUMEQ, PHMAX, PHMIN, LGRV, ITER, LASTIT, KREL
0004 COMMON/SLV/ A(110,30), B(110)
0005 COMMON /MOD/ X(100), Y(100), Z(100), NTYPE(100), IR(100,6), UI(100,6),
0006 NK(100), JUTSP, KITER, IUPDTE, IMOD
0007 COMMON/MEM/ IJ(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
0008 .XJ(200), ALFA(200), MREL(200,2), XL(200), FI(100,12)
0009 COMMON /LOD/ NLND, NLMEM, AX, AY, AZ
0010 COMMON /MAT/ E(5), PR(5), G(5), RHC(5)
0011 COMMON/STIF/ S(12,12), R(3,3), T(12,12), ST(12,12), TF(12), P(12)
0012 COMMON/MOD/ IREL(200,2), JREL(200,2), PHI(100), THETA(200),
0013 XX(100), YY(100), ZZ(100), DX1, PHI1, JITER, ICONV, DX, IMEM,
0014 PHIMX, PHIMN, JCLOSE, ICLOSE, LITER, YMIN, INOD, Y1, DX2
0015 DIMENSION U(12), F(12), NDF(12)
0016 DOUBLE PRECISION A,B,S,R,T,ST,TF,U,F,I,P,FID,PHI,THETA,XX,YY,ZZ
0017 DO 5 ID=1,NUMEM
0018 COMPUTE CONTROL COUNTERS
0019 I=IJ(ID,1)
0020 J=IJ(ID,2)
0021 DU 2 K=1,NQ
0022 KK=K+NO
0023 NDF(K)=NO*I-(NO-K)
0024 NDF(KK)=NO*J-(NO-K)
0025 2 CONTINUE
0026 DO 10 I1=1,NEQ
0027 I1=NDF(I1)
0028 U(I1)=B(I1)
0029 10 CONTINUE
0030 COMPUTE MEMBER FORCES FOR STRUCTURE AXES
0031 NEID
0032 IF (IREL(ID,2),EQ,0,AND,JREL(ID,2),EQ,0) GO TO 11
0033 CALL BM (N,I,J)
0034 GO TO 12
0035 CALL BEAM (N,I,J)
0036 DO 15 I1=1,NEQ
0037 F(I1)=0,0
0038 DO 15 JJ=1,NEJ
0039 F(I1)=F(I1)+S(I1,JJ)*U(JJ)
0040 15 CONTINUE
0041 MEMID(ID)
0042 A*FAY*XA(ID)*RND(M)
0043 XP=XX(JJ)-XX(I)
0044 SP=XL(I)
0045 DO 16 I1=1,NEG
0046 16 CONTINUE
0008740
0008750
0008760
0008770
0008780
0008790
0008800
0008810
0008820
0008830
0008840
0008850
0008860
0008865
0008870
0008880
0008890
0008900
0008910
0008920
0008930
0008940
0008950
0008960
0008970
0008980
0008990
0009000
0009010
0009020
0009030
0009040
0009050
0009060
0009070
0009080
0009090
0009100
0009110
0009120
0009130
0009140
0009150
0009160
0009170
0009180
0009190
0009200
0009210
0009220
0009230
0009240
0009250

```

```

0038 P(II)=0.0
0039 16 CONTINUE
0040 IF (IREL(ID,2).EQ.0.AND.JREL(ID,2).EQ.0) GO TO 30
0041 IF (IRFL(ID,2).GT.0.AND.JREL(ID,2).GT.0) GO TO 35
0042 IF (IREL(ID,2).GT.0.AND.JREL(ID,2).EQ.0) GO TO 40
0043 IF (IREL(ID,2).EQ.0.AND.JREL(ID,2).GT.0) GO TO 45
0044 CALL DLD (1,W,XP,SP)
0045 GO TO 50
0046 CALL DLD (2,W,XP,SP)
0047 GO TO 50
0048 CALL DLD (3,W,XP,SP)
0049 GO TO 50
0050 CALL DLD (4,W,XP,SP)
0051 50 CONTINUE
0052 DO 55 JJ=1,NEQ
0053 FI(ID,JJ)=P(JJ)
0054 55 CONTINUE
0055 DO 20 II=1,NEQ
0056 P(II)=F(II)+FI(ID,II)
0057 20 CONTINUE
0058 C
0059 C TRANSFORM FORCES TO MEMBER AXES
0060 C
0061 C
0062 C
0063 C
0064 C
0065 C
0009260
0009270
0009280
0009290
0009300
0009310
0009320
0009330
0009340
0009350
0009360
0009370
0009380
0009390
0009400
0009410
0009420
0009430
0009440
0009450
0009460
0009470
0009480
0009490
0009500
0009510
0009520
0009530
0009540
0009550
0009560
0009570
0009580
0009590
0009600
0009610
0009620

```

```

00009630
00009640
00009650
00009660
00009670
00009680
00009690
00009700
00009710
00009720
00009725
00009730
00009740
SUBROUTINE MODIFY
COMMON/PAR/ MTYPE,NUNLN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
M BAND,NEQ,NO,NUMEQ,PHMAX,PHMIN,LGRAV,ITER,LASTIT,KREL
1 COMMON/SLV/ A(110,30),B(110)
COMMON/NOD/ X(100),Y(100),Z(100),NTYPE(100),IR(100,6),UI(100,6),
NR(100),JDISP,KITER,IUPDTF,IMCD
1 COMMON/MEM/ ID(200,2),MID(200),XA(200),ZI(200),YI(200),FID(100,12)
1 COMMON/MOD/ XJ(200),ALFA(200),MREL(200,2),XL(200),FI(100,12)
1 COMMON/MOD/ IREL(200,2),JREL(200,2),PHI(100),THETA(200),
XX(100),YY(100),ZZ(100),DX1,PHI1,JITER,ICONV,DX,IMEM,
PHIMX,PHIMN,JCLOSE,ICLOSE,LITER,YMIN,INOD,Y1,DX?,
2 COMMON/STIF/ S(12,12),R(3,3),T(12,12),ST(12,12),P(12)
COMMON/MAT/ E(5),PR(5),G(5),RHO(5)
0001
0002
0003
0004
0005
0006
0007
0008

```



00010230  
00010240  
00010245  
00010250  
00010260  
00010270  
00010280  
00010290  
00010300  
00010310  
00010350  
00010360  
00010365  
00010370  
00010380  
00010390  
00010400  
00010410  
00010420  
00010430  
00010440  
00010450  
00010460  
00010470  
00010480  
00010485  
00010490  
00010500  
00010510  
00010520  
00010530  
00010540  
00010550  
00010560  
00010570  
00010580  
00010590  
00010600  
00010690  
00010700  
00010701  
00010702  
00010703  
00010705  
00010710  
00010715  
00010720  
00010730  
00010733  
00010738  
00010739

```

0023 Z(I)=Z2(I)
0024 5 CONTINUE
0025 IF (IG.NE.2) GO TO 50

0026 DO 10 IM=1,NBEAM
0027 I=ID(IM,1)
0028 J=ID(IM,2)

0029 DY=YY(J)-YY(I)
0030 DX=XX(J)-XX(I)
0031 IF (DX.EQ.0.) GO TO 15
0032 THETA(IM)=DATAN2(DY,DX)
0033 THETA(IM)=ATAN2(DY,DX)
0034 GO TO 10
0035 IF (DY) 20,20,25
0036 20 THETA(IM)=-90./57.2957795
0037 GO TO 10
0038 25 THETA(IM)=90./57.2957795
10 CONTINUE

C
C COMPUTE MEMBER ANGLE CHANGES
C IF (JITER.GT.1) GO TO 31
C DO 30 IN=1,NUMNP
C PHI(IN)=PHMIN
30 CONTINUE
31 NBM1=NBEAM-1
C DO 35 IM=1,NBM1
C J=ID(IM,2)
C PHI(J)=(THETA(IM+1)-THETA(IM))*57.2957795
35 CONTINUE

C
C CHECK FOR JOINT OPENING AND CLOSING
C CALL ANGCHK
C CHECK FOR NEGATIVE OR ZERO Y-COORDINATES
C IF (PHI(KREL).GT.PHMIN.AND.PHI(KREL).LT.PHMAX) CALL SUPCHK
C WRITE (6,1000) IITER,JITER,IUPDIE,IMOD,KREL,PHI(KREL),PHI1,DX1,
1 INOD,YY(INOD),Y1,DX2
1000 FORMAT (5I5,3E15.4,2X,I3,3E15.4)
C DO 60 IM=1,NUMEM
C IREL(IM,1)=IREL(IM,2)
C JREL(IM,1)=JREL(IM,2)
60 CONTINUE

C 50 CONTINUE
C RETURN
C END

```

```

0001 SUBROUTINE BM (IM,I,J)
0002 COMMON/PAR/ MTYPE,MONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
0003 COMMON/MEM/ MBAND,NEQ,NO,NUMEQ,PHMAX,PHMIN,LGRAY,ITER,LASTIT,KREL
0004 COMMON/MOD/ XJ(200),ALFA(200),MREL(200),XL(200),FI(100),
0005 COMMON/STIF/ S(12,12),R(3,3),T(12,12),ST(12,12),P(12)
0006 COMMON/MAT/ E(5),PR(5),G(5),RHO(5)
0007 DOUBLE PRECISION A,B,S,S1,S2,S3,S4,S5,FI,P,R,T,ST,TF,FID,
0008 PHI,THETA,XX,YY,ZZ
0009 M=MID(IM)
0010 YM=E(M)
0011 SM=G(M)
0012
0013 COMPUTE STIFFNESS COEFFICIENTS
0014 S1=XA(IM)*YM/XL(IM)
0015 S2=SM*XJ(IM)/XL(IM)
0016 S3=YM*ZI(IM)/XL(IM)
0017 S4=YM*YI(IM)/XL(IM)
0018 S5=3.*YM*ZI(IM)/(XL(IM)**3)
0019
0020 DO 45 II=1,NEQ
0021 DO 45 JJ=1,NEQ
0022 S(II,JJ)=0.0
0023
0024 45 CONTINUE
0025
0026 S(1,1)=S1
0027 S(1,7)=-S1
0028 S(3,3)=12.*S4/(XL(IM)**2)
0029 S(3,5)=-6.*S4/XL(IM)
0030 S(3,9)=-S(3,3)
0031 S(3,11)=S(3,5)
0032 S(4,4)=S2
0033 S(4,10)=-S2
0034 S(5,5)=4.*S4
0035 S(5,9)=6.*S4/XL(IM)
0036 S(5,11)=2.*S4
0037 S(7,7)=S1
0038 S(9,9)=12.*S4/(XL(IM)**2)
0039 S(9,11)=6.*S4/XL(IM)
0040 S(10,10)=S2
0041 S(11,11)=4.*S4
0042 IF (IREL(IM,2).GT.0.AND.JREL(IM,2).GT.0) GO TO 65
0043 IF (IREL(IM,2).GT.0.AND.JREL(IM,2).EQ.0) GO TO 55
0044 IF (IREL(IM,2).EQ.0.AND.JREL(IM,2).GT.0) GO TO 60
0045
0046 HINGE LEFT MODIFICATION
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500

```

```

00011370
00011380
00011390
00011400
00011410
00011420
00011460
00011470
00011480
00011490
00011500
00011510
00011520
00011530
00011540
00011550
00011600
00011610
00011620
00011630
00011640
00011650
00011660
00011670
00011680
00011690
00011700
00011710
00011720
00011730
00011740
00011750

00011760
00011770
00011780
00011790
00011800
00011810
00011820
00011830
00011840
00011850
00011855
00011860
00011890
00011900
00011910
00011920
00011930
00011940

55 S(2,2)=S5
   S(2,3)=-S5
   S(2,12)=S5*XL(IM)
   S(3,8)=S5
   S(8,12)=-S(2,12)
   S(12,12)=S(2,12)*XL(IM)
   GO TO 65
C
C
C
   HINGE RIGHT MODIFICATION
60 S(2,2)=S5
   S(2,6)=S5*XL(IM)
   S(2,8)=-S5
   S(6,6)=S(2,6)*XL(IM)
   S(6,8)=-S(2,6)
   S(8,8)=S5
C
C
C
   SYMMETRIZE %S<
65 DO 70 II=1,NEQ
   DO 70 JJ=II,NEQ
   S(JJ,II)=S(II,JJ)
   70 CONTINUE
C
C
C
   TRANSFORM %S< TO STRUCTURE COORDINATES
N=IM
CALL ROTATE (1,1,J,N)
75 CONTINUE
RETURN
END

SUBROUTINE ROTATE (NRGT,I,J, ID)
COMMON/PAP/ MTYPE,NONLIN,NUMNP,NBEAM,NSTRUT,NUMEM,NUMAT,NRNP,
1 MBRND,NEQ,NO,NUMEG,PHMAX,PHMIN,LGRAV,ITER,LASTIT,KREL
COMMON/NOJ/ X(100),Y(100),Z(100),NTYPE(100),IR(100,6),UI(100,6),
1 NR(100),JDISP,KITER,IUPDTE,IMCO
COMMON/MEM/ IJ(200,2),MID(200),XA(200),ZI(200),YI(200),FID(100,12)
1 XJ(200),ALFA(200),MREL(200,2),XL(200),FI(100,12)
COMMON/STIF/ S(12,12),R(3,3),T(12,12),ST(12,12),TF(12),P(12)
COMMON/NOJ/ IREL(200,2),JREL(200,2),PHI(100),THETA(200),
1 XX(100),YY(100),ZZ(100),DXI,PHI1,JITER,ICONV,DX,IMEM,
2 PHIMX,PHIMN,JCLOSE,LITER,YMIN,INUD,YY1,DX2
C
C
C
DOUBLE PRECISION S,R,I,ST,TF,CX,CY,CZ,SINA,COSA,RAD,ALPHA,FI,
1 P,FID,PHI,THETA,XX,YY,ZZ
C
C
C
   COMPUTE MEMBER DIRECTION COSINES
IF (ITER.EQ.1) GO TO 5

```

```

0038
0039
0040
0041
0042
0043
0044

0045
0046
0047
0048
0049
0050

0051
0052
0053
0054

0055
0056

0057

0058
0059

0001
0002
0003
0004
0005
0006

0007

```



```

0051      R(3,J)=(CY*CZ*SINA+CX*COXA)/RAD
0052      DO 40 K=1,3
0053      K1=K+3
0054      K2=K+6
0055      K3=K+9
0056      DO 40 L=1,3
0057      T(K,L)=R(K,L)
0058      LI=L+3
0059      T(K1,L1)=R(K,L)
0060      L2=L+6
0061      T(K2,L2)=R(K,L)
0062      L3=L+9
0063      T(K3,L3)=R(K,L)
0064      CONTINUE
0065
0066      GO TO (100,200,300), NROT
0067
0068      TRANSFORM XSK TO STRUCTURE COORDINATES
0069
0070      DO 60 II=1,NEQ
0071      DO 60 JJ=1,NEQ
0072      ST(II,JJ)=0.0
0073      DO 65 KK=1,NEQ
0074      ST(II,JJ)=ST(II,JJ)+S(II,KK)*T(KK,JJ)
0075
0076      GO TO 80
0077
0078      TRANSFORM LOADS TO STRUCTURE AXES
0079
0080      DO 205 II=1,NEQ
0081      TF(II)=0.0
0082      DO 110 JJ=1,NEQ
0083      TF(II)=TF(II)+T(JJ,II)*P(JJ)
0084      CONTINUE
0085
0086      GO TO 80
0087
0088      TRANSFORM FORCES TO MEMBER AXES
0089
0090      DO 305 II=1,NEQ
0091      TF(II)=0.0
0092      DO 305 JJ=1,NEQ
0093      TF(II)=TF(II)+T(II,JJ)*P(JJ)
0094      CONTINUE
0095      RETURN
0096      END

```

```

00012480
00012490
00012500
00012510
00012520
00012530
00012540
00012550
00012560
00012570
00012580
00012590
00012600
00012610
00012620
00012630
00012640
00012650
00012660
00012670
00012680
00012690
00012700
00012710
00012720
00012730
00012740
00012750
00012760
00012770
00012780
00012790
00012800
00012810
00012820
00012830
00012840
00012850
00012860
00012870
00012880
00012890
00012900
00012910
00012920
00012930
00012940
00012950
00012960
00012970
00012980
00012990
00013000
00013010

```

```

0001  SUBROUTINE  ANGCHK
0002  COMMON/PAR/  MTYPE, NONLIN, NUMNP, NSEAM, NSTRUT, NUMEM, NUMAT, NRNP,
0003  COMMON/SLV/  A(110,30), B(110)
0004  COMMON/NOE/  X(100), Y(100), Z(100), NTYPE(100), IR(100,6), UI(100,6),
0005  COMMON/NOE/  NR(100), JDISP, KITER, IUPDTE, IMOD
0006  COMMON/MEW/  ID(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
0007  COMMON/MOD/  XJ(200), ALFA(200), JREL(200,2), MREL(200,2), XL(200), FI(100,12)
0008  COMMON/MOD/  IREL(200,2), JREL(100), DXI, PHI(100), THETA(200),
0009  COMMON/MOD/  XX(100), YY(100), ZZ(100), DXI, PHI(100), JITER, ICONV, DX, IMEM,
0010  COMMON/MOD/  PHIMX, PHIMN, JCLOSE, ICLOSE, LITER, YMIN, INOD, YI, DX2
0011  DOUBLE PRECISION  A,B,FI,FID,PHI,THETA,XX,YY,ZZ,DY,DX
0012  COMPUTE MEMBER RELEASES
0013  DO 36 IM=1,NUMEM
0014  J=ID(IM,2)
0015  IF (J.EQ.KREL) IMEM=IM
0016  36 CONTINUE
0017  IM=IMEM
0018  INITIALIZE ARRAYS
0019  I=ID(IM,1)
0020  J=ID(IM,2)
0021  LASTIT=1
0022  IMODE=0
0023  IUPDTE=0
0024  IF (J.NE.KREL) WRITE (6,1005) IM
0025  1005 FORMAT (18H0J-NODE FOR MEMBER,13,1X,40HDOES NOT MATCH DESIGNATED
0026  RELEASED JOINT)
0027  IF (J.NE.KREL) CALL EXIT
0028  IF (PHI(J).LE.PHMIN.AND.JITER.EQ.1) JCLOSE=1
0029  IF (PHI(J).LE.PHMIN) GO TO 47
0030  IF (PHI(J).GE.PHMAX.AND.IITER.EQ.1) GO TO 45
0031  IF (PHI(J).GE.PHMAX.AND.IITER.GT.1) GO TO 47
0032  IF (JCLOSE.EQ.1.OR.ICLOSE.EQ.1) GO TO 43
0033  JREL(IM,2)=1
0034  IREL(IM+1,2)=1
0035  IF (JREL(IM,2).EQ.1.OR.IREL(IM+1,2).EQ.1) IR(KREL,6)=1
0036  IF (JITER.GT.1.AND.JCLOSE.EQ.1) LASTIT=0
0037  IF (IITER.GT.1) LASTIT=0
0038  IF (IITER.EQ.1) LASTIT=1
0039  ICONV=1
0040  PHI=PHI(J)
0041  IF (IITER.EQ.1) IUPDTE=1
0042  IF (JREL(IM,2).NE.JREL(IM,1).OR.IREL(IM+1,2).NE.IREL(IM+1,1))
0043  I=IMODE+1
00013015
00013020
00013025
00013030
00013035
00013040
00013045
00013050
00013055
00013060
00013065
00013100
00013200
00013300
00013400
00013500
00013600
00013700
00013800
00013900
00014000
00014100
00014200
00014300
00014400
00014500
00014600
00014700
00014800
00014900
00015000
00015100
00015200
00015300
00015400
00015500
00015600
00015700
00015800
00015900
00016000
00016100
00016200
00016300
00016400
00016500
00016600
00016650
00016700
00016800
00016900
00017000
00017100

```

00017200  
00017300  
00017400  
00017500  
00017600  
00017700  
00017800  
00017900  
00018000  
00018100  
00018200  
00018300  
00018500  
00018600  
00018700  
00018800  
00018900  
00019000  
00019100  
00019200  
00019300  
00019400  
00019500  
00019600  
00019700  
00019800  
00019900  
00020000  
00020100  
00020200  
00020300  
00020400  
00020500  
00020600  
00020700  
00020800  
00020900  
00021000  
00021100  
00021200  
00021300  
00021400  
00021500  
00021600  
00021700  
00021800  
00021900  
00022000  
00022100  
00022200  
00022300  
00022400  
00022500

```

C
GO TO 40
45 IF (PHI(J).GT.PHIMX.OR.PHI(J).LT.PHIMN) GC TO 46
ICONV=1
IUPDTE=1
JREL(IM,2)=0
IREL(IM+1,2)=0
IR(KREL,6)=0
PHI1=PHI(J)
KITER=KITER+1
KREL=NR(KITER)
DC 44 IM=1,NUMEM
J=ID(IM,2)
IF (J.EQ.KREL) IMEM=IM
44 CONTINUE
IM=IMEM
IF (PHI(KREL).LE.PHMIN.OR.PHI(KREL).GE.PHMAX) IR(KREL,6)=0
IF (PHI(KREL).LE.PHMIN.OR.PHI(KREL).GE.PHMAX) GO TO 42
JREL(IM,2)=1
IREL(IM+1,2)=1
IR(KREL,6)=1
PHI1=PHI(KREL)
42 IMOD=1
LASTIT=1
DX1=DXX
GO TO 40

C
46 IMOD=0
IUPDTE=0
ICONV=0
IF (JITER.GT.1) GO TO 48
FXL=PHI1-PHMAX
FXR=PHI(J)-PHMAX
X2=DXX
X1=0.0
GO TO 55

C
48 PHTSI=(PHI(J)-PHMAX)*FXL
IF (PHTSI.LT.0.) GO TO 49
FXL=PHI(J)-PHMAX
X1=DX1
GO TO 55

C
49 FXR=PHI(J)-PHMAX
X2=DX1

C
55 DX1=(X1+FXR-X2*FXL)/(FXR-FXL)
IF (JITER.GT.10) DX1=DX1*(1.-0.01*JITER)
LASTIT=1
GO TO 40

C
47 IF (LITER.EQ.1) ICLOSF=1
JREL(IM,2)=0

```

0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
0080  
0081

```

0082 IREL(IM+1,2)=0
0083 IF (JREL(IM,2).EQ.JREL(IM,1).AND.IREL(IM+1,2).EQ.IREL(IM+1,1))
0084   GO TO 51
0085 IR(KREL,6)=9
0086 IUPDTE=0
0087 LASTIT=1
0088 IMODE=1
0089 PHII=PHI(J)
0090 ICONV=0
0091   GO TO 40
0092
0091 51 IUPDTE=1
0092 IF (ITER.EQ.1) LASTIT=1
0093 IF (ITER.GT.1) LASTIT=0
0094 IMODE=0
0095 PHII=PHI(J)
0096 ICONV=1
0097
0097 40 CONTINUE
0098 RETURN
0099 END
0001
0002 SUBROUTINE SUPCHK
0003 COMMON/PAR/ MTYPE, NONLIN, NUMNP, NBEAM, NSTRUT, NUMEM, NUMAT, NRP,
0004   MBEAM, NEG, NQ, NUMEG, PHMAX, PHMIN, LGRAV, ITER, LASTIT, KREL
0005 COMMON/SLV/ A(110,30), R(110)
0006 COMMON/NDI/ X(100), Y(100), Z(100), NTYPE(100), IR(100,6), UI(100,6),
0007   NR(100), JDISP, KITER, IUPDTE, IMOD
0008 COMMON/MEW/ ID(200,2), MID(200), XA(200), ZI(200), YI(200), FID(100,12)
0009 COMMON/MOD/ XJ(200), ALFA(200), MREL(200,2), XL(200), FI(100,12)
0010 IREL(200,2), JREL(200,2), PHI(100), THETA(200),
0011   XX(100), YY(100), ZZ(100), DXI, PHI1, JITER, ICONV, DXX, IMEM,
0012   PHIMX, PHIMN, JCLOSE, ICLUSE, LITER, YMIN, INDO, YI, DX2
0013 DOUBLE PRECISION A,B,FI,FID,PHI,THETA,XX,YY,ZZ,DY,DX
0014
0015   FIND MINIMUM Y-COORDINATE
0016
0017 INDO=1
0018 YMIN=0.0
0019 YMX=0.05
0020 YMN=0.05
0021 DO 15 IN=1, NUMNP
0022 IF (YY(IN).EQ.0) OR (IR(IN,2).EQ.1) GO TO 15
0023 IF (YY(IN).LT.YMIN) INDO=IN
0024 IF (YY(IN).LT.YMIN) YMIN=YY(IN)
0025 15 CONTINUE
0026
0027   CHECK FOR NEGATIVE OR ZERO Y-COORDINATES
0028
0029 IF (YMIN.GT.0) OR (INDO.EQ.1) GO TO 60
0030   GO TO IREL, NBEAM
0031   JREL(IN,2)
0032
0033 6001 0022600
0034 00022700
0035 00022800
0036 00022900
0037 00023000
0038 00023100
0039 00023200
0040 00023300
0041 00023400
0042 00023500
0043 00023600
0044 00023700
0045 00023800
0046 00023900
0047 00024000
0048 00024100
0049 00024200
0050 00024300
0051 00024700
0052 00025200
0053 00025300
0054
0055 00030000
0056 00030100
0057 00030200
0058 00030300
0059 00030400
0060 00030500
0061 00030600
0062 00030700
0063 00030800
0064 00030900
0065 00031000
0066 00031100
0067 00031200
0068 00031300
0069 00031400
0070 00031450
0071 00031500
0072 00031510
0073 00031520
0074 00031600
0075 00031750
0076 00031800
0077 00031900
0078 00032300
0079 00032400
0080 00032500
0081 00032600
0082 00032700
0083 00033200
0084 00033300
0085 00033400

```

```

0019 IF (J.EQ.INOD) IMEM=IM
0020 20 CONTINUE
0021 C IM=IMEM
0022 C
0023 IF (YMIN.LE.0...AND.ITER.EQ.1) GO TO 25
0024 IF (YMIN.LF.0...AND.ITER.GT.1) GO TO 30
0025 LASTIT=0
0026 ICONV=1
0027 YI=YMIN
0028 IUPDTE=1
0029 GO TO 60
0030 C
0031 25 IF (YMIN.LI.YMN) GO TO 40
0032 ICONV=1
0033 IUPDTE=1
0034 JREL(IM,2)=1
0035 IREL(IM+1,2)=1
0036 YI=YMIN
0037 IMOD=1
0038 LASTIT=1
0039 DX2=DX
0040 GO TO 60
0041 C
0042 30 YY(INOD)=0.0
0043 IR(INOD,2)=1
0044 IR(INOD,6)=1
0045 JREL(IM,2)=1
0046 IREL(IM+1,2)=1
0047 IUPDTE=0
0048 LASTIT=1
0049 IMOD=1
0050 ICONV=0
0051 YI=YMIN
0052 GO TO 60
0053 C
0054 35 IUPDTE=1
0055 LASTIT=0
0056 IMOD=0
0057 YI=YMIN
0058 ICONV=1
0059 GO TO 60
0060 C
0061 40 IMOD=0
0062 IUPDTE=0
0063 ICONV=0
0064 IF (JITER.GT.1) GO TO 45
0065 FXL=YI
0066 FXR=YMIN
0067 X2=DX
0068 X1=0.C
0069 GO TO 60
0070 C

```

```

00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500
00035600
00035700
00035800
00035900
00036000
00036100
00036150
00036200
00036300
00036600
00036700
00036800
00036900
00037000
00037100
00037200
00037300
00037400
00037500
00037600
00037700
00037800
00037900
00038000
00038100
00038200
00038300
00038400
00038500
00038600
00038700
00038800
00038900

```

00039000  
00039100  
00039200  
00039300  
00039400  
00039500  
00039600  
00039700  
00039800  
00039900  
00040000  
00040100  
00040200  
00040210  
00040300  
00040400  
00040500

```
45 YTEST=YMIN*FXL  
   IF (YTEST.LT.0) GO TO 50  
   FXL=YMIN  
   X1=DX2  
   GO TO 55  
C  
50 FXR=YMIN  
   X2=DX2  
C  
55 DX2=(X1*FXR-X2*FXL)/(FXR-FXL)  
   IF (JITER.GT.10) DX2=DX2*(1.-0.01*JITER)  
   LASTIT=1  
C  
   IF (DX2.LT.DX1) DX1=DX2  
60 CONTINUE  
   RETURN  
   END
```

0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079