SACKMAN

Report No. 69-19

STRUCTURES AND MATERIALS RESEARCH
DEPARTMENT OF CIVIL ENGINEERING

A COMPUTER PROGRAM FOR THE ANALYSIS OF TUBULAR K JOINTS

by
OJARS GRESTE

Report to
Standard Oil Company of California
San Francisco, California

NOVEMBER 1969

STRUCTURAL ENGINEERING LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY CALIFORNIA

A COMPUTER PROGRAM FOR THE ANALYSIS OF TUBULAR K JOINTS

by

Ojars Greste

University of California
Department of Civil Engineering
Berkeley, California

ABSTRACT

A program for the finite element analysis of tubular K joints is presented. The type of structure consists of two pipes intersecting a third; the type of analysis performed is static-linear-elastic.

The discretization of the structure into an assemblage of quadrilateral and triangular elements is carried out by the computer program. Four degrees of finite element mesh refinement are offered and the meshes can be plotted if suitable equipment is available.

A variety of boundary and loading conditions can be applied. The program provides displacements, moments and membrane forces, and surface stresses of the structure.

ACKNOWLEDGMENT

The work described in this report was carried out under Contract No. 442420-65974 between the University of California and the Standard Oil Company of California. The project was performed under the direction of Professor Ray W. Clough.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The structure for which this program was designed is of a type found in offshore drilling platforms. Typically such platforms are constructed out of tubular members. The structure considered here is a simple type of tubular interconnection consisting of two members intersecting a third in the form of a K.

This computer program provides a linear elastic static analysis of the K joint by means of the finite element method. The principal feature of the program is that the finite element discretization of the structure is performed by the program. Four degrees of refinement of the discretized structure are available. Input data for the program is reduced to a mimimum: the user need specify essentially only the overall geometry of the K joint, degree of refinement desired, material properties, loading and member end support conditions.

Bending and membrane stiffnesses are both considered in the analysis which provides as output the nodal displacements, stress resultants and surface stresses. If plotting equipment is available, a picture of the finite element mesh with nodes and elements numbered can be obtained.

The program was written by Ojars Greste (completed July 1969) and its theoretical development is presented in Reference (1). The analysis part of the program was based on a shell program by C. Philip Johnson (2). This report is a manual for the use of the K joint program.

2. FINITE ELEMENT DISCRETIZATION

2.1 Type of Structure

The general type of K joint that may be considered is shown in Fig. 2.1. The structure is symmetric with respect to the global Y-Z plane and in the analysis only one half of the joint is considered.

The following points should be noted.

i. The three intersecting members may have any lengths subject to certain minimum requirements which are described later.

ii. The angles of inclination θ_1, θ_2 of the branches to the chord may vary between zero and 90 degrees.

iii. The branches are not permitted to intersect each other. The radii of the two branches must be equal.

iv. The branch to chord diameter ratio d/D must be greater than zero and less than or equal to unity.

v. The thickness of the cylindrical surface which forms each member may vary over the pipe.

vi. The chord and branches must each have a uniform elastic modulus E and a uniform Poisson ratio ν throughout. However, values of E and ν for each of the three members may be different.

vii. Any consistent set of units may be used to describe the geometry, material properties and loading of the K joint.

2.2 Substructuring

In analyzing the structure of Fig. 2.1, the approach used is to divide the structure into a number of substructures. These are identified

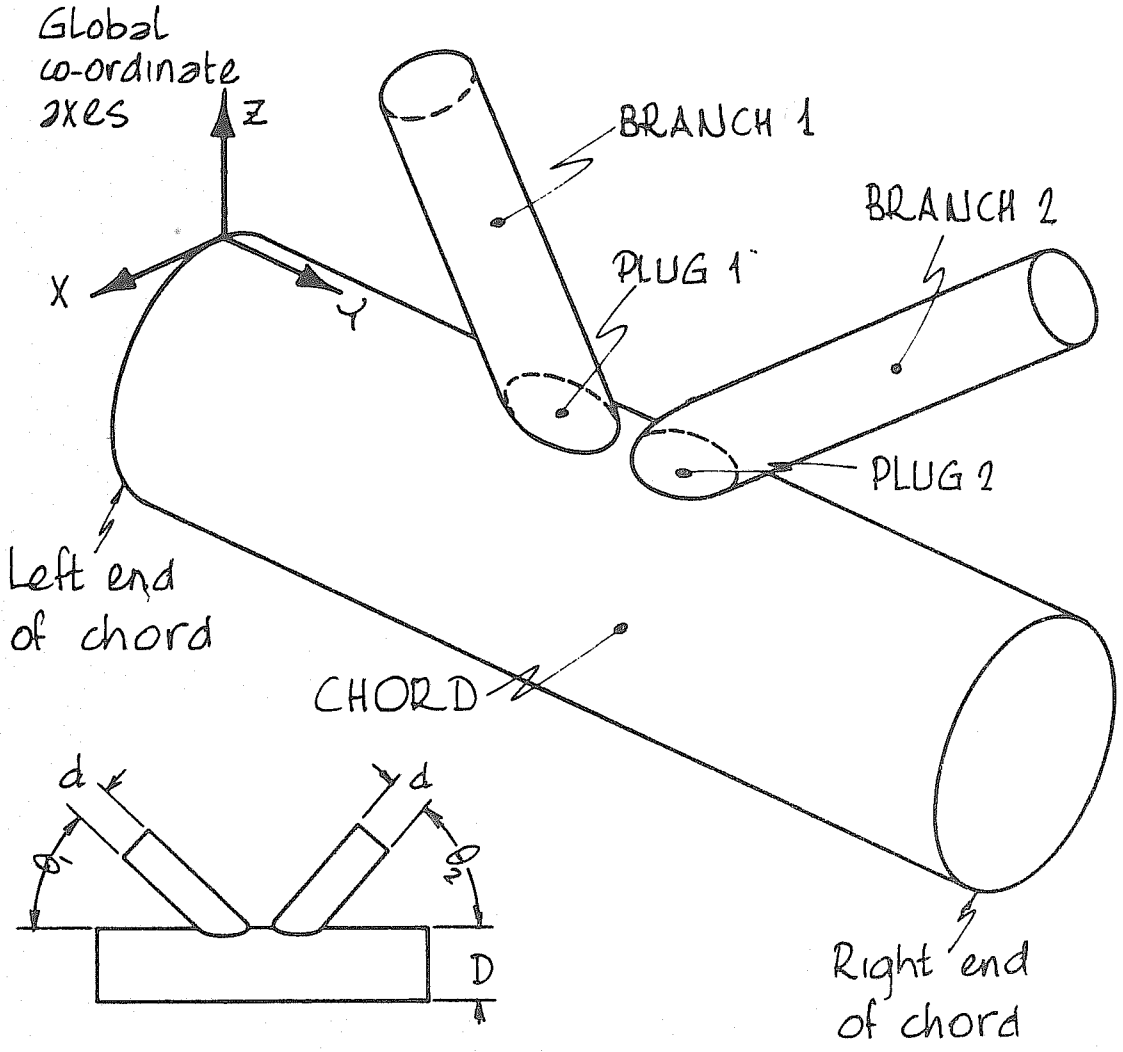


FIG. 2.1 TYPE OF K JOINT

in Fig. 2.1 by the labels, CHORD, PLUG1, PLUG2, BRANCH1 and BRANCH2. Of these five substructures, the chord is referred to as the main structure. Throughout this write-up the word "substructure" is at different times used to refer to all five of the above-mentioned components or just those which exclude the main structure. The context should make it clear which meaning is intended.

Not all substructures need be present in any one case. For example, the chord may have cut-outs in those areas bounded by the walls of the branches. For such a case, the user would indicate that the plugs are absent. Likewise, one or both of the branches may be absent. Note, however, that the chord must always be present. A structure consisting of, say, a single branch pipe is not permissible.

Within the program, each substructure is assigned a substructure number. The program always attempts to number the substructures in the order plug 1, branch 1, plug 2, branch 2, chord. If a particular substructure is absent, then the numbering algorithm will skip over that substructure and number the next one in the list. Thus, if all five substructures are present, plug 1, branch 1, plug 2, branch 2, chord are numbered 1, 2, 3, 4, 5, respectively. If, say, the two plugs are absent, then the three substructures branch 1, branch 2, chord are numbered 1, 2, 3, respectively. The numbers assigned to the substructures are printed in the program output. The user must note these numbers because the output (and part of the input data) of the program is grouped according to substructure number.

2.3 Finite Element Types

Two types of finite elements are available in the program, both of which include membrane and bending stiffness.

i. Triangular element.

Membrane stiffness - constant strain triangle.

Bending stiffness - fully compatible plate bending element after Hsieh, Clough and Tocher (3).

ii. Non-planar quadrilateral element.

Membrane stiffness - an assemblage of four linear strain triangles with linear displacements along exterior sides.

Bending stiffness - an assemblage of four bending elements as per (i) above.

The quadrilateral element has superior stiffness properties compared to the triangle. In the automatic discretization schemes, an effort was made to use the quadrilateral element wherever possible and to use triangles only where quadrilaterals could not be accommodated.

2.4 Mesh Types

Four degrees of mesh refinement are available to the user. Figs. 2.2, 2.3, 2.4 and 2.5 show examples of the four options starting with the coarsest mesh in Fig. 2.2 and progressing in refinement to the finest in Fig. 2.5. The developed surfaces of the tubes are pictured. The same joint configuration applies to all four figures.

For the purpose of numerically defining the structure, the nodes and elements of each substructure are numbered. Each substructure is numbered independently and separate numbering schemes are used to identify the nodes and the elements.

The degrees of refinement of the mesh types relative to the coarse mesh are, in terms of numbers of nodes, approximately 1.0, 2.0, 3.2, 4.3

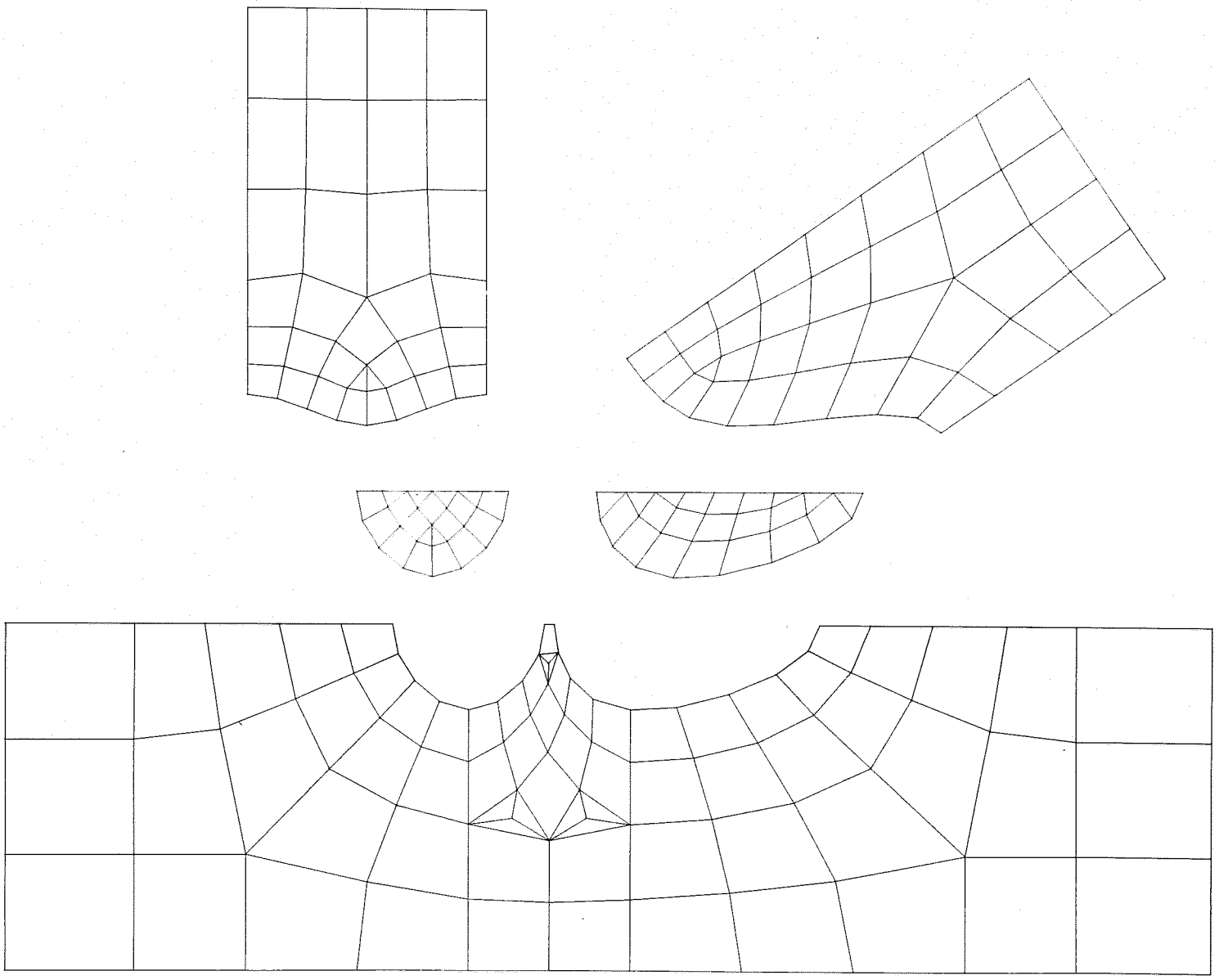


FIG. 2.2 EXAMPLE OF COARSE MESH

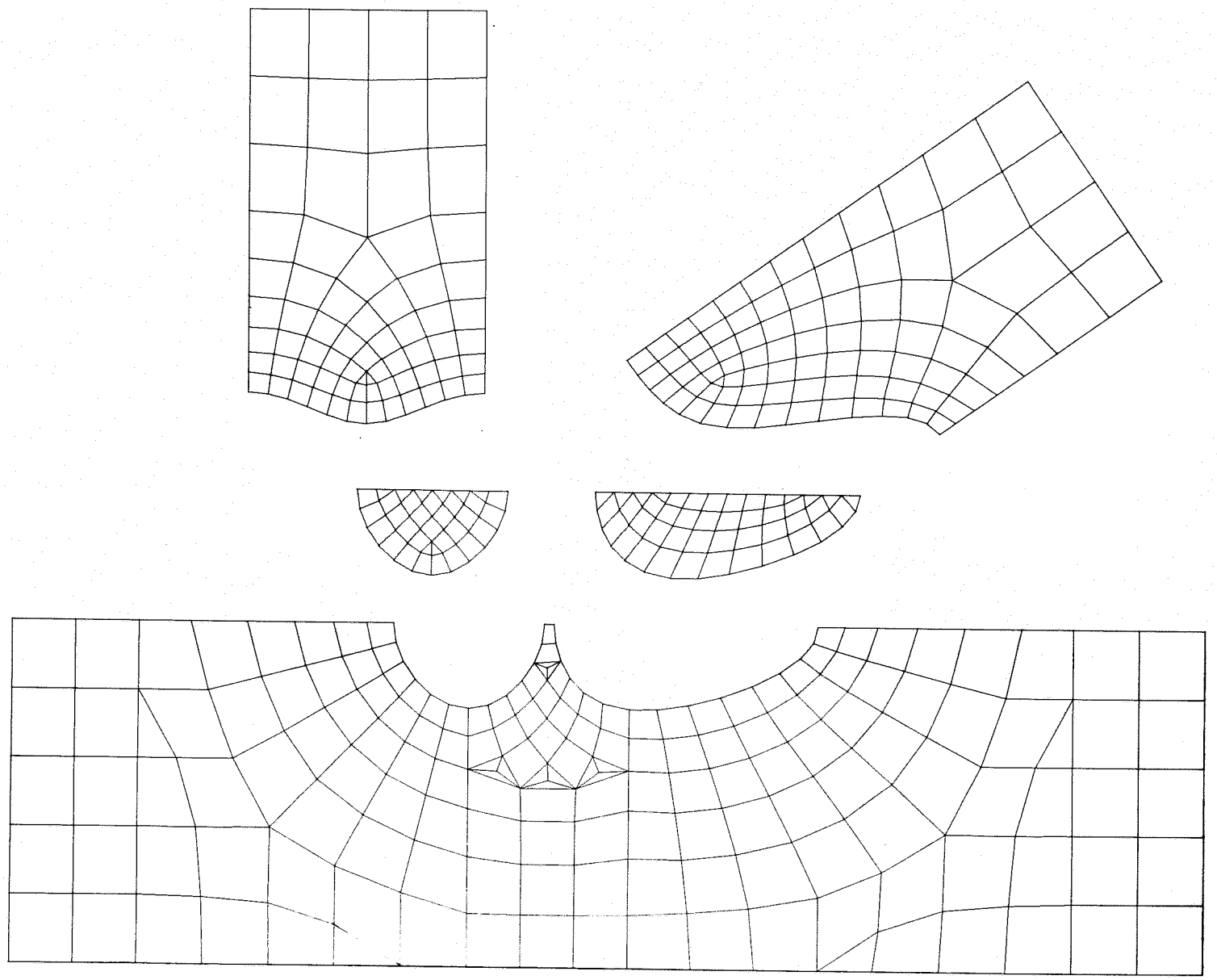


FIG. 2.3 EXAMPLE OF MEDIUM MESH

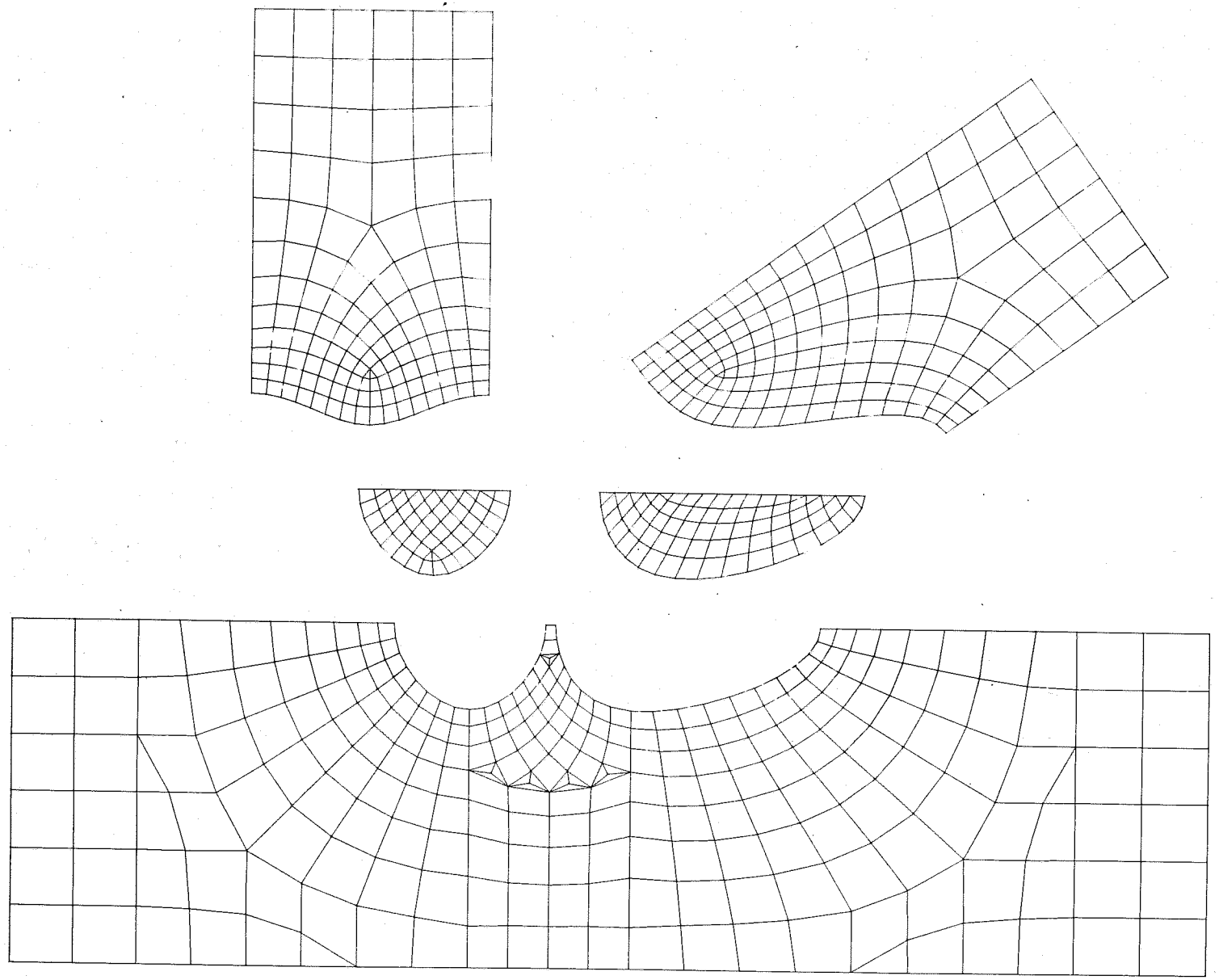


FIG. 2.4 EXAMPLE OF FINE MESH

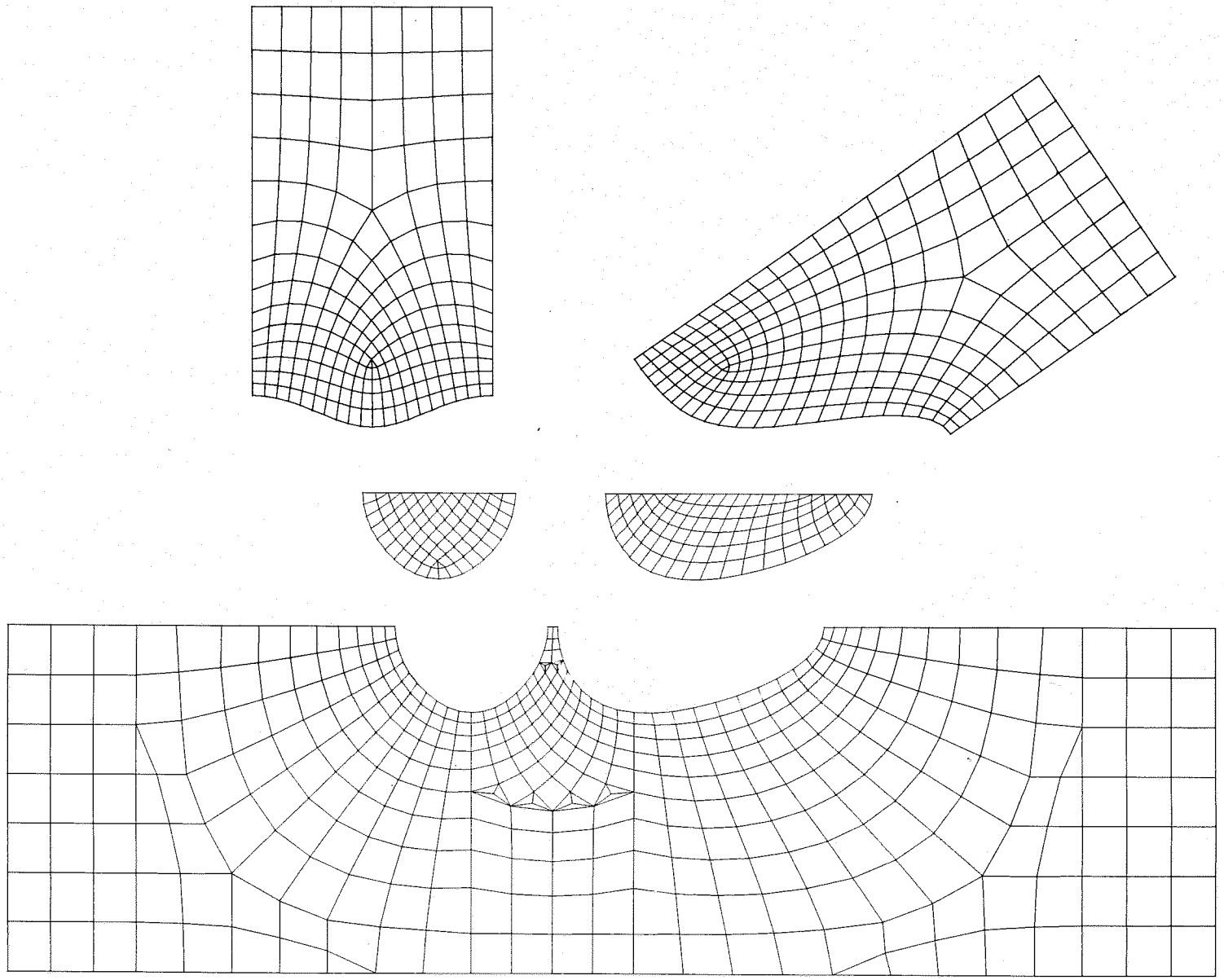


FIG. 2.5 EXAMPLE OF XFINE MESH

for the coarse, medium, fine, extra fine meshes, respectively. In terms of numbers of element, the relative degrees of refinement are approximately 1.0, 2.2, 3.7, 5.5. These ratios will vary slightly with the geometry of the joints.

The numbering scheme used for each substructure mesh causes the stiffness matrix for that substructure to be in banded form. The half band width (half because of symmetry of the stiffness matrix) is important because it is related to the time required for solution of the equations. To obtain the half band width (abbreviated as $\frac{1}{2}$ bw) for a substructure, the entire array of elements (for that substructure) is scanned and the maximum difference between the numbers of any two nodes associated with a particular quadrilateral is recorded. The $\frac{1}{2}$ bw is then given by one plus this number multiplied by five (the number of degrees of freedom per node). In each mesh type the $\frac{1}{2}$ bw is greatest for the main structure. The exact value of the $\frac{1}{2}$ bw will depend on the geometry of the joint in question. Approximate values of the $\frac{1}{2}$ bw for the main structure for the coarse, medium, fine, extra fine mesh types are 105, 150, 200, 250, respectively.

The meshes are generated following a set of basic patterns. These patterns were chosen according to two criteria: first, the elements should be graded in size from the smallest in the juncture region (the area of stress concentration) to the largest in regions away from the junctures; and second, all the quadrilateral elements should be as close as possible to being square in shape (a square being the most desirable* shape). For varying lengths of the chord and branches and varying θ_1,

---

*For the quadrilateral elements used in this program, the element shape which best represents the displacements and stresses in the structure has not been conclusively determined. However, investigations by Mr. Kaspar Willam of U. C. Berkeley suggest that square shaped elements are usually preferable to non-square elements.

θ_2 and d/D, the basic patterns were adjusted (by adding or subtracting elements) to satisfy as closely as possible the second criterion.

Because of the wide range of geometric configurations possible it was necessary, for the mesh patterns chosen, to place limits on the length, θ_1, θ_2 and d/D parameters in order to produce a satisfactory mesh - "satisfactory" in the sense of meeting the two above-mentioned criteria. With reference to Fig. 2.6, the following are approximate minimum dimensions and ranges of parameters θ_1, θ_2, d/D which should result in satisfactory element meshes.

i. $a \geq 1.5 * d$

$b \geq 1.5 * d$

$c \geq 1.5 * D$

$e \geq 1.5 * D$

ii. Approx. $30^\circ \leq \theta_1, \theta_2 \leq 90^\circ$

(with absolute limits $0^\circ < \theta_1, \theta_2 \leq 90^\circ$)

iii. Approx. $0.1 \leq d/D \leq$ approx. 0.95

(with absolute limits $0 < d/D \leq 1.0$)

For values of lengths, θ_1, θ_2 and d/D outside the above ranges the resulting meshes will not necessarily be overly distorted from what might be considered satisfactory. The user should decide for himself whether or not a mesh is acceptable before proceeding to the analysis.

In the course of generating the mesh the program computes minimum branch lengths and minimum values for the distances c and e (Fig. 2.6) which are most desirable to contain the types of mesh schemes being used. If the dimensions of the joint don't satisfy these computed minimum values, an error flag will be set to unity. After the meshes for all substructures

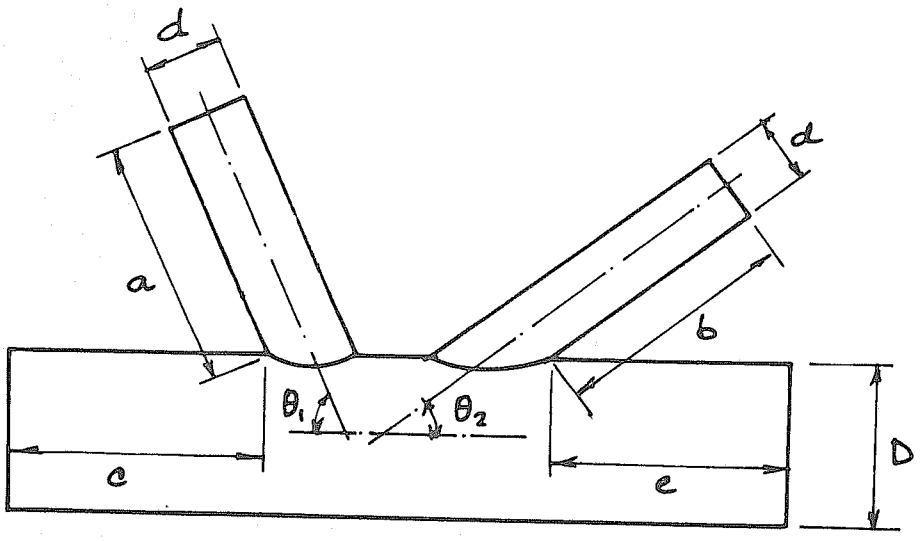
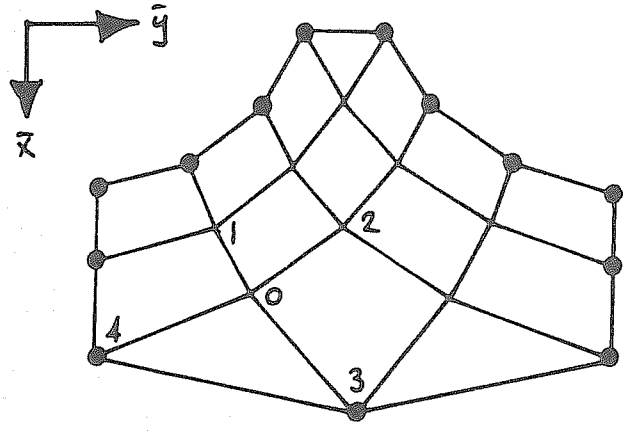


FIG. 2.6 IDENTIFICATION OF "MINIMUM DIMENSION" PARAMETERS



$$\bar{x}_0 = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4)$$
$$\bar{y}_0 = (\bar{y}_1 + \bar{y}_2 + \bar{y}_3 + \bar{y}_4)$$

FIG. 2.7 ITERATIVE SCHEME FOR COORDINATE GENERATION

have been generated, the error flag is examined and if found to equal unity, program execution is terminated. This is an indication that the user should inspect the meshes and decide whether or not they are acceptable to him. If they are considered suitable then in the following run the user can, by means of a data input parameter, instruct the program to override the error flag check after generating the mesh and continue execution. This means that the mesh is generated twice. However, the time taken for generating the mesh is very small compared to the time required to perform the analysis, so that a repeat of the mesh generation does not greatly change the total solution time.

The error flag can be set to unity also when certain array capacities are exceeded. If this occurs, however, the user should not suppress the error flag check because its suppression will lead to incorrect results. A description of all the situations which lead to program error messages is given in Section 5.4.

2.5 Meshes for Non-K Joints

In Section 2.2 it was pointed out that a joint structure having only one branch or no branches at all is still an allowable structure for analysis. However, it should be noted that this program was designed specifically for a K joint (chord plus two branches) and the mesh for the chord is generated on the assumption that two branches are always present. Thus, a joint with a single branch must be treated as an incomplete K joint. For example, suppose that the user wishes to analyze a T joint. He may do so by setting $\theta_1 = 90^o$ and specifying in the input that branch 2 is absent. But for the purpose of generating a mesh for the chord, a value for θ_2 must also be input and the location of the imaginary branch 2 must be specified. In other words, as far as the mesh generation is concerned,

the T joint must be treated exactly as if it were a K joint. The only difference is that for the T joint no mesh will be generated for branch 2. In the analysis the branch 2 substructure is not considered.

Section 7 points out specifically what data must be input and what data may be omitted for the case of a joint with only one branch.

2.6 Co-Ordinate Generation Parameters

This section briefly describes the use of three parameters NLOOP, EPS and NCYCLES. These are precision parameters which are associated with the scheme used for generating nodal point co-ordinates. This scheme is illustrated in Fig. 2.7.

The developed surfaces of the cylinders are considered and the co-ordinates of the nodes are related to local axis systems designated by $\bar{x}$, $\bar{y}$. Certain points in the mesh layout are selected as co-ordinate boundary nodes (not to be confused with displacement boundary nodes). At the co-ordinate boundary nodes (identified in Fig. 2.7 by solid circles) the $\bar{x}$, $\bar{y}$ co-ordinates are assigned fixed values. The co-ordinates of the other nodes are determined iteratively by averagin g the sum of the co-ordinates of the four adjacent nodes as shown in Fig. 2.7. One iterative cycle consists of one averaging operation on all the non-boundary nodes. After NLOOP cycles the relative differences in co-ordinates of two successive cycles are checked. The relative difference at each node is computed as

$$S_j = \frac{|x_j^{\,i} - x_j^{\,i-1}|}{x_j^{\,i-1}} + \frac{|y_j^{\,i} - y_j^{\,i-1}|}{y_j^{\,i-1}} ,$$

where the superscript i denotes the latest iterate and the subscript j denotes the j^{th} node. If all the S_j are less than or equal to EPS, then

the iterative procedure is terminated. If at any node S_j is greater than the quantity EPS, then the iterative procedure is continued for NLOOP more cycles at the end of which another check is carried out on the S_j. A limiting number for the total number of cycles is given by the parameter NCYCLES. The iteration will terminate when either all the S_j are less than or equal to EPS or the total number of cycles equals or exceeds NCYCLES. The user can specify values for NLOOP, NCYCLES and EPS or else allow the program to select the default values 10, 60, and 0.002, respectively. In the test cases which have been run the default values always resulted in meshes with a satisfactory appearance. However, for cases in which the chord or branches are exceptionally long, the default value for NCYCLES may need to be increased. It is up to the user to develop some experience in this regard.

Fig. 2.8(a) shows the $\bar{x}$, $\bar{y}$ local axis systems which are used for each substructure in generating the co-ordinates. Developed surfaces are shown. The same local axis system is used for the chord and the plugs: $\bar{x}$ is in the chord circumferential direction and $\bar{y}$ is in the axial direction. For the branch $\bar{x}$ is in the branch axial direction and $\bar{y}$ is in the circumferential direction. The locations of the origins for these axes are indicated in Fig. 2.8(b). The program transforms the local co-ordinates to the global co-ordinate system X, Y, Z and prints both sets of figures in the output.

2.7 Sub to Main Structure Connectivity

This section is not essential for understanding the use of the program and may be skipped.

The connectivity between the substructures (plug 1, plug 2, branch 1, branch 2) and the main structure (chord) is established by identifying:

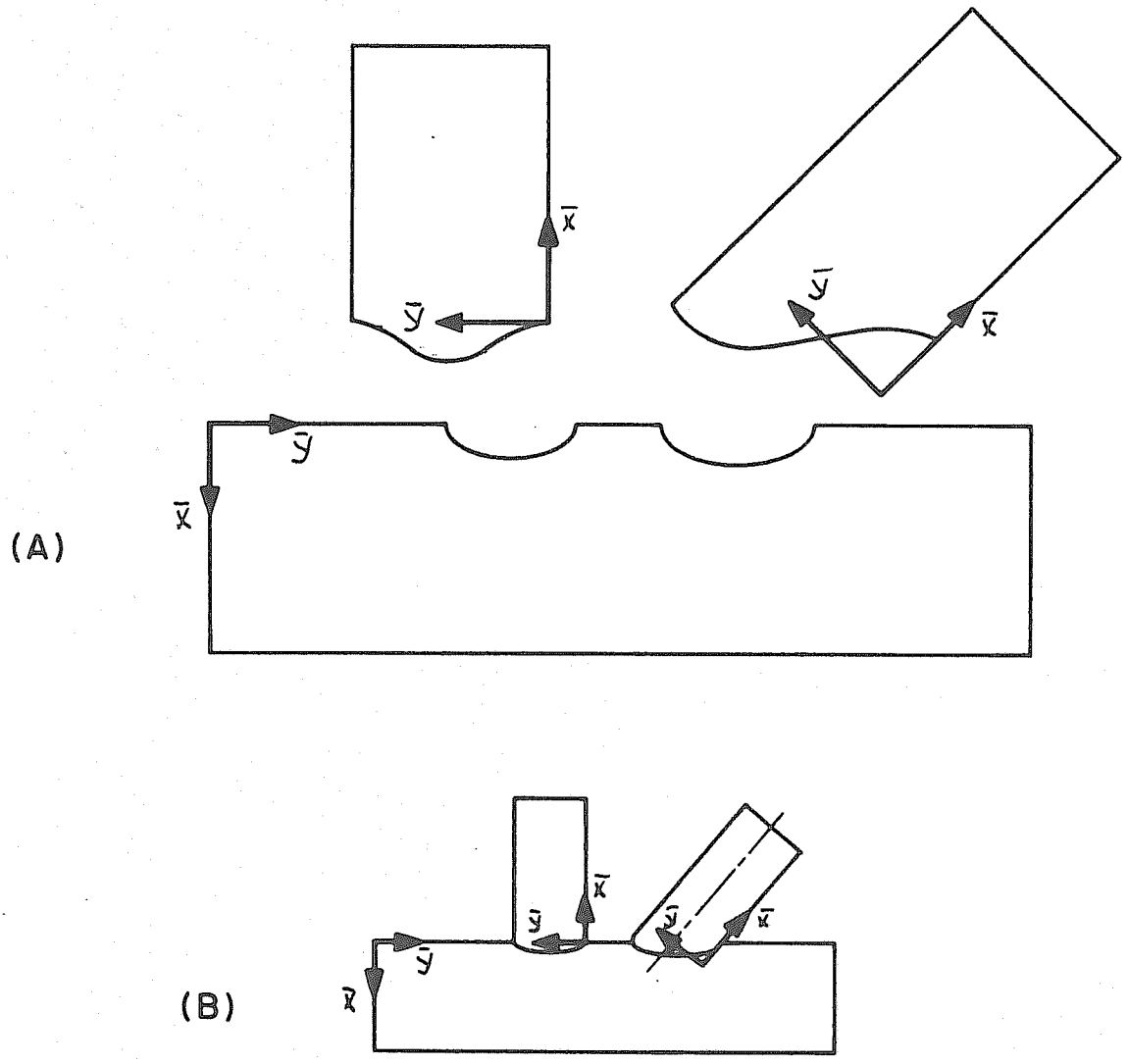


FIG. 2.8 AXES FOR LOCAL CO-ORDINATES

i) for each substructure the number of nodes which connect that substructure to the main structure, and ii) the number of the first node in the group of sequentially numbered nodes of the main structure which form the connection to the substructure in question. The output lists the above information but in terms of numbers of degrees of freedom rather than numbers of nodes. Thus, the variable KSM(I) is the number of degrees of freedom (i.e., number of nodes times 5) which connect the I^{th} substructure to the main structure, and the variable JSM(I) is the number of the first degree of freedom of the main structure which connects to substructure I. If N(I) is the number of the first main structure node which connects to substructure I, then JSM(I) = (N(I) - 1) * 5 + 1. For programming reasons, KSM and JSM for the main structure are assigned the values 1 and 0, respectively.

2.8 Mesh Plotting

A plot of the developed surface of each substructure showing the finite element mesh together with node and element numbers can be obtained if suitable plotting equipment is available. Plotting systems and their associated subroutines often vary from one computer to another. Appendix 2 describes those plotting subroutines used in this program which are peculiar to the plotting system at the U. C. Berkeley Computer Center.

The plotting system at U.C.B. uses a CALCOMP machine which produces plots on an 11 inch wide continuous strip of paper. The mesh plots of the developed surface of each substructure are oriented on the paper as shown in Fig. 2.9. Note that the branch plots have the axial directions of the branches in the long direction of the paper. The dimension of each plot in the 11 inch wide direction (indicated by YL in Fig. 2.9) can be

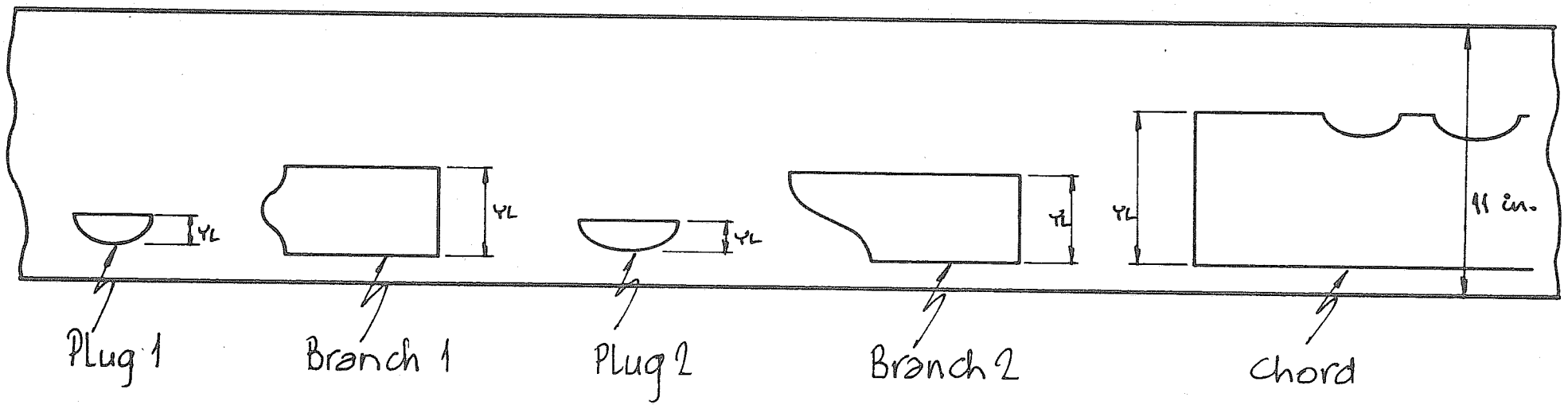


FIG. 2.9 ORIENTATION OF MESH PLOTS ON PLOTTING PAPER

specified by the user. The dimension in the other direction is automatically adjusted so that the same scales are used in the two directions. If the user wants all substructure plots drawn to the same scale he must specify the appropriate dimension YL for each substructure. The input format for this information is given in Section 5.1.

3. DISPLACEMENTS, BOUNDARY CONDITIONS, LOADING

3.1 Co-Ordinate Systems

Fig. 3.1 shows two co-ordinate systems - the global (X, Y, Z) and the tangent (x', y', z'). Each is a right handed rectangular Cartesian system. The global system has as origin the uppermost point of the midsurface of the left end of the chord. The Y axis is parallel to the generator of the chord and the X axis is in the plane tangent to the chord at the origin. The tangent system is spatially dependent and has the nodes as origins. The (x', y') plane is tangent to the cylindrical surface which contains the node in question, with the y' axis parallel to the generator associated with that surface. For nodes at the intersections of the branches with the chord two sets of tangent axes are applicable.

3.2 Displacement Degrees of Freedom

Five displacement components are defined at each node: two rotations (about the x' and y' axes) and three translations. A rotation about the z' axis is not defined. The rotations are always expressed with respect to the tangent axes but the translations can be expressed with respect to either the global or tangent system at the choice of the program user.⊛ The system chosen for the translations is referred to as the base co-

⊛ For interpretation of the displacements at the pipe junction nodes, see Appendix 4.

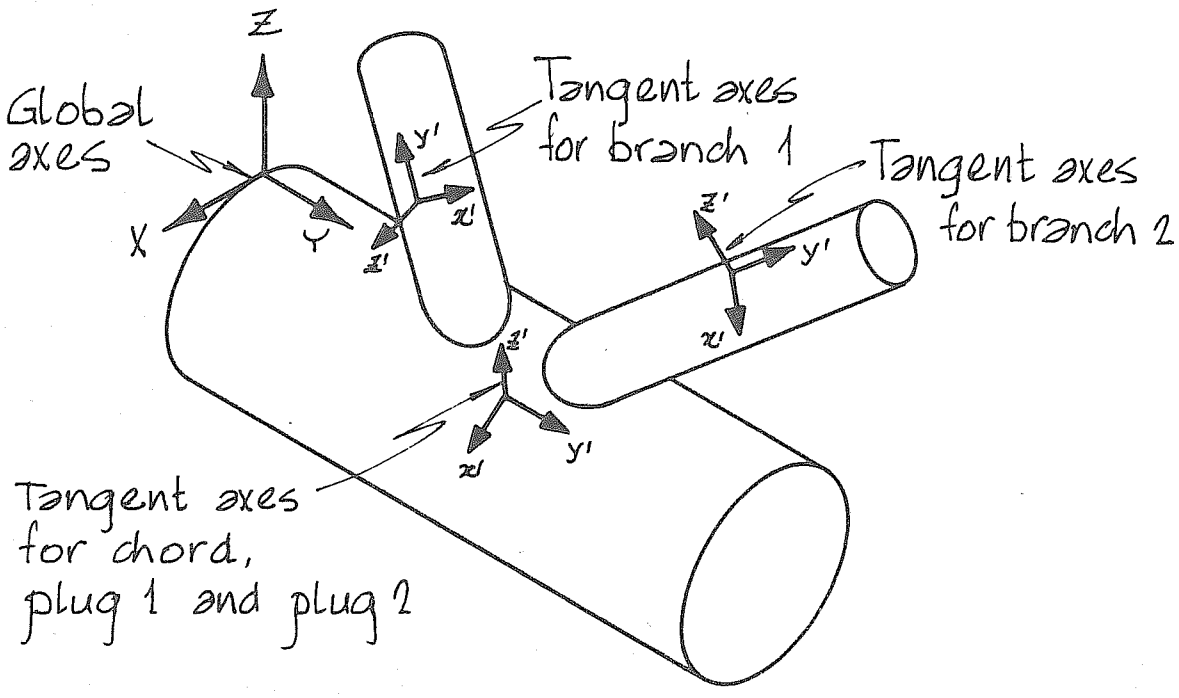


FIG. 3.1 GLOBAL & TANGENT AXES

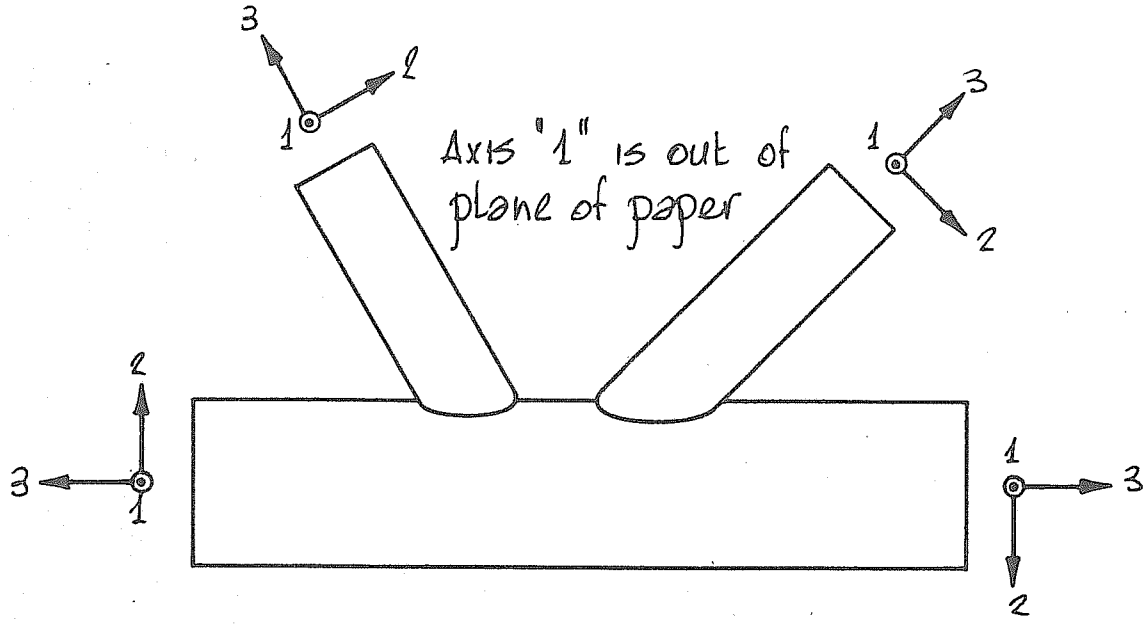


FIG. 3.2 LOAD APPLICATION AXES

ordinate system. The reason for having these two alternate ways of expressing the translations is related to the specification of displacement boundary conditions. A discussion on the selection of the base system is presented in the following section which deals with boundary conditions.

The following notation is used later in referring to the displacements:

D1 = Translation in either global X direction or tangent x' direction.

D2 = Translation in either global Y direction or tangent y' direction.

D3 = Translation in either global Z direction or tangent z' direction.

D4 = Rotation about x' axis.

D5 = Rotation about y' axis.

3.3 Boundary Conditions

A boundary node is defined to be a node at which at least one of the five displacement degrees of freedom has an a priori specified magnitude (zero or non-zero). Since only one half of the symmetric structure is considered in the analysis, the above definition implies that all nodes on the plane of symmetry are boundary nodes. Nodes at the ends of the members will usually also be boundary nodes because of displacement constraints introduced there. By "ends of members" is meant the two ends of the chord and those ends of the branches which don't connect to the chord. Boundary displacements are referenced to the base co-ordinate system.

All boundary nodes except the end boundary nodes have one of two types of boundary conditions which correspond to symmetric and anti-symmetric loading. The user has only to specify whether the loading is symmetric or anti-symmetric; the program will generate the boundary conditions at the appropriate nodes. For the boundary nodes at the ends of

the pipes the boundary conditions will depend on the types of support provided -- whether fixed, free, hinged, etc. To simplify description of the end boundary conditions, the program provides for a standard set of end support types. The list of available types and their identifying numbers is listed in Appendix 1. Thus, the user only specifies the boundary condition identifying numbers for the four pipe ends.

The choice of the base system is made on the basis of which system (global or tangent) is best able to represent the end support conditions in question. Some end conditions (e.g., type 1.225) can only be represented with reference to the global system, while others (e.g., type 1.213) can only be represented via the tangent system. If both global and tangent axes systems are able to represent the end support conditions in question, then the choice of base system is one of convenience or personal preference.

Provision has been made for only a limited number of end support types for both symmetric and anti-symmetric loading. In all these standard types, where a displacement constraint is introduced, the displacement is set equal to zero. However, if the user finds the standard options insufficient, he can provide for the desired boundary conditions in one of two ways. Firstly, an option is provided which allows for the modification of the boundary condition of any node. Once the mesh has been generated and the nodes identified, the user can change the boundary conditions of nodes which have been generated as boundary nodes and add new nodes to the list of boundary nodes. This option also allows for non-zero displacements to be specified. Detailed description of the use of this option is given in Section 5.1. The second way in which the user can provide for end conditions which have not been catered for in the program is to

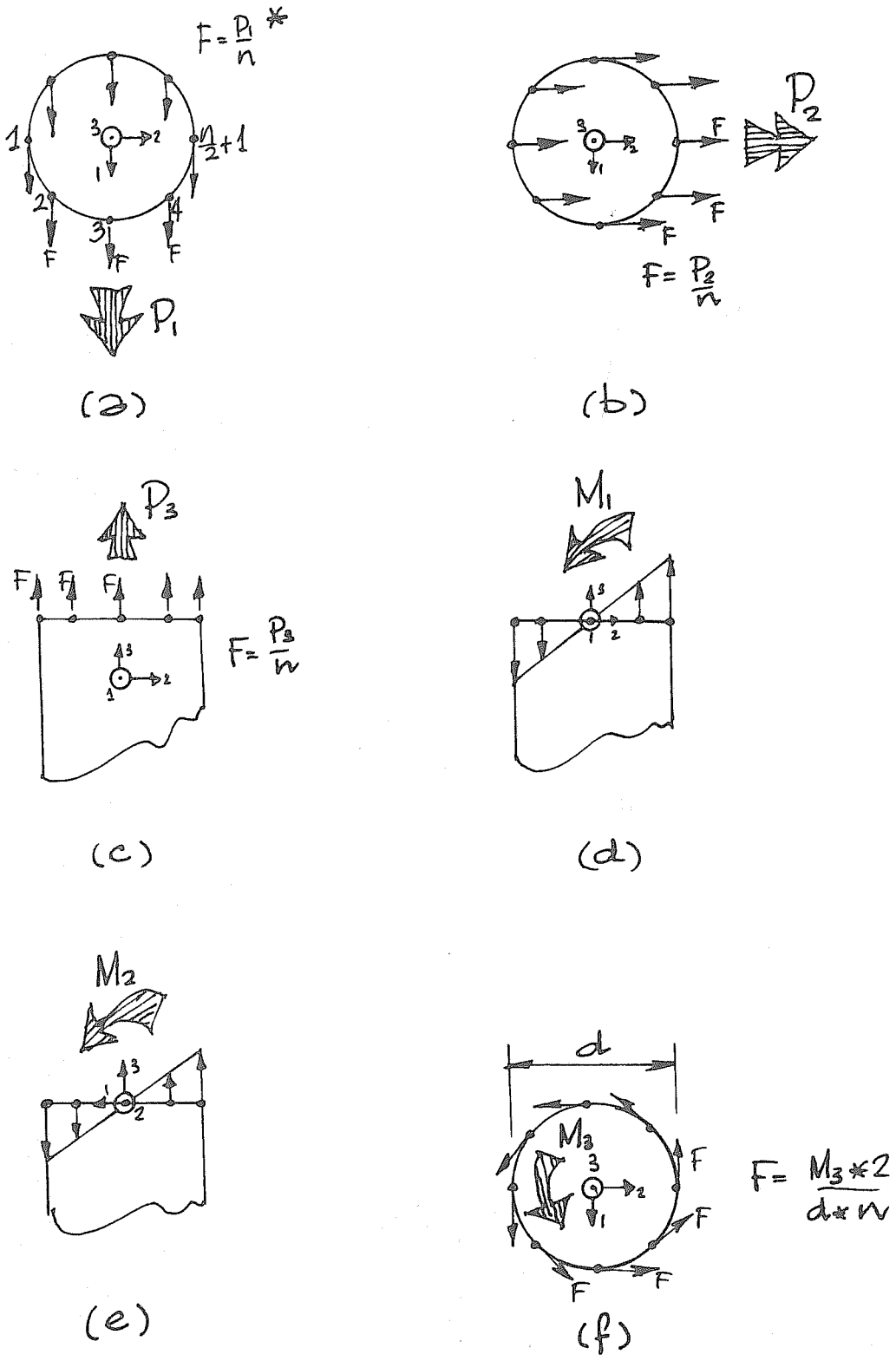
extend the list of end support types by making some simple programming changes. Details of these alterations are given in Appendix 1.

3.4 Loading

At each node five force components, corresponding to the five displacement degrees of freedom, are applied. These are moments about the x' and y' axes and forces in the directions of the base axes.

In order to simplify the input required to describe the forces acting on the structure, only the nodes at the two ends of the chord and at those ends of the branches away from the pipe juncture area may be subjected to (non-zero) loads. At all other nodes the forces are set equal to zero. The applied loads are described by the user in terms of force and moment resultants referred to special axes at the pipe ends. Fig. 3.2 shows the orientation of these axes at the four locations where loads may be applied. At each location the 1 axis is pointing out of the plane of the paper. Six resultant quantities can be applied at each end - three force resultants (P_1, P_2, P_3) in the 1, 2, 3 directions and three moment resultants (M_1, M_2, M_3) about the 1, 2, 3 axes. These quantities can be applied either singly or in any combination provided that all the components of the combination are consistently symmetric or anti-symmetric with respect to the global Y-Z plane.

The program automatically divides the resultant quantities into statically equivalent forces applied at the appropriate nodes and transforms these forces to the base axes. The P_1, P_2, P_3 resultants are decomposed into forces of equal magnitudes applied at the end nodes, as shown in Figs. 3.3(a), (b) and (c). The M_1 and M_2 resultants are decomposed into linearly varying forces (Figs. 3.3(d) and (e)) and M_3 is decomposed into equal forces which are tangent to the cylindrical



* n = number of nodes at end

FIG. 3.3 SCHEME OF LOAD DISTRIBUTION TO END NODES

surface (Fig. 3.3(f)).

For each substructure, the magnitudes of the nodal forces, with respect to the 1, 2, 3 axes corresponding to the substructure in question, which are statically equivalent to the resultant quantities are identified in the printed output as array F. The ordering of these entries corresponds to the node numbering shown in Fig. 3.3(a). Note that these numbers are locally assigned and are not the absolute node numbers. Array F, transformed to the base co-ordinate axes, is identified as array FB. The absolute numbers which correspond to the loaded nodes are then identified and the forces stored in FB are entered into the appropriate location of array P which is the load array (listing the five load components which act at each node) for the substructure.

In addition to the types of loading described above, a uniform pressure load may also be applied. The user has only to specify the magnitude of the pressure; the resulting nodal forces are generated by the program. Note, however, that for plug 1 and plug 2 no forces are generated. Fig. 3.4 shows the surfaces on which forces are generated by a positive uniform pressure. The pressures on the inside and outside surfaces of each plug are equal and hence no resultant force is exerted on the plugs. The chord may have cut-outs in place of the plugs. For a joint with one branch a pressure loading would make sense only if the plug corresponding to the missing branch is physically absent, since no pressure forces will be generated for this plug. Also, in generating the nodal forces due to pressure, it is assumed that there are no end plates at the pipe ends - i.e., axial forces at the ends are absent. If, in the problem to be analyzed, end plates are present, then axial forces are introduced by application of pressure load. However, these axial forces will not be

generated by the program. The user must input the axial force resultants as a separate load case and indicate (by means of a special input parameter) that these forces are to be superimposed on the forces resulting from the pressure load. The form of the input data for this situation is shown by an example in Section 7.

In a single execution of the program any number of load cases can be analyzed provided all are either symmetric or antisymmetric.

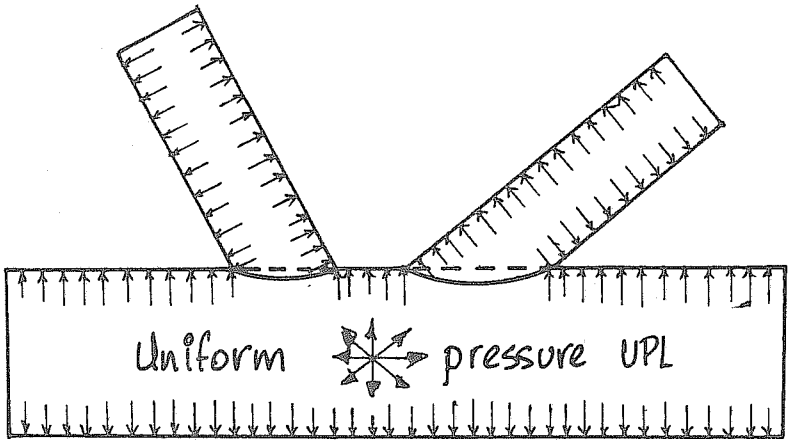
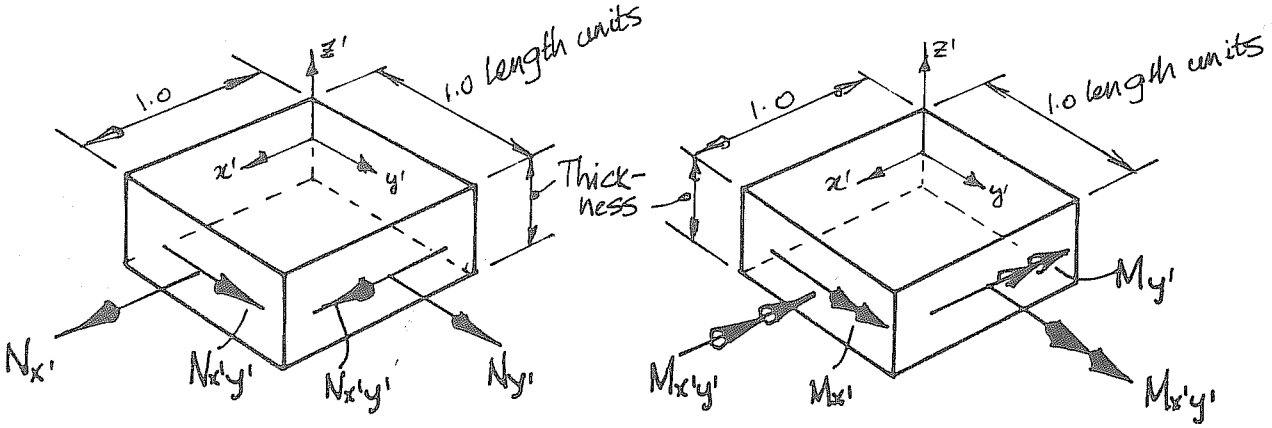
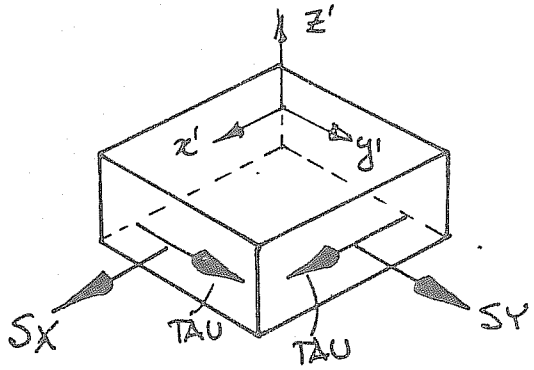


FIG. 3.4 FORCES GENERATED BY POSITIVE UNIFORM PRESSURE



All quantities shown are positive

FIG. 4.1 SIGN CONVENTION FOR MEMBRANE FORCES AND MOMENTS



All quantities shown are positive

FIG. 4.2 SIGN CONVENTION FOR SURFACE STRESS

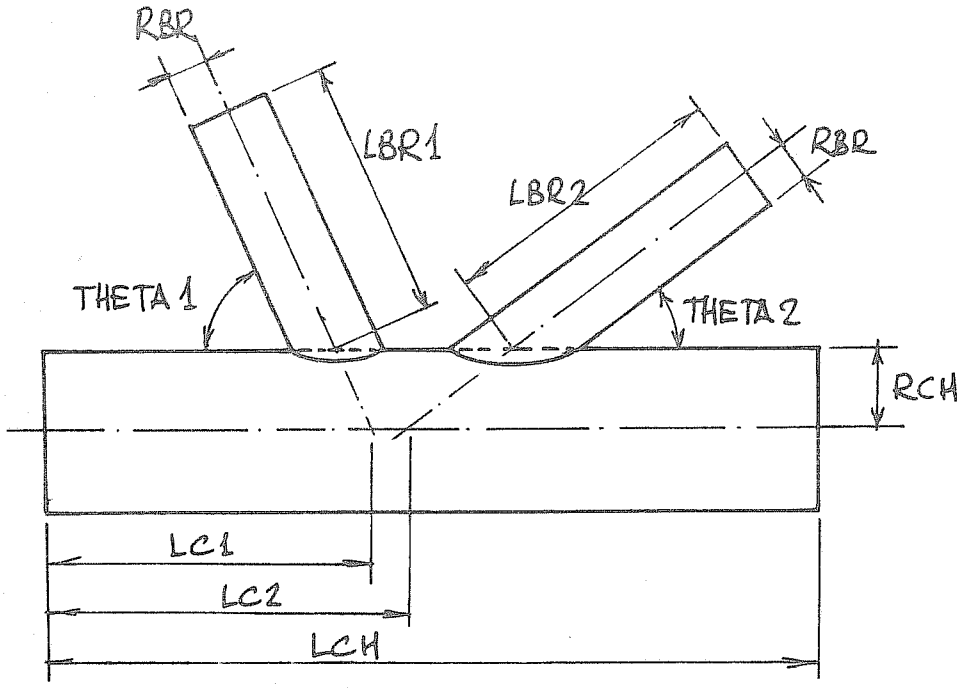


FIG. 5.1 IDENTIFICATION OF K JOINT GEOMETRY

4. FORCES IN SHELL

4.1 Moments and Membrane Forces

The program computes moments (Mx', My', Mx'y') and membrane forces (Nx', Ny', Nx'y') at each nodal point. These are resultant quantities over the pipe thickness and unit width of the pipe surface. They are referred to the tangent co-ordinate axes according to the sign convention shown in Fig. 4.1.⊛ Thus the units of Mx', My', Mx'y' are (length*force/length) and the units of Nx', Ny', Nx'y' are (force/length). The six nodal quantities are obtained by taking an equally weighted average of the respective moments and membrane forces in the elements which connect to the node in question.

4.2 Surface Stresses

The user can optionally obtain from the program, stresses on the inside and outside surfaces of the pipes. The surface stresses SIGMA X, SIGMA Y, TAU XY (whose sign convention is shown in Fig. 4.2⊛) are computed from the moments and membrane forces according to the following formulae:

$$D = 6/T^2$$

$$\text{SIGMA X} = \pm D * Mx' + Nx'/T$$

$$\text{SIGMA Y} = \pm D * My' + Ny'/T$$

$$\text{TAU XY} = \pm D * Mx'y' + Nx'y'/T$$

in which T is the pipe wall thickness. The positive and negative signs are for the outside and inside surfaces, respectively.

Principal stresses (denoted by SIGMA 1, SIGMA 2) are computed from SIGMA X, SIGMA Y, TAU XY using standard stress transformation relationships.

⊛ For interpretation of the moments, membrane forces and stresses at the pipe juncture nodes, see Appendix 4.

5. COMPUTER PROGRAM USAGE

5.1 Input Data

The usual FORTRAN convention is used for writing formats.

A = alphanumeric field

I = integer (must be packed to the right of the field)

F = floating point number (must be punched with a decimal point)

Card 1. Title Card (12A6)

Cols. 1-72 Alphanumeric information for problem identification.

Note: Only columns 1-30 will appear as heading on plot of mesh.

Card 2. Control Card (A10, I5)

This card indicates (i) whether or not the analysis is to be performed after the mesh has been generated, and (ii) whether or not another problem follows the current one.

Cols. 1-8 If only the mesh is to be generated print the word MESHONLY

If the analysis is to be performed print the word ANALYZE

Cols. 10-15 0 if this is the last problem

1 if another problem follows

Card 3. Substructure and Mesh Definition (6A10)

This card indicates which substructures are present in the structure being considered and the degree of mesh refinement desired. If a substructure is not present, then the appropriate field is left blank and the program will not generate a mesh for that substructure. As pointed out in Section 2.2, any of the substructures plug 1, plug 2, branch 1, branch 2 may be absent, but the chord must always be present for the analysis to be carried out. However, if the user wishes only to generate the mesh and not perform the analysis, then he need not necessarily request that the mesh for the chord be generated. For example, he may want to

examine only the meshes for the branches and not the chord or plugs. He would then indicate on this card that the chord and plugs are absent -- i.e., the meshes for these substructures are not to be generated. See Section 7 for an example.

Cols. 1-6 The word CHORDM is a mesh for the chord is to be generated. Otherwise, blank.

11-15 The word PLUG1 if a mesh for plug 1 is to be generated. Otherwise, blank.

21-25 The word PLUG2 if a mesh for plug 2 is to be generated. Otherwise, blank.

31-37 The word BRANCH1 if a mesh for branch 1 is to be generated. Otherwise, blank.

41-47 The word BRANCH2 if a mesh for branch 2 is to be generated. Otherwise, blank.

51-60 Starting in Col. 51, the word COARSE, MEDIUM, FINE or XFINE (See Figs. 2.2, 2.3, 2.4, 2.5) to indicate the mesh type required.

Card 4. Definition of Geometry (5F10.0)

Refer to Fig. 5.1 for an illustration of the variable names used in this and the following card.

Cols.	Variable Name	Meaning
1-10	RCH	Radius of chord
11-20	RBR	Radius of branches
21-30	LCH	Length of chord
31-40	LBR1	Length of branch 1
41-50	LBR2	Length of branch 2

Card 5. Definition of Geometry - Continued (4F10.0)

Cols.	Variable Name	Meaning
1-10	LC1	Location of branch 1
11-20	LC2	Location of branch 2
21-30	THETA1	Angle of inclination of branch 1 (degrees)
31-40	THETA2	Angle of inclination of branch 2 (degrees)

Card 6. Material Properties for Chord (3F10.0)

The chord (including the plugs) must have a uniform elastic modulus and a uniform Poisson ratio throughout. It is assumed at this stage that the chord wall thickness is also uniform throughout. If the thickness varies over the chord, the user must assign a single thickness value on this card and specify the variation at a later stage in the input data description. Card number 11 contains a parameter indicating whether or not any elements are to have their thicknesses changed from the value input on card 6. The description of the format for inputing the thickness changes follows the cards containing the load information.

Cols. 1-10 Elastic modulus of chord and plugs.

11-20 Poisson ratio of chord and plugs.

21-30 Thickness of chord and plugs.

Card 7. Material Properties for Branch 1 (3F10.0)

The remarks regarding the elastic modulus, Poisson ratio and thickness of the chord also apply for the two branches. Each branch must have a single elastic modulus and a single Poisson ratio. Variations in thickness are allowed but only a single thickness value can be assigned here.

Cols. 1-10 Elastic modulus for branch 1.

11-20 Poisson ratio for branch 1.

21-30 Thickness for branch 1.

Card 8. Material Properties for Branch 2 (3F10.0)

Cols. 1-10 Elastic modulus for branch 2.

11-20 Poisson ratio for branch 2.

21-30 Thickness for branch 2.

Card 9. Mesh Plot Dimensions (5F10.0)

This card indicates the heights YL (as shown in Fig. 2.9) of the mesh plots of each substructure. These distances must be in inches.

Cols. 1-10 Distance YL for chord.

11-20 Distance YL for plug 1.

21-30 Distance YL for plug 2.

31-40 Distance YL for branch 1.

41-50 Distance YL for branch 2.

Card 10. Iteration Control Parameters (2I5, F10.0)

This card lists values of the parameters NLOOP, NCYCLES and EPS which were discussed in Section 2.6.

Cols. 1-5 NLOOP or blank.

6-10 NCYCLES or blank.

11-20 EPS or blank.

If this card is left blank the program selects for NLOOP, NCYCLES and EPS the default values 10, 60 and 0.002, respectively.

Card 11. Control Card (9I5, 10X, A10)

Cols.	Variable Name	Value and Meaning
1-5	ICOORD	0, global co-ordinate system is base system.
		1, tangent co-ordinate system is base system.
6-10	IBOUND	0, don't modify boundary conditions.
		1, modify boundary conditions.

Cols.	Variable Name	Value and Meaning
11-15	ITHICK	0, don't modify thicknesses 1, modify thicknesses
16-20	LVECT	Number of load cases
21-25	IUPL	0, no uniform pressure load is applied 1, a non-zero pressure load is applied, the value of which is given on card B.
26-30	ISUPER	0, don't superpose pressure load on other load cases 1, Superpose pressure load on other load cases
31-35	IPLOT	0, don't plot mesh 1, plot mesh
36-40	NUMPLT	0, don't plot node and element numbers 1, plot node and element numbers
41-45	IGO	0, don't compute surface stresses 1, compute surface stresses on inside and outside of pipe wall
56-63		The word CONTINUE if the error flag check is to be over-ridden -- otherwise blank. The conditions under which the error flag check may be over-ridden are discussed in Section 5.4

Some additional explanatory notes on the above . . . i) ICOORD indicates the choice of the base co-ordinate system -- i.e., it indi-

cates whether the translations are to be expressed in global co-ordinates or tangent co-ordinates. ii) IBOUND indicates whether or not the boundary conditions as specified on Card 12 are to be modified. The actual description of any modifications follows the loading information. iii) ITHICK indicates whether or not the thicknesses as input on Cards 6, 7, 8 are to be modified. The description of any modifications follows the loading information. iv) LVECT specifies the number of independent load cases and is computed in the following way:

a) Uniform pressure load (UPL) only . . . . . LVECT = 1

b) UPL superimposed on N load cases . . . . . LVECT = N

c) UPL plus N load cases (no superposition) . . . . LVECT = N+1

d) N load cases . . . . . LVECT = N .

v) ISUPER is the parameter mentioned in Section 3.4 in connection with superposition of forces due to a uniform pressure load on other forces. This option was introduced specifically for the situation in which a joint having end plates at the member ends is subject to a uniform pressure load. In such a situation (as pointed out in Section 3.4) axial forces are introduced in the members in addition to the forces normal to the shell surfaces. The normal forces are generated by the program, but the axial forces are not. The program user must compute the resultant axial forces, input these as a separate load case and specify ISUPER = 1 to indicate that the pressure load forces and axial forces are to be superposed. A description of the loads in this way constitutes a single load case and the response of the structure will be given for the combined loading.

The use of ISUPER = 1 does allow for more than one load case to be analyzed. However, each load case will consist of the pressure loading combined with the loads input on cards 14 and onwards.

An example of the use of this option is given in Section 7.

Card 12. Boundary conditions (A5, 4I5)

Cols. 1-5 Starting in col. 1 the word SYM or ASYM depending on whether the loading is symmetric or antisymmetric respectively.

5-10	Support type identification number for end of branch 1	Select type identification number from Appendix 1
11-15	Support type identification number for end of branch 2	
16-20	Support type identification number for left end of chord	
21-25	Support type identification number for right end of chord	

NOTE: If MESHONLY is specified on card 2 then this is the end of the data cards. If the analysis is to be carried out as well then the cards which follow must be included.

Card 13. Uniform pressure load (F10.0)

Cols. 1-10 Magnitude of uniform pressure load. This must be input as zero if no pressure load is applied. A positive magnitude corresponds to a fluid pressure applied internally to the pipes.

Card 14. Loading (6F10.0)

This card lists the force and moment resultants P_1, P_2, P_3, M_1, M_2, and M_3 applied at the end of branch 1. The six quantities are referred to the (1,2,3) axes at the end of branch 1 as shown in Fig. 3.2.

Cols	1-10	P_1
	11-20	P_2
	21-30	P_3
	31-40	M_1
	41-50	M_2
	51-60	M_3

Cards 15, 16, 17. Loading - continued (6F10.0)

On these three cards the P_1, P_2, P_3, M_1, M_2 and M_3 quantities applied at the end of branch 2 and the two ends of the chord are input following the format for card 14. The resultants are referred to the (1,2,3) axes (Fig. 3.2) at the appropriate ends. Cards 15, 16, 17 are for the branch 2 end, the chord left and right ends respectively.

Repeat cards 14 thru 17 for additional load cases.

NOTE: The load information completes the data deck if both the variables IBOUND and ITHICK on card 11 are zero. If either variable has value one, then the following cards must be included to describe the way in which the thicknesses and/or boundary conditions previously specified are to be modified.

In order to specify the variations in thicknesses from the values input on cards 6,7,8, the user must first have used the MESHONLY option on card 2 and must identify the numbers of the elements for which thicknesses are to be changed. Likewise, in order to describe changes in the boundary conditions input on card 12, the numbers of the nodes must be identified.

Note that the "modifications" discussed here are actually data that must be input in addition to the standard problem description data (cards 1 through load cards) to describe the structure being analyzed. The structure as described by this additional information is analyzed for all load cases -- i.e., it is not possible to introduce modifications for one particular load case and not another.

The following cards are grouped by substructure number.

Substructure No. 1

(A) Next Card (I5) Include this only if ITHICK = 1. The thickness modification is performed by first specifying how many cards are used to describe the modifications for substructure 1.

Cols. 1-5 N Number of cards following.

Note that if ITHICK = 1 but no elements of substructure 1 are to have their thicknesses changed, then N must be input as zero.

(B) Next cards (2I5, F10.0)

Include these cards only if ITHICK = 1 and N > 0. If N > 0 then these N cards indicate the numbers of the elements whose thicknesses are to be changed and the new thickness values.

Cols	Variable Name	Meaning
1-5	I	Number of first element in a group of sequentially numbered elements with thickness TH.
6-10	J	Number of last element in a group of sequentially numbered elements with thickness TH.
11-20	TH	Thickness of the sequentially numbered group of elements - i.e., elements I, I + 1, I + 2, ...J will be assigned thickness TH.

(C) Next Card (I5) Include only if IBOUND = 1.

The number of cards used to describe the boundary condition modifications for substructure 1 is first specified.

Cols. 1-5 M Number of cards following.

Note that if IBOUND = 1 but no boundary condition changes are to be introduced in substructure 1, then M must be input as zero.

(D) Next Cards (I4, 5I1, 1X, 5F10.0) Include only if IBOUND = 1 and M > 0.

If M > 0 then these M cards indicate the numbers of the nodes which are to have their boundary conditions changed or are to be introduced as new boundary nodes.

Cols. 1-4 Node number

Col. 5 = 1 for specified value of displacement D1*
0 if D1 is unconstrained

6 = 1 for specified value of displacement D2
0 if D2 is unconstrained

7 = 1 for specified value of displacement D3
0 if D3 is unconstrained

8 = 1 for specified value of displacement D4
0 if D4 is unconstrained

9 = 1 for specified value of displacement D5
0 if D5 is unconstrained

Cols. 11-20 Specified ** value of D1 (leave blank if this degree of freedom is unconstrained)

21-30 " D2 "

31-40 " D3 "

41-50 " D4 "

51-60 " D5 "

*See Section 3.2 for definition of displacements.

**Specified value may be non-zero.

Next Cards

The sets labelled (A), (B), (C), (D) above are repeated for substructures 2,3,.... NSUBS, where NSUBS is the total number of substructures (including the main structure). Section 2.2 describes the substructure numbering.

Fig. 5.2 illustrates the input data deck setup.

5.2 New Problem

The input cards listed in Section 5.1 (and shown graphically in Fig. 5.2) constitute the description of one problem. The data for a new problem follows the data for the previous one. Note that card 2 of the data deck for each problem must indicate whether or not another problem follows the current one.

5.3 Output

The printed output contains

(i) Echo check of the input data.

(ii) Parameters M1 to M15. These are unimportant for the interpretation of the final results. For their interpretation see Ref. 1.

(iii) For each substructure in turn the following information:

(a) Name of the substructure

(b) Intermediate data generation figures the end of which is signalled by the messages

ITERATION STOPPED
NUMBER OF CYCLES XX
SUM XX

The "number of cycles" and "sum" are NCYCLES and Sj (for the last node inspected) as described in Section 2.6.

(c) Node co-ordinated with respect to local $\bar{x},\bar{y}$, axes

(d) Element node numbers

(e) Displacement boundary nodes and boundary conditions

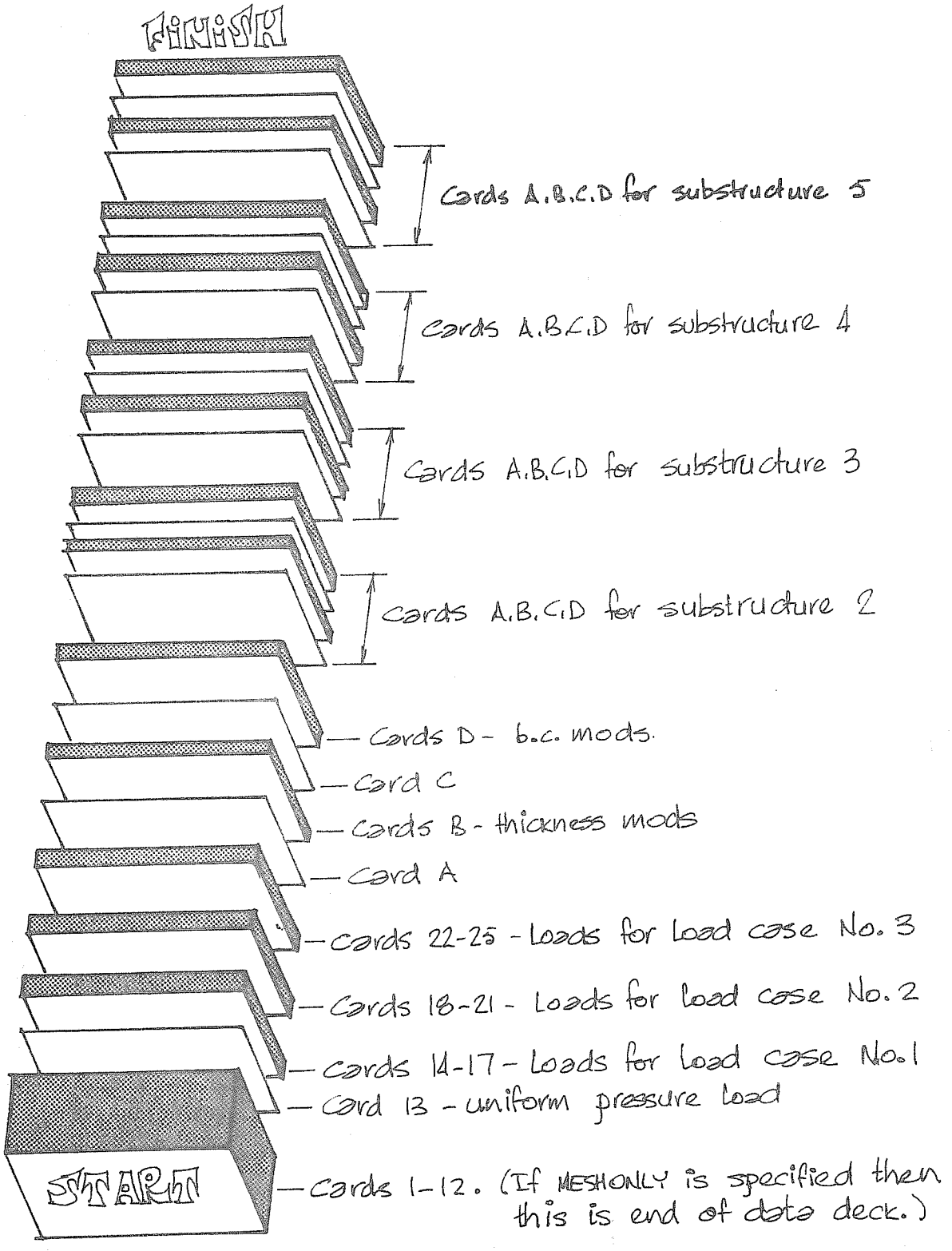


FIG. 5.2 INPUT DATA DECK SETUP FOR ONE PROBLEM

(iv) Echo check of input loads.

(v) For each substructure the load arrays F, FB and P (see Section 3.4).

(vi) Substructure numbers (see Section 2.2).

(vii) The number of elements (NUMELS) number of nodes (NUMPTS) number of displacement boundary nodes (NUMBPS) half band width in terms of node numbers (NBANDS) plus the two parameters JSM, KSM (see Section 2.7) for each substructure.

(viii) The numbers of the nodes at the end of branch 1, end of branch 2 and left and right ends of the chord respectively (array LOC).

(ix) The number of nodes at each of the four above mentioned locations (array NEND).

(x) For each substructure:

(a) Local co-ordinates.

(b) Global co-ordinates.

(c) Element node numbers and material properties.

(d) Displacement boundary conditions.

(e) Various other intermediate information.

(xi) For each substructure:

(a) Nodal forces due to applied pressure (if applied pressure is non-zero).

(b) Nodal forces due to input displacements. These are zero unless non-zero displacements are input.

(c) Sum of nodal forces due to (a) and (b) plus applied loads.

(xii) Displacements$^{\circledast}$ listed by load case and substructure number.

(xiii) For each load case and substructure:

(a) Moments and membrane forces$^{\circledast}$ $M_{x'}$, $M_{y'}$, $M_{x'y'}$, $N_{x'}$, $N_{y'}$, $N_{x'y'}$. (see Section 4.1)

$^{\circledast}$ For interpretation of these quantities at the pipe juncture nodes, see Appendix 4.

(b) Stresses⊛ on outside and inside of pipe (if IGO = 1). See Section 4.2.

5.4 Error Exits

As mentioned in Section 2.4, under certain circumstances an error flag is set which causes termination of program execution. The program prints error messages explaining the reason for termination. The following list describes the situations in which termination occurs.

(i) The lengths of the branches or the dimensions c and e in Fig. 2.6 are less than the minimum values required by the program. If this occurs the user can over-ride the error flag by specifying the word CONTINUE on input card 11.

(ii) The number of nodes or number of elements or number of boundary nodes generated for a particular substructure exceeds the maximum dimensioned value of 700, 600, 160 respectively. This error situation is unlikely to occur as the above maximum values should be ample for all but the longest pipes. If the error does occur, however, the user should choose a coarser mesh.

(iii) The maximum node number difference for any element exceeds the allowed value. If this error occurs, a coarser mesh must be used. This check is included to ensure that the maximum node number difference (which is related to the half band width as described in Section 2.4) does not exceed the value that can be coped with by the available core size. The core requirements for different half band widths are discussed in Section 6. The current "allowed value" for the maximum node number difference is 45 (which corresponds to a half band width of 230). The check for this is made in SUBROUTINE WRITE2 in the IF statement following statement 50. The user can change the allowable value to whatever his available equipment

⊛ For interpretation of the stresses at the pipe juncture nodes, see Appendix 4.

can cope with.

(iv) Two node numbers in an element are equal. This would indicate a program error which, hopefully, will never occur.

(v) The dimensions of certain arrays associated with the co-ordinate generation and mesh plotting are exceeded. This would occur because of a program error.

Note that only under condition (i) of the above, is the error flag check allowed to be over-ridden.

6. PROGRAMMING INFORMATION

The program is written in FORTRAN IV and was developed on a CDC computer at the University of California, Berkeley.

On-line input/output is done via FORTRAN statements READ N, list and PRINT N, list. FORTRAN logical units 1 through 9 are used for intermediate storage via FORTRAN statements WRITE (I) list and READ (I) list. The program uses the overlay feature with two primary overlay levels and no secondary levels. The overlay structure used is described in Appendix 3.

A number of variables in the program have seven-character names. Some machines (for example IBM 360) have a limit of fewer than seven characters for a variable name, hence these names must be changed in adapting the program to such machines.

The CDC 6400 has a 60 bit word length. If the analysis part of the program is to be run on a computer having a smaller word length, precision problems may be encountered in analyzing structures consisting of a large number of elements.

The minimum core size required for execution on the CDC 6400 is 135000_8* (octal) or 47616 (decimal). This amount of core can accommodate a half band width (1/2 bw) of 230 which should be sufficient for most meshes of the FINE type. If additional core storage is available, problems having larger 1/2 bws can be solved. The arrays which need to be expanded in size for a 1/2 bw greater than 230 are all located in a blank COMMON block. In the CDC 6400 the blank COMMON is the last block of storage occupying the core (i.e., it is located after the program, system subroutines, labeled COMMON blocks and dimensioned arrays) and so can be truncated to the required size without interfering with the rest of the program. Fig. 6.1 shows a plot of blank COMMON

---

*The subscript 8 signifies that the number is in octal.

length required vs. 1/2 bw. In the program the blank COMMON block is dimensioned to accommodate a maximum 1/2 bw of 400. For analyzing a joint with the XFINE mesh it is recommended that the MESHONLY option should first be used. This will provide the 1/2 bw of the main structure and the user can then determine what length of blank COMMON is required. For FINE mesh analyses it is recommended that the above procedure should also be followed until the user has acquired sufficient knowledge to be able to predict the 1/2 bw.

Some idea of the core size required may be obtained from the following figures. In the CDC 6400 at U. C. Berkeley, the blank COMMON storage began at location 45105_8. Thus the core sizes required for 1/2 bws of 50, 60, 70, 80 would be 144000_8, 200000_8, 237000_8, 304000_8 respectively.

A CDC 6400 computer will print a message saying that the size of blank COMMON exceeds the size of core that has been requested and that the blank COMMON has been truncated. This is because the program has been dimensioned for a maximum 1/2 bw of 400; a smaller 1/2 bw will use less of the blank COMMON area so that the length of the blank COMMON can safely be truncated.

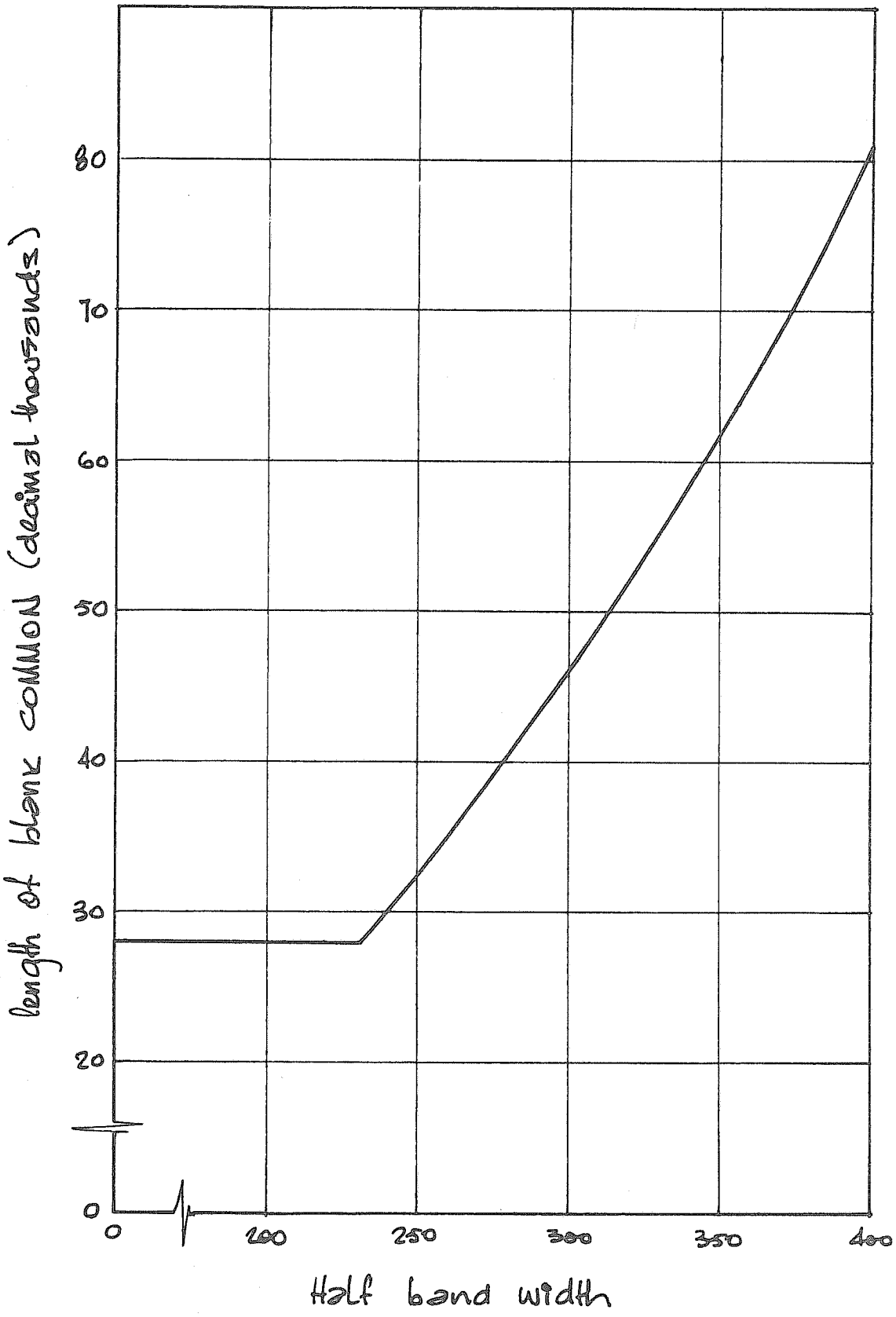


FIG. 6.1 REQUIRED BLANK COMMON SIZE

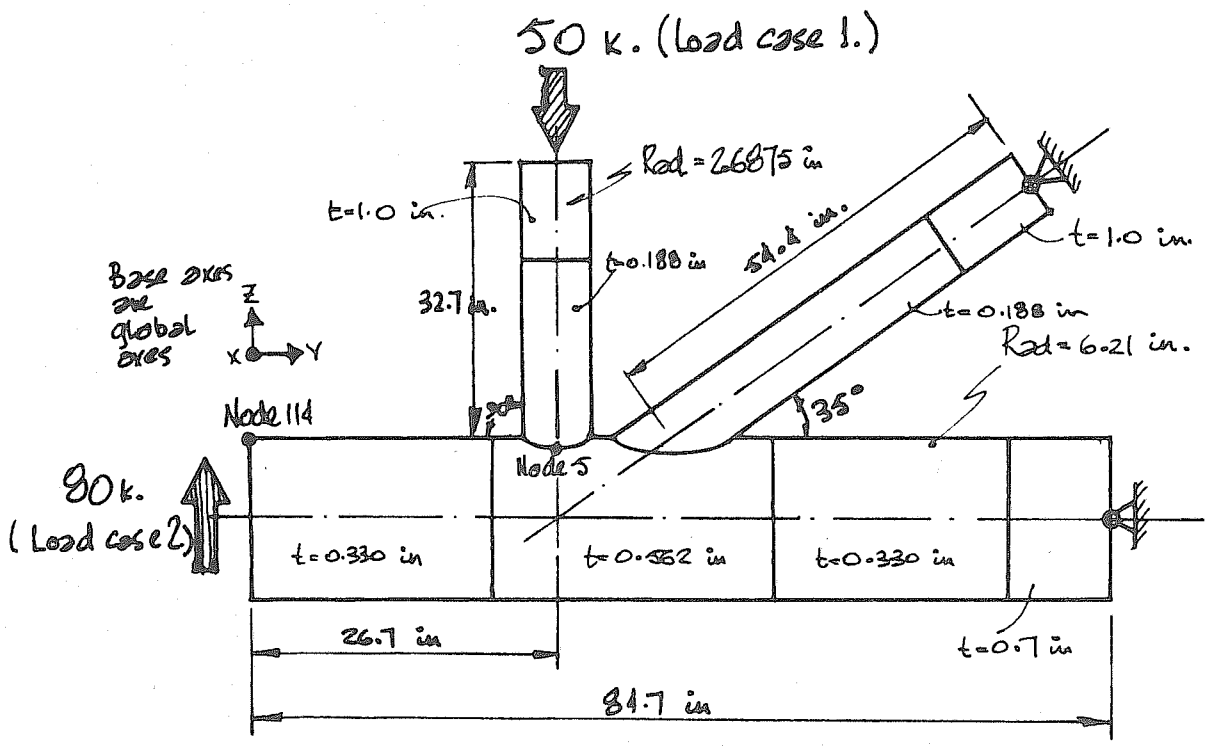


FIG. 7.1 EXAMPLE 1

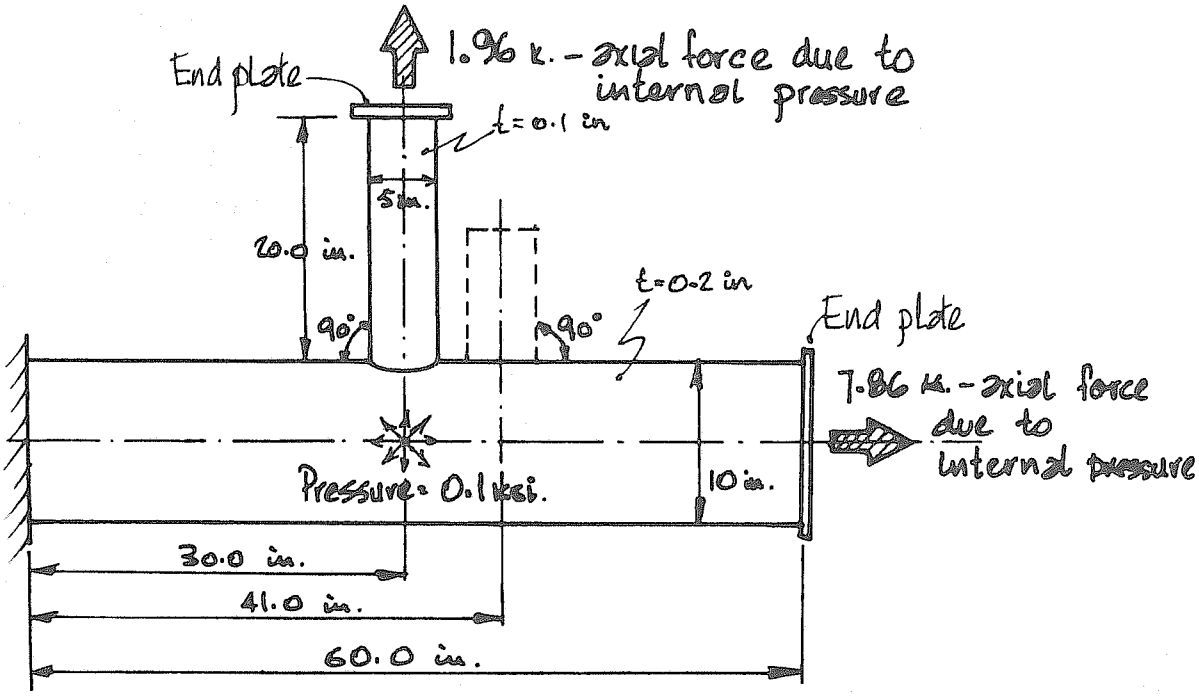


FIG. 7.2 EXAMPLE 2

7. EXAMPLES OF INPUT DATA PREPARATION

Example 1

Fig. 7.1 shows a K joint which was analyzed with a COARSE mesh as a test case of the program. The chord is basically 0.330 in. thick with a local thick wall insert of 0.562 in. The branches are 0.188 in. thick. The increased thicknesses at the branch ends and at the right end of the chord represent an approximation of the end connecting plates. In the first computer run the MESHONLY option was used and the mesh was plotted together with node and element numbers. The input data for this follows.

Card 1. (12A6) EXAMPLE 1

Card 2. (A10, I5) (1)MESHONLY (15)0

Card 3. (6A10) (1)CHORDM (11)PLUG1 (21)PLUG2 (31)BRANCH1 (41)BRANCH2 (51)COARSE

Card 4. (5F10.0) 6.21 2.6875 84.7 32.7 54.4

Card 5. (4F10.0) 26.7 26.7 90.0 35.0

Cards 6, 7, 8. Blank

Card 9. (5F10.0) 9.0 1.28 1.28 3.895 3.895

Card 10. Blank

Card 11. (9I5, 10X, A10) 0 0 1 2 0 0 1 1 1

Card 12. (A5, 4I5) (1)SYM 3 4 11 8

End of data.

Having obtained the numbered mesh, the elements whose thicknesses were to be modified from the basic values of 0.188 in. (branch) and 0.330 in. (chord) were identified. No modifications to the boundary conditions were made. The K joint was actually analyzed only for load case number 1 (Fig. 7.1) but for illustrative purposes the input data shows two load cases. The members were all long enough so that the error flag check never occurred. Therefore, in the analysis run the word CONTINUE on card 11 was omitted. The input data for the

analysis run now follows.

Card 1. (12A6) EXAMPLE 1 - ANALYSIS

Card 2. (A10, I5) (1) ANALYZE (15) 0

Card 3. (6A10) (1) CHORDM (11) PLUG1 (21) PLUG2 (31) BRANCH1 (41) BRANCH2 (51) COARSE

Card 4. (5F10.0) 6.21 2.6875 84.7 32.7 54.4

Card 5. (4F10.0) 26.7 26.7 90.0 35.0

Card 6. (3F10.0) 29000. 0.3 0.330

Card 7. (3F10.0) 29000. 0.3 0.188

Card 8. (3F10.0) 29000. 0.3 0.188

Cards 9, 10. Blank

Card 11. (9I5, 10X, A10) 0 0 1 2 0 0 0 0 1

Card 12. (A5, 4I5) (1) SYM 3 4 11 8

Card 13. (F10.0) 0.0

Card 14. (6F10.0) - - (21-30) -50.0

Cards 15, 16, 17. Blank

Load case number 1

Cards 18, 19. Blank

Card 20. (6F10.0) - (11-20) +80.0

Card 21. Blank

Load case number 2

This ends load input - following cards list thickness modifications.

Card 22.	(I5)	1			Substructure 1
Card 23.	(2I5, F10.0)	1	19	0.562	(plug 1)
Card 24.	(I5)	1			Substructure 2
Card 25.	(2I5, F10.0)	1	20	1.000	(branch 1)
Card 26.	(I5)	1			Substructure 3
Card 27.	(2I5, F10.0)	1	21	0.562	(plug 2)
Card 28.	(I5)	1			Substructure 4
Card 29.	(2I5, F10.0)	1	32	1.000	(branch 2)

Card 30.	(I5)	6			
Card 31.	(2I5, F10.0)	1	39	0.562	Substructure 5 (chord)
Card 32.	(2I5, F10.0)	42	60	0.562	
Card 33.	(2I5, F10.0)	64	69	0.562	
Card 34.	(2I5, F10.0)	75	79	0.562	Substructure 5 (chord)
Card 35.	(2I5, F10.0)	87	91	0.562	
Card 36.	(2I5, F10.0)	102	110	0.700	

This completes the data deck.

The analysis for load case 1 gave the following displacements for node 114 (see Fig. 7.1) of the main structure:

DELTA X = 0.0

DELTA Y = -0.000671302 in.

DELTA Z = -0.0961359 in.

For load case 1, node 5 (main structure) the following moments and membrane forces were obtained.

MX = -0.943913 K. in./in.

MY = -0.488576 K. in./in.

NX = -4.58929 K./in.

NY = -1.98186 K./in.

Example 2

This example presents the input data required for a T joint (Fig. 7.2). As mentioned in Section 2.5 an imaginary second branch must be provided for the purpose of generating a mesh for the chord. In this case branch 2 is the imaginary branch (although branch 1 could have been chosen). The angle θ_2 is arbitrarily chosen to be 90°. Plug 1 in this case is absent -- i.e., the chord has a cut-out in the branch intersection region. Note, however, that plug 2 is present.

The loading considered consists of a uniform pressure load. The chord is cantilevered at the left end and plates are fixed to the chord right end and the branch end. Thus axial forces (whose magnitudes are shown in Fig. 7.2) are introduced in the chord and the branch. The axial forces are to be superimposed on the pressure load forces which are generated by the program. The number of load cases is, therefore, one. The input data is as follows:

Card 1.	(12A6)	EXAMPLE 2				
Card 2.	(A10, I5)	(1) ANALYZE	(5) 0			
Card 3.	(6A10)	(1) CHORDM	-	(21) PLUG2	(31) BRANCH1	- (51) MEDIUM
Card 4.	(5I0.0)	5.0	2.5	60.0	20.0	
Card 5.	(4F10.0)	30.0	41.0	90.0	90.0	
Card 6.	(3F10.0)	30000.	0.3	0.2		
Card 7.	(3F10.0)	30000.	0.3	0.1		
Card 8.	(3F10.0)	30000.	0.3	0.1		
Card 9.	(5F10.0)	(1-10) 9.0	-	(21-30) 2.62	(31-40) 4.5	
Card 10.	Blank					
Card 11.	(9I5, 10X, A10)	0 0 0 1 1 1 1 1 1				
Card 12.	(A5,4I5)	(1) SYM	11	- 1	11	
Card 13.	(F10.0)	0.1				

Card 14.	(6F10.0)	-	-	(21-30) +1.96	
Card 15.	Blank				
Card 16.	Blank				
Card 17.	(6F10.0)	-	-	(21-30) +7.86	
	End of data				

Example 3

This example illustrates the use of input card 3. Suppose that for the T joint shown in Fig. 7.2 only the mesh for the branch was to be generated. The input data would then have the following form:

Card 1.	(12A6)	EXAMPLE 3				
Card 2.	(A10,I5)	(1) MESHONLY	0			
Card 3.	(6A10)	-	-	-	(31) BRANCH1	- (50) MEDIUM
Card 4.	(5F10.0)	5.0	2.5	60.0	20.0	-
Card 5.	(4F10.0)	30.0	41.0	90.0	90.0	

Note that although only the branch mesh is being generated the full geometry of the joint must be described.

Cards 6,7,8		Blank								
Card 9.	(5F10.0)	-		-		-		(31-40) 4.5		
Card 10.	Blank									
Card 11	(9I5, 10X, A10)	0	0	0	0	0	0	1	1	0
Card 12.	(A5, 4I5)	(1) SYM		11		-		-		-

End of data

REFERENCES

1. Greste, O.I., "Finite Element Analysis of K Joints", to be published as a Structural Engineering Laboratory Report, University of California, Berkeley.
2. Johnson, C.P., "The Analysis of Thin Shells by a Finite Element Procedure," Structural Engineering Laboratory Report, No. 67-22, University of California, Berkeley, 1967.
3. Clough, R.W., and Tocher, J.L., "Finite Element Stiffness Matrices for the Analysis of Plate Bending", Proceedings, Conference on Matrix Methods in Structural Mechanics, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, Oct. 1965.

APPENDIX 1. Member End Support Types

This section describes the types of support conditions that the program has provided for at the ends of the chord and the branches. At each of the four ends, three groups of nodes are identified. With reference to Fig. A1 these groups are

(i) Nodes on the plane of geometric symmetry,

(ii) Nodes other than those in (i) but excluding the node which is closest to midway between the two extreme nodes,

(iii) The node which is closest to midway between the two extreme nodes.

The groups (i), (ii), (iii) are identified in Fig A1 by the numbers 1, 2, 3. An explanatory note is required for the group 3 node. For the branches all four mesh types (coarse, medium, fine, xfine) have an even number of equal divisions at the ends. Thus a node is located exactly midway between the two extreme nodes. For the chord, however, the coarse, medium and xfine meshes have an odd number of equal divisions at the ends. The node closest to midway between the extreme end nodes is shown to be the one which is further from the global origin (see Fig. A2).

All nodes of a group are assigned the same boundary conditions but the three groups of boundary conditions may be different. For each node the displacement conditions are represented by a single five digit number from which the constraint conditions for the five displacement degrees of freedom can be recovered. Thus the boundary conditions for each end support type are described by three numbers. These numbers are stored in array IBDTYP (I,J) in which the first subscript is the type identification number and the second is the group number. In the following description of end support types the type identification number is listed against each end support type together with the entries of array IBDTYP.

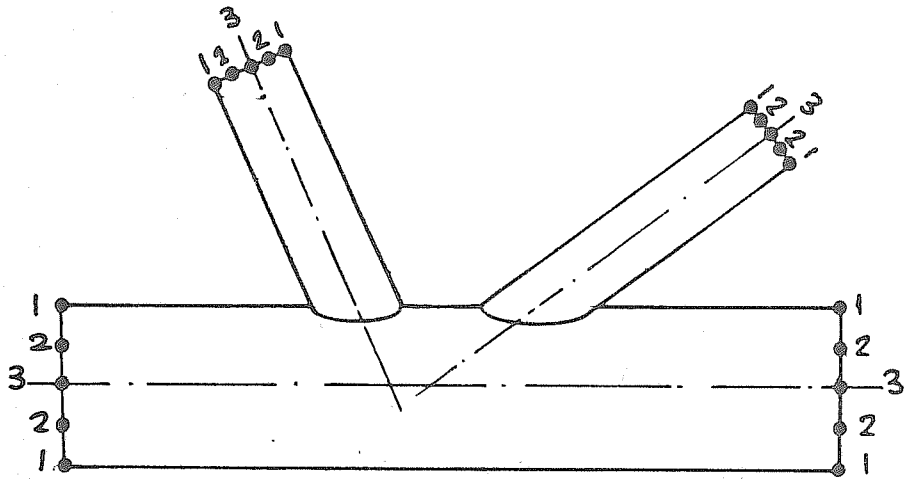


FIG. A1 IDENTIFICATION OF END NODE GROUPS

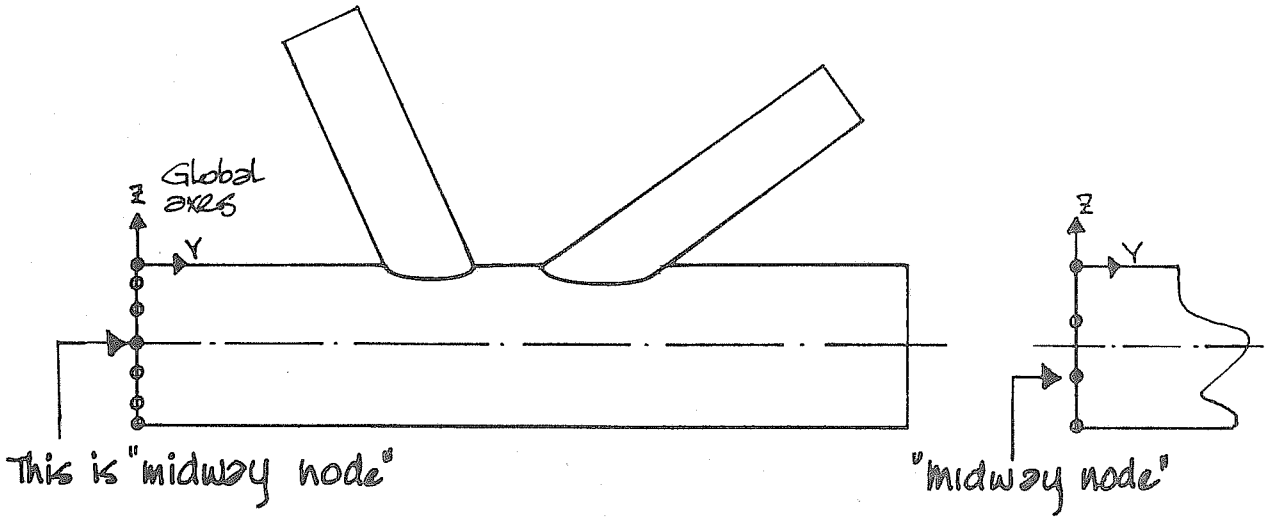


FIG. A2 IDENTIFICATION OF "MIDWAY NODE" ON CHORD END

Types of End Supports

The types of support conditions considered are

(i) free end

(ii) fixed end

(iii) diaphragm at end with different types of roller and hinge constraints.

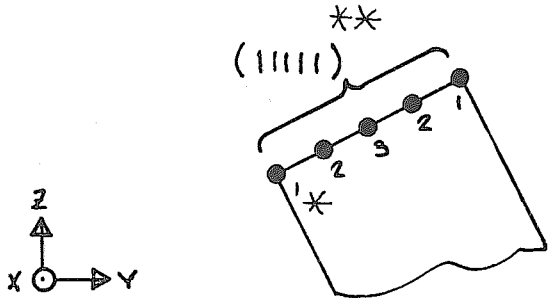
The diaphragm is assumed to have infinite stiffness in its own plane and zero stiffness in bending.

1.0 Symmetric loading.

1.1 Global base axes.

1.11 Branch 1 or 2.

Type 1.111 All nodes fixed.

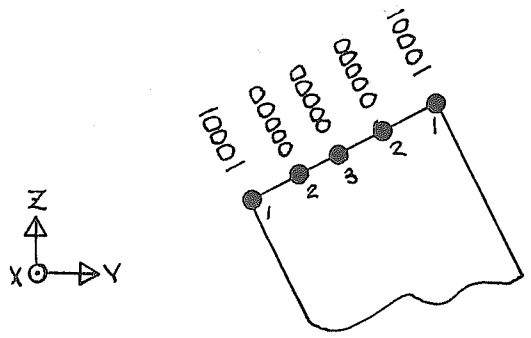


IDENTIFICATION NO. 1

Array IBDTYP

Col. 1	Col. 2	Col. 3
11111	11111	11111
		***

Type 1.112 All Nodes free



IDENTIFICATION NO. 2

10001	00000	00000

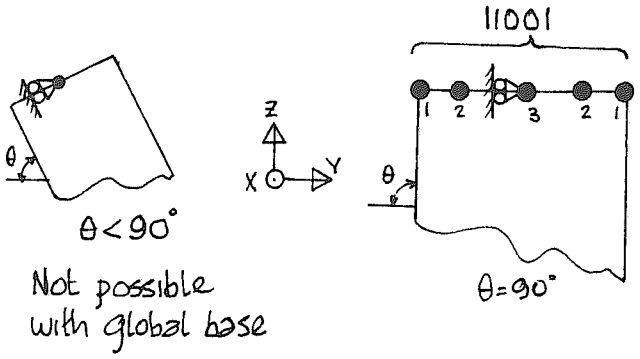
---

* The numbers here represent the groups of nodes previously discussed. They are not node numbers.

** The five degrees of freedom are listed in the order D1, D2, D3, D4, D5. See section 10.

*** The five constraint conditions are stored in array IBDTYP as a single five digit number. The first digit (the leftmost one) gives the constraint for D1; the second digit gives the constraint for D2, and so on.

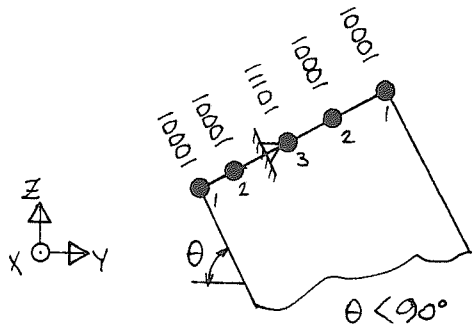
Type 1.113 Diaphragm



IDENTIFICATION NO. 3

11001	11001	11001

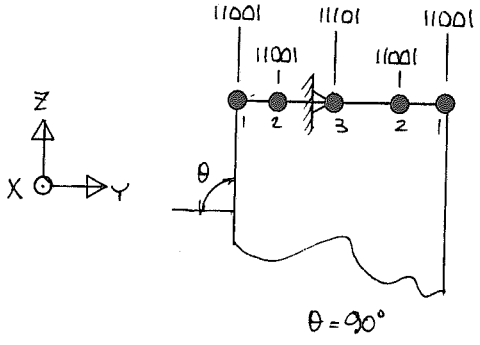
Type 1.114 Diaphragm



IDENTIFICATION NO. 4

10001	10001	11101

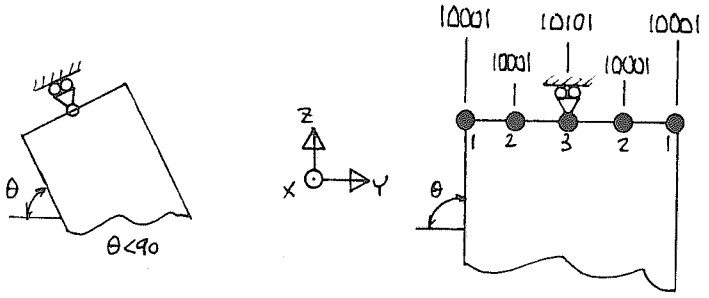
Note: The diaphragm condition is only approximated for $\theta < 90°$. Exact representation is possible if $\theta = 90°$.



IDENTIFICATION NO. 5

11001	11001	11101

Type 1.115 Diaphragm



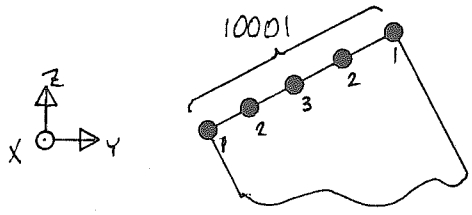
IDENTIFICATION NO. 6

10001	10001	10101

Not possible with global base for $\theta<90°$

Approximate diaphragm representation possible for $\theta=90°$

Type 1.116 Diaphragm - free end

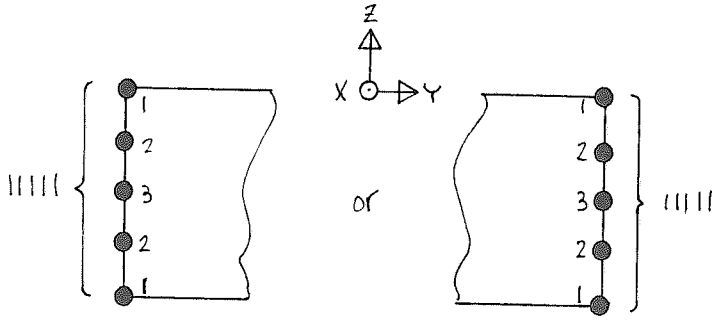


IDENTIFICATION NO. 11

10001	10001	10001

1.12 chord left or chord right end

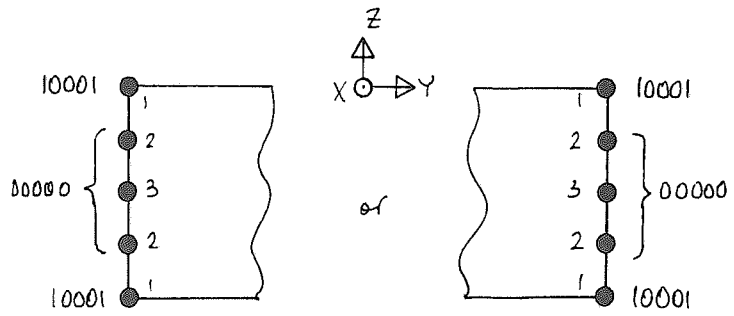
Type 1.121 all nodes fixed



IDENTIFICATION NO. 1

11111	11111	11111

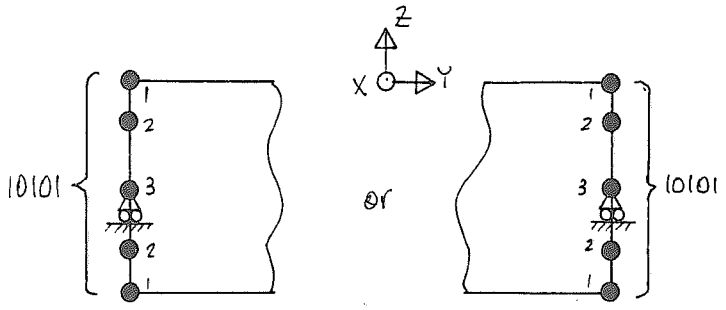
Type 1.122 All nodes free



IDENTIFICATION NO. 2

10001	00000	00000

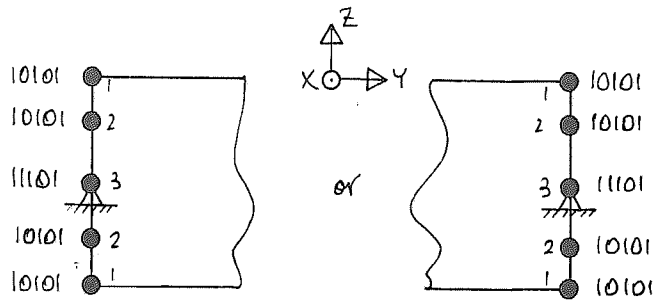
Type 1.123 Diaphragm



IDENTIFICATION NO.,7

10101	10101	10101

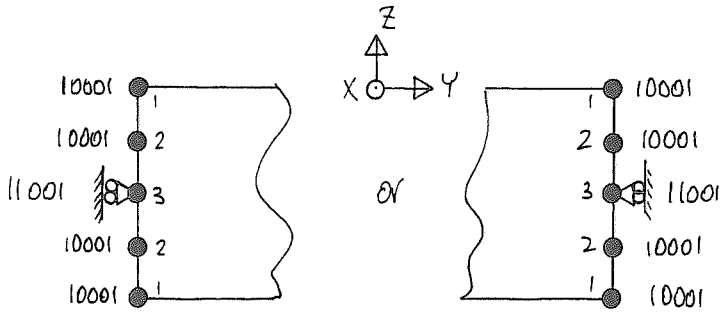
Type 1.124 Diaphragm



IDENTIFICATION NO. 8

10101	10101	11101

Type 1.125 Diaphragm

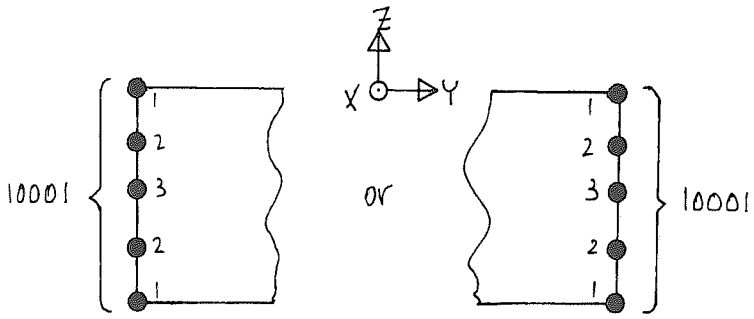


IDENTIFICATION NO. 9

10001	10001	11001

This case is approximated only

Type 1.126 Diaphragm with free end



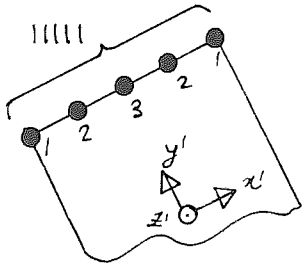
IDENTIFICATION NO. 11

10001	10001	10001

1.2 Tangent Base Axes

1.21 Branch 1 or 2

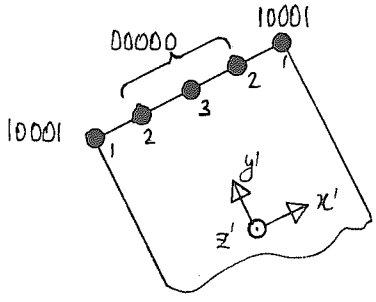
Type 1.211 - All Nodes Fixed



IDENTIFICATION NO. 1

11111	11111	11111

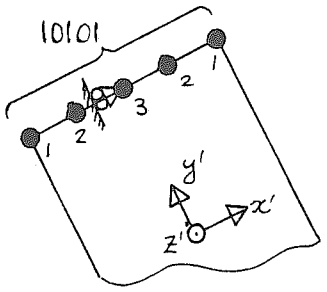
Type 1.212 All Nodes Free



IDENTIFICATION NO. 2

10001	00000	00000

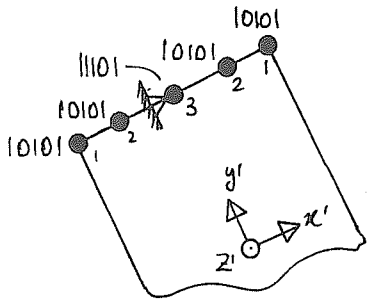
Type 1.213 Diaphragm



10101	10101	10101

IDENTIFICATION NO. 7

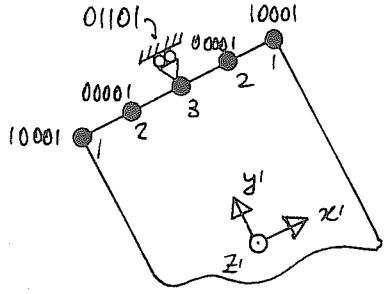
Type 1.214 Diaphragm



10101	10101	11101

IDENTIFICATION NO. 8

Type 1.215 Diaphragm



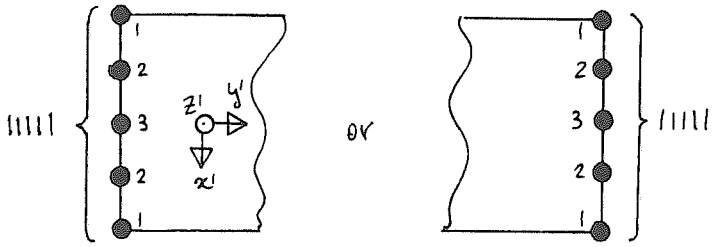
10001	00001	01101

IDENTIFICATION NO. 10

This condition is approximated only.

1.22 Chord left or chord right end

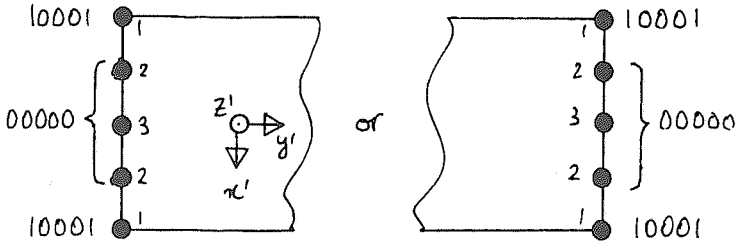
Type 1.221 All nodes fixed



11111	11111	11111

IDENTIFICATION NO. 1

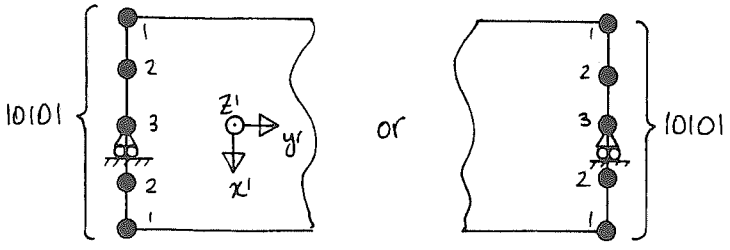
Type 1.222 All nodes free



10001	00000	00000

IDENTIFICATION NO. 2

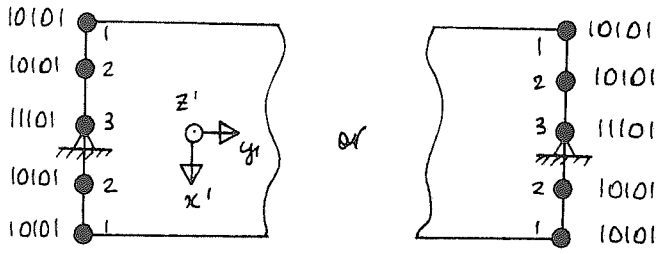
Type 1.223 Diaphragm



10101	10101	10101

IDENTIFICATION NO. 7

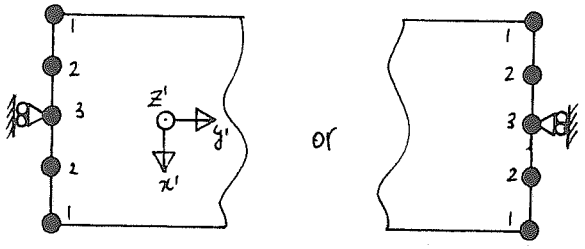
Type 1.224 Diaphragm



10101	10101	11101

IDENTIFICATION NO. 8

Type 1.225 Diaphragm



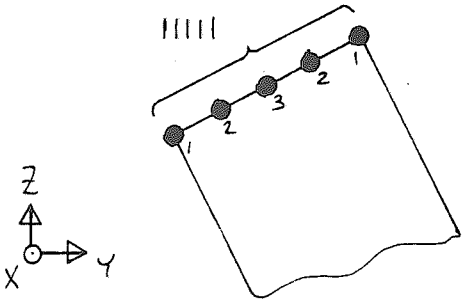
This condition is not possible with tangent coordinates.

2.0 Anti-symmetric loading

2.1 Global base axes

2.11 Branch 1 or 2.

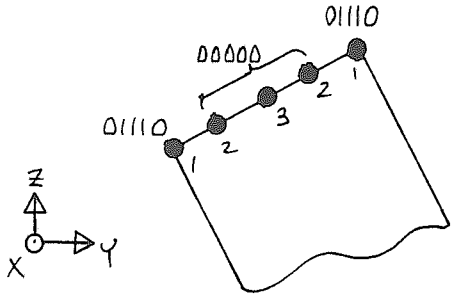
Type 2.111 All nodes fixed.



11111	11111	11111

IDENTIFICATION NO. 1

Type 2.112 All nodes free

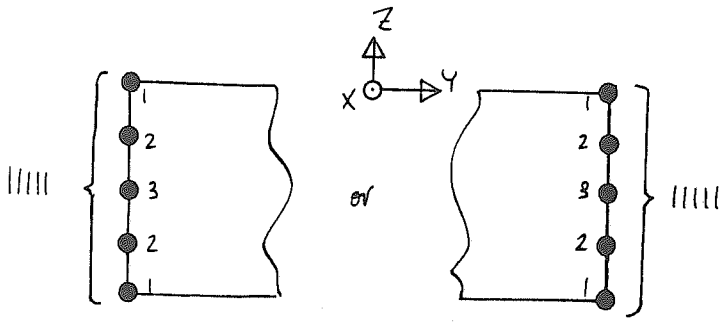


01110	00000	00000

IDENTIFICATION NO.12

2.12 Chord left or chord right end

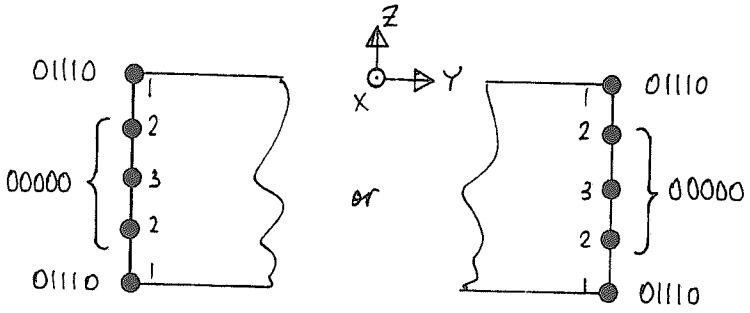
Type 2.121 All nodes fixed.



IIIII	IIIII	IIIII

IDENTIFICATION NO. 1

Type 2.122 All nodes free.



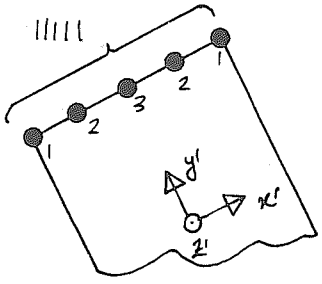
01110	00000	00000

IDENTIFICATION NO. 12

2.2 Tangent base axes

2.21 Branch 1 or 2

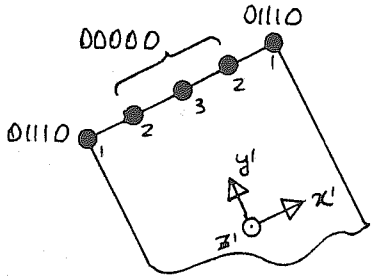
Type 2.211 All nodes fixed



IIIII	IIIII	IIIII

IDENTIFICATION NO. 1

Type 2.212 All nodes free

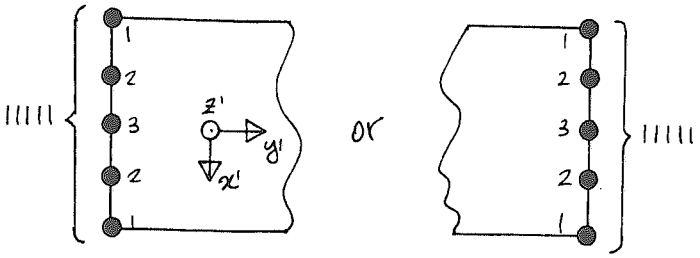


01110	00000	00000

IDENTIFICATION NO. 12

2.22 Chord left or chord right end

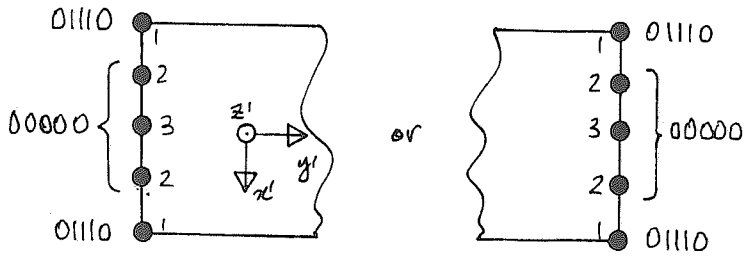
Type 2.221 All nodes fixed



11111	11111	11111

IDENTIFICATION NO. 1

Type 2.222 All nodes free



01110	00000	00000

IDENTIFICATION NO. 12

Modifications to Expand List of Types

The program user can add to the foregoing list of support types. He should identify the displacement constraint conditions for the three groups of nodes, assign a type identification number, and enter the three numbers which describe the constraints into array IBDTYP. The entries of array IBDTYP are set up in a DATA declaration in SUBROUTINE KMESH. The preceding list uses identification numbers 1 to 12 -- i.e., the list occupies the first twelve rows of array IBDTYP. The array is dimensioned for a total of 20 rows and 3 columns.

APPENDIX 2. Description of plotting subroutines.

Plotting of the mesh is done in SUBROUTINE MSPLOT. The central part of this subroutine is the setting up of an array (array ITAG) which lists for each node I, those nodes which are connected to I and whose node numbers are greater than I. The mesh plot itself is produced by going to each node in turn and drawing lines from that node to those adjacent connected nodes contained in ITAG.

The following features of MSPLOT are associated with the U.C. plot system.

(i) Logical units 98 and 99 must be specified on the main overlay PROGRAM card.

(ii) COMMON /CCPOOL/ sets up communication between MSPLOT and the plotting system.

(iii) SUBROUTINES CCBGN, CCALTR, CCPLOT, CCNEXT and CCEND are peculiar to the UC plot system.

The use of /CCPOOL/ and the above mentioned subroutines is described in the pages which follow. From that description the user should be able to decide what alterations are required in MSPLOT to adapt it to the plotting system at hand.

In addition to MSPLOT a FUNCTION SUBPROGRAM, FUNCTION KODE, is used in plotting node and element numbers. This FUNCTION is part of the program. Its purpose is to convert the integer format of the node and element numbers to alphanumeric format. The reason for doing this is that SUBROUTINE CCALTR, which is used for plotting the node and element numbers, can only plot alphanumeric information and not integer numbers. The actual format conversion in FUNCTION KODE is done by the ENCODE/DECODE statements. These statements are a part of CDC 6400 FORTRAN but may not be available for other computers.

The following pages contain a description of the use of SUBROUTINES CCBGN, CCALTR, CCPLOT, CCNEXT and CCEND and COMMON /CCPOOL/.

PAPER COORDINATES

* Paper coordinates are used to specify an "absolute" location on the graph paper. The "paper origin" is taken to be the position of the CAL-COMP plotter pen on the paper at the beginning of the plot. The 1401-plotter operator will initially position the pen (on a clean piece of plotter paper) as far toward the lower tear strip (minus y direction) as the pen will safely move. Thus, unless the user gives the operator special instructions regarding the initial position of the pen, this point near the lower tear strip will become the paper origin. If more than one graph is to be plotted, a call to CCNEXT (details follow) will reposition the pen clear of the previous graph and establish a new paper origin.

* Once the paper origin is established, the point lattice described above, consisting of all points "reachable" by the plotter pen, is superimposed upon the paper. The horizontal or vertical distance between adjacent lattice points is equal to .01 inch, or the distance covered by one horizontal or vertical elemental pen movement. The coordinates of a point in the lattice are defined to be the number of elemental pen movements, in the plus x and plus y directions, respectively, necessary to reach the point from the paper origin. A point 3-1/2 inches to the right and 1-1/4 inches above the initial pen position is denoted in paper coordinates as the point (350.,125.) Effectively a scale of 1 inch=100 units in paper coordinates is established. Negative x and y coordinates would of course indicate that the point in question lies below and to the left of the initial pen position.

* Assuming the standard initial pen position, all plotting should be confined to a rectangle 10-3/4 inches high and 102 inches long, defined in paper coordinates by the vertices: (-200,5),(10000,5), (10000,1080),(-200,1080). Since the y-excess prohibiting microswitches are apparently only accurate to about .05", it is prudent to stay about 5 lattice points away from the top and bottom margins --hence, the values 5 and 1080 as bounds. The limit of 102 inches for graph length is arbitrary, but should be more than adequate.

The values of CCXMIN, CCXMAX, CCYMIN and CCYMAX in the COMMON block /CCPOOL/ are specified in paper coordinates, as is the starting point for lettering in CCALTR. Nevertheless, the concept of paper coordinates may be avoided altogether if the user does not object to being restricted to a 10" by 10" graph. (See section on /CCPOOL/.)

J5 BC XYPL DESCRIPTION

J5 BC XYPL is a MAP subroutine which may be called by FORTRAN IV or MAP programs. This subroutine package allows the user to do initialization and termination tasks for plotting on the CAL-COMP PLOT TAPE; plot curves, alpha-numeric characters, frame and grid lines; and define new origins for plotting when desired.

COMMON /CCPOOL/ XMIN,XMAX,YMIN,YMAX,CCXMIN,CCXMAX,CCYMIN,CCYMAX

Points may be scaled for plotting by specifying the eight scaling numbers in the labeled COMMON block /CCPOOL/. Here XMIN,YMIN are two numbers which are less than or equal to the smallest X and Y in the array of data to be plotted and XMAX,YMAX are larger than or equal to the largest X and Y. The numbers XMIN,XMAX,YMIN,YMAX are preset by the program to 0.,1000.,0. and 1000., respectively, and should be redefined whenever these values are not suitable.

The rectangle described by XMIN,XMAX,YMIN,YMAX is mapped into the rectangle on the paper defined by

CCXMIN,CCXMAX,CCYMIN,CCYMAX as follows, where (X,Y) is a data point to be plotted:

$$CX = CCXMIN + (CCXMAX-CCXMIN) * (X-XMIN)/(XMAX-XMIN)$$
$$CY = CCYMIN + (CCYMAX-CCYMIN) * (Y-YMIN)/(YMAX-YMIN).$$

CCXMIN,CCXMAX,CCYMIN,CCYMAX and the scaled point (CX,CY) are defined in terms of paper coordinates. CCXMIN,...,CCYMAX are preset to 70.,1070., 80.,1080., respectively, to yield the largest possible "labeled" (see CCGRID) square frame (10" by 10") to be safely drawn in the 11 inches between tear strips. Thus, any user who is satisfied with a 10" by 10" graph need not assign any values to CCXMIN,...,CCYMAX in his program. A longer graph can be obtained very simply by adding 100. to CCXMAX for every additional inch of length (beyond 10 inches) desired, and leaving CCXMIN,CCYMIN,CCYMAX alone.

Of course the user may redefine any or all of the scaling numbers to suit his needs. It is possible to obtain different scales for different curves on the same graph by changing the values of XMIN ... YMAX between the plotting of the curves, but in general caution should be exercised when changing scaling numbers in the midst of plotting a graph.

Caution: Following FORTRAN IV rules, the labeled COMMON block /CCPOOL/ must be exactly 8 words long.

CALL CCBGN

Initializes usage of the CAL-COMP PLOT TAPE and CAL-COMP INTERNAL FILE (FORTRAN Unit 98) and defines the present pen position (set by the operator) as the origin (0,0) of the paper coordinates. As described previously, the pen is normally positioned as far as it will move down to the lower edge of the paper. In addition, CCBGN heads the first frame with a line in the Y-direction containing the users job number,

name, the date and start time on the 7094. CCBGN then redefines the (0,0) point to be 6.2 inches to the right of the operator defined (0,0) point.

$$\text{CALL} \left\{ \begin{matrix} \text{CCLTR} \\ \text{CCALTR} \end{matrix} \right\} (x,y,kvtup,ksize \left[\begin{matrix} \text{, nH text} \\ \text{, nH text, n of ch} \\ \text{, chars} \\ \text{, chars, n of ch} \end{matrix} \right])$$

The only distinction between the entries CCLTR and CCALTR lies in the interpretation of the parameters "x" and "y".

"x" and "y" specify the coordinates of the lower left corner of the character raster of the first alphanumeric character to be plotted, assuming that the characters are plotted in the plus x direction. If plotting in the plus y direction, the raster is rotated counter-clockwise 90° about the lower left corner.

CCLTR requires that "x" and "y" be specified in inches from the lower left vertex of the frame specified in CCPOOL. CCLTR facilitates the composition of lettering in relation to the chosen frame.

CCALTR requires that "x" and "y" be specified in paper coordinates, thus providing the ability to specify the location of the lettering "absolutely" on the paper.

"krtup" = 0 or an even integer: the characters are plotted in the "plus x" direction, facing right.

= 1 or an odd integer: the characters are plotted in the "plus y" direction, facing up.

"ksize": The character raster is magnified "ksize" times to 7*ksize hundredths of an inch by 10*ksize hundredths of an inch. Each character (facing right) is defined by connecting points in a 7 x 10 raster of which only 5 x 7 in the lower left corner are actually used. The extra points allow for automatic spacing of the characters as a typewriter does. Each elemental pen motion is repeated "ksize" times to magnify the size of the character. (If the character faces up, the raster is rotated counter-clockwise 90° about the lower left corner).

The string of characters to be plotted may be specified in three different ways:

1. "nHtext": Specifies "n" consecutive alphanumeric characters ("text") to be plotted. The FORTRAN IV compiler packs the consecutive characters 6 to a word, filling out the last word with blanks if "n" is not a multiple of 6, and stores the words in successively higher core storage locations, the last word followed by a word of 12 octal sevens. Since CCLTR interprets the word of all sevens as a flag terminating the string of characters, the actual number of characters plotted (including terminating blanks) will be a multiple of 6, unless "n of ch" is specified.

"n of ch": If specified, only the first "n of ch" consecutive characters will be plotted. "n of ch" should not exceed the "n" in "nHtext".

2. "chars": Denotes a FORTRAN IV array of alphanumeric characters, six to a word, in consecutive order from left to right within each word, then to the next successive word in the array. The user must either supply a word of 12 octal sevens in the array to terminate the character string, or specify the 6th argument...

   "n of ch": If the 6th argument "n of ch" is specified, the test for a word of all sevens will be omitted and the first "n of ch" consecutive characters will be plotted.

3. If only 4 arguments are specified, CCLTR assumes a character string to have already been written in the "CAL-COMP INT FILE", defined here as unit 98. CCLTR first "rewinds" unit 98; then reads from unit 98 and plots the successive "records" (i.e., print lines) one "beneath" the other until all "records" have been read. Words of all blanks are discarded from the end of each "record" and, as before, the number of characters plotted in each "record" will always be a multiple of six. Finally, unit 98 is again "rewound" and initialized for future use.

   * The intention is to simulate "writing" on the plotter paper as one would do on the off-line printer via the FORTRAN WRITE statement.

```
e.g.    WRITE (98,1) (I,I=1,3)
      1 FORMAT (3HDEF/I1,I2,10X,I1)
        CALL CCALTR (X,Y,0,KSIZE,3HABC,2)
        WRITE (98,1) (I,I=4,5)
        CALL CCALTR (X+2.*7.*FLOAT(KSIZE),Y,0,KSIZE)
```

   results in the following string plotted in the "plus x" direction starting at (X,Y), the lower left corner of the character "A", where "b" represents "blank".

```
ABDEFbbb
  1b2bbbbbbbbbb3bbbb
  DEFbbb
  4b5bbb
```

Caution: Care should be taken, particularly when lettering in the plus y direction, to insure that the pen does not run off the plot while attempting to execute the lettering instructions. No provision has been made in the routine to prevent this possibility.

It should be noted again that unless "n of ch" is specified the number of characters plotted will always be a multiple of six. It is therefore possible to plot all specified characters correctly and yet run off the plot while plotting the terminating blanks supplied by {CCLTR / CCALTR}.

Caution: The call to CCGRID, except when used with "6HNOLBLS", writes into the internal file and therefore should not intervene between an instruction which writes information into the internal file and the subsequent lettering of that information through a call to CCLTR or CCALTR.

CCGRID may be called before CCLTR or CCALTR, provided the above mentioned difficulty is avoided.

"5HTICKS", etc., control the labeling of the primary tick marks. Primary ticks are labeled while the grid lines intersecting the frame are labeled.

These options are only active on the axes of the grid which have grid line labels already specified.

"5HTICKS" labels the primary tick marks on any axis specified by the normal label option.

"6HXTICKS" labels only the primary tick marks on the x-axis, if 6HLABELS or 6HXLABLS is also specified.

"6HYTICKS" is similar to 6HXTICKS.

"6HNOTICK" doesn't label any primary tick marks. This option is assumed if none of this group is chosen.

CALL {CCPLOT / CCAPLT} (x,y,n [,{4HJOIN / 6HNOJOIN}] [,ns,nth])

The distinction between CCPLOT and CCAPLT is in the interpretation of the parameters "x" and "y". CCPLOT gives "x" and "y" in relation to XMIN, XMAX, YMIN, YMAX, while CCAPLT uses paper coordinates.

All of the points specified by CCPLOT must fall on or within the rectangle specified by the current values of XMIN, XMAX, YMIN, YMAX.

The limits for plotting with CCAPLT are not set by CCXMIN, CCXMAX, CCYMIN, CCYMAX, but are -600., 100000., 0., 2870., respectively. Thus plotting can be done outside the grid rectangle.

Either of the calls plots the "n" data points (x(1),y(1)),... (x(n),y(n)) specified by the FORTRAN IV arrays "x" and "y", connecting each pair of points (x(i),y(i)), (x(i+1),y(i+1)) for i = 1,...,n-1 by the "best" approximation to the straight line joining them and plotting the symbol(s) specified by "ns" at the first point (x(1),y(1)) and every "nth" point thereafter: (x(n+1),y(n+1)), (x(2n+1),y(2n+1)),...

The approximation is "best" in the sense that each point traversed is at least as near the true line as any other candidate point for the same move. The pen will be down (on the paper) when the path connecting pairs of points is traced if the "4HJOIN" option is specified; up (off the paper) if the "6HNOJOIN" option is taken. If neither "ns" nor "nth" is specified, or "ns" = 0, or "nth" = 0, then no symbols are plotted. If n=0, no points or symbols are plotted, but the "4HJOIN", "6HNOJOIN", "ns", "nth" options are set for possible use by CCETC.

"ns" is interpreted as a string of decimal digits, each of which represents a symbol according to the following table. The symbols specified by each digit are superimposed. (e.g., "2345" would produce an asterisk.) Symbols plotted at a point (x,y) occupy a 9x9 point raster whose center corresponds to (x,y). The center of the raster is located at the center of symbols 1-7; the upper vertex of symbol 8; the lower vertex of symbol 9.

0		no symbol
1	.	dot (actually a small diamond)
2	—	horizontal bar
3	|	vertical bar
4	\	bend dexter
5	/	bend sinister
6	◇	diamond
7	□	square
8	△	delta
9	▽	del

CALL CCNEXT

Spaces approximately 6 inches beyond CCXMAX or the rightmost part of any lettering; empties the output buffer; writes an end-of-file mark on the CAL-COMP PLOT TAPE; writes the job number, name, etc., on a line in the y-direction; and redefines the origin (0,0) of the paper coordinates to be 6 inches to the right of this line at the same ordinate value as before.

CCXMAX should not be redefined before calling CCNEXT, because of its use in spacing beyond the just finished graph.

CALL CCEND

Spaces approximately 6 inches beyond CCXMAX or the rightmost part of any lettering; writes the job number, name, etc., on a line in the y-direction; positions the pen 12 inches to the right of this line; empties the output buffer; writes 1 end-of-file mark, writes on the tape "*bbbbbENDbTAPEbbbb", and 2 end-of-file marks on the CAL-COMP PLOT TAPE, signaling the end of the plot tape to the 1401 plotter program; and rewinds and unloads the tape. Hence, no further plotting can be done after this routine has been called.

APPENDIX 3 - OVERLAY DECK SETUP.

The overlay deck setup is listed below by subroutine name.

OVERLAY (KJOINT,0,0)
PROGRAM KGEN

OVERLAY (KJOINT, 1,0)
PROGRAM KMESH
SUBROUTINE CHSETUP
SUBROUTINE PLUG
SUBROUTINE BRANCHB
SUBROUTINE BRANCHC
SUBROUTINE CHORD
SUBROUTINE ZERO
SUBROUTINE JNCORD1
SUBROUTINE JNCORD2
SUBROUTINE BSTRIP
SUBROUTINE SUBCALL
SUBROUTINE ELSTRP1
SUBROUTINE ELSTRP2
SUBROUTINE ELSTRP3
SUBROUTINE ELSTRP4
SUBROUTINE COMSTRP
SUBROUTINE G1
SUBROUTINE G2
SUBROUTINE G3
SUBROUTINE G4
SUBROUTINE G5
SUBROUTINE G6
SUBROUTINE G7
SUBROUTINE G8
SUBROUTINE BOUNDRY
SUBROUTINE BDIS
SUBROUTINE UPDATE
SUBROUTINE JNCT
SUBROUTINE WRITE2
SUBROUTINE LOADS
SUBROUTINE LOCEND
SUBROUTINE UPROW
SUBROUTINE INVERT
SUBROUTINE ITERATE
SUBROUTINE MSPLOT
FUNCTION KODE

OVERLAY (KJOINT, 2,0)
PROGRAM JOINT
SUBROUTINE SETUP
SUBROUTINE SPARSE

The overlay cards are underlined for emphasis.

In each link the subroutines may be in any order provided that the main subroutine i.e., the PROGRAM is the first one.

SUBROUTINE COMONE
SUBROUTINE ODSHEL
SUBROUTINE SCST
SUBROUTINE SLST10
SUBROUTINE SLCT9
SUBROUTINE SUBSPR
SUBROUTINE BLAYER
SUBROUTINE DIRCOS
SUBROUTINE CTRANS
SUBROUTINE GETNRM
SUBROUTINE NLOAD
SUBROUTINE FORCES
SUBROUTINE FORMK
SUBROUTINE FORMS
SUBROUTINE BIGCON
SUBROUTINE STRESS
SUBROUTINE GDISPL
SUBROUTINE MEMBIO
SUBROUTINE MOMENT
SUBROUTINE MOMT
SUBROUTINE MATMU3
SUBROUTINE PRTRI
SUBROUTINE PQUAD
SUBROUTINE PRNSTS

APPENDIX 4. Interpretation of Displacements, Moments, Membrane Forces and Stresses at Pipe Juncture Nodes.

Some points regarding the interpretation of the displacements at the pipe juncture nodes were brought to the writer's attention after this report had already been typed and was ready for reproduction. These points more appropriately belong in the body of the report but are included as an Appendix because it was more convenient to do so.

It was mentioned in Section 3.1 that for pipe juncture nodes two sets of tangent axes are applicable. The displacements which are printed out for the pipe juncture nodes of each substructure are referenced to the tangent axis system corresponding to the chord. Suppose, for example, that the tangent axes are chosen as the base system. Then for the branches, the displacements which are listed for the juncture nodes are referenced to the tangent system for the chord and not the tangent system for the branch. If the global axes are the base system, then the translations for all nodes are referenced to a common system, but the notations for the branch juncture nodes are still referenced to the tangent system for the chord. Note that the above remarks are only for juncture nodes: for all other nodes the tangent system corresponding to the substructure in question is used.

Unlike the displacements, however, the output moments, membrane forces and stresses at the juncture nodes of each substructure are referenced to the tangent axis system corresponding to that substructure and not the tangent system for the chord. That is, the moments, membrane forces and stresses at the juncture nodes of the branches are referenced to the tangent system appropriate for each branch.