LA-UR -77-1345

Conf-770813--3

TITLE: INGEN: A GENERAL PURPOSE MESH GENERATOR FOR FINITE ELEMENT CODES

AUTHOR(S): William A. Cook, Q-13

SUBMITTED TO: ADINA Conference, August 4-5, 1977
Massachusetts Institute of Technology
Cambridge, Massachusetts

(To be published in the proceedings of the ADINA Conference).

los alamos scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

Form No. 836
St. No. 2629
1/75

UNITED STATES
ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

INGEN: A GENERAL PURPOSE MESH GENERATOR FOR FINITE ELEMENT CODES

by

William A. Cook

ABSTRACT

INGEN is a general purpose mesh generator for two- and three-dimensional finite element codes. The basic parts of the code are surface and three-dimensional region generators that use linear-blending interpolation formulae. These generators are based on an i, j, k index scheme which is used to number nodal points, construct elements, and develop displacement and traction boundary conditions.

I. INTRODUCTION

Approximate numerical solutions to mathematical boundary value problems can be calculated using finite element computer codes. These solutions are dependent upon the mesh used to model the geometry of such problems. Since hand generating simple meshes is a very tedious task and generating complicated meshes a very difficult task see, for example Figs. 1 and 2, it is desirable to have a computer code that will generate meshes that are functional for solving problems without losing the flexibility that the finite element codes have. INGEN is a general purpose meshing code to be used for generating both two- and three-dimensional meshes. The intent, in developing INGEN, has been to include as many options as possible and thus be able to generate a variety of meshes and still take full advantage of the finite element method.

The philosophy involved in developing a nodal point mesh is to distribute the nodal points according to the anticipated field variables. For large gradients, the nodal points should be dense, and for small gradients, the nodal points should be sparse. Therefore, it is necessary to make an estimate of how the field variables change in the different regions of the problem when generating a mesh. The user of the INGEN code accomplishes the corresponding mesh grading for this estimate of the gradients by generating the boundary edges of the mesh with the desired spacing of nodal points using the line and circular arc generators and then using surface and volume (three-dimensional region) generators, both of which preserve this spacing. The surface nodal point generator preserves this spacing by using the nodal points as they are distributed along the boundary edges as the criteria for spacing the surface nodal points. Similarly, the volume nodal point generator calculates the interior nodal points using the surface nodal points as the criteria for spacing of the interior nodal points. Both the surface and volume generators use linear-blending interpolation equations as described in Ref. 1 for calculating nodal point coordinates.

II. MODEL

To generate a three-dimensional mesh, study the body to be modeled and determine which regions of this body can be conveniently enclosed by six separate surfaces. These regions shall be called volumes. The interior nodal points of these volumes can be generated using the volume generator when the surface nodal points are known. Next, onto these volumes designate an i, j, k set of indexes that specify in the different nodal points at the corners of these volumes. Corners are designated as those points where three surfaces meet. Choose the indexes such that one index does not change for each surface. The values of these indexes are not critical since they can easily be changed at any time with a built-in expand option. For each of these volumes determine the surfaces that require known nodal points for use by the volume generator. The nodal points on these surfaces can be generated using the surface generator provided the nodal points along the edges of the surface are known. Finally, for each of these surfaces, determine the edges that require known nodal points for use by the surface generator and generate these using the line and/or circular arc generators. No nodal point associated with either a volume, surface, or edge either needs to be input or generated more than once. When all edges, surfaces, and volumes are determined they are then input into the code and calculated by the code in the order of first the edges, second the surfaces, and third the volumes.

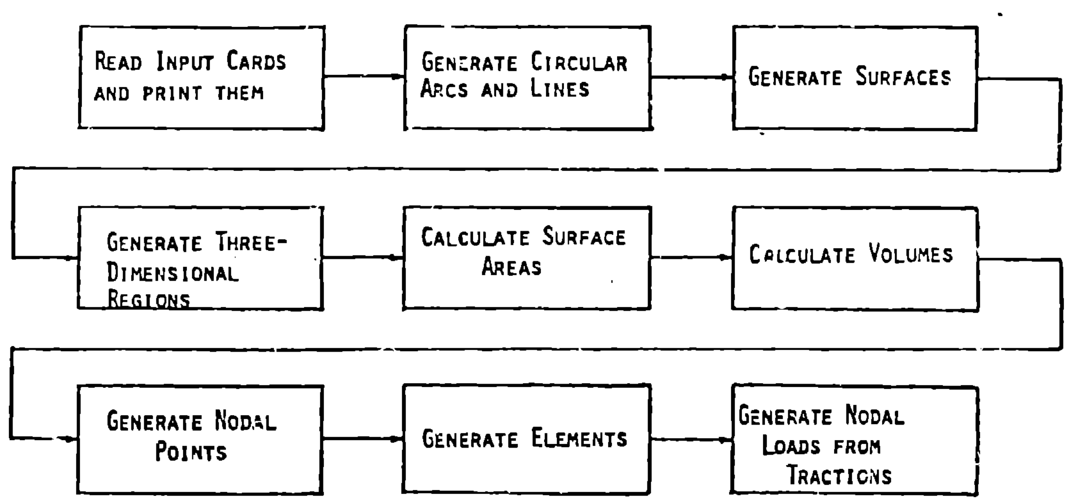
Two-dimensional meshes are developed in a similar manner using the surfaces and edges only.

When boundaries are very irregular, the body being studied may be broken into multiple regions, which have more boundary edges. This will give greater control over the boundary surface nodal point calculations and consequently over interior nodal point calculations.

III. INGEN

The flow of the calculations in the INGEN code are shown in the following chart.

FLOW OF CODE



Special features of INGEN are described in the following sections. INGEN is structured to be compatible with ADINA (see Ref. 7).

Line and Circular Arc Generators

Included in the code is a line generator and a circular arc generator. These are used to generate boundary edge nodal points and if desired can be used to space the nodal points with a geometric progression algorithm. The circular arcs are generated using Eulerian angles as described in Ref. 6.

Boundary Surface Generator

Consider a three-dimensional surface; on the boundary of this surface choose four separate space curves which will serve as boundary edges of this surface (see Fig. 3). Upon this surface a set of nonorthogonal coordinates can be designated which will be called body (normal) coordinates (see Fig. 3). Let these body coordinates (ξ, η) be defined such that along each boundary edge one of the body coordinates is not changing and the other coordinate is the ratio of the distance along that boundary edge to the total length of the boundary edge. One coordinate will always be 0 or 1 and the other coordinate will vary from 0 to 1 and this condition will exist for all four boundary edges. Thus, given a set of nodal points on the boundary edges, the body coordinates can be calculated for each of these nodal points.

To model a surface with a nodal point mesh it is first necessary to designate nodal points along the boundary edges. Assuming these exist, let a set of indexes be defined as follows: let i be a counter used for designating the nodal points along the ξ coordinate and let j be the counter used for designating the nodal points along the η coordinate (see Fig. 4). Then, all nodal points along the boundary edges can be designated with an i,j set of indexes. Thus for every nodal point on the boundary edges, the cartesian coordinates (x,y,z) are known as a function of body coordinates and can be designated with a set of indexes.

The criterion used in determining the interior nodal points on a surface is that the rows and columns of nodal points change from the shape of one boundary edge to the shape of the opposite boundary edge as the rows and columns of nodal points are traversed (see Fig. 5). Using this criterion linear blending interpolation equations can be derived (see Ref. 1). These interpolation equations were first derived by Coons in Ref. 2.

$$
\begin{aligned}
x(\xi,\eta) = {} & (1-\eta)x(\xi,0) + \eta\, x(\xi,1) + (1-\xi)x(0,\eta) \\
& + \xi\, x(1,\eta) - x(0,0)(1-\xi)(1-\eta) - x(1,0)\xi(1-\eta) \\
& - x(0,1)(1-\xi)\eta - x(1,1)\xi\eta
\end{aligned}
$$

Similar equations exist for y and z. Therefore when the boundary edge nodal point coordinates are known the interior nodal point coordinates are known as a function of the body coordinates. There are several ways that the body coordinates can be chosen for the interior nodal points. For example, Gordon and Hall in Ref. 3 allowed the user to input body coordinates. However, in this surface generator the body coordinates are calculated by considering them as a mapping of the original surface onto a unit square. This is easily done since the body coordinates vary from 0 to 1. Again, since the boundary edge nodal points are known as a function of the body coordinates, the location of these points on the unit square are known. Using lines to connect opposite boundary edge nodal points, the interior nodal point body coordinates can be established as the intersection of these lines (see Fig. 6). Once the body coordinates are established for these interior nodal points, the cartesian coordinates can be calculated using the interpolation equations.

An example of a mesh generated with this surface generator is shown in Fig. 5.

Volume Generator

Consider a three-dimensional region. Visualize how such a region may be modeled with six connecting boundary surfaces. For an example, see Fig. 7. Upon this three-dimensional region a set of nonorthogonal coordinates can be designated which will now be called the body coordinates. Note that this is an extension of the body coordinates used for a boundary surface. These body coordinates (ξ,η,γ) can be defined as before as proportional values of the length of the boundary edges. With these coordinates, each boundary surface is a function of two of the three body coordinates with the third being a constant.

Just as the body coordinates were extended in going from a boundary surface to a three-dimensional region, the index scheme can also be extended. In addition to the i and j indexes serving as counters for ξ and η coordinates, let k be a counter for the γ coordinate. Then the surface nodal points will be defined as those nodal points for which one of the indexes does not change. From using the surface generator each nodal point on a boundary surface can be generated; thus each volume boundary nodal point is known as a function of both cartesian coordinates (x,y,z), and body coordinates (ξ,η,γ) and is designated with indexes. The criteria used in determining the interior nodal points is that the surfaces of nodal points change from the shape of one boundary surface to the shape of the opposite boundary surface as the surfaces of nodal points are transversed (note the similarity to boundary surfaces). Using this criterion linear blending interpolation equations can be derived (see Ref. 1).

$$x(\xi,\eta,\gamma) = \tfrac{1}{2}\{(1-\eta)x(\xi,0,\gamma)+\eta x(\xi,1,\gamma)$$
$$+(1-\gamma)x(\xi,\eta,0)+\gamma x(\xi,\eta,1)$$
$$+(1-\xi)x(0,\eta,\gamma)+\xi x(1,\eta,\gamma)\}$$
$$+ C(\xi,\eta,\gamma)$$

where

$$C(\xi,\eta,\gamma) = -\tfrac{1}{2}\{(1-\xi)(1-\eta)(1-\gamma)x(0,0,0)+(1-\xi)(1-\eta)\gamma x(0,0,1)+(1-\xi)\eta(1-\gamma)x(0,1,0)$$
$$+(1-\xi)\eta\gamma x(0,1,1)+\xi(1-\eta)(1-\gamma)x(1,0,0)+\xi(1-\eta)\gamma x(1,0,1)$$
$$+\xi\eta(1-\gamma)x(1,1,0)+\xi\eta\gamma x(1,1,1)\}$$

Similar equations may be defined for $y(\xi,\eta,\gamma)$ and $z(\xi,\eta,\gamma)$. From these equations it can be seen that when the boundary surface nodal point coordinates are known, the interior nodal point coordinates are known as a function of the body coordinates. Again there are several ways that the body coordinates can be chosen for the interior nodal points. In this generator these are calculated by considering the body coordinates as a mapping of the original volume onto a unit cube. This is easily done since the body coordinates vary from 0 to 1. Also, since the boundary surface nodal points are known as a function of the body coordinates, the location of these on the unit cube are known. Using lines to connect opposite boundary surface nodal points (from an index point of view) the interior nodal point body coordinates can be established as the point with the minimum distance to the three lines (see Fig. 8). The details of this approach are explained in reference 1. With the body coordinates established for these interior nodal points the cartesian coordinates can be calculated using the interpolation equations.

<u>i, j, k Index Scheme</u>

The i, j, k index scheme is very useful. This scheme is used for generating nodal points, generating elements, generating boundary conditions (both displacement and traction), eliminating midside nodal points, and calculating surfaces and volumes.

Most of the uses of this index scheme are to set up "DO" loops within the code. However, it also creates control over elements. For example, consider the three-dimensional parabolic element.

(a) When all three indexes are odd the designated point is a corner nodal point.

(b) When two indexes are even and one index is odd there is no nodal point at that point.

(c) When all three indexes are even the designated point is a center point and may or may not be a nodal point as desired.

(d) When two indexes are odd and one even the designated point is a midside nodal point and may be eliminated or centered as desired. Depending upon which index (i, j, or k) is the even index, the midside nodal points may be eliminated on a preferential basis.

Expansion or Contraction Option

One of the most useful options in the INGEN code is the ability to change a mesh without altering all of the input. This option is illustrated in Fig. 9 where the geometrical model on the left was generated and then it was decided to refine the mesh (expansion). The geometrical model on the right was generated. Occasionally it is desirable to make a coarser mesh (contraction); this can also be done in most cases. It is also possible to use contraction for some portions of the mesh and expansion for other portions. Thus there is great flexibility in this option.

The ability to use expansion and contraction on meshes is made possible by using the input of indexes as location counters for index arrays. Thus if there is no expansion or contraction and the index i = 5 is input, the code will use i = 5. However if the i index array is modified with the expansion option when i = 5 is input the code will use the value which has been substituted in the 5th location of the i index array for i.

This option is further illustrated in the following table which was used to generate a coarser (contraction) mesh of the model in Figs. 1 and 2.

TABLE I

Demonstration Model (Fig. 1 and 2)

											1										2										3
Index Number		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
Indexes used for the Original Mesh	i	1				5					0					5					0										
	j	1				5						1									0										
	k	1											2				6		8		0				4			7			0
Indexes used for the	i	1									5					7					9										
	.																														

From the table, notice how the indexes that were used in the input of the original mesh are the only indexes that need to be modified for the modified mesh.

Elements

The following elements can be generated with this code:

(1) Two-dimensional continuum elements with 4 to 8 nodal points.

(2) Three-dimensional continuum elements with 8 to 21 nodal points.

(3) Truss or beam elements with 2 nodal points.

Material properties for these elements are designated with the i, j, k index scheme and can be changed as desired.

IV. EXAMPLE PROBLEMS

Demonstration Model

This example problem is for the geometrics shown in Figs. 1 and 2. The indexes used in generating a mesh of these two bodies and the interface between them are shown in Fig. 10. To generate the mesh of these bodies there were 35 lines and 14 circular arcs generated and there were 23 surfaces and 4 volumes generated. A coarser mesh was also generated as shown in Table I.

Pressurized Concrete Reactor Vessel (PCRV)

The geometric body this mesh represented is essentially the mesh shown in Ref. 4 on page 165. The generated mesh is shown in Fig. 11, which also shows some of the indexes used for this mesh. MOVIE.LASL (Ref. 5) was used to display this mesh. The mesh shown in Fig. 11 was generated with 64 lines, 24 circular arcs, 4 surfaces and 7 volumes. Both a coarser (contraction) mesh and a refined (expansion) mesh was generated from the original mesh. These are compared in Table II. Since displacement and traction boundary conditions are input with indexes, these boundary conditions are generated correctly for both the contraction and expansion.

TABLE II

PCRV MESHES

ITEM	COARSER	ORIGINAL	REFINED
NODES	250	707	3925
ELEMENT (20 Node)	26	96	704
i INDEX	1	1	1
	3	5	9
	5	9	17
j INDEX	1	1	1
	2	2	3
	3	3	5
	5	5	9
	7	7	13
k INDEX	1	1	1
	2	5	7
	3	7	11
	7	11	19
		13	25
	11	19	35
	15	23	43
	16	25	47
	17	29	53

Controlled Thermonuclear Fusion Reactor (CTR) Z Pinch Machine

This mesh was generated as 8 node elements and used the geometric progression option for generating nodal points on the boundary edges. The generated mesh is shown in Fig. 12 along with the indexes that were used. It was generated with 18 lines, 25 circular arcs, 18 surfaces, and 3 volumes. This mesh was used to solve a magnetic potential problem and used truss elements (not shown in Fig. 12) for non-zero potential boundary conditions.

Acknowledgements

The first two versions of this code were developed at Thiokol Corporation with Don Lee and Edwin Dickson doing the programming. Applications of the code to meshing of prestressed concrete reactor vessels was supported by the Nuclear Regulatory Commission's Division of Reactor Safety Research.

References:

1. W. A. Cook, "Body Oriented (Natural) Coordinates for Generating Three-Dimensional Meshes," Int. J. Numerical Methods Eng. 8, 27-43 (1974).

2. S. A. Coons, "Surfaces for Computer-Aided Design of Space Forms," Project MAC, MIT (1964). Revised to MAC-TR-41 (1967). Available through CFSTI, Sills Building, 5285 Port Royal Road, Springfield, Virginia, U.S.A.

3. W. J. Gordon and C. A. Hall, "Geometric Aspects of the Finite Element Method: Construction of Curvilinear Coordinate Systems and Their Applications to Mesh Generation," Research Report, General Motors, Michigan (Sept. 1972).

4. O. C. Zienkiewicz, "The Finite Element Method in Engineering Science," McGraw-Hill, London, 1971.

5. B. E. Brown, "MOVIE.LASL Version 1.0 User's Manual," LA-NUREG-6532-M, Los Alamos Scientific Laboratory, Dec. 1976.

6. H. Goldstein, "Classical Mechanics," (Addison-Wesley Pub. Co., Reading, MASS., 1950).

7. K. J. Bathe, "ADINA, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis," MASS. Inst. of Technology, Cabridge, MASS., Sept. 1975 (revised May 1976).

William A. Cook

FIGURE CAPTIONS

Figure 1 -- Demonstration model

Figure 2 -- Demonstration model

Figure 3 -- Cartesian and body co-ordinate systems for a surface

Figure 4 -- Cartesian and body co-ordinate systems with i, j, indexes

Figure 5 -- Interior nodal points calculated with the surface generator

Figure 6 -- Body co-ordinate system mapped on a unit square

Figure 7 -- Cartesian and body co-ordinate systems for a three-dimensional region

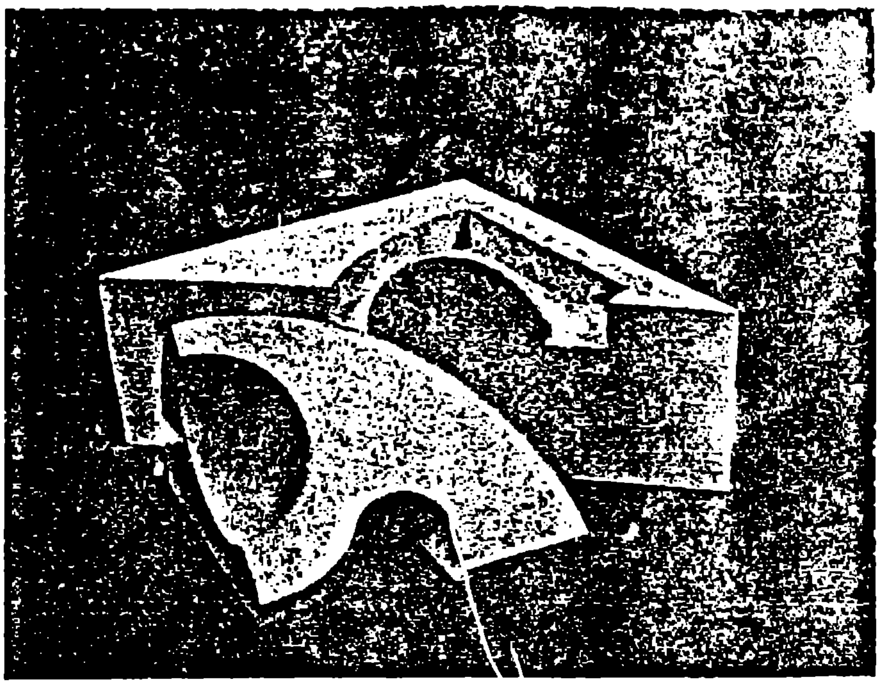
Figure 8 -- Body co-ordinate system mapped on a unit cube

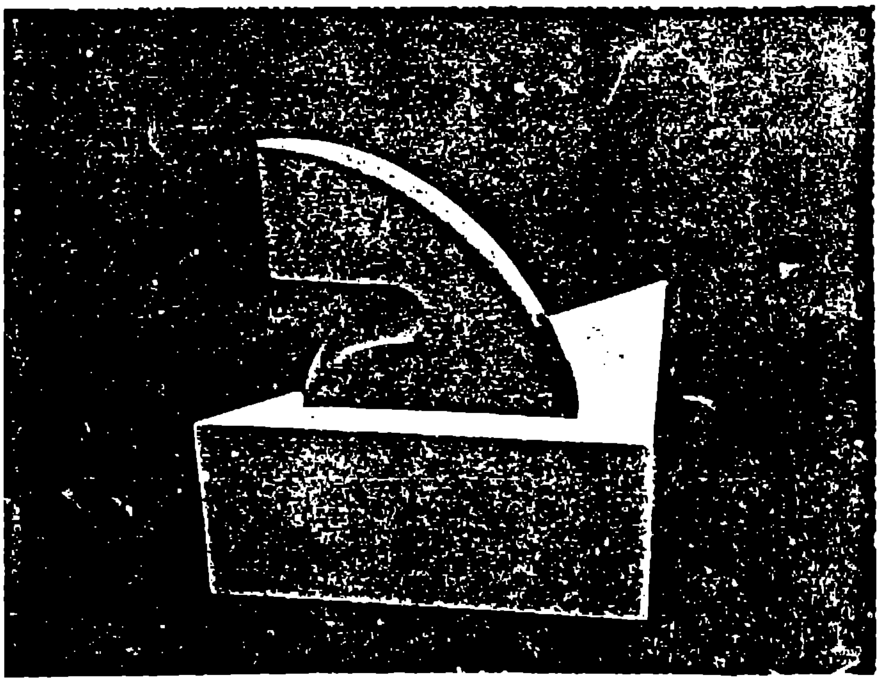
Figure 9 -- Two finite element meshes of the same geometry illustrating the expansion and contraction option

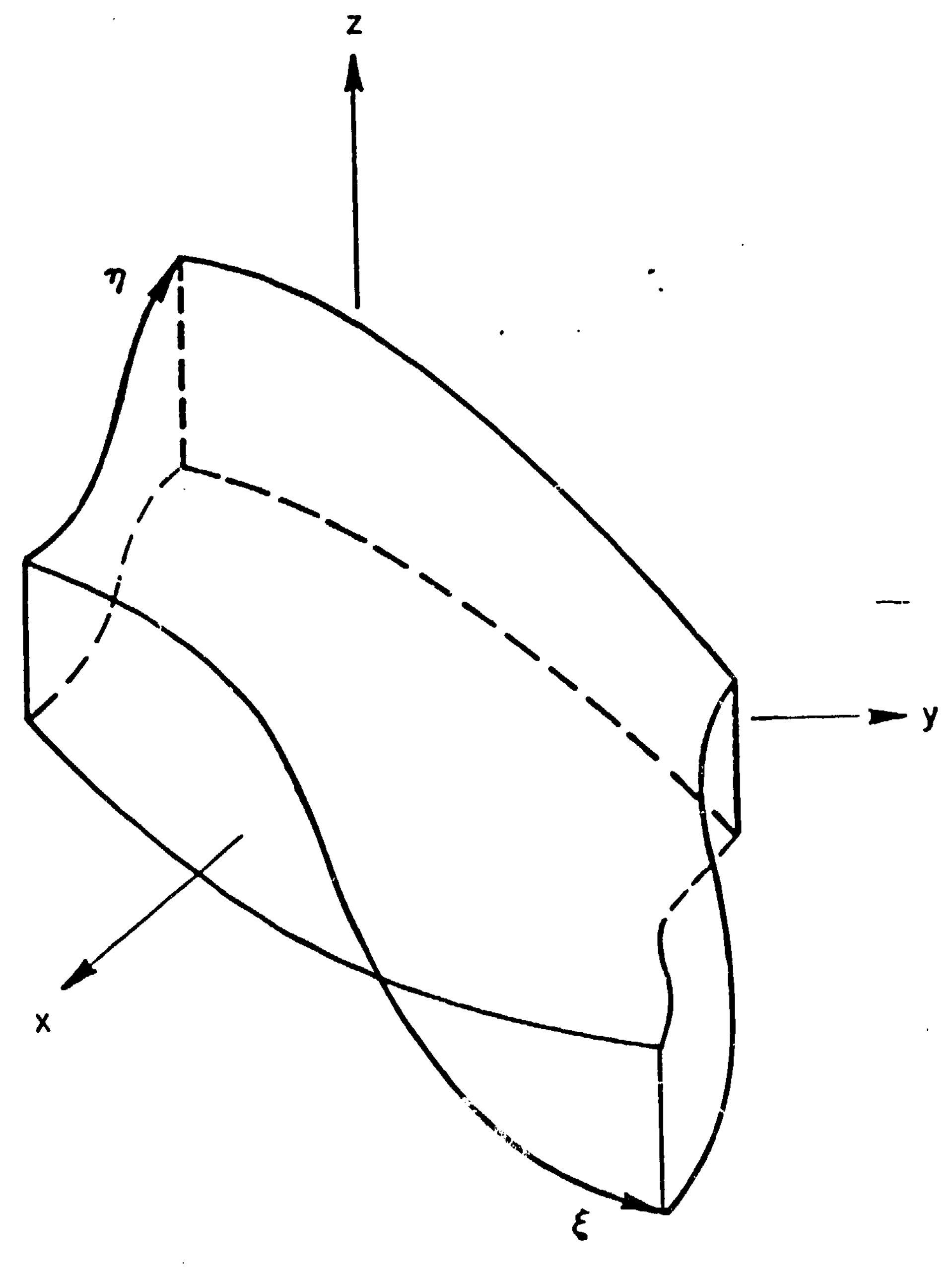
Figure 10-- i, j, k indexes used for the demonstration model (see Figs. 1 and 2)

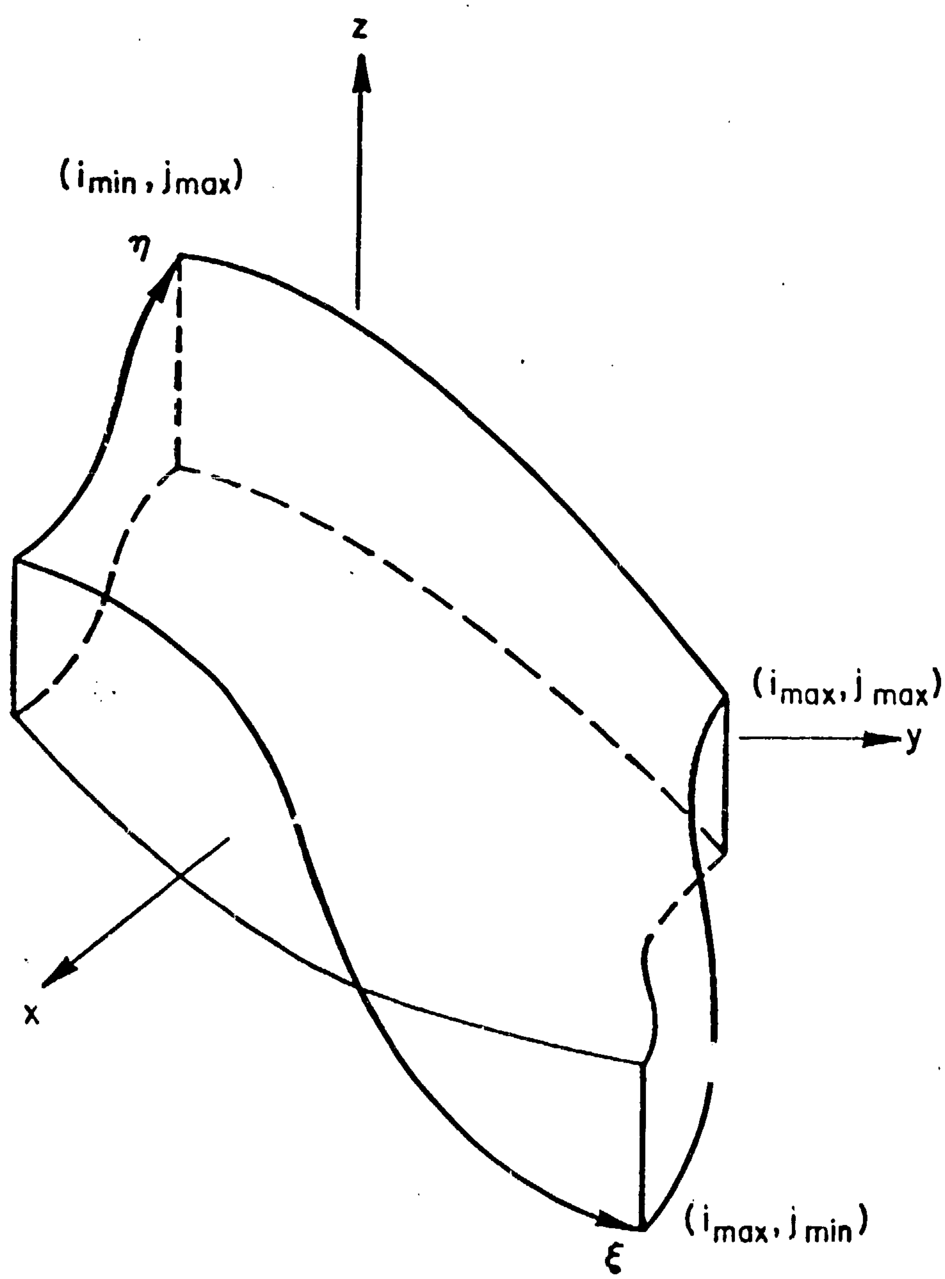
Figure 11-- Pressurized Concrete Reactor Vessel (PCRV) mesh showing the corner nodal points (20 node elements).

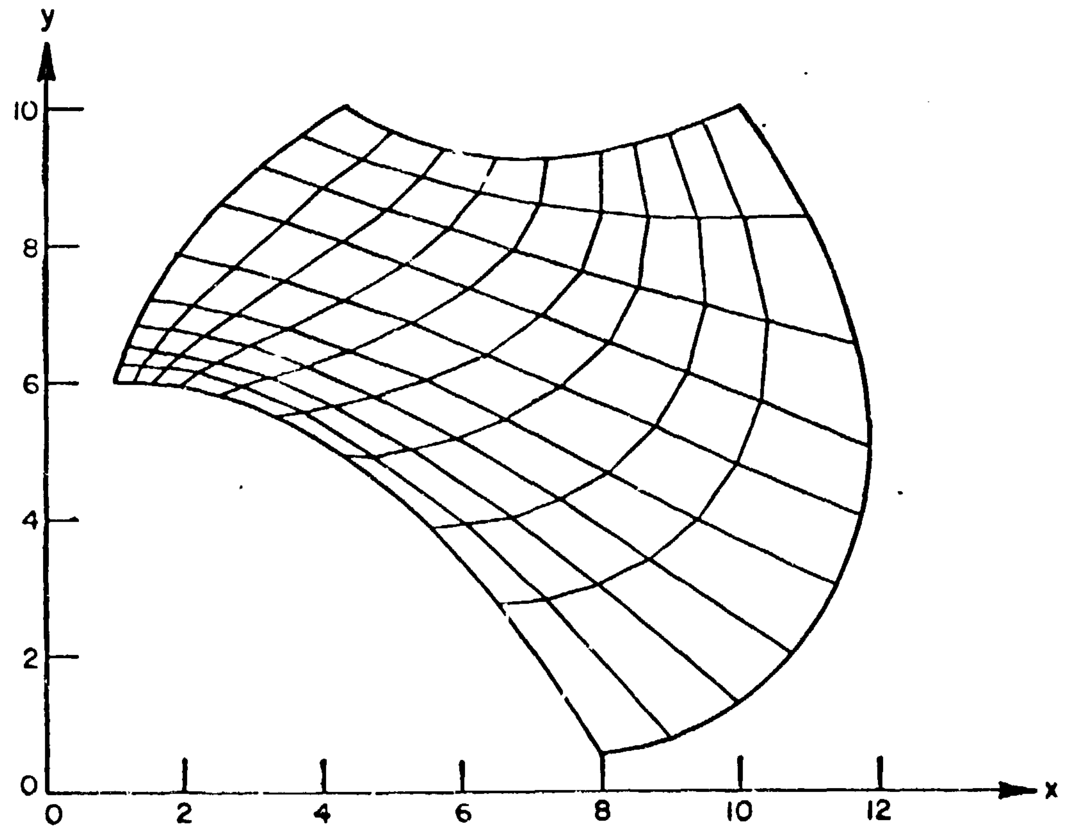
Figure 12-- Controlled Thermonuclear Fusion Reactor (CTR) mesh (8 node elements).



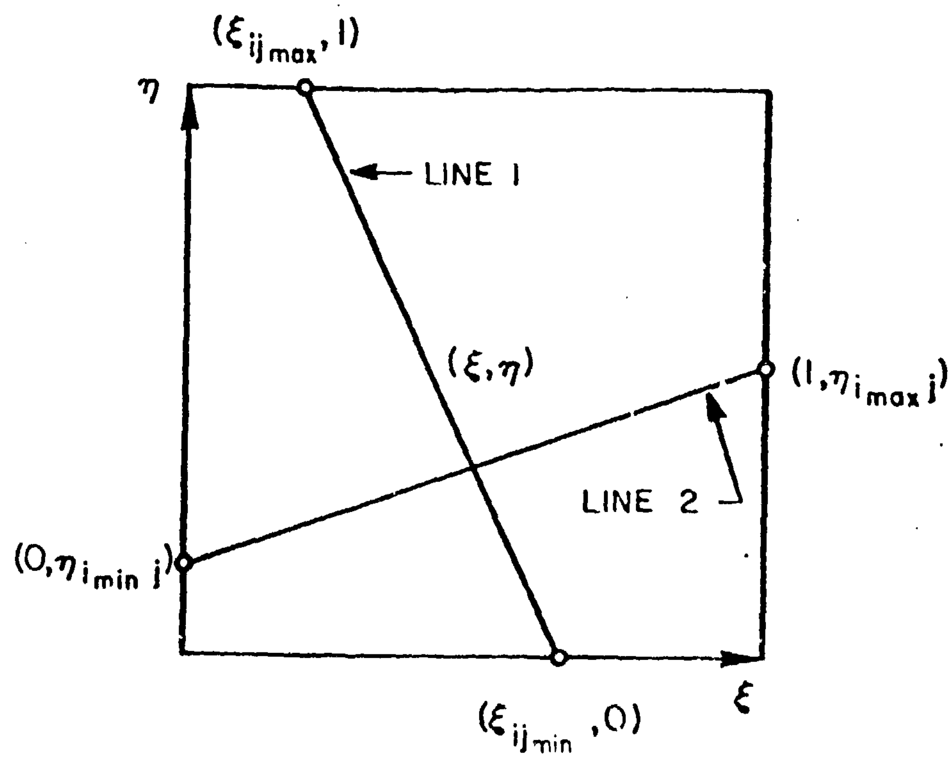



z
η
y
x
ξ

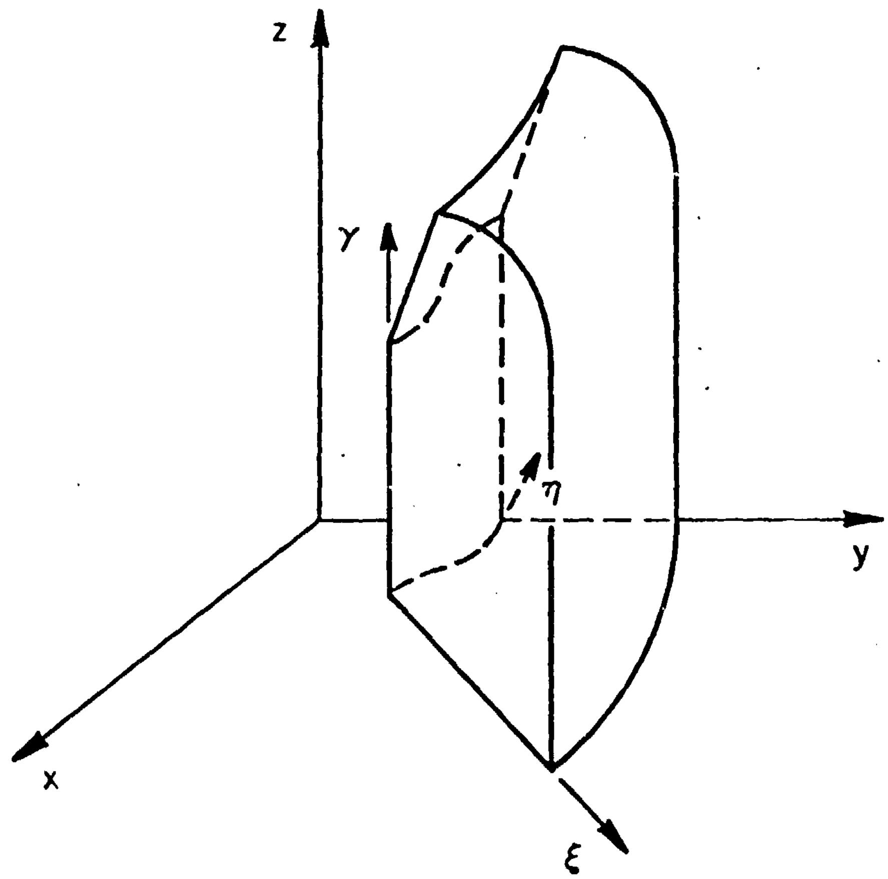

z
(i_{min}, j_{max})
η
(i_{max}, j_{max})
y
x
(i_{max}, j_{min})
ξ



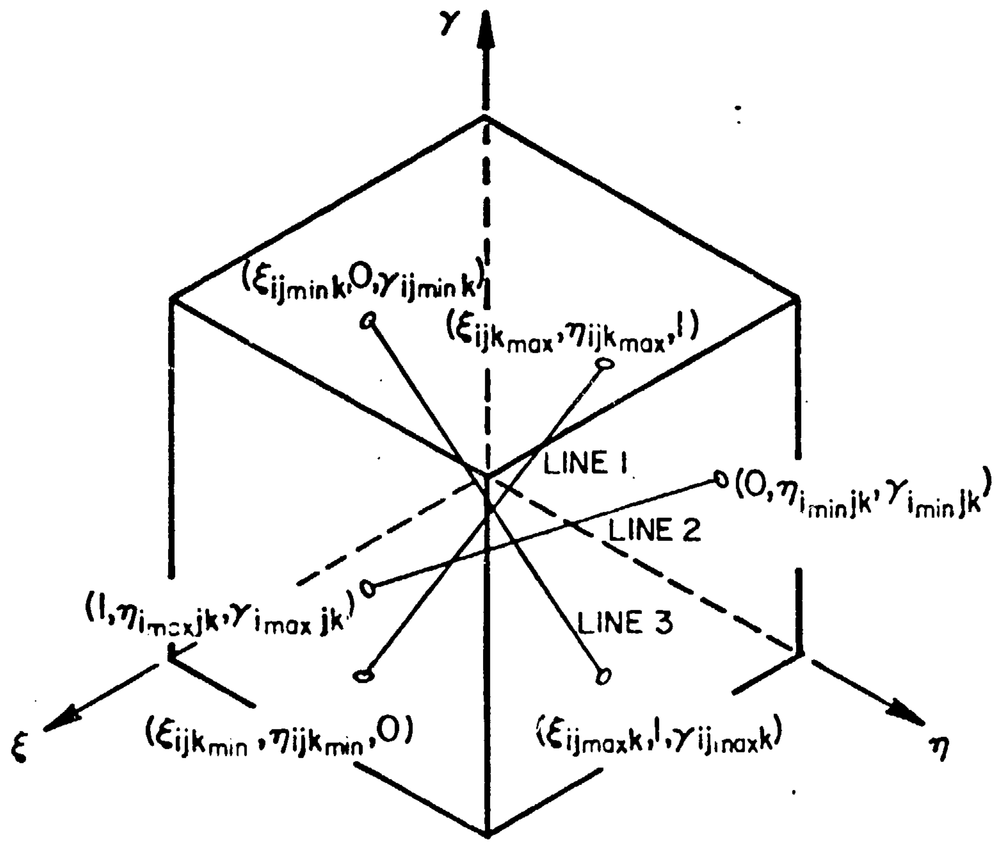
y
10
8
6
4
2
0
0
2
4
6
8
10
12
x

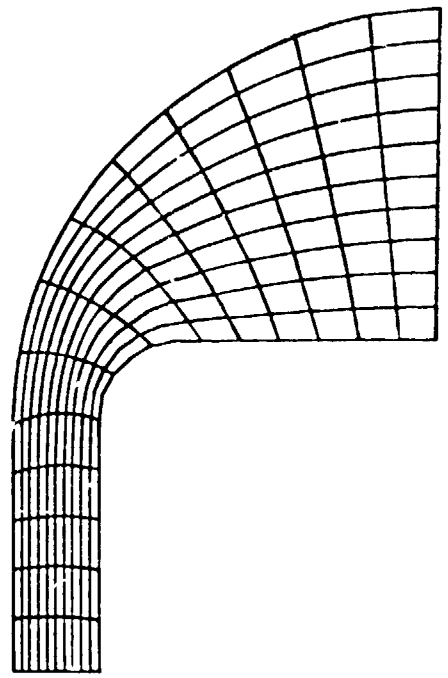


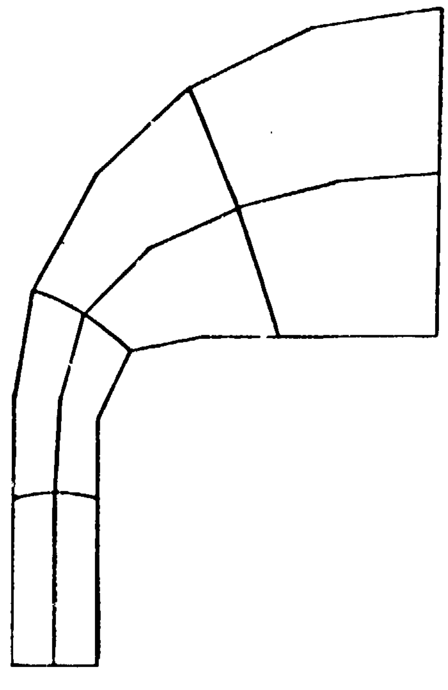
$(\xi_{ij_{max}}, 1)$
η
LINE 1
(ξ, η)
$(1, \eta_{i_{max}j})$
LINE 2
$(0, \eta_{i_{min}j})$
ξ
$(\xi_{ij_{min}}, 0)$

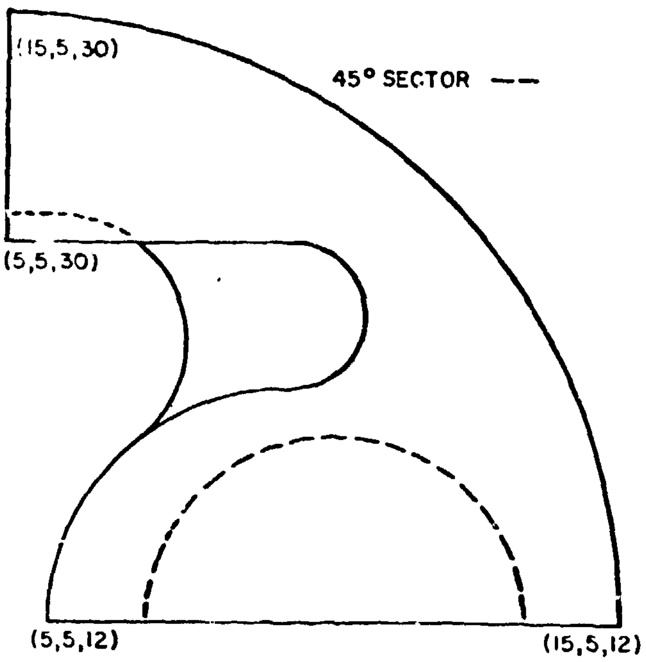


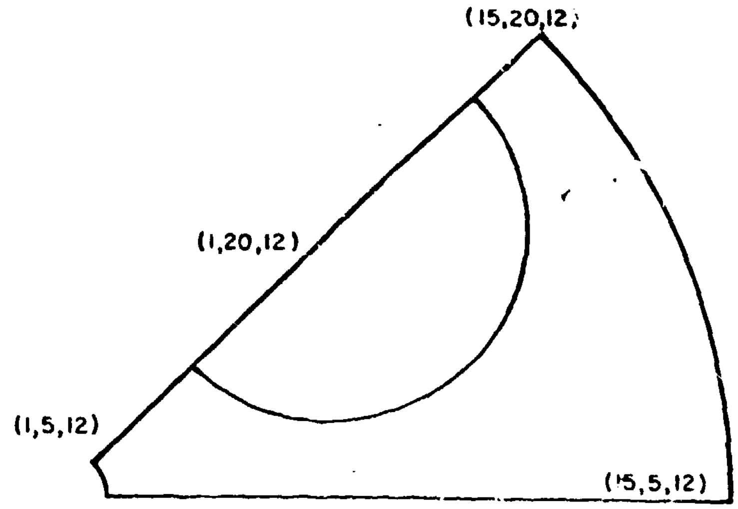
z
γ
η
y
x
ξ

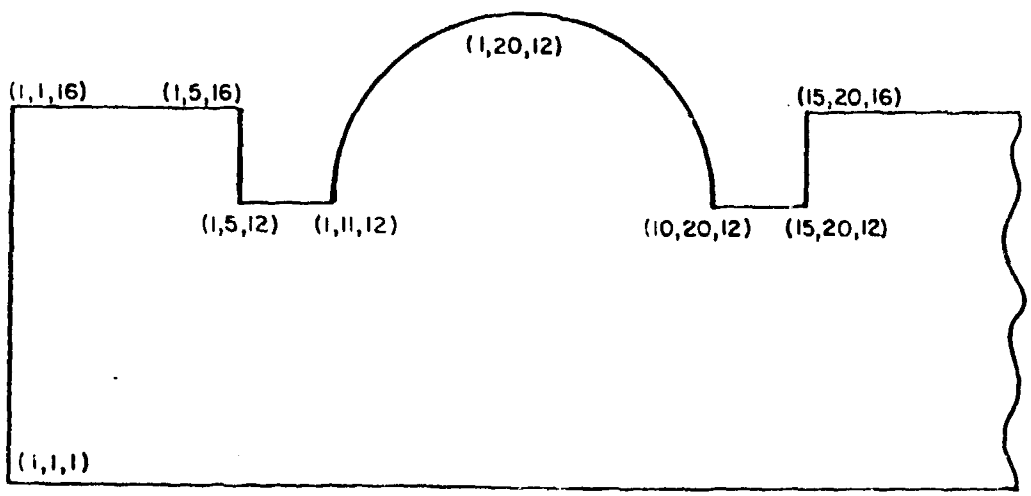

γ
(ξijmink,0,γijmink)
(ξijkmax,ηijkmax,1)
LINE 1
(0,ηiminjk,γiminjk)
LINE 2
(1,ηimaxjk,γimaxjk)
LINE 3
ξ
(ξijkmin,ηijkmin,0)
(ξijmaxk,1,γijmaxk)
η

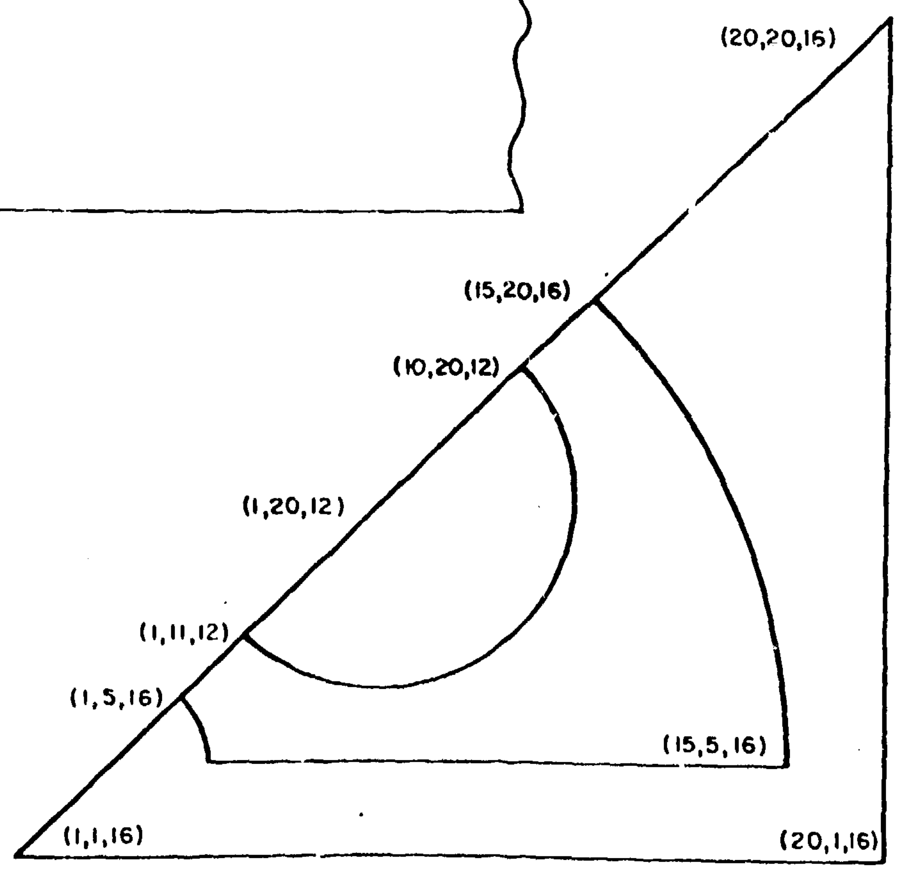


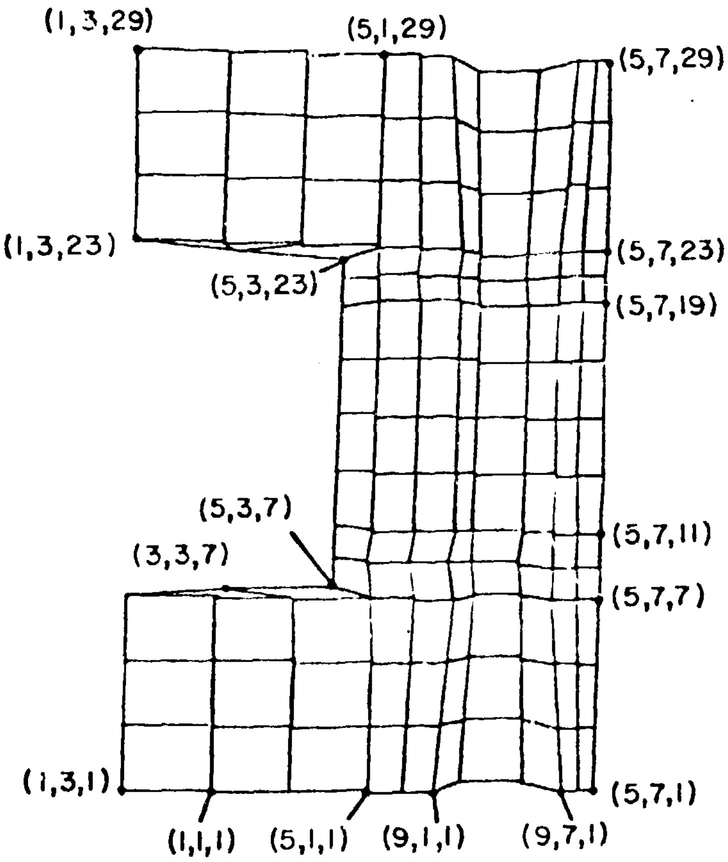



(15,5,30)
45° SECTOR
(5,5,30)
(5,5,12)
(15,5,12)

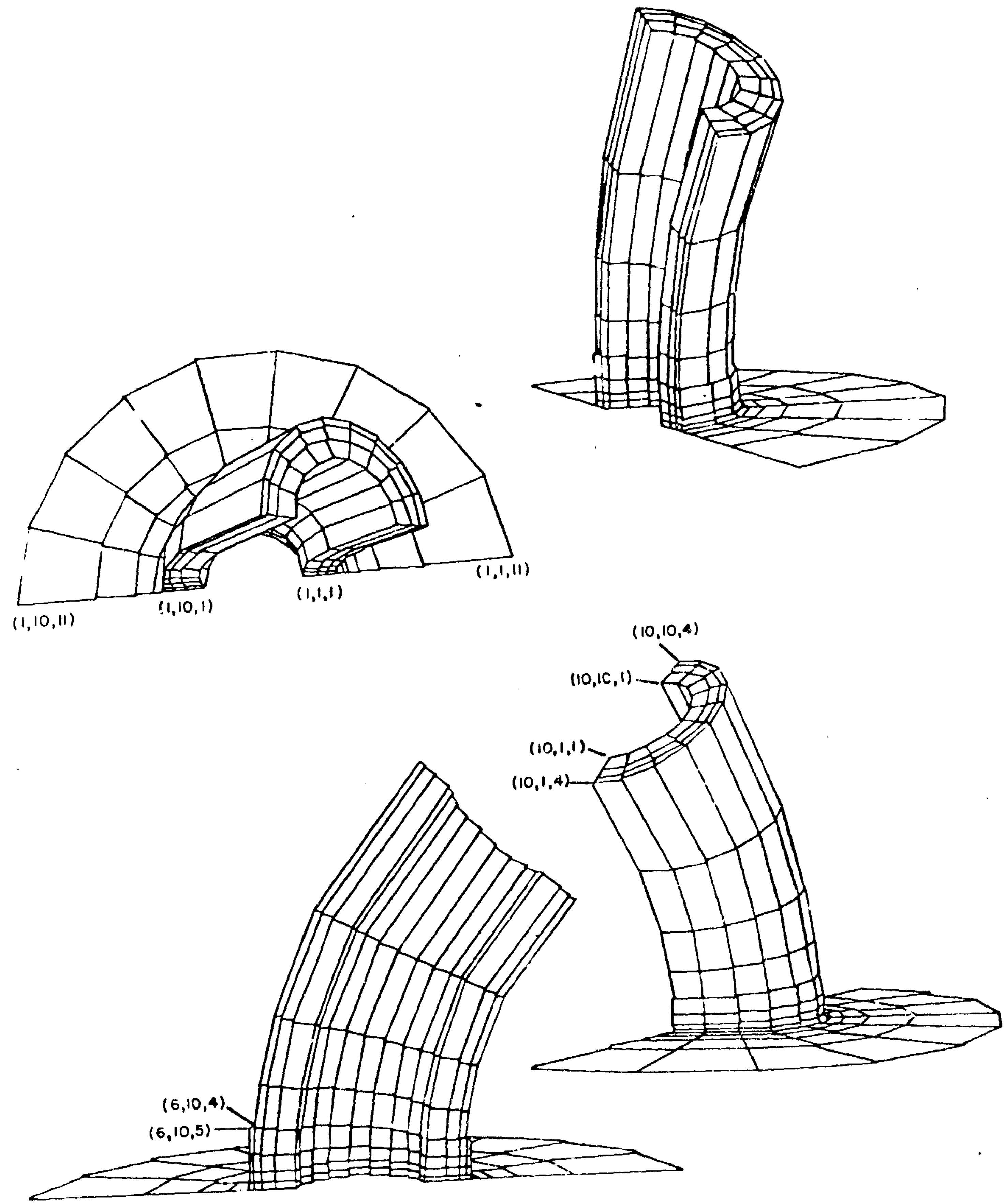

(15,20,12)
(1,20,12)
(1,5,12)
(15,5,12)


(1,20,12)
(1,1,16)
(1,5,16)
(15,20,16)
(1,5,12)
(1,11,12)
(10,20,12)
(15,20,12)
(1,1,1)


(20,20,16)
(15,20,16)
(10,20,12)
(1,20,12)
(1,11,12)
(1,5,16)
(15,5,16)
(1,1,16)
(20,1,16)



(1,3,29)
(5,1,29)
(5,7,29)
(1,3,23)
(5,3,23)
(5,7,23)
(5,7,19)
(5,3,7)
(3,3,7)
(5,7,11)
(5,7,7)
(1,3,1)
(1,1,1)
(5,1,1)
(9,1,1)
(9,7,1)
(5,7,1)



(1,10,11)
(1,10,1)
(1,1,1)
(1,1,11)
(10,10,4)
(10,10,1)
(10,1,1)
(10,1,4)
(6,10,4)
(6,10,5)