AUTOMATIC MESH GENERATION AND FINITE ELEMENT ANALYSIS OF A
TRIAX DOME

by
K. K. Raghu

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Civil Engineering

APPROVED:

Prof. Siegfried M. Holzer, Chairman

Prof. D. A. Garst

Prof. J. R. Llobera
AUTOMATIC MESH GENERATION AND FINITE ELEMENT ANALYSIS OF A
TRIAx DOME

by

K. K. Raghu

Prof. Siegfried M. Holzer, Chairman

Civil Engineering

(ABSTRACT)

An overview of automatic mesh generation is presented and the need for a specialized
mesh generation program for triax domes is discussed. The method of generation of the
triax dome geometry, the terminology, the dome members and loading conditions and
its modelling with ABAQUS is explained briefly.

Development of an automatic mesh generation program, its structure and functions, the
limitations and desirable extensions are also discussed. The program generates nodal
coordinate and element connectivity data and also has mesh refinement features. All the
necessary data required for modelling the triax dome with ABAQUS is generated
automatically by the program.

The model of an existing triax dome at Raleigh, North Carolina, is generated by the
mesh generation program and is analyzed for stresses and displacements using the
ABAQUS finite element analysis program.
Acknowledgements

My sincere appreciation is expressed to my advisor Dr. S. M. Holzer for his continuous support and guidance. Without his help during my entire course work and this thesis work, it would not have been possible. I am grateful to Professor Garst and Professor Loferski for their participation in the committee. I thank Dr. Davalos of University of West Virginia for his advise and helpful tips during the development of the program. Special thanks to Guru and all my friends for their constant help and support. I am very grateful to my mother, father and my family for their love, affection and constant moral support.
Table of Contents

Chapter 1: Introduction ................................................................. 1

Chapter 2: Automatic Mesh Generation: An Overview .......................... 3
  2.1: Introduction........................................................................ 3
  2.2: Discretization of Structures.............................................. 4
  2.3: Mesh Generation in DLEARN Program............................. 5
  2.4: Conclusions .................................................................... 13

Chapter 3: Triax Dome .................................................................. 14
  3.1: The Projection Method.................................................... 15
  3.2: The Triax Dome Terminology.......................................... 15
  3.3: Dome Members and Loads ............................................ 19
  3.4: Modelling with ABAQUS............................................... 19
  3.5: Load Transformation Using Film Elements...................... 21
Chapter 4: Mesh Generation Program .......................................................... 24
  4.1: Overview ......................................................................................... 24
  4.2: Program Structure and Functions .................................................... 25
    4.2.1: Main Triax Node Generation and Projection ............................... 27
    4.2.2: Generation of Intermediate Nodes ............................................ 30
    4.2.3: Beam Element Connectivity .................................................. 32
    4.2.4: Purlin Node Generation ......................................................... 34
    4.2.5: Purlin Element Connectivity .................................................. 35
    4.2.6: Triangular Film Element Connectivity ..................................... 35
    4.2.7: Peripheral Field ....................................................................... 35
    4.2.8: Direction Cosines for the Beam Cross-section ........................... 36
  4.3: Limitations and Desirable Extensions ................................................ 36

Chapter 5: The Finite Element Analysis of A Triax Dome ......................... 40
  5.1: The Raleigh Dome ........................................................................ 40
  5.2: Triangular Shell Elements .............................................................. 41
  5.3: Input File ..................................................................................... 45
  5.4: Boundary Conditions ..................................................................... 47
  5.5: Test Problems ................................................................................ 47
  5.6: Discussion of the results ................................................................. 56
    5.6.1: Effect of the Film Elements ..................................................... 52
    5.6.2: Comparison of Results of the Analysis ....................................... 56

Chapter 6: Conclusions and Recommendations ......................................... 57
  6.1: Conclusions ................................................................................... 57
  6.2: Recommendations ......................................................................... 58
References .............................................................................................................. 59

Appendix A: Spherical trignometric formulae ......................................................... 61

Appendix B: Data sequence for MESHGEN program .............................................. 64

Appendix C: Example input file for ABAQUS ....................................................... 67

Appendix D: Program listing .................................................................................. 71

Vita .......................................................................................................................... 93
1.0 List of Figures

Fig. 2.1 Node generation along a line ................................................................. 7
Fig. 2.2 Node generation over a surface ............................................................. 8
Fig. 2.3 Node generation within a volume .......................................................... 10
Fig. 2.4 Element Generation over a logically rectangular region ...................... 11
Fig. 3.1 Pattern plane and its projection ............................................................. 16
Fig. 3.2 One sector of triax dome model ............................................................. 17
Fig. 3.3 Triax Connectors ...................................................................................... 20
Fig. 3.4 Triax Panels with Film Elements ........................................................... 22
Fig. 4.1 Tree Chart for the Mesh Generation Program ........................................ 26
Fig. 4.2 Top Six Panels in a Triax Dome .............................................................. 28
Fig. 4.3 Projection of the pattern plane ............................................................... 29
Fig. 4.4 Nodes and Elements in One Sector ......................................................... 31
Fig. 4.5 Spherical Triangles .................................................................................. 33
Fig. 4.6 Direction cosines of the local-1 axis ....................................................... 38
Fig. 5.1 Geometry of The Crafts Pavillion Triax Dome, Raleigh, NC ............... 42
<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 5.2</td>
<td>Dimensions of Beam, Purlin and Tension Ring of the Dome</td>
<td>43</td>
</tr>
<tr>
<td>Fig. 5.3.a</td>
<td>Positive Normals for Shell Elements</td>
<td>44</td>
</tr>
<tr>
<td>Fig. 5.3.b</td>
<td>Wind Pressure Distribution Along the Normals</td>
<td>44</td>
</tr>
<tr>
<td>Fig. 5.4</td>
<td>Dome Cap Analysis for Computation of the Film Thickness</td>
<td>46</td>
</tr>
<tr>
<td>Fig. 5.5</td>
<td>Boundary Conditions for The Triax Dome Model</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 5.6</td>
<td>Design Loads for a Triax Dome</td>
<td>49</td>
</tr>
<tr>
<td>Fig. 5.7</td>
<td>Critical Nodes and Elements</td>
<td>53</td>
</tr>
</tbody>
</table>
2.0 List of Tables

Table 5.1 Displacement at Node 1 for Different Film Element Thicknesses ........ 46
Table 5.2 Displacements at Critical Points for Different Methods of Analysis ..... 51
Table 5.3 CPU Time for Different Runs .................................................................. 55
Chapter 1: Introduction

The main objective of this study is to develop an automatic mesh generation program for triax domes. The features of the program should include, mesh generation for domes of any size and triax number. Also, mesh refinement facility is necessary for convergence study.

Because of its economical advantages and superior audio-absorption ability, glue-laminted (glulam) timber domes have become a popular choice for assembly halls, stadiums, etc. (Holzer, 1989). The structural analysis of the glulam dome has not evolved as much as its increasing popularity (Davalos, 1989). Thus, for a detailed study and understanding of the structural response of a triax dome, an efficient and user friendly preprocessor is required.

Using the mesh generation program a triax dome is to be modelled for finite element analysis with ABAQUS. Once the modelling is complete and accurate, a linear finite element analysis needs to be performed for checking the displacements and stresses.
Reducing the modelling time and CPU time for the finite element analysis of the triax dome is also an important objective of this study.
Chapter 2: Automatic Mesh Generation: An Overview

An overview of the automatic mesh generation of continuous media and skeletal structures is presented. The mesh generation method used in a commercial finite element program DLEARN (Hughes, 1987) is also discussed. Finally the need and the usefulness of specialized mesh generation programs is discussed.

2.1 Introduction

The process of manual preparation of finite element mesh data is often complicated and error-prone. Hence, there has been an increased interest in the development of automatic mesh generation programs. These programs describe to the actual finite element solvers, the geometry of the domain to be analyzed. A major part of the time spent in analyzing a model is spent on this task and so this is a major target of automation.
This automation also helps in refining the finite element mesh which is an important method of improving the accuracy of a finite element solution.

2.2 Discretization of Structures

The process of discretization or mesh generation may be automated for all types of structures. For this purpose the structures may be divided into two types:

1. The continuous media, which have an infinite number of degrees of freedom, examples include folded plates, shell structures, etc.

2. the skeletal frames, which constitute an important class of structures, have a fixed number of degrees of freedom and this class includes building frames, reticulated domes, transmission towers, etc.

In discretizing continuous media, the structure is divided into a finite number of smaller manageable elements connected at finite number of points called nodes. Since these structures can be divided into smaller elements of any acceptable shapes, generalized programs may be developed using various popular methods such as Laplacian, Cell decomposition and Triangulation methods.

In generating finite element meshes for skeletal structures, a particular algorithm may need to be selected. The selection of an algorithmic approach depends on the requirements that it must meet (Mark, et. al., 1986). Depending on the amount of user interaction required, these programs can be fairly general in nature and sometimes they can be very specialized in their application. As an example for generalized skeletal
structures, the mesh generation program used in DLEARN - a linear static and dynamic finite element analysis program (Hughes, 1987) is discussed in the next section. The triax dome mesh generation program is a good example for a specialized program.

2.3 Mesh Generation in DLEARN Program

DLEARN program may be used for analyzing three dimensional trusses since it has a 3-d elastic truss element. It also has 4-node quadrilateral, elastic continuum element, with plane stress, plane strain, and torsional axisymmetric capabilities. The program has simple, but labor-saving, node-generation capabilities in one, two, and three dimensions as well as element generation capabilities. This study was conducted in order to understand the mesh generation methods and the algorithm of the program. Also, the study was made to find the possibility of using the algorithms for triax dome mesh generation. Since triax dome has specialized geometric properties, studying DLEARN was helpful in the development of a dedicated mesh generation program for traix domes.

DLEARN has the following data generation capabilities:

1. Coordinate data

2. Boundary condition data

3. Prescribed nodal forces and kinematic boundary conditions

4. Load-time function

5. Kinematic initial condition data
6. Element data

Among the above functions, nodal coordinate data and element data generations are discussed below:

2.3.1 Nodal coordinate generation

Nodes along a line are generated using linear interpolation functions for straight lines and quadratic interpolation for curves (Fig. 2.1). For these generations, 2 generation points (J) need to be input for straight lines and 3 points for curved lines. To get a graded nodal spacing, the third generation point (J = 3) should be placed off the center.

For generation of nodes over a surface four or eight generation points may be defined. The physical space may be two or three-dimensional (Fig. 2.2).

Bilinear interpolation is employed (when J = 4) for equally spaced nodal points along generating lines. Biquadratic serendipity interpolation is employed for curved surfaces (when J = 8), and graded nodal spacing may be achieved by placing generation points 5-8 off center.

For 3-d structures which are brick shaped, 8 or 20 generation points may be defined. In this case the physical space must be three-dimensional (Fig. 2.3). Trilinear interpolation is used for generating equally spaced nodes and triquadratic serendipity interpolation is used for graded nodal spacing.
Nodal generation along a line.

Nodal generation along a line: mapping from local interval to physical space.

Figure 2.1: Node generation along a line[10]
Nodal generation over a surface.

Figure 2.2: Node generation over a surface[10]
2.3.2 Element data generation

The element connectivity may be generated by inputting node numbers in counterclockwise order and then giving the node number increment for directions 1 and 2 (Fig. 2.4). The same algorithm is used for generating truss elements (Fig. 2.5).

Though the above mentioned functions are very useful for various skeletal structures, it still requires lot of input from the user. These generation methods may be used for some of the lattice reticulated dome generations, but, cannot be used for a triax dome mesh generation. This is because the triax dome has a specialized geometry and needs a specialized algorithm for mesh generation.
Figure 2.3: Node generation within a volume[10]

Nodal generation within a volume.

Chapter 2: Automatic Mesh Generation: An Overview
Element generation over a logically rectangular region.

Figure 2.4: Element generation over a logically rectangular region[19]
Nodal ordering for the truss elements.

\[ NEN = 2 \]

\[ N \]

\[ N + \text{INCEL}(1) \]

\[ I \]

\[ I + \text{INC}(1) \]

\[ I + 2 \times \text{INC}(1) \]

\[ \ldots \]

\[ N + N1 \]

\[ I + J1 \]

\[ I + I1 \]

\[ NEN = 3 \]

\[ N \]

\[ N + \text{INCEL}(1) \]

\[ I \]

\[ J \]

\[ I + \text{INC}(1) \]

\[ J + \text{INC}(1) \]

\[ I + 2 \times \text{INC}(1) \]

\[ \ldots \]

\[ N + N1 \]

\[ I + J1 \]

\[ J + J1 \]

\[ I + I1 \]

\[ N1 = (\text{NEL}(1) - 1) \times \text{INCEL}(1) \]

\[ J1 = \text{NEL}(1) \times \text{INC}(1) \]

\[ J1 = (\text{NEL}(1) - 1) \times \text{INC}(1) \]

Generation of truss elements along a line.

Figure 2.5: Generation of truss elements[10]
2.4 Conclusions

As mentioned in Section 2.2, for specialized applications, algorithmic approach is required. Generalized programs will require greater interaction from the user, hence the need for a specialized program for triax dome mesh generation. It is also essential that this program takes minimum input from the user and it will have the mesh refinement features too.
Chapter 3 : The Triax Dome

The Triax Dome, developed and marketed by Timber Structures, Inc. has some special geometric properties and configuration which are illustrated below. The geometry is special because all the members of this lattice dome lie on the great circles of the sphere. The result is that all members have the same curvature, which simplifies the fabrication of the glue-laminated members. Also, this geometric configuration simplifies the modelling of the triax dome for finite element analysis.

In this chapter the details of triax geometry and modelling a triax dome for ABAQUS are presented in brief. The topics are important since they are referred to very frequently in the chapters 4 and 5 where the program development and analysis of test problem on ABAQUS are presented.
3.1 The Projection Method

The triax geometry is obtained by the method of projection (Neal, 1973) illustrated in Figure 3.1. Projecting a network of equilateral triangles, contained in the plane of the tension ring, onto the dome surface by rays originating at the center of the sphere generates a triax dome network.

When the pattern plane is projected to the domical surface, the equilateral triangles get distorted since the projection is along the radial line. This distortion is the least near the apex and it increases as we move towards the periphery. To minimize the distortion of the nodes within an equilateral triangle, only the basic nodes required to define the triangular field are projected. The internal nodes on the sides of the triangles, which define the finite element mesh, are generated on the surface after the projection is complete. This results in equidistant internal nodes.

3.2 The Triax Dome Terminology

The terms associated with a triax dome are described below (Figs. 3.1 and 3.2).

The *Pattern plane* is an imaginary horizontal plane at the level of support work points has a collection of equilateral triangles radiating outwards from the center of the circular plane (Fig. 3.1a)

The pattern plane consists of two sub-fields: A *triax field* which is composed of equilateral triangles contained inside the largest complete hexagon, and a *peripheral*
Figure 3.1 - Pattern Plane and Its Projection
Figure 3.2 - One sector of a triax dome model
field, which is the portion between the triax field and the circle connecting the support work points. It is geometrically impossible to fill the peripheral field with equilateral triangles. Hence, it is triangulated in such a way as to connect dome support points with the outer edge of the triangle module points of the triax field. This triangulation may be done based on the user requirements and also the number of supports provided for the dome.

There are six - 60 degree segments called sectors in a triax field. The sector line which is between the dome center and the support work points, divides the two adjacent sectors. The diameter of a triax dome is the diameter of the circular base of the dome. The radius of the triax dome is the radius of the domical surface. The rise is the vertical distance between the base (support work points) and the apex (center work point) of the dome.

A panel is an equilateral triangular segment of a triax dome. The panel is contained by the beam elements on the three sides and one or more purlins within the panel.

The triax number which can be a whole number or a fractional number is an important parameter in defining the basic triax geometry. It is the number of equilateral triangles contained in one sector line. For a triax number of 4.0, exactly 4 equilateral triangles can be placed on the sector line. If the number is 3.49, then 3 triangles will be placed in the triax field and the fractional part 0.49 times the triangle module will be placed in the peripheral part of the sector line.
3.3 Dome Members and Loads

The different members in a triax dome are the beam elements, purlins, eve purlins and tension ring elements. This skeletal frame work is covered with a timber decking. Two types of connecters are used in the frame work: a steel hub connector for joining six beams, and a steel hanger connector for the purlin to beam connection (Fig. 3.3).

A triax dome may carry concentrated loads or any special loads in addition to the dead load, such as live, wind, snow and seismic loads.

3.4 Modelling on ABAQUS

Beam and purlin elements in a triax dome are curved space elements. In ABAQUS these are modelled as 3 noded isobeam, B32, elements.

After specifying the nodal coordinates, the beam element connectivity, cross section and material properties can be specified. To orient the beam cross section in space, the direction cosines of the local-1 axis of the beam cross section have to be specified. These are specified along with the beam cross section. The computation of direction cosines is presented in chapter 4 (Section 4.2.8).

The steel tension ring is provided to take the tensile forces at the support points of the dome. It constitutes truss elements which are modelled using two noded, solid cross section, C1D2 elements. The film elements used in load transformation are STRI35 3-noded triangular shell, STRI35, elements in ABAQUS. Since, these are curved shell
Figure 3.3 - *Triax Connectors*
elements, though they are 3 noded, the curvature can be defined by specifying direction
sines of the normals at nodes. These direction cosines are input along with the nodal
coordinate data.

The hub connector and the hanger connector may be modelled using 2-noded isobeam,
B31, elements.

3.4 Load Transformation using film elements

The triax domes are usually covered with tongue-and-grove wood decking. But, the
load-carrying capacity of the decking is not considered in this study. Thus, the
distributed loads applied to the decking have to be transformed to the nodes without
actually modelling the decking. One of the methods of transformation from decking to
members is assigning load-tributary areas to the interconnected members
(Davalos, 1989). Though this is an accurate method, a generalized program capable of
transforming the panel loads to the members and in turn to the nodes is required. The
program can be quite complicated since it should be able to work for any number of
beam elements within a panel and a variety of load distributions. Also, manually
calculating and transforming the loads is impractical for such large structures.

In this study, an alternative approach is given to the load handling. A thin film element
is used for the panel load transformation.

Film elements are very thin 3-noded triangular shell elements. These are used in the
place of the wooden decking. It is important to note that the main function of the film
(a) Panel with 12 film elements

(b) Panel with 16 film elements

Figure 3.4: Triax panels with film elements
element is to transform the distributed loads into nodal loads and not to model the decking.

Each panel in a triax dome has several nodes on the beam and purlin elements. The number of nodes depend on the number of elements used to model each member of the panel. Depending on these nodes the panel is triangulated such that the complete panel area is covered with triangular film elements. For instance, a panel with two beam elements on each side and one purlin element requires 12 film elements. The Raleigh dome, which was modelled with two beam and two purlin elements in each panel requires 16 film elements per panel (Fig. 3.4).

The dome framework covered with film elements is subjected to distributed loads and analyzed with ABAQUS. After the load transformation, the film elements are no longer required and they can be eliminated. In order to remove these film elements a two pass analysis is done. Initially, all the nodes in the dome are fixed and the structure analyzed with distributed loads. After obtaining the reactions at the nodes, these reactions are applied as nodal forces and the dome is analyzed for final results. The actual modelling of the film elements and the load application are discussed in Chapters 4 and 5.
Chapter 4: Mesh Generation Program

4.1 Overview

The Triax dome mesh generation program is written in Fortran 77. The input file format and the listing are included in appendix A. The program was developed using structured programming principles (Holzer, 1985). The program structure and functions are presented in section 4.2, the program limitations and extensions are included in section 4.3. Also, all the variables and functions of different routines are described in the listing.

To model a triax dome for ABAQUS, the mesh generation program will generate the nodal coordinates, element connectivities, and direction cosines for orienting the beam cross section in space. A Spherical coordinate system has been used throughout this program.

The program has been generalized for any triax dome mesh generation. It can model domes of any given triax number. The panels in these domes may have any number of beam elements. The number of purlin elements and also the number of purlins in a
panel may also be varied. These features will facilitate in using this program for modelling triax domes of very large diameter too. Also, using varying a number of beam and purlin elements during a preliminary analysis will help in the mesh convergence study. For example, two beam and purlin elements were used per member for modelling the Raleigh Dome, and this was found to be accurate enough by a mesh convergence study (Davalos, 1989).

4.2 Program Structure and Functions

The program structure, consisting of a main program called MESHGEN with 15 subroutines and two supplemental programs PFEMLGEN and FILMGEN, is shown in the tree chart (Fig. 4.1). MESHGEN initializes the variables, reads the parameters which define the triax geometry, calls the subroutines and finally prints the data essential for modelling a triax dome for ABAQUS.

The complete program is explained based on its functions. The individual subroutines are not explained separately. The different functions of the mesh generation program are as follows:

1. Main triax node generation and projection

2. Generation of intermediate nodes

3. Beam element connectivity

4. Purlin node generation
Figure 4.1: Tree chart for the mesh generation program
5. Purlin element connectivity

6. Triangular film element connectivity

7. All the above information for the peripheral field

8. Direction cosines of the local-1 axis for all the beam and purlin elements

The above functions are described in the following sub-sections.

4.2.1 Main triax node generation and projection:

This function is executed by the routine COORD and is called by MESHGEN.

The triax nodes may be classified into two types. The main nodes which occur at the intersection of different beam elements and the internal nodes which occur between the two main nodes. Initially the main nodes on the pattern plane for only one sector are generated. The node numbering is done in concentric circles (Fig. 4.2). The internal nodes, such as 2, 3,....., 8,9,.... (Fig. 4.2), are also taken in the count. Once the nodes are generated on the pattern plane, these need to be projected to the domical surface.

The method of projection, discussed in section 3.1, uses the following simple algorithm:

Generate the nodes on pattern plane in the cylindrical $(r, \theta)$ (Fig. 4.3) system with the origin at the radial center (at node 1).
Figure 4.2: Top six panels in a triax dome
Figure 4.3: Projection of the pattern plane
For projection, change to spherical coordinate system \((r, \theta, \phi)\) with the origin at the spherical center. The coordinates are calculated as follows (Fig. 4.4):

- The \(r\)-coordinate remains the same for all nodes and is equal to the radius of the sphere.
- The \(\theta\)-coordinate is same as \(\theta\) in the cylindrical system, hence it is left unchanged.
- The \(\phi\)-coordinate is calculated from \(r\) and the height of the pattern plane.

Once a node is projected, the corresponding nodes in the remaining sectors are generated by simple rotation, i.e., increasing the \(\theta\) angle by 60 degrees. \(r\) and \(\phi\) remain constant for all these nodes.

### 4.2.2 Generation of intermediate nodes

The routines performing this function include FNUM, SNUM, INGEN, INGEN1, INGEN2 and INGEN3.

All the main nodes (for one sector) are stored in a temporary array. These nodes are chosen in pairs and the internal nodes between these pairs of nodes are generated using spherical trigonometric formulae. The pairs of nodes are selected in a particular sequence by the routines FNUM and SNUM which generate the first and second node of the pair, respectively. For example, the main node pair 1-20 (Fig. 4.4) is chosen first, to generate the internal nodes 2, 8 \\& 14. The procedure is repeated for the subsequent pairs 1-24, 20-24 and so on. The numbers in pairs 1-20, 1-24, etc., are generated by FNUM and SNUM.
Figure 4.4: Nodes and elements in one sector
Knowing the two main node numbers, their coordinates and the number of beam elements in a panel (NELPAN), the internal nodes may be generated using the spherical trigonometric formulae (Appendix A).

The four spherical triangles mentioned in the following algorithm need to be solved for generating each node.

Algorithm:

1. Choose the nodes between which the internal nodes have to be generated (from routines FNNUM and SNNUM)

2. Pass the nodal coordinates, number of internal node to be generated, etc., into the INGEN1 subroutine.

3. Solve the two triangles for $\psi_1$ and $\psi_2$ (Fig. 4.5)

4. Solve the triangle $\psi_1 - \psi_2 - \alpha$ for $\psi_3$

5. Solve the triangle $\psi_3 - \psi_2 - \alpha$ to get the coordinates of the new node

4.2.3 Beam element connectivity (Fig. 4.2)

Beam element connectivity is generated using two routines ELMMAIN and ELMGEN. ELMMAIN stores nodes on each concentric circle in a separate array. The nodes 2, 8, 14, ..., on the radial lines are stored in the C - array. The nodes 20, 21, 22, 23, ... on the
Figure 4.5: Spherical Triangles
concentric hexagonal line are stored in the S - array. ELMGEN generates the connectivity using the arrays generated by ELMMAIN.

The following algorithm illustrates the method of element generation:

Choose the first three arrays S1, C1 & C2 and generate elements 1, 2, ..., 6 in sequence. The node numbers are already stored in proper sequence in these arrays. Repeat the procedure for the next set of arrays.

4.2.4 Purlin Node Generation

In a triax panel, the number of purlins and the number of elements per purlin can vary depending on the size of the panel. A panel size will depend on the size of the dome and the triax number. For large size panels the number of purlins can be more than one. The two important variables to be input for this generation are the number of elements in the first purlin (NELPLN) and the purlin interval (PLNINTV). Subroutine PURLIN performs this function.

Algorithm:

1. Knowing the PLNINTV, pick the two end nodes (8-9, 62-63, ..., Fig. 4.4)

2. Knowing the NELPLN, generate the internal nodes

3. Complete the generation for the remaining 5 sectors by increasing the $\theta$ angle

4. With PLNINTV move to the next end-node pair and generate the internal nodes.
4.2.5 Purlin Element Connectivity

Purlin nodes on each hexagonal line are stored in separate columns of the array and are then used for connectivity generation.

The algorithm is: for each hexagonal line, generate the connectivity for all the purlins in a sequence.

4.2.6 Triangular Film Element Connectivity

Program 2, named FILMGEN, generates the film element connectivity. The input data for this part consists of connectivity for all the film elements in the first sector. Based on this data and also the nodal increment data, film elements in the remaining five sectors are generated.

4.2.7 Peripheral field

The peripheral field in a triax dome is a difficult portion for automatic mesh generation. This part of the program is not totally automated because the peripheral field needs to be triangulated or filled with beam and purlin elements based on the user’s discretion and input. The generalized feature of the program no longer holds good for this portion of the dome. Hence, to generate the beam and purlin elements the data such as the main node numbers and the number of internal nodes to be generated needs to be input. Based on this input data the routines PFMNGEN and PFPLNGEN call the routine
INGEN1 which computes the required nodal coordinate data. Similarly, based on the user input of element connectivity for one sector, routine PFELMGEN computes the element connectivity for the beams and purlins in the remaining sectors.

4.2.8 Direction Cosines for the Orientation of the Beam Cross Section

The connectivity of a beam element orients the centroidal axis of the element in space. In order to orient the beam cross section in space, ABAQUS requires direction cosines for the local-1 axis of the cross section (Fig. 4.6) to be specified along with the beam cross section.

The direction cosines of the local-1 cross sectional axis are computed by taking cross product of the unit normals (OA and OB) at the two end nodes of a beam element. The listing of the program (LOCAL1) which computes the direction cosines, is included in Appendix A.

4.3 Limitations and Desirable Extensions

1. The effect of steel hub connector or a hanger connector (Fig. 3.4), which have different material and stiffness properties than that of a glulam member, is being neglected and the joints are modelled as rigid joints in this study. This is because the mesh generation program does not contain the feature of modelling connector elements. Generating connector elements will facilitate in modelling the flexible
joints and this will yield more accurate results. Hence, a subroutine for generating the connector elements may be incorporated in the program.
Figure 4.6: Direction cosines for the local-1 axis
2. To generate film elements, which are used in load transformation, the connectivity for all the films in one sector needs to be input. This could get very cumbersome for very large domes. Hence, an automatic film generation program which could be based on triangulation algorithm, is required for a quicker and efficient modelling purpose.

3. The load transformation may also be made by assigning load-tributary areas to the interconnected members (Davalos, 1989). For this method also, a generalized program capable of performing the load discretization for different meshes and sizes of triax domes, will be required.
A linear finite element analysis of the Raleigh dome is performed using ABAQUS. In modelling this dome, the greater part of the ABAQUS input file is generated by the mesh generation program discussed in Chapter 4. The format of the input file is presented in Section 5.3. The Raleigh dome specifications and details are presented in Section 5.1 and the effect of using film elements for load transformation is discussed in Section 5.2. In Section 5.4, the different methods used in analyzing the test problem are presented, while Section 5.5 consists of the discussion of the results of finite element analysis.

5.1 The Raleigh Dome

The crafts Pavilion Triax dome was built in Raleigh, NC in 1975. The dome has a span of 133 ft. and a rise above the tension ring of 18 ft. (Fig. 5.1). It consists of a 3-d grid system of six identical sectors composed of curved southern pine glulam members.
The triax dome consists of 132 beams and 108 purlins. The dome rests on a steel tension ring of 1" x 12" cross section. The eve purlins are 3" x 12.25" in cross section. All the above dimensions are shown in Figure 5.2. The decking consists of a 2" tongue-and-groove wooden panels fastened to the beams and purlins with nails. Though the effect of the decking is expected to be significant, it is not included in this study.

5.2 Triangular Shell Elements

As discussed in Chapter 3 (Sec. 3.4) triangular shell elements, which are also called thin film elements in this study, play a very important role in the load transformation. Hence, this section is included with a detailed discussion on film elements.

There are two types of triangular shell elements available in ABAQUS. They are, the STRI3 flat facet element and the STRI35 curved element. Both elements are meant to be used for thin shells only. They do not include transverse shear deformation. For load transformation purpose the STRI35 curved shell element is used.

To define a shell element, we need to specify the nodal coordinates, the direction cosines of the normals at these nodes, the element connectivity, and the section properties. The normals need to be specified to enable the shell elements obtain the proper curvature. If the normals are not specified, they will be flat shell elements.

The shell element connectivity should be given in counter-clockwise direction as shown in the Figure 5.3. This is important for proper computation in ABAQUS. Based on the connectivity data, the positive normal (Fig. 5.3a) is calculated. The positive normals are used for getting
Figure 5.1: Geometry of the Crafts Pavilion triax dome, Raleigh, NC
Figure 5.2: Dimensions of the Beams, purlins and Tension ring of the dome
Figure 5.3a: Positive Normals for Shell elements

Figure 5.3b: Wind pressure distribution along the normals

Wind pressure from ANSI A58.1-82.
stress output and for applying surface pressures, such as wind pressure, which are applied in the radial direction (Fig. 5.3b). These wind forces may be applied using the key word P in ABAQUS and it is positive in the direction of the positive normal.

Section property specifications include the material type and the thickness of the shell. Since these shells are being used only for load transformation, they should be very thin so that the stiffness contribution to the dome is negligible.

Initially, a shell thickness of 1.0E-03 was used in modelling the Raleigh dome. But it was found that these film elements were indeed contributing considerable stiffness to the structure. Hence, to determine the effect of these film elements on the triax dome, a test problem consisting of the top 6 panels of the triax dome, with the necessary boundary conditions was analyzed. Table 5.1 shows the increase in deflection of node-1, with the decrease in shell thickness and finally the convergence of the values. Based on this analysis, a shell thickness of 1.0E-06 is recommended for load transformation.

5.3 Input File

The structure of an input file with all the necessary key words and control statements for analyzing a typical triax dome is presented in the appendix C. This input file will be more-or-less the same for analyzing any type of structure. The data generated by the mesh generation program is shown in the input file. The user needs to input the remaining keywords. The keywords used for getting a mesh plot and a displaced mesh plot are also shown.
Figure 5.4: Dome cap analysis to arrive at a sufficiently thin film element.

Table 5.1: Convergence of displacement values with Reduction in Film Element thickness

<table>
<thead>
<tr>
<th>Film Element Thickness (inches)</th>
<th>Displacement at Node 1 (U, in.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0E-03</td>
<td>0.2011</td>
</tr>
<tr>
<td>1.0E-04</td>
<td>0.2012</td>
</tr>
<tr>
<td>1.0E-05</td>
<td>0.2013</td>
</tr>
<tr>
<td>1.0E-06</td>
<td>0.2013</td>
</tr>
</tbody>
</table>
5.4 Boundary Conditions

Proper boundary constraints need to be specified in order to eliminate the rigid body motions of the triax dome. Also, these boundary constraints must be modelled as closely as possible to the actual support conditions. In order to achieve this and also to simplify the modelling, the following boundary conditions (Fig 5.5) are imposed at the tension ring nodes.

The two nodes 434 and 482, which are lying on the X-axis, are constrained along the direction 2. The two nodes 458 and 506 along the Y-axis are constrained in the direction 1. In addition, all the nodes on the tension ring are restrained in the 3 direction. The net effect of these constraints is the restraint of the dome rotation about the 3rd axis (direction 3) and at the same time allow these constrained supports to move freely in the radial direction.

5.5 Test Problems

The Crafts Pavilion triax dome is chosen as the test problem and a linear finite element analysis is performed. Using the mesh generation program, the necessary data required for ABAQUS is generated.

The dome is analyzed for dead plus snow loads. The dead and snow loads estimated to be 16 psf and 20 psf, respectively, are applied uniformly over the entire dome (Fig. 5.6b). It may be noted that the actual loading is done by projecting the uniformly distributed loads on the domical surface (Fig. 5.6a).
Figure 5.5: Boundary conditions for the triax dome model
Figure 5.6: Design loads for the triax dome
In this study, for simplicity, the loads are applied over the entire surface of the dome. The dome is analyzed using two methods mentioned below:

1. The Film Element Method: The dome, with film elements subjected to the design loads and with the regular boundary conditions (Fig. 5.5) imposed, is analyzed for stresses, displacements and reactions.

2. The Two-pass Method:
   (i) The dome, with film elements and design loads, with all the nodes fixed, is analyzed for the reaction forces at the nodes. (ii) Now, the film elements in the model are completely removed. The reaction forces obtained in the above analysis are applied as the nodal forces and the dome is analyzed for final results.

The above mentioned analyses are performed to determine the effect of film elements on the overall structural response in addition to determining the stresses and displacements in the individual members. In the following section, the results of the ABAQUS runs, the effect of the film elements, etc., are discussed.

5.6 Discussion of the Results

No major unsymmetric response is observed in the linear finite element analysis of the triax dome. A very small, negligible difference (less than 5%), is observed in the displacement of the corresponding nodes in different sectors.

Table 5.2 shows the displacements of a few selected nodes. The maximum displacement is observed at one of the purlin nodes close
Table 5.2: Displacements at critical points for different methods of analysis

<table>
<thead>
<tr>
<th>Node Number</th>
<th>Film Elem. Method(1)</th>
<th>Two-pass Method(2)</th>
<th>Difference b/w 1 &amp; 2 (in.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.581</td>
<td>1.574</td>
<td>0.007</td>
</tr>
<tr>
<td>7</td>
<td>1.598</td>
<td>1.592</td>
<td>0.006</td>
</tr>
<tr>
<td>34</td>
<td>1.479</td>
<td>1.493</td>
<td>0.014</td>
</tr>
<tr>
<td>93</td>
<td>1.184</td>
<td>1.187</td>
<td>0.003</td>
</tr>
<tr>
<td>180</td>
<td>1.047</td>
<td>1.052</td>
<td>0.005</td>
</tr>
<tr>
<td>1019</td>
<td>1.643</td>
<td>1.641</td>
<td>0.002</td>
</tr>
<tr>
<td>1067</td>
<td>1.352</td>
<td>1.362</td>
<td>0.010</td>
</tr>
<tr>
<td>1167</td>
<td>1.094</td>
<td>1.104</td>
<td>0.010</td>
</tr>
</tbody>
</table>
to the apex and it is 1.641 inches. The maximum displacement in the main nodes, i.e.,
the main beam element node, is at node 7 and it is 1.592 inches. The maximum stress
occurs in a beam element on the sector line (Fig. 5.7) of the triax dome. The element
number 202, with connecting nodes 284, 336 and 378, is subjected to a maximum
compressive stress of 1239 psi. All the corresponding elements on other sector lines also
showed stresses in the same range.

5.6.1 Effect of the Film Elements

The following results were obtained after the two methods of analyses mentioned in the
section 5.5 were performed.

Table 5.2 includes the comparison of the displacements obtained at nodes after analyzing
by the two methods. Since the difference is negligible, the effect of film elements on the
overall structure may be ignored. Hence, if it is required, the triax dome may be
completely analyzed without removing the film elements. But, the following discussion
illustrates the benefits of removing the film elements.

The two-pass method of analysis, in which the film elements are removed, is a very
effective method for analyzing very large domes and also for non-linear analysis. Its
benefits are discussed below:

One of the important factors of consideration in any finite element analysis is the
computer (CPU) time consumption. In order to reduce this CPU time and make these
computationally intensive programs efficient, several time
Figure 5.7: Critical nodes and elements
and memory saving algorithms are used. The examples include Skyline storage scheme (Holzer, 1985), Element-By-Element and Static Condensation methods (Hughes, 1987). Though these are not discussed in this study, an effort was made to minimize the computer time consumption in any possible way, however small the the time savings may be.

Table 5.3 lists the CPU time required for analyzing the triax dome by each method. It is interesting to note that, with little extra effort in preparing the input data file, the two-pass method can be more efficient than the film element method. To perform this efficient two-pass method, two separate sets of beam element data need to be prepared. The first set of beam element data, which will be used for the first pass analysis, will have the following format.

*ELEMENT..., ELSET = XXXX
{Input the connectivity of all the elements here}

*BEAM SECTION, ELSET = XXXX
{Input the cross section }

The direction cosines of the local-1 axis of the cross section are not specified here. This helps in the reduction of the input file size which in turn reduces the CPU time considerably. For the second pass the direction cosines have to be mentioned. As discussed in Section 4.2.8, ABAQUS has a default orientation for all space beam elements if direction cosines are not specified. This default orientation may not be used for the actual analysis. But, since the objective of the first pass in the two pass analysis is to do the load transformation, the beam cross section orientation will have negligible influence on the transformation of loads.
Table 5.3: CPU time for different runs

1. Film element method 670

2. Two-pass method
   (a) First-pass analysis:
      (i) With direction cosines of local-1 axis 650
      (ii) Without direction cosines of local-1 axis 150
   (b) Second-pass analysis 280
This is due to the fact that during the load transformation each film element with all the 3 nodes lying on beams or purlins are being fixed. The local-1 axis direction cosines may then be mentioned in the input data for the second pass analysis.

The input data for both first and second passes may be generated using the mesh generation program by specifying the particular option.

5.6.3 Comparison of The Results of the Analysis

The results of the analysis performed in this study closely matched with the analysis done by using load projection method (Huang, 1989). In the load projection method (Fig. 5.6a), the loads are applied on the projected area of the triangular panels.

Since the net total load applied in this analysis is more than the loads in projected method, the displacements calculated in this study should have been higher than the ones in projection method but, it is observed to be otherwise. For example, the maximum displacement in the projection method occurred at the apex (Node 1) and it is 1.8 inches. Whereas, in this study, the maximum deflection is observed at node 1019 (close to the apex) and is 1.65 inches. The same observation was made at other nodes too. This difference may be attributed to the different methods used in load discretization and it needs to be studied in detail.
Chapter 6: Conclusions and Recommendations

6.1 Conclusions

The mesh generation program can accurately model the triax domes for finite element analysis. The model was found to be accurate geometrically, which was checked by the mesh plots. The accuracy of the finite element model is reflected in the final results. The program requires implementation of several features which are included in section 6.2. The results of the finite element analysis of triax domes, with and without film elements, proves that the effect of the shell element stiffness on the overall structural response can be negligible.

The direction cosines of the normals must be specified when using the three-noded triangular shell elements. The direction cosines of the local-1 axis of the beam cross-section must be specified for beam elements.
Finally, it can be mentioned that specialized mesh generation programs are very useful for performing detailed analyses of structures, such as triax domes. The mesh refinement facility helps in examining the dome against design codes.

6.2 Recommendations

The following features should be implemented in the mesh generation program, to increase the program efficiency and accuracy in modelling.

1. Modelling connector elements for analyzing them as flexible joints.

2. Automating the film element generation will be useful for load transformation.

3. Routines capable of loading the dome partially.

4. For uniform surface loading the loads should be projected on to the triangular films as shown in Figure 5.5a. This projected loading data should be generated automatically.

5. Element generation based on random material property generation may be included.

6. Spherical coordinate system is successfully used in this study and is proved to be very effective for spherical domes.
References


(7) Hibbit, Karlsson & Sorensen, Inc., ABAQUS.


Appendix A : Spherical Trignometric Formulae

Knowing the two end node coordinates, the intermediate nodes on the domical surface may be generated using the spherical trignometric formulae. From $\theta_1$, $\phi_1$ and $\theta_2$, $\phi_2$ of the two end nodes, the third side $\psi$ of the triangle $\theta - \phi - \psi$ is calculated as follows:

$$\psi_1 = \cos^{-1}[\cos \phi_1 \cos \theta_1]$$

$$\psi_2 = \cos^{-1}[\cos \phi_2 \cos \theta_2]$$

The included angle ($\phi\psi$) of the sides $\theta$ and $\psi$ is,

$$\theta \psi_1 = \cos^{-1}\left(\frac{\tan \theta_1}{\tan \psi_1}\right)$$

$$\theta \psi_2 = \cos^{-1}\left(\frac{\tan \theta_2}{\tan \psi_2}\right)$$

The included angle $\psi_{12}$ of the sides $\psi_1$ and $\psi_2$ of the two triangles is,

$$\psi_{12} = (\theta \psi_1 - \theta \psi_2)$$
From $\psi_{12}$, $\psi_1$, $\psi_2$, the third side $\alpha$, and the included angle ($\alpha \psi_2$) of the sides $\alpha$ and $\psi_2$ are calculated as below:

$$\alpha = 2 \times \tan^{-1}\left[\frac{\cos B_1 \times \tan \frac{(\psi_1 + \psi_2)}{2}}{\cos B_2}\right].$$

$$\alpha \psi_2 = (2B_1 - (B_1 + B_2))$$

where,

$$B_1 = \tan^{-1}\left[\frac{\cos\left(\frac{\psi_2 - \psi_1}{2}\right)}{\cos\left(\frac{\psi_2 + \psi_1}{2}\right) \tan\left(\frac{\psi_{12}}{2}\right)}\right]$$

$$B_2 = \tan^{-1}\left[\frac{\sin\left(\frac{\psi_2 - \psi_1}{2}\right)}{\sin\left(\frac{\psi_2 + \psi_1}{2}\right) \tan\left(\frac{\psi_{12}}{2}\right)}\right]$$

The angle $\alpha$ is modified based on the location of the internal node between the two end nodes.

If the internal node is located midway then $\alpha$ is taken as half the original $\alpha$. If it is located at 1/4 point from the end node, then $\alpha$ is taken as 0.25 of the original $\alpha$.

Using the modified $\alpha$, $\psi_2$ and $\alpha \psi_2$, the third side $\psi_3$ and the included angle $\psi_{23}$ is calculated as follows:

Appendix A : Spherical Trignometric Formulae
\[ \psi_3 = 2 \times \tan^{-1} \left[ \cos B_1 \times \tan \left( \frac{\alpha + \psi_2}{2 \cos B_2} \right) \right] \]

\[ \psi_{23} = (B_1 - B_2) \]

where,

\[ B_1 = \tan^{-1} \left[ \frac{\cos \left( \frac{\psi_2 - \alpha}{2} \right)}{\cos \left( \frac{\psi_2 + \alpha}{2} \right) \tan \left( \frac{\alpha \psi_{12}}{2} \right)} \right] \]

\[ B_2 = \tan^{-1} \left[ \frac{\sin \left( \frac{\psi_2 - \alpha}{2} \right)}{\sin \left( \frac{\psi_2 + \alpha}{2} \right) \tan \left( \frac{\alpha \psi_{12}}{2} \right)} \right] \]

Finally, \( \theta_3 \) and \( \phi_3 \) coordinates of the intermediate node are given by,

\[ \theta_3 = \tan^{-1} \left[ \tan(\psi_3) \cos(\theta \psi_3) \right] \]

\[ \phi_3 = \sin^{-1} \left[ \sin(\psi_3) \sin(\theta \psi_3) \right] \]
Appendix B: Data Sequence for MESHGEN Program

Input variable definition:

- DIA: Diameter of the triax dome
- RISE: Rise of the dome
- TRIAX: Triax number
- NELPAN: Number of beam elements in one panel side
- KEY: 0 - to suppress the generation of peripheral field
  1 - to generate the peripheral field
- TRNUM: First node number on the tension ring = last node number in the triax field + 1
• **NSET**: Number of intermediate nodal sets to be generated. Equal to the number of data lines that will follow

• **NODE1**: First end node number

• **NODE2**: Second end node number

• **NMN**: Number of mid nodes to be generated

• **MN(I)**: Mid node numbers

• **INC**: Increment number for generating the corresponding node in the next sector

• **NPASS**: Number of concentric circles along which the beam elements need to be generated

• **NSET**: Number of beam elements input in the following data lines

• **ELINC**: Element increment number

• **NINC1**: Node 1 increment number

• **NINC2**: Node 2 increment number

• **NINC3**: Node 3 increment number

• **ELNUM**: Element number
Data sequence (Free format)

-----------------------------------MESHGEN----------------------------------- DIA, RISE, TRIAX, NELPAN, KEY, TRNUM

(If KEY = 1, then input the following data)

-----------------------------------PFMNGEN-----------------------------------

NSET, INC

NODE1, NODE2, NMN, (MN(I), I = 1,NMN) (Same sequence for NSET lines)

-----------------------------------PFPLNGEN-----------------------------------

NSET, INC

NODE1, NODE2, NMN, MN (Same sequence for NSET lines)

-----------------------------------PFELMGEN-----------------------------------

(Data for generating beam elements)

NPASS

NSET, ELINC, NINC1, NINC2, NINC3 (Same sequence for NPASS lines)

ELNUM, NODE1, NODE2, NODE3 (Same sequence for NSET lines)

(Data for generating purlin elements)

NPASS

NSET, ELING, NINC1, NINC2, NINC3 (Same sequence for NPASS lines)

ELNUM, NODE1, NODE2, NODE3 (Same sequence for NSET lines)

(data for generating eve purlins)

TRNUM

-----------------------------------END OF DATA-----------------------------------

Appendix B : Data Sequence for MESHGEN Program
Appendix C : Example input file for ABAQUS

*HEADING
POINT LOAD ON SINGLE RING CAP
*NODE, NSET = MAIN, SYSTEM = S
**NODAL COORDINATE DATA
   1,1600.00098,  0.00000,  90.000000,-0.000000, 0.0000000, -1.0000000
   2,1600.00098,  0.00000,  87.62790, -0.0413903, 0.0000000, -0.9991431

*NODE, NSET = TRNODE, SYSTEM = S
**PURLIN NODE COORDINATE DATA
   14,1600.00098,  29.99969,  85.88910, 0.0, 0.0,-1.0
   15,1600.00098,  89.99969,  85.88910, 0.0, 0.0,-1.0

*NSET, NSET = FIXNODES
MAIN, TRNODE
**BEAM ELEMENT CONNECTIVITY
*ELEMENT, TYPE = B32, ELSET = BEAM1
**MAIN BEAMS (6.75" * 11.00")
   1, 1, 2, 8
2, 1, 3, 9
*ELEMENT, TYPE = B32, ELSET = EVEPLN
  7, 8, 14, 9
  8, 9, 15, 10
*ELEMENT, TYPE = CID2, ELSET = TRING
**TENSION RING ELEMENT CONNECTIVITY
  6001, 8, 9
  6002, 9, 10
*ELEMENT, TYPE = STRI35, ELSET = FILM
** FILM ELEMENT CONNECTIVITY
  2001, 1, 2, 3
  2002, 1, 3, 4
*BEAM SECTION, SECTION = RECT, MATERIAL = WOOD, ELSET = BEAM1
  6.75, 11.00
*BEAM SECTION, SECTION = RECT, MATERIAL = WOOD, ELSET = EVEPLN
  3.00, 12.25
*SOLID SECTION, MATERIAL = STEEL, ELSET = TRING
  12.0
*SHELL SECTION, MATERIAL = WOOD, ELSET = FILM
  1.0E-03, 3
*MATERIAL, NAME = WOOD
*ELASTIC
  1.800E+06, 4.625
*DENSITY
  5.7E-05
*MATERIAL, NAME = STEEL
*ELASTIC
2.900E+07, 0.3
*DENSITY
7.3E-04
*BOUNDARY
8, 2
15, 1
11, 2
18, 1
TRNODE, 3
*STEP, LINEAR
*STATIC
*DLOAD
FILM, BZ, -250.0
*NODE FILE, NSET = FIXNODES, SUMMARY = NO
U
RF
*EL FILE, SUMMARY = NO
S
*PLOT
LOAD, CAP WITH FILMS
12, 12, 10, 10, 1, 1, 1, 0.5
*VIEWPOINT
500.0, 0.0, 0.0
*DISPLACED, HIDE
U

*END STEP
Appendix D : Program Listing

*****************************************************************************
* MESHGEN *
*****************************************************************************

PROGRAM MAIN
DIMENSION A(3,900), TEMP(3,300), PLN(3,900), N1(300), N2(300)
INTEGER PEL(3,500)

INTEGER NREP, LNUM, NUMSUP, NELPAN, NINTN, PNUM, LPNUM, TRNUM

REAL DIA, RADI, RISE, TRIAX, SL, HT

READ TRIAX PARAMETERS

READ(5,*)DIA, RISE, TRIAX, NELPAN, KEY, TRNUM

COMPUTE BASIC PARAMETERS OF A TRIAX DOME

SL = DIA/(2*TRIAX)
HT = SQRT(SL**2-(SL**2)/4)
PI = 3.14159265358979323846D0
RAD = PI/180.0
DEG = 1.0/RAD
RADI = DIA/(2.0*SIN(PI-2*ATAN(DIA/(2.0*RISE))))
PATHT = RADI-RISE
NINTN = NELPAN * 2 - 1

COMPUTE NUMBER OF MAIN NODES

LNUM = 1
DO 100 I = 1,INT(TRIAX)
    LNUM = LNUM + 6*(I**2-1)*NINTN + 6*I*(NINTN+1)
100 CONTINUE

CALL COORD(NELPAN, NINTN, TRIAX, SL, HT, PATHT, RADI, PI, TEMP, A)
CALL FNUM(TRIAX, N1)
CALL SNUM(TRIAX, N2)
CALL INGEN(TRIAX, NELPAN, NINTN, PI, TEMP, N1, N2, A, DEG, RAD)
CALL PURLIN(TRIAX, NELPAN, NINTN, PI, DEG, RAD, LPNUM, NUMPEL, 1
A, PLN, PEL)
CALL PLNPOST(LPNUM, LNUM, NUMPEL, A, PLN, PEL)

IF (KEY .NE. 0) THEN

Appendix D : Program Listing
GENERATE NODES ON THE TENSION RING

PHI = ATAN(PATHT/(DIA/2.0)) * DEG
A(2,TRNUM) = 0.0
A(3,TRNUM) = PHI
THETA = 60.0 / (2*NELPAN*(INT(TRIAX) + 1))
DO 5 NN = 1, (INT(TRIAX) + 1) * NELPAN * 2 * 6
   A(2,TRNUM+NN) = THETA * NN
   A(3,TRNUM+NN) = PHI
5 CONTINUE

IF (KEY .NE. 0) THEN
   CALL PFMNGEN(P1, DEG, RAD, LNNUM, A)
   LNNUM = LNNUM + (INT(TRIAX) + 1) * NELPAN * 2 * 6
   IF (TRIAX.EQ.INT(TRIAX)) LNNUM = LNNUM-6
ENDIF

CALL NORMAL(NELPAN, NINTN, TRIAX, TRNUM, LNNUM, RAD, PI, RAD, A)
CALL TRING(NINTN, TRNUM, LNNUM)
CALL ALINNNORM(RADI, PI, RAD, A, P)

WRITE(6,10)
10 FORMAT(15X,'TRIAX DOME PARAMETERS'/)
WRITE(6,20)DIA, RADI, RISE, TRIAX, SL, HT, NREFP, LNNUM, NUMSUP, LPNUM
20 FORMAT(5X,'PATTERN DIAMETER' (DIA) = ',F8.3/
$ 5X,'RADIUS OF THE DOME' (RADI) = ',F8.3/
$ 5X,'HEIGHT OF THE DOME' (RISE) = ',F8.3/
$ 5X,'TRIAX NUMBER' (TRIAX) = ',F8.3/
$ 5X,'LENGTH OF BASIC TRIANGLE' (SL) = ',F8.3/
$ 5X,'HEIGHT OF BASIC TRIANGLE' (HT) = ',F8.3/
$ 5X,'NO. OF REFERENCE POINTS' (NREFP) = ',I4/
$ 5X,'NO. OF GENERATION POINTS' (LNNUM) = ',I4/
$ 5X,'NO. OF SUPPORT POINTS' (NUMSUP) = ',I4/
$ 5X,'NO. OF PURLIN NODES' (LPNUM) = ',I4/

WRITE(7,30)
30 FORMAT('***NODAL COORDINATE DATA'
J = 3
A(1,900) = RADI
DO 200 L = 1,LNNUM
   WRITE(7,35)L, A(1,900), A(2,L), A(3,L)
200 CONTINUE

WRITE(8,40)
40 FORMAT('***PURLIN NODE COORDINATE DATA'
DO 300 PNUM = 1,LPNUM
   WRITE(8,45)PNUM+1000, A(1,900), PLN(2,PNUM), PLN(3,PNUM)
300 CONTINUE
WRITE(9,50)
50 FORMAT('**PURLIN ELEMENT CONNECTIVITY')
DO 400 NNN = 1, NUMPEL-1
WRITE(9,55) NNN+1000, PEL(1,NNN), PEL(2,NNN), PEL(3,NNN)
55 FORMAT(3X, I5, ',', I5, ',', I5, ',', I5, ',', I5)
400 CONTINUE
END

******************************************************************************
*                        COORD                                      *
******************************************************************************
SUBROUTINE COORD(NELPAN, NINTN, TRIAX, SL, HT, PATHT, RADI,     
1                  PI, TEMP, A)
DIMENSION A(3,900), TEMP(3,300)
C
C SUBROUTINE TO GENERATE THE NODAL COORDINATES OF THE TRIAX DOME
C
A(2,1) = 0.0
A(3,1) = 90.0
TEMP(2,1) = 0.0
TEMP(3,1) = 90.0
M = 2
N = 2 + 6*NINTN
DO 300 I = 1, INT(TRIAX)
   IF (I.GT.1) THEN
      N = 2
      DO 100 J = 1, I
        N = N + 6 * NINTN * (J*2-1)
        IF (J.LT.I) N = N + 6 * J * (NINTN+1)
      100 CONTINUE
   ENDIF
A(1,N) = SL * I
A(2,N) = 0.0
A(3,N) = 90.0 - ATAN (A(1,N)/PATHT) * 180.0/PI
DO 1 I1 = 1,5
   INC = I1 * I * (NINTN + 1)
   A(2,N+INC) = A(2,N) + 60.0 * I1
   A(3,N+INC) = A(3,N)
  1 CONTINUE
*** WRITE(6,*)'N',N,A(3,N)
TEMP(2,M) = A(2,N)
TEMP(3,M) = A(3,N)
M = M + 1
C
DO 200 K = 1, (I-1)
   N = N + NELPAN * 2
   A(1,N) = SQRT ((HT*I)**2. + (SL/2.*I - SL*K)**2)
   A(2,N) = 30. - ATAN ((SL/2.*I - SL*K) / (HT*I)) * 180./PI
   A(3,N) = 90. - ATAN (A(1,N)/PATHT) * 180./PI
DO 2 I2 = 1,5
   INC = I2 * I * (NINTN + 1)
   A(2,N+INC) = A(2,N) + 60.0 * I2
   A(3,N+INC) = A(3,N)
  2 CONTINUE
*** WRITE(6,*)'N+INC',N+INC
2 CONTINUE
* WRITE(6,*)'N',N,A(3,N)
    TEMP(2,M) = A(2,N)
    TEMP(3,M) = A(3,N)
    M = M + 1
200 CONTINUE
C
    N = N + NELPAN * 2
    A(1,N) = SL * I
    A(2,N) = 50.0
    A(3,N) = 90. - ATAN (A(1,N)/PATHT) * 180./PI
* WRITE(6,*)'N',N,A(3,N)
    TEMP(2,M) = A(2,N)
    TEMP(3,M) = A(3,N)
    M = M + 1
C
300 CONTINUE
RETURN
END

*******************************************************************************
*                        FNNUM                              *
*******************************************************************************
SUBROUTINE FNNUM(TRIAX, N1)
DIMENSION N1(300)
N = 1
M = 1
DO 300 I = 1, INT(TRIAX)
    DO 100 J = 1, I
        N1(N) = M
        N = N + 1
        N1(N) = M
        N = N + 1
        M = M + 1
    100 CONTINUE
    DO 200 K = 1, I
        N1(N) = M
        N = N + 1
        M = M + 1
    200 CONTINUE
    M = M - I
300 CONTINUE
RETURN
END

*******************************************************************************
*                        SNUM                                         *
*******************************************************************************
SUBROUTINE SNUM(TRIAX, N2)
DIMENSION N2(300)
N2(1) = 2
N2(2) = 3
N2(3) = 3
N = 4
M = 4
DO 300 I = 2, INT(TRIAX)
N2(N) = M
N = N + 1
M = M + 1
DO 100 J = 1, (I-1)
   N2(N) = M
   N = N + 1
   N2(N) = M
   N = N + 1
   M = M + 1
100  CONTINUE
   N2(N) = M
   M = M - (I-1)
   N = N + 1
   DO 200 K = 1, I
      N2(N) = M
      N = N + 1
      M = M + 1
200  CONTINUE
300  CONTINUE
RETURN
END

***********************************************************************

* INGEN
***********************************************************************

SUBROUTINE INGEN(TRIAX, NELPAN, NINTN, PI, TEMP, N1, N2, A, DEG, RAD)
DIMENSION A(3,900), TEMP(3,300), N1(300), N2(300)
NN = 2
M = 1
DO 600 J = 1, INT(TRIAX)
   DO 300 K = 1, J*2
      IF (J.EQ.1 .AND. K.EQ.2) TEMP(2,1) = 60.0
      N = NN
      THETA1 = TEMP(2,N1(M)) * RAD
      THETA2 = TEMP(2,N2(M)) * RAD
      PHI1 = TEMP(3,N1(M)) * RAD
      PHI2 = TEMP(3,N2(M)) * RAD
      DO 200 L = 1, NINTN
         F = (1.0 - L/(NINTN + 1.0))
         IF (THETA1.EQ.THETA2 .AND. (THETA1.EQ.0.0
1         .OR. THETA1.EQ.(60.0*RA)) THEN
            F1 = F
            F2 = 1.0 - F
            A(2,N) = THETA1 * DEG
            A(3,N) = (F1 * PHI1 + F2 * PHI2) * DEG
            ELSE
               CALL INGEN1(PHI1, PHI2, THETA1, THETA2,
               F, N, PI, A)
1        ENDIF
   600  CONTINUE
   IF (A(2,N).LT.59.90) THEN
      INC = I4 * (2*J -1)
      A(2,N+INC) = A(2,N) + 60.0 * I4
      A(3,N+INC) = A(3,N)
      ENDIF

Appendix D : Program Listing 75
4 CONTINUE
N = N + 6 * (J*2-1)
IF (K.EQ.1 .AND. L.EQ.NINTN) MM = N + 1
200 CONTINUE
M = M + 1
NN = NN + 1
300 CONTINUE
N = MM
DO 500 KK = 1, J
THETA1 = TEMP(2,N1(M)) * RAD
THETA2 = TEMP(2,N2(M)) * RAD
PHI1 = TEMP(3,N1(M)) * RAD
PHI2 = TEMP(3,N2(M)) * RAD
DO 400 LL = 1, NINTN
F = (1.0 - LL/(NINTN + 1.0))
CALL INGEN1(PHI1,PHI2,THETA1,THETA2,
F,N,PI,A)
1 DO 5 I5 = 1,5
INC = I5 * J*(NINTN + 1)
A(2,N+INC) = A(2,N) + 60.0 * I5
A(3,N+INC) = A(3,N)
5 CONTINUE
N = N + 1
400 CONTINUE
N = N + 1
M = M + 1
500 CONTINUE
NN = 2
DO 550 I = 1,J
NN = NN + 6 * NINTN * (I*2-1) + 6 * (NELPAN*2) * I
550 CONTINUE
600 CONTINUE
RETURN
END

******************************************************************************
*                          INGEN1                                           *
******************************************************************************
SUBROUTINE INGEN1(PHI1,PHI2,THETA1,THETA2,F,N,PI,AORPLN)
DIMENSION AORPLN(3,900)
C
DEG = 180.0/PI
RAD = 1.0/DEG
PSI1 = ACOS (COS(PHI1) * COS(THETA1))
PSI2 = ACOS (COS(PHI2) * COS(THETA2))
C
THEPSI1 = ACOS (TAN(THETA1) / TAN(PSI1))
THEPSI2 = ACOS (TAN(THETA2) / TAN(PSI2))
PSI12 = ABS(THEPSI2-THEPSI1)
*
WRITE(6,*)'***N***',N
*
WRITE(6,*)'PSI',PSI1*DEG,PSI2*DEG,'THEPSI',
*
1 THEPSI1*DEG,THEPSI2*DEG,'PSI12',PSI12*DEG
C
IF (PSI12.LE.0.0001) THEN
C
NOTE: THIS IS ONLY FOR 1 ELEM TRIAX
IF (PSI1.LT.PSI2) THEN
   PSI3 = PSI1 + F * (PSI2-PSI1)
ELSE
   PSI3 = PSI2 + F * (PSI1-PSI2)
ENDIF
THEPSI3 = THEPSI1
GO TO 100
ELSE
CALL INGEN2(PSI1,PSI2,PSI12,ALPHA,ALPSI2,DEG)
WRITE(6,*)'ALPHA',ALPHA*DEG,'ALPSI2',ALPSI2*DEG
ALPHA = F * ALPHA
CALL INGEN3(ALPHA,PSI2,ALPSI2,PSI3,PSI23,DEG)
C
WRITE(6,*)'PSI3',PSI3*DEG,'PSI23',PSI23*DEG
IF (THEPSI1.GT.THEPSI2) THEN
   THEPSI3 = PSI23 + THEPSI2
ELSE
   THEPSI3 = PSI23 + THEPSI1
ENDIF
100 AORPLN(3, N) = ABS(ASIN (SIN(THEPSI3) * SIN(PSI3))) * DEG
AORPLN(2, N) = ABS(ATAN (TAN(PSI3) * COS(THEPSI3))) * DEG
ENDIF
WRITE(6,*)'*****N*****', N
C
RETURN
END

******************************************************************************
* INGEN2 *
******************************************************************************
SUBROUTINE INGEN2(PSI1,PSI2,PSI12,ALPHA,ALPSI2,DEG)
C
B1 = ATAN (COS((PSI2-PSI1)/2) / (COS((PSI2+PSI1)/2))
1   * TAN(PSI12/2))
B2 = ATAN (SIN((PSI2-PSI1)/2) / (SIN((PSI2+PSI1)/2))
1   * TAN(PSI12/2))
WRITE(6,*)'B',B1*DEG,B2*DEG
ALPHA = 2 * ATAN (COS(B1) * TAN ((PSI1+PSI2)/2) / COS(B2))
ALPSI2 = ABS(2 * B1 - (B1+B2))
C
RETURN
END

******************************************************************************
* INGEN3 *
******************************************************************************
SUBROUTINE INGEN3(ALPHA,PSI2,ALPSI2,PSI3,PSI23,DEG)
C
B1 = ATAN (COS((PSI2-ALPHA)/2) / (COS((PSI2+ALPHA)/2))
1   * TAN(ALPSI2/2))
B2 = ATAN (SIN((PSI2-ALPHA)/2) / (SIN((PSI2+ALPHA)/2))
1   * TAN(ALPSI2/2))
WRITE(6,*)'B',B1*DEG,B2*DEG
PSI3 = 2 * ATAN (COS(B1) * TAN ((ALPHA+PSI2)/2) / COS(B2))
PSI23 = ABS(B1 - B2)
C
RETURN
END

*****************************************************************************
* PURLIN
*****************************************************************************
SUBROUTINE PURLIN( TRIAX, NELPAN, NINTN, PI, DEG, RAD,
1       LPNUM, NUMPEL, A, PLN, PEL)
C
C DIMENSION A(3,900), PLN(3,900), PEL(3,500)
C
C INTEGER PLNINTV, PNUM, NUMI, PINTN, LPNUM
C
C NELPAN = 2
PLNINTV = 2
C
C PNUM = 1
NN = 2
DO 400 I = 1, INT(TRIAX)
   DO 300 J = PLNINTV, NINTN, PLNINTV
      N = NN + 6 * (J-1) * (2*I-1)
      PNUMI = PNUM
C
C CALCULATION OF THE NO. OF PURLIN NODES
C IN ONE SECTOR OF A PURLIN CIRCLE
C
C NPSECT = I * ((J / PLNINTV) * NELPAN * 2)
1 + (I - 1.0) * ((INT((NINTN - (J - PLNINTV)) / PLNINTV))
1 * NELPAN * 2)
C
C WRITE(6,*)'NPSECT', NPSECT
C
DO 200 K = 1, 2*I-1
   IF (MOD(K,2) .NE. 0) THEN
      PINTN = (J / PLNINTV) * NELPAN * 2 - 1
   ELSE
      PINTN = (INT ((NINTN - (J - PLNINTV)) / PLNINTV))
1 * NELPAN * 2 - 1
   ENDF
THETA1 = A(2,N) * RAD
THETA2 = A(2,N+1) * RAD
PHI1 = A(3,N) * RAD
PHI2 = A(3,N+1) * RAD
PLN(2,PNUM) = A(2,N)
PLN(3,PNUM) = A(3,N)
DO 5 I5 = 1, 5
   INC = I5 * NPSECT
   PLN(2,PNUM+INC) = PLN(2,PNUM) + 60.0 * I5
   PLN(3,PNUM+INC) = PLN(3,PNUM)
5 CONTINUE
PNUM = PNUM + 1
DO 100 L = 1, PINTN
   F = 1.0 - L / (PINTN + 1.0)
   CALL INGEN1 (PHI1, PHI2, THETA1, THETA2, F,
1

PNUM, PI, PLN)

DO 6 I6 = 1, 5
   INC = I6 * NPSECT
   PLN(2,PNUM+INC) = PLN(2,PNUM) + 60.0 * I6
   PLN(3,PNUM+INC) = PLN(3,PNUM)
6 CONTINUE
   PNUM = PNUM + 1
100 CONTINUE
   N = N + 1
200 CONTINUE
   PNUM = PNUMI + 6 * (PNUM - PNUMI)
300 CONTINUE
   NN = 2
   DO 350 JJ = 1, I
      NN = NN + 6 * NINTN * (2*JJ-1) + 6 * JJ * (NINTN+1)
350 CONTINUE
400 CONTINUE
   LPNUM = PNUM - 1.0
C
   CALL PELMGEN(TRIAX, NINTN, NELPLN, PLNINTV, NUMPFL, PEL)
C
   RETURN
END

*******************************************************************************
*   PFMGEN   *
*******************************************************************************

SUBROUTINE PFMGEN(PI, DEG, RAD, LNUM, A)
DIMENSION A(3,900), NN(3)
C
   READ(5,*),NSET, INC
C
   DO 300 NN = 1, NSET
      READ(5,*), NODE1, NODE2, NMN, (MN(I),I=1,NMN)
      WRITE(6,*), 'NNN ', NODE1, ' INC ', INC
C
      THETA1 = A(2,NODE1) * RAD
      THETA2 = A(2,NODE2) * RAD
      PHI1 = A(3,NODE1) * RAD
      PHI2 = A(3,NODE2) * RAD
      WRITE(6,*), 'THETA', THETA1*DEG, THETA2*DEG, 'PHI', PHI1*DEG
      1      PHI2*DEG
C
   DO 200 I = 1, NMN
      F = (1.0 - I/(NMN + 1.0))
      N = MN(I)
      WRITE(6,*), 'F', F, 'N', N
      IF (THETA1 .EQ. THETA2 .AND. (THETA1 .EQ. 0.0 .OR. THETA1 .EQ. (60.0 * RAD))) THEN
          F1 = F
          F2 = 1.0 - F
          A(2,N) = THETA1 * DEG
          A(3,N) = (F1 * PHI1 + F2 * PHI2) * DEG
          WRITE(6,*),'******'
      ELSE

Appendix D : Program Listing
CALL INGEN1(PHI1, PHI2, THETA1, THETA2, F, N, PI, A)
* WRITE(6,*')'N', N
ENDIF
* WRITE(6,*')'N', N
* CHECK
   DO 100 I1 = 1,5
      A(2,(N+INC*I1)) = A(2,N) + 60.0 * I1
      A(3,(N+INC*I1)) = A(3,N)
   100 CONTINUE
* WRITE(6,*')'N', N
200 CONTINUE
*N WRITE(6,*')A(2,33), A(3,33), A(2,34), A(3,34)
300 CONTINUE
LNNUM = N + 5 * INC
RETURN
END
*****************************************************************************************************************************************
* PPFLngen
*****************************************************************************************************************************************
SUBROUTINE PPFLngen(PI, DEG, RAD, A, PLN, LPNUM)
DIMENSION A(3,900), PLN(3,900)
C
READ(5,*')NSET, INC
C
DO 300 NN = 1, NSET
READ(5,*') NODE1, NODE2, NMN, MN
C SINCE MN VALUE IS 1000 + SOMETHING, IT SHOULD BE REDUCED BY 1000
MN = MN - 1000
* WRITE(6,*')'NODE1', NODE1, 'INC', INC
C
THETA1 = A(2,NODE1) * RAD
THETA2 = A(2,NODE2) * RAD
PHI1 = A(3,NODE1) * RAD
PHI2 = A(3,NODE2) * RAD
* WRITE(6,*')'THETA', THETA1*DEG, THETA2*DEG, 'PHI', PHI1*DEG
* 1 ,PHI2*DEG
C
DO 200 I = 1, NMN
   F = (1.0 - 1/(NMN + 1.0) )
   N = MN + I - 1
   WRITE(6,*')'F', F, 'N', N
   IF (THETA1 .EQ. THETA2 .AND. (THETA1 .EQ. 0.0 .OR. THETA1 .EQ. (50.0 * RAD)))) THEN
      F1 = F
      F2 = 1.0 - F
      PLN(2,N) = THETA1 * DEG
      PLN(3,N) = (F1 * PHI1 + F2 * PHI2) * DEG
* WRITE(6,*')'*****'
   ELSE
      CALL INGEN1(PHI1, PHI2, THETA1, THETA2, F, N, PI, PLN)
* WRITE(6,*')'N', N
   ENDIF
   WRITE(6,*')'N', N
* CHECK
DO 100 I1 = 1.5
   PLN(2,(N+INC*I1)) = PLN(2,N) + 60.0 * I1
   PLN(3,(N+INC*I1)) = PLN(3,N)
100   CONTINUE
*   WRITE(6,*)'N', N
200   CONTINUE
*   WRITE(6,*)PLN(2,33), PLN(3,33), PLN(2,34), PLN(3,34)
300   CONTINUE
   LPNUM = N + 5 * INC
RETURN
END

******************************************************************************
*   PELMGEN
******************************************************************************

C
SUBROUTINE PELMGEN(TRIAX, NINTN, NELPLN, PLNINTV, NUMPEL, PEL)
C
INTEGER PEL(3,500)
INTEGER FPNODE, LPNODE, PLNINTV
C
   FPNODE = 1001
   NUMPEL = 1
C
   DO 300 I = 1, INT(TRIAX)
      DO 200 J = PLNINTV, NINTN, PLNINTV

C
   CALCULATION OF THE NO. OF PURLIN NODES
   IN ONE SECTOR OF A PURLIN CIRCLE
C
   NPSEQ = I * ((J / PLNINTV) * NELPLN * 2)
1      + (I - 1.0) * ((INT((NINTN - (J - PLNINTV))) / PLNINTV))
     1
      * NELPLN * 2)
C
   LPNODE = FPNODE + 6 * NPSEQ - 1
   DO 100 K = FPNODE, (LPNODE-2), 2
      PEL(1,NUMPEL) = K
      PEL(2,NUMPEL) = K + 1
      PEL(3,NUMPEL) = K + 2
      NUMPEL = NUMPEL + 1
100   CONTINUE
      PEL(1,NUMPEL) = LPNODE - 1
      PEL(2,NUMPEL) = LPNODE
      PEL(3,NUMPEL) = FPNODE
      NUMPEL = NUMPEL + 1
      FPNODE = LPNODE + 1
   200   CONTINUE
   300   CONTINUE
C
*   WRITE(6,10)
*10   FORMAT(8X,'NUMPEL',5X,'NODE 1',11X,'NODE 2',11X,'NODE 3'/)
*   DO 900 NNN = 1, NUMPEL-1
*   WRITE(6,*)(NNN+1000),(PEL(III,NNN),III=1,3)
*900   CONTINUE
RETURN

Appendix D: Program Listing
END

************************************************************************************

*   PLNPSTP
*  

SUBROUTINE PLNPSTP(LPNUM, LNUM, NUMPEL, A, PLN, PEL)

C
DIMENTION A(3,900), PLN(3,900)
INTEGER PEL(3,500)

C
DO 400 I = 1, LPNUM
   * IF(MOD(PLN(2,I),59.99) .LE. 0.10) THEN
      DO 300 J = 1, LNUM
         IF(ABS(A(2,J) - PLN(2,I)) .LE. 0.10 .AND.
            1           ABS(A(3,J) - PLN(3,I)) .LE. 0.10) THEN
            DO 200 K = 1, NUMPEL
               DO 100 L = 1,3
                  IF(PEL(L,K) .EQ. I+100) PEL(L,K) = J
               100 CONTINUE
            200 CONTINUE
   END IF
   END IF
400 CONTINUE
RETURN
END

************************************************************************************

*  NORMAL
*  

SUBROUTINE NORMAL(NELPAN, NINTN, TRIAX, TRNUM, LNUM, RADI, $PI, RAD, A)

DIMENTION A(3,900)
INTEGER TRNUM

C
SUBROUTINE TO GENERATE THE DIRECTION COSINES OF THE NORMALS AT
THE BASIC TRIAX NODES

C
DC'S FOR NODE-1

C
DCN1 = 0.0
DCN2 = 0.0
DCN3 = -1.0
N = 1
WRITE(3,5)N, RADI, A(2,N), A(3,N), DCN1, DCN2, DCN3
5 FORMAT(3X, I4, 3(' ', F10.4), 3(' ', F10.6))
N = 2 + 6*NINTN
DO 300 I = 1, INT(TRIAX)
   IF (I.GT.1) THEN
      N = 2
      DO 100 J = 1, I
         N = N + 6 * NINTN * (J**2-1)
         IF (J.LT.I) N = N + 6 * J * (NINTN+1)
      100 CONTINUE
   END IF
300 CONTINUE
ENDIF

C
DO 1 I1 = 1, 6
   INC = (I1 - 1) * I1 * (NINTN + 1)
   DCN1 = - COS(A(2, N+INC)*RAD) * COS(A(3, N+INC)*RAD)
   DCN2 = - SIN(A(2, N+INC)*RAD) * COS(A(3, N+INC)*RAD)
   DCN3 = - SIN(A(3, N+INC)*RAD)
   WRITE(3, 10) N+INC, RADI, A(2, N+INC), A(3, N+INC), DCN1, DCN2, DCN3
1 CONTINUE

C
DO 200 K = 1, (I-1)
   N = N + NELPAN * 2
DO 2 I2 = 1, 6
   INC = (I2 - 1) * I2 * (NINTN + 1)
   DCN1 = - COS(A(2, N+INC)*RAD) * COS(A(3, N+INC)*RAD)
   DCN2 = - SIN(A(2, N+INC)*RAD) * COS(A(3, N+INC)*RAD)
   DCN3 = - SIN(A(3, N+INC)*RAD)
   WRITE(3, 20) N+INC, RADI, A(2, N+INC), A(3, N+INC), DCN1, DCN2, DCN3
20 CONTINUE
200 CONTINUE

C
300 CONTINUE

C
DCN1 = 0.0
DCN2 = 0.0
DCN3 = -1.0
DO 400 NN = TRNUM, LNUM, (NINTN + 1)
   WRITE(3, 30) NN, RADI, A(2, NN), A(3, NN), DCN1, DCN2, DCN3
30 CONTINUE
400 CONTINUE
RETURN
END

************************************************************************************
* TRING
************************************************************************************

SUBROUTINE TRING(NINTN, TRNUM, LNUM)
INTEGER TRNUM

SUBROUTINE TO GENERATE THE TENSION RING ELEMENTS

I = 1
DO 100 NN = TRNUM, (LNUM - 4), (NINTN + 1)
   WRITE(4, 10) 6000 + I, NN, NN+NINTN+1
10 FORMAT(3X, I5, 2(' '), I5))
   I = I + 1
100 CONTINUE
WRITE(4, 10) 6000 + I, NN, TRNUM
RETURN
END

*****************************************************************************
* ELMGEN *
*****************************************************************************

PROGRAM MAIN
C
C PROGRAM TO COMPUTE THE BEAM ELEMENT CONNECTIVITY
C
INTEGER C(10,15,100), S(5,150)
C
TRIAX = 3.4345
NELPAN = 2
NINTN = NELPAN * 2 - 1
N = 2
DO 400 I = 1, INT(TRIAX)
   DO 200 J = 1, NINTN
      DO 100 K = 1, 6*(I*2-1)
         C(I,J,K) = N
      * WRITE(6,*)'I',I,'J',J,'C',C(I,J,K)
         N = N + 1
   100 CONTINUE
   CONTINUE
   DO 300 L = 1, 6*I*(NINTN+1)
      S(I+1,L) = N
   * WRITE(6,*)'I',I,'S',S(I+1,L)
      IF (L.EQ.1) S(I+1,6*I*(NINTN+1)+1) = N
         N = N + 1
   300 CONTINUE
   CONTINUE
   CALL ELMGEN(NELPAN,NINTN,TRIAX,C,S)
END

*****************************************************************************
SUBROUTINE ELMGEN(NELPAN,NINTN,TRIAX,C,S)
C
INTEGER C(10,15,100), S(5,150), B(3,500)
C
DO 50 JJJ=1,6
   S(1,JJJ)=1
   CONTINUE
   NUMEL = 1
C
   DO 800 I = 1, INT(TRIAX)
      DO 600 J = 1, NELPAN
         NODE1 = J*2-2
         NODE2 = J*2-1
         NODE3 = J*2
         IF (I.EQ.1 .AND. J.LT.NELPAN .OR.
            1 J.GT.1 .AND. J.LT.NELPAN) THEN
            DO 100 K = 1, 6 * (I*2-1)
               IF (J.EQ.1) THEN
                  B(1,NUMEL) = S(1,K)
               ELSE
                  B(1,NUMEL) = C(I,NODE1,K)
               ENDIF
            100 CONTINUE
         ELSE
            B(1,NUMEL) = C(I,NODE3,K)
         ENDIF
   600 CONTINUE
   CONTINUE
   RETURN
END
B(2,NUMEL) = C(I,NODE2,K)
B(3,NUMEL) = C(I,NODE3,K)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  NUMEL = NUMEL + 1

100 CONTINUE
ELSEIF ((I.GT.1 .AND. J.EQ.1) .AND. NELPAN.GT.1) THEN
  INCC = 1
  INCS = 1
  DO 300 KK = 1, 6
    DO 200 L = 1, I*2-2, 2
      B(1,NUMEL) = S(I,INCS)
      B(2,NUMEL) = C(I,NODE2,INCC)
      B(3,NUMEL) = C(I,NODE3,INCC)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  NUMEL = NUMEL + 1
  INCC = INCC + 1
  B(1,NUMEL) = S(I,INCS)
  B(2,NUMEL) = C(I,NODE2,INCC)
  B(3,NUMEL) = C(I,NODE3,INCC)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  NUMEL = NUMEL + 1
  INCC = INCC + 1
  INCS = INCS + NINTN + 1
  IF (L.EQ. I*2-3) THEN
    B(1,NUMEL) = S(I,INCS)
    B(2,NUMEL) = C(I,NODE2,INCC)
    B(3,NUMEL) = C(I,NODE3,INCC)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  NUMEL = NUMEL + 1
ENDIF

200 CONTINUE
  INCC = KK *( I*2-1 ) + 1

300 CONTINUE
ELSEIF (J.EQ. NELPAN) THEN
  INCC = 1
  INCS = 1
  DO 500 KKK = 1, 6
    B(1,NUMEL) = C(I,NODE1,INCC)
    B(2,NUMEL) = C(I,NODE2,INCC)
    B(3,NUMEL) = S(I+1,INCS)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  NUMEL = NUMEL + 1
  INCC = INCC + 1
  INCS = INCS + NINTN + 1
  DO 400 LL = 1, (I-1)
    B(1,NUMEL) = C(I,NODE1,INCC)
    B(2,NUMEL) = C(I,NODE2,INCC)
    B(3,NUMEL) = S(I+1,INCS)

* WRITE(6,*),'NUMEL',NUMEL,(B(JJ,NUMEL),JJ=1,3)
  INCC = INCC + 1
  NUMEL = NUMEL + 1
  B(1,NUMEL) = C(I,NODE1,INCC)
  B(2,NUMEL) = C(I,NODE2,INCC)
B(3,NUMEL) = S(I+1,NICS)

WRITE(6,*)'NUME',NUMEL,B(JJ,NUMEL),JJ=1,3
NUMEL = NUMEL + 1
INCC = INCC + 1
INCS = INCS + NINTN + 1

400 CONTINUE
500 CONTINUE
ENDIF
600 CONTINUE
NODE1 = 1
NODE2 = 2
NODE3 = 3

DO 700 JJ = 1, NELPAN*I*6
    B(1,NUMEL) = S(I+1,NODE1)
    B(2,NUMEL) = S(I+1,NODE2)
    B(3,NUMEL) = S(I+1,NODE3)
    NODE1 = NODE1 + 2
    NODE2 = NODE2 + 2
    NODE3 = NODE3 + 2
    NUMEL = NUMEL + 1

700 CONTINUE
800 CONTINUE
WRITE(10,10)
10 FORMAT('**BEAM ELEMENT CONNECTIVITY')

DO 900 NNN = 1, NUMEL-1
    WRITE(10,20)NNN, B(1,NNN), B(2,NNN), B(3,NNN)
20 FORMAT(3X,I5,','','','','I5,','','I5)
900 CONTINUE
RETURN
END

******************************************************************************

******************************************************************************

PROGRAM MAIN

C
C DIRECTION COSINES OF THE NORMALS AT ALL THE NODES
C
PI = 3.14159265358979323846DO
RAD = PI/180.0

C

DO 100 I = 1, 529
    READ(4,*)N, A1, A2, A3
    DCN1 = - COS(A2*RAD) * COS(A3*RAD)
    DCN2 = - SIN(A2*RAD) * COS(A3*RAD)
    DCN3 = - SIN(A3*RAD)
    WRITE(6,10)N, A1, A2, A3, DCN1, DCN2, DCN3
100 CONTINUE
10 FORMAT(1X, I5, 3(’,’, F10.5), 3(’,’, F10.7))

C

DO 200 J = 1, 341
    READ(5,*)N, P1, P2, P3
    DCN1 = - COS(P2*RAD) * COS(P3*RAD)
    DCN2 = - SIN(P2*RAD) * COS(P3*RAD)
    DCN3 = - SIN(P3*RAD)
WRITE(7,10)N, P1, P2, P3, DCN1, DCN2, DCN3
200  CONTINUE
    STOP
    END

***********************************************************************
*                             LOCAL1                               *
***********************************************************************

PROGRAM MAIN

C
C DIRECTION COSINES OF THE LOCAL 1 AXIS OF THE BEAM CROSS SECTION
C
DIMENSION A(3,1500)
INTEGER ELNUM
C
PI = 3.14159265358979323846D0
RAD = PI/180.0
DEG = 1.0/RAD
C
READ(3,*)NUMNODE
DO 100 K = 1, NUMNODE
    READ(3,*)N, A(1,N), A(2,N), A(3,N)
100  CONTINUE
C
READ(4,*)NUMNODE
DO 200 J = 1, NUMNODE
    READ(4,*)N, A(1,N), A(2,N), A(3,N)
200  CONTINUE
C
READ(5,*)NUMEL, CS1, CS2
DO 300 I = 1, NUMEL
    READ(5,*)ELNUM, N1, N2, N3
C
    PICK THE NODAL COORDINATES FOR CALCULATING DC'S
    THETA1 = A(2,N1) * RAD
    THETA2 = A(2,N3) * RAD
    PHI1  = A(3,N1) * RAD
    PHI2  = A(3,N3) * RAD

    WRITE(7,*)THETA1*DEG, THETA2*DEG, PHI1*DEG, PHI2*DEG
C
C DIRECTION COSINE CALCULATION
C
    C1 = SIN(THETA1) * COS(PHI1) * SIN(PHI2)
        1
    - SIN(THETA2) * COS(PHI2) * SIN(PHI1)
C
    C2 = - COS(THETA1) * COS(PHI1) * SIN(PHI2)
        1
    + COS(THETA2) * COS(PHI2) * SIN(PHI1)
C
    C3 = COS(THETA1) * COS(PHI1) * SIN(THETA2) * COS(PHI2)
        1
    - COS(THETA2) * COS(PHI2) * SIN(THETA1) * COS(PHI1)
C
PRINT THE *ELEMENT AND *BEAM SECTION CARDS
C
WRITE(6,10)ELNUM
10  FORMAT(' *ELEMENT, TYPE=B32, ELSET=B', I4)
WRITE(6,20)ELNUM, N1, N2, N3
20 FORMAT(3X, I5, ', ', I5, ', ', I5, ', ', I5)
WRITE(7, 30) ELNUM
30 FORMAT('BEAM SECTION, SECTION=RECT, MATERIAL=WOOD, ELSET=B', I4)
WRITE(7, 40) CS1, CS2
40 FORMAT(F6.2, ', ', 2X, F6.2)
WRITE(7, 50) C1, C2, C3
50 FORMAT(4X, F7.4, ', ', F7.4, ', ', F7.4)
300 CONTINUE
STOP
END
PROGRAM MAIN

INTEGER TRNUM, TREL, NELPAN, B(3,100), P(3,100), T(3,100)

TRIA = 3.5
NELPAN = 2.
* WRITE(6,*)'MAKE SURE YOU CHANGE THE TRIAX NO.', 'TRIA = ', TRIAX

GENERATE BEAM ELEMENT CONNECTIVITY
READ(5,*)NPASS
DO 250 NP1 = 1, NPASS
READ(5,*)NSET, ELINC, NINC1, NINC2, NINC3
DO 200 N1 = 1, NSET
READ(5,*)ELNUM, NODE1, NODE2, NODE3
IF (NP1.EQ.1 .AND. N1.EQ.1) INUMB = ENUM - 1
ELNUM = ENUM - INUMB
B(1,ELNUM) = NODE1
B(2,ELNUM) = NODE2
B(3,ELNUM) = NODE3

DO 100 I = 1, 5
   NUMB = ENUM + ELINC * I
   B(1,NUMB) = NODE1 + NINC1 * I
   B(2,NUMB) = NODE2 + NINC2 * I
   B(3,NUMB) = NODE3 + NINC3 * I
* WRITE(6,*)NUMB, (B(K,NUMB), K=1,3)
100 CONTINUE
200 CONTINUE
250 CONTINUE

* IF (KEY.EQ.0) THEN

GENERATE PURLIN ELEMENT CONNECTIVITY
READ(5,*)NPASS
DO 450 NP2 = 1, NPASS
READ(5,*)NSET, ELINC, NINC1, NINC2, NINC3
DO 400 N2 = 1, NSET
READ(5,*)ELNUM, NODE1, NODE2, NODE3
IF (NP2.EQ.1 .AND. N2.EQ.1) INUMP = ENUM - 1
ELNUM = ENUM - INUMP
P(1,ELNUM) = NODE1
P(2,ELNUM) = NODE2
P(3,ELNUM) = NODE3

DO 300 J = 1, 5
   NUMP = ENUM + ELINC * J
   P(1,NUMP) = NODE1 + NINC1 * J
   P(2,NUMP) = NODE2 + NINC2 * J
   P(3,NUMP) = NODE3 + NINC3 * J
* WRITE(6,*)NUMP, (P(K,NUMP), K=1,3)
300 CONTINUE

Appendix D: Program Listing 89
400 CONTINUE
450 CONTINUE
* ENDIF
C GENERATE TENSION RING ELEMENTS
READ(5,*)TRNUM
TREL = 0
LNUM = TRNUM + (6 * (INT(TRIAX) + 1) * 2 * NELPAN) - 2
DO 500 NN = TRNUM, LNUM, 2
   TREL = TREL + 1
   T(1,TREL) = NN
   T(2,TREL) = NN + 1
   T(3,TREL) = NN + 2
* WRITE(6,*)'TREL', TREL, (T(KK,TREL), KK = 1,3)
500 CONTINUE
   T(3,TREL) = TRNUM
C BEAM ELEMENT CONNECTIVITY
WRITE(11,10)
10 FORMAT('**BEAM ELEMENT CONNECTIVITY(PERIPHERAL FIELD)')
DO 600 II = 1, NUMB
   WRITE(11,40)II+INUMB, B(1,II), B(2,II), B(3,II)
600 CONTINUE
C T-RING ELEMENT CONNECTIVITY
WRITE(11,20)
20 FORMAT('**T-RING ELEMENT CONNECTIVITY')
DO 700 I2 = 1, NUMP
   WRITE(11,40)I2+INUMP, P(1,I2), P(2,I2), P(3,I2)
700 CONTINUE
C PURLIN ELEMENT CONNECTIVITY
WRITE(11,30)
30 FORMAT('**PURLIN ELEMENT CONNECTIVITY')
DO 800 I3 = 1, TREL
   WRITE(11,40)I3+5000, T(1,I3), T(2,I3), T(3,I3)
800 CONTINUE
STOP
END
PROGRAM MAIN

C GENERATION OF FILM ELEMENT CONNECTIVITY
C
INTEGER NUMEL, TNC, CNUM, REFMN1, REFPN1, REFPN2, LMINC, UMINC
1     LPINC, UPINC, ELCNUM, ELINC, N1, N2, N3, PRENUM, E(3,2000)
C
NUMEL = 0
PRENUM = 0
READ(5,*)TNC, NSPEL
C TNC - TOTAL NUMBER OF CIRCLES
C
DO 200 I = 1, TNC
   READ(5,*)CNUM, NEL, REFMN1, REFMN2, REFPN1, REFPN2, LMINC,
1     UMINC, LPINC, UPINC, ELINC
   IF (ELINC .EQ. 0) ELINC = NEL
C MODIFY REFPN1 & REFPN2
   REFPN1 = REFPN1 - 1
   REFPN2 = REFPN2 - 1
   IF (REFPN2 .LT. 1000) REFPN2 = 10000
   DO 100 J = 1, NEL
      READ(5,*)ELNUM, N1, N2, N3
      E(1,ELNUM) = N1
      E(2,ELNUM) = N2
      E(3,ELNUM) = N3
C INCREATING N1:
   IF(N1 .LT. 1000) THEN
      DO 1 K1 = 1, 5
         F,ELNUM+ELINC*K1) = N1 + LMINC*K1
1     CONTINUE
      ELSE
         DO 2 K2 = 1, 5
            E(1,ELNUM+ELINC*K2) = N1 + LPINC*K2
2      CONTINUE
   ENDIF
C INCREATING N2:
   IF(N2 .LT. 1000) THEN
      DO 3 K3 = 1, 5
         E(2,ELNUM+ELINC*K3) = N2 + UMINC*K3
3     CONTINUE
      ELSE
         DO 4 K4 = 1, 5
            E(2,ELNUM+ELINC*K4) = N2 + UPINC*K4
4      CONTINUE
   ENDIF
C INCREATING N3:
   IF(N3 .LT. 1000) THEN
      IF (N3 .GT. REFMN1 .AND. N3 .LT. REFMN2) THEN
         DO 5 K5 = 1, 5
            E(3,ELNUM+ELINC*K5) = N3 + LMINC*K5
5        CONTINUE
ELSE
    DO 6 K6 = 1, 5
    E(3,E1NUM+ELINC*K6) = N3 + UMINC*K6
    CONTINUE
ENDIF
ELSE
    IF (N3 .GT. REFPN1 .AND. N3 .LT. REFPN2) THEN
        DO 7 K7 = 1, 5
        E(3,E1NUM+ELINC*K7) = N3 + LPINC*K7
        CONTINUE
    ELSE
        DO 8 K8 = 1, 5
        E(3,E1NUM+ELINC*K8) = N3 + UPINC*K8
        CONTINUE
    ENDIF
100 CONTINUE
    IF (PRENUM .NE. CNUM) THEN
        PRENUM = CNUM
        NUMEL = NUMEL + 6 * ELINC
    WRITE(6,*)'CNUM', CNUM, 'NUMEL', NUMEL
ENDIF
C
MODIFY THE NODES FOR THE LAST 3(?) ELEMENTS OF THE CIRCLE
C
    IF (E(1,LEL-2) .EQ. REFPN2) E(3,LEL-2) = REFPN1
    IF (E(1,LEL-1) .EQ. REFPN2) E(1,LEL-1) = REFPN1
    IF (E(3,LEL-1) .EQ. REFPN2) E(3,LEL-1) = REFPN1
    E(1,LEL) = REFPN1
    E(3,LEL) = REFPN2
200 CONTINUE
    IF(NSPEL .NE. 0) THEN
        DO 250 II = 1, NSPEL
            READ(5,*)E1NUM, N1, N2, N3, ELINC, NINC1, NINC2, NINC3
            E(1,E1NUM) = N1
            E(2,E1NUM) = N2
            E(3,E1NUM) = N3
            DO 9 K9 = 1, 5
                E(1,E1NUM+ELINC*K9) = N1 + NINC1*K9
                E(2,E1NUM+ELINC*K9) = N2 + NINC2*K9
                E(3,E1NUM+ELINC*K9) = N3 + NINC3*K9
9 CONTINUE
250 CONTINUE
ENDIF
C
    DO 300 L = 1, E1NUM +ELINC*5 + 100
        IF (E(1,L) .NE. 0 .AND. E(2,L) .NE. 0 .AND. E(3,L) .NE. 0) THEN
            WRITE(6,10)L+2000, E(1,L), E(2,L), E(3,L)
10 FORMAT(3X, I5, ',', I5, ',', I5, ',', I5, ',', I5)
        END IF
300 CONTINUE
STOP
END
Vita

K. K. Raghu was born in Bangalore, India on May 9, 1962. He graduated from R. V. College of Engineering in 1985. He worked as a project assistant at Indian Institute of Science, Bangalore and in September, 1987, went on to attend graduate school at Virginia Polytechnic Institute and State University, studying structural engineering. He is currently completing the requirements for the Master of Science degree in Civil Engineering.

[Signature]