N89-22956

Definition of NASTRAN Sets By Use Of Parametric Geometry

by

Terry V Baughn, Associate Professor
Department of Civil & Mechanical Engineering
Southern Methodist University
Dallas, Texas

Mehran Tiv, Graduate Student
Department of Civil & Mechanical Engineering
Southern Methodist University
Dallas, Texas

SUMMARY

Many finite element preprocessors describe finite element model geometry with points, lines, surfaces and volumes. One method for describing these basic geometric entities is by use of parametric cubics which are useful for representing complex shapes. The lines, surfaces and volumes may be discretized for follow on finite element analysis. The ability to limit or selectively recover results from the finite element model is extremely important to the analyst. Equally important is the ability to easily apply boundary conditions. Although graphical preprocessors have made these tasks easier, model complexity may not lend itself to easily identify a group of grid points desired for data recovery or application of constraints. A methodology is presented which makes use of the assignment of grid point locations in parametric coordinates. The parametric coordinates provide a convenient ordering of the grid point locations and a method for retrieving the grid point ID's from the parent geometry. The selected grid points may then be used for the generation of the appropriate set and constraint cards.

LIST OF SYMBOLS

x,y,z	cartesian coordinates
F_1 - F_4	parametric blending functions
L	symbol for a line
R	a vector representing a point in space
$\mathbf{R}_{,\xi}$	derivative of the vector **R**
S	symbol for a surface
$\mathbf{S}_1$ - $\mathbf{S}_4$	vectors that contain algebraic coefficients
V	symbol for a volume
ξ	parametric variable
$(\ )^T$	transpose of a vector

INTRODUCTION

The integration of computer aided drafting programs and finite element preprocessors has led to a common practice of passing basic geometry to the finite element model building programs. The basic geometry may consists of lines, points, arcs and splines which describe the geometry of the part the user wants to analyze. Production finite element pre and post processors make it extremely easy for the structural analyst to make models with a large number of elements and grid points. Generally the input to the preprocessor is kept to a minimum and the program will generate automatically the grid points and elements required to discretize the model. The user may need to identify grid points to apply forces, moments, displacements, temperatures, heat transfer coefficients or select a series of grid points for data recovery. Since the generation is automatic, the user has or little no control of the grid point ID's and will have to rely on the graphics capability of the preprocessor to identify them. The process of selecting the grid point ID's for set or constraint cards can become very time consuming and subject to error if they are manually selected by the user. Most of the advanced model building codes have graphics driven techniques to assist in the selection process. In the absence of a robust preprocessor or if the preprocessor has limited grid selection capability an alternate method may be employed using a neutral file of the model.

Many of the pre and post processors are associated with a specific finite element code. The disc files used for the graphics display of the models can be very large. A common method for reducing the disc space required for storing a finite element model or for transferring the model from one code to another is by use of a neutral file. [1][1] The neutral file is not specific to any code but is a data base of finite element model information. The data base should contain information on grid points, element connectivity, materials, constraints, coordinate systems, etc and model geometry. One of the main objectives of the neutral file is to provide an interface between the pre and post processor and external programs. The model data is written to a neutral file which may be used to extract the desired grid point information. The neutral file is generally readable by the system editor.

The selection of the grid points is based on the relationship between the basic or parent geometry and the finite element data. The basic geometry may then be discretized with the finite element grid points and elements. For example a line may have many grid points placed along its length. The finite element model data base in the form of a neutral file should retain the relationship between the line and the associated grid points. In general the neutral file should contain the relationship between all the geometric data used to generate the model and the associated finite element data. If the basic geometric entities are written in terms of parametric geometry, a procedure may be developed to easily obtain grid point ID's on lines, surfaces and volumes.

PARAMETRIC GEOMETRY

Parametric cubics serve as the mathematical foundation to describe the basic geometric entities such as lines, surfaces and volumes. First consider the development of a parametric curve. A parametric line or curve can be generated from

$$\mathbf{R} = S_1\xi^3 + S_2\xi^2 + S_3\xi + S_4 \tag{1}$$

where $\mathbf{R}$ represents a point in space (x,y & z) and ξ is the parametric variable ranging from zero to one. The vectors $\mathbf{R}$ and $\mathbf{S}$ are indicated by bold type. The four vectors, $\mathbf{S}_1$ through $\mathbf{S}_4$ each have three components and may be defined in terms of the end conditions of the line. A parametric curve in a cartesian coordinate system is shown in Figure 1. Equation (1) can be written using the end point information as,

$$\mathbf{R} = (F_1(\xi)\ F_2(\xi)\ F_3(\xi)\ F_4(\xi))\ (\mathbf{R}(0)\ \mathbf{R}(1)\ \mathbf{R}_{,\xi}(0)\ \mathbf{R}_{,\xi}(1))^T \tag{2}$$

---

[1] Numbers in square brackets represent references

where,

$$
\begin{aligned}
F_1(\xi) &= 2\xi^3 - 3\xi^2 + 1 \\
F_2(\xi) &= -2\xi^3 + 3\xi^2 \\
F_3(\xi) &= \xi^3 - 2\xi^2 + \xi \\
F_4(\xi) &= \xi^3 - \xi^2.
\end{aligned}
\tag{3}
$$

Equation (2) is referred to as the geometric form of the parametric cubic curve. [3] The geometric form of a line or curve provides a technique for determining the value of $\mathbf{R}(\xi)$ continuously over the curve with only knowing the starting and ending values and the starting and ending derivatives.

Surfaces are generated from 4 parametric curves and volumes may be defined by four surfaces spaced along a parametric coordinate. The parametric coordinate, ξ, is given a subscript to define the two and three dimensional geometric entities. A sketch of a parametric cubic surface and volume with the parametric coordinates are shown in Figure 2. In Figure 2, the sides of the surface and volume are shown as straight but they may also be curved. It is convenient to define edges on the surface and surfaces on the volume. Figure 2 also shows the edge and surface definitions. Clearly, edges could now be defined from the surfaces on the volume.

After the model has been defined, grids are assigned to the basic entities by specifying the number of grid points desired in a specific parametric direction. Consider the surface shown in Figure 3 which has been discretized with four grids in the ξ_1 direction and three grids in the ξ_2 direction. Recall that the parametric coordinates range in value from zero to one. The origin for the surface is identified by the arrows indicating the parametric directions. The one direction is uniformly divided into thirds to accommodate the requested number of grid points and the two direction is divided into halves. The parametric coordinates of the grid point indicated by the arrow in Figure 3 are (2/3, 1/2). The simple example shown in Figure 3 illustrates how the parametric coordinates are assigned and implies how a procedure could be developed for extracting grid point ID's from the parametric geometry. The grid point locations are stored in terms of their parametric coordinates. For example, in Figure 2, all grids on edge 3 would have parametric coordinates of $(1, \xi_2)$ and all grid points on surface 6 of the volume in Figure 2 would have parametric coordinates of $(\xi_1, \xi_2, 1)$. Search routines can be employed to located grid points with specific parametric coordinates based on the user input of which edge or surface information is needed. Next the program which makes use of the neutral file to extract grid point ID's will be discussed.

PROGRAM DESCRIPTION

A computer program was written in Fortran 77 which makes use of a neutral file for extracting the grid point ID's. Assume that a finite element model has been written to a neutral file and the user wishes to extract grid point ID's. Referring to Figure 4, the program will first ask for the name of the neutral file and also prompt the user for the name of a file where the extracted information is to be stored. Once the requested information has been entered, the program will respond with a summary of the model. Each line surface and volume is assigned an ID number and upon execution of the program these ID numbers appear in the output to the screen and are also written to the designated file. Thus, the user is informed about the content of the model as to the total number of grid points and the individual totals for the number of lines, surfaces and volumes. The user is asked if more information is wanted on lines, surfaces or volumes. If "none" is selected, the program will stop depending on wether a new model is to be investigated. The point "A" was designed into the program which will be convenient as a return location to request more information. If more information on lines, surfaces or volumes was requested, the program moves to location "B" on Figure 5.

The program flow chart shown in Figure 5 describes the procedure programed to extract grid point ID's from an individual line, surface or volume. Grid point ID's may also be obtained for the entire model. If a surface is selected, the user may also specify any edge on the surface. If a volume is selected, the user may also specify a surface on the volume. The program returns the user to point "A" on Figure 4 to request more information. The grid point ID's that were requested are written to an output file.

EXAMPLE PROBLEM

Consider a rather simple model consisting of Line 31, Surface 12, and Volume 17 as shown in Figure 6. The model does not represent a real object but was designed to illustrate the utility of the grid point extraction program. The ID numbers for the geometric entities would be assigned by the preprocessor. Grid points have been assigned to each of the geometric entities and are also shown in Figure 6. The parametric coordinates are displayed to identify the edges and faces. Upon executing the program the analyst would enter the neutral file name containing the model and any name for an output file. The program will respond with the following summary:

Total No. of Grids: 23
No. of Lines: 1
No. of Surfaces 1
No. of Volumes 1

Suppose the objective is to recover grid point ID's on face 4 of Volume 17, (ie, grid points 18, 19, 22, and 23). These grid points can be identified by simply knowing which volume ID and which face. The face ID requires knowing the orientation of the parametric coordinate system of that particular volume. The preprocessor must provide this information graphically. Once the model summary has been written to the screen and output file, the program will prompt the user for more information on a L,S or V. The user will input "V" and the program will respond with:

Volumes 1
With the following ID's:
17

The next prompt asks for the ID number of the volume, in this case 17.

Volume 17 No. of Grids 8
16 17 18 19 20 21 22 23

The volume face ID is requested and the number 4 is input. Face 4 is located at the arrowhead end of the ξ_2 direction. The program responds with:

Face 4
Volume 17
18 19 22 23

The face request may be repeated. If no other faces are to be investigated, the program returns to the location "A" in Figure 4 which asks the user if any additional lines, surfaces or volumes are to be investigated.

CONCLUDING REMARKS

Most finite element preprocessors can identify a group of grid points, but there is no convenient method for writing this information to a NASTRAN set card. A program has been written to interrogate a neutral file representing a finite element model which over comes this deficiency. The program will extract grid point ID's which are associated with the parent geometry of the model. The program is external to the pre and post processor which provides additional capability for the analyst. The procedure makes use of the parametric description of lines, surfaces and volumes. The program is very useful for adding additional constraint or set cards to an existing model.

A logical extension to the work presented is to add the capability to define lists of grid point ID's that are not wholly contained in a geometric entity. It is often desirable to write constraint and data recovery sets that contain the grid point ID's from portions of several entities. These lists may then be used to write new constraint and data recovery sets without modifying the basic model geometry. Boolean operators may be used to generate the new sets.

REFERENCES

1. Ross, B.A., J.J. Cox, T.N. Hurst, S.E. Hurst, "Rosetta Stone For FE Modeling," Computers In Mechanical Engineering, Vol 3, No. 4, January 1985, pp 43-51.

2. Stanton, E.L., L.M. Crain, T.F. Neu, "A Parametric Cubic Modelling System General Solids of Composite Materials," International Journal For Numerical Methods in Engineering, Vol. 11, 1977, pp 653-670.

3. PDA Engineering, PATRAN USERS MANUAL,Vol 2, Release 1.5 Chapter 28.0, 2975 Redhill Avenue, Costa Mesa, Ca. 92626

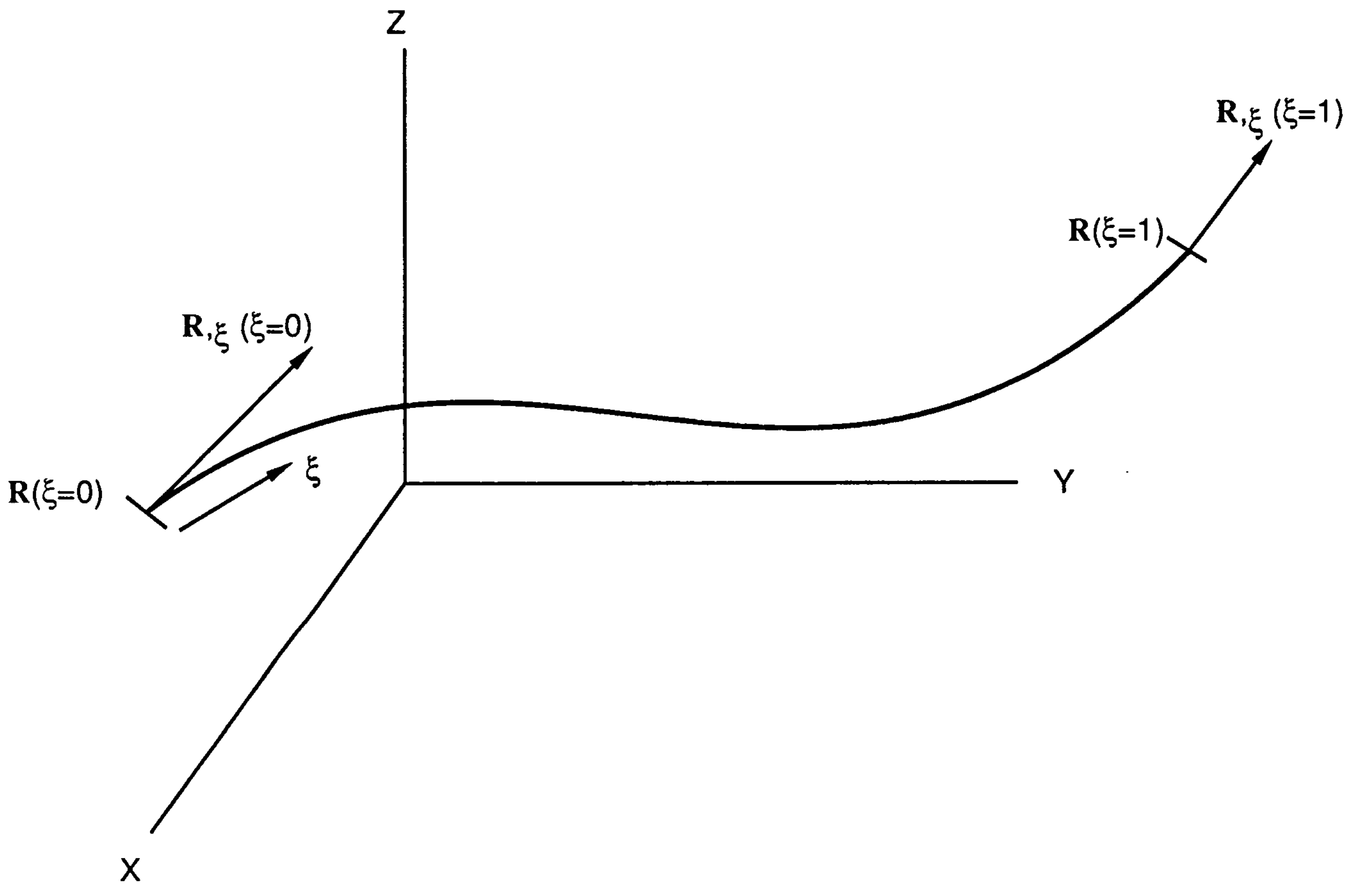

Figure 1 A Parametric Cubic Curve in a Cartesian Coordinate

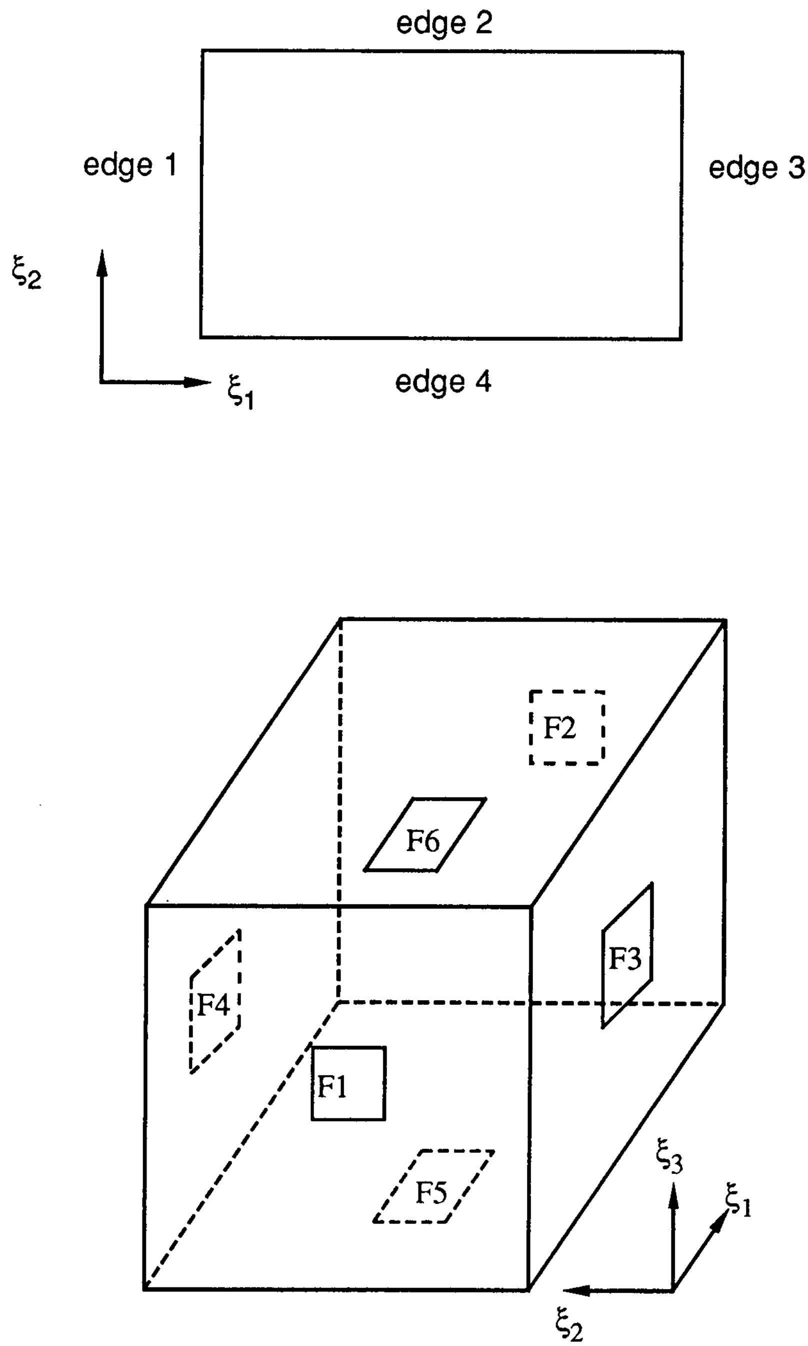

Figure 2. The Parametric Description of A Surface and Volume

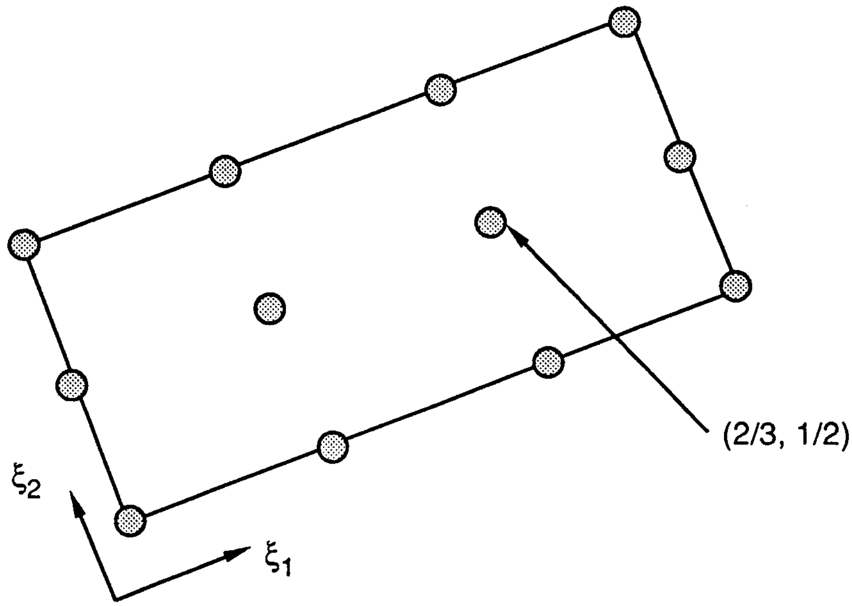

Figure 3. Grid Points on a Surface Located in Parametric Coordinates

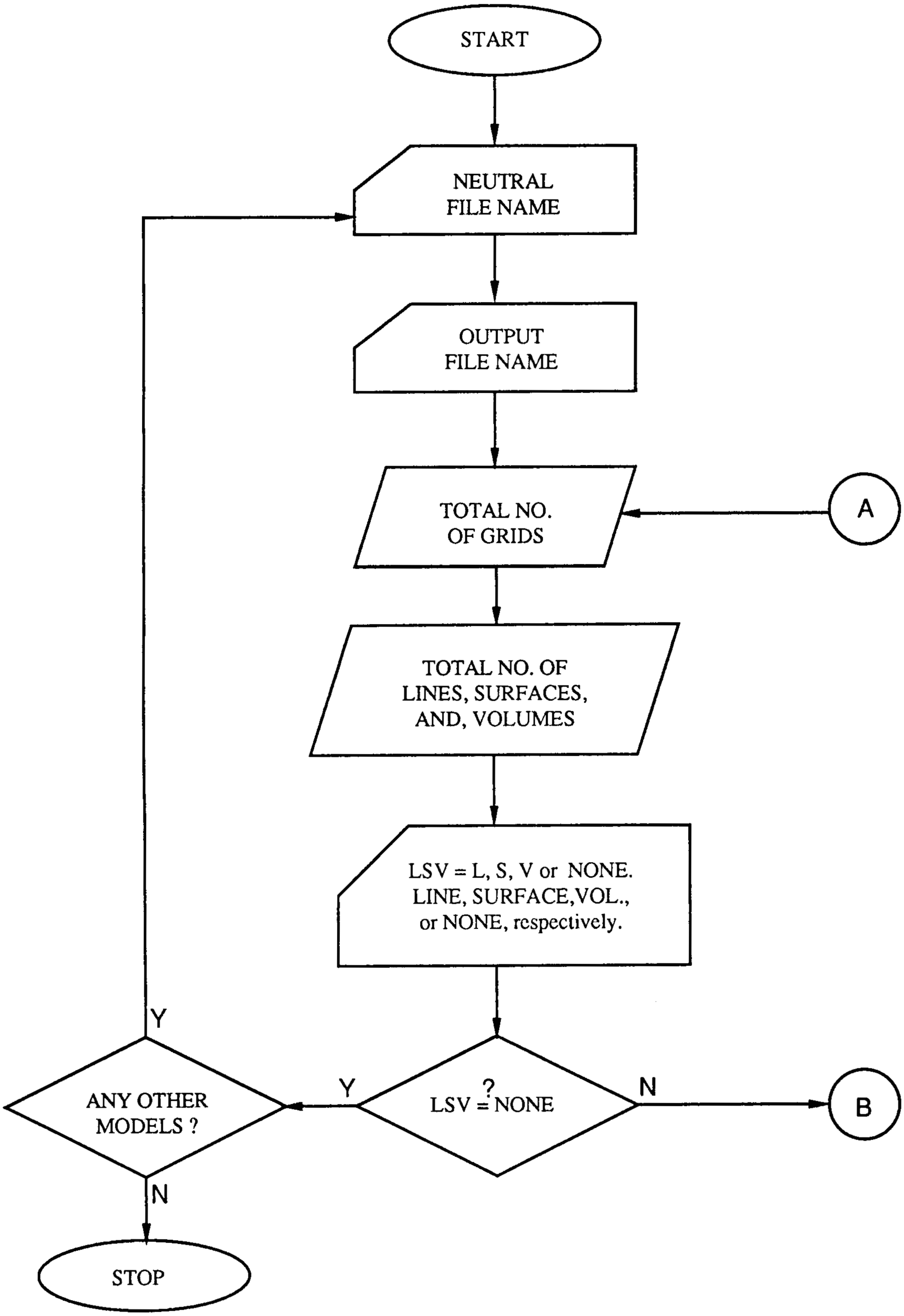

Figure 4 Program Flow Chart for General Model Information

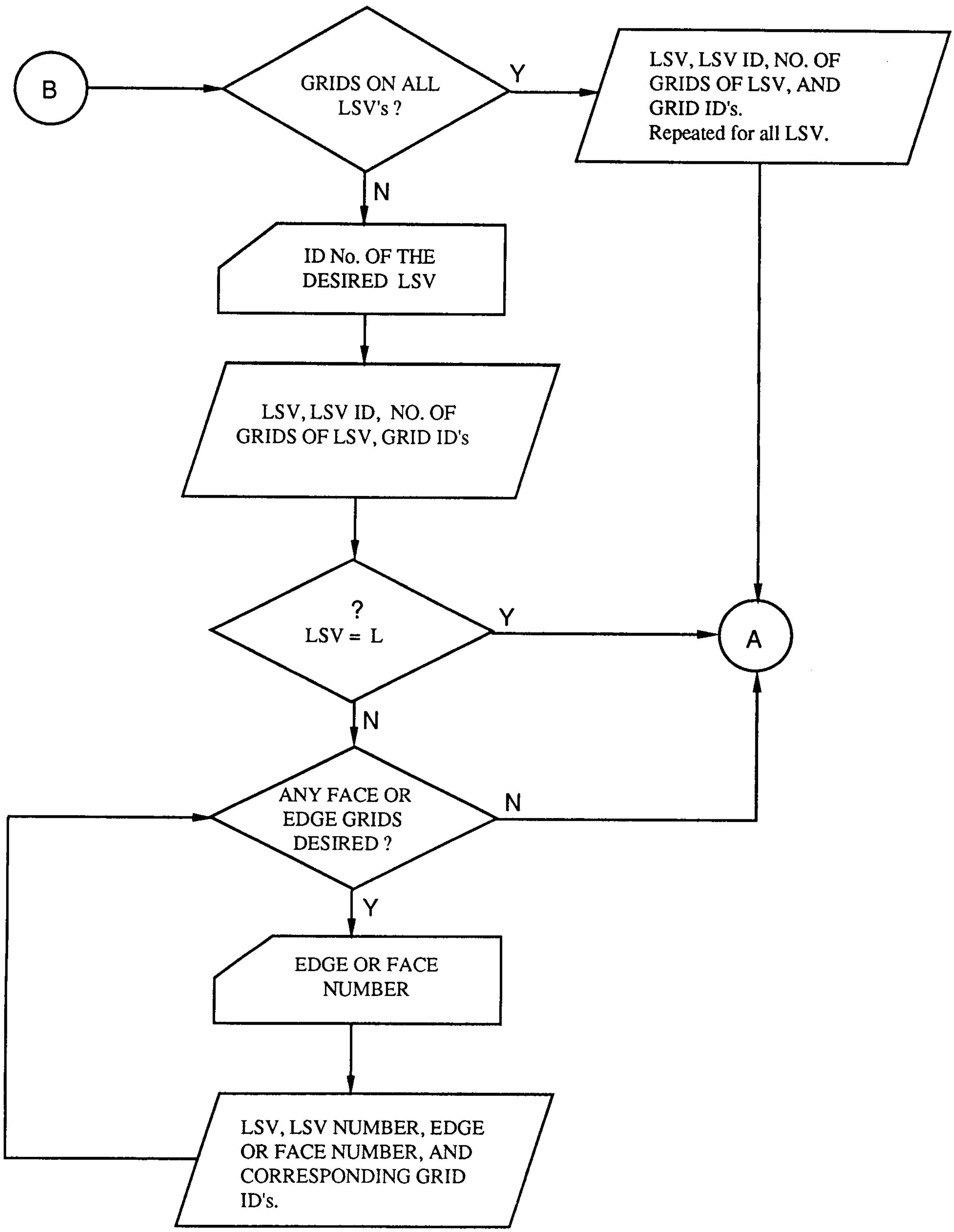

Figure 5 Program Flow Chart for Grid Point ID Extraction

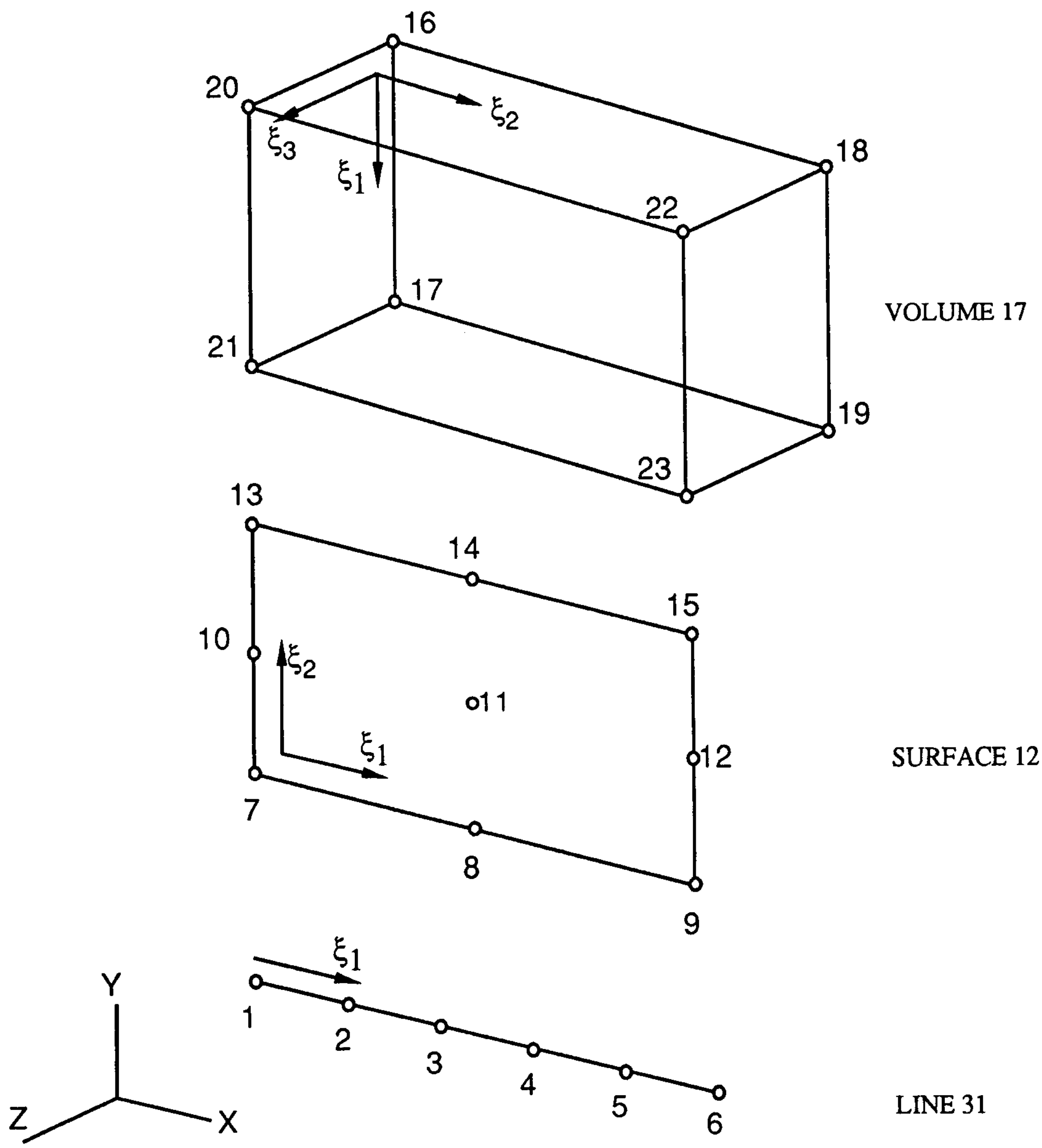

Figure 6 - Line, Surface, Volume with Assigned Grid Points