

1986

# Interactive design and NC machining of free form surfaces (phase II) /

M. Sevket Tarhan  
*Lehigh University*

Follow this and additional works at: <https://preserve.lehigh.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

---

## Recommended Citation

Tarhan, M. Sevket, "Interactive design and NC machining of free form surfaces (phase II) /" (1986). *Theses and Dissertations*. 4728.  
<https://preserve.lehigh.edu/etd/4728>

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact [preserve@lehigh.edu](mailto:preserve@lehigh.edu).

**INTERACTIVE DESIGN AND  
NC MACHINING  
OF FREE FORM SURFACES**

**(PHASE II)**

**BY  
M. SEVKET TARHAN**

A Thesis  
Presented to the Graduate Committee  
of Lehigh University  
in Candidacy for the Degree of  
Master of Science  
in  
Mechanical Engineering

Lehigh University  
1986

# CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering.

12/8/1986  
Date

*Tulga Orroy*  
Professor in Charge

*Charles J. FE*  
Chairman of Department

## ACKNOWLEDGMENTS

I wish to thank Dr. Tulga Ozsoy for all the help he provided for the completion of this thesis. He made himself available at any time with his invaluable technical experience and original ideas.

I would like to acknowledge the support of Bridgeport TEXTRON who provided funds for this project and especially Mr. Marvin Kreithen for his special interest and involvement.

I thank Mr. Greg Loney for his excellent work during the first phase of the project which formed the basis of my development study.

Finally, thanks to my family, especially to my cousin Tijen, for their encouragement.

# Table of Contents

<b>ABSTRACT</b>	<b>1</b>
<b>1. INTRODUCTION</b>	<b>3</b>
1.1 BACKGROUND	3
1.2 PROBLEM STATEMENT	4
1.3 ORGANIZATION OF THESIS	7
<b>2. MATHEMATICS OF CURVES AND SURFACES</b>	<b>8</b>
2.1 INTRODUCTION	8
2.2 MATHEMATICAL DEFINITIONS	9
<b>3. ENHANCEMENTS ON CISPA's ALGORITHMIC WORK AND DATA STRUCTURE</b>	<b>12</b>
3.1 FILE ORGANIZATION AND HANDLING	12
3.2 INTERACTIVE FEATURES	16
3.3 PICTURE MANIPULATION	20
3.4 INTERACTIVE BEGINNING POINT AND PRIMARY CUT DIRECTION SELECTION	23
3.5 CURVE TRANSFORMATIONS	27
3.6 ENHANCED ROUGH TO DEPTH MILLING ALGORITHM	29
3.7 NC TOOL PATH TRIM WITH IMPOSED 2-D BOUNDARY CURVES	32
<b>4. PROGRAM STRUCTURE / PROGRAM OPERATION</b>	<b>45</b>
4.1 HARDWARE	45
4.2 MAJOR MODULES AND PROGRAM OPERATION	45
<b>5. CONCLUSION</b>	<b>52</b>
5.1 SUMMARY	52
5.2 LIMITATIONS	53
5.3 FUTURE STUDY	54
<b>REFERENCES</b>	<b>55</b>
<b>Appendix A. CASE STUDY</b>	<b>57</b>
<b>Appendix B. MENU TREE FOR PROGRAM OPERATION</b>	<b>66</b>
<b>VITA</b>	<b>73</b>

## List of Figures

<b>Figure 3-1:</b>	Flowchart For Model File Concept	15
<b>Figure 3-2:</b>	Flowchart For Continuous Selection	18
<b>Figure 3-3:</b>	Flowchart For Default Name Creation	19
<b>Figure 3-4:</b>	Flowchart For Picture File Regeneration	22
<b>Figure 3-5:</b>	Display of Arrows at Surface Corners for User-Selection of Beginning Point and Primary Cut Direction	24
<b>Figure 3-6:</b>	Patch Structure	25
<b>Figure 3-7:</b>	NC Tool Path with Selected Primary Cut Direction	26
<b>Figure 3-8:</b>	RTD Example	30
<b>Figure 3-9:</b>	RTD Flowchart	31
<b>Figure 3-10:</b>	Digitizing of the Boundary Curve	34
<b>Figure 3-11:</b>	Algorithm For Trimming Operation For "Outside" Case	35
<b>Figure 3-12:</b>	Example for NC Tool Path Trim (outside)	37
<b>Figure 3-13:</b>	Example for NC Tool Path Trim (inside)	38
<b>Figure 3-14:</b>	Flowchart for NC Tool Path Trim	39
<b>Figure 3-15:</b>	Algorithm For Final Pass Calculations	41
<b>Figure 3-16:</b>	Grid Structure for Final Pass Algorithm	42
<b>Figure 3-17:</b>	Flowchart for Final Pass Algorithm	44
<b>Figure 4-1:</b>	Subroutine Call Structure for CISPA	50
<b>Figure A-1:</b>	Primary Curves with Tangent Constraints	59
<b>Figure A-2:</b>	Gencur Surface with Arrows Displayed for Beginning Point and Primary Cut Direction Selection	60
<b>Figure A-3:</b>	Tool Path for Parametric Machining and Imposed 2-D Boundary Curve	61
<b>Figure A-4:</b>	Trimmed Path (outside)	62
<b>Figure A-5:</b>	Trimmed Path (inside)	63
<b>Figure A-6:</b>	Tool Path for Final Pass Around the Boundary	64
<b>Figure A-7:</b>	Tool Path by Roughing to Depth	65
<b>Figure B-1:</b>	Menu Tree For Program Operation	66

## ABSTRACT

Today, CAD/CAM is considered as the key element of most design and manufacturing applications. Interactive NC tool path creation and verification is one of the very early application areas of CAD/CAM technology. It provides an environment in which geometry definition by interactive menus helps the user to create his model in 3-D and graphical feedback insures detection of the input errors before NC machining has started.

During the first phase of the development study by Greg Loney, a program called CISPA was developed. CISPA, Computer Interactive Surfaces Pre-APT, is build upon routines taken from the execution phase of APT 4, a processor designed to handle free form curves and surfaces. In CISPA, a menu-driven front-end with graphical feedback guides the user through curve and free form surface definitions resulting in a mathematical model which may be used to generate NC tool paths for milling of 3D shapes. Improvements were made in the definition and calculation of step size basing the input on geometric attributes. Routines were also added to supplement error handling in the APT complex and increase flexibility in milling surfaces, by rough-to-depth.

Second phase of the development work which formed the basis of this thesis consists of new additions for design and NC machining of free form surfaces, as well as enhancements for geometric and machining data handling, picture manipulations. For the storage of curves and NC tool paths, more flexible model and machining file concepts were implemented, new options for picture manipulation were added, and curve creation by transformation was provided. Existing rough-to-depth algorithm was modified by better definition of the move between part surface and clearance plane. The user is also

provided with the option for definition of beginning point and parametric primary cut direction for generation of NC tool paths on free-form surfaces. Another new feature is NC tool path trim which allows surface machining of a cavity by imposing general side walls created as the projection of a 2-D curve onto the surface.





# Chapter 1

## INTRODUCTION

### 1.1 BACKGROUND

During the last years Computer Aided Design / Computer Aided Manufacturing (CAD/CAM) has proven that it provides very powerful tools which could increase the productivity. Computational Geometry has supplied the mathematical representations of various components of such systems, such as free-form curves and surfaces and with advanced graphics devices, designers have the opportunity of having all the necessary graphical feedback during their use for various design and manufacturing applications. Designing with free form surfaces is essential in many application areas such as found in the aerospace, automobile and shipbuilding industries. Their mathematical representations present difficulties in input definition phase and there is still a great need for new methods to improve this.

With the advent of the computer, computational methods were devised dividing the surface into an assembly of curvilinear, quadrilateral patches, each patch mathematically defined by one formula. In 1963, Ferguson described the patch system in terms of parametric rather than cartesian coordinates. This coordinate definition is now in standard use for many reasons (4). A general theory of surface patches was described by Coons (1964) which showed how four boundary curves can be blended into a smooth patch using general blending functions allowing any order of continuity between patches. The properties of spline curves which are represented by an  $n$ 'th order polynomial were found to be useful in representing the boundary curves. Later B-splines were suggested

by Schoenberg (1964) (2) and applied to curve and surface definitions by Reisenfeld (1967). The introduction of the control polygon by Bezier (1971) (3) has created another dimension in curve and surface definitions. Here, the input is an open or closed polygon of straight lines of which vertices are used in obtaining the definition. The resulting curve is a loose approximation of the input polygon.

The application of the preceding surface formulations on the computer creates easily desired geometry elements such as surface point, normal, curvature etc. The result is a large array of points connected by curves in a 3-D wire frame representation. Computer graphics supplies different options for better visualization of this structure.

## 1.2 PROBLEM STATEMENT

All CAD/CAM systems used in NC machining applications generate a cutter location file in APT or like language. This file is then submitted to the post-processor of the specific NC machine and final control data is prepared. The form of input for geometry and machining parameters and the verification of generated NC tool paths are two distinguishing subjects in the mentioned process. In this aspect, there are currently three distinct levels:

1. A file containing the geometry and machining parameters in a certain format is read by the system. No immediate verification is available.
2. File input is accompanied with graphical verification so that the user can detect the input errors before NC machining has started.
3. User creates the geometry and NC tool paths in an interactive mode. Graphical feedback is made available him or her during the whole process.

The problem initially brought to Lehigh University's CAD Lab by

Bridgeport-Textron Controls Division and it was stated as ' the development of methods for implementing NC milling strategies for 3-D free form curves and surfaces '. This was resulted in a joint project between Bridgeport-Textron and Lehigh whose main goal was: develop a system where by 3-D space curves and surfaces may be created, and later used to define NC tool paths. This system should be inexpensive, yet reliable, easy to use and run on existing hardware contained in Lehigh's CAD Lab, namely a DEC VAX 11/780 mini-computer and raster graphics terminal, such as DEC VS11.

With two years work, a prototype system CISP A (Computer-aided Interactive Surfaces Pre-Apt) with menu-driven input and a sophisticated graphics control had been prepared and submitted to the company as the first version. The description of this work has been presented also to the Mechanical Engineering Dept. of Lehigh University as a master thesis (1). Routines in CISP A might be grouped into four different sections:

1. Interactive front end.
2. Mathematical definitions.
3. NC tool path creation.
4. Graphics control.

The origin of the mathematical definitions part and some of the NC tool path creation modules was another sculptured surface processor named CASPA03 (Computer Aided Sculptured Pre-Apt, Version 3) developed by CAM-I (Computer Aided Manufacturing International, Inc.). CASPA was lacking of an interactive input feature and its NC machining modules were in need of modifications and algorithms for tolerance control. The addition of these together with enhancements on its graphics control eliminated CASPA creating

a new package called CISPA, although CASPA's language input may still be used with a software provided switch.

After a thorough investigation of the earlier version of CISPA, some modifications of the existing modules, as well as additions of new features were decided to be necessary. A development plan has been prepared ordering the suggestions from Bridgeport-Textron development people, the original writer Gregory Loney and the project manager Dr. Tulga Ozsoy. Then work proceeded chronologically as follows:

- Standardization of all picking sessions: pick by name or select by cursor.
- Model file and machining file manipulations: rename, save and retrieve operations.
- Picture manipulation: blanking and unblanking of curves, surfaces, NC tool paths and picture file regeneration.
- Introduction of new data structures: tables for retrieving curves, deleting and blanking of entities, saved file structure.
- Curve transformation: translation, scaling and rotation of existing curves, move or copy options for transformed curves.
- Interactive beginning point and primary parametric cut direction selection for NC tool paths to be created on surfaces.
- NC tool path trim with imposed boundaries generated by projection of 2-D closed curves on to the existing surface.
- Enhanced RTD algorithm: to prevent the problems existed with the earlier work.

The rest of thesis will detail the enhancement work on CISPA encompassed in the above list.

### 1.3 ORGANIZATION OF THESIS

Contained in this thesis is a summary of the earlier version of CISPA and a thorough description of the enhancement work including algorithmic discussion, database organization and program operation.

Chapter 3 introduces a brief explanation and formulation of synthetic curves and surfaces used in CASPA. Algorithmic work is discussed with flow charts in chapter 4. Newly introduced data structures are also explained here. Chapter 5 presents program operation and program structure information such as the hardware required, description of major modules, etc. A summary and conclusion is included in chapter 6. Appendix A contains a case study. The menu flowchart can be found in Appendix B.

# Chapter 2

## MATHEMATICS OF CURVES AND SURFACES

### 2.1 INTRODUCTION

The geometry definition modules of CISPA offers the creation of the following entities :

1. Points

2. Curves

- conic
- cubic spline
- combined curve
- flow curve

3. Sculptured Surfaces

- tabulated cylinder
- surface of revolution
- mesh-type
- mesh-type with flow curves

The design strategy proceeds by defining points, fitting curves to these points, if necessary, combining or setting different flow structures to the simple curves, arranging them in space and fitting a surface to the final curve structure.

Conics are represented with rational parametric cubic equations, for cubic splines, cubic polynomials are used. Sculptured surfaces are made of bi-cubic patches. The following section will explain how to set above mentioned forms.

## 2.2 MATHEMATICAL DEFINITIONS

All of the definitions in this section use projective space rather than the Euclidean space which became a convention in computational geometry because of the convention of manipulating projective elements (9).

Conics, in general, can be represented by the following rational parametric form:

$$(x \ y \ w) = (u^2 \ u \ 1) [A]$$

To find the matrix A, first solve for the unknown k's in the below equations.

$$(k_1 \ 2k_2 \ k_3) \begin{bmatrix} x_0 & y_0 & w_0 \\ x_t & y_t & w_t \\ x_1 & y_1 & w_1 \end{bmatrix} = (x_m \ y_m \ w_m)$$

where  $(x_0 \ y_0 \ w_0)$  and  $(x_1 \ y_1 \ w_1)$  are the endpoints of the conic section and  $(x_t \ y_t \ w_t)$  is the intersection point for the tangents at the end points. The point  $(x_m \ y_m \ w_m)$  is the parametric midpoint. The matrix A is then:

$$A = \begin{bmatrix} k_1 & -2k_2 & k_3 \\ -2k_1 & 2k_2 & 0 \\ k_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 & y_0 & w_0 \\ x_t & y_t & w_t \\ x_1 & y_1 & w_1 \end{bmatrix}$$

Each segment, arc between two consecutive points, of a piecewise cubic spline has its own parametric form which may be written in matrix form as  $r = [U][C][S]$ .

$$r(u) = (1 \ u \ u^2 \ u^3) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & -3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ r(0) \\ r(1) \end{bmatrix}$$

Position and tangent vectors at each endpoint,  $r$  and  $r$ , are needed for the above formulation called a Ferguson curve. If the curve is to be fit through a series of points  $(r_1, r_2, \dots, r_n)$ ,  $n-1$  segment equations of the above form should be established. That implies the determination of all tangent vectors. The

following recurrence relation between the tangents at three successive points has been derived from C2 continuity requirement for the composite curve and may be used to find out all the internal tangents  $(r_2, r_3, \dots, r_{n-1})$ , given  $r_1$  and  $r_n$ .

$$r_{i-1} + 4r_i + r_{i+1} = 3(r_{i+1} - r_{i-1}) \quad , \quad i=2,3,\dots,n-1$$

The type of surface used in CASPA under the APT complex is called a Ferguson surface, because the surface is made up of bi-cubic patches bounded by the segments of Ferguson curves. The parametric form of the bi-cubic patches takes the following matrix form:

$$r(u,v) = F(u) Q F^T(v)$$

where

$$F(u) = [ f_1(u) \ f_2(u) \ f_3(u) \ f_4(u) ]$$

and

$$Q = \begin{bmatrix} r(0,0) & r(0,1) & r_v(0,0) & r_v(0,1) \\ r(1,0) & r(1,1) & r_v(1,0) & r_v(1,1) \\ r_u(0,0) & r_u(0,1) & r_{uv}(0,0) & r_{uv}(0,1) \\ r_u(1,0) & r_u(1,1) & r_{uv}(1,0) & r_{uv}(1,1) \end{bmatrix}$$

The F vector is made up of interpolation functions which supply C1 continuity for the patch.

$$\begin{aligned} f_1(u) &= 1 - 3u^2 + 2u^3 & f_2(u) &= 3u^2 - 2u^3 \\ f_3(u) &= u - 2u^2 + u^3 & f_4(u) &= -u^2 + u^3 \end{aligned}$$

Matrix Q contains geometric data from the patch corners: upper right section contains positional information, upper left and lower right partitions hold tangent information in the u,v directions, and lower left section is filled with parametric cross derivatives called 'twist vectors'.

In certain cases, there is no need for the storage of the whole Q matrix since the parametric patch equation takes a simpler form containing less



elements. Bi-linear surface which may be defined with the following equation is a good example for such cases.

$$r(u,v) = [ r(0,0)(1-u) + r(1,0)u ](1-v) + [ r(0,1)(1-u) + r(1,1)u ]v$$

A proper flag assignment based on the distinction of special subtypes of Q permits less storage and greater computational speed.

# Chapter 3

## ENHANCEMENTS ON CISPA's ALGORITHMIC WORK AND DATA STRUCTURE

### 3.1 FILE ORGANIZATION AND HANDLING

CISPA uses different files for storage and retrieval of different types of data. External files used by the system for geometric, machining and graphical data had the following assigned names in the previous version.

1. CSAVESET.DAT : curve input data from saved set.
2. CLSF.DAT : NC tool paths in binary form.
3. FOR007.DAT : single NC tool path as punched GOTO cards.
4. FOR004.DAT : canonical forms for curves and surfaces.
5. FOR008.DAT : graphics point file ( picture file ).
6. ERRORS.DAT : error messages data file.

As the list implies, the user was not allowed to specify the geometry and machining files. At the beginning of the interactive session, above prenamed files were opened and used during the session instead of opening a scratch file and copying it to the named model or machining file in case that the user wishes to save his work. Also, there is no need for keeping canonical form and graphics files after the program exit. Previous version of CISPA, however, was not considering these facts. A new file organization and handling scheme has been introduced to improve the data management by releasing existing weaknesses.

At the very beginning of the execution phase of the program, the files

FOR004.DAT and FOR008.DAT are opened as 'scratch' files which are automatically deleted by the operating system at the end of the program operation.

For storing curves, a more flexible model file concept has been introduced. After the selection of menu-driven mode, the user is asked to identify the model file. If no name is entered, CISPASAVED.DAT is used as default. If such a file exists, its first records, containing the number of saved curves and the end records of their input data in the file, are read into NSAVE and the array CURVFLOG, respectively. In the file, first record of each saved curve contains the name of the curve. Using record numbers in CURVFLOG, these names are found and read into the array NAMSAV which is later accessed for the selection of curves to be retrieved. As the last step of the initialization, CISPA.SCR is opened as a scratch file for storing input information. During the program operation, input data of every retrieved or newly created curve is written into CISPA.SCR and the appropriate location of the array ICATALOG is loaded with the number of the end record. Since the index of ICATALOG serves as the identifying entity number, surfaces have also an element in ICATALOG reserved, although currently, synthetic curves are the only type of entities which can be saved. Such elements are assigned the same value as the preceding one, because the begin record of the  $n$ th curve is looked at ICATALOG( $N-1$ ) which should contain the end record of preceding curve. A negative number in ICATALOG indicates deleted curve. If the user wishes to save the currently active curves, the current model file is overwritten with the content of the ICATALOG and the input data of undeleted curves in the scratch file. The model file may be renamed any time during the session

allowing the transfer of a group of curves from one model file into a new one. A detailed flowchart of the algorithm is given in figure 3-1.

Applying almost the same procedure described above, a similar file concept for the NC machining section of CISPA has been introduced to provide flexibility in storing and retrieving NC tool path data. After the user entered the machining file name, the contents of this file are copied into another scratch file named CISPETH.SCR, and all of the saved NC tool paths in the file are made active while they are displayed. That means, user can not selectively determine which paths should be retrieved from the file and which not. Other than this, there is no difference between geometric definitions and machining sections of CISPA for 'delete', 'rename' and 'save' operations. Another external file referred in the machining section is the punch file. During the punch file generation, the point data of the specified NC tool path is extracted from CISPETH.SCR and written to a file, opened under the given name, in the form of GOTO/ cards.

Before ending the program operation, active machining and model files are closed with one of the 'delete' or 'keep' options. Two different flags which show the saved status of these files are checked to determine which option applies for each case. As mentioned before, all scratch files are deleted automatically at the exit from the program.

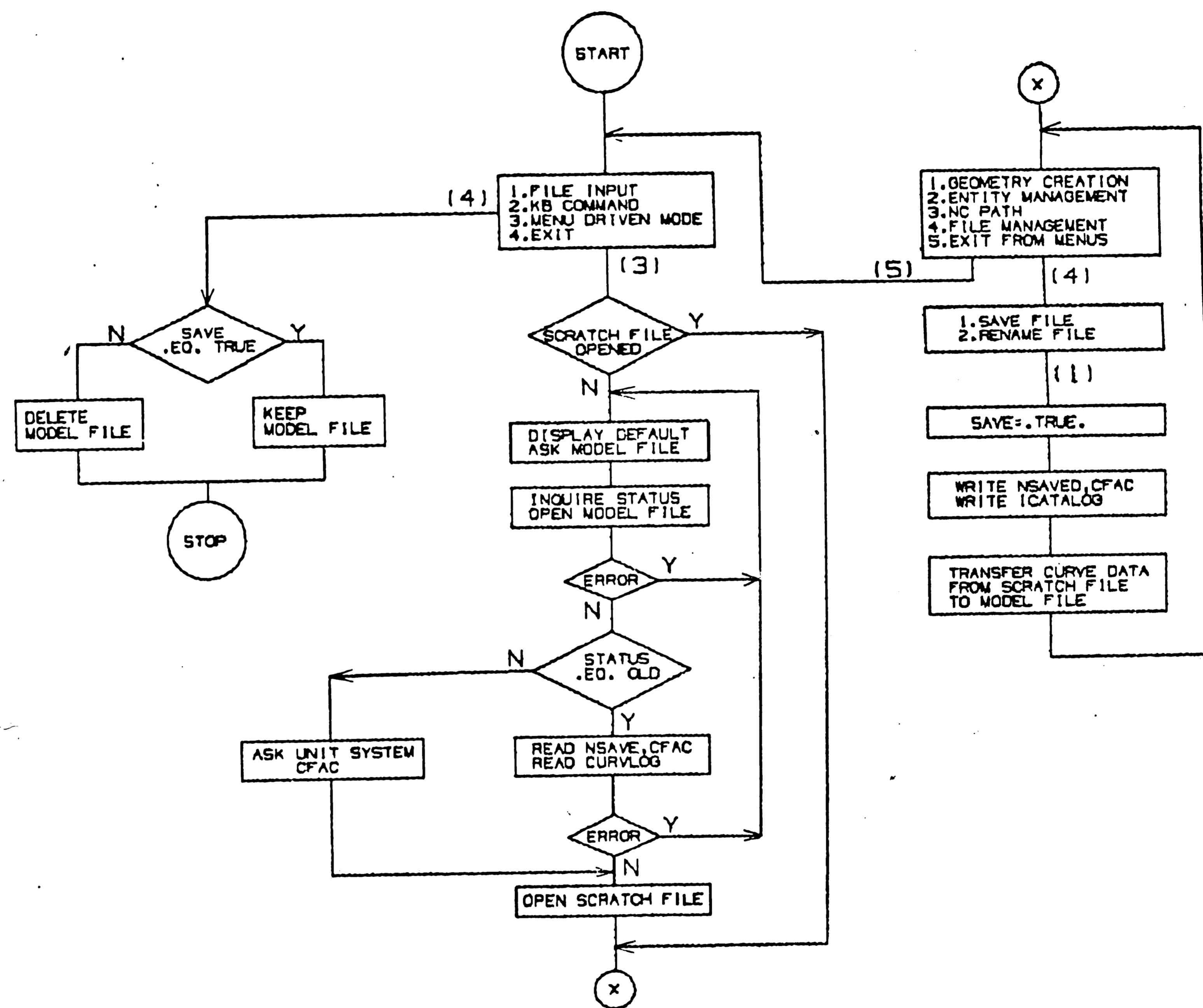


Figure 3-1: Flowchart For Model File Concept

### 3.2 INTERACTIVE FEATURES

As mentioned earlier, CISPA essentially provides a menu-driven front end for a set of APT routines used in CASPA for the creation of curve and surface geometry. Interactive features of its previous version have been explained in 'CISPA Package Manual' (7) in detail. Among these features, interactive entity selection and default name creation for entities were found to be requiring a thorough modification. The explanation of newly implemented methods follow in the next paragraphs.

Throughout CISPA, there are several points where an interactive entity selection is needed. In the previous version, some of these selections can be made by use of cursor or by entering the name of the entity to be picked via keyboard. Standardization of all selection types was considered to be necessary and a unique selection method, offering both 'pick by cursor' and 'select by name' options, was implemented in every program module which employs an interactive entity selection. As first, display of surface and NC tool path names was added to be able to offer 'select by name' option. NC tool path names appear at the first point of the path. Surface names are displayed at the parametric mid-point whose location in the picture file is calculated once and stored in the array IBLANK. If the user prefers 'select by name' option, all active, unblanked entities of the specified type are listed and the user is prompted to enter a name from the list. In case that no name is entered or entered name can't be found in the list, selection session ends with a message saying that no entity is selected. In the machining modules, blanked entities are also included in the list.

'Pick by cursor' is the other option of the entity selection menu. If a

curve is to be picked, the cursor coordinates are compared with all curve points in the picture file projected onto the screen. The curve having the closest projection point to the current position of cursor is selected and the selection point is marked by a circle symbol. For NC tool path and surface picking, only screen locations of displayed entity names are checked versus current cursor location. In blanking and deleting modules, 'pick by cursor' method is applied as continuous selection. Until carriage return has been hit, the user is prompted to pick another entity by depressing the space bar, and the selected entity numbers are entered into the selection list. Any entity picked by mistake may be deleted from the list by hitting the 'R' (Reject) key. Alternate action, in this case 'enter by name' option may be invoked by hitting the '\$' key. When carriage return is hit, requested operation (blanking or deleting) is applied on the entities in the list. The selection session can only be ended by depressing the '!' (Menu display control) key which generally backs up one menu level. Figure 3-2 is the flow diagram of continuous selection algorithm.

In CISP, there are two options available for naming curves and surfaces. The user can either accept the displayed default name by 'CR' hit or type a name of his choice. The default name is a concatenation of a descriptive letter ('C' for curve, 'S' for surface) and a number which is incremented by one with the naming of each entity. The naming procedure necessitates a checking operation to see if the default name or the typed name was previously used. In the previous version of CISP, the checking operation wasn't working properly allowing duplicate names. The flowchart in figure 3-3 explains the new method implemented to avoid any ambiguities. Valid keywords of CASPA, as well as

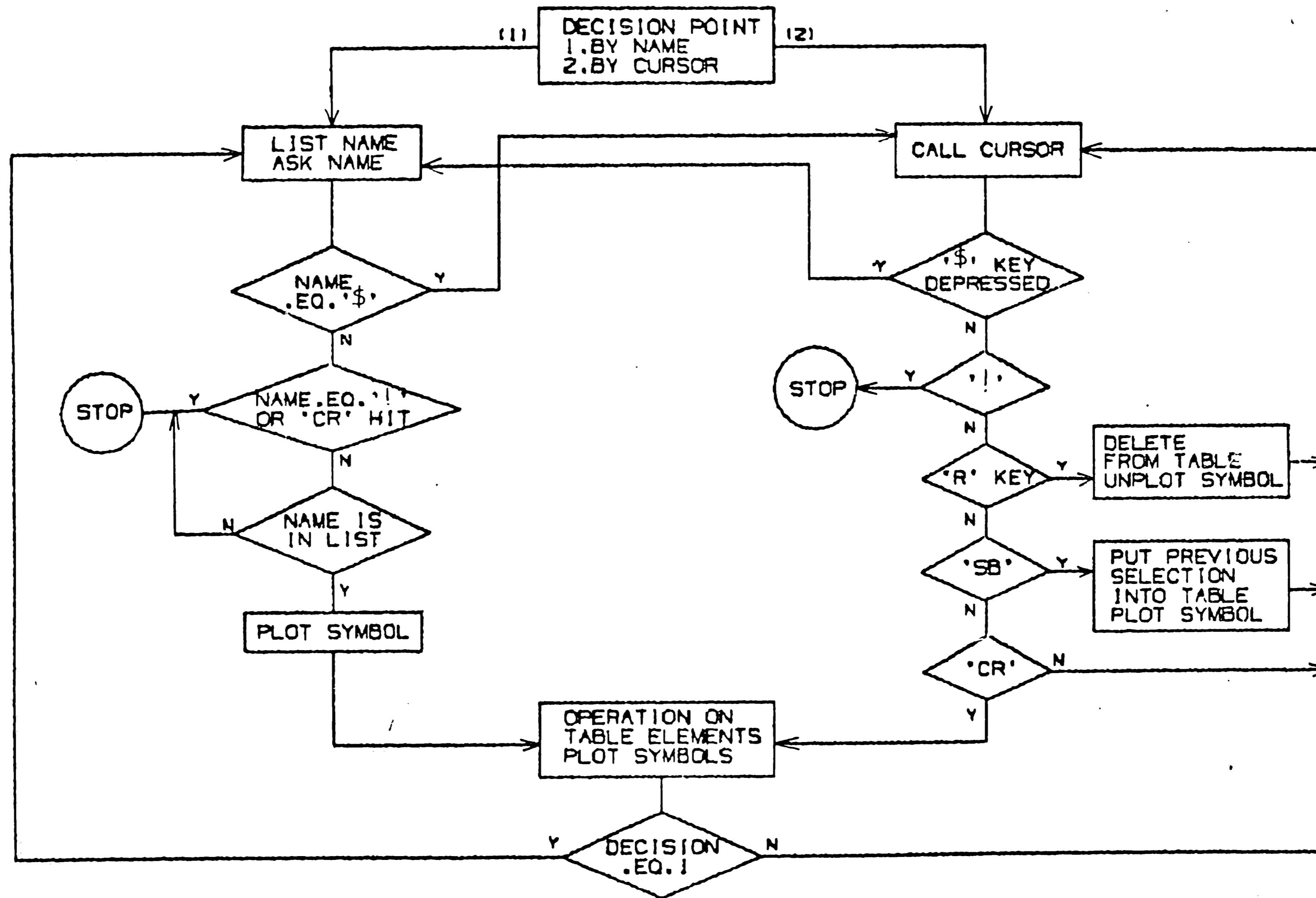


Figure 3-2: Flowchart For Continuous Selection



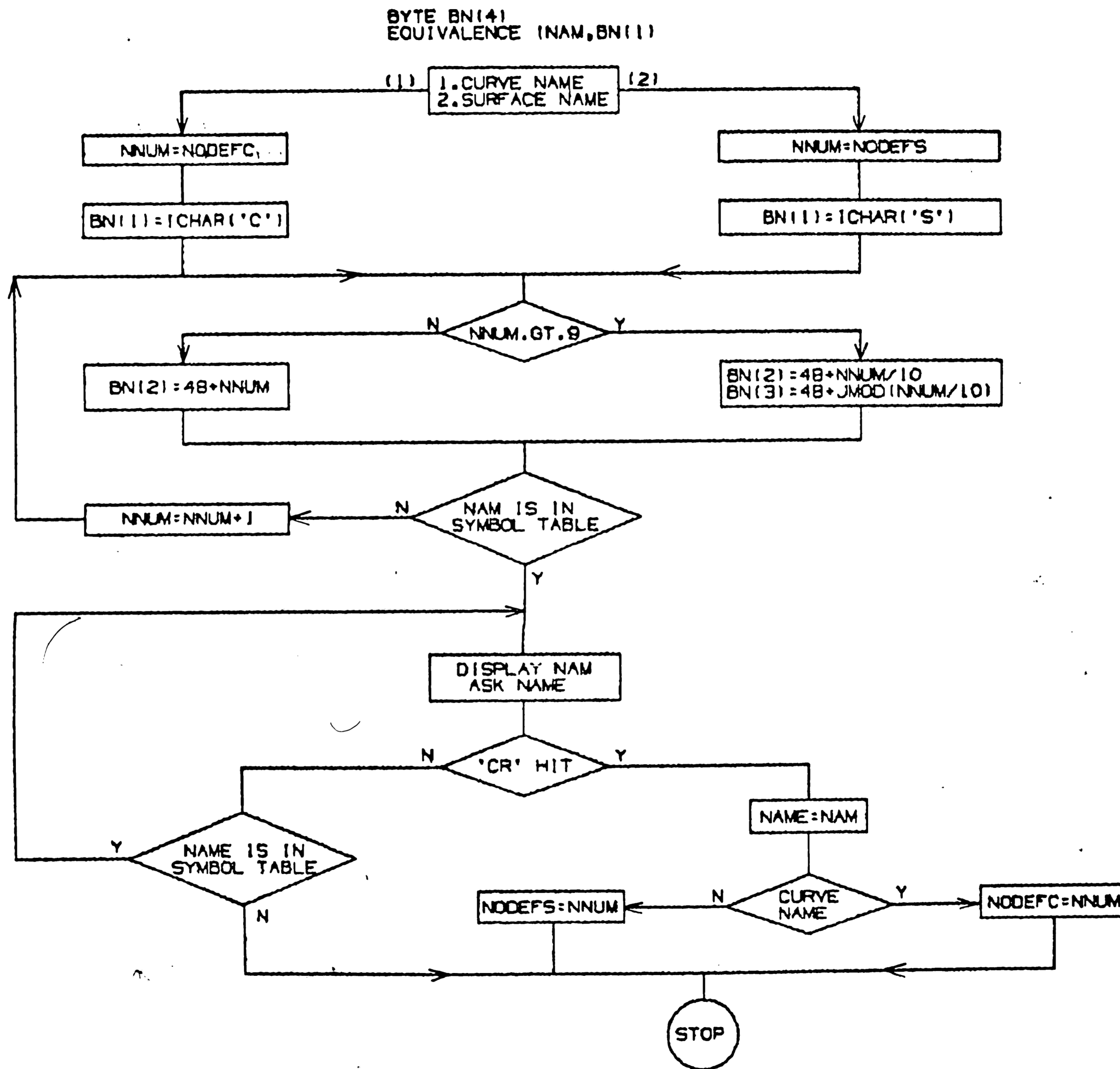


Figure 3-3: Flowchart For Default Name Creation

entity names are stored in a symbol table and a search routine called SWORDS is responsible for checking and entering new names into this table. Same routine is called for checking the duplicates of entered new names either by default or user typed. Since the symbol table has a certain format holding the data in integer form, necessary conversions and initializations have to be done before the search routine is called.

### 3.3 PICTURE MANIPULATION

At each decision node a global menu is activated by pressing the ' G ' key on the keyboard. The global menu of previous version of CISPA offers the options redraw, view manipulation, clear graphics or dialog and hard copy. In the new version, two additional options, blanking / unblanking and regenerate, have been added.

The addition of a blank / unblank option was considered necessary for a better visualization. Quite often, depending on the eye point entities happen to be placed on top of each other. This necessitates blanking operation to get a clear view. Curves, surfaces and NC tool paths to be blanked can be selected with the cursor or keying in their names, as explained in the preceding section. After a successful selection, the names of blanked entities disappear, meaning that they will not be displayed at the next redraw operation. In the unblanking part, the user first gets a list of blanked entities of the specified type. Then, typing names from the list, blanked entities may be displayed again.

Actually, blanking and unblanking are simple flag setting operations. The arrays IBLANK ( for surfaces and curves ) and NCBLANK ( for NC tool paths ) contain blanking flags which are set to 1 at blanking and to 0 at unblanking.

During the redraw operation, these flags are checked before calling the display routine for each entity.

Regenerate function cleans up the graphics point buffer and/or graphics point file from deleted entities. The graphics point file is never used unless the current point buffer (holding 500 points) fills. Then the entire buffer is written to disk and additional incoming points are augmented to the file. Whenever there is a need for accessing to a point which is not in the buffer, appropriate section of the file is copied to the buffer and the index of the point is calculated.

If the size of the graphics point data, NNUM, is less or equal to 500, regenerate routine deals only with the arrays P, ENT and ICON which contain dimensional coordinates, entity codes and display codes respectively. A negative entity code indicates deleted point. Every element of the array ENT is checked one at a time, and data of the undeleted points are transferred into three temporary arrays. After the last point has been checked, arrays P, ENT and ICON are initialized with data in the temporary arrays.

In case that NNUM exceeds 500, above explained procedure is applied to the file FOR008.DAT which is a sequential file holding six-word binary records of the graphics point file data. The last word of each record is reserved for the entity code which determines if the point will be skipped or not. After the file has been rewritten with undeleted point data, pointers for the buffer are reset to zero and regenerate routine returns. Figure 3-4 illustrates the algorithm described in the last two paragraphs.

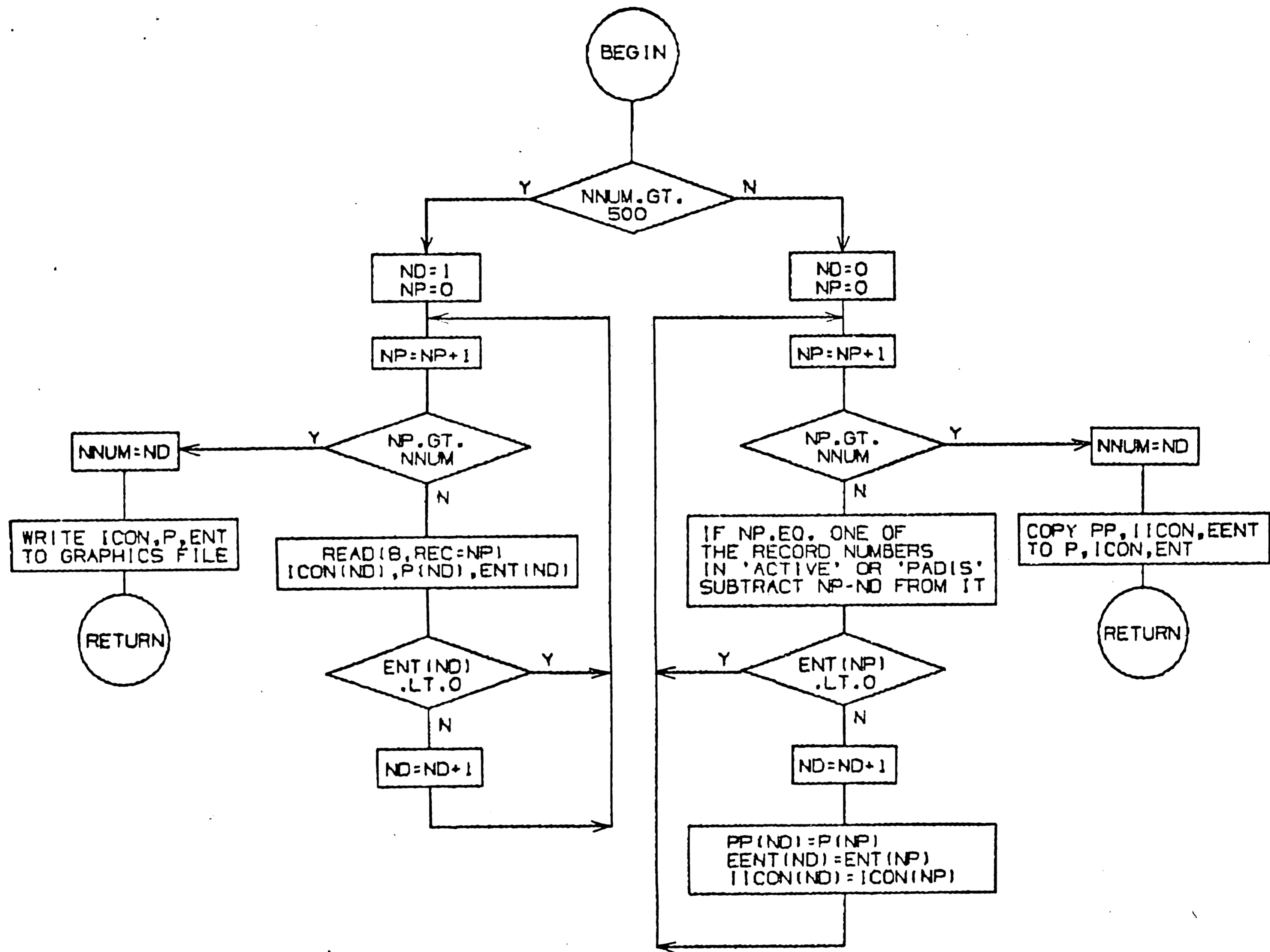


Figure 3-4: Flowchart For Picture File Regeneration

### 3.4 INTERACTIVE BEGINNING POINT AND PRIMARY CUT DIRECTION SELECTION

In the previous version of CISP A, the beginning point and forward stepping direction of cutter paths are set by default. For milling purposes it is sometimes desirable to make this user-defined. The new version prompts the user to pick one of the eight arrows, displayed pairwise at the corners of the surface ( see fig. 3-5 ).

Sculptured surfaces are defined as an assemblage of interconnected parametric surface patches (see fig. 3-6) which are defined as a function of two parameters  $u$  and  $v$ . A topology table, stating which patches lie across each boundary of any patch in the surface, is always part of the data which defines a sculptured surface. In CISP A, cutter paths may start at one of the four corner points of the surface and proceed primarily in  $u$  or  $v$  direction. Before a machining module is called, corner patches are identified with the help of information in the topology table and two arrows are displayed at the appropriate vertex of each corner patch. Selection of one of these arrows gives the starting patch and primary move direction. During the creation of the cutter location data, this information is used to direct the digitizing of the patch definitions. The basic structure used in the machining modules of previous CISP A version which utilize variable step size and scallop height algorithms have not been altered, but some modifications have been done to be able to offer above mentioned directional flexibility. As an example, refer to figure 3-6 and assume that the arrow at the right upper corner pointing downwards has been selected for the generation of a new parametric tool path on the surface. The path will consist of passes created holding  $u$  constant and

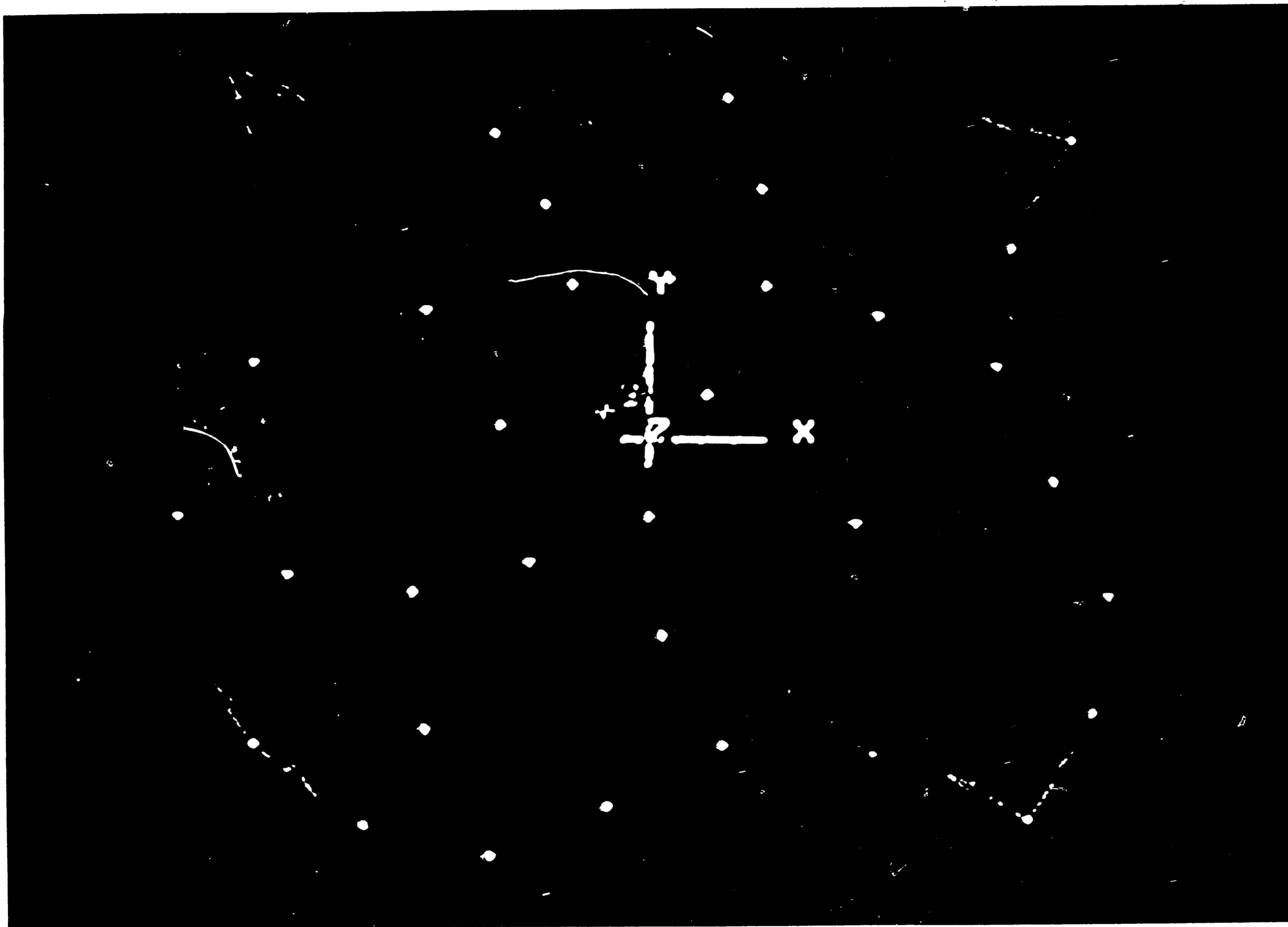


Figure 3-5: Display Of Arrows at Surface Corners for User-Selection of Beginning Point and Primary Cut Direction

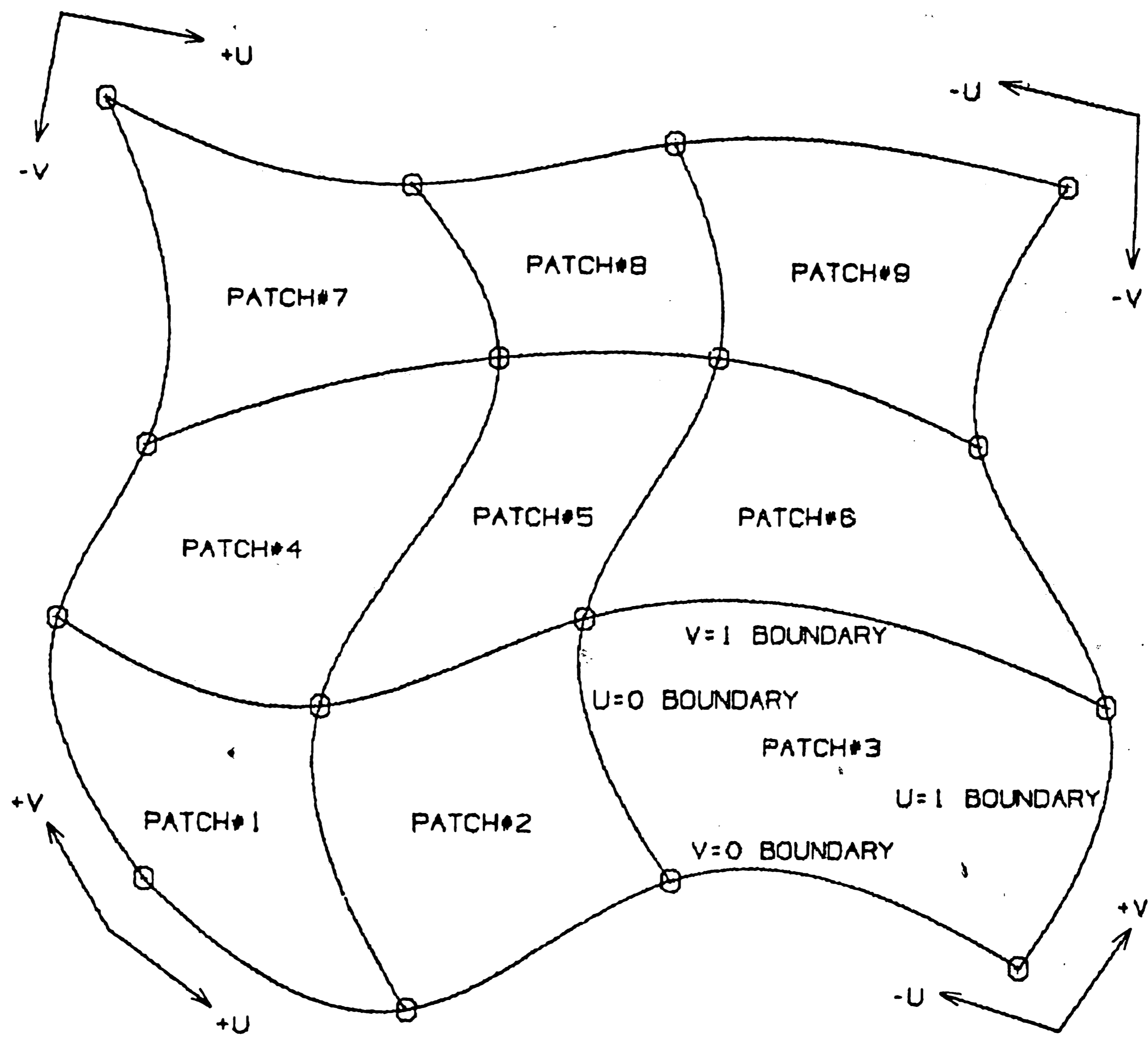


Figure 3-6: Patch Structure

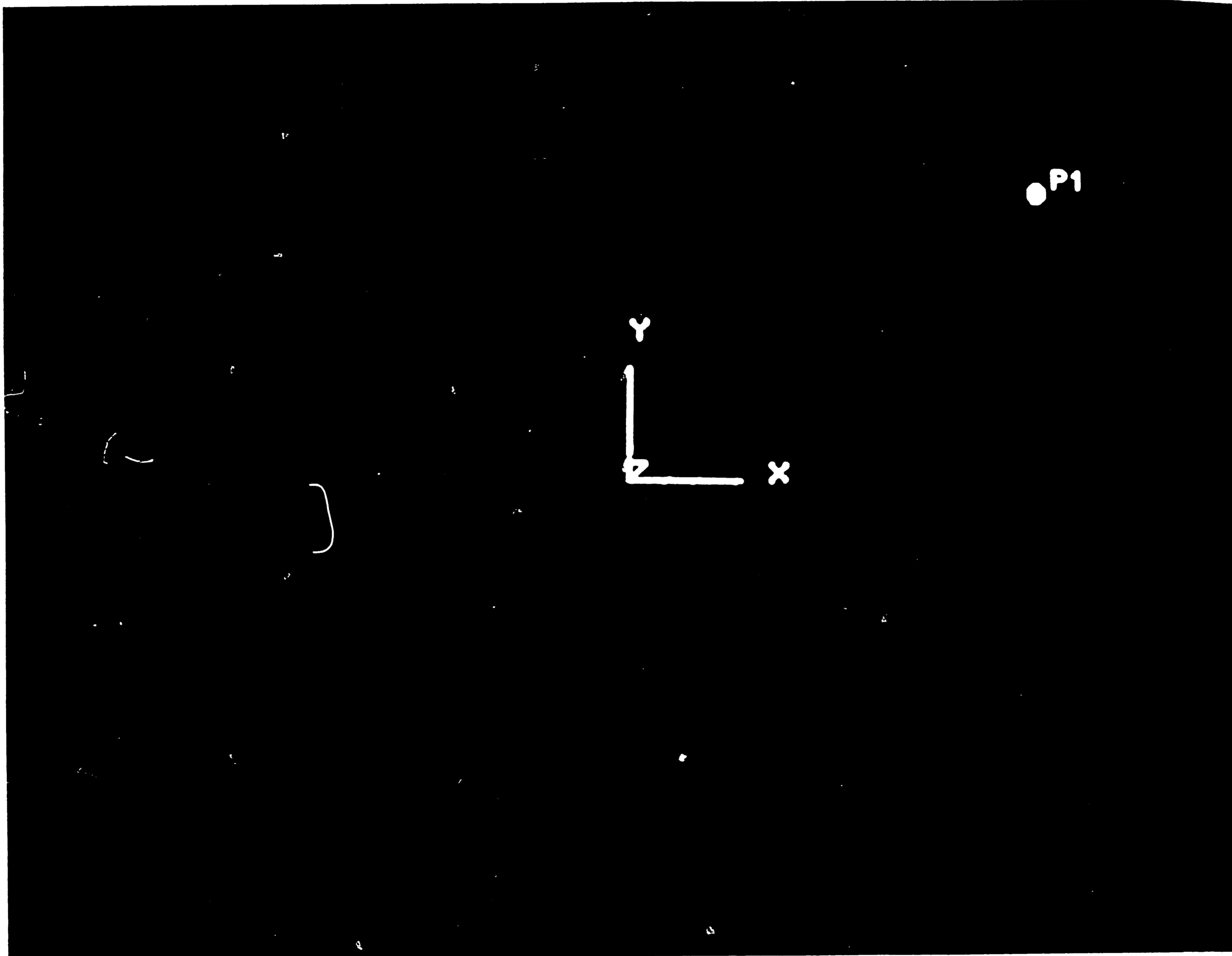


Figure 3-7: NC Tool Path With Selected Primary Cut Direction



varying  $v$  either from 1 to 0 ( move downwards ) or from 0 to 1 ( move upwards ). As moving downwards, '  $v = 0$  ' boundaries are checked to determine next patch to be milled and a zero patch number indicates the end of the pass. For the next pass which will move upwards '  $v = 1$  ' boundaries are the ones to be checked. After milling of three consecutive patches in  $v$ -direction, '  $u = 0$  ' boundary is checked to find out whether there is another three-patch block left. The NC tool path generated for the above case can be found in figure 3-7. Obviously, if another arrow had been selected, digitizing limits and boundaries to be checked would be different.

### 3.5 CURVE TRANSFORMATIONS

Another new capability added to CISPA is curve transformations. Currently, linear translation, overall scaling with respect to an arbitrary point and rotation about an arbitrary axis are the implemented types. All of these three types of transformations use the same user-friendly input phase where three different options for point entry for transformation point are offered. As a new curve is being generated by transforming another one, the user may delete or keep the original curve by choosing one of the ' move ' and ' copy ' options. At the end, the user is also asked if he wishes to continue the transformations with the new transformed curve or to select another one. Same features were also added to ' modify curve ' module.

Input data of the curve to be transformed, is copied from the scratch file to the work array TEM. For 'continue' case this step is skipped, since TEM is updated with the input data of the transformed curve. According to the transformation type, the necessary changes on the stored input data are made and the array TEM is transferred to the corresponding APT modules for final

process. In translation and scaling, constraint vector sections of the input data are not changed. For translating a curve, entered translation vector is added to the junction point coordinates. If overall scaling has been selected, vectors connecting the scaling origin to the junction points are multiplied by the scaling factor and the results are added to the point coordinates. The following formula is applied to every junction point for overall scaling.

$$P = P + sf ( P - SO )$$

where

P : junction point  
 SO : scaling origin  
 sf : scaling factor

For rotation about an arbitrary axis, the user enters two points on the axis and the desired rotation angle. To rotate the junction points, first, coordinate system is translated to one of the given axis-points by subtracting axis point coordinates from those of junction points. Multiplication with rotational matrix given below and translation back to the original coordinate system are the next steps. Mathematical formulation of the explained transformation procedure follows.

$$P = ( P - X_1 ) [R] + X_1$$

where

P : junction point coordinates  
 $X_1$  : first point on axis  
 $X_2$  : second point on axis

and

R : rotational matrix

$$\begin{bmatrix} n_1^2 + (1 - n_1^2)q_1 & n_1n_2(1 - q_1) + n_3q_2 & n_1n_3(1 - q_1) - n_2q_2 & 0 \\ n_1n_2(1 - q_1) - n_3q_2 & n_2^2 + (1 - n_2^2)q_1 & n_2n_3(1 - q_1) + n_1q_2 & 0 \\ n_1n_3(1 - q_1) + n_2q_2 & n_2n_3(1 - q_1) - n_1q_2 & n_3^2 + (1 - n_3^2)q_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$Q \quad \therefore \text{rotation angle (counterclockwise), } q_1 : \cos(Q) \text{ , } q_2 : \sin(Q)$$
$$n = (n_1, n_2, n_3) = X_2 - X_1$$

In the case of constraint vectors, vector components are multiplied by rotation matrix.

### 3.6 ENHANCED ROUGH TO DEPTH MILLING ALGORITHM

A NC tool path generated by a rough to depth allows the milling of a surface in incremental layers (see fig 3-8). The tool path in each of these layers is defined by projecting a previously defined path on the surface onto the bottom plane of that layer and finding intersection points between the passes and the top plane where the cutter may leave or enter the layer. These intersection points are determined by searching a point ahead to indicate whether the path is passing through one of the boundary planes and then applying line-plane intersection algorithm. Beginning from the top surface of the blank, the cutter removes material in layers until the part surface is reached. Then the cutter rides up on the surface as long as the path remains inside the currently machined layer. If an exit point on the upper boundary plane is detected, it moves to a part clearance plane, makes a rapid move on this plane to a location above the next intersection point which will be an entry location. In the case of intersection with lower plane, the portion of the original path between the current and the next intersection point is projected to this plane by simply changing z-coordinates of cutter location points. After the machining of each layer, cutter lifts up to the part clearance plane and before plunging into the next layer, it moves to a location on the clearance plane above the first machining point in that layer. As the rough to depth module of

5: GLOBAL MENU

INPUT SUBSEQUENT Z STEPS (0.1E+01): 0.5

ROUGH TO DEPTH PARAMETERS

- 1 REFERENCE PLANE
- 2 CLEARANCE PLANE
- 3 PART CLEAR PLANE
- 4 FIRST DELTA Z
- 5 SUBSANT DELTA Z
- 6 PROCESS RTD

6: GLOBAL MENU

GENERATING RTD

ACCEPT "CR" OR

REJECT, "R" THE

NO PATH

- 1 MACHINING PAR.
- 2 GENERATE PATH
- 3 DELETE PATH
- 4 FILE TERMINATE

3: GLOBAL MENU

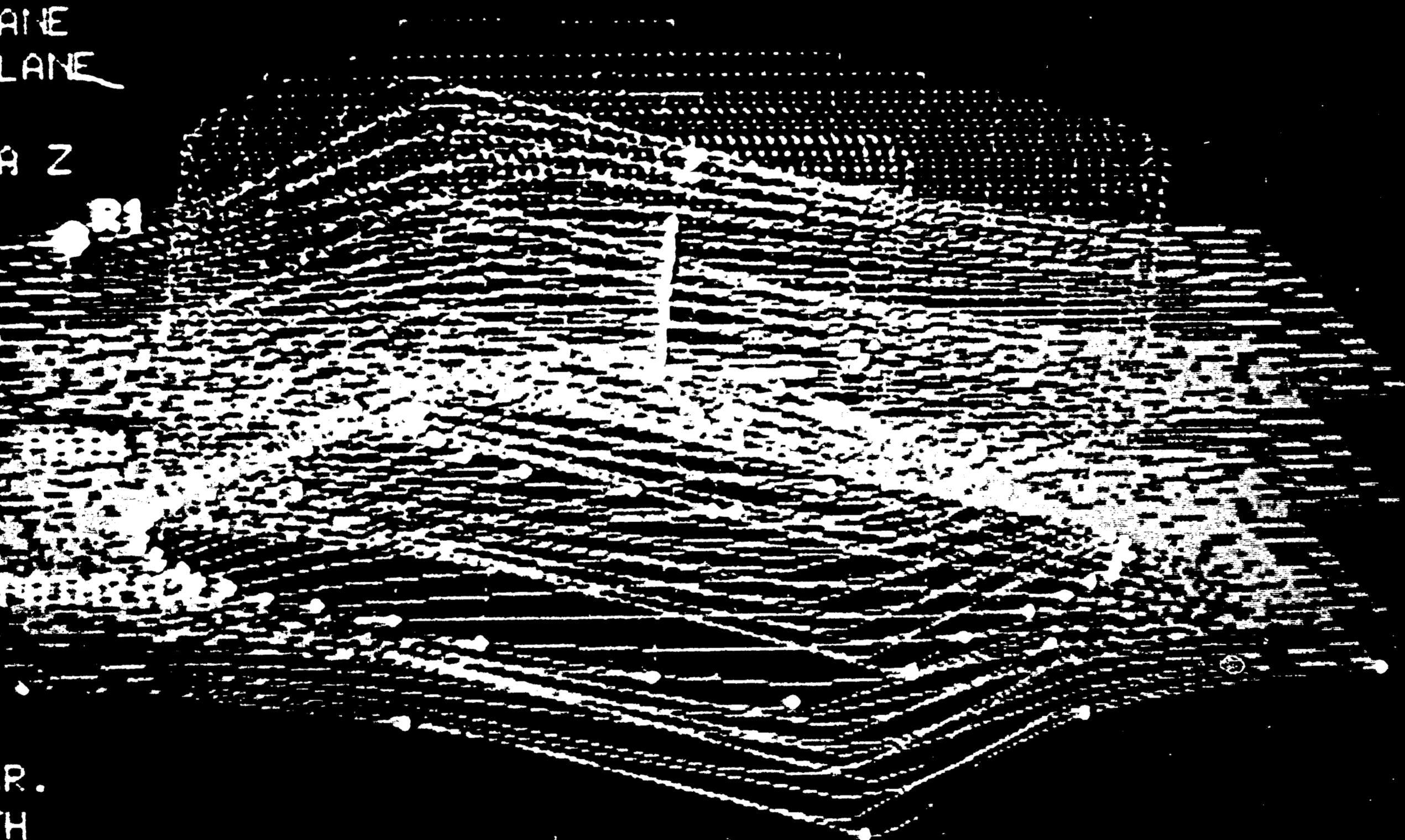


Figure 3-8: RTD Example

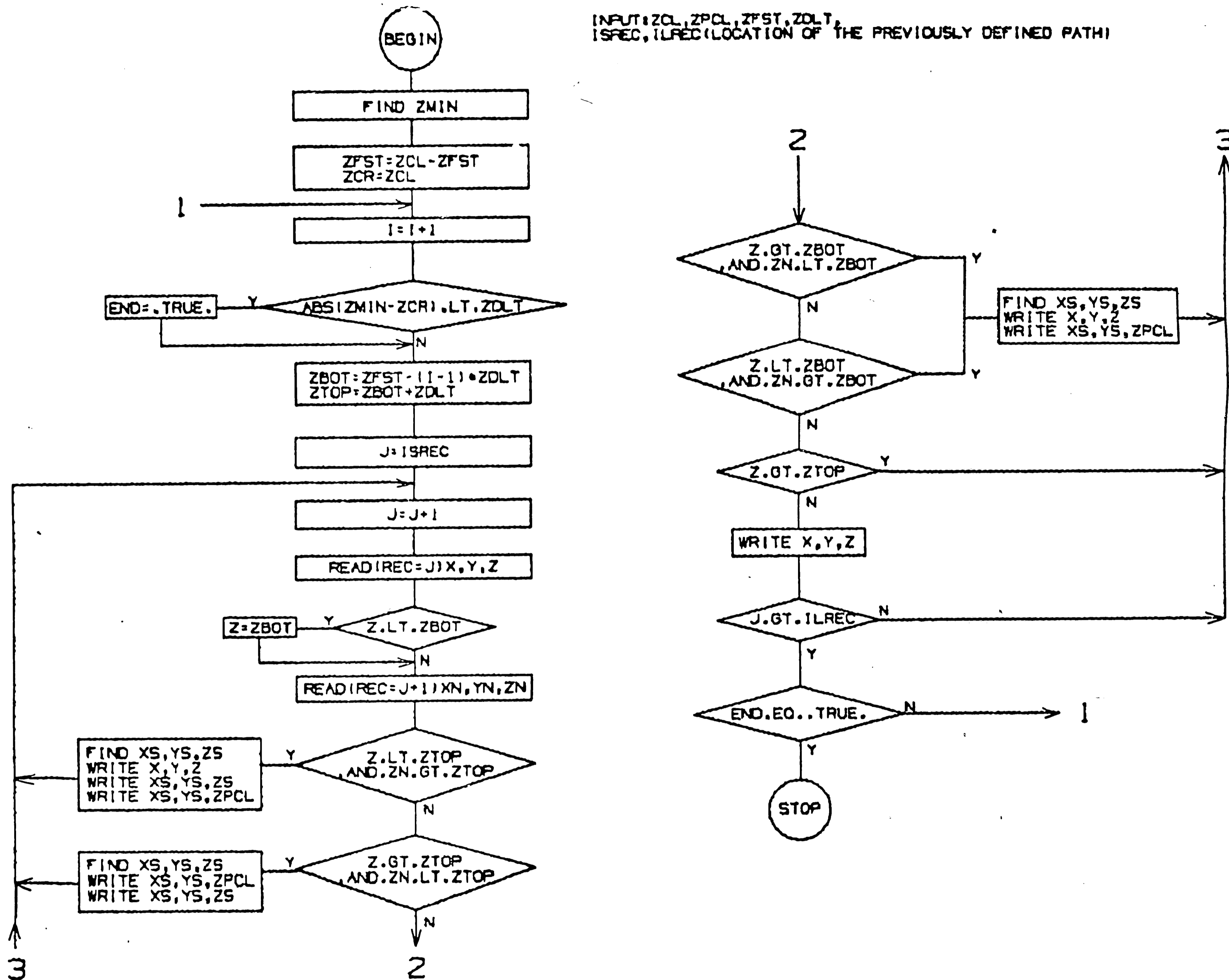


Figure 3-9: RTD Flowchart

the previous version of CISPAs was tested, some errors have been encountered concerning tool motion between the part clearance plane and the next entry point on the current layer. It was decided to create a new program module for RTD machining rather than modifying the existing code. Figure 3-9 demonstrates applied algorithm in detail.

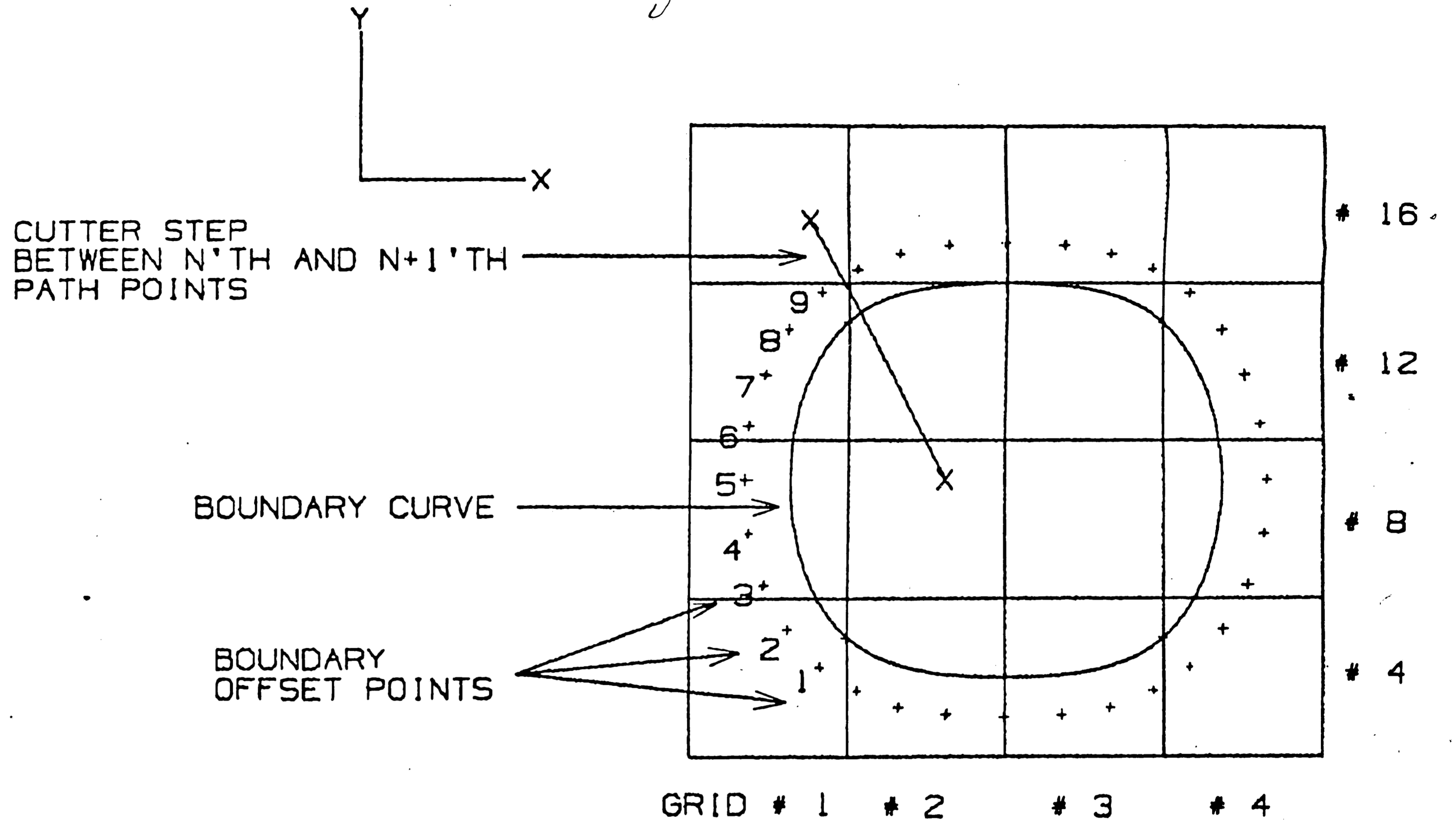
### **3.7 NC TOOL PATH TRIM WITH IMPOSED 2-D BOUNDARY CURVES**

The parametric machining modules of CISPAs use previously defined surface definitions to create cutter location data. Resulting NC tool paths have the natural boundaries of the sculptured surfaces as their own boundaries. In some applications, however, the user may wish to replace the natural boundary by another closed surface curve. Surface trimming where the definitions of the original surface (host surface) and the general boundary defined on it are used to create the mathematical formulation of the trimmed surface (12) would be a solution to the problem. In this case, NC tool path is created on the trimmed surface by taking account of the new boundaries. During the development work on CISPAs a different approach was taken and its implementation was justified by various test cases with satisfactory results.

In this approach, the main idea is that the NC tool path, generated on the original surface, is trimmed rather than the surface itself. That means, the user must first create a NC tool path with the natural boundaries of the surface and then submit it together with a closed boundary curve to the trimming module. The projection of the 2-D boundary curve in the z-direction onto the surface is considered as the new boundary. Inside or outside part of the original tool path to be trimmed is defined as the new tool path.

Algorithmically this approach requires finding the intersections between the original tool path and the given boundary curve. Each NC tool path consists of line segments which connect consecutive cutter location points and represent steps in the milling operation. The problem is further reduced down to finding intersection between two line segments by representing the given boundary curve as a combination of line segments. The offset (in or out) for the boundary curve is calculated according to chordal deviation algorithm (1) for a given tolerance, as if the profile machining is used for defining a single pass around the curve (see fig. 3-10). Since the line segments connecting these points represent the boundary with applied tolerance, intersection points can be determined by checking whether the projection of a step to the x-y plane is crossing the projection of a boundary segment or not.

Before the search for intersection points starts, some arrangements are made to put the data in a tabular form to avoid unnecessary checking and thus increase the response time. The minimum and maximum x-y values of the point data, representing the boundary curve, define a box in which intersections can occur. This min-max box is further divided into square grids of user entered size. Indices of the first and last boundary points in each grid are registered into the corresponding location of the two dimensional array ICAT. A zero value in this array indicates an empty grid. Fig 3-10 demonstrates an example case on how ICAT is formed. Next the trimming operation is started by checking sequentially the x-y coordinates of the cutter locations listed in the given NC tool path to be trimmed. The algorithm used for this trimming operation for 'outside' case is given in figure 3-11.



ARRAY ICAT :

GRIDS :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FIRST POINT	1	29	26	24	3	0	0	21	6	0	0	17	0	10	14	0
LAST POINT	2	32	28	25	5	0	0	23	9	0	0	20	0	13	16	0

Figure 3-10: Digitizing Of The Boundary Curve



```

for each path segment do
  if ( path segment is outside of min-max box ) then
    move to the end-point
  else
    find grid N1 of beginning-point
    find grid N2 of end-point
    for each grid in rectangle of N1,N2 do
      for I = ICAT(N,1) to ICAT(N,2) do
        intersect ( path segment, I )
      enddo
    enddo
  endif
enddo

procedure intersect ( path segment, I )
  if ( path segment crosses  $B_{1,1} - B_1$  ) then
    if ( beginning-point is outside ) then
      move to the intersection point
      lift up to the clearance plane
    else
      move in the clearance plane above the intersection point
      plunge to intersection point
      move to the endpoint
    endif
  endif
endif

```

**Figure 3-11:** Algorithm For Trimming Operation For "Outside" Case

Each cutter location point is checked against if it is inside the min-max box. If then, its grid number is calculated and the preceding cutter location point is assigned to the closest grid. These two grids define a rectangle in which the end points of current cutter step lies. Line segments which connect consecutive boundary points inside this rectangle, are checked against for a possible intersection with the line segment defined by the current step on the path. In the example case in fig 3-10, end-point for a cutter step is in grid 6 and beginning-point is in grid 13. So, boundary points 3 through 13 which lie in grids 5,6,9,10,13,14 form the portion of the given boundary to be checked against for a possible intersection with the current cutter step. After the

intersection module returns the coordinates of the entry point to the boundary, search for the next intersection point starts which will be an exit location from the boundary. Same procedure is repeated until the last step on the path is checked. If the user has selected the 'outside' option, points of the original tool path which fall inside the boundary curve are skipped. At the intersections where the original path enters the bounded region, tool moves up to the user-defined clearance plane and makes a rapid move to a location on the clearance plane above the next intersection where the exit from the region occurs (see fig. 3-12). In the case of 'inside' option, just the opposite is accomplished. The trimmed tool path starts at the first intersection point where the original path crosses the boundary curve. At the exit the tool moves to the clearance plane and doesn't plunge before the next intersection location has been detected (see fig. 3-13). A detailed flow diagram for the trimming operation can be found in figure 3-14.

For the application of an intersection algorithm, both line segments to be checked for crossing are defined in parametric form such that each parameter varies from 0 to 1 between beginning and end points. If a parameter value of the intersection point for a given line is not between 0 and 1, intersection routine returns with no-crossing flag set.

parametric line equation for a path segment (cutter step):

$$P = P_{n-1} + t_1(P_n - P_{n-1})$$

parametric line equation for a boundary segment :

$$B = B_{k-1} + t_2(B_k - B_{k-1})$$

intersection point :  $P = B$  ---

$$t_1(p_{n,x} - p_{n-1,x}) + t_2(b_{k-1,x} - b_{k,x}) = b_{k-1,x} - p_{n-1,x}$$

MINIMUM CLEARANCE HEIGHT: (.200E+01)1

TRIMMING PARAMETERS

INSIDE OF OUTSIDE

TOOL SIZE

CLEARANCE PLANE

TO STEP FOR BOUNDARY PASS

G: GLOBAL MENU

PATH TRIM

.GLOBAL PARAMETERS

.GENERATE TRIMMED PATH

G: GLOBAL MENU

OPERATING TRIMMED PATH

ACCEPT "CP" OR REJECT "R" THE PATH?(CR)

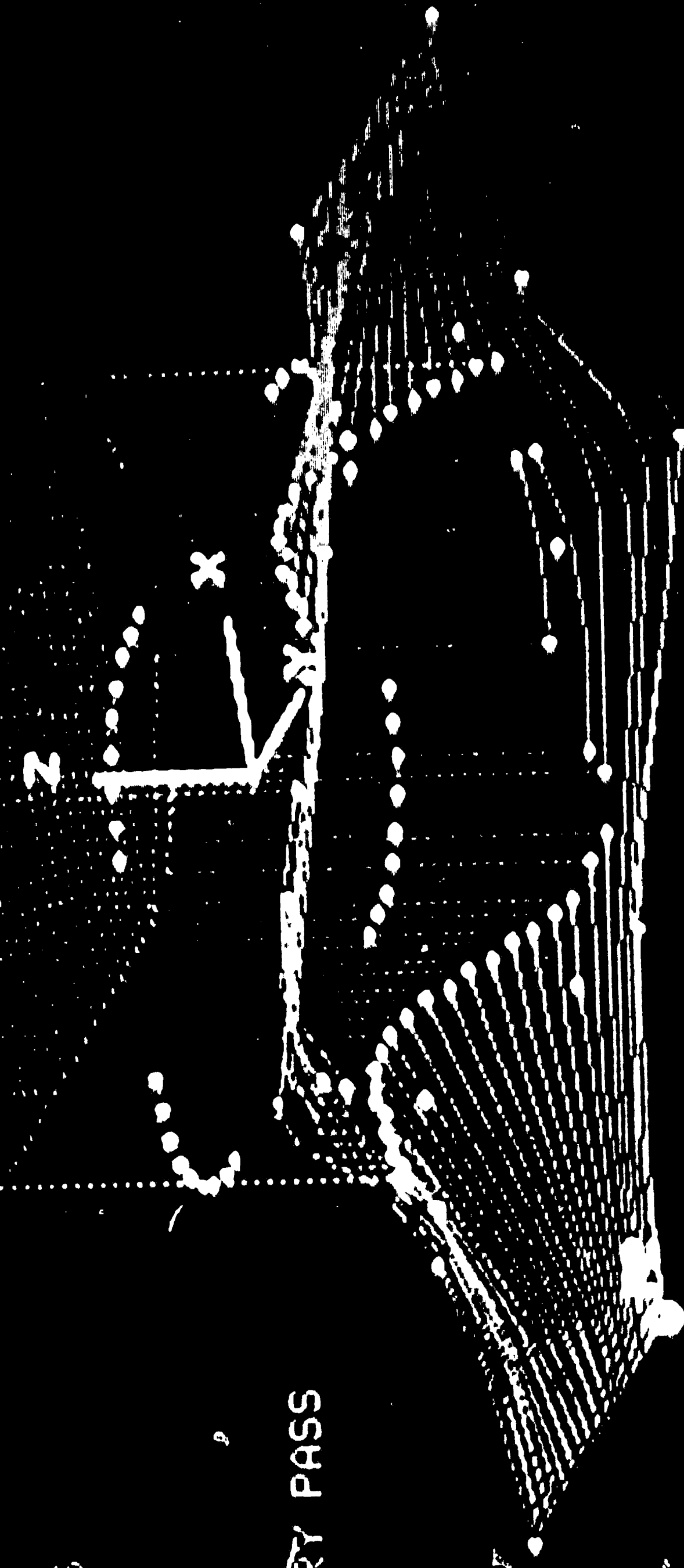


Figure 3-12: Example For NC Tool Path Trim (outside)

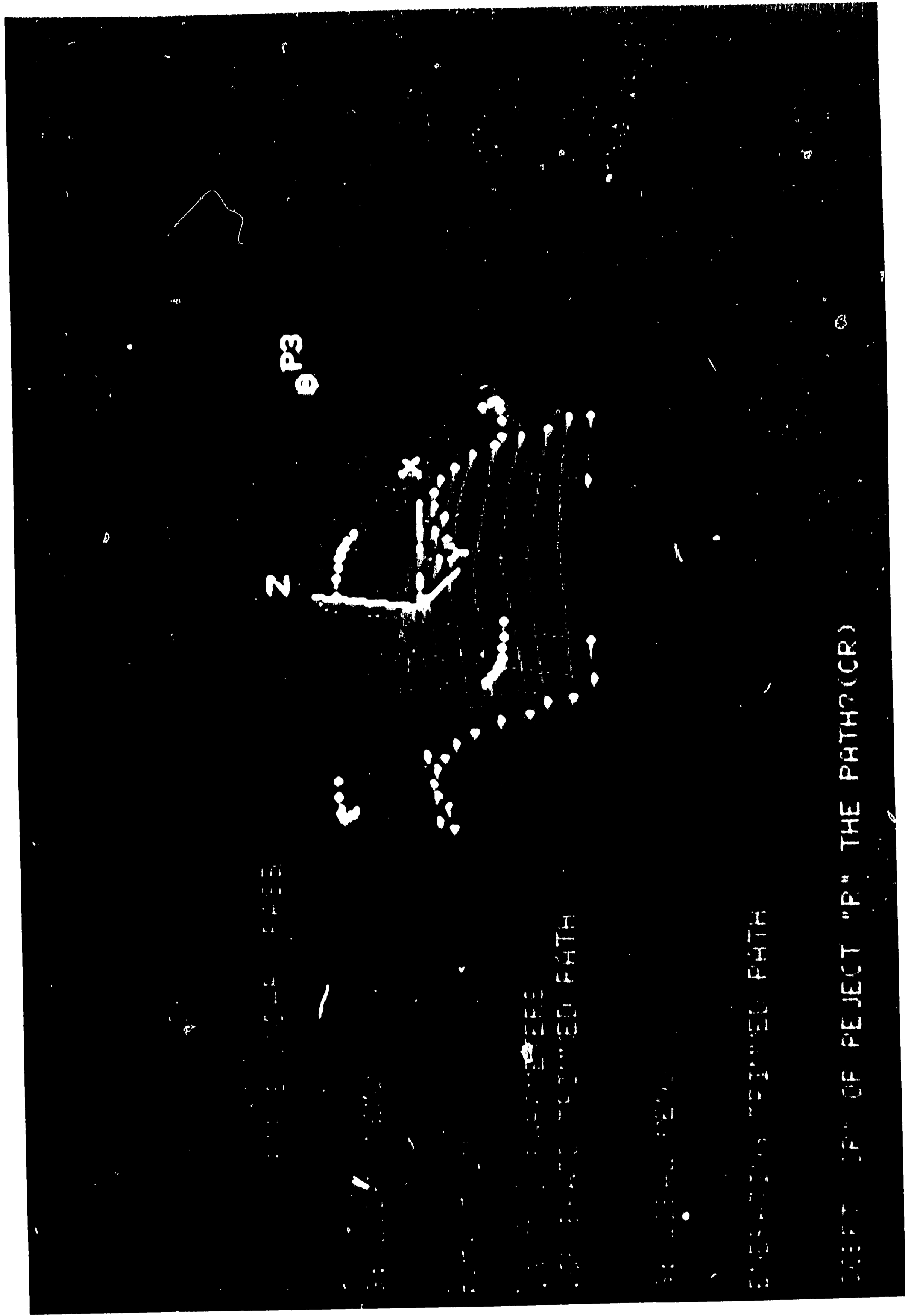


Figure 3-13: Example For NC Tool Path Trim (inside)

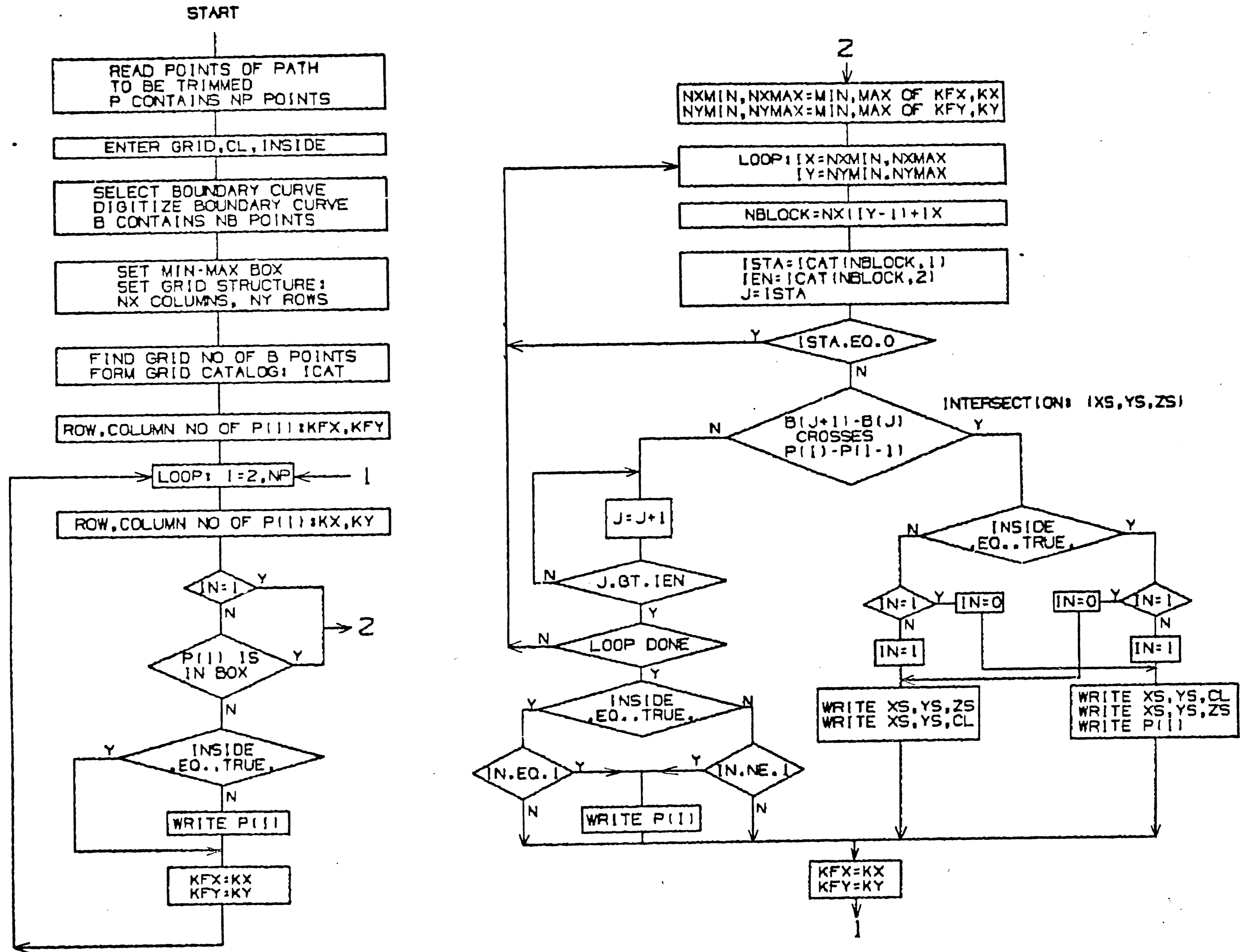


Figure 3-14: Flowchart For NC Tool Path Trim

$$t_1(P_{n,y} - P_{n-1,y}) + t_2(b_{k-1,y} - b_{k,y}) = b_{k-1,y} - P_{n-1,y}$$

in matrix form

$$[A] t = c$$

$$t = [A]^{-1} c$$

It should be noted that, with the grid structure, there is no need to check all boundary segments against each cutter step. The selection of grid size determines how many line segments of the boundary are to be checked each time. A small grid size decrease the number of segments to be checked making a fast response possible. In some regions, however, if the distance between consecutive boundary points is much larger than the grid size, detection of intersection might fail. Especially for boundary curves having straight line portions, a large grid size is suggested. Currently, there are two limitations for the definition of the boundary curves. First one is the  $C^1$  continuity requirement which doesn't allow combining curves with different end slopes. If the current digitizing algorithm is expanded by a trimming feature which will trim the boundary offset curves at the intersection points,  $C^1$  continuity requirement loses its validity. Another limitation is that the closed boundary curve must be defined in clockwise direction with respect to +z-axis. Wrong direction causes offset points generated on the wrong side of the curve.

After the implementation of the tool path trim module, case studies have shown that the trimmed paths leave excess material in the vicinity of the boundary. A final pass tangent to the boundary was considered as an appropriate solution and the module was expanded to allow such a finishing cut around the boundary. For generation of this additional path, first, offset (in or out) of the 2-D boundary curve is calculated with user-defined step size

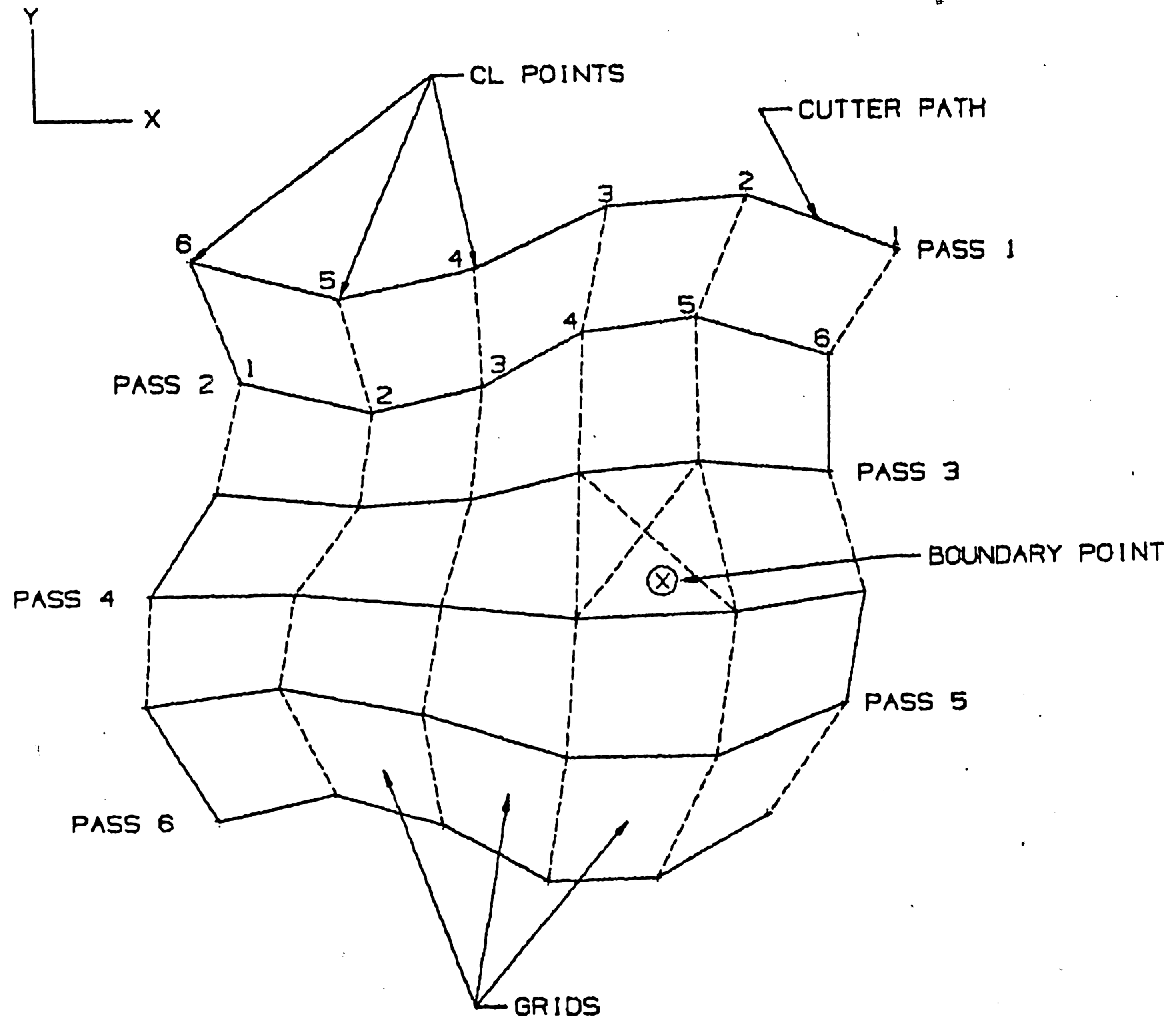
tolerance. Then z-coordinates of the resulting offset points are determined by applying an algorithm given in figure 3-15.

```
for each boundary point do
  for each pass do
    for each point on the pass do
      define grid
      if ( grid contains B.P. ) then
        find two triangles containing B.P.
        set their plane equations
        solve for two different z-coordinates of B.P.
        take the average
      endif
    enddo
  enddo
enddo
```

**Figure 3-15:** Algorithm For Final Pass Calculations

The exact solution for this problem would be to find the intersection of a line which passes through the boundary point and is parallel to z-axis with the original surface. Instead of dealing with surface canonical forms and successive iterations, original cutter location data which contains digitized information about the surface geometry is used by this algorithm. Each boundary point falls inside the projection of a quadrilateral grid formed by four cutter location points which lie pairwise on 2 consecutive passes ( see fig. 3-16 ). Beginning from the first set of four points, every grid is checked until the grid of the boundary point whose z-coordinate has to be determined is found. Corner points of this grid define four different triangles ( see fig. 3-16 ). The next step is to find which two triangles contain the boundary point. Then, the plane equations of these triangles are formed and knowing x and y coordinates, z coordinate is calculated. The average of z values resulting from two different plane equations is accepted as the z coordinate of the cutter location.

$P_1, P_2, P_3$  : corner points of triangle



42

Figure 3-16: Grid Structure for Final Pass Algorithm



$B = (b_x, b_y, b_z)$  : boundary point

plane equation :

$$n P = d$$

where

$$n = ( P_2 - P_1 ) \times ( P_3 - P_2 )$$

$$d = n P_1$$

substitute B into the plane equation:

$$n_x b_x + n_y b_y + n_z b_z = d$$

$$b_z = ( d - n_x b_x - n_y b_y ) / n_z$$

In order to be able to form the grid structure, the number of points on each pass should be known. Parametric machining modules of previous version of CISPA was not outputting this information into the machining file. Therefore, they have been modified in such a way that the point records are preceded by a three-word record. First word is the number of points on the pass and the last two are loaded with -9999 to indicate the special type of the record. If the number of points on 2 consecutive passes are not equal, last two points of the pass having less points are repeatedly used for the rest of the grids. Flow diagram of the final pass algorithm is given in figure 3-17.

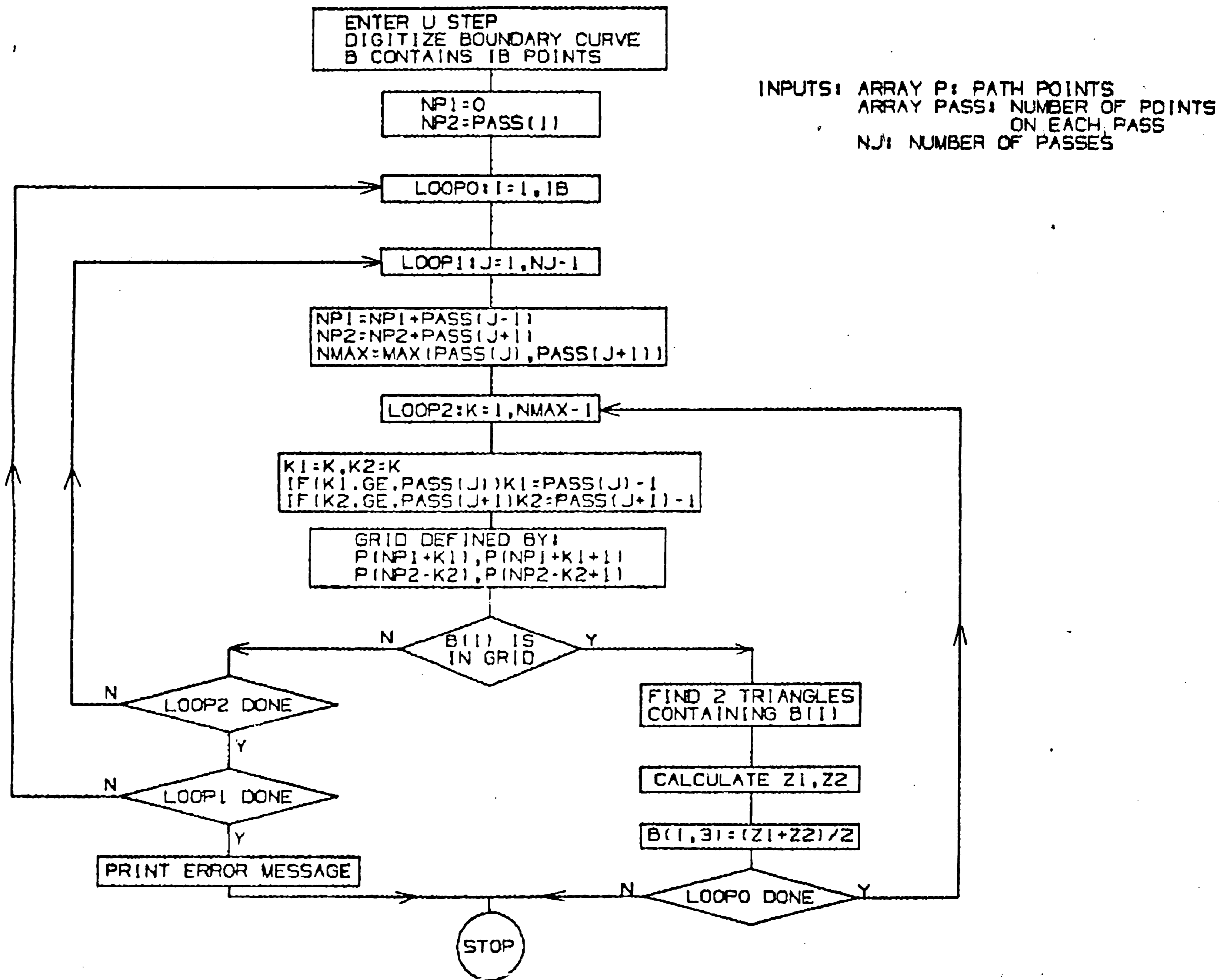


Figure 3-17: Flowchart For Final Pass Algorithm

# Chapter 4

## PROGRAM STRUCTURE / PROGRAM OPERATION

### 4.1 HARDWARE

Currently, CISPA version II may be run either on one of the host computers VAX 11/780 and VAX 8300 with VMS, or on the Micro VAX II with Micro VMS operating systems. The graphics may be displayed on Tektronix 4107 or 4109 raster graphics terminals with a 640 x 480 pixel resolution. DEC VS11 raster graphics terminals with 512 x 512 resolution may also be used.

### 4.2 MAJOR MODULES AND PROGRAM OPERATION

As mentioned in chapter 1, CISPA consists of the following software modules.

1. Graphics package
2. Mathematical definitions
3. Interactive front end

The graphics driver of CISPA is called GRAPH3D, permitting display on both DEC VS11 and Tektronix 4107/4109. Information about the routines of GRAPH3D which can be used for basic plotting, advanced graphics, transformations for real time motion, and interactive input purposes may be found in the reference manual 'GRAPH3D Graphics Package' (8). Only lower level graphics routines of GRAPH3D Library for plotting of points and symbols, drawing of lines, and cursor control, are utilized by CISPA.

Mathematical definitions module is responsible for creation of curve- and surface canonical forms and consists of some CASPA routines which have been directly taken from the APT4 execution phase. APT107 is the main module for definition of sculptured surfaces, synthetic curve definition is accomplished by APT108. Storage and retrieval of resulting canonical forms are the tasks of APT094. Reference (11) contains detailed information about the form of input stream to above mentioned modules.

Interactive front end may be further subdivided into the following submodules.

1. Geometry creation
2. Entity management
3. File management
4. NC machining
5. Display and view control

Geometry creation module consists mainly of routines which prepare and dispatch formatted data to the APT complex. Routine MSCURV guides definition of different types of curves. Available synthetic curve types and modules called by MSCURV to form their input data, can be found in (1). A previously defined curve may be transformed in 3-D space by selecting the newly added option in the synthetic curve menu. To transform a curve, MODCURL loads its input data into the work array TEM, and CURTRAN makes the necessary changes according to the user-entered transformation type and data. Successfully defined curves are saved on scratch file CISPA.SCR by executing the routine SAV CURV.

MSSURF displays the menu showing available sculptured surface subtypes

and calls the specific routine of the selected subtype. This module accomplishes the selection of curves forming the basis of the surface and dispatches the locations of canonical forms together with other necessary data to the APT107 complex. Curve ( or surface ) selection in any part of CISPAC is guided by the routine PCKENT which calls PCKNAME in the ' Select by Name ' case or PCKCURV ( or PCKSURF ) if ' Pick by Cursor ' option has been selected.

Currently, entity management section works only with curves. The routine RETRIEV reads curves input data from user-specified model file and passes it to the synthetic curve processor. It makes calls to READ\_WORDS which parses a string of characters to obtain the list of names to retrieve, and READ\_COM to read the input data from an external storage area. At the end of RETRIEV, a call to SAV\_CURV puts the input data of a retrieved curve into the scratch file CISPAC.SCR. DELETE is responsible for deletion of points and curves. For the selection of curves to be deleted, routines PCKNAME and PCKCURV are called repeatedly offering continuous selection.

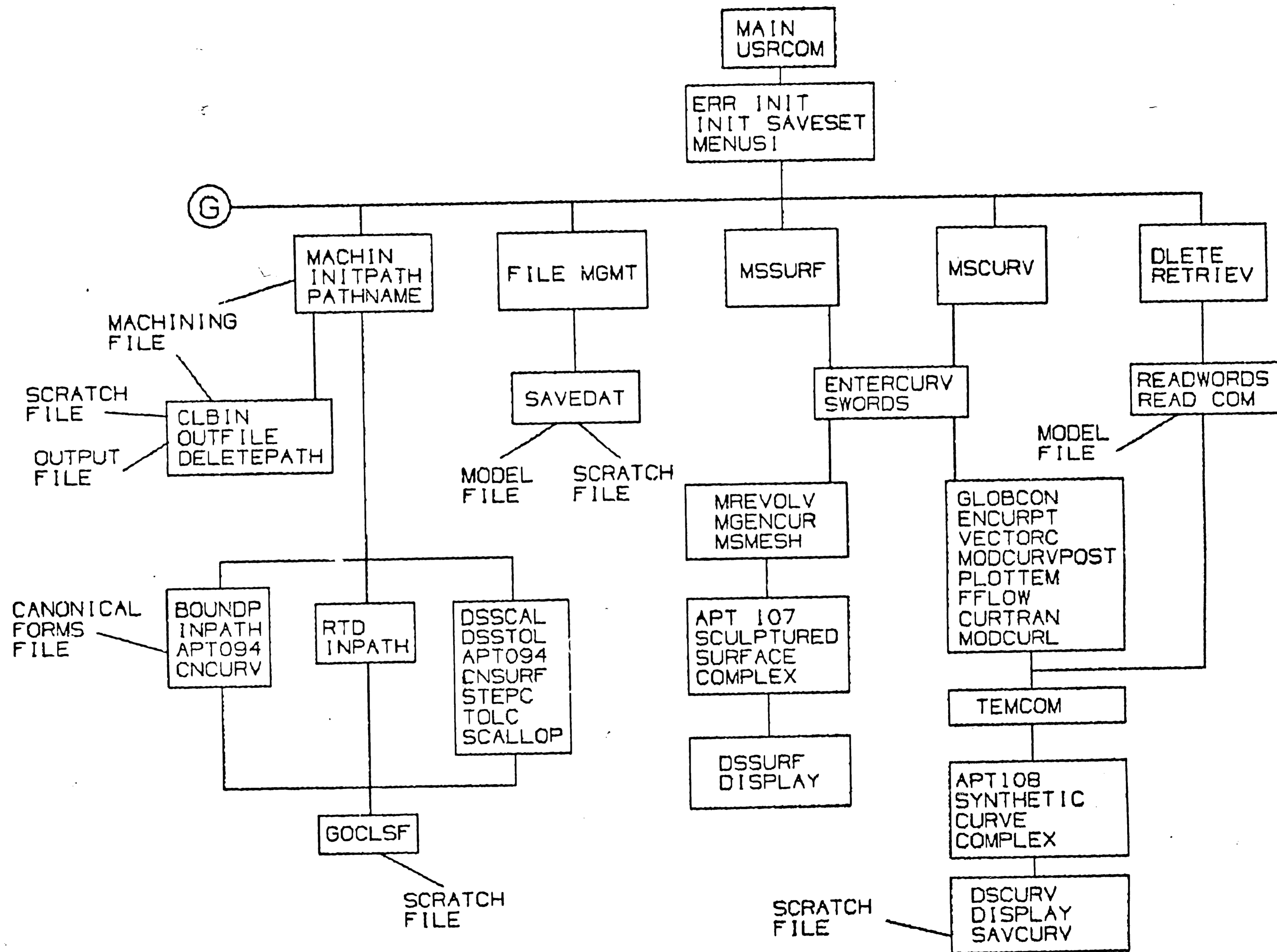
File management appears as an option of the main menu for menu driven mode. The routine FILE\_MGMT allows renaming and saving of the model file. Active curves are saved by copying undeleted parts of the scratch file CISPAC.SCR to the model file.

NC machining section is controlled by the main routine MACHIN which first calls INIT\_PATH for the identification of the machining file. If an existing file name is entered, INIT\_PATH copies its contents to the scratch file CISPAC.SCR and makes all stored NC tool paths active. After this step, MACHIN allows specification of machining parameters and calls the corresponding routine for generation of a cutter path of selected type. DSSTOL

and DSSCAL are parametric machining routines allowing variable primary step size based on chordal deviation and variable secondary (sideways) step size based on scallop height. Before one of these two routines is called, beginning point and primary cut direction are determined by executing UVDIR. This routine displays arrows at the corner points of the surface by calling DRWARW, and after selection of an arrow with cursor, starting patch number and parametric cut direction are returned. Besides parametric machining, Rough to depth and NC tool path trim are the other two options of the 'Surface Milling' menu displayed by MACHIN. The module RTD reads the data of a previously defined tool path and determines its projections to user-defined cutting planes. Resulting so called rough to depth tool path allows milling of the surface in incremental layers. NC tool path trim is accomplished by BOUNDP which trims a previously defined tool path according to the user-selected 2-D boundary curve. If the user wishes to create an additional tool path tangent to the boundary, BOUNDP calculates z-coordinates of the boundary points by calling INTERSECT. Rename and save operations on the machining file are the tasks of the routine CL BIN. Active tool paths are saved by copying undeleted parts of the scratch file CSPATH.SCR to the machining file. Finally, routine OUTFILE is responsible for writing cutter location data of a specified tool path to a user-defined output file in the form of GOTO/ cards.

Display and view control features of CISPA are embodied in the global menu which can be accessed by hitting the 'G' key. The routine GLOBME guides this section by calling the corresponding routine assigned to each option. Newly implemented BLANK\* allows the selection of entities to be blanked or

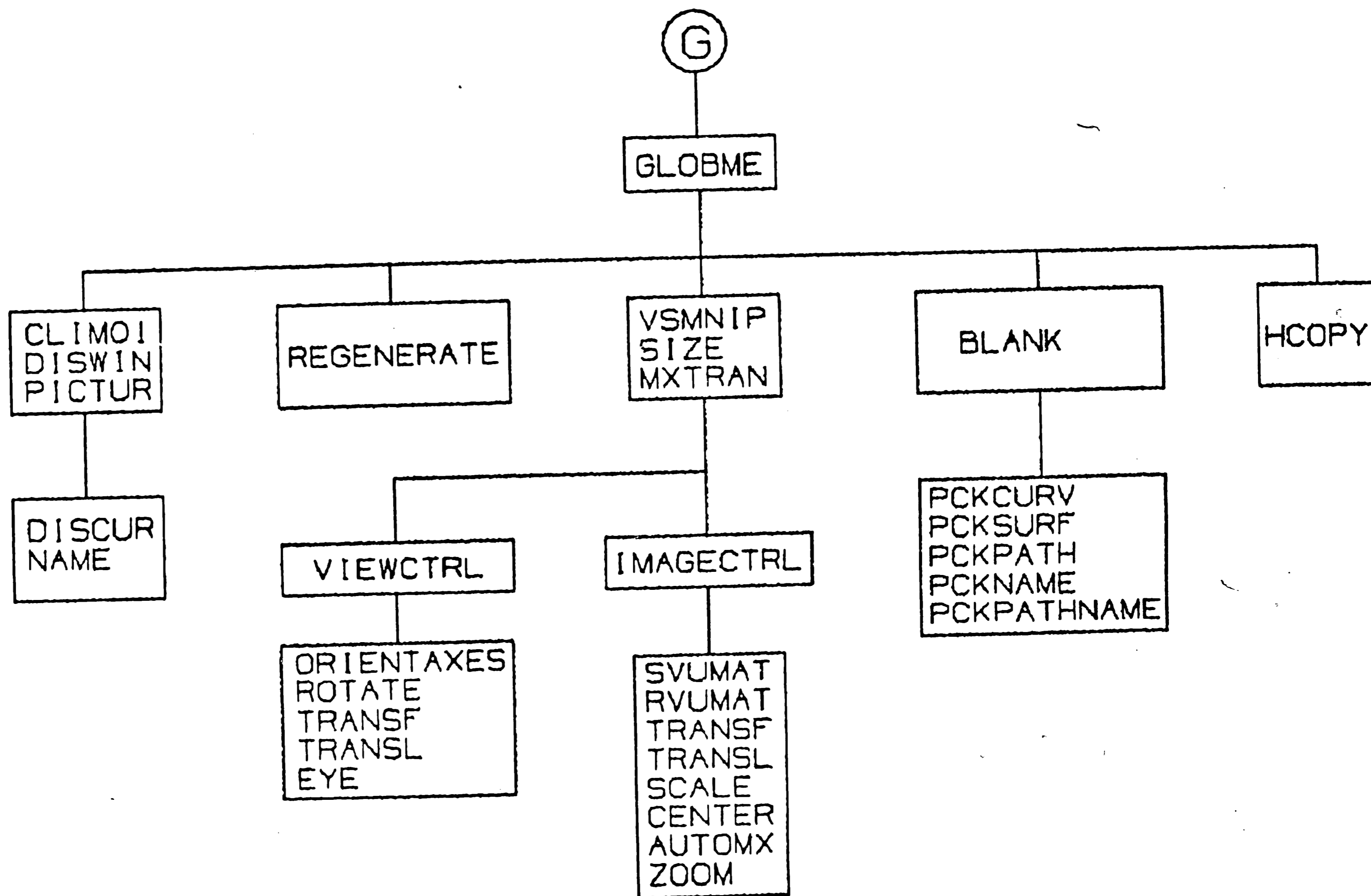
unblanked and sets corresponding flags. REGENERATE cleans the picture file from gaps caused by deleted entities. If view manipulation has been selected, GLOBME calls VSMNIP which serves as a switch to transfer the program operation control to the routines for display and view control under the GRAPH3D package. The functions they provide are discussed in detail in the 'CISPA.LU PACKAGE MANUAL' (5). Refer to Fig. 4-1 for the complete subroutine call structure of CISPA.



50

Figure 4-1: Subroutine Call Structure For CISPA





# Chapter 5

## CONCLUSION

### 5.1 SUMMARY

CISPA is a menu-driven CAD/CAM package which allows creation and NC machining of free form surfaces. The origin of CISPA's mathematical definitions section is another sculptured surface processor named CASPA03 developed by CAM-I. During the first phase of the development study, the basic structure of an interactive front end for a set of APT routines used in CASPA has been formed. Modification of CASPA NC machining modules, as well as addition of new algorithms like step size based on geometric attributes were also accomplished in this phase. Second phase of the development study consists of the enhancement work explained in previous chapters.

Chapter 2 introduces mathematical formulations of synthetic curves and surfaces used in CASPA. Curves are represented in parametric cubic rational form. For sculptured surfaces, bi-cubic patches with a two parameter formulation are utilized.

Newly implemented algorithms and data structures are discussed with flow charts in chapter 3. For a more efficient manipulation of curve and NC tool path related data, model and machining file concepts were implemented. Some interactive features like entity picking and default name creation were improved. Addition of a blanking/unblanking feature and curve transformations has contributed in the creation of a more powerful system. NC machining module of CISPA has been also modified: beginning point and primary cut direction for tool paths for surface parametric machining was made user-selectable, errors

encountered by the existing rough-to-depth routine were removed with the implementation of a new RTD algorithm and finally a NC tool path trim feature was added. This feature allows the user to create a NC tool path which can be used for milling the bottom surface of a cavity surrounded by general walls which are generated by projection of a 2-D curve onto the surface.

Next, program structure/operation is explained with information about the hardware required and description of major modules. During the first phase of the development study, CISPA was implemented on VAX 11/780 with DEC VS11 raster graphics terminals used for graphics display. However, experience of the last months has shown that Micro VAX II - Tektronix 4107/4109 provides a workstation configuration with a good response.

## 5.2 LIMITATIONS

As mentioned earlier, CASPA's language input is still available through a software switch. But currently, there is no data compatibility between command mode and menu driven mode of CISPA. For example, input data of the entities created in the command mode is not written to the scratch file. That means, these entities can not be retrieved, modified or i.e. they are not recognized by the menu driven mode.

Number of entities that can be created and stored in a session, is limited. For NC tool paths this limit is 20, and only a total number of 50 curves and surfaces may be created. Currently, storage and retrieval of surfaces is also not available. Therefore, if a surface is to be used in another session, all curves input to its definition are to be stored.

The database of CISPA contains curve input data and not canonical forms. Each time a curve is retrieved, its canonical form is formed by

processing the input data. For a more efficient retrieval operation, however, both input data and canonical form should be stored.

### 5.3 FUTURE STUDY

Implementation of the following items will be emphasized during the continuation of the enhancement work on CISPA.

- Working coordinate system, working plane definition for entity and tool path creation.

- Automatic constraint definition for 2-D curves to improve their construction. In this way, untrained users will be offered more traditional creation techniques for arc, conic type of curves.

- Automatic constraint definition for surfaces of different categories. This will increase the current number of surface subtypes which can be created without requiring the user application of special constraints.

- Utilizing a ring type data structure for more efficient storage of input and canonical form data. This could be done with additional coding work on the existing database manipulation routines (13) for adding, deleting, modifying entity related data. Such a data structure is also more suitable for line command-to-interactive menu and vice versa switching mode process.

## REFERENCES

1. Loney, G.C., "Interactive Design and NC Machining of Free Form Surfaces", CAD Laboratory, Department of Mechanical Engineering and Mechanics, Lehigh University, 1985.
2. Schoenberg, I.J., "Contributions to the Problem of Approximation of Equidistant Data by Analytical Functions", Q. Appl. Math., vol. 4, pp. 45-99, 112-141, 1946.
3. Rogers, D.F. and Adams, J.A., "Mathematical Elements for Computer Graphics", McGraw-Hill Book Co., 1976.
4. Faux, I.D. and Pratt, M.J., "Computational Geometry for Design and Manufacture", John Wiley and Sons, New York, 1979.
5. Loney, G.C., Ozsoy, T.M. and Smith, R., "CISPA.LU Package Manual", CAD Laboratory, Department of Mechanical Engineering and Mechanics, Lehigh University, 1985.
6. CAM-I, "Consolidation of User Documentation for the CAM-I Sculptured Surfaces SSX6, SSX6A, and SSX7 System Releases", Publication No. PS-78-SS-02, Computer Aided Manufacturing International, Inc., Arlington, Texas, 1978.
7. CAM-I, "Documentation for Computer Aided Sculptured Pre-APT System ( CASPA.03 )". Publication No. PS-78-SS-01, Computer Aided Manufacturing International, Inc., Arlington, Texas, 1978.
8. Ozsoy, T.M., Bhalla, S., Summer, R., and Tress, D., "GRAPH-3D Graphics Package", CAD Laboratory, Department of Mechanical Engineering and Mechanics, Lehigh University, 1985.
9. CAM-I, "Mathematics for Representing Conics as Coon's Rational Parametric Cubic Curves", Publication No. R-75-SS-04, Computer Aided Manufacturing International, Inc., Arlington, Texas, 1975.
10. Tektronix, Inc., "4107, 4109 Programmers Reference", Tektronix, inc. Beaverton, OR., 1983.
11. CAM-I, "CAM-I Sculptured Surfaces Users Course", Publication No. TM-77-SS-01, Computer Aided Manufacturing International, Inc., Arlington, Texas, 1977.
12. Chen, J.J. and Ozsoy, T.M., "Sculptured Surface Modeling Using B-Spline Surfaces with General Boundaries", ASME Paper 85-DEC-79,

11'th Design Automation Conference, Sept. 10-13, 1985.

13. Kivanc, L. and Ozsoy, T.M., "Geometric Data Base Manipulation Routines Reference Manual", CAD Laboratory, Lehigh University, 1984.

## Appendix A

### CASE STUDY

A brief example follows showing the steps needed to create a surface and four types of NC cutter paths generated for milling.

1. Enter the CISPA program by picking the option MENU DRIVEN MODE and specify model file name and system of units.

2. Create primary curves C22 and C7 using the SPLINE definition, under the synthetic curve option, and inputting the proper tangential directions as constraints. Then C21 and C23 are defined by translating and scaling the curve C22 with 'copy' option. Finally, C15 is created by translating C7 in x-direction. The view may be changed by utilizing view manipulations under the global menu ( fig. A1 ).

3. These curves are re-parameterized using the ARC option under the flow menu ( attachment of a flow structure ) and a GENCUR type of surface is fit to the resulting curves.

4. Machining section of CISPA is started by specifying machining file name. First, parametric machining is selected and the NC tool path P1 is generated with machining tolerances of 0.02 in. in the primary direction and a scallop height of 0.02 in. in the second. The radius of the ballend mill is specified as .25 in. and the arrow on the bottom side of the surface pointing left ( fig. A2 ) is picked by cursor to indicate the beginning point and the primary cut direction.

5. Creation of a trimmed path is the next step, but first, following preparations have to be done. Using the BLANK option of the global menu, everything is blanked except the NC tool path created in step 4. For the

creation of the closed boundary curve, Geometry creation module is selected again and the curve C13 is created by first defining 8 single curves of SPLINE type and then combining them under the COMBINE option. Figure A3 shows what should appear on the screen after blanking of all these individual curve segments.

6. By using NC PATH TRIM option under the surface milling menu in machining module, P1 is trimmed first by selecting 'outside' and then 'inside' option and two new NC paths P2 ( see fig. A4 ) and P3 ( see fig. A5 ) are generated. In both cases a grid size of 1.5 in. and a clearance plane of  $z=1$  in. are input.

7. P3 is accepted and a final pass is requested by entering 'Y'. In this way, NC tool path P6, tool path around the boundary, which is the projection of 2-D boundary curve on the original surface ( see fig. A6 ), is created.

8. Finally, ROUGH TO DEPTH option of surface milling menu is utilized to create R1 ( see fig. A-7 ) by using P1 created in step 4. Following values are assigned to rough to depth machining parameters:

reference plane : 0.0 in.

clearance plane : 1.5 in.

part cl. plane : 1.0 in.

first z-step : 1.25 in.

subsequent z-steps : 0.5 in.

9. Each NC tool path is written to a different ASCII punch file for post processing. FILE TERMINATE then PUNCH FILE options allow the selection of NC tool paths to be transferred into the punch file.

10. Machining and model files are saved before the exit from CISPA.





Figure A-1: Primary Curves with Tangent Constraints

SURFACE MILLING

- 1 PARAMETRIC MACHINING
- 2 ROUGH TO DEPTH
- 3 NC\_PATH TRIM

G:GLOBAL MENU

SELECT SURFACE TO BE MILLED

- 1 BY NAME
- 2 BY CURSOR

SURFACE SELECTION LIST:  
S1

ENTER SURFACENAME#S1

ENTITY S1 SELECTED  
ENTITY NUMBER 11

ENTER NC PATH NAME (P1 ):  
TYPE OF SURFACE (1-PATCH,2-MESH): 2

SELECT FORWARD STEPPING

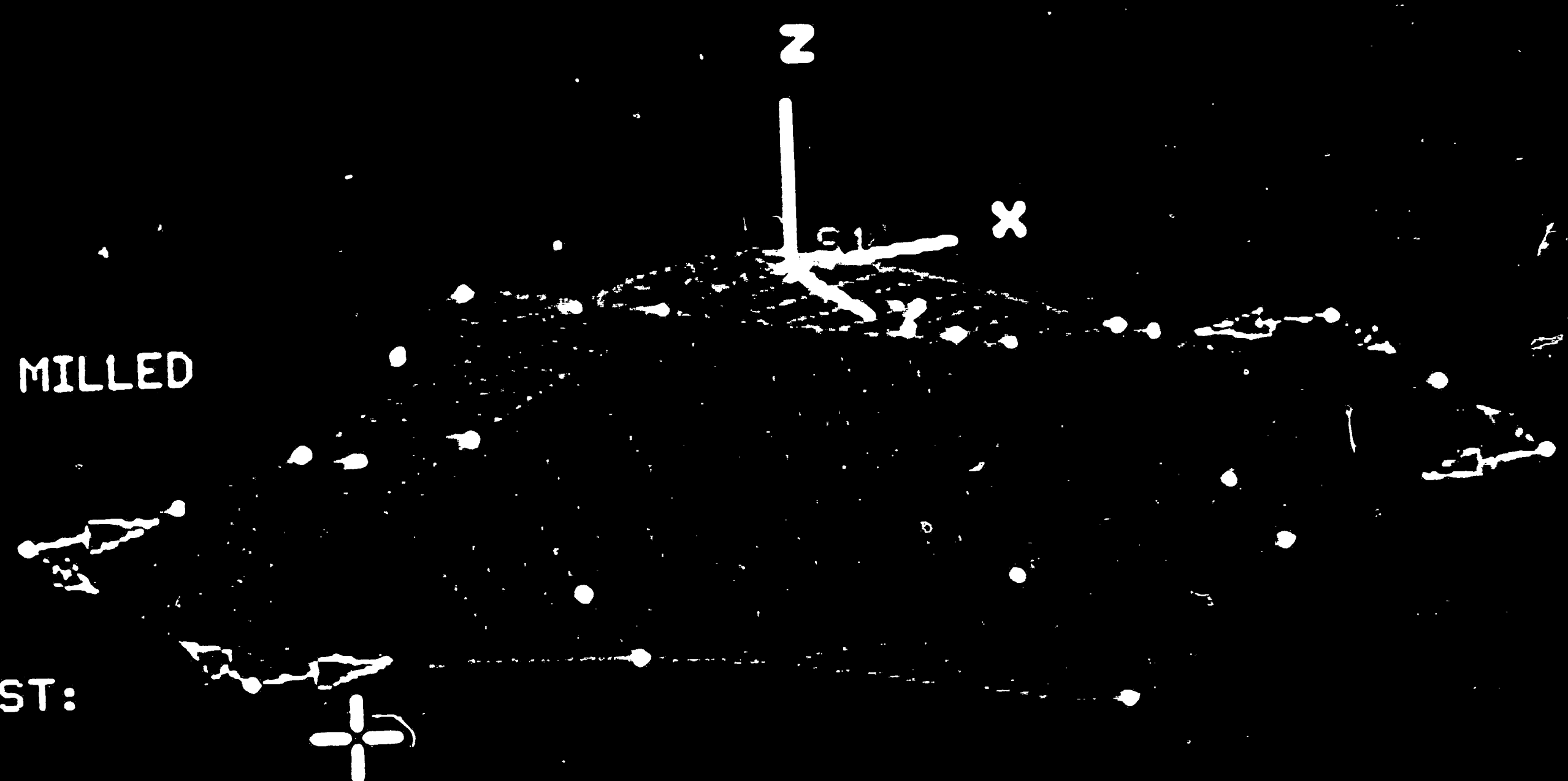


Figure A-2: Gencur Surface with Arrows Displayed for Beginning Point and Primary Cut Direction Selection

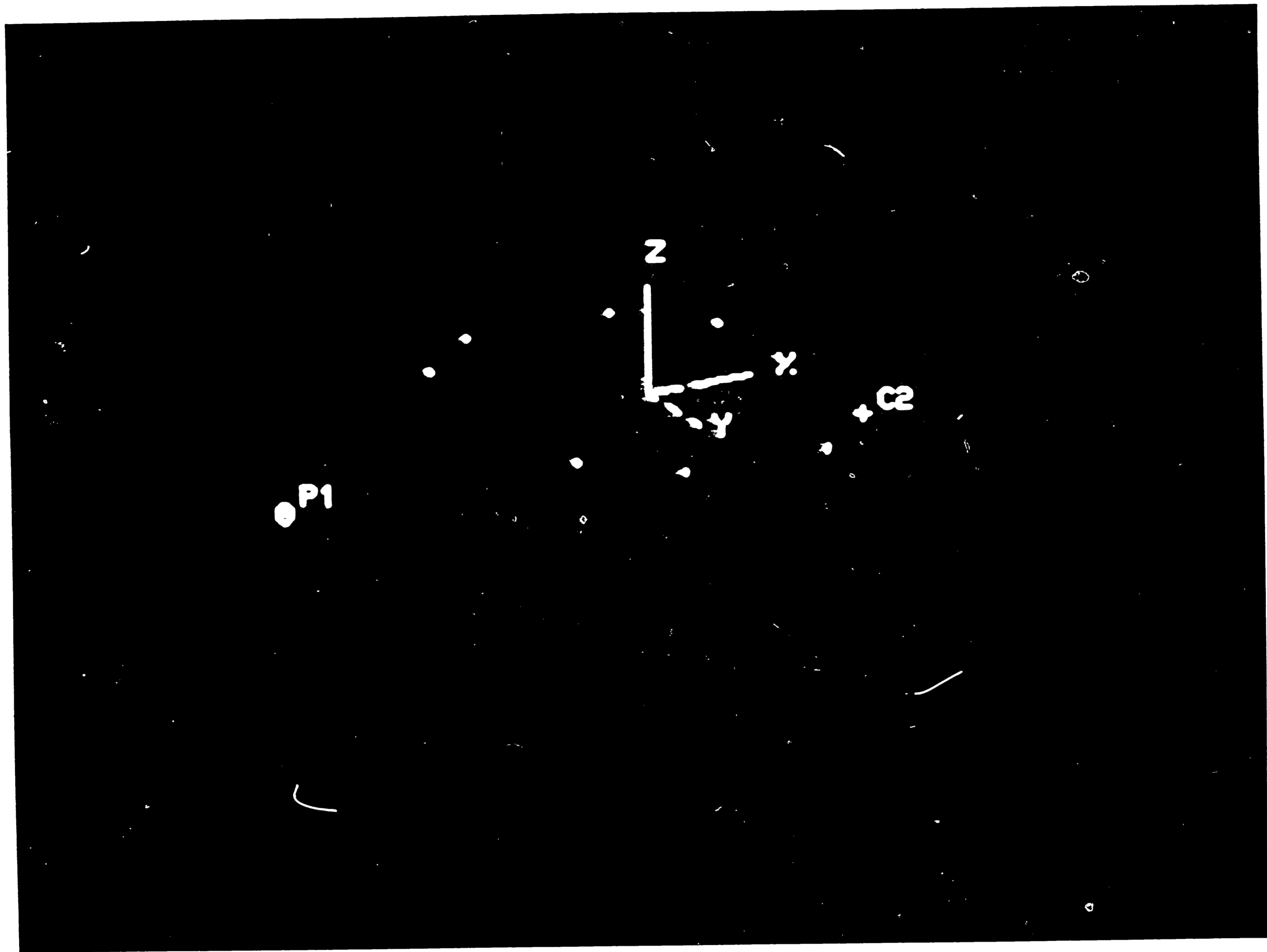


Figure A-3: Tool Path For Parametric Machining and Imposed  
2-D Boundary Curve



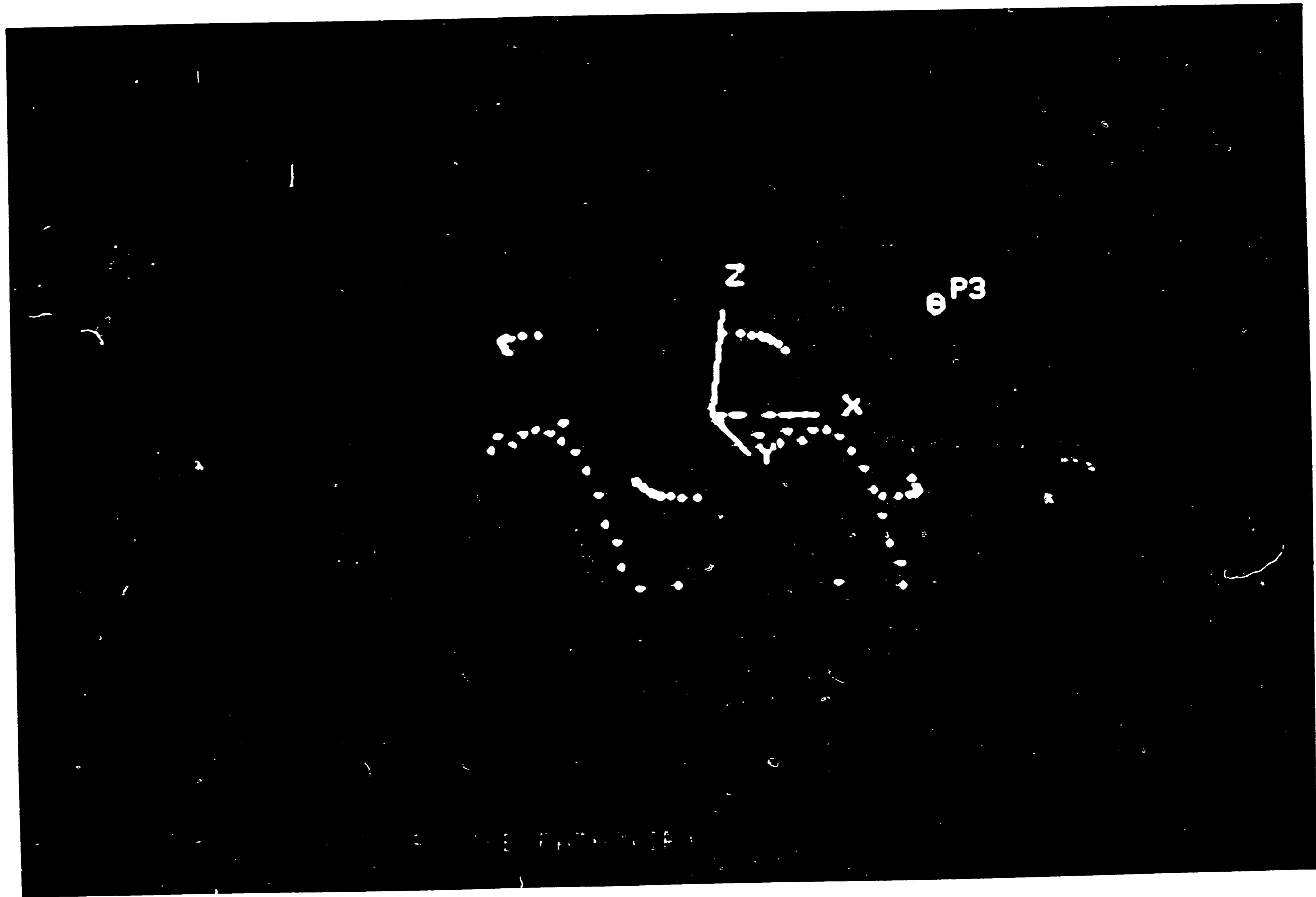


Figure A-5: Trimmed Path (inside)

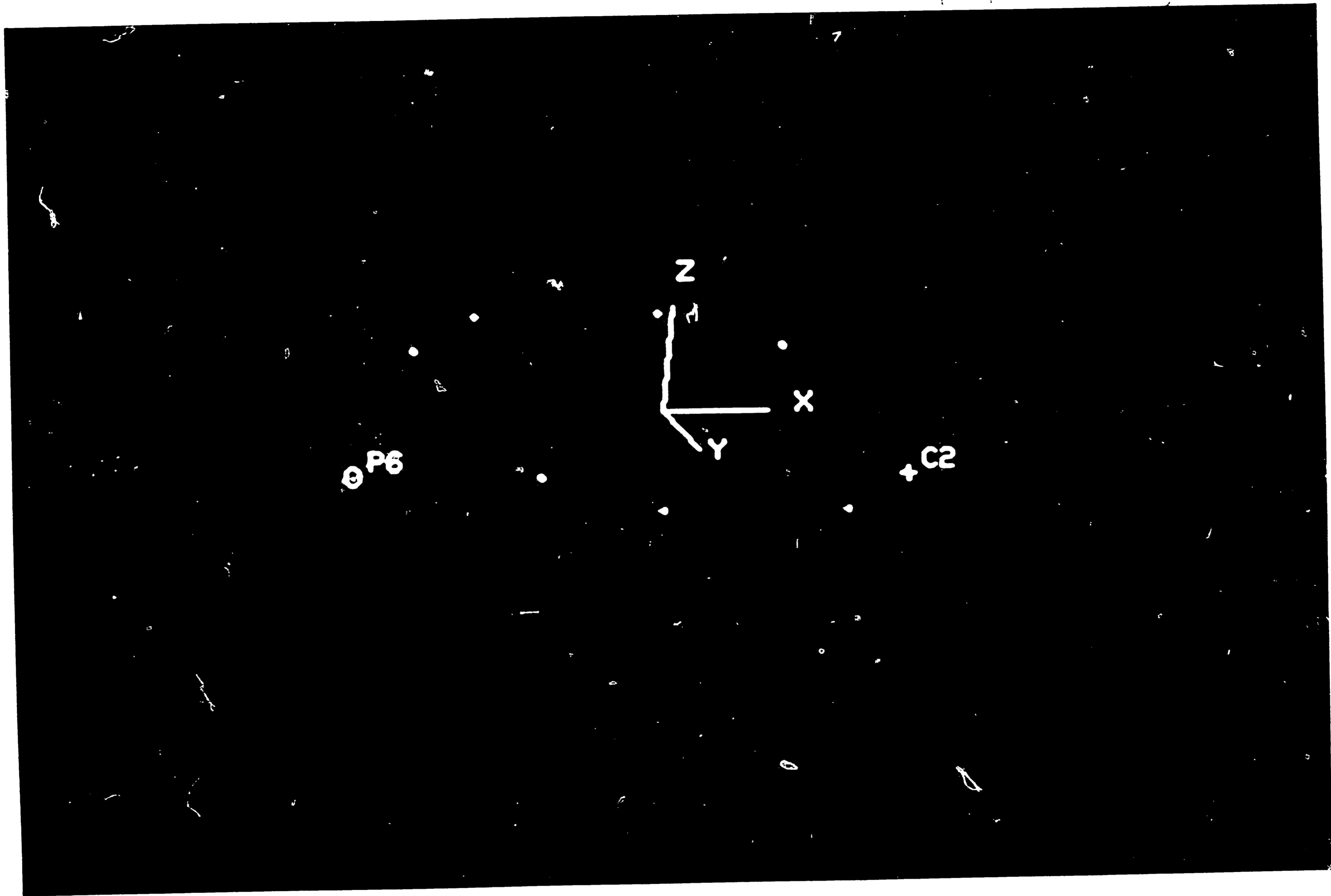
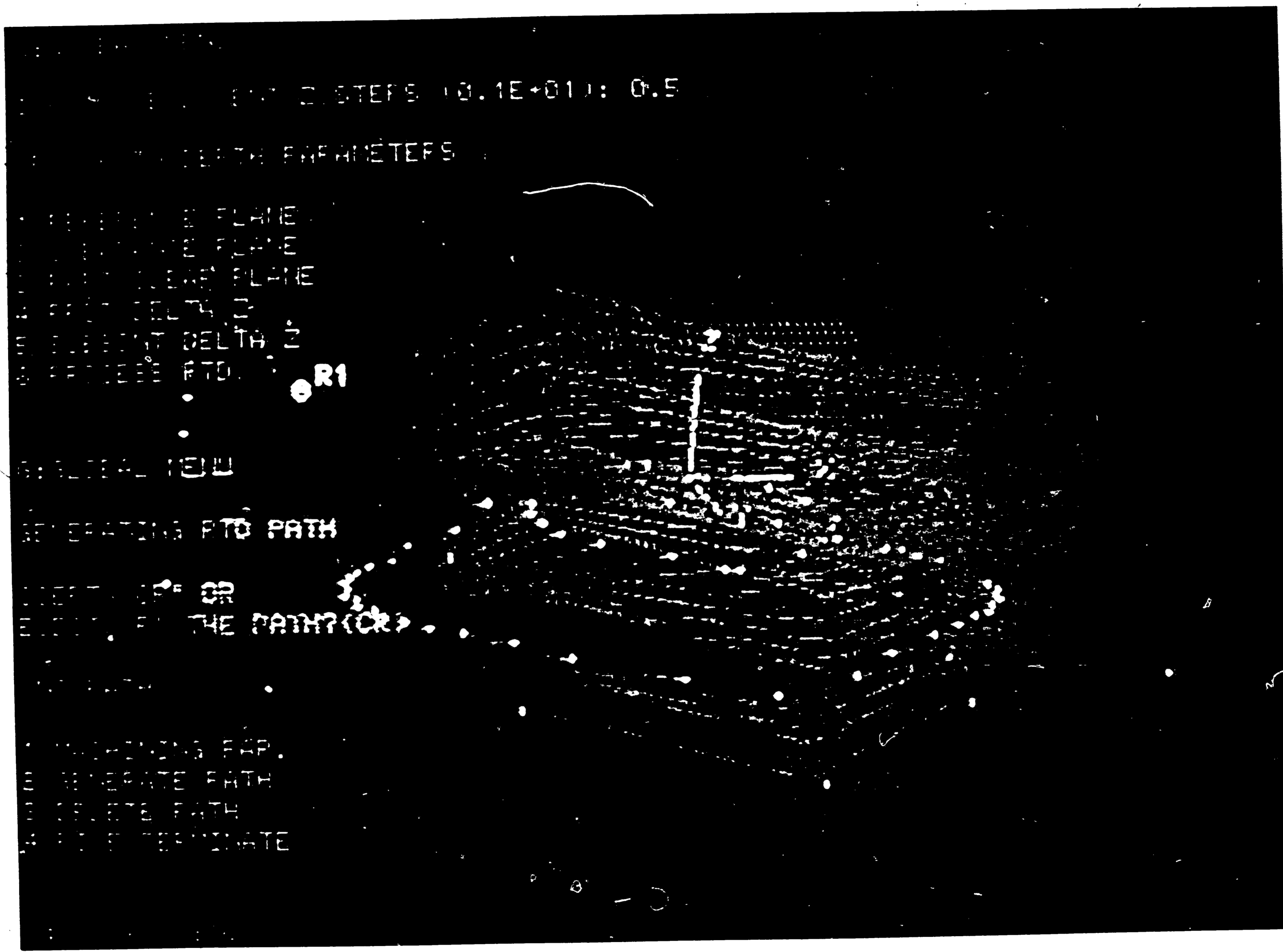


Figure A-6: Tool Path For Final Pass Around the Boundary



65

Figure A-7: Tool Path by Roughing to Depth

# Appendix B

## MENU TREE FOR PROGRAM OPERATION

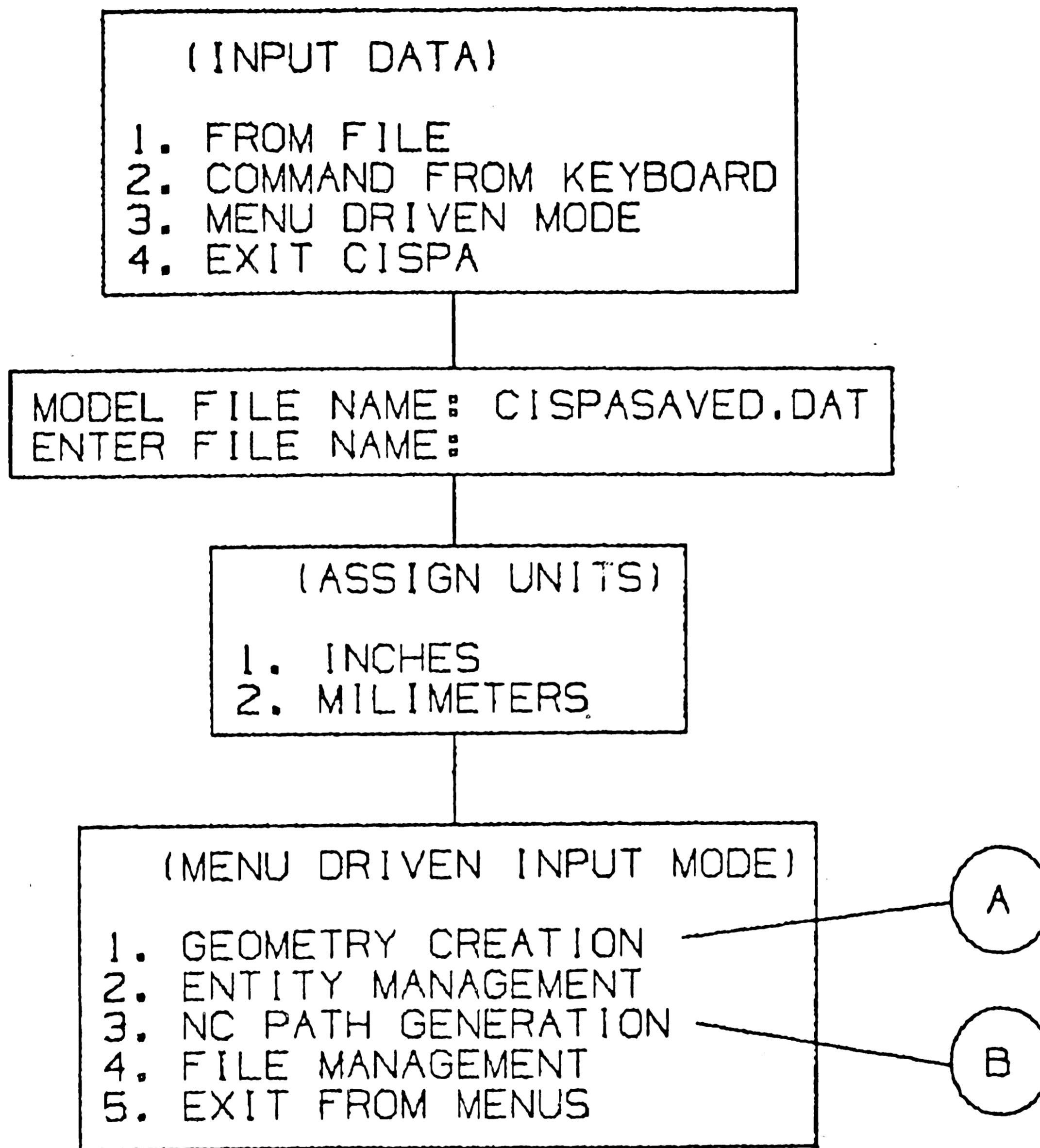


Figure B-1: Menu Tree For Program Operation



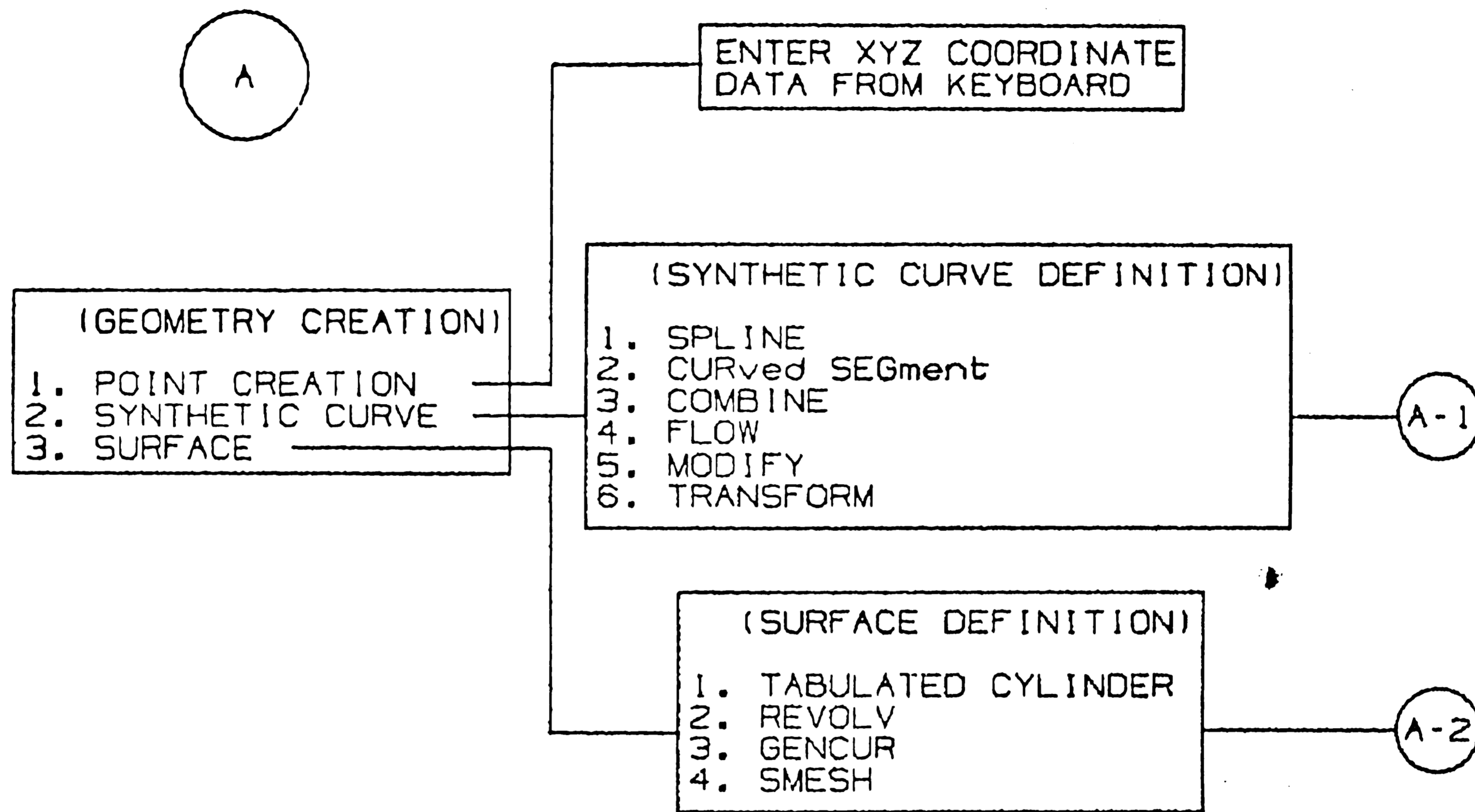


Figure B-1: Continued

A-1

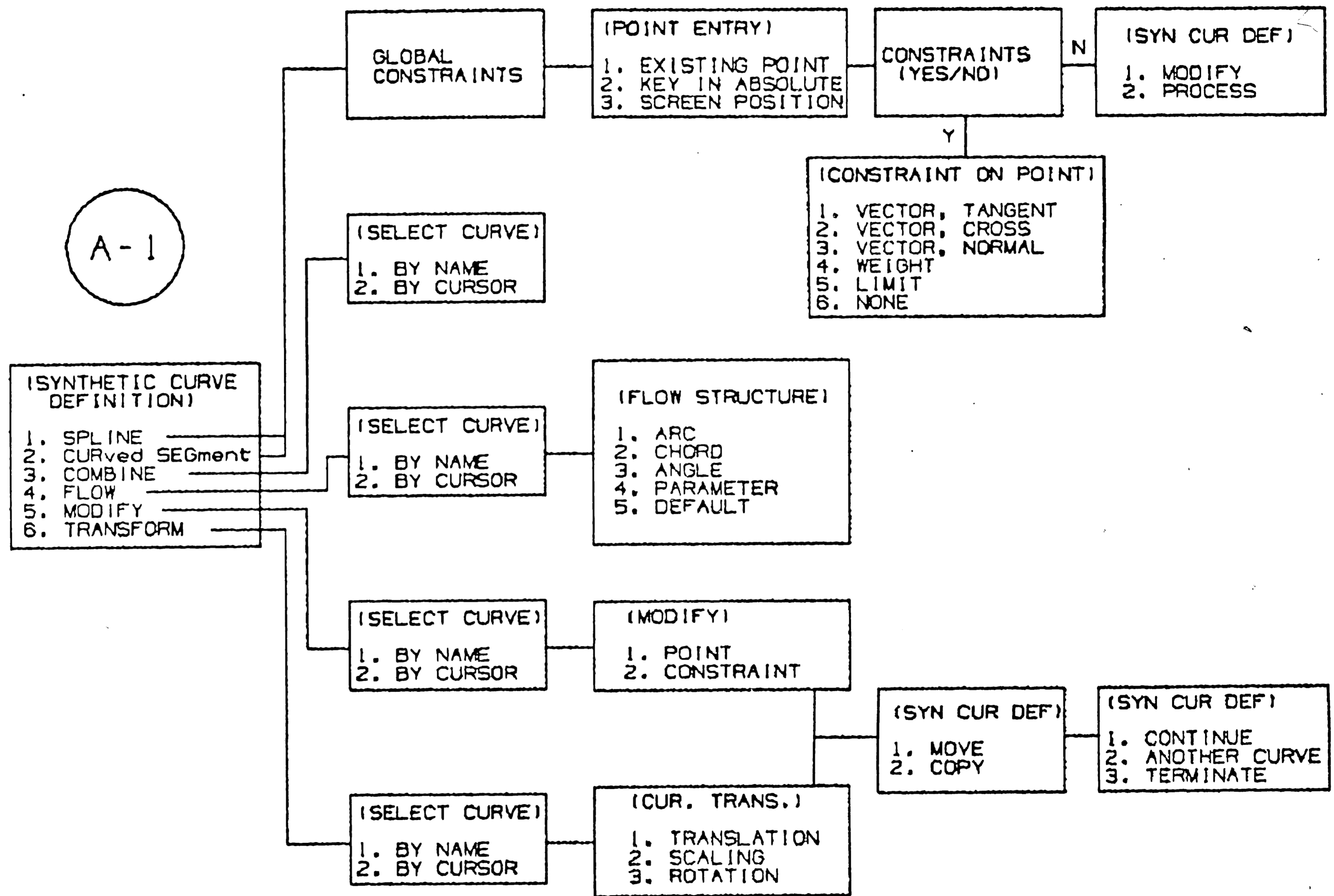


Figure B-1: Continued

A-2

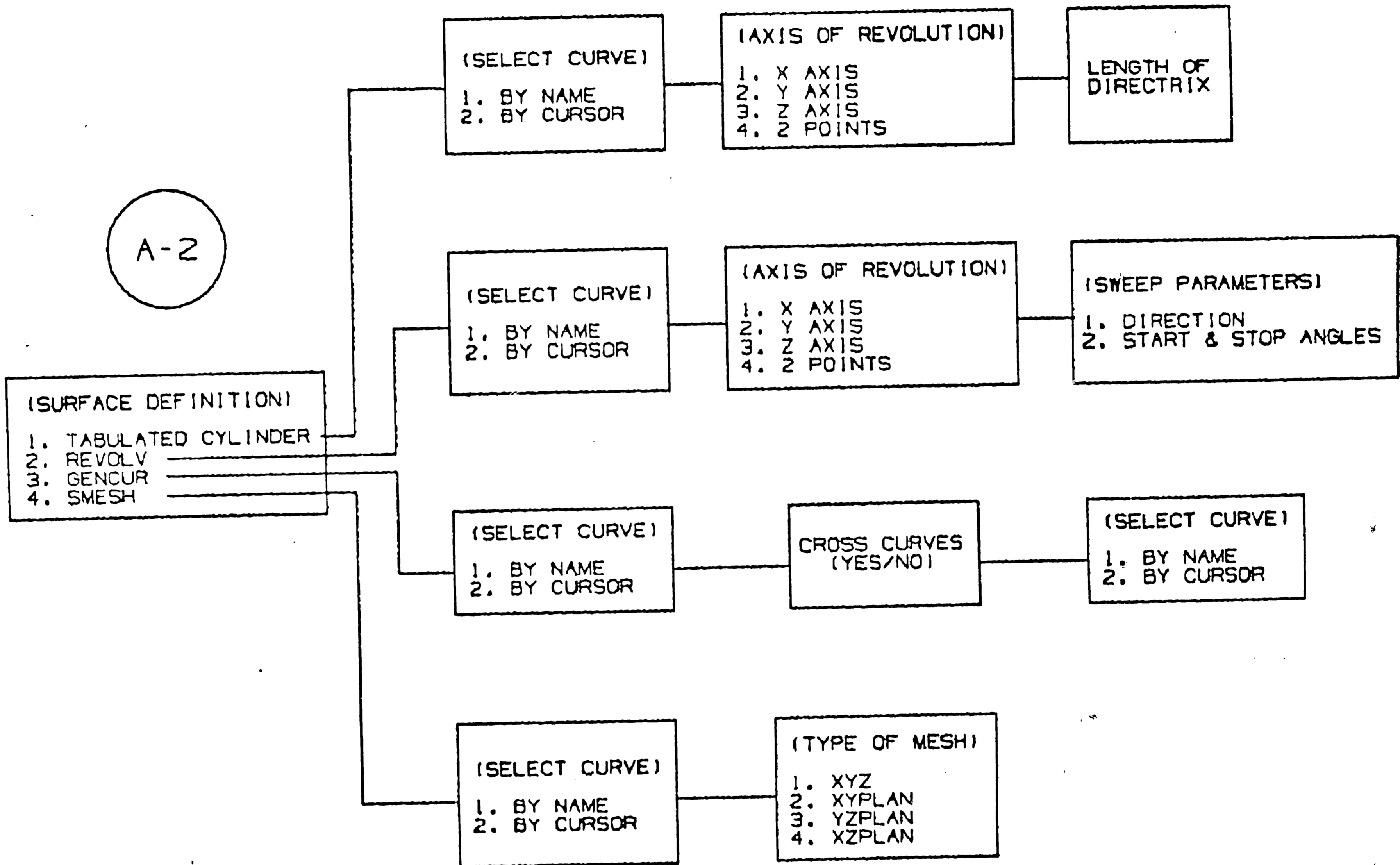


Figure B-1: Continued

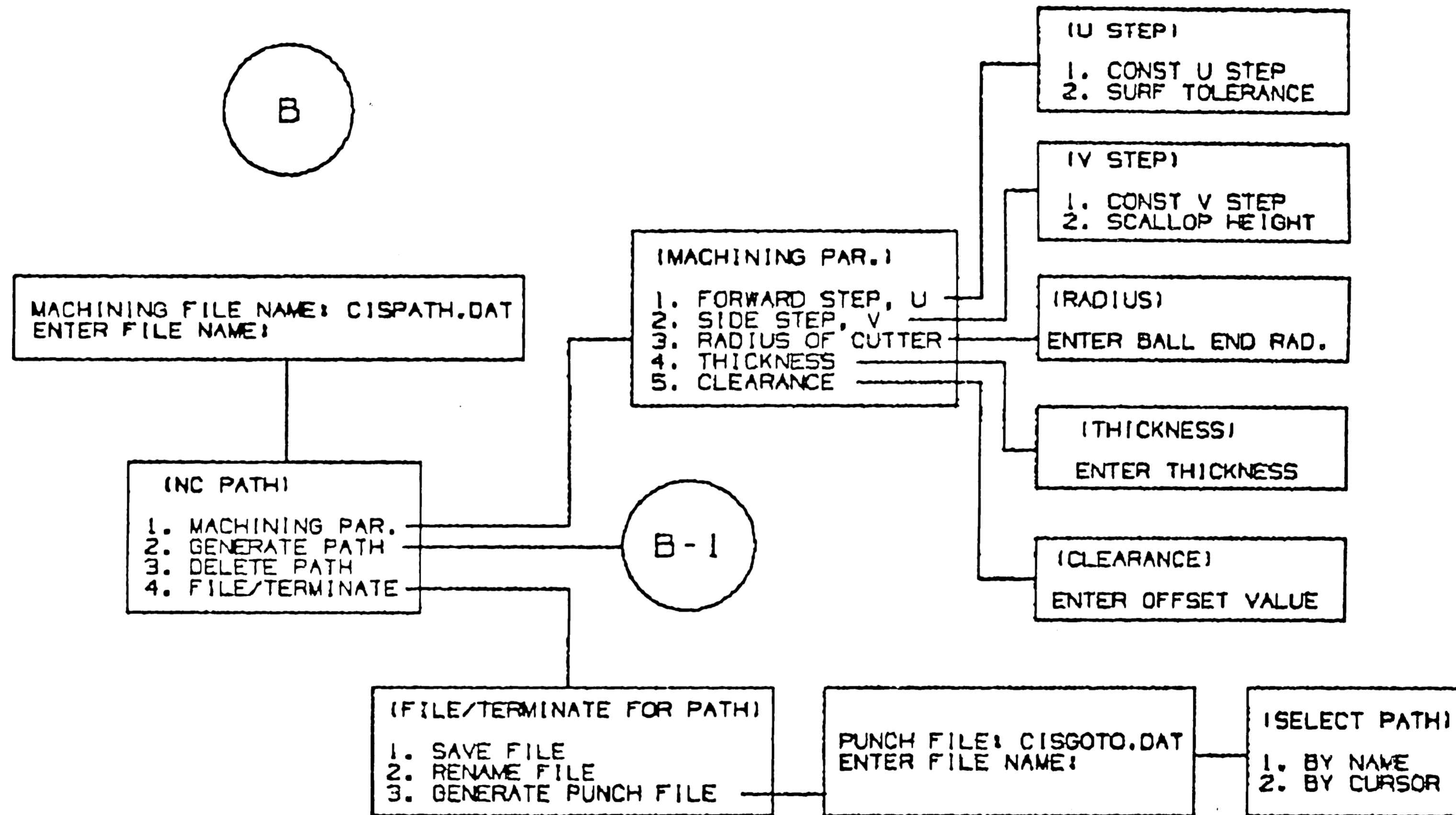


Figure B-1: Continued

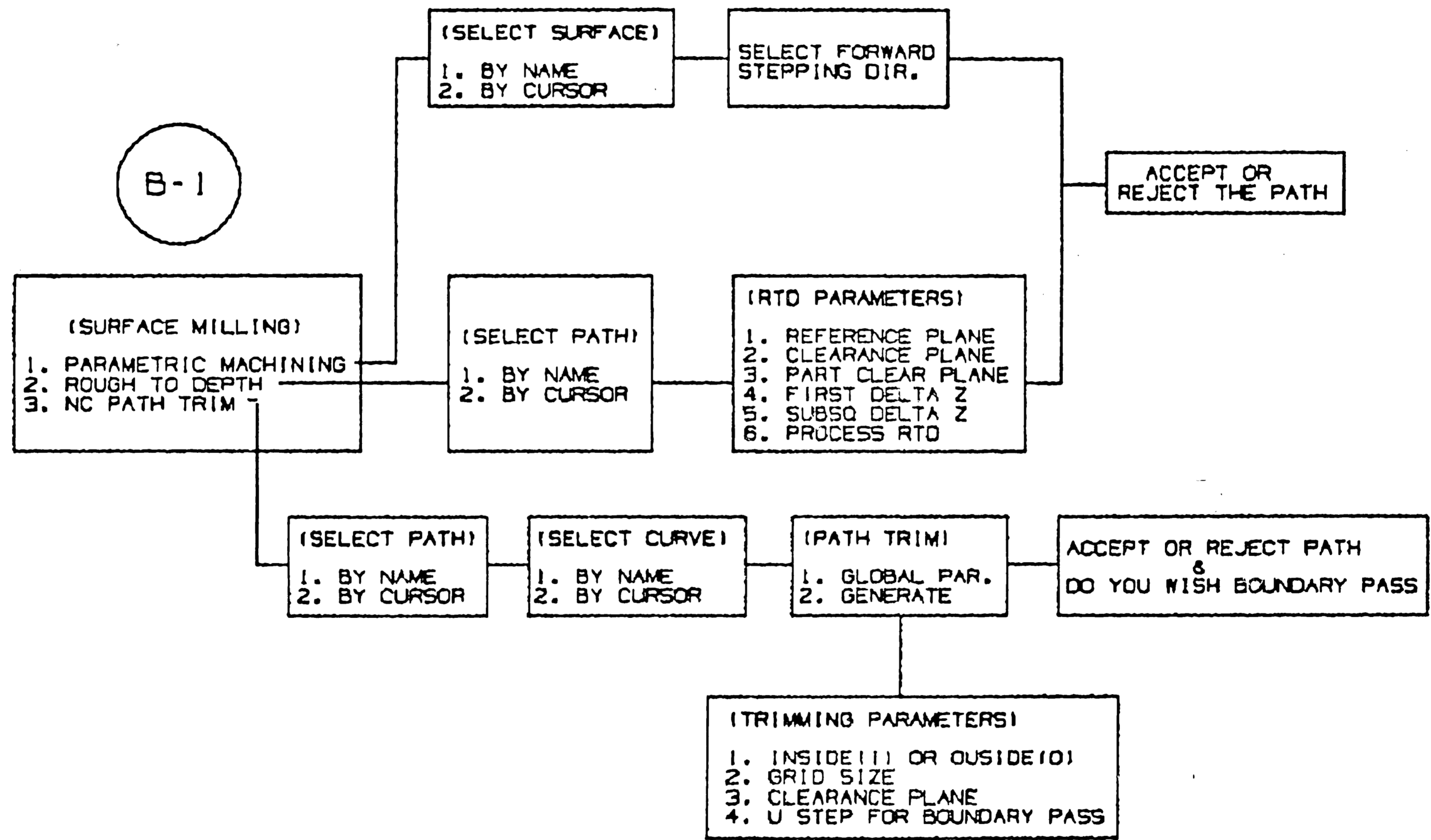


Figure B-1: Continued

# GLOBAL MENU

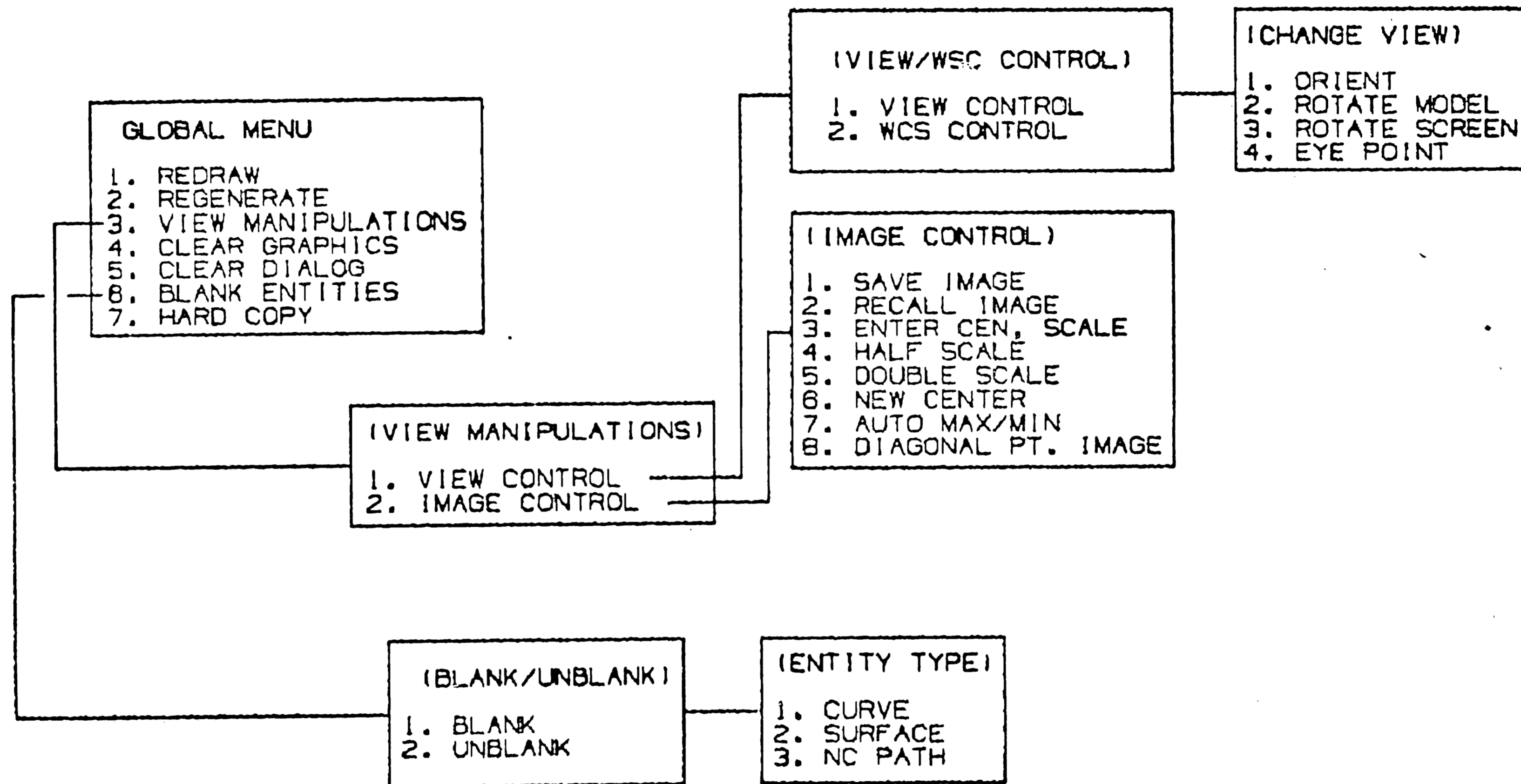


Figure B-1: Continued

## VITA

M. Sevket Tarhan was born January 20, 1963 in Istanbul, Turkey. He obtained a Bachelor's of Science degree, in Mechanical Engineering from the Bogazici University, Istanbul in 1985.

In the same year, he decided to continue his education at Lehigh University where he could specialize in robotics, CAD/CAM and finite element analysis, working toward a Master of Science in mechanical engineering.

His parents are Servet and Sumer Tarhan of Istanbul, Turkey.