TABLES

FIGURES

APPENDICES

1. INTRODUCTION

In recent years there has been a rapid recognition that computer-aided-design and manufacture (CAD/CAM) is a major technology that will significantly affect production methods and systems within manufacturing industry. This importance is reflected in the major finding of the Cabinet Office's ACARD CAD/CAM report (1) in 1980: 'The principal benfit of CAM. as well as CAD, is the improvement in productivity that can be achieved..... an increase by a factor of three or so may be expected with current equipment.' Sir Kenneth Corfield, Chairman of the Engineering Council, commented in September 1982: 'All designers should be aware of CAD/CAM and its possibilities.' This is also true for production engineers and NC programmers.

1.1 THE USE OF CAD/CAM IN ENGINEERING

All aspects of design and manufacturing need to use and convey information relating to the shapes of objects. From the very beginning of engineering, object information has been transimitted via drawings (i.e. formalised sketches), which had to be manually produced. When computers were introduced into the design and manufacturing environment, there arose a need for computer to understand the shapes of objects. Computer, however, could not readily be made to understand shapes. Consequently, special application languages were developed so as to introduced into computer the geometry of the objects geometry from which necessary calculations could be performed. In addition, computer graphic

device were developed to enable man to deal with computer based object in a form that could be readily understood and also to allow visualization, through graphic means, of the results of calculations that have been performed. Though much attention has been focused on this subject (e.g. APT geometries, aircraft and automobile type sculptured surfaces), the overall task of describing complete 3D objects of common engineering shapes has only just commenced. Initial works (2 - 8) in this area has progressed significantly. However, an acceptable standard means of describing and processing complete 3D objects, until recently, remains elusive. Neverthess, computer based modelling of 3D objects is believed achievable and as such can be expected to have an enormous impact on present day design, manufacturing and planning techniques.

1.2 THE ADVENT OF 3D SOLID MODELLING

Recently, 3D solid modelling has become recognised as the best way to communicate information about physical objects. With a complete computerised representation of the geometry of an object together with other relevant details of its shape, it is now possible to integrate design, analysis and manufacturing activities by serving all engineering functions with a single model representation.

Traditional CAD/CAM system vendors have often focused on drafting productivity. These systems deal with usually only the 2D representation of a part and do not deal with the complete geometry of it. Some systems maintain 3D information in terms of the x, y,

z co-ordinates of the points and lines of an objects to permit a wire-frame display of the object, with all lines visible regardless of the view point. There is no information in the geometry representation as to whether a plane bounded by these visible lines represented actual solid material on one side or the other (i.e. whether a point is inside or outside the model). Thus, the model represented by these lines does not really represent the object. It could not be used for analysis of the properties of the object (i.e. moment of inertia, centre of gravity etc.) or for rapid and straight-forward modification, manipulation and visualization (i.e. hidden line removal) of the object from various perspectives. Traditional systems are constrained by having to deal with the drafting function and because of the approach they use to represent geometry prohibits them from doing anything more profound than facilitating the conventional orthogonal or isometric drawings. Although line drawing systems can favourably affect drafting productivity, but not the total engineering/manufacturing productivity.

Some vendors describe solid geometric modelling as visualization capabilities, whereby a wire-frame type of 3D model is displayed as solid object with opaque surfaces that are coloured and shaded to provide a realistic representation of the model. This capability can have a positive impact on productivity, in that it facilitates visual checking of the general shape of the object that might result in design changes prior to actual manufacturing. The ability to see what an object would look like in a photographic form could conceivably save much time and effort in making pattern

or engineering prototypes. Beneficial as this capability may be, such systems are not providing a true computerised 3D model of the full solid geometry of the object.

Solid modelling can be appropriately defined as a function that creates a 3D picture of an object (with automatic hidden line removal) as well as a full set of 3D data about that object. The 3D information includes dimensions, moments, masses, volumes, surface area and other basic physical properties of a particular object. The ultimate forms of solid modelling systems are those that permit unlimited representations of solid objects, by allowing an arbitrary definition of boundary curves and surfaces (i.e. boundary representation, B-rep) or in terms of primitives of geometry (i.e. constructive solid geometry CSG). These forms of modelling together with Boolean functions allow virtually any engineering shape to be created, manipulated, analysed and visualised (see Figure 1.1)

1.3 AN EXAMPLE OF 3D SOLID MODEL

To illustrate the concept of 3D solid modelling, a simple example is constructed using the solid modelling technique (9). At the basic level of construction, various edges (straight line to circular arc etc) can be constructed one at a time, to form a closed loop to define a 'sheet' consisting of 2 faces (Fig. 1.2a). In Fig. 1.2b a face has been 'lifted' to provide an object with positive volume. At a higher level, holes and islands can be added using the Boolean operations 'SUBTRACT' and 'UNITE' to generate the

more sophisticated geometry which are illustrated in Figure 1.2c to Figure 1.2f.

1.4 FUTURE CAD/CAM SYSTEM BASED ON 3D SOLID GEOMETRIC MODELLER

In future, a geometric model will form the master database representation of a part. This is in marked constrast to the present convention that an engineering drawing being the master representation of a part. The key function of a design office will be to produce a computer geometric model of the part intended to be manufactured. The geometric model will be the database containing all necessary information presently conveyed by the engineering drawing. It can be seen in this perspective that the production of an engineering drawing (from the geometric model) will be a secondary function. All engineering operations downstream from the design office, i.e. process planning, NC programming and machining etc. will take the geometric model as their primary input data. They will access the required information from the geometric model via the enquiring facilities of the geometric modeling system (GMS). Some of these operations will add additional information to the master geometric model and hence produce secondary geometric models. In NC machining, a machining model, MM, may be produced. A possible schema for a general GMS is shown in Figure 1.1

A new generation of industrial geometry system is just emerging. Unlike most of the current commercially available systems, the new systems are based on unambiguous 3D solid geometric modellers and they are potentially capable of supporting current and future

application automatically. The current state of the art can be summarised as follows:

1) A body of solid modelling theory exists, and the software technology for building geometric modelling systems is now reasonably well understood.

2) Interactive graphics technology is adequate for building good human-user interface for solid modellers.

3) There is a body of experience with research and industrial prototype GMSs, but there is no production experience with solid modelling because they are just beginning to be used in industry.

Currently, sophisticated production grade solid modellers exit today. Broadly, these systems, and others already announced or likely to be announced in the next few years representing a technology plateau: they are based on the best technologies of the 1970s. These technologies are now relatively stable, increasingly well understood, and are unlikely to change dramatically (or to be superseded) in the next few years. If solid modelling is to move to a higher technological plateau in the late eighties or early nineties, a number of open issues must be resolved. The most obvious requirement will be the integration of a GMS into a company's manufacturing/management system. Other issues are summarised in the following:

(1) Integration of 3D solid modeller with sculptured surfaces

In order to be able to represent any arbitrary shape, there is a need to incorporate sculptured surface capability into GMS and the ability to deal with 'blends' in solid modellers.

(2) Representation theory and technology

More efficient and flexible algorithms are needed to enable highly efficient GMS to be constructed. Better representation technique for dimensioning and tolerancing and practical algorithms for forms and features classification.

(3) User Interface

More experience with using solid modeller in industrial environment is needed to guide the design of better human-users interfaces for next generation of geometric modeller.

(4) Hardware

Solid modellers pose computational requirements that are heavier and different in character from that of current wire-frame based systems. It is likely that new, special purpose hardware will be needed for achieving high performance in a GMS. Fortunately, many of the required computation can be done in parallel, and therefore the prospects for high speed hardware based on VLSI technique are probable.

(5) System communication

As various system of GMS being developed and multiplied in industry. there will be a need for these systems to communicate with each other. Some excellent initial works have been done to achieve this end, i.e. the IGES and XBF of CAM-I (10, 11). Another related area is the separation of application software development and that of GMS. Again, CAM-I has initiated work in this area in the form of the Application Interface specification (AI) (12).

(6) Application software development

The most important characteristic of solid modellers is their potential capability of supporting automatically a great variety of applications. However, relatively few algorithms are known for driving application software from geometric modellers, and much research is needed to bring to fruition the revolutionary level of automation promised by solid modelling technology.

Perhaps the two most active areas of current research are the automatic generation of meshes for finite element analysis and the automatic generation of NC programs.

1.5 AIMS OF THESIS

A new generation of industrial geometry system is just emerging. Unlike most of the current commercially available system, the new systems are based on unambiguous three dimensional (3D) solid geometric modellers and they are capable of supporting current and future applications automatically.

It is the advent of 3D solid geometric modelling and its impact on the next generation of computer aided machining NC system that this research work attemps to investigate.

This thesis investigates the computer aided manufacturing of geometric models and aims to provide a new link between CAD and CAM.

This thesis intends to research on the following areas:

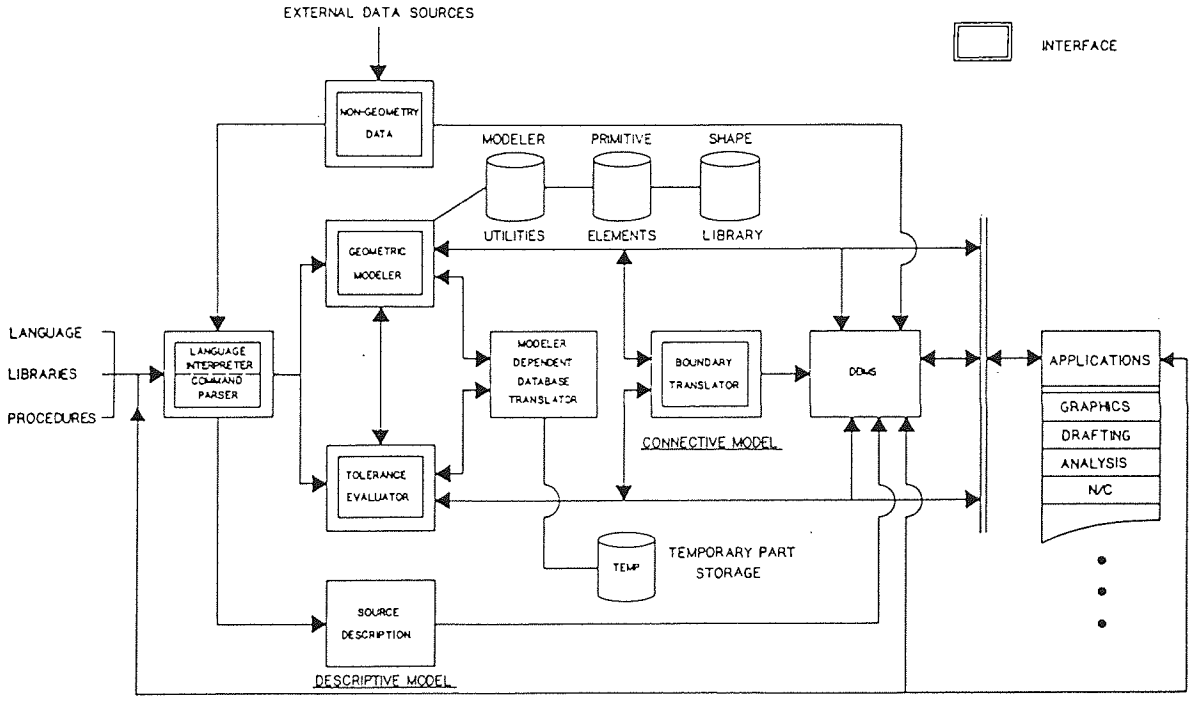
1) To develop the concept and algorithms for the automatic generation of geometric information (i.e. APT geometry required for NC manufacturing applications) directly from geometric models.

2) To develop the theory and algorithms for the interactive generation of a complete APT part program and the automatic generation of cutter path for 3D geometric models.

3) To design and implement an experimental software interface (ROMAPT) between a bounded geometric modeller (ROMULUS) and an unbounded NC processor (APT) based on the above theory.

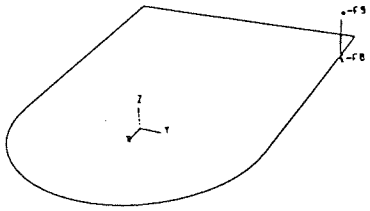
4) To machine an engineering part using ROMAPT.

FIGURE 1 1· A GENERAL GEOMETRIC MODELLING SYSTEM

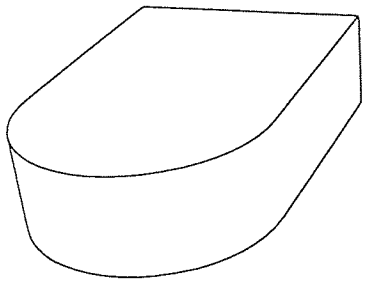


GEOMETRIC MODELING SYSTEM

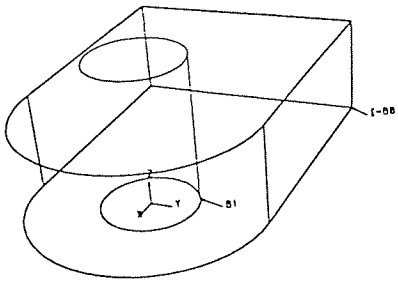
FIGURE 1 2: GENERATION OF A 3D SOLID GEOMTRIC MODEL



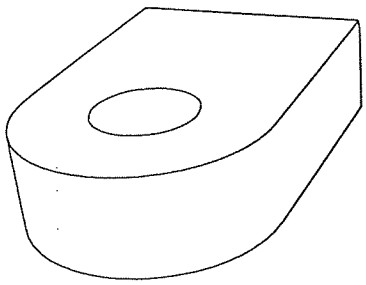
a 'sheet' - defined by a Closed Loop



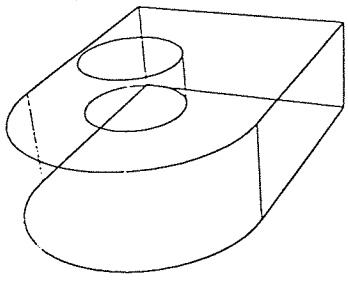
b Generation of a Volume by 'Lifting'



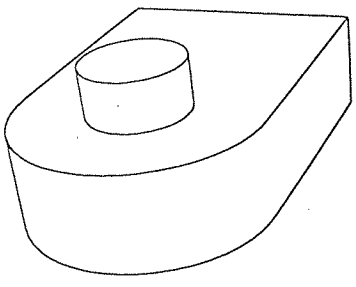
c Wire-frame View of the Original Object B0 and a Cylinder B1



d Hidden Line View of Boolean Operation B0-B1



e Wire-frame View of Boolean Operation B0 + B1



f Hidden Line View of Boolean Operation B0 + B1

2. BACKGROUND OF CAD/CAM AND GEOMETRIC MODELLING DEVELOPMENT

Geometry is central to the design and production of mechanical and other discrete engineering components. With the introduction of mass production and job specialization, engineering drawings were used as the means of specifying geometry between design function and production function.

The widespread use of mini computers, plotters and graphic display units have triggered off the development of computer aided drafting systems in the 1970s. These systems generally improves the quality and productivity of the drafting function. The trend is such that more computers will be introduced to improve the quality and productivity of design and manufacturing functions. This is the beginning of computer integrated manufacturing (CIM). It is envisaged that geometric modelling will be the key element in the development of CIM.

The following describes some of the main stream techniques used in geometric modelling systems.

2.1 WIRE FRAME MODELLING

The wire-frame technique is used by many current commercial CAD systems. These are usually simple 2D systems for designing printed circuit board (PCB), large scale integrated circuit (LSI), 2D

draftings and similar tasks.

These systems store the geometry by simple datastructures consisting lists of points, lines and arcs. The application programs scan through the datastructure to produce photomasks or engineering drawings. These systems usually require only simple analytic geometry, co-ordinate transformations for scaling, rotation, translation and projection.

Due to the popularity of these systems and partly because of the saturation of the 2D market, gradually 3D wire-frame systems were introduced particularly for the mechanical engineering sector. These systems have been found useful because of the following advantages:

1) Model creation is simple and fast.

2) Viewing of model is fast.

3) Geometric entities (points and edges) can be updated quickly.

4) Computer resource (i.e. power and storage) is low.

However, the limitations of the wire frame systems have also become apparent:

1) Ambiguity: The wire frame model is ambiguous.

2) Impossible object: The wire frames system tolerate impossible or meaningless objects.

3) Hidden line removal: The hidden line in a model can only be removed manually and not automatically.

4) Silhouette line: The silhouette line of the model (view dependent) cannot be generated automatically.

5) Sectioning: The sectioning of the model cannot be done automatically.

6) Mass properties: The total surface area and volume of the model cannot be generated automatically.

7) Interference Checking: No automatic interference facility.

The above mentioned deficiencies of the wire frame models is due to their lacking of geometry completeness.

2.2 SURFACE MODELLING

Some of the ambiguities of wire fame models are overcome with surface models by providing more information describing the surface. These can be created by attaching surface elements to the

edge frame.

The most common type of surface models are:

1) Plane: The model is a flat plane created between two parallel straight lines.

2) Tabulated cylinder: This model is a curved plane between two arbitrary parallel curves.

3) Ruled surface: This model is the interpolation of two curves.

4) Surface of revolution: This model is created by revolting and arbitrary curve through an arc about an axis.

5) Sweep surface: This model is created by sweeping an arbitray curve through another arbitrary arc.

6) Fillet surface: This model is a cylindrical face joining two other adjacent surface in a continuous manner.

7) Sculptured surface: This model is a general scheme to represent complex sculptured surface.

2.2.1 SCULPTURED SURFACE MODELLING

The sculptured surface modelling evolved from the work of Ferguson , Coons, Bezier, Gordon and Riesenfeld (13 - 16). This Technology was initially developed to replace the lofting process used in the design of sculptured objects (turbine blades, ship's hull, aircraft and car bodies etc.).

One of the earliest technique was that of the Ferguson patch in 1963 (13). This technique used parametric rather than cartesian co-ordinates in its curve and surface representation. This allowed the surface to be defined mathematically and to be displayed and transformed easily by the computer via matrix algebra.

Later in 1964, Coons (14) of MIT developed a very general theory of surface patches (Coon's patch). Coons showed how four arbitrary boundary curves can be blended into a single smooth path and that inter patch continuity of both gradient and curvature may be achieved.

Although Coon's work was very significant, it was difficult for non-mathematician to use easily. In 1971, Bezier introduced his UNISURF (15) system which was based on Ferguson's work. It is a practical system which allows curves and surface to be designed easily by non-mathematical minded users.

By 1974, Gordon and Riesenfeld started to exploit the properties of B-spline for use in surface design, particularly the facility for purely local modification of curves and surfaces without affecting other existing curves and surfaces.

Sculptured surface work has been concerned primarily with interpolation and approximation and has strong roots in the mathematical theory of spline and numerical analysis. However, the development of sculptured surface theory has had remarkably little effect on the development of solid modelling theory, algorithms and systems. But this is envisaged to be changed very soon with the converging of the two powerful technologies.

Aparting from the advantages of fast model creation, fast editing and modification of surface and small data storage requirement, there are still a number of disadvantages for surface modellers. These disadvantages are:

1) No connectivity information between surfaces thus results in ambiguities.

2) Mass properties calculation are limited to single surface.

3) Calculation of intersection between sculptured surfaces is a complex problem.

4) User is expected to detect interference between surface models.

5) Machining of sculptured surface is still under development.

2.3 SOLID MODELLING

To overcome the majority of the deficiencies mentioned above, the solid modelling technique was developed using unambiguous representation of solids.

The following traces a brief history of solid modelling development. In Britain Dr Ian Braid of the Cambridge University CAD group developed the BUILD-1 (2) system in 1973 and BUILD-2 (3, 4) in 1978. The BUILD group was continued in the Engineering Department of Cambridge University (under the leadership of Dr Graham Jared). Dr Ian Braid joined Shape Data Limited and have contributed to the development and launching of the ROMULUS geometric moodeller (9) in 1978. The other significant developments include the Geometric Modelling Project at Leeds University (8) and the MEDUSA system (17) of Cambridge Interactive System (CIS).

the U.S.A. Baumgart's (18) idea of applying Euler operator and wing-edge data structure has significantly influenced the development of geometric modelling including Eastman's GLIDE-1 (19) of Carneigie Mellon University and BUILD-2 of Cambridge University. In 1972, the Rochester University developed the PADL-1 and later PADL2 and influenced the development of GMSOLID. In 1976, CAM-I launched its Geometric Modelling Project (20).

In Japan, Hokkaido University started TIPS-1 (7) and Tokyo University launched GEOMAP (21) in 1973. TIPS was further

developed by Cornell University under a CAM-I contract.

In Europe, the COMPAC (22) system was developed by the Berlin Technical University in 1969. This was followed by the development of the PROREN (23) system at the Ruhr University and the launching of EUCLID (24) in France and EUKLID (25) in Switzerland.

In the commercial field, developments have been rapid. In 1980, Evan and Sunderland started marketing ROMULUS. This is followed by many geometric modelling product announcements by CAD/CAM vendors like APPLICON (based on Synthavision) and Computervision (Solid Design) etc.

Currently, there are three major classes of solid models: idealized, approximate and complete models. They are described in the following.

2.3.1 IDEALIZED MODEL

An idelaized model is a form of solid object representation which does not explicitly contain all boundary geometric entities (i.e. not completely evaluated). The model contains the necessary information to evaluate all required geometry. Both sheet model (for sheet metal work) and graph model (for piping system) are examples of this type of model.

A sheet model represents a solid object by a set of single surfaces with associated surface thickness. A graph model represents a piping system by the description of centre-lines of pipes and associated cross-sections of pipes. Presently, only experimental systems are implemented for the idealized model.

2.3.2 APPROXIMATE MODEL (FACETTED MODEL)

The approximate model (facetted model) uses polygonal schemes to approxiamte solid object with ploy hedra having hundreds or thousands of planar polygonal faces. Within limits, the user generally has some control over the accuracy of the approximate representation by specifying the number of facets used.

This type of model contains only planar faces, hence all spatial sorting calculations involving face and edge comparisions are made very simple (by solving simple linear equations).

If only few faces are used to model an object, the data storage requirement is low and the hidden line removal operation is extremely fast. This is advantageous for a quick visualization of the object but the resultant picture of the object can be crude or even misleading. Mass properties calculations may be similarly highly inaccurate. If a very large number of faces are used to represent the object, the accuracy of the model would be within acceptable limits (but still not as good as that of the complete

models). But this would require very large data storage and will slow down drastically picture generation and manipulation.

Generally, facetted modellers do not support adjacent relationships of vertices, edges and faces. Consequently, local modifications (i.e. tweaking etc.) are not possible without full remodelling. Another advantage of facetted modellers is that they can approximate any surface shape including sculptured surfaces.

Because of the relative simplicity of approximate models, many CAD/CAM vendors offer modellers based on this technique. However, for application that require high accuracy, for example NC machining, a complete representation of the solid is prefered.

2.3.3 COMPLETE MODEL

A complete model contains all the boundary surface of the solid object. The model is made up of faces. Faces meet in edges which are located between vertices. Faces can be plane, concave or convex and could have holes in them. Each edge has two vertices and meets only two adjacent faces. There are two main classes of complete model: Boundary Representation (B-rep) and Constructive Solid Geometry (CSG) models.

B-rep models contain within the database the geometric entities (points, curves and surfaces) and the topology elements (vertices, edges and faces). The topology relates various geometries

together. All geometric information in B-rep are evaluated.

CSG models represent solid objects in terms of Boolean algebra (i.e. union, intersection and difference etc.) operated on volume elements (primitives). CSG models are unevaluated trees.

The complete modelling is the most advanced represenation of solid object. A complete and unambiguous description of the object is stored within the computer. Many deficiencies of the previous modelling technique have been eliminated. With the geometric completeness of this type of model, automation of various applications based on this techinque are possible. The main advantages of complete model are given in the following:

1) No ambiguity.

2) Automatic hidden line removal and sectioning.

3) Automatic calculation of mass properties.

4) Automatic interference detection.

5) Considerable future potential for automation of applications.

Untill recently solid modellers require large main-frame computers to run. Their run time are lengthy and expensive and hence their use to industry were limited. The main disadvantages of complete

modellers are as in the following:

1) High demand on computer resource.

2) Geometric coverage does not yet cover sculptured surface.

3) Medium to very large data storage is required.

4) Application software based on complete modelling have not been fully developed.

However, progress in computer technology, in both hardware (cheaper and faster) and software (robust and greater functionality) are rapid. It is now possible to implement solid modelling system on some of the relatively inexpensive 32-bit mini-computer available today. With so many advantages, many CAD/CAM ventors are now offering modelling options to their systems.

However, much remain to be done. Better and efficient algorithms are required for model display and manipulation (particularly for hidden line removal). More application software based on complete models. The incorporation of sculptured surface into solid modelling.

The following concentrates on CSG and B-rep models.

2.4 CSG MODELS

This is a building block approach to model solid object. The idea is that large and complicated solid object can be built up by some ordered additions and subtractions of simple primitive volume elements (i.e. cube, cylinder, sphere and cone etc.) using Boolean operations.

The primitive volume can be made up of unbounded half-space. For example, a cube is made up of 6 planar half-spaces. A planar half-space is the infinite plane defined by:

$$ax + by + cz + d = 0$$

In general, a 3D half-space is a set:

$$\{ p : f (p < 0) \}$$

where,

p: is a point in Euclidean 3D space.

f = 0 : defines a surface.

CSG models can be described as a binary tree and has the advantage of providing a compact model description in the computer. However, there are also disadvantages for CSGs:

1) To represent complicated engineering part, hundreds of primitives may be required, thus making the input process very lengthy.

2) The datastructure is unevaluated. The relationships between vertices, edges and faces and their adjacency are not stored explicitly. Hence, the model does require excessive computing to evaluate the faces and edges for displaying or mass properties calculations. It is not efficient for application like NC which require direct access to edges and faces.

3) Direct operation like 'tweaking' on the model is not always possible. Local modifications (i.e. adding fillet, chambers etc.) are not possible without remodelling.

To eliminate some of the above disadvantages, the current trend is such that CSGs will also provide boundary description.

2.5 BOUNDARY REPRESENTATION

A boundary model store all boundary geometry explicitly and completely. The model of a solid object is a single region in 3D space in which all points are connected through the solid. The

model is bounded by faces, these being bounded by edges. A complete representation of such a model can be made up of its topology and geometry. The topology of the model defines the relationship between vertices, edges and faces in terms of their connectivity. The geometry of the model defines the definite shape of the object. The geometry is made up of points (referenced by vertices), curves (referenced by edges) and surfaces (referenced by faces).

There are two methods of constructing boundary models. The first method is an extension of wireframe and surface models into solid model. The user is required to add faces and their topology to the model interactively. The advantage is that this method provide an upgrade route for those were already committed to wireframe systems. But the manual conversion process may be complicated by the introduction of erroneous models.

The second method is the sweeping of 2D profiles along linear or circular arc to generate a solid model. As the process is automatic, this provides a very reliable way of modelling solids.

In practice, many commercial modellers using a combination of modelling techniques (i.e. B-rep and CSG) to model solid.

Alternatively. a boundary model of a solid object can be regarded as essentially a graph of vertices and edges embedded in a surface topologically equivalent to a sphere. A FACE is made up of a set

of edges forming in a continuous, non-self intersecting path on a bounded region of a continuous and non-self intersecting surface. An EDGE joins two VERTICES. A set of boundary edges is called a LOOP. The complete set of faces defining one complete surface of an object is called a SHELL. An object is made up of shells. To complete the boundary representation, spatial geometries of point, curves and surfaces are attached to the topology of vertices, edges and faces via links (i.e. pointers).

The geometry and topology must be well formed and consistent and that the object must be non-self intersecting. In addition, the topology is required to conformed to Euler-Poincare's (4) equation:

$$V - E + 2F - L = 2 (S - H)$$

where,

V: number of vertex

E: number of edge

F: number of face

L: number of loops

S: number of shells

H: number of through holes.

Because of the evaluated representation of boundary models, It facilitates a powerful set of 'tweaking' operations (any face can be selected, moved, rotated or draft angle, fillet and chambers added etc.). As the datastructure allows direct access to all the topological and geometrical entities, it lends itself for application like NC machining which often require direct access to face, edge and their adjacencies. For the above reasons mentioned, this research concentrates on boundary (graph-based) model representation.

2.5.1 TOPOLOGICAL RELATIONSHIP BOUNDARY MODEL

The relationship between topological entities of boundary model may be described (in term of graph theory) in the following.

1) There is a vertex at every join of edges.

2) An edge has two vertices (i.e. a start vertex and an end vertex) which define its bounds.

3) An edge lies common between two adjacent faces. (It always lies in at least one face but not more than two faces.)

4) A loop (i.e. a circuit in graph theory) is an ordered sequence of edges that formed a closed path.

5) A face that defines an open surface is always bounded by at least one loop.

The following discusses the trade-off between data storage and computing speed.

2.5.2 A SIMPLE DATA STRUCTURE FOR BOUNDARY MODEL

A simple datastructure (Figure 2.1) which stores minimum data to represent a boundary model requires only the following:

1) For each object, a list of the shells of the object.

2) For each shell, a list of the faces of the shell.

3) For each face, a list of the edges in the face.

4) For each edge, a list of the vertices (i.e. 2) of the edge.

5) The corresponding links to geometry (i.e. point, curve, surface) for corresponding topological element (i.e. vertices, edges, faces).

Although that data storage requirements are small, but the algoritms required for scanning the datastructure (say to reach every entity) can be time consuming.

2.5.3 ALTERNATIVE DATA STRUCTURE FOR BOUNDAPY MODEL

To have a faster scanning algorithms, an alternative datastructure for the boundary model which store all necessary pointers (forward and backward) is given in the following (Figure 2.2):

1) For each object, a list of the shells.

2) For each shell, a list of the faces and the owning object.

3) For each face, a list of the edges and the owning shell.

4) For each edge, a list of the vertices and the face in which the edge belongs.

5) For each vertex, a list of the face meeting at the vertex.

6) The corresponding links to geometry for faces, edges and vertices.

Using this datastructure, a more efficient scanning algorithm (i.e. to reach every entities) is possible but at the expense of more storage.

2.5.4 A BALANCED DATA STRUCTURE FOR BOUNDARY MODEL

The various trade-offs of computing speed, data storage and program development time can be described as (26):

MODEL = DATA + STRUCTURE + ALGORITHM

For different applications, the datastructure can be modified accordingly to suit the corresponding requirements of the application. To support application program like automatic generation of machining data for NC manufacturing, a balanced datastructure (see Figure 2.3) can be constructed of the following:

1) An object has an unordered list of shells.

2) A shell has an unordered list of faces.

3) A face has an unordered list of loops.

4) A loop has an ordered list of connected edges (all belong to a single face) and references the owning face.

5) An edge has an ordered list of its vertices (maximum 2) and an ordered list of loops (maximum 2) in which it lies.

6) An vertex has an ordered list of edges which meet at the vertex.

7) The corresponding links to geometry for faces, edges and

vertices.

This datastructure minimizes the various searching time required for the most used topological items (like face, loop, edge and vertex) for NC application tasks, for example:

a) Given a face, what are the adjacent faces, so that check surfaces may be obtained.

b) Given one boundary edge of a face, what are the other boundary edge in order, so that a cutter may be driven round a profile.

c) Given a face, what is the minimum gap between loops to avoid cutter collision.

This datastructure also minimizes the data storage for other items (i.e. object, shell etc.) at the expense of searching time.

On the whole, this is a well balanced datastructure.

2.6 COMMUNICATION BETWEEN MODELLERS AND APPLICATION PROCESSORS

Due to the diverse developments and applications of various geometric modeller systems, there arises a need for modellers to communicate with each other, and with application processors like NC.

2.6.1 CAM-I EXPERIMENTAL BOUNDARY FILE (XBF)

The basic requirement for a model representation to be used for archiving and communication purpose is that a model in another form can be translated into this representation and back again without essential loss or change of information.

It is essentially a static form of communication between different model representation. During the translation, the geometrical information of the object models is unaltered with only its representation changed to a standard form.

This concept is used by the CAM-I Experimantal Boundary File Representation (XBF) which is based on the ANSI Y14.26M standard (27) and the IGES standard (28) for transmission of graphical data.

2.6.2 CAM-I APPLICATION INTERFACE (AI)

Until recently, there is no standardization that allows direct communication between different modelllers and application programs requiring modelling services.

As differing application requirement imposes different access path which affects the design the datastructure and algorithm for efficient programming development and execution, the standardization of application programs' internal model representation is undesirable.

A practical approach is a dynamic means of communication between modellers and application programs, which would make modelling services of an underlying modeller available to and yet transparent to any application program which requires them. This is the concept of the CAM-I Application Interface (AI).

The AI is designed to allow modeller independent communication between an application program and a modeller. The AI effectively shields application program from an underlying geometric modeller.

The AI is a set of entry points (callable FORTRAN subroutines) in the geometric modeller that allows the application program to access and manipualate the modeller and obtain information about the model (i.e. the creation, modification and interrogation of the model). The flow of data through the AI entry point arguments include not only boundary geometry but boundary topology and CSG information as well.

The data flow corresponds to that between a nominal geometric model and an application program. The AI maps the datastructure of the underlying geometric model to that of the nominal model (Figure 2.4).

The use of AI in an application program requiring direct access to the underlying modeller may not be desirable or possible. An alternative is to use the CAM-I XBF which may be used by the application program without accessing the modeller directly.

The XBF can be constructed by a translating processor via calls to AI accessing an internal model. Similarly, another translator can be used to read an XBF file and translates it into an internal model representation via calls to AI.

Both the AI and XBF together with the two translators have been partially implemented and tested in the Lucas Research Centre.

2.7. MANUFACTURING AND NC APPLICATIONS

Computer aided manufacturing (CAM) has a longer history than computer aided design (CAD) and geometric modelling. The following outlines some of the recent developments.

The development of manufacturing techniques is characterised by the trend to a higher productivity along with an increase of flexibility. In mechanical engineering industry, this higher level of productivity is to be achieved by an almost constant number of employees. Hence, there is an urgent need for automation.

Numerical Control (NC) is an important element of this automation. NC is the directing of a machine tool process through the execution of a preceded instruction sequence. The technique represented perhaps the most significant modern advancement in the manufacturing industry. It is most widely applied to

metal-removing machining operations such as milling, drilling, boring and turning. It is also being used in an increasing variety of applications which include welding, automatic drafting, electronic component placement and wiring.

Each NC instruction sequence represents a detailed, step-by-step, set of codes of controlling a particular action of a machine tool. Generation of code at this level is often too complex to be done routinely by hand especially for complicated parts. Hence, a number of NC processors, among them, APT, have been implemented and are in widely use.

A NC processor is a computer application program which accepts as input user-oriented language statements that describe the NC operations to be performed. Commonality among many types of NC devices is obtained by designing NC processors to produce common interface code termed Cutter Location Data (CLDATA) or file (CLFILE). The CLDATA, in turn, is used as input to another computer application program called a post-processor which produces the code for the particular NC device used. This output is normally in the form of punched tape (NC tape) for convenient storage or reading by the device's controller during the execution of program.

The advantages of NC machine tools that represents an effective means for rationalisation and automation of the manufacturing process are recognised and utilised by industry. With the use of NC, the machining process is preceded by process planning activity

to ensure an optimal interaction between the machine tool, tools and the workpiece for the accomplishment of the manufacturing task.

2.8 APT

APT is the acronym for automatically programmed tools. It denotes a language and also a computer program to process statements written in that language. The result of such processing is a NC control tape for automatic directing of a NC machine tool. APT was originally developed jointly by the Massachusetts Institute of Technology (MIT) and member companies of the Aerospace Industries Association of America (AIAA). A version called APT3 was completed by this group in 1961. At that time, the AIAA estabilised the APT Long Range Program in order to continue the development of APT to meet current and future needs of the users, and to extend the benefits of APT to other industries.

The widespread acceptance of APT language for use in NC processing (especially in U.S.A.) has resulted in a need for APT processors for a wide variety of computing equipment. Among many computer manufacturers, IBM took up APT3 and further developed the system and designated it as APT-AC (29). The other relatives of APT includes EXAPT, ADAPT, NELAPT and UNIAPT etc.

In 1964, a system regorisation study was initiated by the APT Project. This resulted in the specification of a completely new

system design which allows computer manufacturers the full range of choice over the mode of implementation they desire while at the same time minimising duplication of effort. The design was subsequently implemented and designated the APT4 system (30).

Recently, CAM-I has further developed the APT4 system to include sculpture surface machining capability (and designated as SSR1) through its Sculptured Surface Project (SSP) (31).

2.8.1 AN APT PART PROGRAM EXAMPLE

The APT part prgram consists of usually 5 sections: program indentification, geometry, machine description, motion and end.

The following is an example of an APT part program showing the various sections (Figure 2.5).

1) Program indentification

The program indentification section consists usually the following statements:

```
PARTNO EXAMPLE

CLPRNT
```

```
MACHIN/ PLOTT, 1

MACHIN/ CINTA5, 4, OPTION, 2

UNITS/ IN

INTOL/ 0.001

INTOL/ 0.001
```

2) Geometry section

The following is the geometry section containing APT geometry statements:

```
PST = POINT / -4, 4, 0

P1 = POINT / 0,0,0

P2= POINT / 1,0,0

P3 = POINT / 1,1,0

P4 = POINT / 0,1,0

L1 = LINE / P4, P1
```

L2 = LINE / P1, P2

L3 = LINE / P2, P3

L4 = LINE / P3, P4

3) Machine description

The machine description section usually consists of the following:

LOADTL / 1

CUTTER / 2.0

COOLNT / ON

SPINDL / 150

FERDAT / 100

4) Motion section

The motion section consists of statements controlling the motion of the tool:

FROM / PST

GO / TO, L4, TO, PS, TO, L1

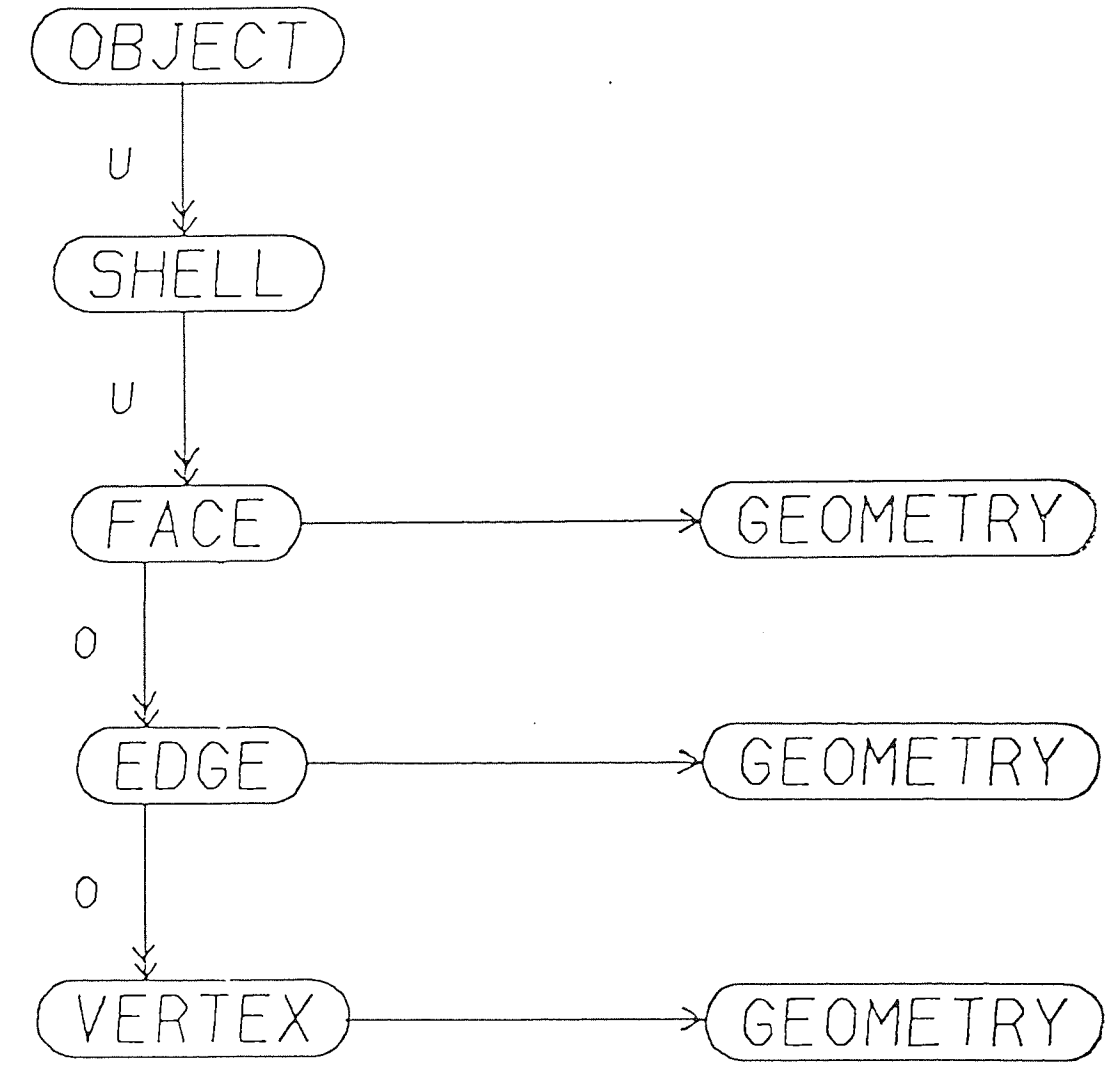
```
TLRGT, GORGT / L1, PAST, L2

TLRGT, GOLFT/ L2, PAST, L3

TLRGT, GOLFT/ L3, PAST, L4

TLRGT, GOLFT/ L4, PAST, L1

GO TO / PST
```

5) End section

The end section consists of the following:

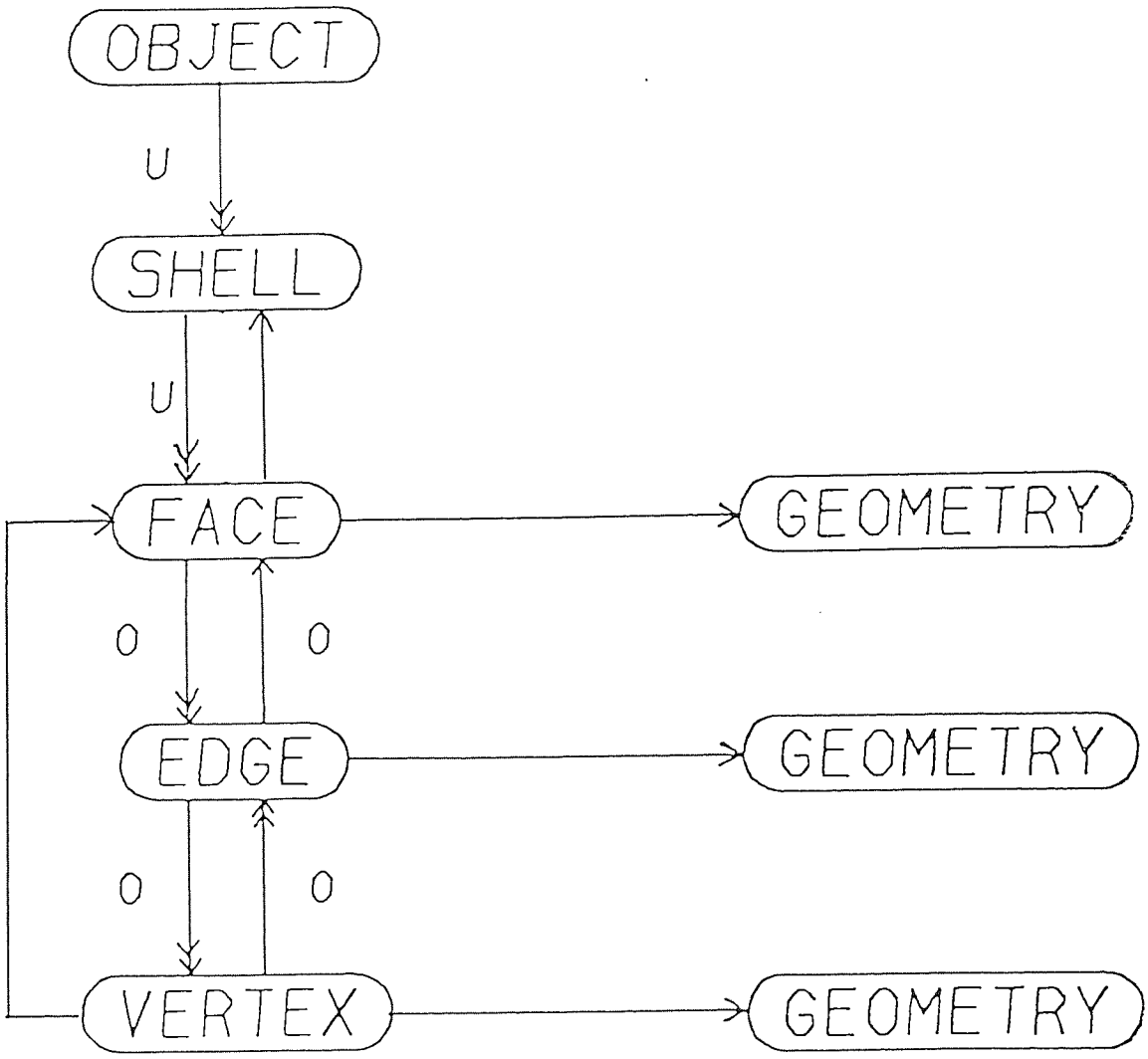
```
COLLNT / OFF

SPINDL / OFF

END

FINI
```

FIGURE 2.1: Minimum data boundary model



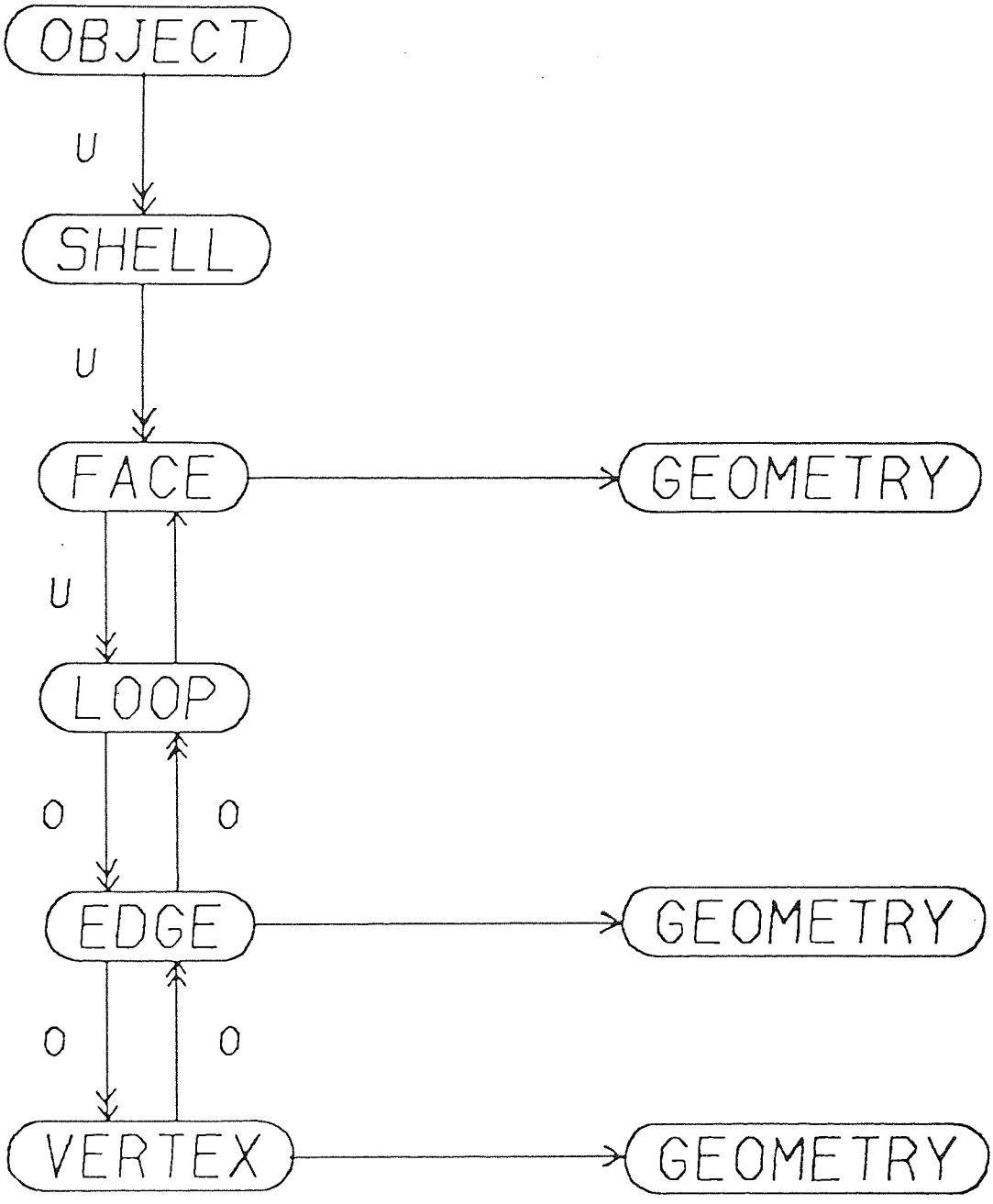
O = ORDERED U = UNORDERED

FIGURE 2.2 Boundary model with more data



O = ORDERED U = UNORDERED

FIGURE 2.3: Balanced boundary model



O = ORDERED U = UNORDERED

FIGURE 2.4 Communication between a nominal geometric model and an application program

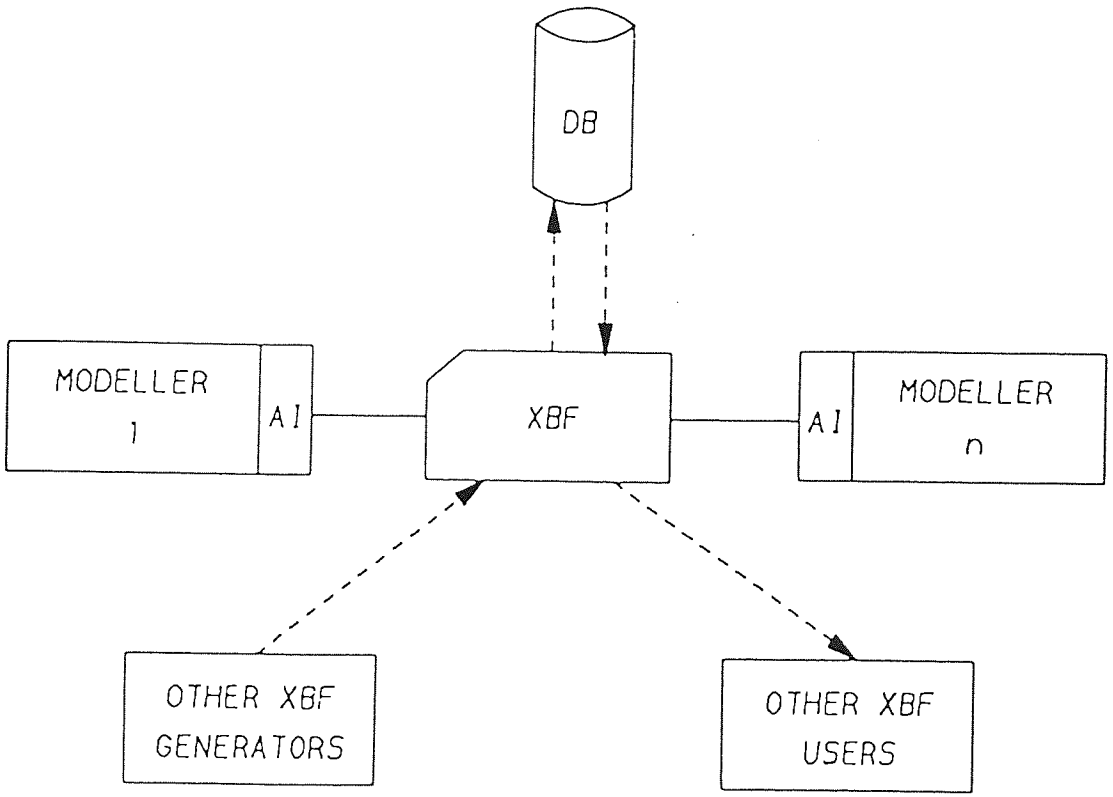
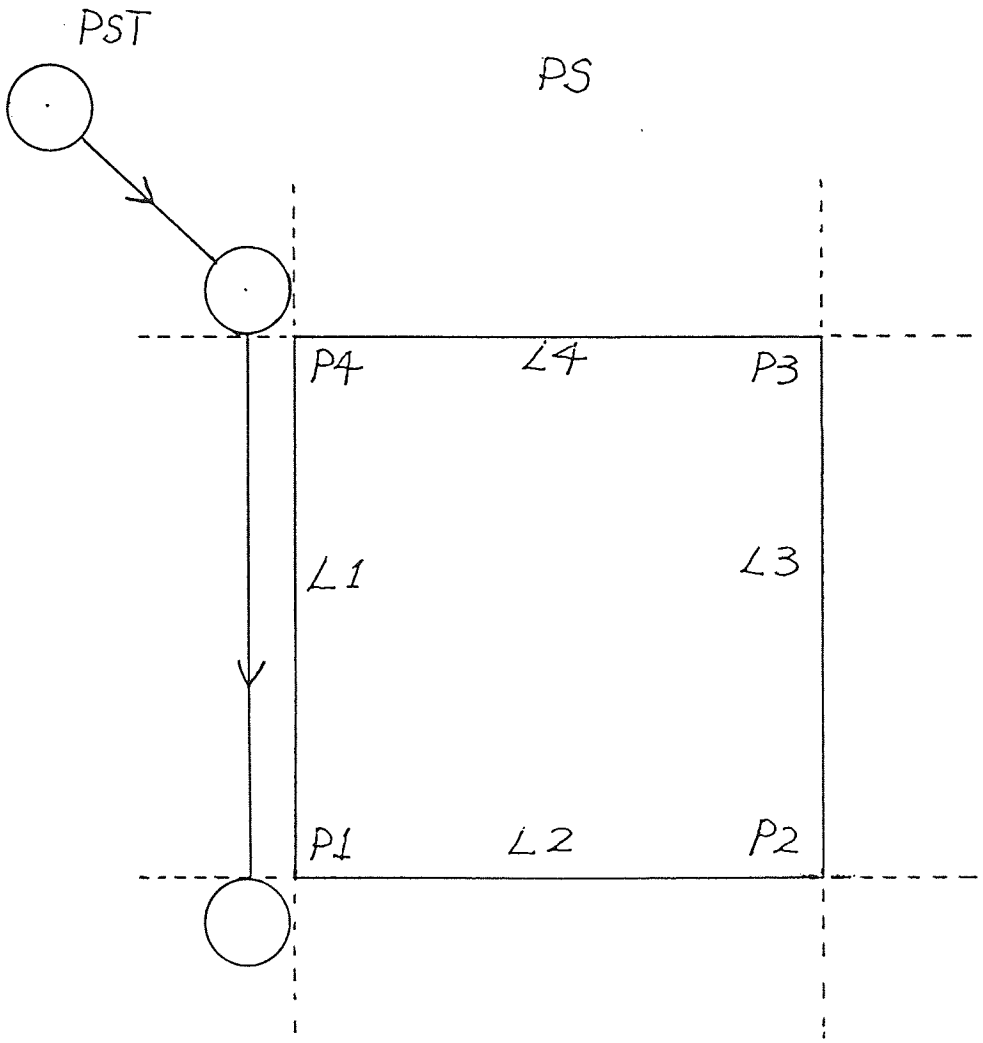


FIGURE 2.5 Simple contouring example



3. THE LUCAS SET UP

Lucas Industries Limited is a large British based company with annual sales of approximately 1,200 million pounds and employing 47,000 people worldwide. The main business of the company is supplying components to car and truck manufactuers and to the aerospace industry. Lucas Industries is organized into few main product groups each having its own R & D facilities. There is also a corporate Research Centre which acts as a centre of expertise for the whole Lucas group.

There is an Engineering Computing Group (of which the author is a member) based at the Lucas Research Centre. This group has had many years experience in developing CAD/CAM systems for use both within and without the Lucas group. These systems are developed and tested on the Research Centre's PRIME 550 II computer and then transferred to the Lucas group's IBM (370, 3033) computers for general usage throughout the company.

The Engineering Computing Group have developed an extremely comprehensive finite element analysis system which includes stress, thermal, dynamic and electro-magnetic analysis. This is marketed under the trade name FELSET (32).

Another suite of programs is marketed under the trade name SURFSET (33). This is a sculptured surface design suite and has been used to design a wide range of articles from shoes, bottles to engine

housings. An interface has been developed between the design suite and NC machining via APT (by the author). A further interface has been developed between the design and analysis systems enabling the complete automated analysis and machining of sculptured parts. A new version of the design suite (PROTEUS), based on B-spine surface representation is just completed.

For the large class of engineering components which are made up of simple analytic geometric entities such as cubes, cylinders, cones and spheres etc. can only be modelled by complex surface representation technology in an inefficient and limited manner at present. These components are to be handled by off-the-shell commercial CAD/CAM systems based on wire-frame technology (i.e. Computervision CADDS, IBM CADAM etc).

In future, It is envisaged that CAD/CAM systems based on solid geometric modelling will be used. Hence, the current research work in the CAD/CAM area in the Lucas Research Centre will concentrate on the application of solid modelling technique.

The Research Centre has just completed a contract from CAM-I to investigate the Geometric Modelling Project's Experimental Boundary File and the Application Interface (34).

Lucas is a member of the CAM-I organization and is sponsors of three CAM-I projects (the Geometric Modelling Project, the Sculptured Surface Project and the Framework Project), and has a vested interest in CAD/CAM.

The following describes the various CAD/CAM facilities and their applications in a manufacturing environment within Lucas (which is very typical of a large manufacturing company).

3.1 COMPUTER FACILITIES

The various computing facilities available in the Lucas Research Centre for CAD/CAM research and development are as in the following:

1) COMPUTER : PRIME 550 II

2) MAIN MEMORY: 2 MB

3) DISK : 2 X 80 MB disk drive

4) MAGNETIC TAPE UNIT: 1 unit (800 b.p.i 9 tracks)

5) GRAPHIC TERMINAL: 3 X TEKTRONIC 4010, 2 X TEKTRONIC 4010, 1 X ADM 5.

6) PLOTTER: 1 X TEKTRONIC 4611, 1 X BENSION

7) COMMUNICATION LINK: HASP link to IBM 370, 3303 to run APT-AC or SSR1 NC processors.

8) CAD PACKAGES: SURFSET, PROTEUS, FELSET.

9) DRAFTING SYSTEM: MEDUSA.

10) GEOMETRIC MODELLER: ROMULUS, MEDUSA, TIPS

11) OTHER CAD SYSTEMS IN LUCAS: COMPUTERVISION CADDS, IBM CADAM.

For details, see Appendix 1.

3.2 SURFSET SYSTEM

Since 1970 the Engineering Computing Group has been involved in the research and development of a comprehensive CAD/CAM package now known as SURFSET suite of programs. The initial aim of the project was to provide a complete CAD/CAM system for the design and manufacturing of automative small lamps. During the development of the system, a modular approach was taken to allow future enchancement and additional software to be added easily.

The system is based on Coons and Bezier patches. A new version based on B-spline has also been developed. The system has been later modified to allow a general design and machining capability for doubly curved objects including the followings:

1) Automotive small lamps.

2) Automatic head lamps.

3) Aircraft engine fairings.

4) Turbine blades.

5) Bottles.

A pilot system (CADAPT) has also been developed to link the SURFSET CAD system to the APT-AC system for tooling applications. The objective is that Bezier patches used in SURFSET by the designer can be made available to an APT part programmer, and used directly in the APT part program.

3.3 APT FACILITIES

APT-AC (Advanced Countouring (29)) and SSR1 (Sculptured Surface Relese 1 of CAM-I (31)) NC processors are available in the Lucas Group Computing (LGC) Centre's IBM 3033 computer.

The processing in APT is essentially consisting of translating the input to internal representation, calculating the points on the actual cutter path and then the postprocessor producing the paper tape required for a particular controller of a machine tool.

The APT language consists basically of geometry, machining and post-processor words (see Section 2.8.1).

The geometry statement:

P1 = POINT / x, y, z

defines a point P1, in Cartesian co-ordinates.

The machining statement:

GO TO / P1

instructs the cutter to move to the point P1.

The post-processor statements:

COOLNT / ON

switches on the coolant for the machine.

FEDRAT / 300

sets the feedrate of the machine.

The post-processor statements are only acted on in the post-processor. They are checked generally for syntax initially in

APT and are passed on to the post-processor to be checked for validity for the particular machine.

APT is a very powerful language and is being used successfully by many Lucas NC departments. APT is particularly suited for task required long and complex NC programming.

Sometimes quite complex calculations can be run using the geometry part of APT. APT has many definitions for such thing as points, lines, circles and conics. The APT geometry can be used by engineers (even with very little training in APT) for design calculation. Moreover, APT can be also of use for new machines without post-processors. This is done by taking the output from the calculation section of APT and doing the rest of the part programming manually would still be worthwhile in many cases.

A very useful aid in checking a part program is to get a graphical plot of the cutter path. This is obtained from APT simply by the addition of few post-processor words to the original program:

MACHIN / PLOTPP, 1

at the beginning of the part program.

MACHIN / ON, XYZ, 1

at the beginning of the machine statements.

A separate post-processor is available for generating such plots.

3.3.1 APT-AC SYSTEM

Due to the popularity of the IBM APT-AC systems particularly in U.S.A., it has become the defacto industrial standard.

The APT-AC system is organized in five sections (Figure 3.1):

1) Control section

This section supervises the flow of the part program through the various sections of the NC processor. It carries out all input and output functions. When a section has completed its task, it returns control to this section.

2) Translation section

This section converts the input APT part program into an intermediate represenatation file (PROFIL) which forms the input to other sections of APT. It translates the programmer's description of the geoemtry of the part into an internal compact (canonical) form.

3) Calculation section

This section takes the PROFIL information and calculates the

successive cutter locations (which define the part). These cutter locations together with post-processor commands are stored in a file called CLDATA (cutter location file).

4) Edit section

This section transforms, prints or plots the CLDATA according to the programmer's requirement.

5) Post-processor

This section selects and executes the post-processor specified by the program. The post-processor takes the CLDATA as input and generates machine tool readable instructions (i.e. G79 for milling, G81 for drilling, G02 for circular interpolation etc. for A Cintimatic 3VT milling machine with Acromatic 5 controller). When all post-processing task is completed, this section returns control to the Control section.

Standard IBM APT is not as good at handling lathes for turning applications with the efficiency it handles drilling and milling applications. Accordingly, a set of APT macro (machining sequence) have been developed for use in a lathe APT part program. Using these macros, it is possible to describe the desired shape and the initial shape as input, the system will generate a number of cuts to achieve the desired shape.

3.3.2 APT4 SYSTEM (SSR1)

An APT4 processor is an APT computing system based on the APT3 (see Section 2.8) system but further developed by the APT Long Range Program.

Logically, the APT4 (30) system is similar to APT-AC except it does not have an explicit control section. The flow of the system is controlled by a master program called Supervisor. The APT4 systems consists of the following (Figure 3.2):

1) Translator

It is a table driven program which translate the input APT language into an intermediate language (IL). The output is stored in a file called ELFILE.

2) Execution Complex

It takes the ELFILE as input and selects and executes the necessary geometry, motion and other APT routines from the Subroutine Library. The output is stored in a file called EXFILE.

3) Subroutine library

It contains all computer independent standard FORTRAN 4 programs of APT4.

4) CL Editor

It takes the EXFILE and produces a general cutter location file (CLFILE).

5) Post-processing

The CL Editor selects the post-processor specified by the program. The post-processor then converts the CLFILE into a machine readable instructions.

6) Load complex

This program initializes the APT4 tables used by the Translator.

Due to the interest of aerospace and automobile industries, the CAM-I Sculptured Surface Project has charged itself with the task to provide sculptured surface machining capability in APT4. The new system is designated as SSR1 and is available through CAM-I.

3.4 CADAPT SYSTEM

This is a link between the SURFSET CAD system and the IBM APT-AC system to provide NC facilities for sculptured surface machining. This is necessary because the ouput from the SURFSET system is not

compatible with that of the SSR1 system. The objective of this system is that surfaces (Bezier or B-spine) used in the SURFSET system by the designer can be made available to an APT part programmer who can then used the surfaces directly in the APT part program.

3.5 NC MACHINES

For milling applications, the Lucas Research Centre has designed and developed a light-weight NC 3-axis milling machine (The Lucas Model Making Machine) intended for machining soft material such as ploy urethane foam. This machine was orginally constructed for the purpose of providing visualization aids for the SURFSET CAD system. The machine can be used to produce models of components rapidly.

In the longer term, the Research Centre will acquire an industrial grade 3 axis NC milling machine for development work. In the meantime, prototype components are to be machined either using the Cintimatic 3 VT milling machine with Acramatic 5 contoller at Aston University or at a suitable machine in a Lucas operating company.

It is envisaged that with some modifications, the system developed for milling applications would be applicable for turning. For turning applications, there is an INDEX G30/150 automatic lathe with Plessey 7360 control system available if required.

3.6 CAD/CAM SYSTEM

For evaluation purposes. the Lucas Research Centre has acquired the ROMULUS and TIPS geometric modellers.

The MEDUSA drafting and modelling system (2 stations) is also installed by the Research Centre in the Drawing Office.

Both Computervision CADDS and IBM CADAM syetems and others are used throughout the Lucas Group's operating companies.

3.7 COMPUTER INTEGRATED MANUFACTURING (CIM)

The primary aim of large scale CAD/CAM research and development particularly in industry is at present the integration of the various processes between receiving an order to the dispatching of the goods, i.e. the approach of Computer integrated manufacturing (CIM) (Figure 3.3).

The geometric modeller is the key element to this approach. The geometric model is the centre of the database consisting not only the geometric information but many more (i.e. tolerance, material, surface finishes, machining data, product or even customer information etc.) for all manufacturing functions from marketing,

production to distribution.

3.8 GEOMETRIC MODELLING IN A MANUFACTURING ENVIRONMENT

After an initial study of Faux's work on users' needs on geometric modellers (35) and a feasibility study, the following users' requirements for CAD/CAM systems based on geometric modelling technology in an manufacturing environment are indentified.

3.8.1 REQUIREMENTS ON CAD

Generally, the most natural way to communicate information about geometry is by means of sketches. So as far as possible, the dialogue between the user and a CAD system should be graphical. The primary output from a CAD system is therefore an engineering drawing which conveys manufacturing information from design function to the production function. The requirements on a CAD system based on geometric modelling in an manufacturing environment are listed in the following:

1) Ability to generate 2D engineering drawing of a design. This ability is supported by a good set of geometric construction facilities to assist in the creation of sketches.

2) Ability to generate a 3D model from 2D views. (This process is more natural to designers who interpret 2D drawings into 3D object. Alternatively, the ability to generated 3D object from 2D constructs would be more akin to the habit of some designers. as it is easier to think in term of 2D.)

3) Ability to view (with option on automatic hidden line removal) and section (with option on automatic hatching) the 3D model. (As it is necessary to check the model corresponds to the design intend of the designer.)

4) Ability to calculate automatically mass properties of the 3D object, i.e. area, volume, weight, centre of gravity etc. (This is necessary to aid design calculation.)

5) Ability to compare two models and display any difference between them. (So that the designer can compare the new design with previous issues.)

6) Ability for the semi-automatic or automatic code classification of the 3D object.

7) Ability to generate geometry information from the 3D model as input to other design analysis programs.

8) Ability to generate finite element information from the 3D model for analysis.

9) Ability to carry out tolerance build-up calculations.

10) Ability to produce drawing with views and sections generated by (3).

11) Ability to add (under user control) textual notes, dimensions and tolerance information to any view or section on a drawing.

12) Ability to store and retrieve standard parts i.e. nuts, bolts, clamps, fixtures etc and output them on a drawing.

13) Ability to generate assemblies from previously drawn components.

14) Ability to explode assemblies to generate isometric view with option on automatic hidden line and surface removal. (This facility is needed for technical illustrations and manuals).

15) Ability to modify the model or drawing locally so that only the modified geometry need to be specified (to minimize work). The affected attributes, if any, of the model (i.e. dimensions, tolerances etc.) should be updated automatically and display for user's approval or further modification. (If other departments have accessed the original model for applications then these department must be informed when a new model is issued.)

16) Ability to include details of drawing updates, issue number, draughtsman's indentification etc. in the finished drawing.

3.8.2 REQUIREMENTS ON CAM

Interrogation and extraction of data from the geometric model are of crucial importance for all applications of geometric model including manufacturing. The requirements for computer aided manufacturing are described in the following:

1) Ability to compare a new issue of a drawing or model with the previous issue and indentify what has been altered.

2) Ability to derive manufacturing information (i.e. material, tolerances, finishing requirements etc.) from the drawing or model.

3) Ability to extract manufacturing features from the model.

4) Ability to check for similar NC programs already written that can be fully or partially reused.

5) Ability to access the tool library to derive details and geometry of tools and clamps etc. for proper collision checking.

6) Ability to keep record of which tools are used on which task, and which task used which tools. (This information will help the user or the system to select the proper tools.)

7) Ability to modify the model to be machined to produce a manufacturable model. The modification instruction can be stored and used later.

8) Ability to take the output from (1) and to alter the NC program to reflect these changes without complete reprogramming.

3.8.3 REQUIREMENTS FOR PROCESS PLANNING

Before a part can be manufactured, it has to go through a process planning stage. This function indentifies the appropriate manufacturing process required and machine tools needed together with the general machining strategies.

Most industrial process planning systems are based on the variant concept together with a parts classification coding scheme (usually based on group technology). Generative process planning based on geometric modelling is still under research. The following is some of the envisaged requirements for a general planning system based on geometric modelling (36):

1) Ability to extract features from a geometric model.

2) Ability to display features graphically and in a form usable to existing planning system.

3) Ability to generate a parts classification code from a geometric model (based on features) automatically.

4) Ability to transmit feature information to a generative planning system when available.

5) Ability to identify modifications to the model and effect neccessary changes reflecting those modifications in the planning process.

3.8.4 REQUIREMENTS FOR NC

The requirements for NC systems based on geometric modelling are as in the following:

1) Ability to extract information from the geometric model about object geometry, material, tolerance, finishes etc.

2) Ability to modify the contours (or the model itself) to be machined for roughing cuts etc.

3) Ability to generate extra geometry (using construction facilities available) for roughing or area clearance operation etc.

4) Ability to define or extract features from the geometric model.

5) Ability to deine the tool path by choosing a tool from the tool library and directing (either graphically via a cursor or name of entity) the tool to the face or profile of the geometric model to be machined.

6) Ability to define the workpiece, the part, tools and clamps and the facilities to detect clashing of these objects.

7) Ability to define a group of faces or holes (or features) to be machined together.

8) Ability to add extra machining instructions (i.e. feeds, spindle, speeds, offset, coolants etc.) and post-processor information. These information need to be able to be inserted at specific points on the tool path or part program.

9) Ability to display the tool paths when they are generated.

10) Ability to store cutter locations in a file.

11) Ability to output a complete APT part program or a form of APT 'GO TO / ' statements (compatibe with IBM APT-AC or SSR1).

3.8.5 REQUIREMENTS ON OPERATION

A CAD/ACM system is envisaged to provide facilities to meet the following operation requirements:

1) Ability to operate the screen cursor by digitized pen on a tablet.

2) Ability to display more than one view on the screen at any time.

3) Ability to switch either picture or text on or off the screen.

4) Ability to select entities by using cursor (or other means, i.e. names, labels, symbol etc.) in any of the displayed views.

5) Ability to window in and out freely.

6) Ability to modify, backtrack or correct easily any mistakes made without remodelling. (The design of an object is usually an interative process. This process will be greatly expediated if model can be edited easily.)

7) Ability to keep a journal file of all operations and facilities to save any intermediate results. So that the model or results can be recreated easily.

It is the ease of modification and operation that distinguishes a good and practical production grade system.

3.9 FUTURE TREND OF GEOMETRIC MODELLING IN MANUFACTURING

In contrast to to the current engineering practice that the engineering drawing is the master representation of the part, a geometric model will be the master representation of the part for all functions (i.e. from marketing, design, production to distribution).

In future, the primary function of the design function is to produce a part geometric model which captures all design and manufacturing information presently represented by the engineering drawing.

As can be seen in this perspective, the production of the engineering drawing from the design function will be only a secondary function (for departments or companies not yet computerized or for legal and patent purposes). It should be emphasized that the engineering drawings are to be generated from the geometric model and not the other way round as done currently.

All engineering functions (engineering, production etc.) downstream from the design function will accept the geometric model of the part as the master input. These functions will interrogate the geometric model to generate the necessary information required. They may also generate additional information to the master model (if that is intended) or generate secondary models. For example, in manufacturing function, a manufacturing model (Section 6.1) may

be generated from the master model. Similarly, a machining model (Section 6.3) may be generated for NC machining.

3.10 SPECIFICATION OF A PRACTICAL CAD/CAM SYSTEM BASED ON GEOMETRIC MODELLING

It has been pointed out (in Section 3.) that Lucas is interested in developing a practical CAD/CAM system based on geometric modelling. Two areas most interested to Lucas are indentified as design and manufacture. (This is typical of engineering and manufacturing companies.)

In view of the future trend and current requirements identified above on geometric modelling, the following is an initial specification of a practical CAD/CAM system integrating geometric modeller, design, drafting and NC manufacturing. When this system is working, then the later objectives would be to have a system satisfying most of the requirements idenified above (in Section 3.8).

3.10.1 THE MODELLING SYSTEM

The modeller is to have both topology and geometry handling capabilities:

1) Topological functions

The modeller will have topological routines for building up datastructures (see Section 2.5) representing the basic shape entities in form of topological elements, i.e. objects, faces, edges and vertices together with their relations (i.e. connectivities and adjacencies etc).

The modeller will have the ability to construct 2D profiles and 'lift' these to become 3D objects.

The modeller also has the capability to carry out general intersection calculation between objects and the Boolean operations (i.e. union, intersection, difference etc) on objects.

2) Geometrical functions

The modeller will have facilities to handle the following geometrical entities:

1. Surface:

Plane, cylinder, cone, sphere, sphere.

2. Track:

Straight line, arc, circle, ellipse.

3. Point.

The modeller can store, display, transform and moidify all the above geometrical entities. In effect, the modeller is a 3D solid geometric modeller.

3.10.2 THE INPUT SYSTEM

The function of the input system is to input to the system the geometry of the components (2D, 2 1/2 D and 3D) to be manufactured.

As far as possible, the input system is to be conform to the requirements of design engineers and draughtsmen (see Sections 3.8.1, 3.8.5).

The user interface will be implemented for interactive processing. It would appear to the user as if it were a intelligent draughting system. It will calls necessary modelling facilities to execute geometric processing.

3.10.3 THE OUTPUT SYSTEM

The output system will generate ouput as required by the design,

drafting and NC functions (see Sections 3.8.2, 3.8.3, 3.8.4). Specifically, the output system will output the following as required by the user:

1) Generate views suitable for graphic display or plotting.

2) Engineering drawing.

3) Provide APT source geometry statements as required by NC programmer for both 2 1/2D and 3D parts.

4) Generate and display cutter path automatically for 2 1/2 D parts.

5) Generate a complete APT part program for the manufacture of the part as required for 2 1/2D and 3D parts.

It is expected that the main economic payback will be in the improving of lead time in both design and manufacture made possible by the system.

3.11 IMPLEMENTATION OF SYSTEM

In order to reduce the long lead time for the development of the above CAD/CAM system, it is decided to buy the various components of the system, whenever possible and only to develop those parts

which are not available commercially.

3.11.1 SELECTION OF THE ROMULUS GEOMETRIC MODELLER

As discussed in Section 2.5, a B-rep modeller is more suited for drawing and NC manufacture applications because of its datastructure. ROMULUS is a commercially available B-rep geometric modeller developed by Shape Data Limited (9).

ROMULUS is based on the considerable research work of Dr Ian Braid's BUILD system (2) at the Cambridge University. It also represents the state of the art of geometric modelling. Recent research work of Grayer (37) has showed that for 2 1/2D objects, NC tapes can be generated automatically from B-rep base modeller.

A detailed study was then carried out to compare various facilities available in ROMULUS and that of MEDUSA (a facetted modeller) and TIPS (a half space modeller) (7, 17). It was concluded that ROMULUS was most suited for our requirements.

In consideration of the above, together with the fact that Shape Data Limited will made the source code of ROMULUS available for research, it was therefore decided to select ROMULUS as the central module for the integrated CAD/CAM system

3.11.1.2 ROMULUS DATASTRUCTURE

As much of the research work will be developed around ROMULUS, it is important to have an appreciation of the ROMULUS datastructure which is described briefly in the following (Figure 3.4):

1) An object (i.e. body) has an unordered list of faces.

2) A face has an ordered list of loops and its owning body.

3) A loop has an ordered list of vertices and its owning face.

4) Each vertex points to an edge.

5) An edge points to two vertices (which in turn defines the edge).

6) Two co-edges point to and define a curve.

7) A curve points to two co-edges and relates to two adjacent loops.

8) The corresponding links to geometry for faces, edges and vertices.

The ROMULUS datastructure is similar to the datastructure proposed for NC manufacturing applications discussed in Section 2.5.4.

3.11.2 SELECTION OF THE INPUT SYSTEM

There are many drafting packages available commercially. The MEDUSA system offers a very unique feature of the ability to create 3D ROMULUS models from 2D constructs which contains four types of information:

1) 2D views of object components.

2) Linking information defining how these 2D views are related and how to generated the 3D object from them.

3) Information indicating how various components are to be assembled into objects.

4) Information of which views, projections and sections are to be output.

The views and sections generated by the ROMULUS system are to be stored in MEDUSA 2D sheets which can then be annotated and dimemsioned in MEDUSA to produce full engineering drawings. The configuration of the system is shown in Figure 3.5.

It is this unique feature together with the fact that MEDUSA has been proved to be a good drafting system that it was selected as the input system (with the added advantage that it performs the engineering drawing for the envisaged output system).

3.11.3 DESIGN OF THE OUTPUT SYSTEM (ROMAPT)

CAD systems, whether they are drafting aids, complex surface representation or geometric modellers, are usually based on a bounded geometry. In the past this had let to considerable problems when trying to use a system such as APT to handle the CAM end of the total process.

The use of APT in this way has really been a relatively short term solution to the problem of providing an fully integrated CAD/CAM system. This, however, does not mean APT will be redundant in a few years. APT has been proven a useful manufacturing tool, and it is very unlikely to be replaced (partly because of the huge investment made in APT). It will certainly be used increasingly in an interactive, graphical mode as hardware costs fall. While drawings are still the chief means of communication, its popularity is certain.

The ultimate solution to the above problem is the development of a bounded geometry NC processor making use of the full potential of solid geometric modellers. Currently, CAM-I's Advanced Numerical Control Project (ANC) is addressing the problem (with an initial forecasted budget of US$4 million for a 2 1/2D bounded NC processor) (38).

As there is no commercial available system, this output system is

to be developed in house. Within Lucas Industries, there is a substantial investment in APT and with it a concomitant pool of NC programming expertise. With due consideration for the above mentioned factors, the output system (ROMAPT) is initially designed to output NC information (i.e. APT geometry) from the modeller for use in APT part program. Later, it is intended to develop algorithms to gain insight for the requirements of a bounded NC processor.

This is the kernel of this research work.

FIGURE 3.1: APT-AC organization

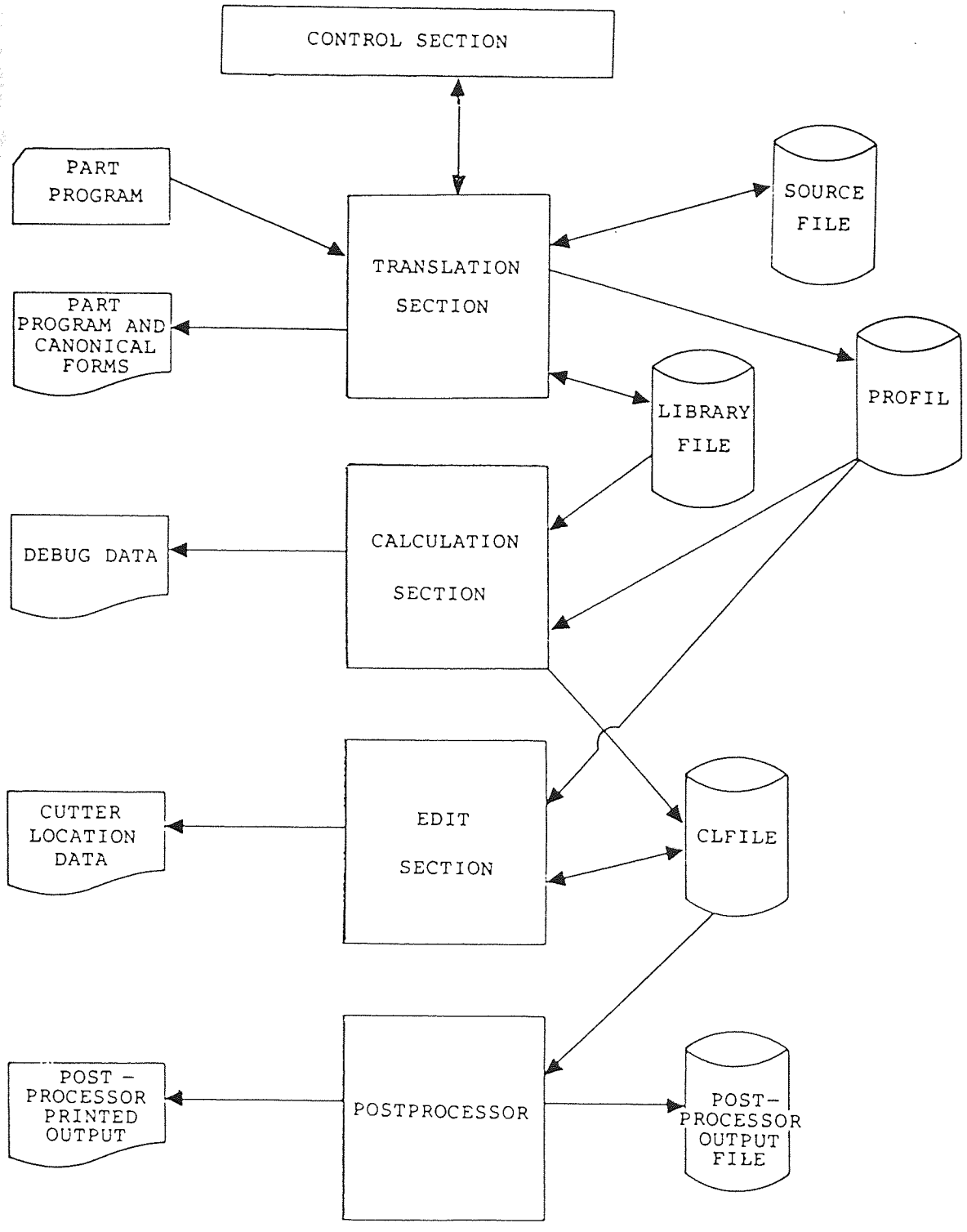


FIGURE 3.2· APT4 organization

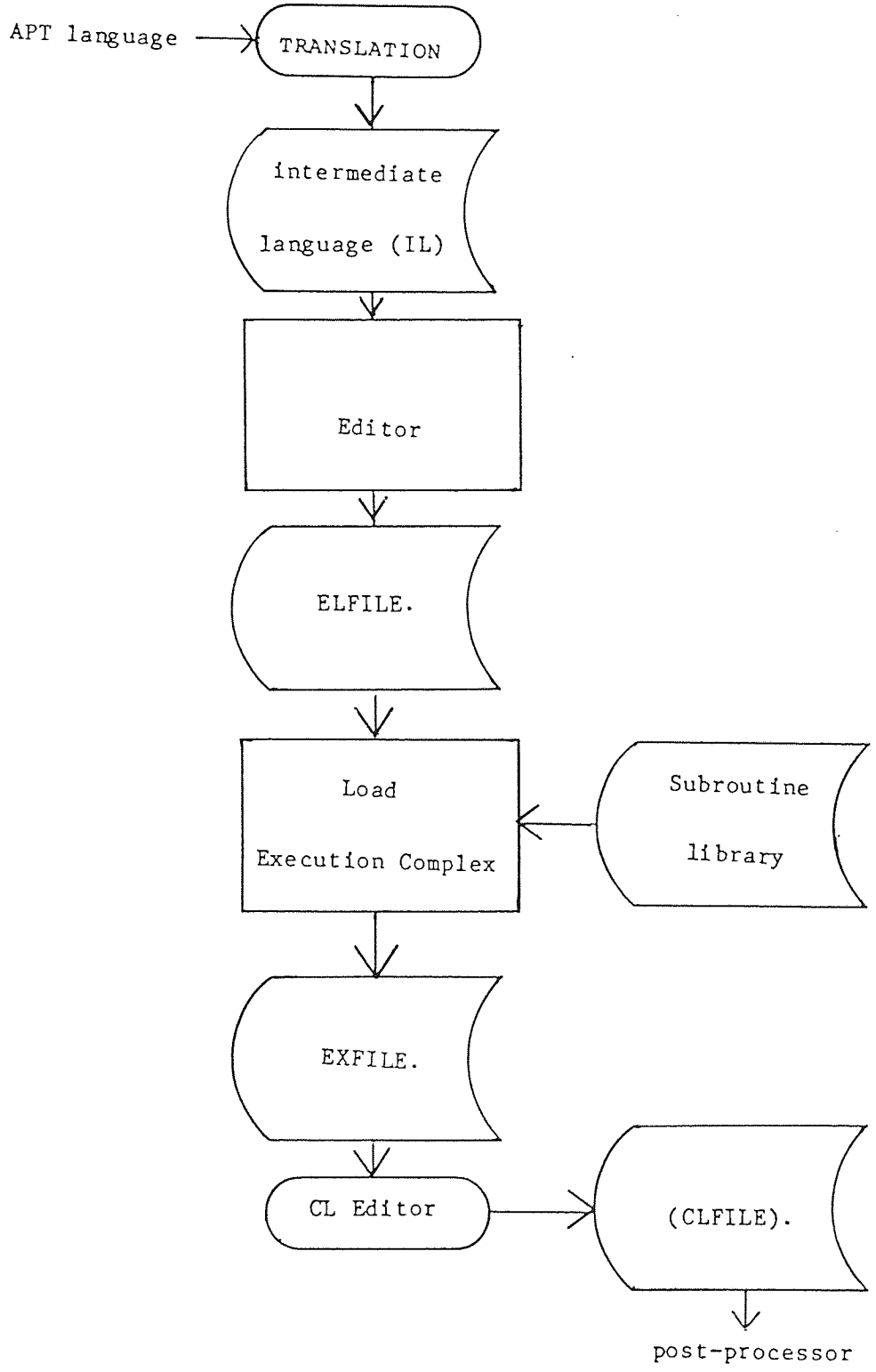


FIGURE 3.3 Computer integrated manufacturing

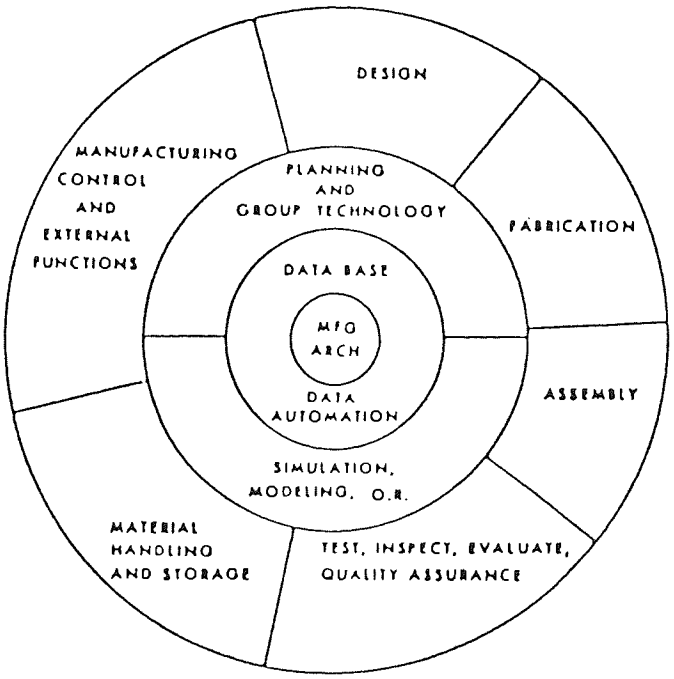
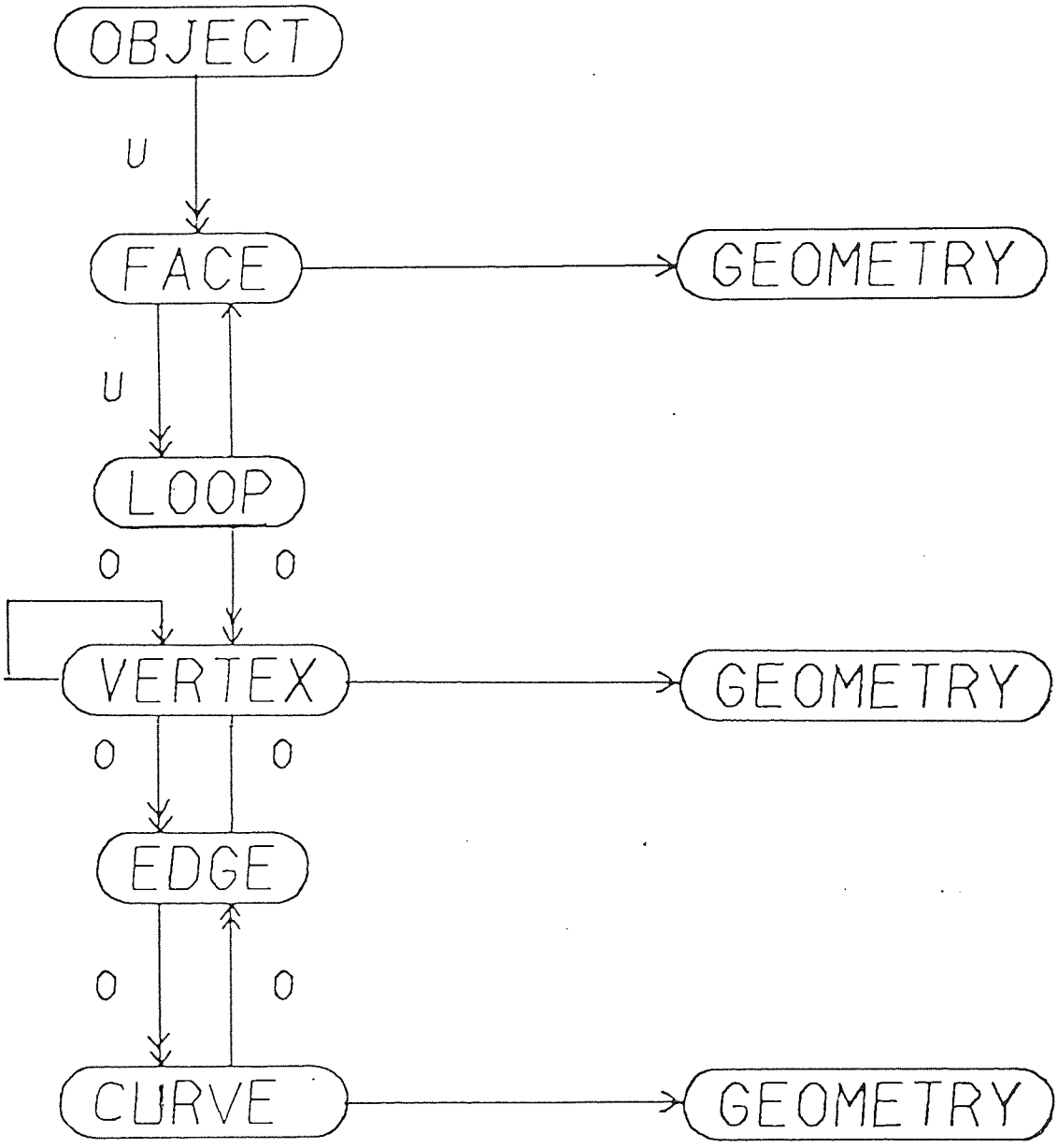
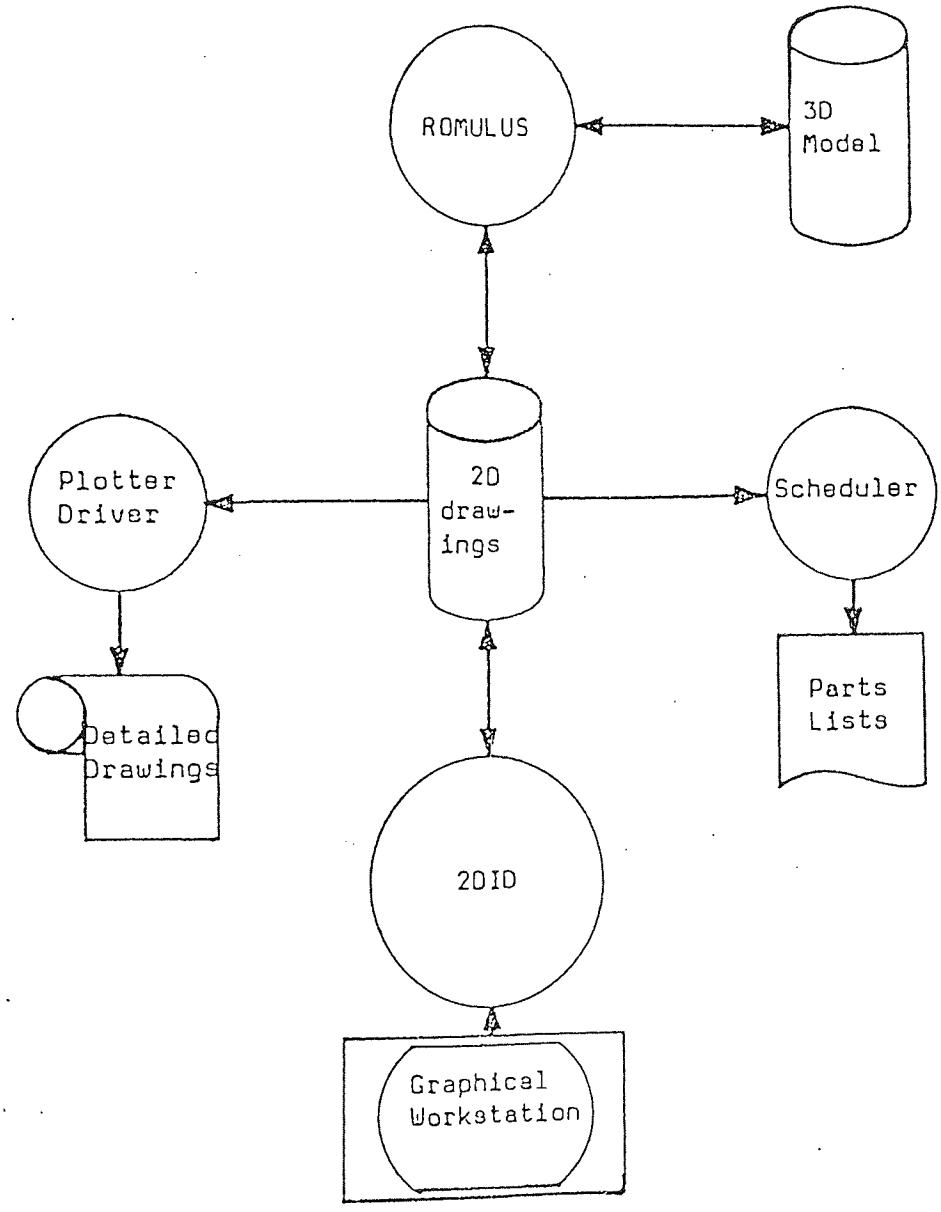


FIGURE 3 4· Romulus data structure



O = ORDERED U = UNORDERED

FIGURE 3.5 Romulus and Medusa configuration



4. ROMAPT A NEW LINK BETWEEN CAD AND CAM

The use of numerically controlled machine tools is now firmly established within manufacturing industry. Design and manufacturing systems based on an increased armoury of mathematical techniques for manipulating shapes by computer will greatly increase the use and effectiveness of numerically controlled machine tools from now on.

One aspect of this armoury, namely, the advent of geometric modelling techniques for the description and manipulation of 3-D shapes should further this effectiveness. It is essential therefore, to establish a link between geometric modelling and NC machine tools to provide an integrated computer-aided-design and manufacturing system.

However most geometric modelling systems are "bounded" in a mathematical sense - i.e. lines and planes are finite, whereas the controlling computer language for many NC machine tools, namely APT is "unbounded" - i.e. lines and planes are infinite. APT systems, however, are presently only linked to wire-frame drafting systems. The combination of a geometric modeller and APT will provide a powerful manufacturing system for industry from the initial design right through part manufacture using NC machines. This research work investigates the theory, developes the concept and algorithms, and the design of an experimental software interface (ROMAPT) between a bounded geometric modeller (ROMULUS of Shape Data Limited) and an unbounded NC processor (APT).

APT is used because of its popularity in industry. ROMULUS is used because it represents the state of the art of solid modelling. Although ROMULUS is basically a boundary representation modeller, the current trend is such that constructive solid modellers (CSG), i.e. PADL and half-space modeller, i.e. TIPS etc, will provide boundary representation as well. Hence, the technique developed here will also be applicable to other modellers.

4.1 ROMAPT DESIGN CONCEPT

The most straight-forward approach is to design a bounded NC processor to interface with a bounded geometric modeller. However, the manufacturing industry has invested substantially over the years in APT and similar NC processors. It is very unlikely that the industry would be willing to shed this investment instantly. The American National Stardard Institue ANSI has set 1983 as the target date for the launch of a revised APT language standard to cater for the additional requirement of intelligent CNC, robot, and inspection machines etc. It, therefore, appears that the revised APT language will become even more widely used.

A bounded NC processor with access to a geometric modeller would have the the advantages over APT in geometry related computing. For postprocessing, it would require something like APT's postprocessing capabilities. Moreover, the current state of development of geometric modelling is such that a universal

accepted bounded geometric modeller standard does not yet exist. Hence, it is not possible at present to design a universal bounded NC processor. Currently, CAM-I is addressing this problem.

The investment needed to develop n bounded NC processors for n bounded geometric modellers could be excessive. The alternative approach is to design a link between a state of the art bounded geometric modeller and a popular unbounded NC processor like APT. Using this approach, the theory, concept and algorithms required for the link can be readily tested, verified and refined. The theory and algorithms developed can then be used in future to develop an modeller-independent link between solid modelling and APT. The geometric modeller will be used to provide data for NC calculation and to resolve ambiguities in APT processing, i.e. to provide APT with bounded geometry intelligence to overcome some unreliability problems in APT (caused by the part-seeking algorithms of APT).

This amounts to a develop a technique of binding the unbounded APT system to the bounded ROMULUS solid modeller.

4.2 APPROACHES TO NC MACHINING DEVELOPMENT

It is assumed that most of the machining strategy and technology information required for the NC manufacture will be provided by a manual or a computerized process planning system with access to the

geometric modeller (see Section 3.8.3). This system will supply the following information:

1) Operations sequence.

2) Machine tool selection

3) Clamping selection.

4) Sub-operation sequence.

5) Cutting tool selection.

6) Cut distribution.

7) Selection of cut data (feed, speed and spindle).

The ROMAPT system is envisaged to provide the following:

1) Description of the part and the blank stock.

2) APT geometry statement as required.

3) Tool path calculation.

4) Complete APT part program.

The following NC machining development approaches based on a

geometric part model are attempted.

Automatic APT Geometry Generation

1) ROMAPT (Stage 1)

This is a software link between ROMULUS and APT to generate APT geometry statements automatically.

2) AIAPT

This is a modeller-independent implementation of ROMAPT (stage 1) using the CAM-I Application Interface subroutine specifications (12).

Automatic NC Cutter Path Generation

3) Bounded Cutter Path Generator

This is an automatic cutter location generator for a geometric model.

4) ROMAPT (Stage 2)

This is an extension of ROMAPT (stage 1) to provide automatic generation of APT motion (cutter path) statements.

4. ROMAPT (Stage 1)

To allow for the diversity of possible manufacturing processes and machining strategies that may be used, the interface can be designed to output APT geometric source statements. In this case the output is readable to the NC programmer; he can check and edit, if required, the APT source to suit the task. Conceptually, this approach is very straight-forward and easy to use and was adopted with the following objectives:

1. Ease of use.

2. Enabling the NC programmer to understand the geometric model better.

3. Providing APT geometric source statements as required.

Within manufacturing industries, there is a substantial investment in APT and with it a concomitant pool of NC programming expertise. With due consideration for the above mentioned factors, the ROMAPT system was designed to convert bounded geometric information held in the geometric modeller to standard unbounded APT geometry statements. The experience and expertise of the NC programmer is

then used to complete the NC part program. This further allows the NC programmer to concentrate on the machining strategy rather than the geometry.

When this approach (stage 1) has been proven, then more automation (stage 2) in the form of automatic cutter path generation can be developed for the system.

4.3.1 ROMAPT SYSTEM CAPABILITY (STAGE 1)

Traditionally, a NC programmer is given a set of engineering drawings of the part he wishes to make. He has to understand and interpret the drawings accurately before he proceeds to write his NC part program. This is where much time is consumed and many errors are possible.

Working with a geometric modeller, in addition to the set of engineering drawings given, will help to reduce the above mentioned drawbacks. However, the NC programmer would need to have some facilities to enable him to query and understand the part given in the form of a geometric model defined by a data structure (Figure 3.4).

4.3.2 ROMAPT (STAGE 1) SPECIFICATION

In view of the requirements indentified (see Section 3.8), the

ROMAPT (stage 1) system is specified initially as in the following:

1) The system will take a 3D geometric model of an engineering component as input (i.e. a get facility).

2) The user can select the required machining angle for the model (i.e. transformation facilities like move and rotate) and is presented with a view from this angle on a tektronic compatible screen (i.e. view facility).

3) The user can plot a drawing of the component with or without hidden line removal (i.e. plot and hidden line removal).

4) Labels can be attached to geometric items for which the APT part programmer required the geometry. On the initial view, either all items are labelled or none are labelled. The programmer can then remove or request labels on individual items (i.e. name generation, labelling and plot selection facilities).

5) The labels given will be generated by the program and be of the following format:

AXXXXX

where, A is: P for a point

C for a circle.

L for a straight line.

E for other edge.

F for face

XXXXX is a number starting from 1.

e.g. P1, C2, L8, E21, F34.

6) Labelling will initially only refer to points, lines and circles (for 2 1/2D parts), and later will be exteneded to cover other items including surfaces for 3D parts.

7) At any time, the view can be redrawn with the relevant labelling. The facility to window any part of the view is to be included. In the event of the particular label of the item not being clear, the user will be able to request the label for that particular item to be displayed away from the model.

8) The user will be able to point to any particular circlular arc and be given the centre, radius and the start and end points of the arc (for drilling applications).

9) The user will be able to construct extra lines and horizontal planes to the model (for NC roughing apllications i.e. grid and plane facilities).

10) The user will be able to store the modified model with extra construction geometries (i.e. save facility).

11) The user will be able to generate APT geometry source statements for all geometric entities in the model including construction geometries. (i.e. APT enquire facilities).

12) For a modified object, the user will be able to compare and alter NC instruction only for the modified area without having to treat it as a new job (i.e. monitor file facility).

4.3.2.1 COMPUTER SOFTWARE DESIGN

The following general principles have been applied in the development of the system:

1) The ROMAPT system will be an extension and compatible to ROMULUS. (To a user, ROMAPT is ROMULUS with extra commands and facilities for APT geometry generation.)

2) All codings will be done in standard ANSI FORTRAN 66. (As ROMULUS is implemented in the same language.)

3) As far as possible, ROMULUS software routines will be used (to take advantage of the latest development in ROMULUS).

4) New routines are initially designed to accomodate changes (a modular approach) due to the continuous development of ROMULUS and ROMAPT and to help to facilitate program diagnosis.

5) Later, the system will be tuned for operational efficiency.

4.3.3 GRAPHIC MANIPULATION

Supplementary to manipulation facilities (i.e. transformations like rotation, scaling, moving and visualization like viewing, hidden line removal, silhoutte line, plotting etc) provided by ROMULUS, further manipulating facilities have been developed to help a user's understanding of the geometric model. All these additional facilities will help a user to generate APT geometry statements for a component. These facilities are described in the following (see Appendix 2 for the full set of commands):

COMMAND	FUNCTION
1. VIEW	Facility to see the object via a display screen.
2. GENERATE	Generates name for unnamed geometric entities (points, edges, faces). naming conventions: P : point E : edge F : face
3. CANCEL	Cancels name(s) of geometric entities (point, edge, face).
4. LABEL	Labels name(s) of required geometric entities (point, edge, face) on a display drawing.

5. ENQUIRE — Outputs required geometric information of the model of the part.

6. SHIFT — Shifts the picture on display in X and Y direction.

7. WINDOW — Provides windowing facilities on current view. So that a more complex geometric model can be handled.

8. SELPLOT — Provides plotting facilities on current view selectively.

9. CONVENTION — Converts names to APT convention:

P: points

L: straight lines

C: circles

F: face

4.3.4 LOOP HANDLING FACILITIES

To enable the user to handle the different faces and profiles of the geometric model conveniently, various loop handling facilities are provided in the ROMAPT system as follows:

1) SELECT face LOOP

Function: to display face and its associated loop labels.

2) SELECT face [LOOP] n

Function: to select a face or a loop within a face for operation.

3) SELPLOT face [LOOP] n

Function: to select a face or a loop within a face for plotting.

4) APTLABEL [FACE/ POINT/ EDGE]

Function: To label all faces, or points, or edges.

5) APTLABEL [POINT/ EDGE] face [LOOP] n

Function: To label all points or edges of the loop n of face specified.

4.3.5 MATHEMATICS OF GRAPHIC MANIPULATION

The ability to represent and view a 3D object together with the transformation facilities like rotation, shifting, scaling and projection of the object is foundamental to the understanding of the shape of the object. All these operations can be implemented

in the computer neatly by a set of matrix transformations (39).

The transformation from homogeneous co-ordinates (i.e. a four dimensional position vector [X Y Z H] representing a three dimensional point [x y z]) to ordinary co-ordinates is given by the equation:

$$[X\ Y\ Z\ H] = [x\ y\ z\ 1]\ T$$

or alternatively,

$$[x^*\ y^*\ z^*\ 1] = [X/H\ Y/H\ Z/H\ 1]$$

where T is the transformation matrix and is given by:

$$\begin{bmatrix} a & b & e & p \\ d & e & f & g \\ h & i & j & r \\ l & m & n & s \end{bmatrix}$$

Let the matrix T be partitioned into its various components which are of interest:

1) a 3 x 3 matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \end{bmatrix}$$

Local shearing and rotation is produced by this 3 x 3 matrix.

2) 1 x 3 row matrix

$$[l \quad m \quad n]$$

This 1 x 3 row matrix produces the translation.

3) 3 x 1 column matrix

$$\begin{bmatrix} p \\ g \\ r \end{bmatrix}$$

This 3 x 1 column matrix produces perspective transformations.

4) s

This s term produces global scaling.

4.3.6 ACCESSING THE DATASTRUCTURE OF THE MODEL

The Generate, Cancel, Label and Enquire and other facilities mentioned above (Section 4.3.3) requires access to the data structure of the ROMULUS boundary geometric model (Section 3.11.1).

The boundary data structure can be described in term of a tree with root (i.e. body), node (i.e. face or loop), leaf (i.e. vertices or edges) and link (pointers). (see Figure 4.1).

An algorithm for accessing every entities of such structure is as follows:

1) Start at root.

2) If all link entities are marked, then go to step (7).

3) Go down the tree to any unmarked link until a leaf is found or a node with all links being marked.

4) If a leaf is reached, mark the link and return to the previous node and go to step (3).

5) If the node is the root, go to step (2).

6) Go up the tree through the link to a node, mark the link and go to step (3).

7) Unmarked all links.

8) Stop

This algorithm will form the basic of a variety of algorithms accessing the geometric model.

4.3.7 APT GEOMETRY STATEMENT GENERATION

The ROMAPT program scans through the data structure held in the geometric model in a systematic manner: firstly the body, then the faces followed by the points and edges (see Section 4.3.6). This geometric data (bounded) is then converted into standard APT geometry format (unbounded). The APT geometric statements for a component are output accordingly. The components are processed individually. For each component (i.e. a body), the APT geometric definition for every face will be output and its associated points and lines will be output accordingly in a systematic manner.

To provide compatibility among many APT processors, standard APT-AC/APT4/SSR1 (29, 30, 31) geometric definitions are employed as

in the following.

1) POINT

P1 = POINT/ x, y, z

The co-ordinates of point P1 are given in (x, y, z).

2) EDGE

E1 = LINE/ P1 , P2

E1 is a straight line defined between point P1 and point P2.

E2P = POINT / x , y , z

E2 = CIRCLE/ CENTER , E2P , RADIUS , t

$$ P1 TO P2

E2 is a circular arc / circle defined between point P1 and point P2 with centre at E2P (x, y, z) and radius of t units.

```
E3 = ELLIPS/ INTOF , F1 , F3

$$ P3 TO P4
```

E3 is an ellipse between point P3 and point P4 and is defined by the intersection of plane F1 and cylinder F3.

3) FACE

```
F1P = POINT/ x , y , z

F1V = VECTOR/ u , v , w

F1 = PLANE/ F1P , PERPTO , F1V
```

F1 is a plane defined by a point on the plane at F1P (x, y, z) and the normal to the plane: vector F1V (u, v, w).

```
F3P = POINT/ x , y , z

F3V = VECTOR/ u , v , w
```

F3 = CYLNDR/ F3P , F3V , t

F3 is a cylinder defined by a point F3P (x, y, z) on the cylinder axis and a vector F3V (u, v, w) along the cylinder axis and the radius (t) of the cylinder.

F4P = POINT/ x , y , z

F4V = VECTOR/ u , v , w

F4 = CONE/ F4P , F4V , d

F4 is a cone defined by the vertex point F4P (x, y, z), the axis vector F4V (u, v, w) and the half-angle (d) of the vertex of the cone.

F5P = POINT/ x , y , z

F5 = SPHERE/ CENTER , F5P , RADIUS , r

F5 is a sphere defined by the center F5P (x, y, z) and the radius (r).

F6P = POINT/ x , y , z

F6V = VECTOR/ u , v , w

F6 = TORUS/ F6P , F6V , r1 , r2

F6 is a torus defined by the center point F6P (x, y, z), the axis vector F6V (u v, w) the major radius (r1) and the minor radius (r2).

As there is no formal definition for torus in APT4/SSR1, this has to be defined indirectly as a surface of revolution.

A torus type has to be defined as a surface of revolution of a synthetic curve in SSR1. (Both the synthetic curve and the surface of revolution types are non-compatible with APT-AC, and they are unique to the SSR1 system).

As an example, to represent a torus with:

CENTRE : 0,0,0

NORMAL : 0,0,1

MAJOR RADIUS : 5

MINOR RADIUS : 1

The following procedures of SSR1 are required:

```
P0 = POINT / 0,0,0

PX = POINT / 0,0,5

PP1 = POINT / -6,0,0

PP2 = POINT / -5,0,1

PP3 = POINT / -4,0.0

PP4 = POINT / -5,0,-1

C1 = SCURV / CURSEG, PP1, PP2, PP3

C2 = SCURV / CURSEG, PP3, PP4, PP1

C = SCURV / COMBIN, C1, C2

TOR = SSURF / REVOLV, C, AXIS, P0, PX, CCLW, 0, 360
```

The torus generated is a form of sculptured surface (31).

Note that any APT part program containing the standard APT-AC torus definition will not be accepted by the SSR1.

Only the most elementary forms of geometry definitions (i.e. a subset of the lowest common demoninator) of the APT family is used to allow maximum transportability among many different APT processors.

4.3.8 ROMAPT (STAGE 1): MODE OF OPERATION

The ROMAPT interface consists mainly of 3 commands in 3 different modes of operations:

1) APTALL (dump mode)

Generates APT geometric statements for all geometric entities of the body concerned.

2) APTOPTIMAL (optimised mode)

Generates an optimal set of APT geometric statements (mainly for 2 1/2 D parts).

3) APTENQUIRE (interactive mode)

Generates APT statements for requested geometric entities

interactively to enable a NC programmer to select the minimum set of APT geometric statements for the component concerned.

APTENQUIRE can be used to pinpoint the particular entities required, thus an even smaller set of geometric statements can be obtained by the NC programmer.

4.3.8.1 GEOMETRY OF ROMULUS

There are three types of geometry entities i.e. point, track and surface represented in ROMULUS (9). They are stored in a computer real array (EQ) and are described in the following:

1) Point type

POINT

EQ(1), EQ(2), EQ(3) stores x, y, z co-ordinates of a point.

2) Track tpye

STRAIGHT LINE

EQ(1), EQ(2), EQ(3) stores the point on the line.

EQ(4), EQ(5), EQ(5) stores the normalized vector along the line.

CIRCLE

EQ(1), EQ(2), EQ(3) stores the centre of the circle.

EQ(3), EQ(4), EQ(5) stores the normalised normal to the plane of circle.

EQ(6) stores the radius of the circle.

ELLIPSE

EQ(1), EQ(2), EQ(3) stores the centre of the ellipse.

EQ(4), EQ(5), EQ(6) stores the normalised normal to the plane of ellipse.

EQ(7) stores the semi-major axis of the ellipse.

EQ(8), EQ(9), EQ(10) stores the normalized vector along the major axis.

EQ(11) stores the minor axis.

INTERSECTION CURVE

EQ(1), EQ(2) references the two intersecting surfaces.

3) Surface type

PLANE

EQ(1), EQ(2), EQ(3) stores a point on the plane (closest to origin).

EQ(4), EQ(5), EQ(6) stores the normalized normal to plane.

CYLINDER

EQ(1), EQ(2), EQ(3) stores a point on centre line.

EQ(4), EQ(5), EQ(6) stores the normalized vector along the axis.

EQ(7) stores the radius.

CONE

EQ(1), EQ(2), EQ(3) stores a point on axis

EQ(4), EQ(5), EQ(6) stores the normalised vector along axis.

EQ(7) stores radius at the specified point.

EQ (8) store the sine of the half angle of divergence of cone.

TORUS

EQ(1), EQ(2), EQ(3) stores the centre of torus.

EQ(4), EQ(5), EQ(6) stores the nomalized normal to plane of torus.

EQ(7) stores major radius.

EQ(8) stores minor radius.

SPHERE

EQ(1), EQ(2), EQ(3) stores the centre of sphere.

EQ(4) stores radius.

As can be seen, the geometry entities in ROMULUS corresponds to those APT geometry defined in Section 4.3.7.

4.3.8.2 ALGORITHMS FOR APT GEOMETRY GENERATION

As ROMULUS usually generalizes circle into generalised ellipse for processing, this may caused problem for APT to generate a circular interpolation for a circle represented by a generalized ellipse. To overcome this problem, generalized ellipses are coverted back into true circles, if applicable, before they are output into APT

geometry statements.

Algorithms for the APT geometry generation facilities of APTALL, APTAPTIMAL and APTENQUIRE are described in the following:

A) APTALL

1) Start at the object.

2) If all faces are marked, go to step (11).

3) Go to an unmarked face, output the corresponding surface geometry and mark the face.

4) Go to an unmarked loop of the face, mark the loop.

5) If all vertices in the loop are marked, go to step (7).

6) For the current loop, go to an unmarked vertex, output the corresponding point geometry and mark the vertex, go to step (5).

7) If all edges in the loop are marked, go to step (9).

8) For the current loop, go to an unmarked edge (via an vertex), output the corrresponding track geometry if the coresponding curve has not been marked, and mark the edge and the corresponding curve

if not already been marked.

9) If all loops in the face are marked, go to step (2).

10) Else go to step (4).

11) Unmarked all entities.

12) Stop.

B) APTOPTIMAL

APTOPTIMAL attempts to minimize the APT geometry required for 2 1/2D parts. All planes with surface normal orthogonal, and cylinders with axis in parallel, to the orientation of the tool axis are ignored .

The algorithm for APTOPTIMAL is similar to that of APTALL except that step (3) is modified as in the following:

1) Start at the object.

2) If all faces are marked, go to step (11).

3) Go to an unmarked face, mark the face. If the face is a plane and the surface normal is orthogonal or the face is a cylinder with

the axis parallel to the tool axis, then ignored the face, go to step (2). Else output the corresponding surface geometry of the face.

4) Go to an unmarked loop of the face, mark the loop.

5) If all vertices in the loop are marked, go to step (7).

6) For the current loop, go to an unmarked vertex, output the corresponding point geometry and mark the vertex, go to step (5).

7) If all edges in the loop are marked, go to step (9).

8) For the current loop, go to an unmarked edge (via an vertex), output the corrresponding track geometry if the coresponding curve has not been marked, and mark the edge and the corresponding curve if not already been marked.

9) If all loops in the face are marked, go to step (2).

10) Else go to step (4).

11) Unmarked all entities.

12) Stop.

C) APTENQUIRE

The algorithm for APTENQUIRE is similar to APTALL except in two aspects:

a) The user supply the name of the face, loop, edge or point required.

b) If the direction of the axis of a circle is orthogonal to the tool axis, the circle is converted into a straight line defined by the projection of the circle onto a horizontal plane.

The algorithm for APTENQUIRE is given in the following:

1) Start at the object.

2) If all faces are marked, go to step (10).

3) Go to an unmarked face, mark the face. If the face is the one specified, output the surface geometry if required. If more face required, go to step (2). If no more entity is required, go to step (10). Else go to step (4).

4) If all loops in the face are marked, go to step (3).

5) Go to an unmarked loop in the face, mark the loop. If the loop is the one required, go to step (6). Else go to step (4).

6) If all vertices in the loop are marked, go to step (8).

7) For the current loop, go to an unmarked vertex, mark the vertex. If the vertex is the one specified, output the corresponding point geometry. If more vertices are required, go to step (6). If no more entity required, go to step (10). Else go to step (8).

8) If all edges in the loop are marked, go to step (5)

9) For the current loop, go to unmarked edge, mark the edge. If the edge is the one specified, output the corresponding geometry. If more edges are required, go to step (8). If no more entity is required, go to step (10). Else go to step (8).

10) Unmarked all entities.

11) Stop.

4.3.9 ROMAPT EDITING FACILITY

The output of various ROMAPT APT facilities (see Appendix 2) is a set of APT geometry source statements on a file required for a NC APT part program. To complete the part program, the user will have to supply the necessary program description, tolerance, dimension, tool definition, spindle, feed, speed, and various post-processor words via an editor. It may be inconvenient for the user to get in

and out of ROMAPT to do the editing.

To facilitate APT part programming development within the ROMAPT environment, an editor command has been added to allow a user to add APT commands (i.e. control statements and post-processors etc.) via the screen to the file. The command is as follows:

CNC EDIT [APT statements]

Function: To input user's APT statements.

Hence a complete APT part program can be written (with the geometry generated by ROMAPT) into a file via a screen. This program is then ready to be processed in APT to generate a NC tape for the machine tool.

4.3.10 ROMAPT CONSTRUCTION GEOMETRY

The ROMULUS geometric modeller stores primarily the boundary description (i.e. points, edges, faces) of the object. Using ROMAPT, various profiles of the object can be defined in terms of APT geometry. (Hence, the profiling part of NC part programming can be made easier.) However, to machine the object, extra geometric entities (i.e. points, line, planes which are not part of the geometric model of the object) are needed for NC applications like roughing, area clearance and pocketing.

The construction geometry facilities in ROMULUS is intended for use in the construction of the model and not for use in NC applications directly. Therefore, the existing construction geometry facilities have been extended to provide extra construction geometry required by the NC programmer to complete the geometry description part of his part program.

The construction geometry facility consists of three parts: construction, displaying and manipulation. The construction part provides facilities to create construction geometries for NC applications. The displying part provides facilities to view construction geometries selectively. The manipulation part provides facilities to delete or convert construction geometry into APT format for NC applications.

4.3.10.1 CONSTRUCTION

As frequently required in NC programming is a need to define a horizontal grid of parallel straight lines (each separated by the cutter diameter) on a plane for area clearance or pocketing and a series of parallel plane (each separated by the cutter diameter) to a horizontal base plane for roughing. The construction facilities are described in the following:

CNC GRID [CURSOR/ vector 1, vector 2]

Function: To define a horizontal grid of straight lines (each separated by the amount specified by the user via the cutter definition) for area clearance operation in NC applications.

CNC PLANE [CURSOR/ height]

Function: To generate a set of parallel planes (each separated by the amount specified by the user via the cutter definition) to a horizontal base plane for roughing application.

4.3.10.2 DISPLAYING

This facility is provided so that the user can view construction tracks and surfaces conveniently:

CL2 [TRACK/ SURFACE]

Function: To view all tracks or surfaces generated.

CL2 [track/ surface]

Function: To view a track or surface.

4.3.10.3 MANIPULATION

A further facility is provided so that the user can inspect and get or delete construction geometry in APT format.

CANCEL [track/ TRACK/ surface/ SURFACE]

Function: To cancel a track or face, or all tracks and surfaces.

APTINFORM [track/ TRACK/ surface/ SURFACE]

Function: To inspect a track/surface or all tracks/surfaces.

APTENQUIRE [track/ TRACK/ surface/ SURFACE]

Function: To get APT geometry on a track/surface or all tracks/surfaces.

4.3.11 APT POCKET FACILITY

Automatic pocket facility for planes has been implemented to be compatible with IBM APT-AC standard (29). The syntax for the command is:

CNC POCKET [CURSOR] [CURRENT/ PSURF/ POINT/ ZAXIS/ SET]

Function: To output APT POCKET statements with the following options:

1) CURRENT: pocket on the currently selected face.

2) PSURF: pocket on user-generated planes.

3) POINT: pocket on points given (maximum 20, covex polygon).

4) ZAXIS: pocket on user defined Z-planes.

5) SET: user can redefine all default values of the pocket.

4.3.12 AUXILIARY FACILITIES

Various auxiliary facilities are provided to aid the user to generate a APT part program:

CNC [CUTTER] [FLAT/ BALL] radius

Function: To define the cutter and its size.

CNC [HEADER/ MACHINE/ END]

Function: To output default APT statements for the header section,

or machine section, or end statement.

CNC RETRACT height

Function: To define the retracting height for the cutter.

CNC [TOLERANCE] intol, outtol

Function: To define the tolerances intol and outtol for APT processing.

CNC [POINT/ DISTANCE] [CURSOR]

Function: To find out the co-ordinates of a point, or a distance between two specified points.

CNC [SET] point vector

Function: To change the point co-ordinate by a vector.

4.4 WORKED EXAMPLE

To prove the validity of the above concept, a worked example of a NC part program was carried out using the APT geometry generated by ROMAPT on an artifical mechanical part Gehause (Figures 4.2a and 4.2b) designed by using ROMULUS.

The generation of the APT geometry for the Gehause is shown in Figure 4.3.

After the geometry of the part has been generated, the NC programmer completes the NC program by adding the necessary APT machining instructions. The APT part program is then processed by APT and a post-processor to generate a NC tape. The cutter path is shown in Figures 4.4a and 4.4b, and the part has been machined successfully on a Lucas Model Making Machine.

4.5 DISCUSSION

The following is a discussion on some of the features of the ROMAPT (stage 1) system.

1) Names

Names of POINT/ EDGE/ FACE are the usual means of communication between the user and the system. These names are generated automatically by ROMULUS (though not consecutively). There are also facilities for cancelling and re-generating (consecutively) of names in ROMAPT. These names can be replaced by user defined names via the renaming function.

2) Machining strategy

In using ROMAPT (stage 1), the NC programmer has complete freedom in deciding the machining strategy and in the choosing of every separate tool path.

3) Accuracy

The accuracy of the system is limited inherently by the computer hardware used. Rounding errors may occur as a result of a trade-off between computing time and resolution. In testing for approximate equality, ROMULUS makes use of two quantities: REABS for large values and RENOR for normalized quantities. On entry to ROMULUS. REABS and RENOR are set to be 0.002 and 0.00002 respectively for the PRIME computer implementation. These factors account for the occurence of rounding errors in the sixth places after the decimal (Figure 4.5). Fortunately, these rounding errors are usually acceptable for machining.

4) Circle and ellipse

Both circle and ellipse are stored internally with references to their plane and cylinder. If the component is tilted, these references will remain the same and the circle remains unchanged. However. for machining, if machining angle is not adjusted accordingly, when the component is tilted, then the cutter path for a circle becomes an ellipse.

As ROMULUS works to a comparision accuracy of three decimal places and APT works to a higher accuracy, problem may be expected with

tangency condition (i.e. if a line is tangent to a circle in ROMULUS and both are ouput to APT in canonical form, APT may well not recognise them as being tangent and thus may fail on machining calculation.) However, this problem has not occurred during the various test runs. In the event of a problem occuring, this is envisaged to be remedied by the following:

a) Output the APT statements specifying tangency rather than the simple canonical form.

b) To run ROMAPT on a double precision version.

4.6 CONCLUSION FOR ROMAPT (STAGE 1)

It has been demonstrated that the foundamental concept behind ROMAPT i.e. the binding the unbounded geometry of APT via a bounded 3D solid geometric modeller (ROMULUS) is valid.

FIGURE 4.1: Tree structure

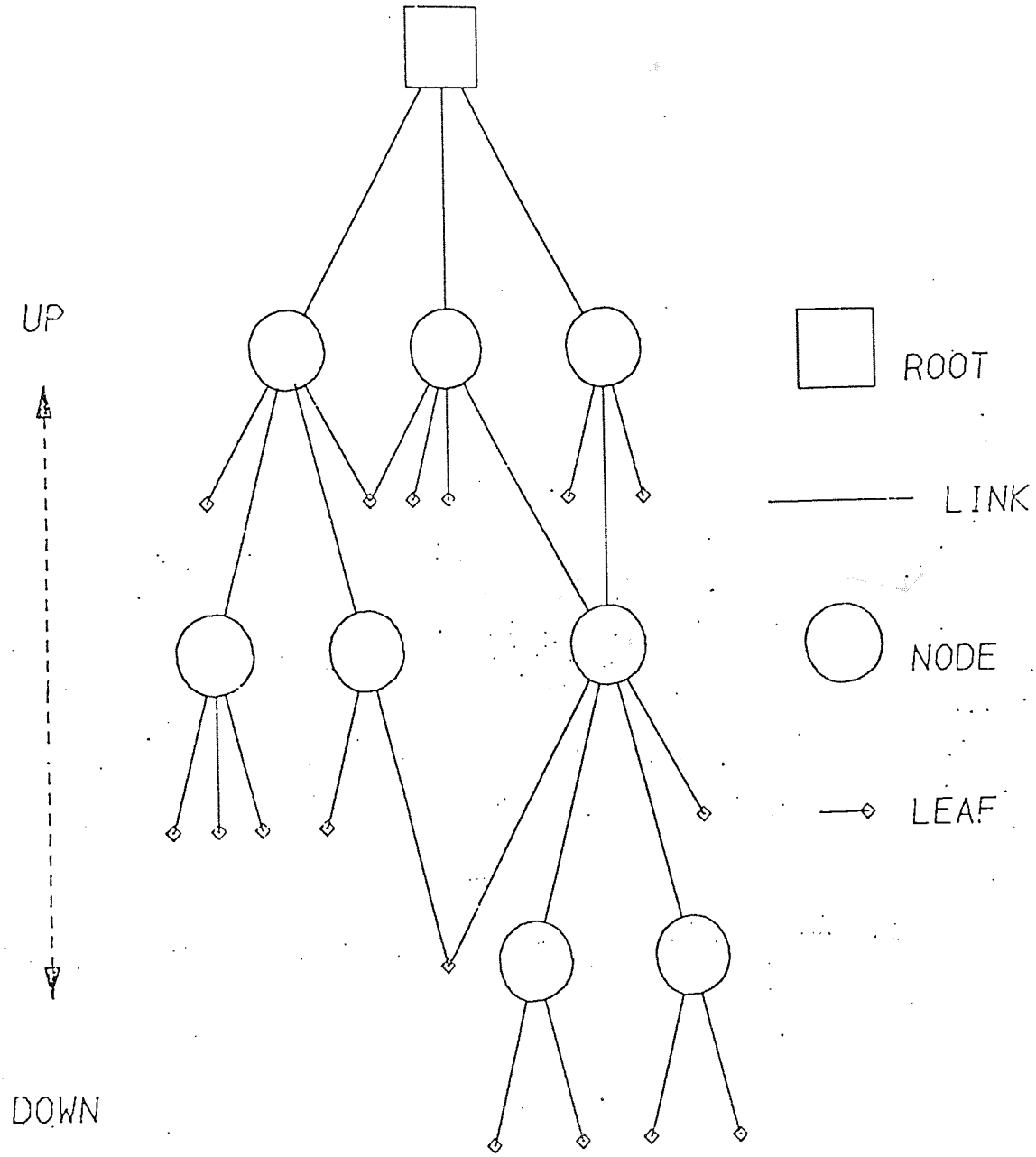


FIGURE 4 2a Gehause (hidden line view)

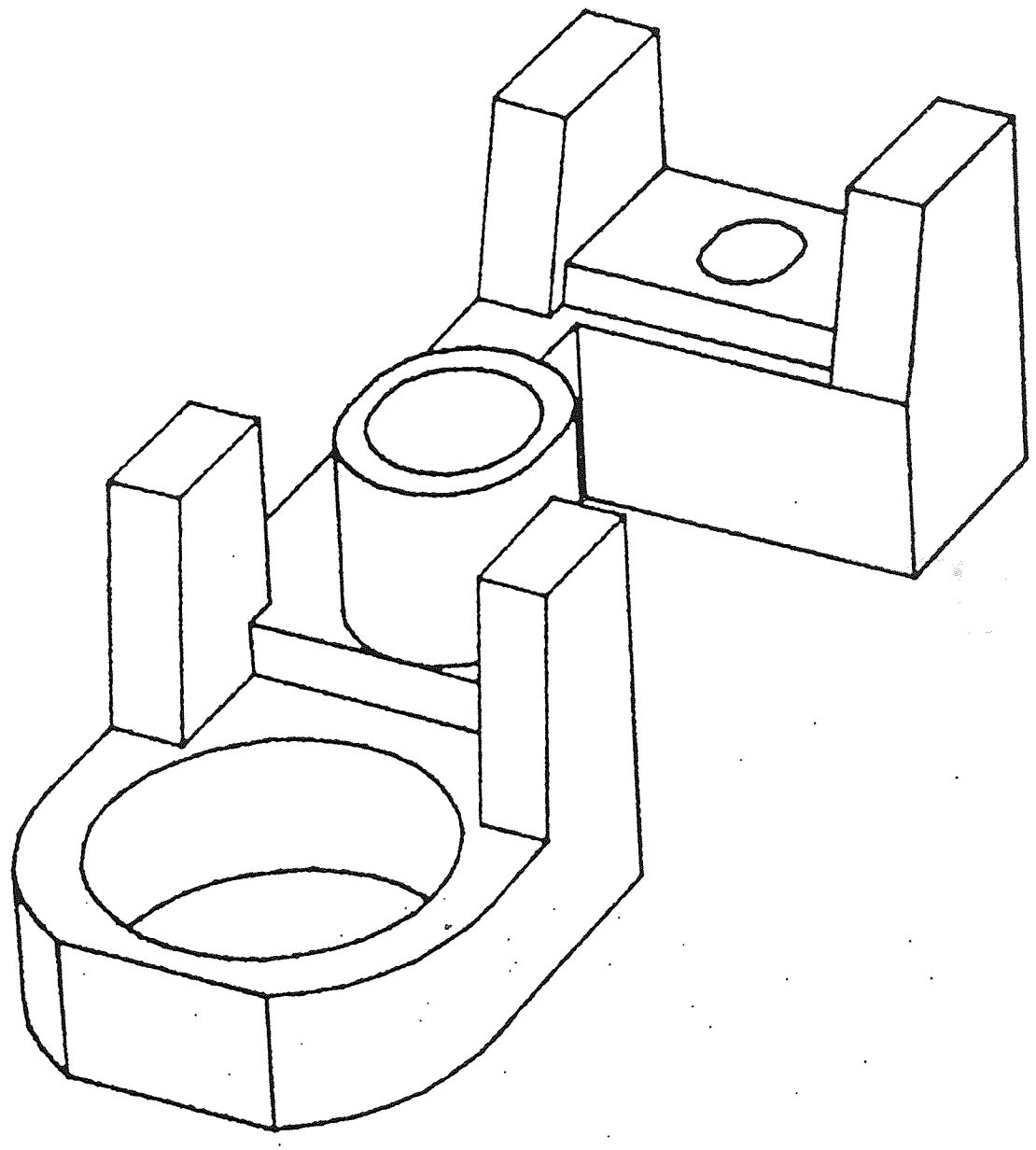


FIGURE 4.2b Gehause (orthogonal view)

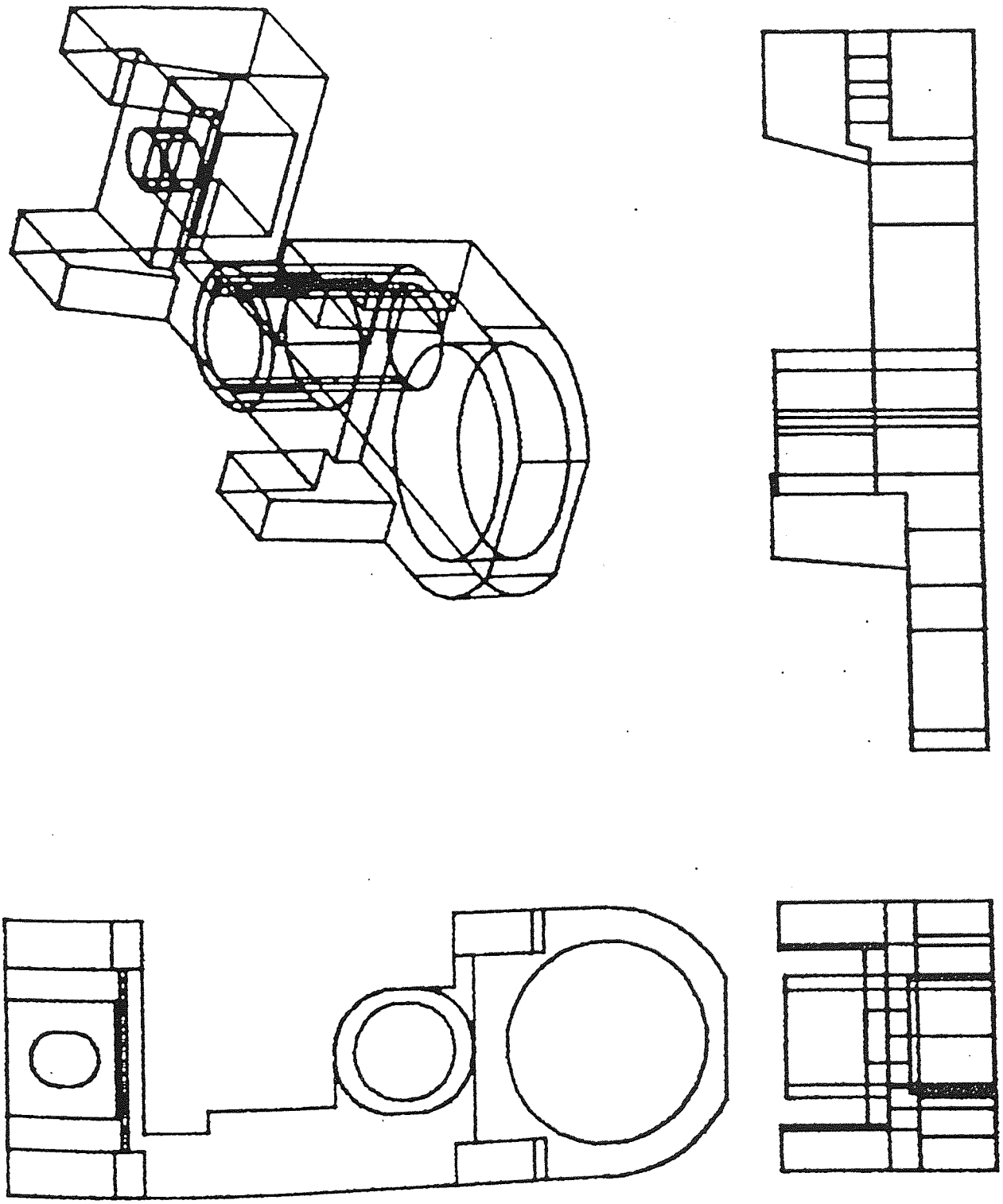


FIGURE 4.3· ROMAPT operations

```
@# MONITOR FILE CREATED  14:21 26 APR 82  BY  PTFC
@#  ROMULUS  VERSION 3.3
 OPTION TERMINAL 4010
@# ACCESS THE FILE CONTAINING THE GEOMETRIC MODEL : GEHOUSE
 GET ROMTEMP>BODY.G
@# LIST BODIES
 LIST
@# GENERATE NAMES FOR POINT/EDGE/FACE OF THE MODEL IF NOT
@# ALREADY DONE SO BY THE PROGRAM
 GENERATE GEHOUSE
@# SKETCH A VIEW OF THE BODY
 SKETCH
@# PLOT A HIDDEN LINE VIEW OF GEHOUSE  FIGURE 4a
 HLPLOT
@# PLOT A 3D ENGINEERING (ORTHOGONAL) VIEW OF GEHOUSE : FIGURE 4b
 OPTION ORTHO
 SKPLOT
@# MONO VIEW
 OPTION MONO
 SKETCH
@# USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
@# PLOT THE WINDOW VIEW : FIGURE 5a
 SKPLOT
@# LABEL ENTITIES INTERESTED
 APTLABEL FACE
@# PLOT THE FACE TOP WITH LOOPS LABELLED : FIGURE 5b
 SEPLOT TOP
 SELECT TOP LOOP
 APTLABEL TOP
@# PLOT THE FACE TOP WITH POINTS/EDGES ON LOOP 1 : FIGURE 5c
 SEPLOT TOP
 APTLABEL POINT TOP LOOP 1
 APTLABEL EDGE TOP LOOP 1
@# RELEASE THE WINDOW
 VIEW RELEASE
 SKETCH
@# SELECT THE BOTTOM FACE BASE
 SELECT BASE
@# PLOT THE FACE BASE WITH LOOPS LABELLED : FIGURE 6a
 SEPLOT BASE
 SELECT BASE LOOP
 APTLABEL BASE
@# PLOT THE FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIGURE 6b
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
@# USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
@# PLOT THE ENLARGED FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIG. 6c
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
@# THE ABOVE PROCEDURE CAN BE REPEATED FOR OTHER FACES
@# UNTIL THE GEOMETRY OF THE PART IS UNDERSTOOD . THE USER CAN PROCEED TO..
@# WHEN THE GEOMETRY IS UNDERSTOOD , THE USER CAN PROCEED TO..
@# OUTPUT THE APT GEOMETRY SELECTIVELY AS IN THE FOLLOWING : FIGURE 7b
 OUTPUT APTDATA
 APTENQUIRE BASE
 APTENQUIRE POINT BASE LOOP 1
 APTENQUIRE EDGE BASE LOOP 1
 OUTPUT
@# STOP KEEP
 OK.
```

FIGURE 4.4a Cutter path generated for the Gehause in the x-y plane

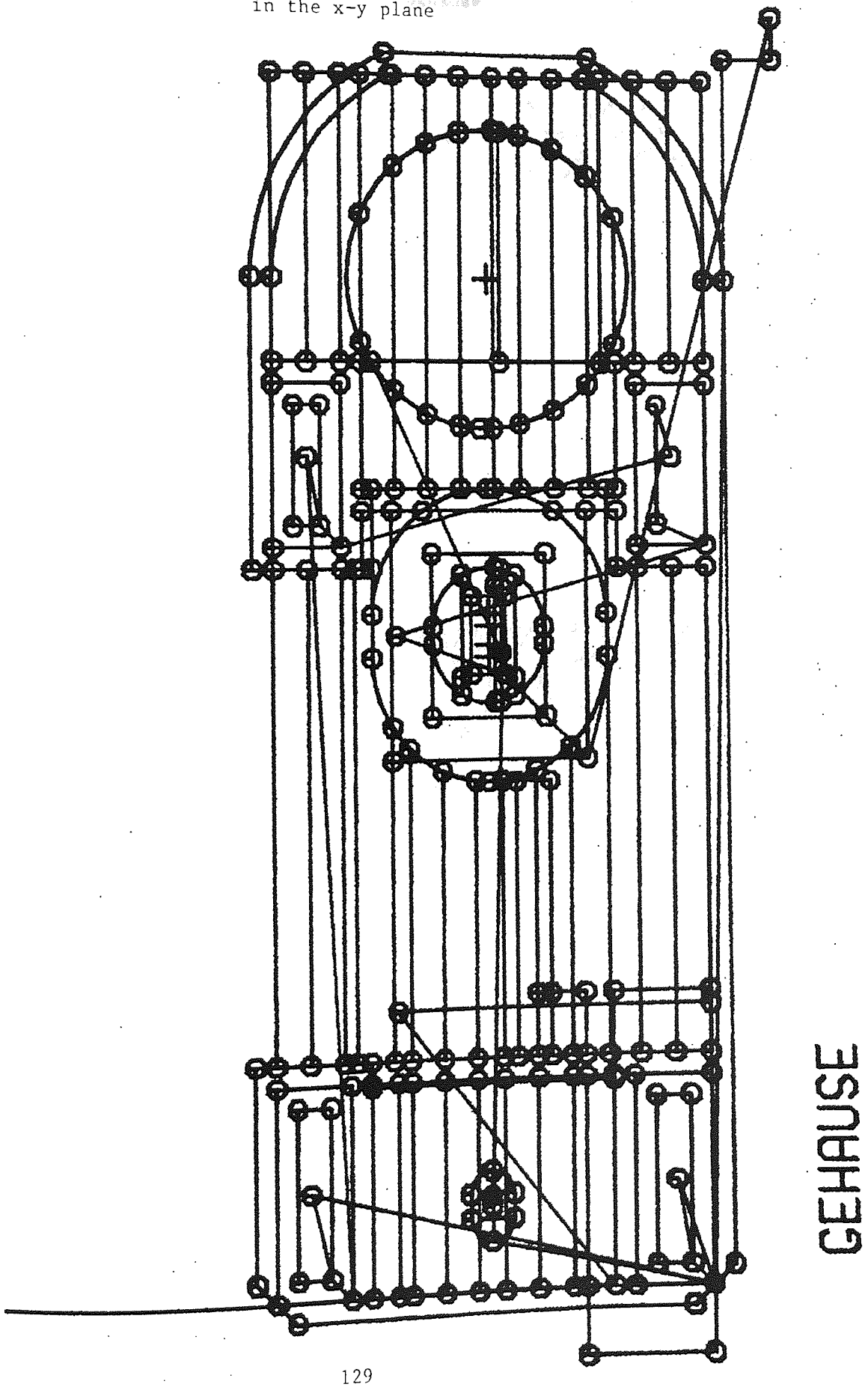
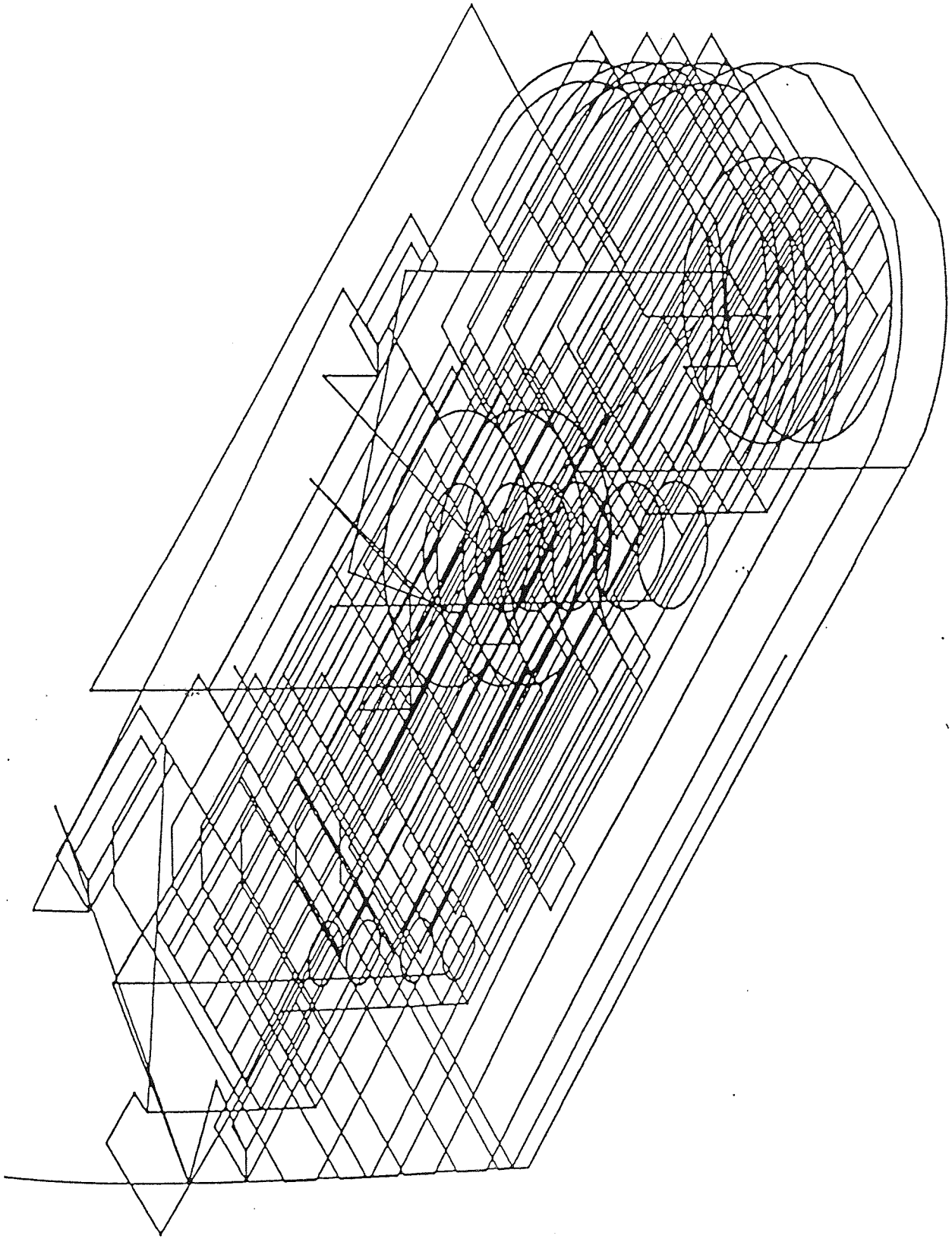


FIGURE 4.4b: Perspective view of the cutter path generated for the Gehause



5. AIAPT

Although ROMAPT is a working system, it suffers the inconvenience of only functioning within the ROMULUS modeller. In order to have a modeller-independent version of ROMAPT, the concept of AIAPT is described in the following.

The Geometric Modelling Project of CAM-I have developed specifications for a set of FORTRAN subroutines to act as an Application Interface (AI) to geometric modellers (see Section 2.6). This specification is available as CAM-I Report R-80-GM-04 and Addenda. The interface effectively shields application programs from an underlying geometric modeller. AIAPT will perform the same functions as ROMAPT but instead of calling ROMULUS routines calls AI routines instead, thus freeing the program from any modeller dependancies. It should then be possible for AIAPT to work with any geometric modeller that supports a boundary description provided that there is an implementation of the AI available for the modeller.

In Lucas, there is also an implementation of an experimental database (LEDB) with some geometric modelling driving functions for graphical display. A large portion of the CAM-I Application Interface (AI) subroutines have been implemented on this database. Using this database as a test bed, the AIAPT implemetation can then be developed and tested.

5.1 AI NOMINAL BOUNDARY MODEL

The AI nominal boundary model for solids (see Section 2.6.1) can be described as a graph of vertices and edges situated in a surface toplogical equivalent to a sphere.

The edges are connected in an ordered manner in Loops. The Loops bound portion of the surface into Faces. Faces of a surface are grouped into a Shell. A real object may consist of one outer shell with optionally one or more inner shells which representing voids.

The datastructure of the AI nominal boundary model is shown in Figure 2.3.

5.2 DEVELOPMENT OF AIAPT

Graphic manipulation facilities (i.e. viewing, windowing and transformation like scaling, shifting and rotation) very similar to ROMAPT (see Section 4.3.3) are implemented for AIAPT.

Similarly, the three APT geometry generating facilities (i.e. APTALL. APTOPTIMAL and APTENQUIRE) have also been implemented in AIAPT.

The algorithms employed are described in the following.

5.3 ALGORITHMS FOR AIAPT

The algorithms for accessing the AI nominal datastructure is similar to those of ROMAPT (see Section 4.3.8.2) except that a new topological element Shell is included.

A) APTALL (AI)

1) Start at the object.

2a) If all shells are marked, go to step (11).

2b) Go to an unmarked shell, mark the shell.

2c) If all faces are marked, go to step (2a).

3) Go to unmarked face of the shell, output the corresponding surface geometry and mark the face.

4) Go to an unmarked loop of the face, mark the loop.

5) If all vertices in the loop are marked, go to step (7).

6) For the current loop, go to an unmarked vertex (via an edge),

output the corresponding point geometry and mark the vertex, go to step (5).

7) If all edges in the loop are marked, go to step (9).

8) For the current loop, go to an unmarked edge, output the corrresponding track geometry, mark the edge.

9) If all loops in the face are marked, go to step (2c).

10) Else go to step (4).

11) Unmarked all entities.

12) Stop.

B) APTOPTIMAL (AI)

APTOPTIMAL attempts to minimize the APT geometry required for 2 1/2D parts. All planes with surface normal orthogonal, and cylinders with axis in parallel to the orientation of the tool axis are ignored.

The algorithm for APTOPTIMAL is similar to that of APTALL except that step (3) is modified as in the following:

1) Start at the object.

2a) If all shells are marked, go to step (11).

2b) Go to an unmarked shell, mark the shell.

2c) If all faces are marked, go to step (2a).

3) Go to an unmarked face, mark the face. If the face is a plane and the surface normal is orthogonal or the face is a cylinder with the axis parallel to the tool axis, then ignore the face, go to step (2c). Else output the corresponding surface geometry of the face.

4) Go to an unmarked loop of the face, mark the loop.

5) If all vertices in the loop are marked, go to step (7).

6) For the current loop (via an edge), go to an unmarked vertex, output the corresponding point geometry and mark the vertex, go to step (5).

7) If all edges in the loop are marked, go to step (9).

8) For the current loop, go to an unmarked edge, output the corrresponding track geometry, mark the edge.

9) If all loops in the face are marked, go to step (2c).

10) Else go to step (4).

11) Unmarked all entities.

12) Stop.

C) APTENQUIRE (AI)

The algoritm for APTENQUIRE is similar to APTALL except in two aspects:

a) The user supply the name of the face, loop, edge or point require.

b) If the direction of the axis of a circle is orthogonal to the tool axis, the circle is converted into a straight line defined by the projection of the circle onto a horizontal plane.

The algorithm for APTENQUIRE is given in the following:

1) Start at the object.

2a) If all shells are marked, go to step (10).

2b) Go to an unmarked shell, mark the shell.

2c) If all faces are marked, go to step (2a).

3) Go to an unmarked face, mark the face. If the face is the one specified, output the surface geometry if required. If more face required, go to step (2c). If no more entity is required, go to step (10). Else go to step (4).

4) If all loops in the face are marked, go to step (3).

5) Go to an unmarked loop in the face, mark the loop. If the loop is the one required, go to step (6). Else go to step (4).

6) If all vertices in the loop are marked, go to step (8).

7) For the current loop (via an edge), go to an unmarked vertex, mark the vertex. If the vertex is the one specified, output the corresponding point geometry. If more vertices are required, go to step (6). If no more entity required, go to step (10). Else go to step (8).

8) If all edges in the loop are marked, go to step (5)

9) For the current loop, go to unmarked edge, mark the edge. If the edge is the one specified, output the corresponding geometry. If more edges are required, go to step (8). If no more entity is required, go to step (10). Else go to step (8).

10) Unmarked all entities.

11) Stop.

5.4 AIAPT RESULTS

To demomstrate the principle of AIAPT, a geometric model of a 3D object (an alternator end bracket, see Figures 7.3a) is designed by using ROMULUS. This model is translated into a XBF file via a translator. This XBF is then read by the Lucas experimental database (LEDB) with AI implemented. As a result, a nominal boundary model of the 3D object is generated in the LEDB. AIAPT is then used to generate the APT geometry source statement for the nominal boundary model held in the LEDB.

This set of APT geometry is then submitted (as the geometry part of a APT part program) for APT processing. The result is successful (see Figures 7.3b and 7.3c).

It has been demonstrated that the concept behind AIAPT i.e. the communication of application program (AIAPT) via AI to any underlying geometric modeler is valid.

It further demonstrated the portability of AIAPT in that it can be used to generated APT geometry with any geometric modeller that

supports a boundary model representation provided that an AI implementation is available.

As the AI specification is presently only partially implemented, further development work will be implemented in ROMAPT as there is more proven software available from the ROMULUS library. However, the algorithms developed for ROMAPT can be readily transferable to AIAPT (as demonstrated) when the AI is fully implemented.

6. NC AUTOMATION (ROMAPT STAGE 2)

The previous sections have established a valid link between a geometric modeller and APT at the geometry level. The following describes how this link can be extended to provide automatic generation of NC cutter path.

6.1 CONCEPT OF MANUFACTURING MODEL

As geometric modeller was first developed, it was charged with the objective to represent the geometrical aspect of the design intent of the designer. No or little provision was made for the fact that the geometric model thus created has to be manufactured using certain manufacturing process. (Some may argue that this should be the function of application software for manufacturing.) Perhaps this is one of the reasons why 3D solid modeller, inspite of its enormous potentials, has not been much widely used in industry for production.

With the current state of the art of geometric modelling, the modeller would be unable to inform the designer that a certain design or construct is not manufacturable (for certain manufacturing process) or machinable (for certain machining conditions or tools). It is the task of the designer/production engineer to ensure that a certain geometric model is suitable for manufacture. Hence, the concept of designing a manufacturing model that is machinable (i.e. design for manufacture).

An anology in conventional manual engineering practice exits. A design usually has to go through the loop of design and production a number of times before a final production design is attained. Similarly in geometric modelling, facilities should be provided for the designer/production engineer to modify the design to generate a manufacturing model suitable for production. This tpye of modification is usually local and is known as tweaking in geometric modelling.

Before a part is made, it usually goes through a process planning stage (see Section 4.2). The function of process planning is to detail how the part is to be manufactured depending on the various features of the part.

A feature that is manufacturable is termed a manufacturing feature. A manufacturing model can be defined as being made up of a number of maunfacturing features. In NC processing, a machinable manufacturing feature is called a machinable feature.

6.2 MACHINING FEATURES

The machining model MM is defined as a geometric model made up of a number of machining features. A machining feature is defined as an entity of interest for numerical control. Various machining features are described in the following:

1) POINT feature

The point feature relates to point-to-point operation of NC machining usually connected with hole drilling or tapping.

2) PROFILE feature

The profile feature relates to the boundary of a face, which is usually made up of a number of edges bounded in an ordered manner (i.e. a loop entity with a bounded face). This feature usually associated with the profiling operation of NC machining.

3) FACE feature

This feature relates to a bounded region (e.g. a face) of a surface. This feature usually relates to the pocketing facility of NC machining.

The above three types of machining features will be useful for NC applications for precision casting parts (where only a skin of the part need to be machined) and finishing purposes. For roughing or NC operations on a blank, the volume feature is required.

4) VOLUME FEATURE

The volume feature is a portion of that volume need to be removed from the blank or stock to give the part desired (i.e. volume =

stock - part). This feature usually relates to area or volume clearance operation of NC machining.

6.3 MACHINABLE MODEL

A machinable model can then be defined as a geometric model made up of a number of machinable features.

$$MM = \sum_i Fmi$$

6.4 MACHINING STRATEGY

There are two approaches to execute machining strategy.

6.4.1 VOLUME APPROACH (DELTA-VOLUME)

This method uses the concept of the volume feature (see Section 6.2) and is used by the CAM-I ANC Project (38). The model of the volume to be removed is the difference between the stock model and the finished part model. This volume is partitioned into a number of machinable delta volumes each with its machining strategy calculated. When all delta volumes have been machined, the

required part is produced.

6.4.2 POINT-PROFILE-FACE APPROACH (REVERSE)

This method uses the point, profile and face features (see Section 6.2) to determine the final cut strategy (according to surface and tolerance requirements). The movement of the tool along cutting strategy will generate a machining volume. This volume is added to the finished part and the process is repeated until the expanded part model is equal to the stock. The part is then machined in reverse order.

6.5 THEORY OF AUTOMATIC MACHINING

Early works of Woo (40) and Grayer (37) on automatic NC machining of geometric models concerned only with 2 1/2 D parts. For 3D parts, it is a subject under intensive research (41).

As formal theory for computer aided automatic machining of 3D solid geometric model has not yet existed and as reported work in this area is nearly non-existing, the author has no choice but to invent his own theory. The theory is developed for boundary representation solid modeller. The current trend is such that constructive solid modeller (CSG) will provide boundary

representation, so that theory developed here will be applicable for both types of geometric modellers.

The function of NC programming is to convert the manufacturing features defined above (see Section 6.1) into specific machining strategies (Si). Therefore, a machinable model (AMM) that can be machined automatically can be defined as:

$$AMM = \sum_{i} Li\ Si\ Fmi$$

where,

Fmi: is the machinable feature.

Si: is the specific machining strategy.

Li: is the link between consecutive Si.

For i = 0, the term L(0) S(0) Fm(0) is the starting point of the cutter. For n machinable features, the term L(n+1) S(n+1) Fm(n+1) is the finishing point of the cutter (i.e. when i = n+1).

6.6 INFORMATION REQUIREMENT

Whether a part is machinable automatically or not depends on its features. To compute the machinability of features, the following

information is required. (The information indicated below is not an exhausting list but merely an indication of what is require.)

6.6.1 GEOMETRY INFORMATION

The following information is needed:

1) The kind of geometric entities (i.e. point, track, surface or volume).

2) The co-ordinates of points.

3) The type and equation of track (i.e. line, circle), surface (i.e. plane, cylinder, cone, etc) or volume (i.e. cuboid, sphere etc).

4) The direction of track and surface normal for faces (at a given point).

5) The intersection of geometric entity A with geometric entity B.

6.6.2 TOPLOLOGY INFORMATION

The following topological information is needed:

1) The kind of topological entity (i.e. vertex, edge, loop, face, shell and object).

2) The geometry entity attached to the topological entity.

3) The point co-ordinate of a vertex.

4) The vertices of an edge.

5) The vertices of a loop.

6) The edges of a loop.

7) The adjacent loops of an edge.

8) The loops of a face.

9) The faces of a shell.

10) The shells of an object.

11) The shape of track or surface (i.e concave, convex or smooth at a given point).

The details of accessing a boundary geometric model is given in Section 4.3.6.

6.7 GENERAL MACHINING FUNCTIONS

The following general machining functions are developed to access, and process information from, a geometric model selectively and systematically so as to compute efficiently whether a specific feature is machinable or not.

These functions are designed to operate on boundary geometric models. The associated algorithms for these functions are initially developed to operate on ROMULUS boundary models. As demonstrated in Chapter 5, these functions can be implemented in the AI environment. Hence, different boundary models can be handled via AI.

The algorithms given below are simple implementation of the functions. More sophistications can be incorporated, if required, for more complicated objects.

6.7.1 MACHINING CONDITION Mc

Some of the following functions are assumed to take place under certain machining condition Mc:

Mc (r1, r2, v, f, s, m, l, x) = 0 (not machinable)

= 1 (machinable)

where,

r1: is the radius of the cutter

r2: is the distance from the cutter end to the centre of the fillet cutter

(Note: r1, r2 can be replaced by the formal cutter definition).

v: is the cut vector

f: is the feed rate of the cutter

s: is the revolution per minute (r.p.m) of the cutter

m: is the material of the work-piece

l: is the coolant

x: any other factors (i.e. tear and wear of tool)

6.7.1.1 ALGORITHM FOR MACHINING CONDITION Mc

The algorithm for the machining condition Mc is as follows:

1) Set machine flag M = 0 (i.e. not machinable).

2) Read and check tool definition (r1, r2) against the tool library. If tool not acceptable, stop.

3) Read material (m) and check against the material file. If material not acceptable, stop.

4) Read feedrate (f) and check against the feedrate table. If feedrate not acceptable, stop.

5) Read r.p.m. speed (s) and check against the speed table. If not acceptable, stop.

6) Read the coolant (l), and check against the coolant table. If coolant not acceptable, stop.

7) Read any other parameters and check against the relevant tables. If not acceptable, stop.

8) Else set machine flag M = 1 (i.e. machinable condition).

The necessary libraries, files and tables are usually obtained from the machine shop, recommendations from the machine tool manufacturers or based on rules of thumb. It is assumed here that, in practice, all machining parameters are provided via the process planning function.

The rest of the functions concentrate on the geometrical aspects of machining.

6.7.2 HOLE FUNCTION Fh

The point feature can be described as either machinable or not by the hole function:

Fh (r1,p1,r2,p2,v1,r,face,Mc) = 0 not machinable

1 machinable

where,

Mc: is the machining condition

face: is the cylindrical face defining the hole

r: is the radius of the cylinder

v1: is the vector defining the axis of the cylinder

r1: is the radius of the 'top' circle of the cylinder

p1: is the centre of the 'top' circle

r2: is the radius of the 'bottom' circle

p2· is the centre of the 'bottom' circle

The Hole Function is equal to 1 (i.e. machinable) when r=r1=r2 and face is cylindrical. The function is 0 (i.e. not machinable) when r1 not equal r2 and/or that face is not cylindrical.

6.7.2.1 ALGORITHM FOR HOLE FUNCTION

The algorithm for the hole function Fh is as follows:

1) Set machining flag M = 0 (i.e. not machinable).

2) Identify the face where hole is required. If the surface defining the hole is not a cylindrical type, stop.

3) Identify the two circular edges of the cylindrical face.

4) If the radii of the two circular edges are not equal, stop.

5) Else, set M = 1 (i.e. machinable).

6) Stop.

6.7.3 CONNECTIVITY FUNCTION Fc

The connectivity function Fc defines the type of connectivity between two consecutive edges in a given loop of a face.

Fc (vx,e1,e2,face,sense) =

0 no connection

1 line/line

2 line/circle

3 circle/line

4 circle/circle

5 any others

where,

vx: is the node between edge e1 and edge e2

e1: is the edge preceeding vx

e2: is the edge following vx

face: is the face where connection takes place

sense: is the sense of direction of the loop containing edges e1 and e2.

The connectivity depends on the instrinsic geometrical properties of the edges. (Type 5 of Fc can be defined in terms of the two intersecting surfaces.)

6.7.3.1 ALGORITHM FOR CONNECTIVITY FUNCTION

The algorithm for the connectivity function Fc is as follows:

1) Set connectivity M = 0 (i.e. no connection).

2) Identify the face (i.e. the part surface).

3) Identify the loop.

4) Identify the edge (E1) and get its track geometry type.

5) Identify the next edge (E2) and get its track geometry type.

6) Set M according to the following combination of connectivity types (for E1 and E2):

1: straight line/ straight line.

2: straight line/ circle.

3: circle/ straight line.

4: circle/ circle.

5: any other type combinations.

7) Stop

6.7.4 INTERSECTION FUNCTION Fi

The intersection function defines the classes of intersection within each connectivity types.

Fi (Fc) = 0 co-incidence
1 no intersection
>1 classes of intersection

where,

Fc: is the connectivity function

The intersection class depends on the number and manner of intersections between the two edges.

1) LINE/LINE

The intersection class of the line/line case is defined as:

1 no intersection
2 one intersection
3 one 'down' intersection
4 one 'up' intersection

The 'up' or 'down' intersection will affect the direction of the cutter to turn 'left' or 'right' (see Section 8.2.6).

2) LINE/CIRCLE

The intersection class of the line/circle and circle/line cases are defined as:

1 no intersection

2 one tangential intersection

3 two intersections

3) CIRCLE/CIRCLE

The intersection class of the circle/circle case is defined as:

1 no intersection

2 one tangential intersection

3 two intersections

6.7.4.1 ALGORITHM FOR INTERSECTION FUNCTION

The algorithm for the intersection function Fi is as follows:

1) Set intersection flag I = 1 (i.e. no intersection).

2) Get the connectivity flag M from the connectivity function Fc. If M = 0 stop

3) Identify the two edges and their track geometries.

4) Calculate intersections, if any, of the two edges.

5) If no intersection, stop

6) If co-incidence, set I = 0 and stop.

7) If one intersection, set I = 2.

8) If M = 1, go to step (10).

9) If two intersections, set I = 3, stop.

10) If one 'down' intersection, set I = 3, stop.

11) If one 'up' intersection, set I = 4, stop.

12) Stop.

6.7.5 RAISE FUNCTION Fr

The raise function defines whether an adjacent face f2 of the

selected part surface f1 is a raise or not.

Fr (r,vx1,vx2,e,f1,f2) = 0 no raise

1 raise

where,

r: is the radius of the cutter specified

e: is the boundary edge between adjacent faces f1 and f2.

vx1: forward vertex of edge e

vx2: backward vertex of edge e

f1: the selected part surface

f2: the adjacent face to f1

An adjacent face f2 is a raise, if an intersection occurs between face f2 and a cutter of radius r, such that it is placed on the boundary edge with the cutter axis along the surface normal of the selected part surface f1.

6.7.5.1 ALGORITHM FOR RAISE FUNCTION

The algorithm for the raise function Fr is as follows:

1) Set the raise flag N = 0 (i.e. not raise).

2) Read the cutter radius r.

3) Identify the part surface f1.

4) Identify the adjacent face f2. If f2 is coplanar to f1, stop

5) Identify the edge e common to both faces.

6) Calculate the track geometry of the edge e.

7) Calculate the mid-point p of the track.

8) Calculate the surface normal n of the part surface f1 at the point p.

9) Place an imaginary cutter of radius r at the mid-point p of the track such that the cutter axis lies along the surface normal n of the part surface f1.

10) Calculate the track geometry of the circle c with centre at the mid-point p and radius r of the cutter.

11) Calculate the intersections, if any, between circle c and adjacent face f2.

12) If no intersection, stop.

13) If there is intersection, set N = 1 (i.e. raise face)

14) Stop.

6.7.6 LOOP FUNCTION Fl

The loop function is a function of Mc, Fc, Fi, and Fr, which defines whether a profile feature is machinable or not.

Fl (Mc, Fc, Fi, Fr) = 0 not machinable
1 machinable

where,

Mc: is the machining condition

Fc: is the connectivity function

Fi: is the intersection function

Fr: is the raise function

A loop is machinable when both the connectivity and intersection

functions of its edges can be defined properly and that any adjacent raise face is not inclined more than 90 degree towards the centre of the loop.

6.7.6.1 ALGORITHM FOR LOOP FUNCTION

The algorithm for the loop function Fl is as follows:

1) Set machining flag M = 0 (i.e. not machinable).

2) Identify the part surface fl.

3) Identify the loop and all edges in the loop.

4) If machining condition Mc not acceptable, stop.

5) Calculate the connectivity function Fc of all edges in the loop.

6) Check the connectivity function of each edge pair. If not acceptable (i.e. no connection), stop.

7) Calculate the intersection function Fi of all edges in the loop.

8) Check the intersection function of each edge pair. If not acceptable (i.e. no intersection or co-incidence), stop.

9) Identify all raise faces in the loop.

10) If any of the raise face is inclined more than 90 degree towards the centre or interior of the loop, stop.

11) Set M = 1 (i.e. machinable)

12) Stop.

6.7.7 CUT FUNCTION Cf

The cut function defines whether a cut vector is machinable or not.

Cf (Mc, v, Fr1, Fr2,.....Frn-1, Frn) = 0 not machinable

1 machinable

where,

Mc: is the machining condition

v: is the cut vector

Fr1, Fr2 : raise functions (in pairs) for the 1st loop

Fr(n-1), Frn : raise functions (in pairs) for the n/2th loop.

n: is an even number denotes number of intersections between the cut vector v and the loops of the face.

A cut vector is machinable when it forms a proper intersection of the part surface.

6.7.7.1 ALGORITHM FOR CUT FUNCTION

The algorithm for the cut function Cf is as follows:

1) Set cut function flag M = 0 (i.e. not machinable).

2) Identify the part surface f1.

4) Identify the cut vector v. If v is not acceptable, stop.

5) Calculate the intersections of the cut vector v and the face f1.

6) If there is no intersection, stop.

7) For each intersection point, identify the adjacent face and its raise function.

8) If any raise function is not acceptable, stop.

9) Else, set M = 1 (i.e. machinable)

10) Stop.

6.7.8 REGION FUNCTION Rf

The region function, which is a function of Cf, defines whether a face feature is machinable or not.

Rf (Cf) = 0 not machinable

1 machinable

where,

Cf: is the cut function

A region is machinable when all associated cut vectors for that region are machinable.

6.7.8.1 ALGORITHM FOR REGION FUNCTION

The algorithm for the region function is as follows:

1) Set the region function flag M = 0 (i.e. not machinable)

2) Identify the part surface fl.

3) Calculate all cut vectors required for the face fl.

4) For each cut vector, calculates its cut function Cf.

5) If any cut function Cf is not acceptable, stop.

6) Set M = 1 (i.e. machinable).

7) Stop.

7. BOUNDED NC CUTTER PATH GENERATOR

The following describes the development of a bounded NC cutter path generator based on a geometric part model.

7.1. BOUNDED VS UNBOUNDED NC PROCESSORS

Existing APT based NC systems are based on the philosophy that the direction and the position of the tool is known and the problem is to 'find' the part. With bounded geometry description of the part, the problem of the NC processor becomes reversed: the desription of the part is known and the problem is one of finding the position of the tool and its direction to generate the tool path required for the part and process involved.

7.2. CONCEPT OF OFFSET

On a closer examination, the cutter path required for the bounded NC cutter path generator is the cutter offset of the part concerned. In the case of a profile feature, the cutter path is the cutter offset of a loop of a face. For a face feature, the cutter path is the cutter offset of the face.

As for milling operations, the machining requirement usually

consisted of a combination of profiling and pocket milling. The following concentrates on the development of functions required for automatic profiling and face milling facilities.

7.2.1 POINT OFFSET FUNCTION Op

The offset function of a point, Op, is defined as the cutter offset of the point along the surface normal, nl, of the part surface, fl, such that, it does not violates check surfaces f2, f3 with surface normals n2, n3.

7.2.1.1 ALGORITHM FOR POINT OFFSET FUNCTION

The alogorithm for the point offset function is as follows:

1) Identify the part surface fl and calculate its surface normal nl.

2) Locate the point pl interested.

3) Identify the adjacent faces around the point p at a distance equal to the cutter radius r.

4) If no adjacent face is identified, go to step (11).

5) Calculate the raise function Fr for all adjacent faces identified above.

6) If there is more than 2 raise faces, stop.

7) Identify the two raise faces f2, f3 and the intersection points p2, p3 from step (3).

8) Calculate the surface normal n2, n3 of faces f2, f3 at points p2, p3.

9) If there is one raise face f2, offset point p1 (from p2) by r along the surface normal n2 of face f2. Go to step (11).

10) If there are 2 raise faces f2 and f3, offset point p1 from p2 and p3 by r along the surface normal n2 and n3 of faces f2 and f3.

11) If the cutter is ball-ended, offset point p1 by the cutter radius r along the surface normal n1 of the part surface f1.

12) Stop.

7.2.2 TRACK OFFSET FUNCTION

The track offset function, Ot, is defined as the cutter offset of that track by the cutter radius along the surface normal n1 of the

check surface f1.

7.2.2.1 MATHEMATIC FOR OFFSET FUNCTION

A straight line L can be represented by a point P and a vector V, i.e.

$$L = f(P, V)$$

If the cutter radius r and direction N of offset are given, then the track offset L′ for the line L is given by:

$$L' = f(P', V)$$

where $P' = P + r * N$

A circle C can be represented by a centre point P and a radius R, i.e.

$$C = f(P, R)$$

The track offset C′ for the circle C is:

$$C' = f(P, R + r)$$

An ellipse E can be represented by its centre P and its two radii,

r1 and r2, i.e.

$E = f (P, r1, r2)$

The track offset function E′ for the ellipse E is:

$E' = f (P, r1 + r, r2 + r)$

The above representations are used in ROMULUS.

7.2.2.2 ALGORITHM FOR TRACK OFFSET FUNCTION

The algorithm for the track offset function Ot is as follows:

1) Identify the part surface f1.

2) Identify the edge e1.

3) Identify the adjacent face f2 and its surface normal n2 (e1 is common between f1 and f2).

4) Get the track geometry t for the edge e1.

5) Offset t by r (cutter radius) along the surface normal n2 away from check face f2.

6) Stop.

7.2.3 EDGE OFFSET FUNCTION Oe

The edge offset function, Oe, can be defined as the cutter offset of that edge, away from the check surface, f1, of surface normal, n1, and delimited by the offsets of adjacent surfaces s2 and s3 with surface normals n2 and n3.

Alternatively, the edge offset can also be expressed in terms of point offsets:

$$Oe = \sum_{i} Opi$$

7.2.3.1 ALOGORITHM FOR EDGE OFFSET FUNCTION

The algorithm for the edge offset function Oe is as follows:

1) Identify the part surface f.

2) Identify the loop in f.

3) Identify edge e1 and the adjacent check surface f1. (e1 is the

edge common to f and f1.)

4) Calculate the raise function of f1.

5) If f1 is not a raise face, go to step (7).

6) If f1 inclined more than 90 degree towards the centre or interior of the loop, stop.

7) Identify e1's forward edge e2 and forward vertex vx2 and the adjacent check surface f2. (e2 is the edge common to f and f2).

8) Check the connectivity and intersection functions between e1 and e2. if not acceptable, stop.

9) Identify e1's backward edge e3 and backward vertex vx3 and the adjacent check surface f3. (e3 is the edge common to f and f3).

10) Check the connectivity and intersection functions between e1 and e3, if not acceptable, stop.

11) Get the track equation t1 for edge e1 and offset it by r (cutter radius) along the sense of the surface normal n1 of f1 (i.e. calculate the track offset function Ot of t1).

12) Get the track equation t2 for edge e2 and offset it by r along the sense of the surface normal n2 of f2 (i.e. calculate the track offset function for t2).

13) Get the track equation t3 for edge e3 and offset it by r along the sense of the surface normal n3 of f3 (i.e. calculate the track offset function of t3).

14) Calculate the intersection P1 (nearest to vx2) between t1 and t2, and the intersection P2 (nearest to vx3) between t1 and t3.

15) If there is no intersection, stop.

16) The edge offset required is the track t1 with P1 and P2 as end points.

17) Adjust t1, P1, P2 if necessary according to the shape of the cutter.

18) Stop.

7.2.4 LOOP OFFSET FUNCTION Ol

The loop offset function, Ol, can be defined as the cutter path of that loop of the face (profile feature).

$$Ol\ (Oe) = \sum_i Oei \quad \text{iff Fl exits.}$$

where,

Oe· is the edge offset function.

The validity of Ol is established by the loop function Fl.

7.2.4.1 ALGORITHM FOR LOOP OFFSET FUNCTION

The algorithm for the loop offset function Ol is as follows:

1) Identify the part surface fl.

2) Identify the loop in fl.

3) Calculate the loop function Fl of the loop.

4) If Fl is not machinable, stop.

5) Identify all edges in the loop.

6) For every edge, calculate the edge offset function Oe.

7) If any Oe is not acceptable, stop.

8) Store all Oe in an ordered manner (in a machining file).

9) Stop.

7.2.5 CUT VECTOR OFFSET FUNCTION Oc

The cut vector offset function, Oc, can be defined as the cutter path of a cut vector. v, across a face.

The validity of Oc is established by the cut function, Cf.

7.2.5.1 ALGORITHM FOR CUT VECTOR OFFSET FUNCTION

The algorithm for the cut vector offset function is as follows:

1) Identify the part surface fl.

2) Get the cut vector V.

3) Calculate the cut function Cf of V.

4) If Cf is not machinable, stop.

5) Identify all intersection points Pi between V and fl.

6) For every intersection point Pi, calculate the point offset function Opi.

7) If any Opi is not acceptable, stop.

8) Store all Opi in an ordered manner together with the linking information between successive Opi.

8) Stop.

7.2.6 REGION OFFSET FUNCTION Or

The region offset function, Or, can be defined as the cutter pattern for the selected part surface (face feature).

$$Or\ (Oc) = \sum_i \quad Oci \qquad \text{iff } Oci \text{ exits}$$

where,

Oc: is the cut vector offset.

The validity of Or is established by the region function, Rf.

Alternatively, as a region can be machined by cutter path pattern of a profile feature:

$$Or = \sum_j \quad Olj \qquad \text{iff} \quad Olj \text{ exists}$$

7.2.6.1 ALGORITHM FOR REGION OFFSET FUNCTION

The algorithm for the region offset function Or is as follows:

1) Identify the part surface f1.

2) Calculate the region function Rf of f1.

3) If Rf is not machinable, stop.

4) Calculate all cut vector offset functions Oci for the part surface.

5) Store all Oci in an ordered manner together with the linking information for successive Oci (in a machining file).

6) Stop.

7.3 AUTOMATIC CUTTER PATH GENERATOR FOR 2D AND 2 1/2D PARTS

Using the above concept, the automatic cutter path generator for 2D and 2 1/2D parts is developed. To demonstrate the applicability of the above theory, an engineering test part has been created by using ROMULUS and then the geometric model is machined by using ROMAPT to generate the cutter path automatically (see Chapter 9).

The automatic generation of cutter paths is described in the following.

7.3.1 MACHINING OF HOLE

The cutter path for the machining of the hole defined by a circular edge can be generated by the command:

HOLE [edge 1.....edge n] [tool] [radius]

where,

[edge]: name of the circular edge.

[tool]: options for either flat-end or ball-end cutter.

[radius]: radius of cutter.

The HOLE command uses the hole function Fh (see Section 7.2.3).

The machining strategy S built in the HOLE command is to start machining from the centre of the top circle plus offset to the centre of the bottom circle plus offset.

7.3.1.1 ALGORITHM FOR HOLE

The algorithm of HOLE is given in the following:

1) Identify the selected face.

2) Identify the top and bottom circular edges C1 and C2 and their centres P1 and P2 (of the selected hole).

3) Define the cutter radius r so that it equals the radius of C1 (and C2).

4) Start machining from P1 to P2.

5) Move cutter from P2 to P1

6) Output and display the cutter path.

7) Repeat steps (2) to (6) for other holes specified.

8) Stop.

7.3.2 MACHINING OF A LOOP (PROFILE FEATURE)

The cutter path for a profile can be generated automatically by the command:

PROFILE [edge1, edge2...] [LOOP] [tool] [radius] [options]

where,

[edge]: name of edges.

[LOOP]: options for the whole loop.

[tool]: options for either flat-end or ball-end cutter.

[radius]: radius of the cutter.

[options]: either the current selected face or user-defined Z-planes.

The PROFILE command uses both the edge offset function Oe and the loop offset function Ol (see Sections 7.2.3 and 7.2.4).

The machining strategy built in the PROFILE commands is to start machining with the start vertex of the edge specified and follows the logical direction of the loop or the order of edges specified.

7.3.2.1 ALGORITHM FOR PROFILE

The algorithm of PROFILE is given in the following:

1) Identify the selected part surface.

2) Identify the starting edge E.

3) Identify the cutter and its radius r.

4) Calculate the edge offset function Oe for E.

5) Output the offset edge in the machining file and display the cutter path.

6) Repeat steps (4) to (5) for all edges specified or all edges in the loop.

7) Stop.

7.3.3 MACHINING OF FACE (FACE FEATURE)

The cutter path for the machining of a face of the part can be generated by the command:

MACHINE [face] [edge] [tool] [radius] [options]

where,

[face]: name of face.

[edge]: name of the edge indicating the cutting direction.

[tool]: options for flat-end or ball-end cutter.

[radius]: radius of the cutter.

[options]: options for the cutter to cut TO, ON, PAST.

The MACHINE command uses both the cut vector offset function Oc and the region offset function Or.

The machining strategy built in the command MACHINE uses a zig-zag machining method. The user specified the cut vector via an edge. All intersections between the cut vector and the loops of the part surface are calculated. These intersection points are offseted appropriately by r (cutter radius) if necessary.

Machining starts on the first intersection point to the second point. The cutter then retracts to a specified height, moves over to the top of the subsequent intersection (3rd) point. Machining resumes on this point to the next (4th) intersection point. This process continues untill all intersection points (of the current cut vector) have been processed.

The next cut vector is generated from the previous cut vector (i.e. offset the previous cut vector by the cutter diameter 2r for a

flat-end cutter, or 1/5 of r for a ball-end cutter, via the track offset function Ot). The linking strategy between successive cut vector is such that the finishing point of the previous cut will be retracted to a specified height, and the cutter then moves over to the top of of the first cut of the cut vector.

The whole procedure is repeated until the whole surface is machined.

A special feature of both commands PROFILE and MACHINE is that the cutter path is displayed simultaneously on the screen as it is generated. This facility helps the user to have a visual check on the performance of the system.

7.3.3.1 ALGORITHM FOR MACHINE

The algorithm of MACHINE is given in the following:

1) Identify the selected part surface (PS).

2) Identify the cutter definition and its radius r.

3) Identify all loops in the face.

4) Identify the edge E for generating cut vectors v.

5) Calculate the region offset function Or of the PS.

6) If Or is not machinable, stop.

7) All intersections between the cut vectors and the edges in the loops of the face are calculated. (Hole loops are ignored. Cut vectors for parallel edges are appropriately adjusted.)

8) For each intersection point, check whether the adjacent face F (not the PS) is a raise or not.

9) If F is a raise, offset the intersection point (via the point offset function Op) away from the obstrusting face F to avoid collision.

10) Output and display the cutter path.

11) Stop.

7.3.4 ROUGHING

The output of the above automatic cutter path generator is a cutter location file containing the centre of the cutter in x, y, z co-ordinates (Figures 7.1). This machining data will move a cutter across the surface of the model. For machining on precision casting parts where only finishing is required, the above machining

will suffice. However, for machining on a blank block, roughing processing will be necessary.

Accordingly a roughing processor (ROUGH) is developed to generate a series of roughing passes at a specified cutting depth. The method is to reproduce the final cut but offset (via the cut vector offset function) vertically by a certain amount specified by the user. This process is repeated until the top of the block is reached. Any cut that would be made above the block is truncated to save unnecessary machining movement and time. The whole process is reversed for actual cutting (i.e. the Reverse approach of machining of Section 6.4.2).

Figure 7.2 shows a simple final machining pass with four roughing passes generated by ROUGH.

The machinig file output of PROFILE and MACHINE will have one or more separate machining sequences. Each machining sequence will contain a header record, the machining data record and an end record.

The header record contains the cutter indentifier, cutter diameter, spindle speed (via default values set for the Lucas Model Making machine) and initial setting position and the the height (along z-axis) of the block from which machining is to be made. The machine data record contain the cutter location of the centre of the cutter in x, y, z co-ordinates. The end record is a terminator.

If all information in the header record are the same for various sequences, these sequences will be combined into one and a single set of rough passes will be produced from them all. If all information in the header record of sequence is different from the previous sequence, then all previous sequences are grouped together and roughing passes generated and output before this present and sebsequent sequences are dealt with. The user can also re-output the input sequences combined into a single output sequence (i.e. editing the sequences) or not generating roughing passes for a particular set of sequences (i.e. a hole sequences).

The command to use the rough processor is as follows:

ROUGH input output

where,

input: is the file containing the machining data generated by PROFILE or MACHINE.

output: is the file in which the resultant roughing passes are written into.

7.3.5 DISPLAYING OF MACHINING DATA

After the generation of roughing passes, it is desirable to have some kind of graphic check on the machining data. The DISPLAY processor is therefore developed.

The command to use the DISPLAY processor is as follows:

DISPLAY input

where,

input: is the machining data file.

The DISPLAY processor allows the user to view the machining data (both the roughing and final passes) by the following facilities:

OBJECT: to read the machining data.

VIEW: to see the perspective view of the machining sequences.

VIEW 3: to see the perspective view and the three orthogonal projections of the machining sequences.

ROTATE: to rotate the dview.

WINDOW: to window on portion of the machining sequences.

7.3.6 MANIPULATION OF THE CO-ORDINATE SYSTEM

If for some reasons, the co-ordinate system of the machining data is found to be not suitable for manufacture. The change of the co-ordinate system for the machining data can be effected by the following processor:

MODIFY input output

where,

input: is the machining data file.

output: is the machining data file with the new co-ordinate system. The user can change the co-ordinate system by input the new co-ordinate directions. For example, replacing X' by -Z, Y' by +Y and Z' by +X.

7.3.7 MACHINING PROCESSING

The machining data file can now be further processed to generate a NC tape. There are two approaches:

7.3.7.1 SPECIAL POST-PORCESSOR FOR THE LUCAS MODEL MAKING MACHINE

A special post-processor to convert the machining data file into a NC tape into a suitable machine readable format for the Lucas Model Making Machine has been developed.

The command to use the special post-processor is as follows:

PROCESS input [MODEL]

where,

input: is the file containing the machining data.

[MODEL]: is the key word to activate the post-processor.

After the processing, a paper tape will be produced. The NC tape can then be used on the Lucas Model Making Machine to machine the part on soft foam material.

7.3.7.2 SIMPLE APT TRANSLATOR

For machining on harder material like aluminium, proper machine tool has to be used. Hence the development of this simple APT translator. The translator converts the machining data from PROFILE or MACHINE into a simple APT part program. Each machining sequence is converted into a separate APT part program. Information at the header record of the machining data i.e. cutter

diameter, spindle speed and initial setting poit are converted into corresponding APT statements as in the following:

LOADTL / 1

UNITS /

SPINDL /

FROM /

RAPID, GO TO /

Each machining data record is converted into a simple APT

GO TO / x, y z statements.

The end record generated a APT

FINI statement.

For a flat-end cutter, the x, y, z co-ordinates are the same as that machining data (assuming machining a horizontal planar surface.) For ball-end cutter, the x, y, z is offset by the radius r vertically (for 2 1/2 D parts) w.r.t. machining record in the machining data file.

This resultant APT part program can then be processed with a

suitable post-processor to produce a NC tape for the machine tool.

7.3.8 IMPLEMENTATION

The processors ROUGH, DISPLAY, MODIFY and PROCESS are developed as an enchancement to the software suite SURFSET designed for sculptured surface applications.

These processors are originally developed for sculptured surface processing using ball-end cutter only. They are modified to cater for machining data generated based on a geometric models.

The decision to make use of any available existing working software is to minimize lead time and avoiding 're-inventing' the wheel syndrome.

7.3.9 DEMONSTRATION OF RESULTS

To demonstrate the applicability of the above system, commands HOLE (for the point feature), PROFILE (for the profile feature) and MACHINE (for the face feature) have been used to machine successfully an engineering 2 1/2 D part model (i.e. an end bracket of an alternator) on the Lucas Model Making Machine (see Figures 7.3a - d).

The above example has demonstrated the validity of the concept of the automatic 2 1/2 D cutter path generator mentioned above.

If the cutting strategy (Si) of every machining features (Fmi) and the links (Li) between them can be defined and stored in the computer, it can be seen that the automatic machining model is possible:

$$AMM = \sum_{i} Li\ Si\ Fmi$$

7.3.10 EXTENSION OF SYSTEM TO 3D

The system has been successfully extended to cover inclined plane and horizontal cylinder machining using ball-end cutter.

Although the above approach works effectively for 2D, 2 1/2 D and limited 3D parts (with inclined plane and horizontal cylinders) applications, for fully 3D applications it is envisaged to have the following disadvantages:

1) As the output is a cutter location file, it is difficult to understand and debug. The NC programmer cannot check conveniently whether the program has performed correctly or not.

2) The part program thus generated tends to be much larger than that generated by manual APT programming.

3) To develope the system to a production grade system for tooling would require a substantial investment especially in machining technology and post-processing development.

4) Complexity of developing offset functions for specific shape or forms of tool.

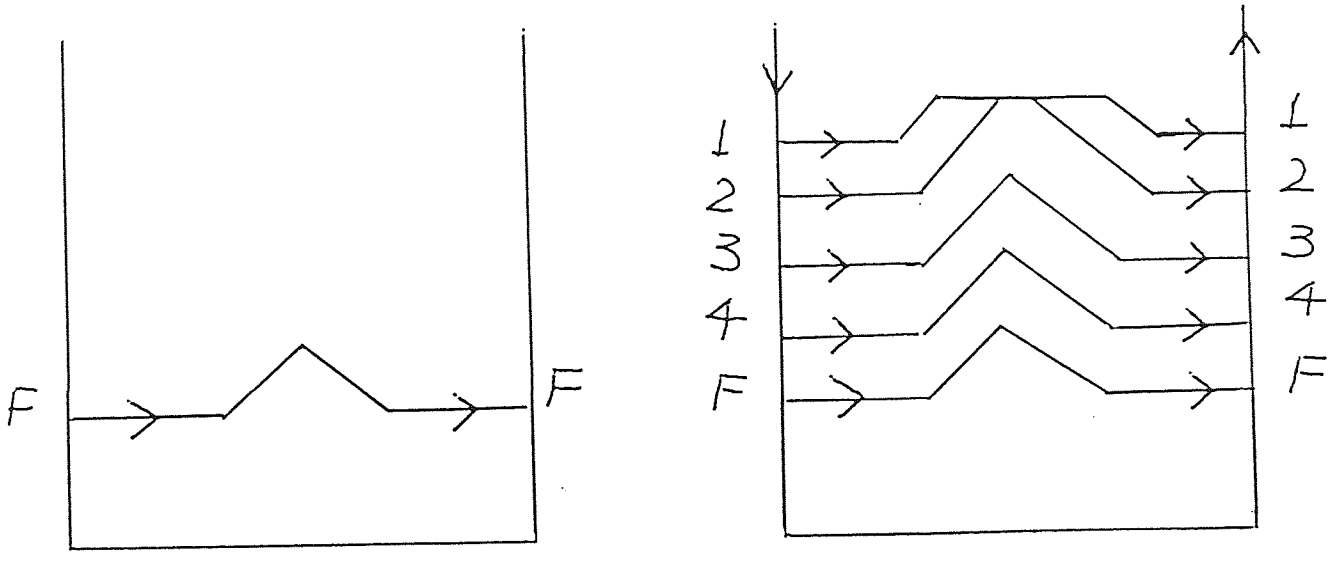
7.3.11 PARAMETER MODEL

The above problem can be tackled by using the concept of modifying interactively the geometric model face by face with suitable offset to generate a machining model consisting of parameterized faces (i.e. a parameter model). These faces are the envelope of the cutter of the centre, and which can be machined parametrically. Currently CAM-I's ANC Project is addressing this problem with a $4 million to produce a 2 1/2D bounded NC processors.

FIGURE 7.1 The cutter location co-ordinates

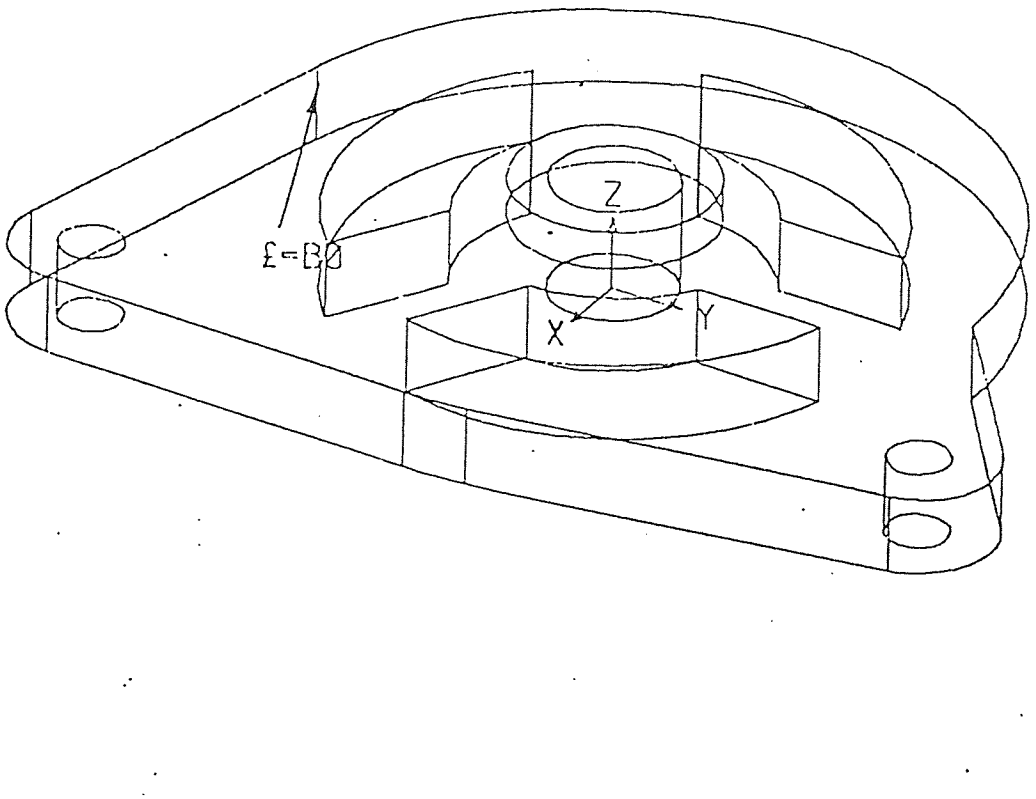
```
MACHINE  CM
FLATEND   0.400000E 01 0.000000E 00 0.000000E 00 0.400000E 01
P  0.2900000E 02  0.9100000E 02 -0.1000000E 02
L  0.2900000E 02 -0.4000000E 01 -0.1000000E 02
L  0.1024999E 02 -0.4000000E 01 -0.1000000E 02
L  0.1024999E 02  0.6099995E 02 -0.1000000E 02
L -0.1025000E 02  0.6099998E 02 -0.1000000E 02
L -0.1025000E 02 -0.4000000E 01 -0.1000000E 02
L -0.2900000E 02 -0.4000000E 01 -0.1000000E 02
L -0.2900000E 02  0.9099995E 02 -0.1000000E 02
L -0.2758068E 02  0.9996135E 02 -0.1000000E 02
L -0.2346157E 02  0.1080457E 03 -0.1000000E 02
L -0.1704586E 02  0.1144614E 03 -0.1000000E 02
L -0.8961578E 01  0.1185806E 03 -0.1000000E 02
L -0.7602492E-04  0.1200000E 03 -0.1000000E 02
L  0.8961432E 01  0.1185807E 03 -0.1000000E 02
L  0.1704573E 02  0.1144615E 03 -0.1000000E 02
L  0.2346147E 02  0.1080458E 03 -0.1000000E 02
L  0.2758064E 02  0.9996150E 02 -0.1000000E 02
L  0.2900000E 02  0.9100000E 02 -0.1000000E 02
L  0.2900000E 02  0.9100000E 02 -0.1000000E 02
E
```

FIGURE 7.2 Roughing passes



F : the finishing cut

FIGURE 7 3a· The alternator end-bracket



FIFURE 7.3b Cutter path (milling) generated for the alternator end-bracket in the x-y plane

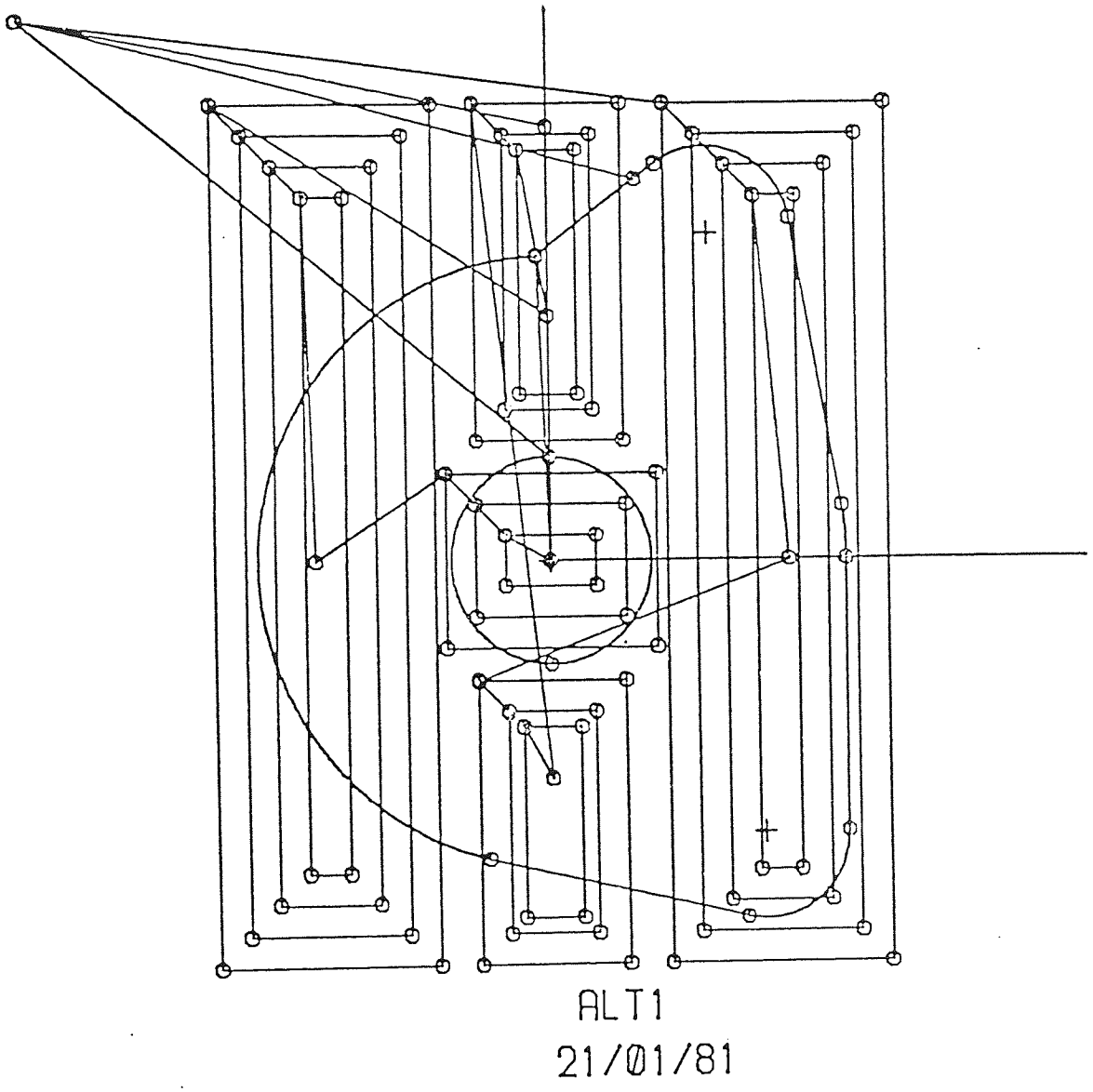


FIGURE 7.3c: Perspective view of the cutter path (milling) of the alternator end-bracket

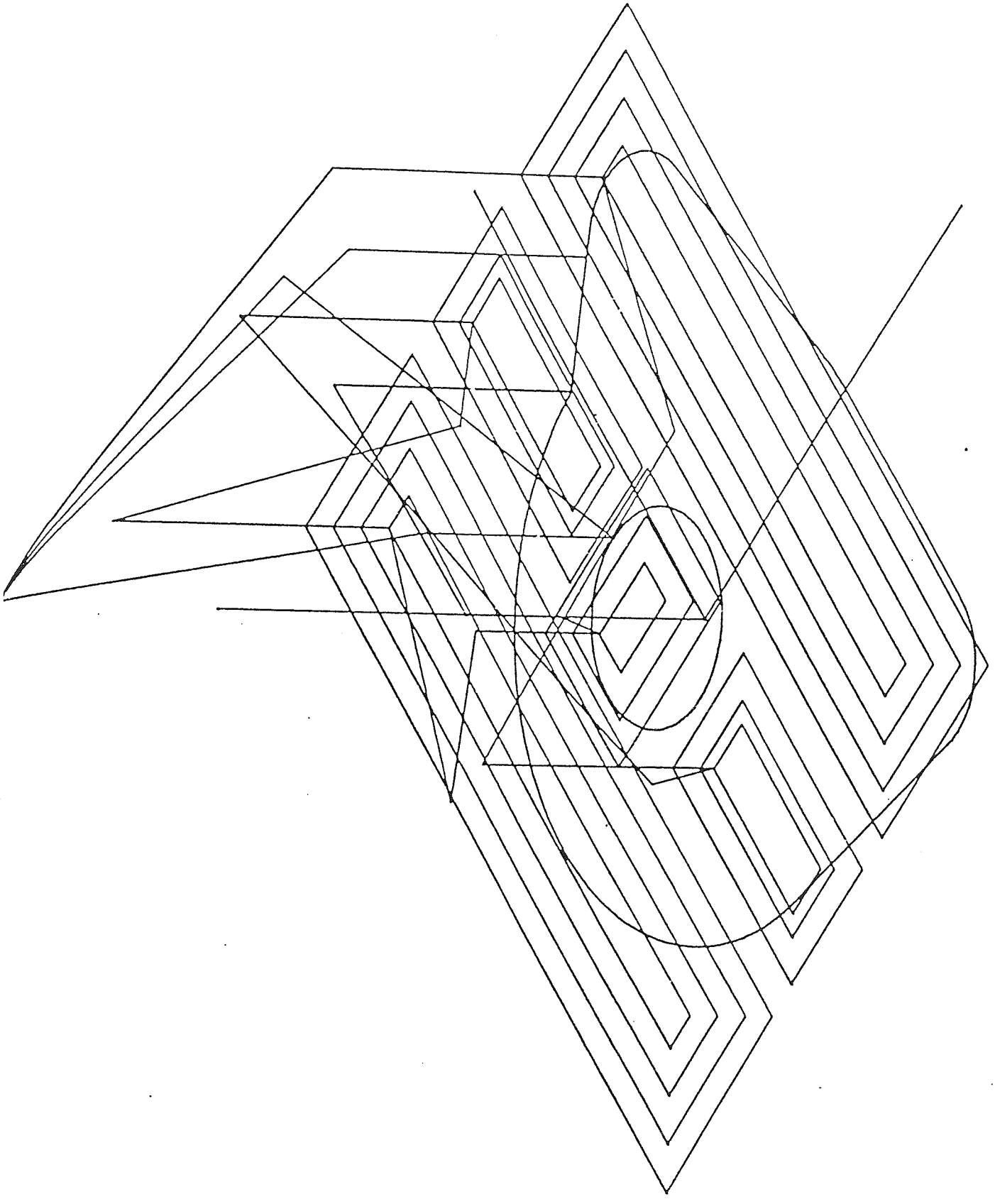
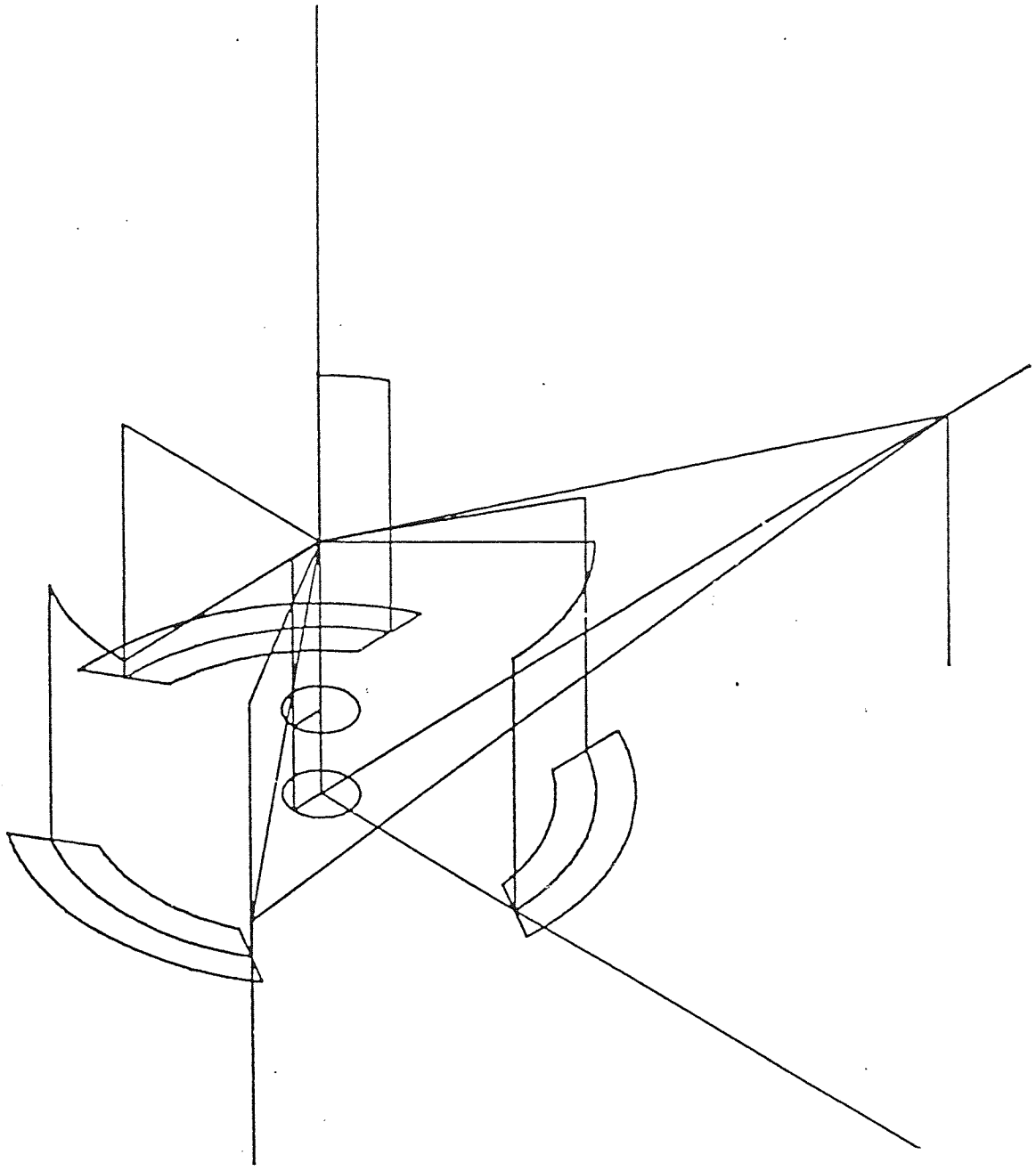


FIGURE 7.3d Perspective view of the cutter path (hole and pocketing) of the alternator end-bracket



8. ROMAPT (STAGE 2)

This is an extension of the ROMAPT (stage 1) concept. Instead of outputing the cutter locations, the corresponding APT motion statements (for cutter path generation) are output.

Together with the APT geometry generation capability and some auxiliary facilities, a complete APT part program can be generated using ROMAPT.

The following describes the development of the ROMAPT 2 system.

8.1 APT MOTION STATEMENTS

There are four types of APT motion statements.

1) Direct

GO TO / P1

This moves the cutter centre (i.e. tool tip) to point P1.

2) Incremental

GODLTA / -n

This moves the cutter in the Z-direction.

3) Start-up

GO / drive surface (DS), part surface (PS), check surface (CS)

The moves the cutter from an initial position to the location defined by the drive surface, part surface and check surface.

The following APT statements of the part prgram example (see Section 2.8.1):

FROM / PST

GO / TO, L4, TO, PS, TO, L1

moves the cutter from point PST to reach the drive surface, L4, part surface, PS and check surface L1.

In general, the start-up can be defined as:

GO / TO , DS , TO , PS , TO , CS

ON ON

PAST PAST

where ON, TO, PAST are final positional modifiers.

4) Contouring

TLON	,	GOFWD	/ DS	,	TO	, CS
TLRGT		GOBACK			ON	
TLLFT		GORGT			PAST	
		GOLFT			TANTO	

The contour statement defines the relative position of the tool (via positional modifiers· TLON, TLRGT, TLLFT) to the drive surface (DS) e.g. sitting on the DS (TLON), touching the DS on the right (TLRGT), or touching the DS on the left (TLLFT). The statement also defines the direction of the tool (via directional modifiers: GOFWD, GOBACK, GORGT, GOLFT) would move along the DS e.g. forward (GOFWD), backward (GOBACK), turn right (GORGT) or turn left (GOLFT). The tool is to move until it is touching (TO) or sitting on (ON), or passing (PAST) or tangential to (TANTO) the check surface CS. Again using the part program example in Section 2.8.1.

TLRGT, GORGT / L1, PAST, L2

This directs the cutter to turn right and touches on the right of the DS (L1) and moves the cutter along L1 until it passes CS (L2).

It is this relationship of the cutter to the DS, PS and CS that defines the cutter path.

To generate the APT motion statement automatically, the development of the following functions are therefore required.

8.2 FUNCTIONS FOR AUTOMATIC GENERATION OF APT MOTION STATEMENT

These functions are developed as an extension to those discussed in Section 6.7

8.2.1 SENSE FUNCTION Fs

The Sense Function, Fs, defines the sense (clockwise or anti-clockwise) of the cutter when machining round a feature (i.e. a loop).

Fs = 1 anti-clockwise

2 clockwise

8.2.2 INITIAL FUNCTION Fin

The position function defines the relative position (outside, on or inside) of the starting point of the cutter in relation to the feature (i.e. a loop).

Fin = OUT (outside)

ON (sitting on)

IN (inside)

8.2.3 GLOBAL RAISE FUNCTION Fg

The global raise function Fg is an arbitrary function to regard all faces of a feature (i.e. a loop) are raise faces without affecting the value of their raise function Fr.

Fg (Fl) = 0 no change

1 all faces are raise

where Fl: is the loop function.

8.2.4 START-UP FUNCTION Fst

In APT motion part programming, there is a need to define the start-up point for the cutter. The start-up function, Fst, is developed for this purpose. It is a function of Mc, Fs, Fin, Fg, Fc, Fi and Fr.

Fst (Mc, Fs, Fin, Fg, Fc, Fi, Fr) = 0 not machinable

> 1 machinable
(start-up identified)

where,

Mc: is the machining condition

Fs: is the sense function

Fin: is the global raise flag

Fc: is the connectivity function

Fi: is the intersection function

Fr: is the raise face function

The start-up function identifies the drive surface (DS), part surface (PS) and check surface (CS) in relation to the cutter and

defines the first cutter contact point on the PS.

8.2.4.1 RULES OF START-UP FUNCTION, Fst

The rules of the start-up function are developed for a loop as in the following (Figure 8.1).

Let,

PS: part surface

E1: 1st edge in the loop

F1: adjacent face of E1

Fr1: raise function of E1

En: n-th (i.e. the last) edge in the loop

Fn: adjacent face of En

Frn: raise function of En

The rules for defining DS and CS are such that the first edge (E1)

and last edge (En) will normally be the drive and check surface respectively. However, if their adjacent faces Fl and Fn are raise face, then Fl and Fn would replace El and En as drive and check surface respectively (see Table 8.1).

TABLE 8.1 RULES OF ASSIGNING DRIVE AND CHECK SURFACES

INPUT		OUTPUT	
Frl	Frn	DS	CS
0	0	El	En
1	0	Fl	En
0	1	El	Fn
1	1	Fl	Fn

The initial relation of the cutter to the loop is defined by the initial function Fin. Of the three positional modifiers : TO, ON, PAST for DS and CS, the PAST modifer is not necessary.

As the positional modifier for the PS is always TO, hence, there are in total only 4 combinations. They are assigned an arbitrary Start-up Function , Fst, value as follows:

Fst = 1 = GO/ ON, DS, TO, PS, ON, CS

2 = GO/ TO, DS, TO, PS, ON, CS

3 = GO/ ON, DS, TO, PS, TO, CS

4 = GO/ TO, DS, TO, PS, TO, CS

The rules for defining the start-up function Fst are as in the Table 8.2.

TABLE 8.2 RULUS FOR START-UP FUNCTION

INPUT			OUTPUT
Fg	Frl	Frn	Fst
1	-	-	4
0	0	0	1
0	1	0	2
0	0	1	3

0 1 1 4

8.2.4.2 ALGORITHMS OF START UP FUNCTION

The algorithm of start-up function is given in the following:

1) Identify the selected face (PS).

2) Identify the loop.

3) Identify the 1st edge E1 and the last edge En and their adjacent raise faces Fl, Fn respectively, if any.

4) Store E1 in DS and En in CS.

5) Replace DS by Fl if Fl is a raise face and CS by Fn if Fn is a raise face.

6) If global raise function Fg = 1, or Fl is a raise face, output ´GO / TO. DS ´, go to step (8).

7) Else output ´ GO / ON, DS ´.

8) Output ´ , TO, PS ´.

9) If Fg = 1 or Fn is a raise face, output ´ , TO, CS ´ and go to

step (11).

10) Else output ' , ON, CS '.

11) Stop.

8.2.5 POSITION FUNCTION Fp

In NC machining, there is a need to determine the relative position of the cutter w.r.t. the current drive surface. This is equivalent to finding out whether a point is inside or outside the material of the geometric model.

Fp (Fin, Fs, Fc, Fi, Fr, Cf) = 0 indeterminate
1 TLON (tool on)
2 TLLFT (tool left)
3 TLRGT (tool right)

where,

Fin: initial function

Fs: sense function

Fc: connectivity function

Fi: intersection function

Fr: raise function

Cf: cut function

The position function Fp defines the next correct orientation of the cutter in relation to the current drive surface, i.e. tool on the surface, tool on the left of the surface or tool on the right hand side of the surface.

8.2.5.1 RULES OF POSITION FUNCTION Fp

The rules of the position function Fp are developed as in the following (Figure 8.1).

Let,

PS: part surface

Fin: initial function (OUT/ ON/ IN)

Fs: sense function of the loop

The other variables (i.e. El, Fl, Frl, En, Fn, Frn) are the same as those defined in Section 8.2.4.1

There are 3 positional modifiers: TLON, TLLFT TLRGT. They are assigned to an arbitrary value of the position function Fp:

Fp = 1 = TLON

2 = TLLFT

3 = TLRGT

The rules for defining the position function Fp are given in Table 8.3.

TABLE 8.3 RULES OF POSITION FUNCTION

INPUT					OUTPUT	
Fin	Fs	Fg	Frl	Frn	Fm	MODIFIER
OUT	1	1	-	-	3	TLRGT
IN	1	1	-	-	2	TLLFT
OUT	2	1	-	-	2	TLLFT
IN	2	1	-	-	3	TLRGT

ON	-	0	0	0	1	TLON
OUT	1	0	0	0	1	TLON
OUT	1	0	1	0	3	TLRGT
OUT	1	0	0	1	1	TLON
OUT	1	0	1	1	3	TLRGT
IN	1	0	0	0	1	TLON
IN	1	0	1	0	2	TLLFT
IN	1	0	0	1	1	TLON
IN	1	0	1	1	2	TLLFT
OUT	2	0	0	0	1	TLON
OUT	2	0	1	0	2	TLLFT
OUT	2	0	0	1	1	TLON
OUT	2	0	1	1	2	TLLFT
IN	2	0	0	0	1	TLON

IN	2	0	1	0	3	TLRGT
IN	2	0	0	1	1	TLON
IN	2	0	1	1	3	TLRGT

8.2.5.2 ALGORITHM FOR POSITION FUNCTION

The algorithms for the position function Fp is given in the following:

1) Set Fp = 0 (i.e. not machinable or indeterminate).

2) Identify the selected face (PS).

3) If cut vector Cf not acceptable, stop.

4) Identify the loop.

5) Identify the sense function Fs, the initial function Fin, the global raise function Fg.

6) Identify the 1st edge El and last edge En and their adjacent face Fl and Fn and calculates their raise function Frl and Frn respectively.

7) Calculate the position function according to the Table 8.3.

8) Set the appropriate value to Fp.

9) Stop.

8.2.6 DIRECTION FUNCTION Fd

In NC machining, there is a need to determine which direction the cutter should go next w.r.t. the current tool position. This requirement leads to the development of the direction function. This function is based on the topology and geometric structure of the geometric model.

Fd (Fs, Fc, Fi, Fr, Cf) = 0 indeterminate
1 GOFWD (go forward)
2 GOLFT (go left)
3 GORGT (go right)

where,

Fs: is the sense function

Fc: connectivity function

Fi: intersection function

Fr: raise function

Cf: cut function

The direction function defines the next move for the cutter, i.e. forward left or right of the current direction.

It should be emphasized that this function should also be applicable to robotic movement or other manipulator that require a direction evaluator.

8.2.6.1 RULES FOR DIRECTION FUNCTION Fd

There are two methods to define the direction of the cutter i.e. the point and the line methods.

8.2.6.2 POINT METHOD

This method uses the point direction statement of APT to give a direction to the cutter via a point (P1). This point is usually provided by the next point in the loop. If next edge is an arc, then the mid-point of the edge is used (Figure 8.2).

INDIRP / P1

This indicates a direction for the cutter to move from the current cutter position (P0) along the vector, V.

V = P1 - P0

In this method, the direction function Fd for the cutter is always GOFWD i.e. along the direction given by V, i.e.

TLON , GOFWD / DS, ...

TLLFT

TLRGT

This method has two main advantages:

1) Simple calculation for direction function

The direction is always easily determined by the next point in the loop (or mid-point of the edge if circular).

2) Suitable for 3D application

As the direction is indicated by a vector V, this method is well suited for 3D machining applications.

However, this method would fail on some special cases when the radius of the cutter is greater than the length of the forward edge (see Figures 8.3a, 8.3b).

To remedy this situation, the line method is used.

8.2.6.2.1 ALGORITHM FOR DIRECTION FUNCTION (POINT METHOD)

1) Identify the part surface (PS).

2) Identify the loop.

3) Identify the current edge.

4) Identify the next point P1 in the loop.

5) Get the co-ordinate of the point.

6) Output ' INDIRP / P1 '.

7) Output ' GOFWD '.

8) Stop.

8.2.6.3 LINE METHOD

In this method, formal direction modifiers: GOFWD, GOLFT and GORGT are used to define the direction for the cutter i.e. go forward, go left, go right respectively.

If the current edge (E1) is tangent (i.e. TAN = 1) to the next edge (E2) (i.e. the Fc = 2 or 3, Fi = 2), the direction for the cutter is GOFWD (see Figure 8.4).

With reference to Figures 8.5a and 8.5b, let

Fs: sense of the loop

P1: 1st point in the loop

P2: 2nd point in the loop

P3: 3rd point in the loop

Two evaluator functions required by the Direction Function Fd are described in the following:

1) SENSE EVALUATOR Es

The sense evaluator determines the relative sense for the cutter

(when machining round the loop).

Es (P1, P2, P3) = 1 (-ve, anti-clockwise)

2 (+ve, clockwise)

where P1, P2, P3 are defined as above.

2) DIRECTION EVALUATOR Ed

The direction evaluator Fd determines the relative direction for the cutter:

Ed (Fs, Es) = 1 (GOLFT)

2 (GORGT)

where,

Fs: sense function

Es: sense evaluator

(In the computer, Ed is stored as variable IFRL2).

The direction of the cutter (i.e. Fd) can be determined by the

direction evaluator Ed via a simple vector cross-product multiplication.

Let,

V1 = P2 - P1

V2 = P3 - P2

The co-ordinate system is assumed to be right-handed (with normal pointing out of the paper).

For sense function Fs = 1 (anti-clockwise), the cross product V2 X V1 = -ve (i.e. normal is 'into' the paper) defines Es = 1 (i.e. anti-clockwise) and Ed = 1 (i.e. GOLFT).

Similarly, the cross-product V2 X V1 = +ve (i.e. normal is 'out' of the paper), defines Es = 2 (i.e. clockwise) and Ed = 2 (i.e. GORGT).

For Fs = 2 (clockwise), then the cross-product V2 X V1 = +ve defines Es = 2 (clockwise) and Ed = 1 (GORGT).

Similarly, the cross-product V2 X V1 = -ve defines Es = 1 (anti-clockwise) and Ed = 2 (GOLFT).

Hence, the rules for the direction function Fd is as in the

following:

TABLE 8.4 RULES FOR DIRECTION FUNCTION

INPUT			OUTPUT		
Fs	Es	TAN	Ed	Fd	MODIFER
1	1	0	1	2	GOLFT
1	2	0	2	3	GORGT
2	2	0	1	3	GORGT
2	1	0	2	2	GOLFT
-	-	1	-	1	GOFWD

8.2.6.4 ALGORITHM FOR DIRECTION FUNCTION (LINE METHOD)

The algorithm for the direction function Fd is given in the following:

1) Identify the part surface.

2) Identify the loop.

3) Identify the current edge E1 and its end points P1 and P2.

4) Identify the next edge E2 and its end points P2 and P3.

5) Calculate the connectivity function Fc and intersection function Fi of the two edges.

6) If the two edges are tangent (i.e. Fc = 2 or 3, Fi = 2), then output the modifier 'GOFWD'. stop.

7) Identify the sense function Fs.

8) Calculate the cross-product V2 X V1 and set the direction evaluator Ed flag accordingly.

9) Assign the value for the direction function Fd and output the modifier according to the rules of Table 8.4.

8.2.7 RELATION FUNCTION Fre

In NC machining, there is a need to determine the relative position of the cutter w.r.t. to the current check surface CS.

This function is related to the topology and geometric structure of the geometric model.

Fre (Fin, Fg, Fc, Fi, Fr, Cf) = 0 indeterminate

1 TANTO

2 ON

3 TO

4 PAST

where,

Fin: initial function

Fg: global raise function

Fc: connectivity function

Fi: intersection function

Fr: raise face function

Cf: cut function

This function defines the relationship between the cutter and the current check surface i.e. TANTO, ON, TO or PAST.

8.2.7.1 RULES OF RELATION FUNCTION Fre

The four final positional modifiers: TANTO, ON, TO and PAST are used to define the position of the cutter w.r.t. the current check surface (CS). These modifiers are assigned to an arbitrary value of the relation function Fre:

Fre = 1 = TANTO

2 = ON

3 = TO

4 = PAST

With reference to Figures 8.1a and 8.1b, let

E1: 1st edge in the loop.

F1: adjacent face of E1.

Fr1: raise face function of F1.

E2: 2nd edge in the loop.

F2: adjacent face of E2.

Fr2: raise face function of F2.

The rules defining the relation function Fre are given in the Table 8.5.

TABLE 8.5 RULES FOR RELATION FUNCTION

INPUT					OUTPUT		
Fin	Fg	Fr1	Fr2	TAN	Fre	MODIFIER	
OUT	1	-	-	1	1	TANTO	
IN	1	-	-	1	1	TANTO	
OUT	1	0	0	0	4	PAST	
OUT	1	1	0	0	4	PAST	*
OUT	1	0	1	0	3	TO	
OUT	1	1	1	0	4	PAST	*
IN	1	0	0	0	3	TO	*
IN	1	1	0	0	3	TO	
IN	1	0	1	0	3	TO	
IN	1	1	1	0	4	PAST	*

ON	0	0	0	0		2	ON
ON	0	0	0	1		1	TANTO
OUT	0	0	0	0		2	ON
OUT	0	1	0	0		2	ON
OUT	0	0	1	0		3	TO
OUT	0	1	1	0		3	TO	*
IN	0	0	0	0		2	ON
IN	0	1	0	0		2	ON
IN	0	0	1	0		3	TO
IN	0	1	1	0		3	TO	*

(* Denotes those rules that can be reversed, see Section 8.2.7.3.)

However, the above general rules would fail on some special cases when both Fr1 = 1 and Fr2 = 1 (see Figures 8.6a, 8.6b).

In Figure 8.6a, according to the rules in the Table 8.5, the positional modifier is TO i.e. E1, TO, E2 which is wrong. The modifier should be PAST (i.e. E1, PAST, E2 as in Figure 8.6b) to give the correct cutter path as required.

To overcome the above special cases, the Correlation Evaluator Ec and the Reverse Evaluator Er are used.

8.2.7.2 CORRELATION FUNCTION Ec

The correlation function Ec correlates the surface normal of the adjacent face of an edge and the sense of direction of the next edge in the loop.

Ec (E1, E2, P1, F1, N, V) = 0 (same sense)
1 (opposite sense)

where,

E1: 1st edge in the loop.

E2: 2nd edge in the loop.

P1: point between the above 2 edges. (P2 is the end point of E2.)

F1: adjacent face of edge E1.

N: surface normal of F1 (at P1).

V: sense of direction of E2 (along the direction of the loop).

The Ec defines whether the normal N of face F1 is of the same sense or opposite with the sense of the direction V of edge E2 (Figures 8.7a, 8.7b).

If E2 is a staright line, then V = P2 - P1. If E2 is circular, the V is the tangent of edge E2 at point P1.

The Ec can be calculated by a simple dot product. The dot product N.V = +ve defines Ec = 0 (i.e. same sense). Conversely, the dot product N.V = -ve defines Ec = 1 (i.e. opposite sense). The rules for the correlation function Ec is given in the Table 8.6.

TABLE 8.6 RULES FOR CORRELATION FUNCTION

INPUT	OUTPUT	
N.V	Ec	SENSE
+ve	0	same
-ve	1	opposite

8.2.7.3 REVERSE FUNCTION R

The reverse function R defines whether the rule of certain special case (as given in Table 8.5) should be reversed.

R (Fg. Ec) = 0 (no change)

1 (reverse)

where,

Fg: is the global raise function.

Ec is the correlation evaluator.

The relationship between Fg, Ec and R is given in Table 8.7.

TABLE 8.7 RULES FOR REVERSE FUNCTION

INPUT		OUTPUT
Fg	Ec	R
0	0	0
0	1	1

1	0	1
1	1	0

The effect of the reverse function R on the rules of Table 8.5 is given in Table 8.8.

TABLE 8.8 REVERSE RULES FOR RELATION FUNCTION

INPUT						OUTPUT	
Fin	Fg	Fr1	Fr2	TAN	R	Fre	MODIFIER
OUT	1	0	0	0	1	3	TO
IN	1	0	0	0	1	4	PAST
OUT	1	1	1	0	1	3	TO
IN	1	1	1	0	1	3	TO
OUT	0	1	1	0	1	4	PAST
IN	0	1	1	0	1	4	PAST

8.2.7.4 ALGORITHM FOR RELATION FUNCTION Fre

The algorithm for the relation function Fre is given in the following:

1) Identify the selected part surface (PS).

2) Identify the loop.

3) Identify the current edge E1 and its adjacent face F1. and its raise function Fr1.

4) Identify the next edge E2 and its adjacent face F2 and its raise function Fr2.

5) Identify the initial function Fin.

6) Identify the global raise function Fg.

7) Identify the sense function Fs.

8) Calculate the tangency TAN between E1 and E2.

9) Calculate the reverse function R.

10) Assign appropriate value to Fre according to Table 8.5 and Table 8.8.

11) Stop.

8.2.8 LOOP MODIFIER FUNCTION Fmo

The loop modifying function Fmo defines whether a loop is closed or open.

Fmo = CLOSE

OPEN

CLOSE indicates the whole loop will be machined.

OPEN indicates only part of the loop will be machined.

8.3 AUTOMATIC MACHINING FUNCTION FOR A LOOP Ml

The automatic machining function for a loop defines whether a profile feature can be machined automatically or not.

Ml (Fmo, Fin, Fg, Fl, Fst, Fp, Fd, Fre) = 0 (not machinable)

1 (machinable)

where,

Fmo: loop modifying function.

Fin: initial function.

Fg: global raise function.

Fl: loop function.

Fst: start-up function.

Fp: position function.

Fd: direction function.

Fre: relation function.

A loop can be machined automatically when its Fmo, Fin, Fg, Fl, Fst, Fp. Fd and Fd can be defined properly.

As many of the above functions, particularly, Fp, Fd and Fre need to access the data in the loop from the geometric model frequently, this would require scanning the geometric model many times to access the data. It is computationally more efficient to have a special data structure to store the data of the loop concerned. The above functions can then access the data much more rapidly via the special data structure. This data structure is described in the following.

8.4 LOOP DATA MODEL

The loop model data structure is used to represent a profile feature. The data is stored in an array LOOP and its items are described in the following.

(1) sense of loop (i.e. Fs).

(2) global raise flag (i.e. Fg).

(3) no. of elements n in the loop.

(4) minimum distance.

(5) minimum radius.

(6) flag OUT/ ON/ IN (Fin).

(7) flag CLOSE/ OPEN.

(8) flag MANUAL/ AUTO.

(9) pointer to tool.

(10+1) 1st point.

(10+2) 1st edge.

(10+3) raise flag (Fr).

(10+4) raise face.

(10+5) connectivity flag (between the 1st and the nth edges).

(10+6) intersection flag (between the 1st and the nth edges).

(10+7) double flag.

(10+8) omit flag.

(10+9) concave flag.

(10+10) reverse flag (R).

(10n+1) nth point.

(10n+2) nth edge.

(10n+3) raise flag (Fr).

(10n+4) raise face.

(10n+5) connectivity flag (between the nth and the n-1th edges).

(10n+6) intersection flag (between the nth and the n-1th edges).

(10n+7) double flag.

(10n+8) omit flag.

(10n+9) concave flag.

(10n+10n) reverse flag (R).

The loop model, LM, can be expressed in the following form:

LM = H + nE

where,

H: is the header description of the loop.

n: no. of elements in the loop.

E: elment of the loop.

The header description of the loop model is designed to hold the following data (currently 10 records):

(1) Sense of loop (Fs)

This defines the loop is to be machined in the same sense (i.e. anti-clockwise for an outer loop) or opposite (i.e. clockwise) of the loop.

(2) Global raise flag (Fr)

This flag allows the user to machine the loop as if all adjacent faces are raise faces.

(3) No. of elements n

This defines how many (n) elements of the loop.

(4) Minimum distance

This defines the critical minimum distance between 2 edges of the loop (or of different loops of the same face), which determines the maximum size of the cutter allowed to contour through these 2 edges.

(5) Minimum radius

This defines the smallest radius of the circular edges, if any, in

the loop. This also determines the maximum size of the cutter to contour through those edges.

(6) CLOSE/ ON/ IN flag (Fin)

This defines the relative start position of the tool in relation to the loop.

(7) CLOSE/ OPEN flag

This flag allows the user to manipulate the loop as either close or open (to suit machining requirement).

A close loop implies the cutter will start machining at the starting vertex of a specified edge, through the whole loop and back to the start vertex.

An open loop implies the cutter will start machining on an end vertex of a specified edge, through the number of edges specified and end on the start vertex of the last edge.

This allows the user to have some flexibilities to change the contour of the profile to suit practical machining strategy. For example, an open loop can be used to machine only one edge , or more but less than n, even there are n edges in the loop.

(8) AUTOMATIC/ MANUAL flag

This allows the user to generate the data for the elements of the loop based on the geometry and topology (i.e. Fc, Fi, Fr etc.) of the loop automatically or manually by the user.

The manual mode allows the user to overwrite (or modify) the geometry and topology of the loop to suit practical machining strategy (e.g. one or more edges can be omitted).

If required, the user can use this facility to construct a new loop model to suit machining requirement. Hence, a maximum flexibility is allowed.

(9) Tool

This stores the radius of the cutter.

(10) Unused.

This header description is designed to be extensible to cope with future requirement. The details of an element of a loop is described in the following.

8.4.1 ELEMETS OF LOOP MODEL (INTELLIGENT DATA)

Each element of the loop model, LM, consists of m (currently 10)

records as in the following (Figure 8.8).

RECORD(1): Point

This is pointer to the name of the point (i.e. Pn).

RECORD(2): Edge

This is pointer to the name of the edge (i.e. En, which is the backward edge of Pn).

RECORD(3): Raise function, Fr

This defines whether the adjacent face (i.e. Fn of En) is a raise face (=1) or not (=0) (Section 6.7.5).

RECORD(4): Raise face

This is a pointer to the name of the adjacent face (i.e. Fn).

RECORD(5) Connectivity function Fc

This defines the connectivity function, Fc, between En and En-1 (Section 6.7.3).

RECORD(6): Intersection function Fi

This defines the intersection function Fi between En and En-1

(Section 6.7.4).

Record 1 to 6 provides all information required by the loop function F1. As record 7 to 10 will provide some form of intelligence to the loop model, they are discussed in some details as follows.

8.4.2 INTELLIGENT DATA OF LOOP MODEL

The following descibes the intelligent data record 7 to 10 of the loop model.

8.4.2.1 REVERSE FUNCTION R flag (RECORD (10))

The reverse function R has been described in detail in Section 8.2.7.3.

The reverse function R is used to reverse the rules (Table 8.8) (for some special cases) for controlling the relative position of the cutter in relation to the current drive surface (DS) and check surface (CS) (i.e. En and En+1 respectively, see Section 8.2.7.3):

R = 0 (no change of rules)

1 (reverse rules)

8.4.2.2 CONCAVE FUNCTION Fcon flag (RECORD(9))

The concave flag defines whether the current edge (En) is concave or convex (assuming the edge is circular).

Fcon (En) = 0 (concave)

1 (convex)

The concave flag may affect the direction of the NC contouring of the loop. If the edge is convex, the sense is usually forward (i.e. GOFWD). Conversely, if the edge is concave, the sense is backward (i.e GOBACK).

In a special case where a straight line (E1) is tangent to a concave circular edge (E2), an extra line (L1) and logic has to be constructed to contour properly (see Figure 8.9).

8.4.2.2.1 ALGORITHM FOR CONCAVE FLAG Fcon

The algorithm for the concave function flag Fcon is given in the following:

1) Identify the circular edge E1.

2) Identify the adjacent face F defining the cylinder which contains E1.

3) If the inside of F is solid, set Fcon = 0.

4) If the outside of F is solid, set Fcon = 1.

5) Stop.

8.4.2.3 OMIT FUNCTION FLAG Fo (RECORD (8))

In NC machining, there are certain cases where the edge (E1) concerned is not suitable or convenient to machine (say for a certain cutter size). The edge is then temporarily ignored or deferred later (see Figures 8.10a, 8.10b).

The omit function Fo allows a user to omit one or more edges in the loop for the current contouring.

Fo (En) = 0 (retain)
1 (omit)

The contouring would proceed as if the edge (or edges) are not there. Therefore, care must be taken to ensure that omitting an edge (or edges) will not generate an impossible case (i.e. edges

do not intersect etc.).

8.4.2.3 ALGORITHM FOR OMIT FUNCTION FLAG Fo

The algorithm for the omit function flag Fo is given in the following:

1) Set Fo = 0 (i.e. retain).

2) Identify the edge En.

3) Get the cutter radius r.

4) Get the track equation of the edge En.

5) If the edge is circular, go to step (7).

6) If the length of the line is greater than the diameter (2r) of the cutter, set Fo = 1, stop.

7) If the radius rl of the edge is less than r, set Fo = 1.

8) Stop.

8.4.2.4 DOUBLE FUNCTION FLAG Df (RECORD(7))

In APT contouring, the position of the cutter as defined by the drive surface (DS) and check surface (CS) can be ambiguous as shown in Figures 8.11a - d.

The cutter position can be determined by calculating the number of intersections (INT) between the edge En and the track geometry of the consecutive edge En+1. If there is one intersection, the cutter is located at the 1st position. If there is two intersections, the cutter is located at the 2nd position. (Hence, the corresponding APT statement has to be repeated.)

The double function Df is developed to resolve the above APT ambiguity. The following Segment Function is required by Df.

8.4.2.4.1 SEGMENT FUNCTION Fseg

As can be seen in Figure 8.11d, the final cutter position sometimes depends on another parameter: the height of the segment h (Figure 8.12a).

Let,

r: is the cutter raidius.

r1: is the radius the circular edge C.

P1, P2: are the two intersection points between staight line L and circle C.

d: is half of the cord length between P1 and P2.

e: is the perpendicular distance from the centre P0 of C to d.

h: is the segment height.

The h is calculated as follows:

$d = (P1 - P2) / 2$

$e = \sqrt{(r1*r1 - d*d)} = \sqrt{((r1-d)(r1+d))}$

$h = r1 - e$

If $h > 2r$, the 1st PAST postion of the cutter is inside the circle (Figure 8.12b).

If $h < 2r$, the 1st PAST position of the cutter is outside the circle (i.e. the cutter is forced to the 2nd position) (Figure 8.12c).

The segment function Fseg defines whether the segment height h is

larger than the cutter diameter 2r.

Fseg (En, En-1, h, 2r) = 0 (i.e. h < 2r)

1 (i.e. h > 2r)

where,

En: the current edge in the loop.

En-1: the preceeding edge in the loop.

h: is the height of the segement.

r: is the cutter radius.

8.4.2.4.2 SEGEMENT FLAG SEG

The segment flag , SEG, defines the conditions whether the segment function should be invoked.

SEG (En, En-1, INT, Fin, Fc, Fi, Fcon) = 0 (not invoke)

1 (invoke Fseg)

where,

En: the current edge in the loop.

En-1: the preceeding edge in the loop.

INT: no. of intersections between edge En-1 and track geometry of En.

Fin: initial function.

Fc: connectivity function.

Fi: intersection function.

Fcon: concave function.

The rules for invoking the Fseg is given in the Table 8.9.

TABLE 8.9 RULES FOR INVOKING SEGMENT FUNCTION

INPUT					OUTPUT
Fin	Fc	Fi	INT	Fcon	SEG
-	1	-	-	-	0
-	4	-	-	-	0

OUT	3	3	2	0	0
OUT	3	3	2	1	1
OUT	2	3	2	0	1
OUT	2	3	2	1	0
IN	3	3	2	0	1
IN	3	3	2	1	0
IN	2	3	2	0	1
IN	2	3	2	1	0
-	-	-	1	-	0

8.4.2.4.3 DEFINITION OF DOUBLE FUNCTION Df

The double function, Df, defines the exact position of the cutter as determined by the corresponding drive surface (DS) and check surface (CS).

Df (En, En-1, Fc, Fi, INT, SEG, Fseg) = 1 (1st position)

2 (2nd position)

where,

En: is the current edge.

En-1: is the preceeding edge.

Fc: connectivity function.

Fi: intersection function.

INT: no. of intersection between edge En-1 and the track geometry of En.

SEG: segment flag.

Fseg: segment function.

8.4.2.4.4 RULES OF DOUBLE FUNCTION Df

The rules for the double function Df is given in Table 8.10.

TABLE 8.10 RULES FOR DOUBLE FUNCTION

INPUT				OUTPUT
Fc	INT	SEG	Fseg	Df
-	1	0	-	1
-	2	0	-	2
-	1	1	1	1
-	2	1	1	2
-	1	1	0	1
-	2	1	0	1
1	-	-	-	1
4	-	-	-	1

8.4.2.4.5 ALGORITHM FOR DOUBLE FUNCTION Df

1) Set double function flag M = 1 (i.e. 1st position).

2) Identify the edge En.

3) Identify the preceeding edge En-1.

4) Identify the connectivity function Fc between En and En-1.

5) If Fc = 1 or 4, stop.

6) Calculate the intersection (INT) between the edge En-1 and the track geometry of the edge En.

7) If INT = 1, stop.

8) Test the segment flag SEG, if SEG = 0, stop.

9) If SEG = 1, calculate the segment function Fseg.

10) If Fseg = 0, stop.

11) If Fseg = 1, set M = 2 (i.e. 2nd position)

12) Stop

8.5 ALGORITHM FOR AUTOMATIC MACHINING FUNCTION FOR A LOOP M1

The algorithm for automatic machining function for a loop is given in the following:

1) Identify the selected part surface PS and the loop concerned.

2) Read the loop modifying function Fmo for CLOSE/ OPEN option.

3) Read the global raise face function Fg.

4) Evaluate the loop function Fl and generate (or read) the intelligent data to set up the loop model.

5) Identify special cases, if any (i.e. single cylinder etc). Move the cutter to the starting point PST.

6) Evaluate the start-up function Fst for the first edge in the loop and output the APT start-up statement according to the rules in Table 8.2.

7) Evaluate the position function Fp of the current edge and output the APT positional modifier statement according to the rules in Table 8.3.

8) Evaluate the direction function Fd of the current edge and output the APT statement according to the rules in Table 8.4.

9) Evaluate the relation function Fre of the current edge and output the APT final position modifier statement according to the rules in Tables 8.5 and 8.8.

10) If necessary, evaluate the double function Df of the current edge. If Df = 2, output the current complete APT statement again (to place the cutter in the 2nd position).

11) Repeat steps (6) to (8) for all other edges specified in the loop.

12) Return cutter to the starting point.

13) Stop.

8.6 IMPLEMENTATION OF AUTOMATIC PROFILING

The following describes a practical computer implementation of the above algorithm. The operation of the system is illustrated in the following (for details see Chapter 9).

The data structure required for the generation of APT motion statements can be set up either manually or automatically:

CNC [LOOP] [option 1] [option 2] [option 3]

where,

[LOOP]: is the key word for the generation of the data structure for the loop.

[option 1]: indicates whether the loop is OPEN or CLOSE.

[option 2]: indicates whether the cutter is situated IN, ON, OUT of the loop.

[option 3]: indicates whether the MANUAL or AUTOMATIC mode is required.

The manual method allows more flexibility (to suit the technological requirement of the machining).

The automatic generation of the required APT motion statement is effected by the command:

CNC [APTLOOP] [options]

where,

[APTLOOP]: is a keyword for the generation of the APT motion statement for the loop.

[option]: define either the currently selected face or user-defined Z-planes.

The corresponding APT motion statements is then generated automatically. (This assumes the required geometry statements have been generated previously by using the APTENQUIRE command.)

8.6.1 ROUGHING FOR LOOP

If roughing is needed, then before the above APT contouring instructions are output, it is preceeded by using the APT thickness statement (29):

THICK / THPS, THDS, THCS1, THCS2, THCS3

where,

THPS: thickness for part surface.

THDS: thickness for drive surface.

THCS1, THCS2, THCS3: thickness for first, second and third check surfaces. The APT processor assumes that the last thickness specified is to apply for the remaining surfaces.

The thickness specified above would then be left for the finishing cut (see Figure 8.13a). The thickness statement can also be used repeatedly to generate a pocketing effect, if required (see Figure 8.13b).

As can be seen above, a machinable profile feature can be machined automatically (via an APT part program).

8.7 FACE MODEL

The concept of the loop model can be easily extended to the face model (FM) for a face feature.

The face data model is used to represent a face feature. The data is stored in an array (MACH) and its items are described in the following:

(1): pointer to the face.

(2): no. of loops n in the face.

(3): minimum distance between loops.

(4): minimum radius in the face.

(5): pointer to the tool.

(6): pointer to feed rate.

(7): pointer to speed.

(m) : the last pointer for the header of face model.

(m+1): pointer to links (machining strategy): to the 1st loop.

(m+2): pointer to the 1st loop.

(m+3): pointer to links to the 2nd loop.

(m+4): pointer to the 2nd loop.

(m+2n-1): pointer to links to the nth loop

(m+2n): pointer to the nth loop

(m+n+1): pointer to links to return.

Using the APTLOOP facility, all the loops (if machinable) can be machined automatically as above. The area left between loops can be machined by the area clearance facility described in the following.

8.7.1 AREA CLEARANCE

Area clearance facility is provided as an extension to the construction facility of GRID (see Section 4.3.10.1) for 2D or 3D part surface. A horizontal grid of straight lines (each separated by the amount specified by the the user indirectly) bounding that area are generated. The area is defined by the two cursor positions. ROMAPT generates a series of APT motion statement corresponding to a zigzag path for the cutter to traverse the area using those lines generated as drive surfaces and the boundray of the grid as check surface. The user has the option to machine on or within the grid. The command for this facility is as follows:

CNC CLEAR [CURSOR/ vector 1, vector 2]

Function: to clear off an area defined by two cursor positions or by two vectors.

8.7.2 ALGORITHM FOR AREA CLEARANCE

The algorithm for the area clearance operation is given in the following:

1) Identify the selected part surfac (PS).

2) Read the two cursor positions, or the two vetors, (P1 and P2)

defining the area for clearance.

3) Generate a grid of straight lines (as drive surfaces) each separated by the cutter diameter (or by the scallop distance if the part surface is inclining or non-planar) to cover that area.

4) Move the cutter from the starting point PST to point P1.

5) Move the cutter to traverse the area in a zigzag path using the lines within the grid as drive surfaces and the grid as check surface.

6) Return the cutter to the starting point.

7) Stop.

8.7.3 MACHINING OF HOLE

The machining of hole facility in APT is an extension of the HOLE command (see Section 7.3.1).

To provide further flexiblity in machining holes, two offset quantities are allowed (OFF1 and OFF2). The machining starts from the starting point PST to the centre P1 of the top circular edge plus the first offset (P1 + OFF1) to the centre P2 of the bottom circular edge minus the second offset (P2 - OFF2). The cutter is

then retracted to P1 + OFF1 before return to the starting point PST.

8.7.4 ALGORITHM OF MACHINING HOLE

The algorithm for machining holes is given in the following:

1) Identify the selected part surface.

2) Identify the top and bottom circluar edges C1 and C2 and their centres P1 and P2 (of the hole specified).

3) Get the cutter radius r.

4) Read the two cutter offsets OFF1 and OFF2.

5) Offset P1 and P2 by OFF1 and OFF2 respectively.

6) Start machining from the start point PST to P1 + OFF1.

7) Move the cutter from P1 + OFF1 to P2 - OFF2.

8) Retract the cutter from P2 - OFF2 to P1 + OFF1.

9) Return the cutter to PST.

10) Repeat steps (2) to (9) for other holes specified.

11) Stop.

8.8 FACE MILLING

As an alternative, a face can be machined using the method developed in MACHINE.

The face milling facility is an extension of the MACHINE facility (see Section 7.3.3) to generate APT motion statements to machine the selected 2D or 3D faces.

The cut vector is translated in straight line tracks (as check surfaces) for 2D part surface or into cut planes (as check surfaces) for 3D part surface.

All the intersection points between the cut vectors and the loops of the face are stored in the order they occur (in array POI) together with their edges (where intersection occur) or adjacent faces if raised (in array FINT).

The intersection points together with their associated edges or faces are re-ordered in ascending order according to the vector distance of the intersection points and the initial cutter position. The APT motion statements are output using the

intersection point, Pint, as direction, i.e.

INDIRP / Pint

The cut vector is used as drive surface. The selected face is the part surface. The associated edge or face of the intersection point is used as the check surface. The machining starts with the 1st intersection point to the 2nd. The cutter then retracts to a specified height, moves over to the top of the subsequent intersection (3rd) point. Machining resumes on this point to the next (4th) intersection point. This process continues until all intersection points have been processed. The whole procedure is then repeated for the next cut vector until the whole face is machined.

8.8.1 ALGORITHM FOR FACE MILLING

The algorithm of MACHINE is given in the following:

1) Identify the selected part surface (PS).

2) Identify the cutter definition and its radius r.

3) Identify all loops in the face.

4) Identify the edge E for generating cut vectors v.

5) Calculate the region offset function Or of the PS.

6) If Or is not machinable, stop.

7) All intersections between the cut vectors and the edges in the loops of the face are calculated. (Hole loops are ignored. Cut vectors for parallel edges are appropriately adjusted.)

8) Sort the intersection points in an ascending order according to the vector distance of the intersection point and the initial cutter position.

9) Move the cutter to the 1st intersection point to the 2nd point, using the cut vector as the drive surface.

10) Retract cutter and move it over to the top of the next intersection point.

11) Resume machining on this point to the subsequent point.

12) Repeat steps (10) to (11) for other intersection points of the current cut vector.

13) Repeat steps (7) to (12) for other cut vector until the whole face is machined.

14) Stop.

8.9 OBJECT MODEL (FINISHING)

The face model can be further extended to represent a complete object.

The object model can be used to represent a finished (i.e. machined) object. The data is stored in an array (OBJECT) and its items are in the following:

(1): pointer to the object.

(2): no. of faces n.

(3): pointer to tool.

(4): pointer to feed rate.

(5): pointer to speed.

(m): the last record of the header.

(m+1): pointer to links (machining strategy) to the 1st face (from

the starting point).

(m+2): pointer to the 1st face.

(m+3): pointer to links to the 2nd face.

(m+4): pointer to the 2nd face.

(m+2n-1): pointer to links to the nth face.

(m+2n): pointer to the nth face.

(m+2n+1): pointer to links to return (to the starting point).

8.9.1 BLOCK MODEL (ROUGHING)

The block model can be used to represent the blank block of the original stock for roughing machining. The data is stored in an array (BLOCK) and its elements are described in the following:

(1): pointer to block.

(2): pointer to object.

(3) no. of volume features n (see Section 6.2)

(4) pointer to tool.

(5) pointer to feed rate.

(6) pointer to speed.

(m): the last record of the header.

(m+1): pointer to links (machining strategy) to the 1st volume feature (from the starting point).

(m+2): pointer to the 1st volume feature.

(m+3): pointer to links to the 2nd volume feature.

(m+4): pointer to the 2nd volume feature.

(m+2n-1): pointer to links to the nth volume feature.

(m+2n): pointer to the nth volume feature.

(m+2n+1): pointer to links to return (to the starting point).

The above various models i.e. loop, face, object and block are data models of the machining model MM. (see Section 6.3).

8.10 DEVELOPMENT AND RESULTS

The main effort of this research lies on the development of the above mentioned theory and the implementation of the software (ROMAPT) based on the theory.

To demonstrate the applicability of the above theory, an engineering part has been created by using ROMULUS and then the geometric model is machined by using ROMAPT to generate the cutter path automatically.

The details of the design and manufacturing of the part is given in Chapter 9.

FIGURE 8 1: Contouring example

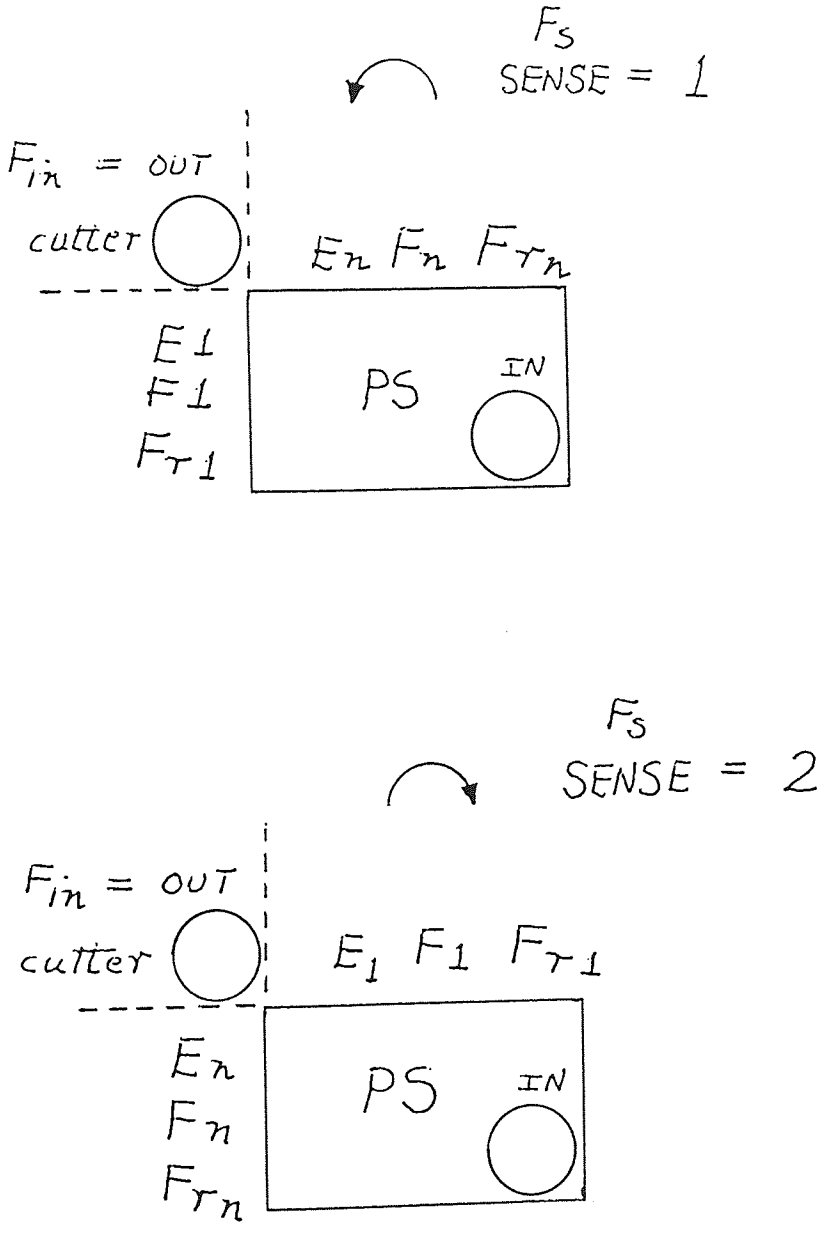


FIGURE 8 2· Point method

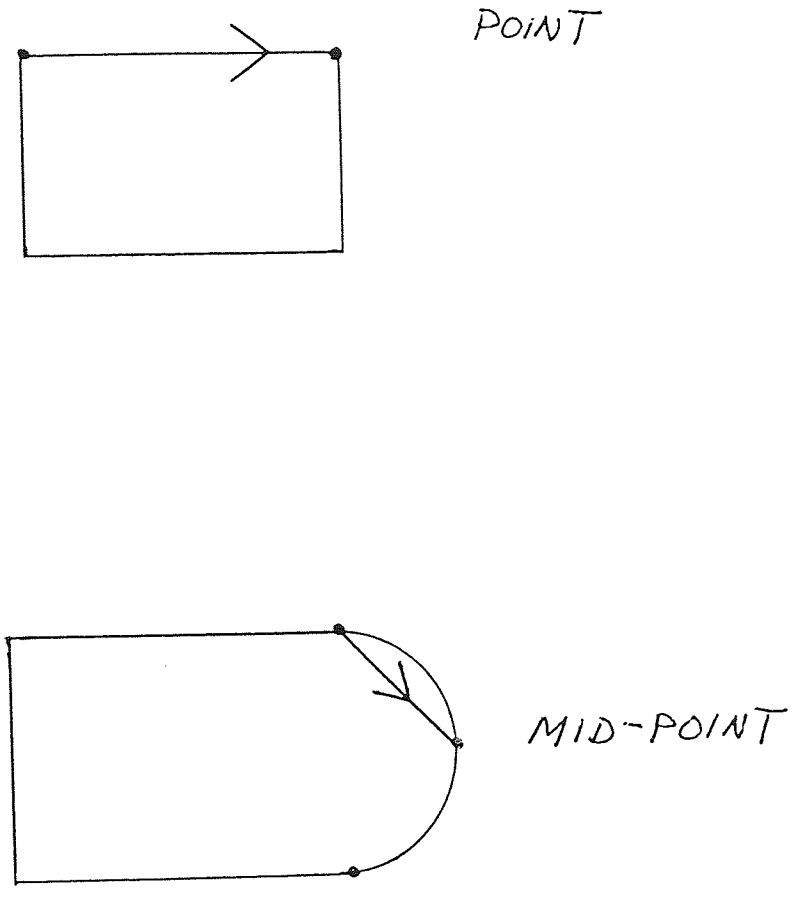


FIGURE 8 3a· Normal case of point method

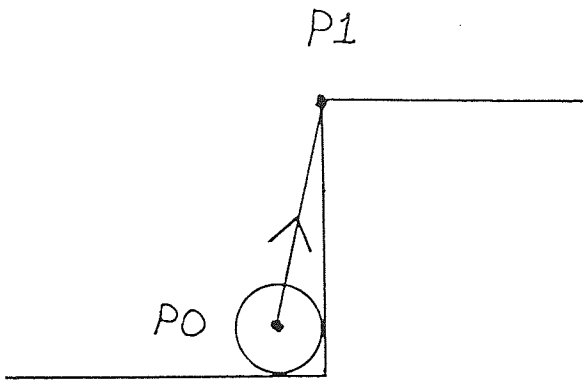


FIGURE 8.3b· Problem case of point method

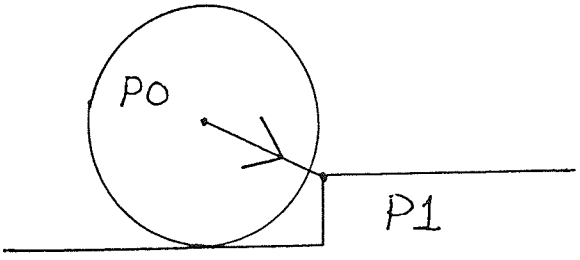
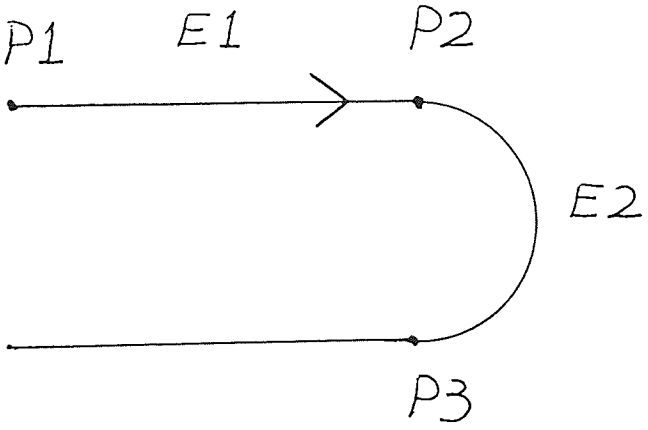


FIGURE 8 4: Tangential case



TAN = 1 (TANGENT)

$F_c = 2$ or 3

$F_i = 2$

FIGURE 8 5a Direction function (same sense)

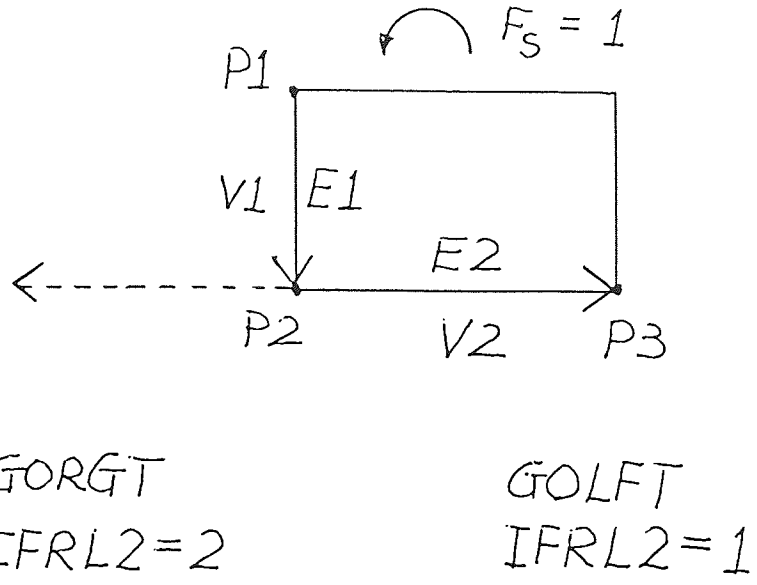


FIGURE 8.5b Direction function (opposite sense)

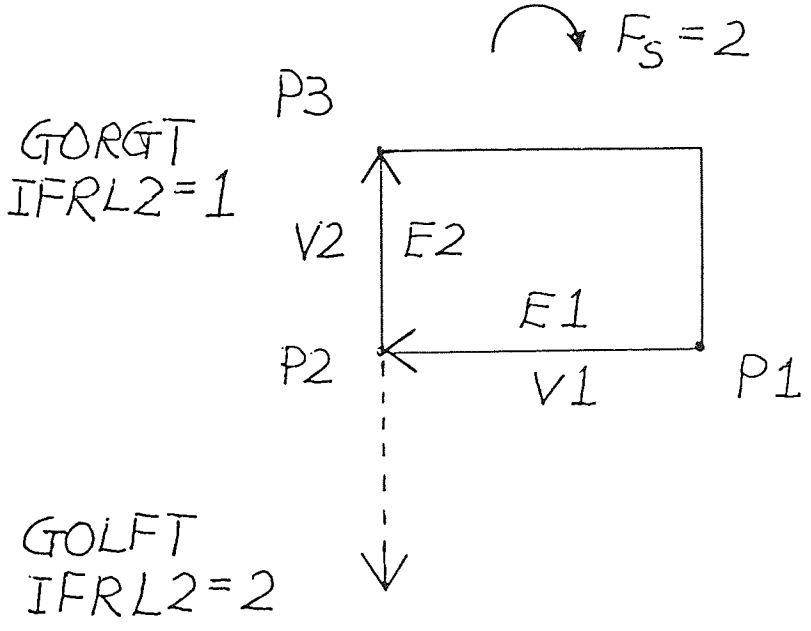


FIGURE 8.6a· Problem case of normal rules

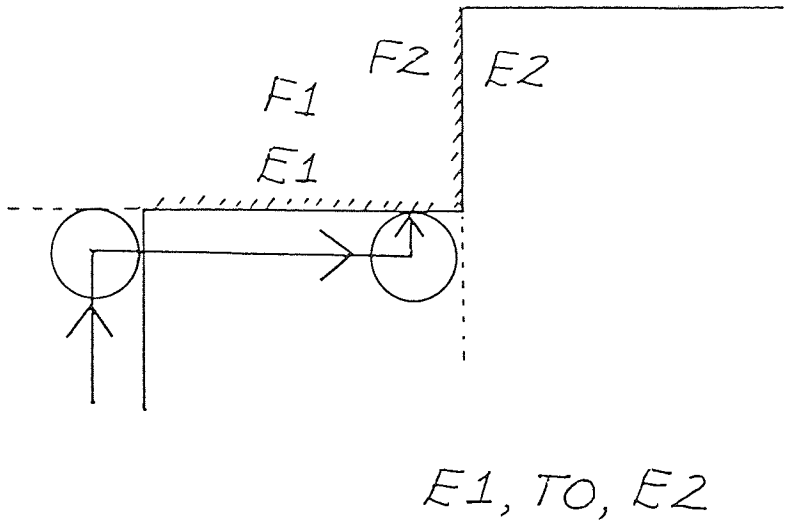


FIGURE 8.6b· Reverse of rules

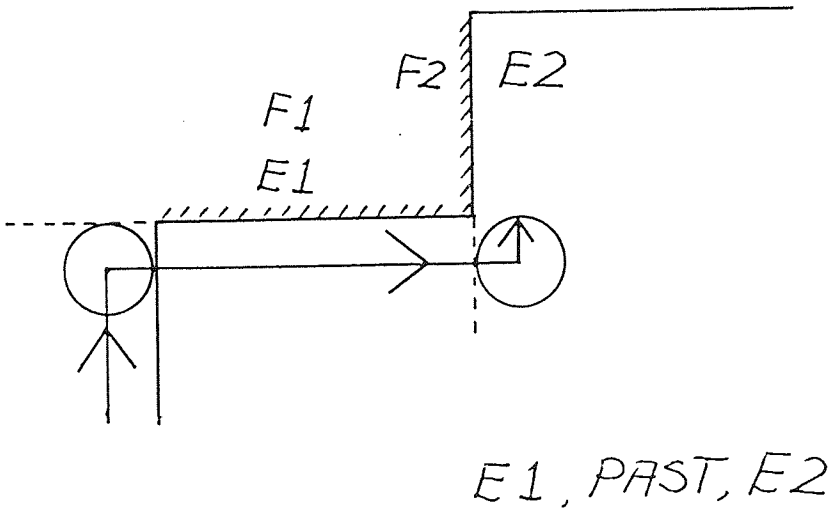


FIGURE 8.7a: Correlation function (same sense)

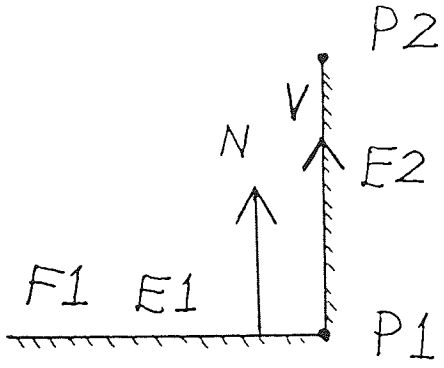


FIGURE 8.7b: Correlation function (opposite sense)

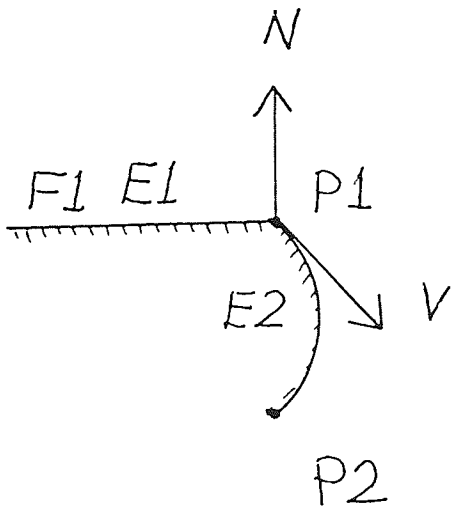


FIGURE 8.8 An element of the loop model

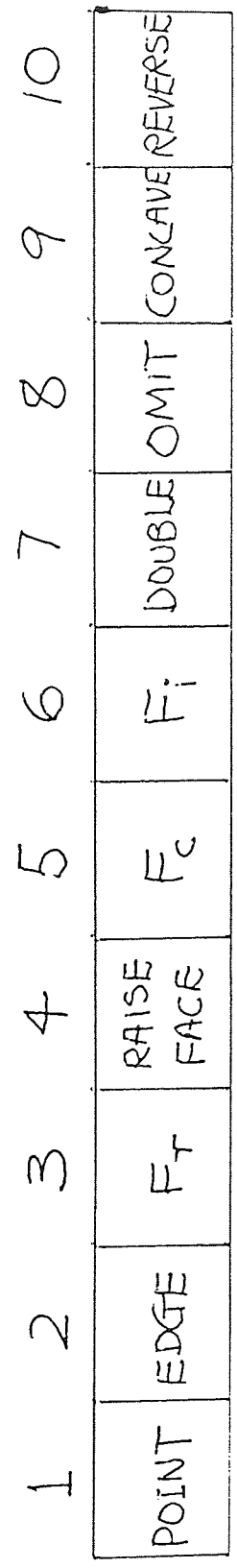


FIGURE 8.9 Special concave case

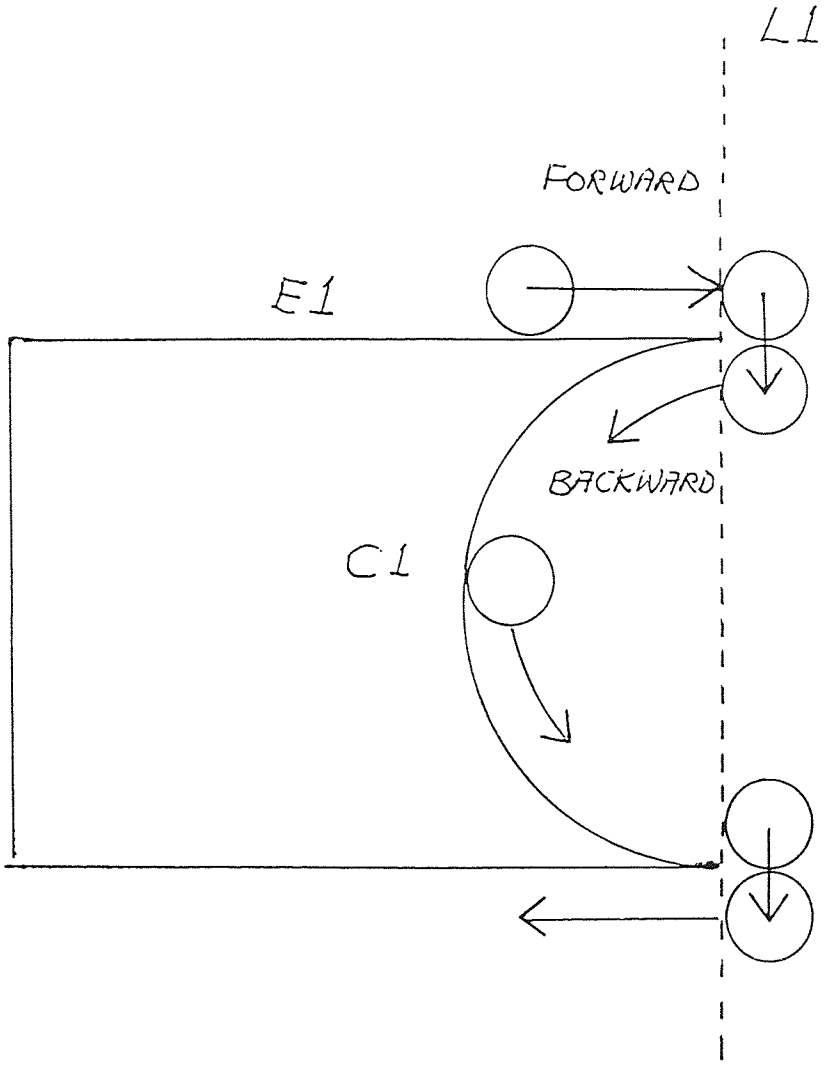


FIGURE 8.10a: Omit case (radius too small)

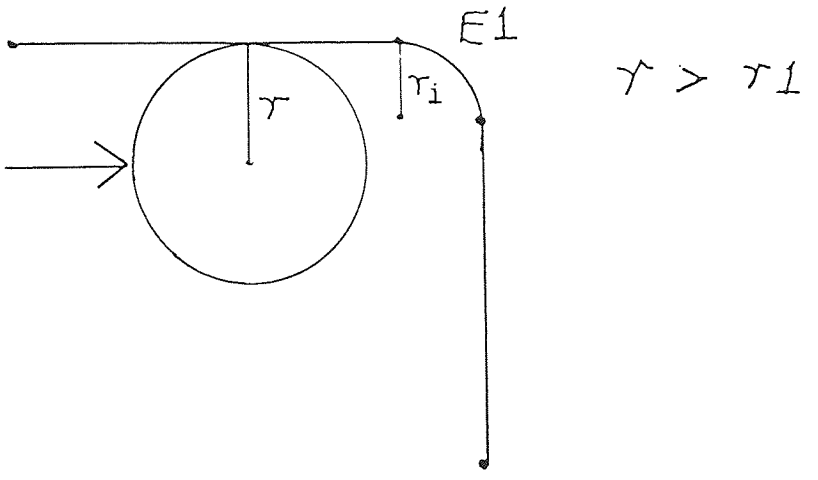


FIGURE 8.10b Omit case (edge too small)

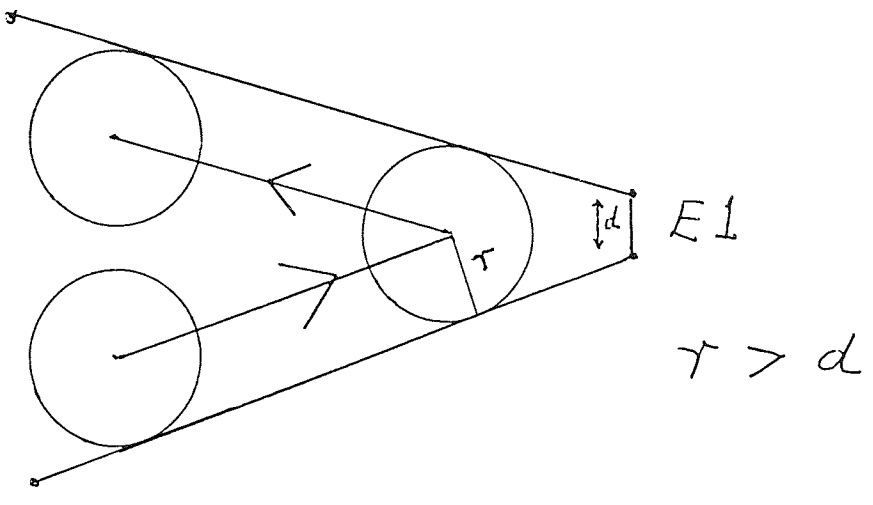


FIGURE 8.11a Double function (convex edge)

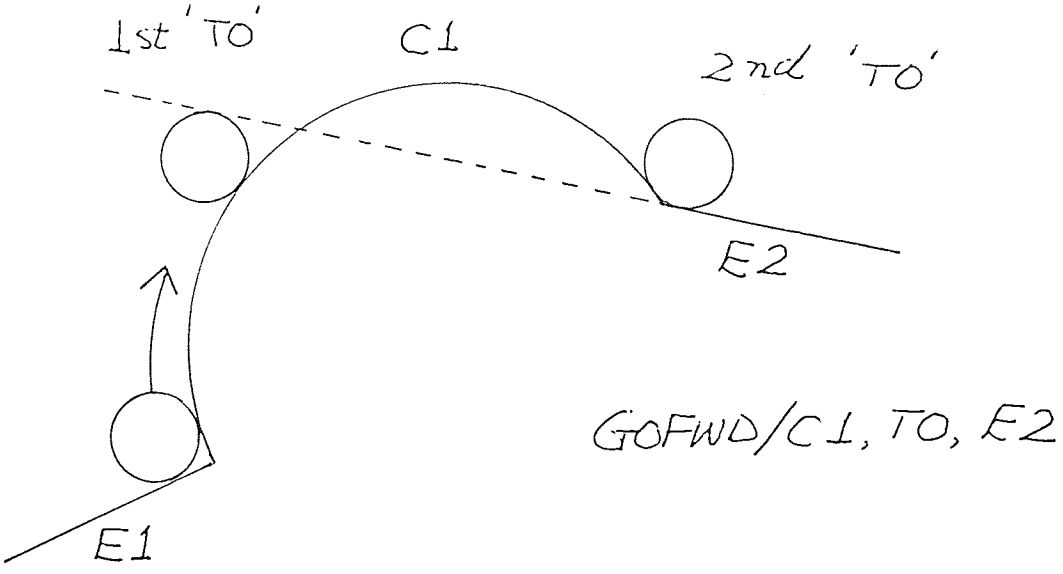


FIGURE 8.11b Double function (concave edge)

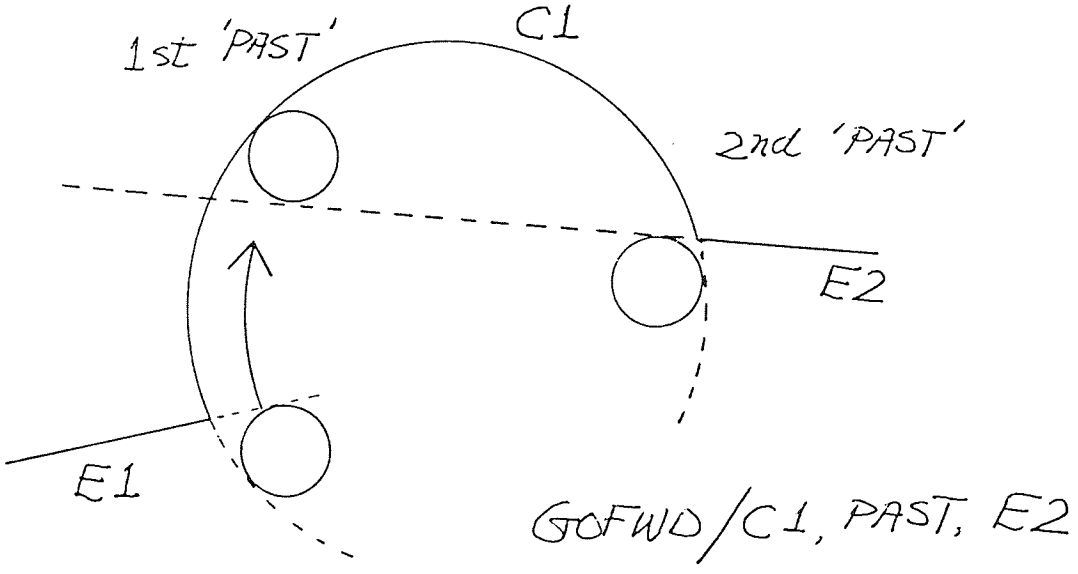


FIGURE 8.11c Double function (two possible positions)

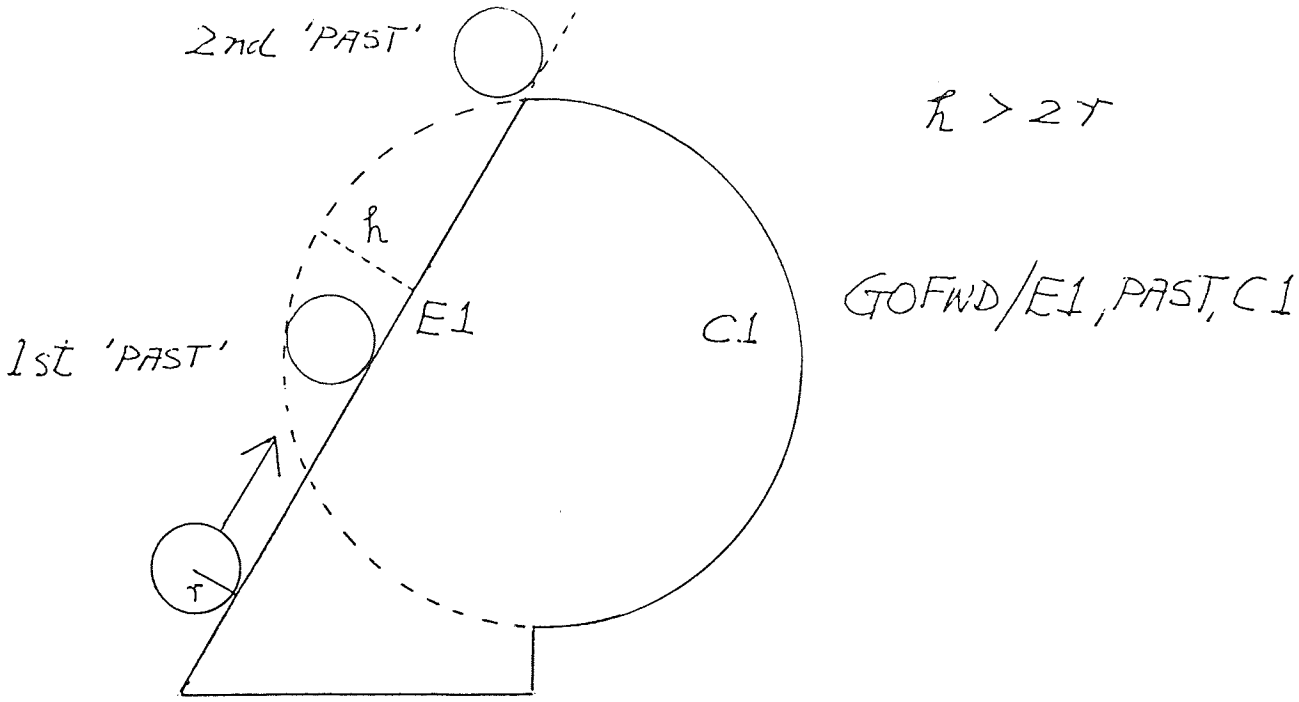


FIGURE 8.11d Double function (only one possible position)

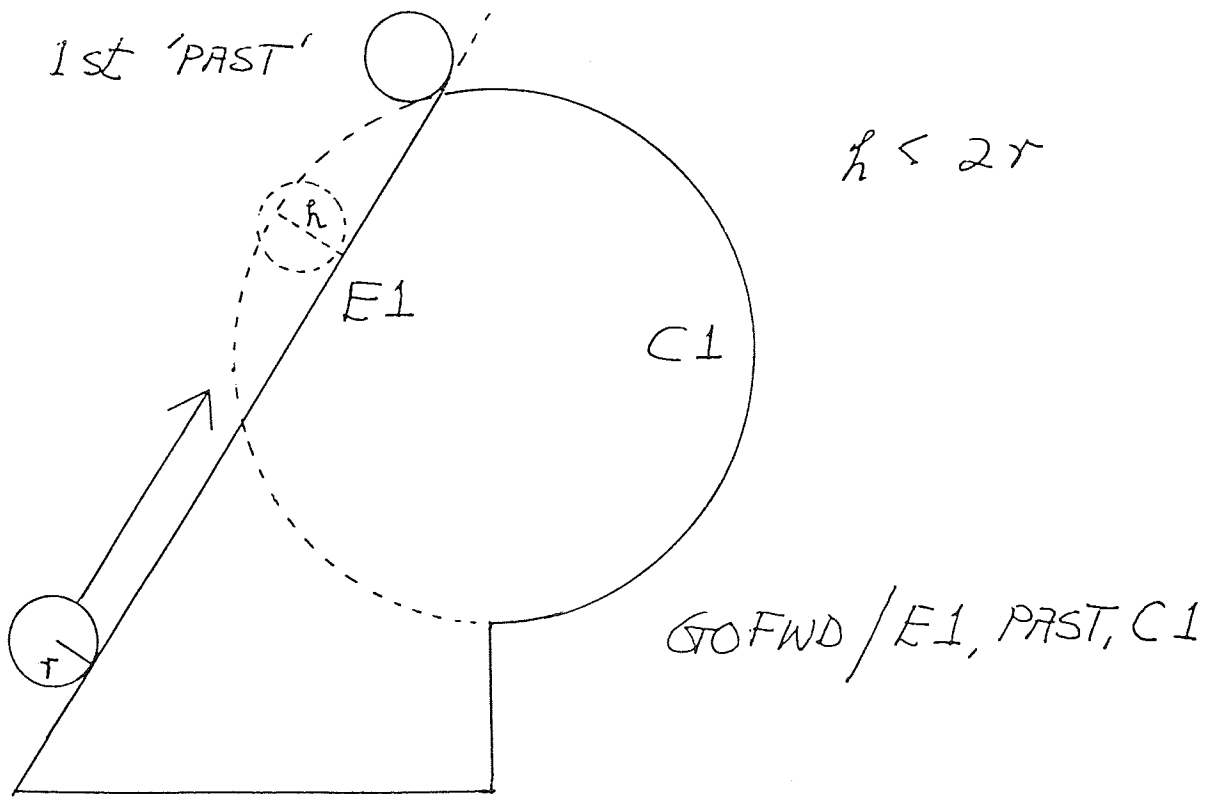


FIGURE 8.12a Segment function

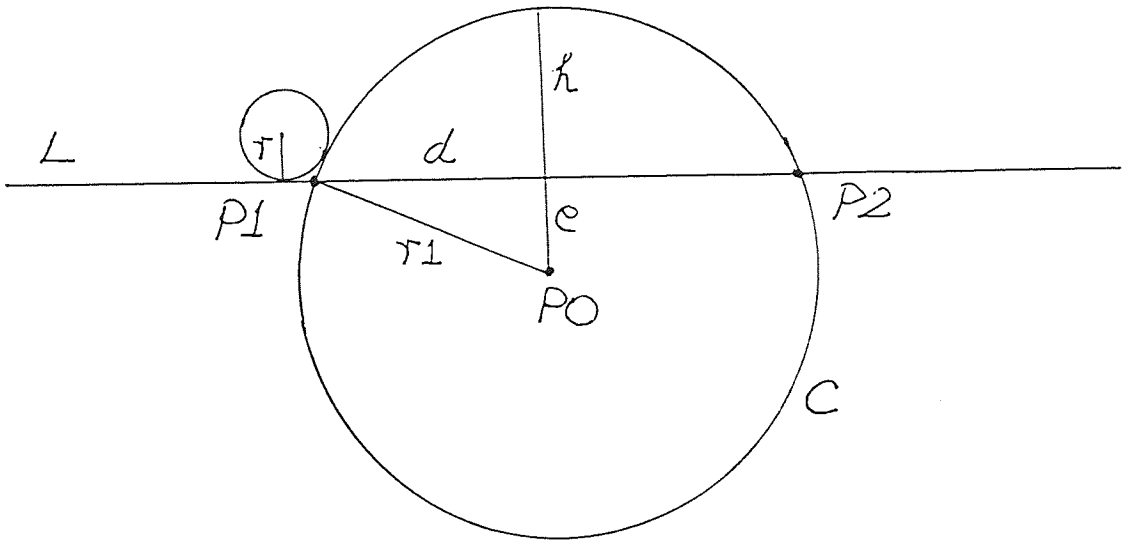


FIGURE 8 12b Segment function (two possible positions when h > 2r

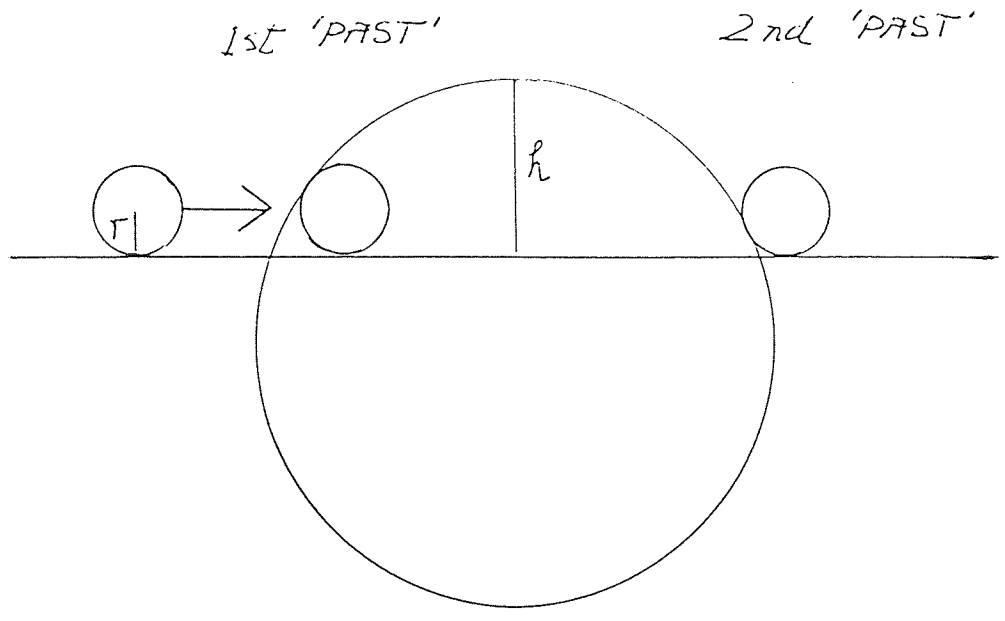


FIGURE 8.12c Segment function (only one possible position when h < 2r

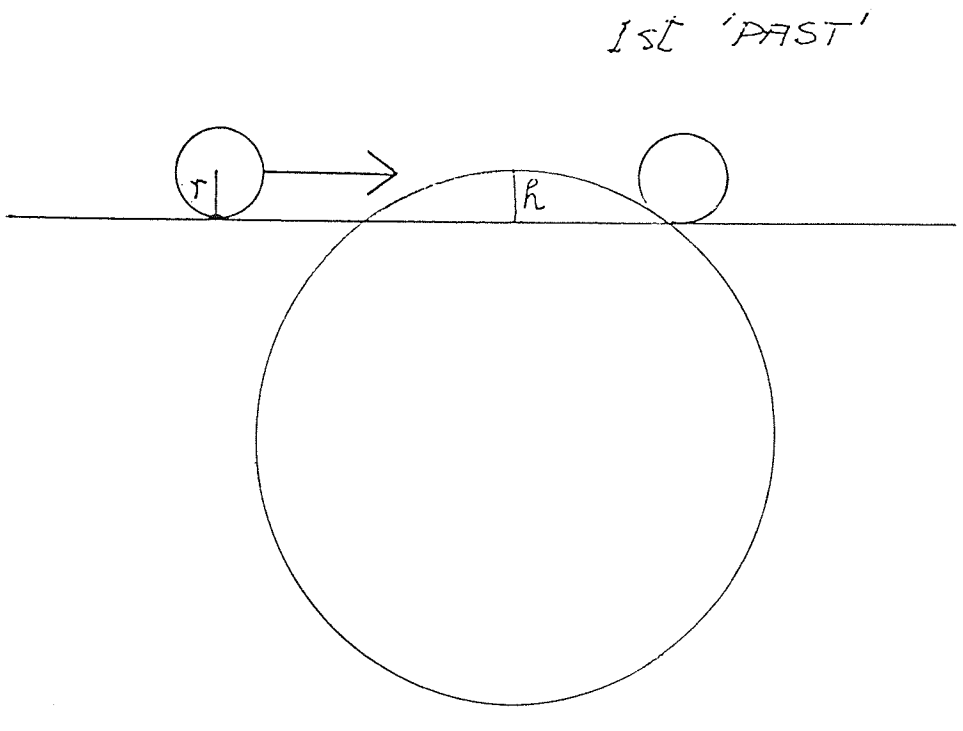


FIGURE 8.13a APT thickness statement example

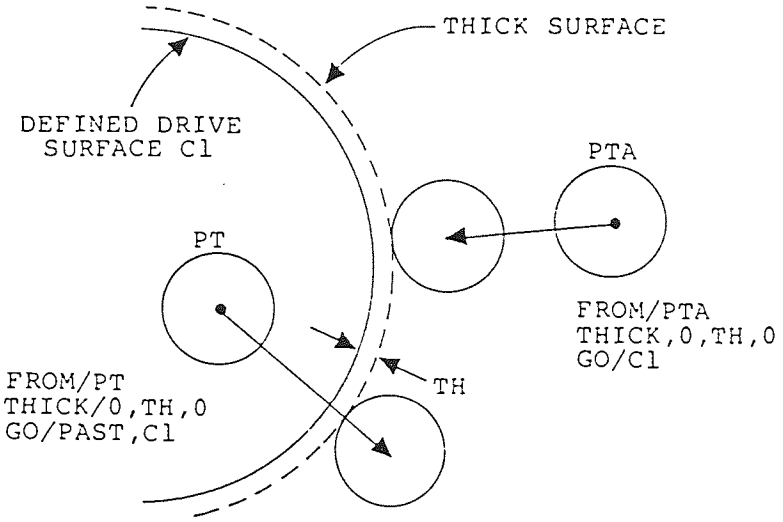
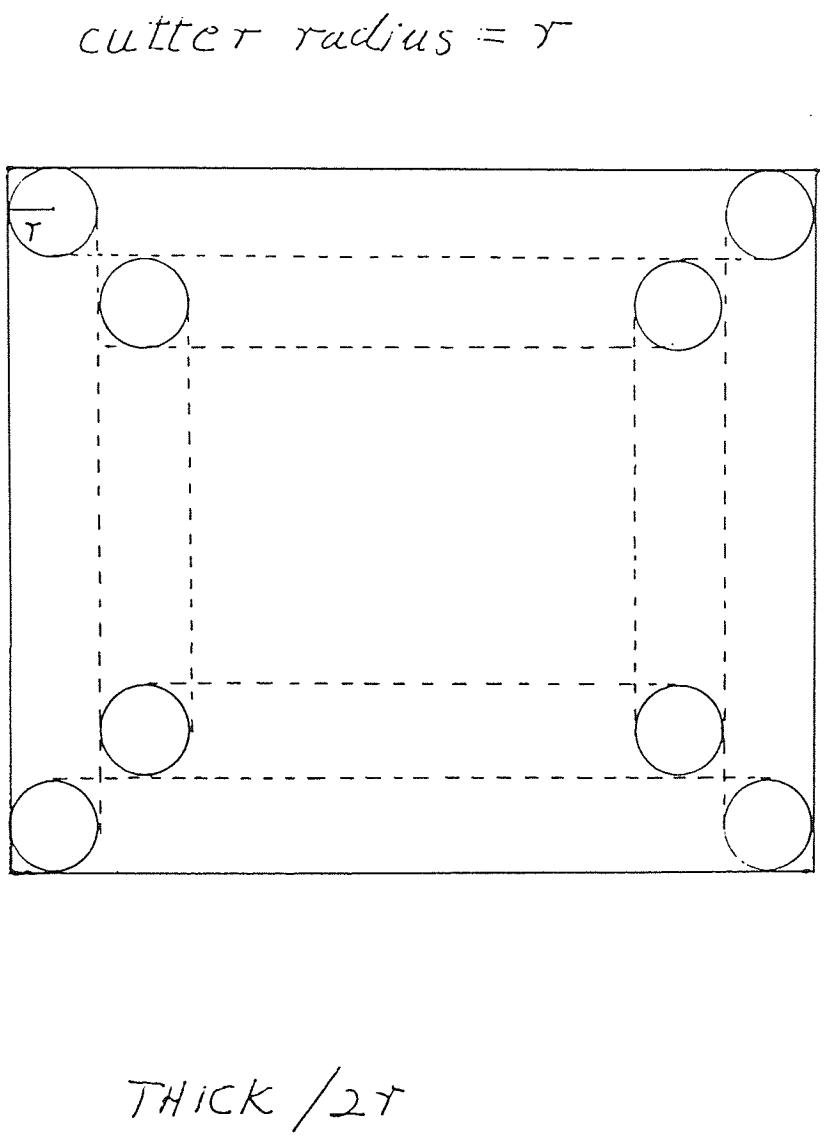


FIGURE 8.13b Pocketing via thickness statement



9. WORKED EXAMPLE OF ROMAPT

To demonstrate the capability of the ROMAPT CAD/CAM system. an engineering part has been created by using ROMULUS and the the geometric model is machined by using ROMAPT to generate the cutter path automatically (Figure 9.1). The following describes how the part is designed.

9.1. DESIGN

A face of the part can be defined as a set of ordered lines joining various points in a closed manner:

LINE point 1 point 2

(meaning: join point 1 and point 2 by a straight line.)

LINE TO point 3

(meaning: join point 2 and point 3 by a straight line.)

. .....

LINE TO point 1

(meaning: complete the loop by joining a straight line to the

first point, point 1.)

The straight line type used above can be circular arc type.

The above procedure is shown in Figure 9.2 and the face (F0) created is shown in Figure 9.3.

The face F0 is then 'swept' (i.e. linear sweep) along a vector direction (in this case, the X-axis) to give it a solid volume (Figure 9.4).

To complete the design, 3 extra shapes i.e. 1 rectangular block and 2 cylinders are needed (as shown in Figures 9.5, 9.6 and 9.7).

Rectangular block primitive can be created by the command:

CUBE point x, y, z

where,

point: is the centre of the bottom face of the cube.

x, y, z: is the dimension of the cube in X- Y- Z-axis.

Cylinders can be created by the command:

CYLINDER point r l

where,

point: is the centre of the bottom face of the cylinder.

r: is the radius of the cylinder.

l: is the height of the cylinder.

These extra shapes are then 'added' to the original body (via the Boolean operation) as shown in Figures 9.8, 9.9 and 9.10.

The completed part and its hidden line view, orthogonal wire-frame projections, silhouette and sectioning view are shown in Figures 9.11 to 9.17 respectively. As can be seen the part can be built up very easily.

9.2 AUTOMATIC APT GEOMETRY GENERATION FOR THE ENGINEERING PART

The name of various faces of the part can be labelled as shown in Figures 9.18 and 9.19. The APT geometry statement of a top face (F1) of the Body Machining part can be generated by the following commands:

APTENQUIRE F1

(meaning: output APT geometry statement for face F1.)

APTENQUIRE POINT F1

(meaning: output APT geometry statments for all point in face F1.)

APTENQUIRE EDGE F1

(meaning: output APT geometry statements for all edges in face F1.)

This is shown in Figure 9.20. Similarly any other points, edges and faces can be output automatically.

9.3 MANUFACTURING

The workpiece for manufacturing can be a blank block of material or precision casting (in this case, only the finishing cut is required).

For this exercise, a blank block of aluminium 54mmX120mmX85mm is used (Figure 9.21a). The NC machine used was a Cintimatic 3VT milling machine with Acramatic 5 A controller in Aston University (Figure 9.21b). The block is mounted and clamped as shown (Figure 9.22).

The machining plan consists of the following steps:

1. Mill around the part along the outer profile (Figure 9.23).

2. Mill around the inner profile of the part (Figure 9.24).

3. Mill pockets as necessary.

4. Repeat the above steps for various machining depth.

9.4 CONSTRUCTION OF PLANES

As a milling machine can only cut to a certain depth, there is a need for the construction of a set of parallel horizontal planes corresponding to the cutting depth required. (Ususally, the cutting depth is a fraction of the cutter radius). These planes can be generated automatically by the command:

CNC PLANE CURSOR

(The cursor is used to define the upper and lower limits for the planes.)

The resultant planes are shown in Figures 9.25, 9.26 and 9.27.

9.5 MACHINING OF OUTER PROFILE

The outer profile of the part can be considered to be made of 4 edges: E7, E4, E21 and E18 as shown in Figure 9.28.

The cutter path for the outer profile can be generated automatically by the command:

PROFILE [edge1, edge2...] [LOOP] [tool] [radius] [options]

where,

[edge]: name of edges.

[LOOP]: options for the whole loop.

[tool]: options for either flat-end or ball-end cutter.

[radius]: radius of the cutter.

[options]: either the current selected face or user-defined Z-planes.

The cutter path for cutter radius of 4.0 and 19.05 mm are shown in Figures 9.29 and 9.30.

9.5.1 GENERATION OF APT MOTION STATEMENTS

The relationship (topology) of various geometric entities: points, edges and faces of the outer profile is shown in Figure 9.31.

The data structure required for the generation of APT motion statements can be set up either manually or automatically:

CNC [LOOP] [option 1] [option 2] [option 3]

where,

[LOOP]: is the key word for the generation of the data structure for the loop.

[option 1]: indicates whether the loop is OPEN or CLOSE.

[option 2]: indicates whether the cutter is situated IN, ON, OUT of the loop.

[option 3]: indicates whether the MANUAL or AUTOMATIC mode is required.

The manual method allows more flexibility (to suit the technological requirement of the machining). The manual method is shown in Figure 9.32. The data structure generated is shown in Figure 9.33. The automatic generation of the required APT motion statement is effected by the command:

CNC [APTLOOP] [options]

where,

[APTLOOP]: is a keyword for the generation of the APT motion statement for the loop.

[option]: define either the currently selected face or user-defined Z-planes.

The corresponding APT motion statements is shown in Figure 9.34. (This assumes the required geometry statements have been generated previously by using the APTENQUIRE command.)

The header, tolerance, cutter definition and end statements can be generated by ROMAPT as shown in Figure 9.35.

Finally, post-processor, coolant, spindle and feedrate statments can be inserted to complete the NC program. (Note that the spindle speed and feed rate can be obtained from workshop machining standard tables recommended by NC/tool manufacturers.)

The cutter path verification for the outer profile is shown in Figures 9.36 and 9.37.

9.6 MACHINING OF THE INNER PROFILE

The relationship (topology) of various geometrical entities: points edges and faces of the inner profile is shown in Figure 9.38. The cutter path for the inner profile is generated (for a cutter radius of 6.35 mm) as shown in Figure 9.39.

ROMAPT can also recommended a maximum cutter size for certain critical areas (i.e. passing the gap between loops) as shown in Figure 9.40.

9.6.1 GENERATION OF APT MOTION STATEMENTS

The data structure required for the generation of APT motion statement for the whole loop is generated automatically as shown in Figures 9.41 and 9.42. The APT motion statements are generated as shown in Figure 9.43. Similarly, the APT motion statement can be generated for the cylinder. The cutter path is shown in Figure 9.44. Finally, the task is completed by an area clearance operation.

CNC CLEAR CURSOR

This is followed by the definition of the area by 2 cursor positions (as shown in Figure 9.45). The cutter path and the APT motion statement generated are shown in Figures 9.46a, 9.46b and 9.47.

The cutter path verification for the whole inner profile is shown in Figures 9.48 and 9.49.

Note that the machining is in 2 steps each cut to a depth of 5 mm.

9.7 MACHINING OF SLANTING PLANES OF THE PART

The work piece is rotated 180 degree along the Y-axis and mounted and clamped as shown in Figure 9.50. (This particular machining strategy is adopted because of the clamping method used in the Aston University NC Laboratory.)

As the NC machine can only cut to a depth of 5 mm each time, hence a number of parallel horizontal planes each separated by 5 mm are required for the machining. This is carried out as shown in Figure 9.51, 9.52 and 9.53.

For each layer, an area clearance operation is carried out as shown in Figure 9.54. The resulting cutter path and APT motion statements are shown in Figures 9.55 and 9.56. The whole NC cutter path verification is shown in Figures 9.57, 9.58 and 9.59.

Finally the two slanting planes can be machined similarly by the area clearance operation. The cutter path verification is shown in Figures 9.60, 9.61 and 9.62. The finished part is shown in Figure 9.63.

9.8 3D MACHINING

To demonstrate that ROMAPT is a full 3D NC machining system, the cutter path for a cylindrical part surface with a cylindrical island is shown in Figures 9.64 and 9.65.

9.9 CONCLUSION

As can be seen, a working CAD/CAM system (ROMAPT) for prototype making of Lucas engineering parts has been developed in the Lucas Research Centre.

It is envisaged that Lucas operating companies will use this new technology to improve the lead time for the prototype making of their various engineering parts.

FIGURE 9.1 Lucas Girling Part (Body Machining 412X6475)

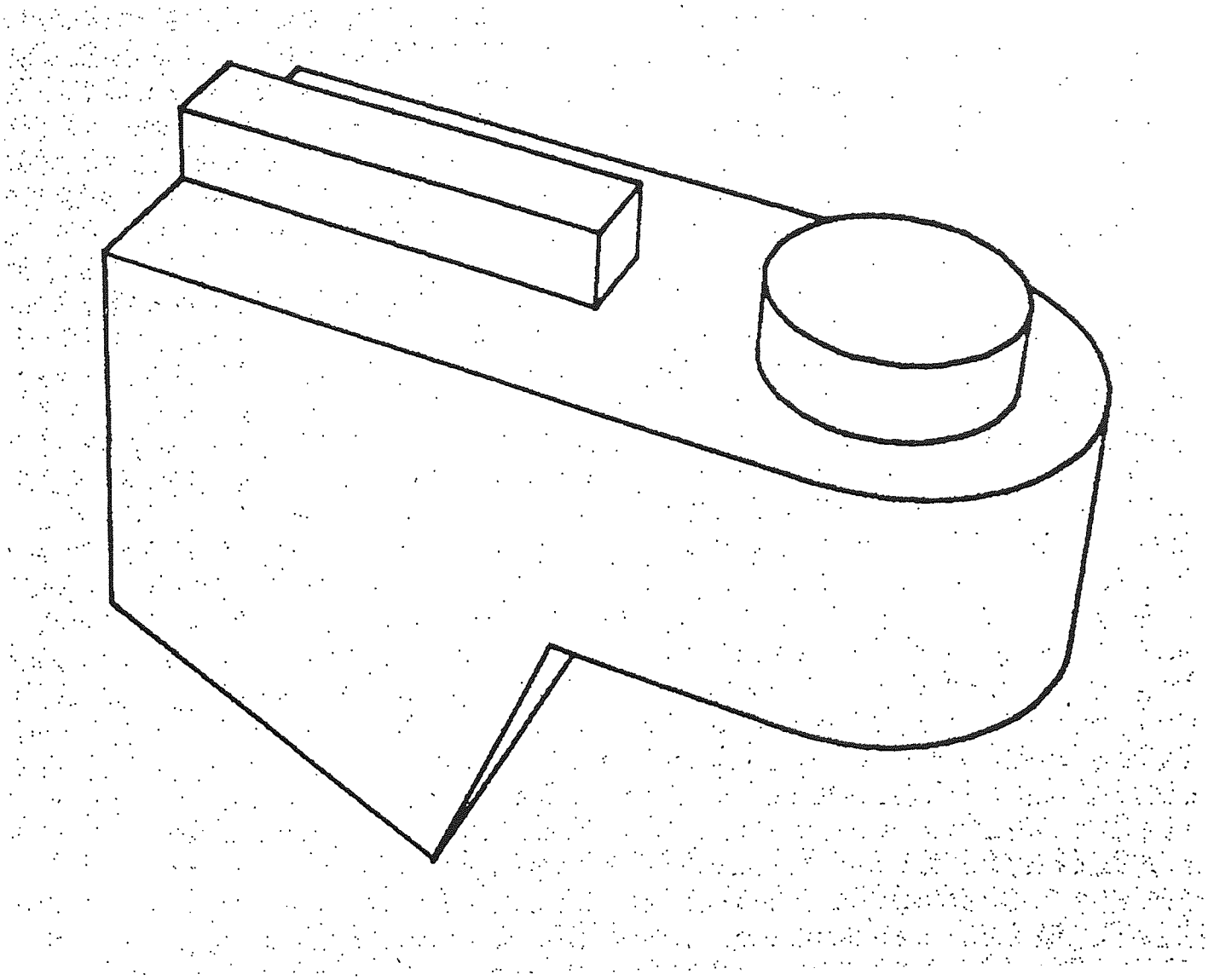


FIGURE 9.2 Romulus procedure for creating a plannar face

```
ROMULUS
* @I C.LUCAS1
@* MONITOR FILE CREATED  11:56 19 JUL 83  BY  BTFC
@*    ROMULUS VERSION 4.0 (LUCAS)
 CREATE B0
  NEW BODY CREATED - NAME: ^-B0
 LIST

  OWNER: ^

  BODIES:  #=B0

 LINE 25.0,91.0,-10 25,0,-10
 LINE TO 25,0,-62
 LINE TO 25,47.4,-82.71
 LINE TO 25.0,61.6,-43.7
 LINE TO 25.0,91.0,-43.7
 LINE TO 25.0,91.0,-10
  1 FACE CREATED
  FACE: F13 HAS A PLANAR SURFACE
@I
*
```

FIGURE 9.3 A planar face created by Romulus

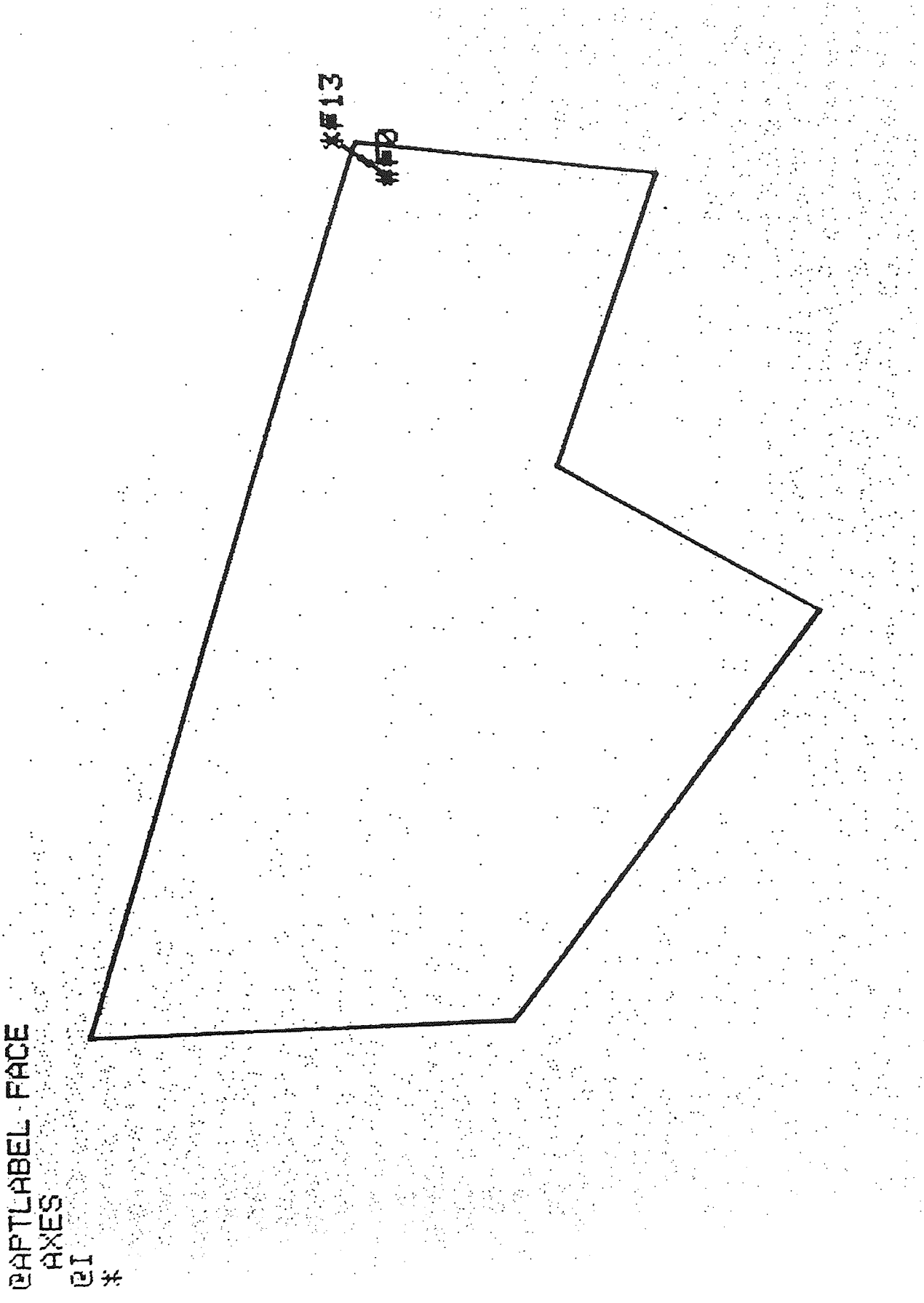


FIGURE 9.4 Volume (BO) generated via sweeping

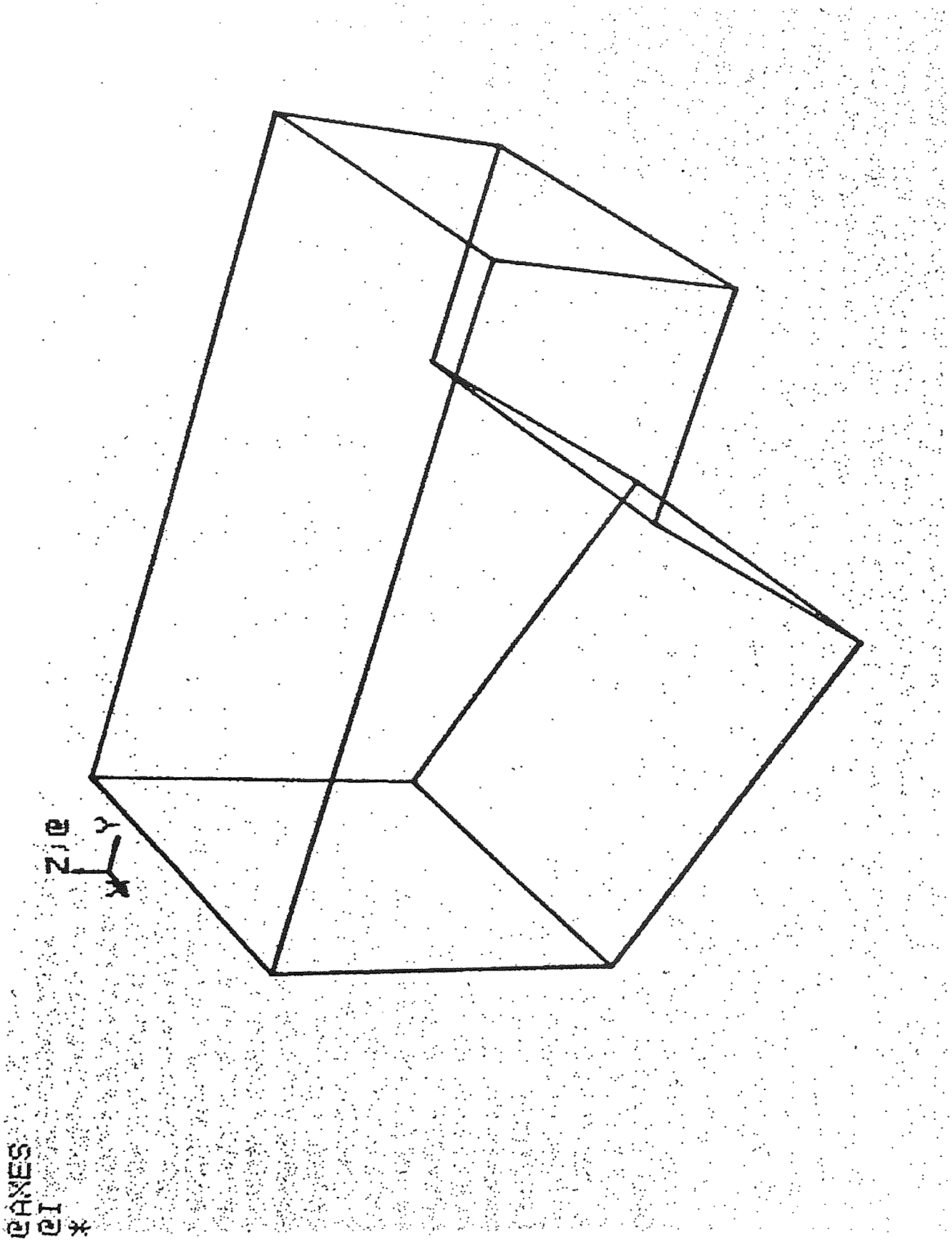


FIGURE 9.5 Procedure for a rectangular block B1

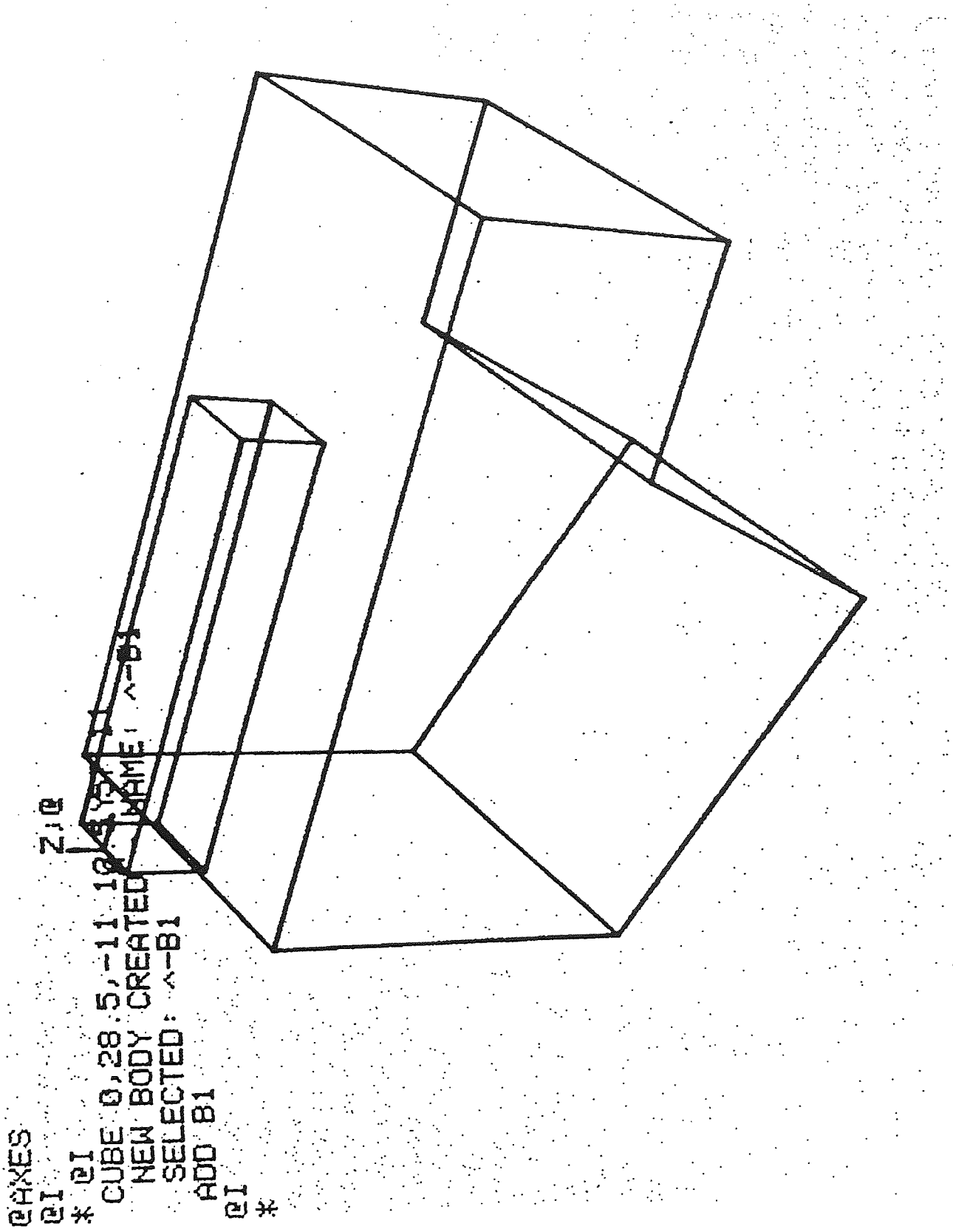


FIGURE 9.6 Procedure for a cylinder B2

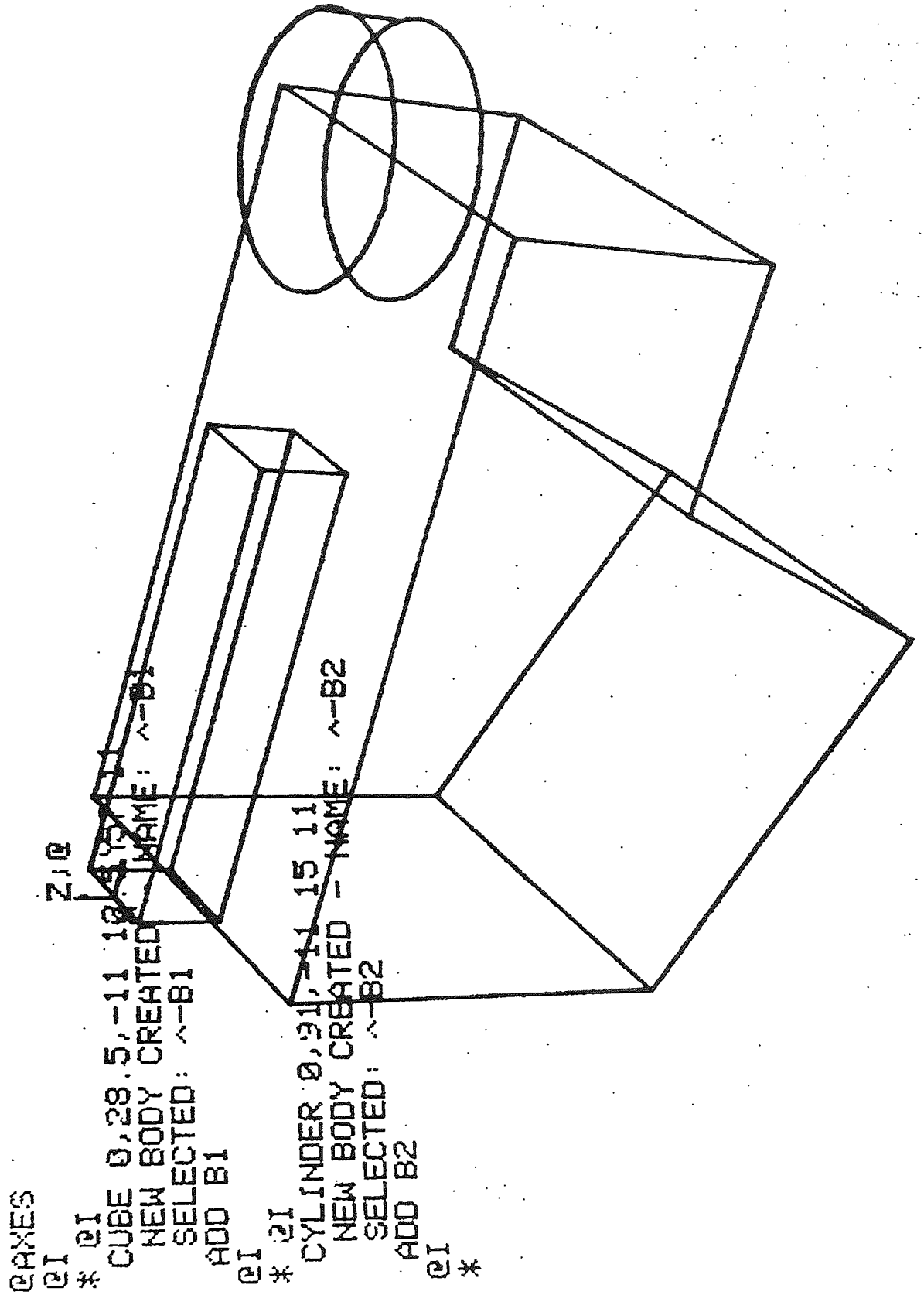


FIGURE 9.7 Procedure for an other cylinder B3

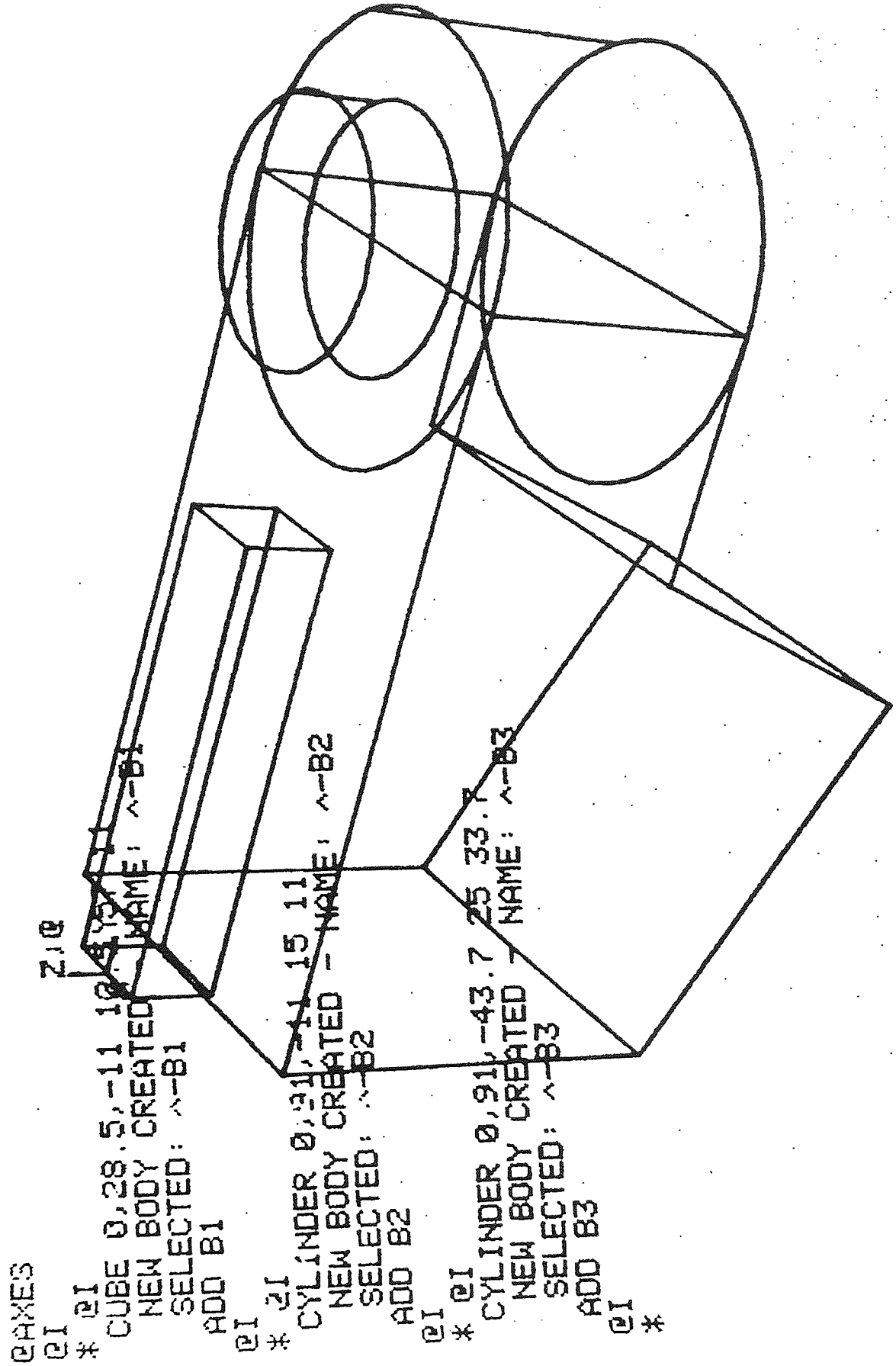


FIGURE 9.8 Boolean operation BO = BO + B3

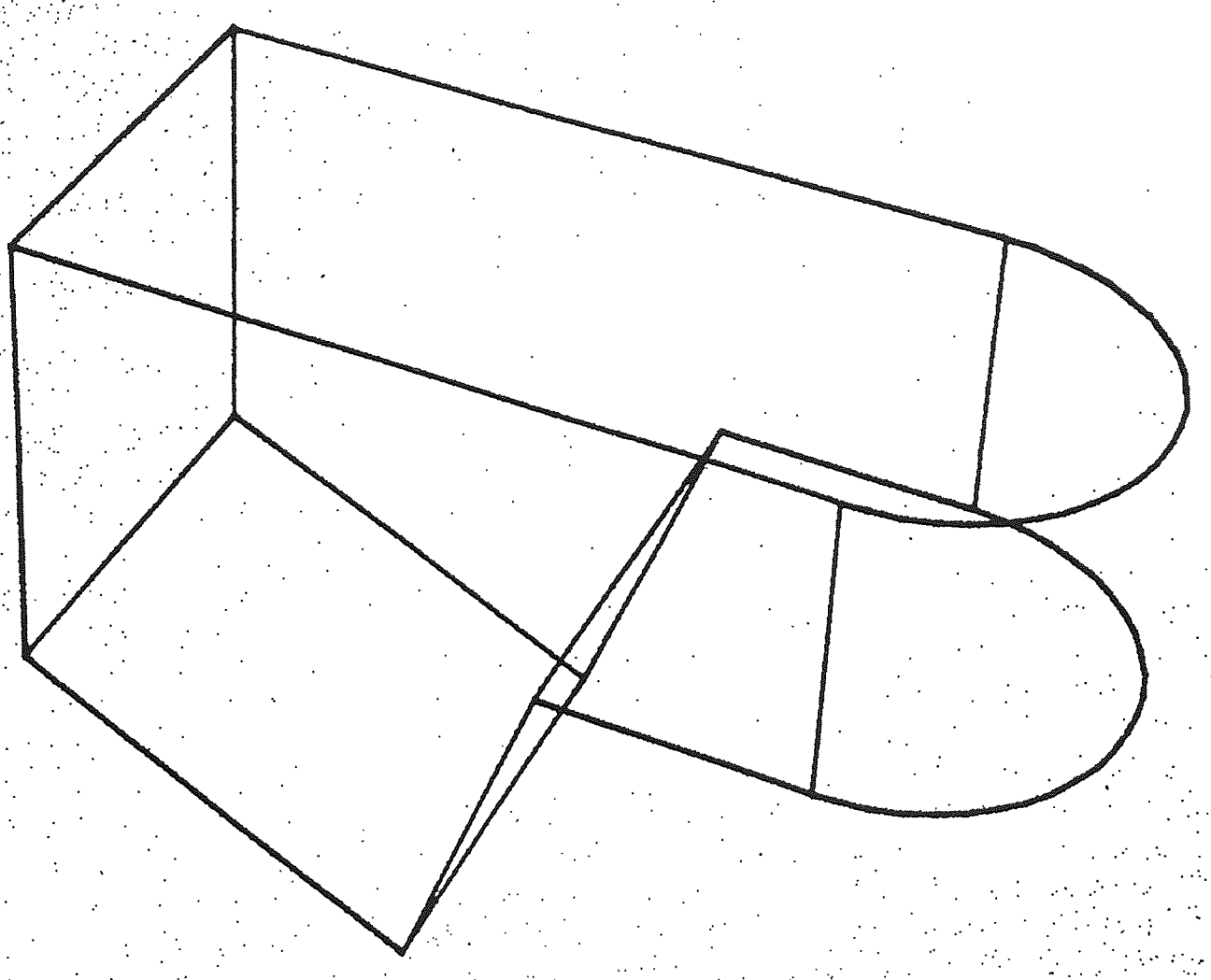


FIGURE 9.9 Boolean operation B0 = B0 + B2

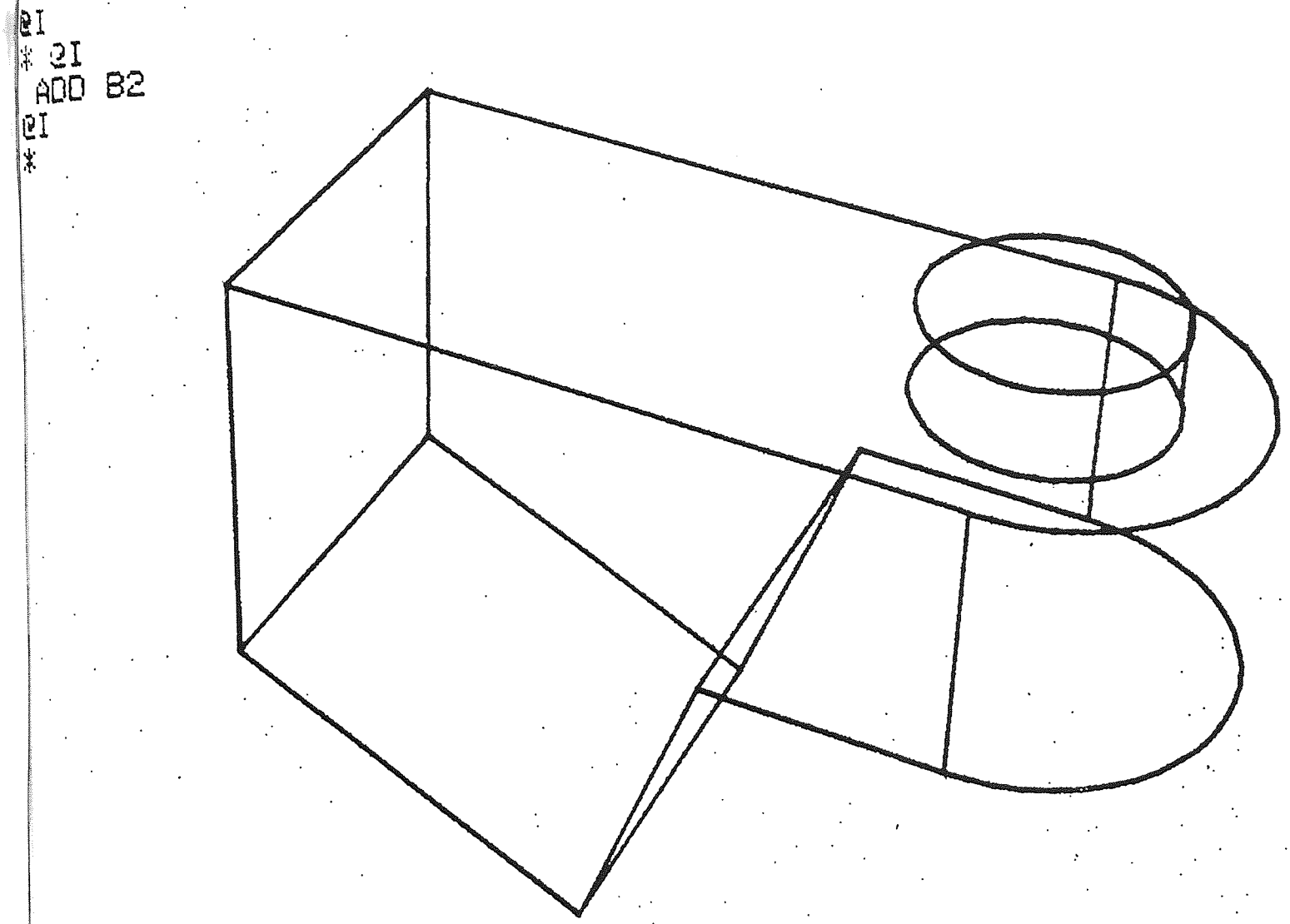


FIGURE 9 10 Boolean operation B0 = B0 + B1

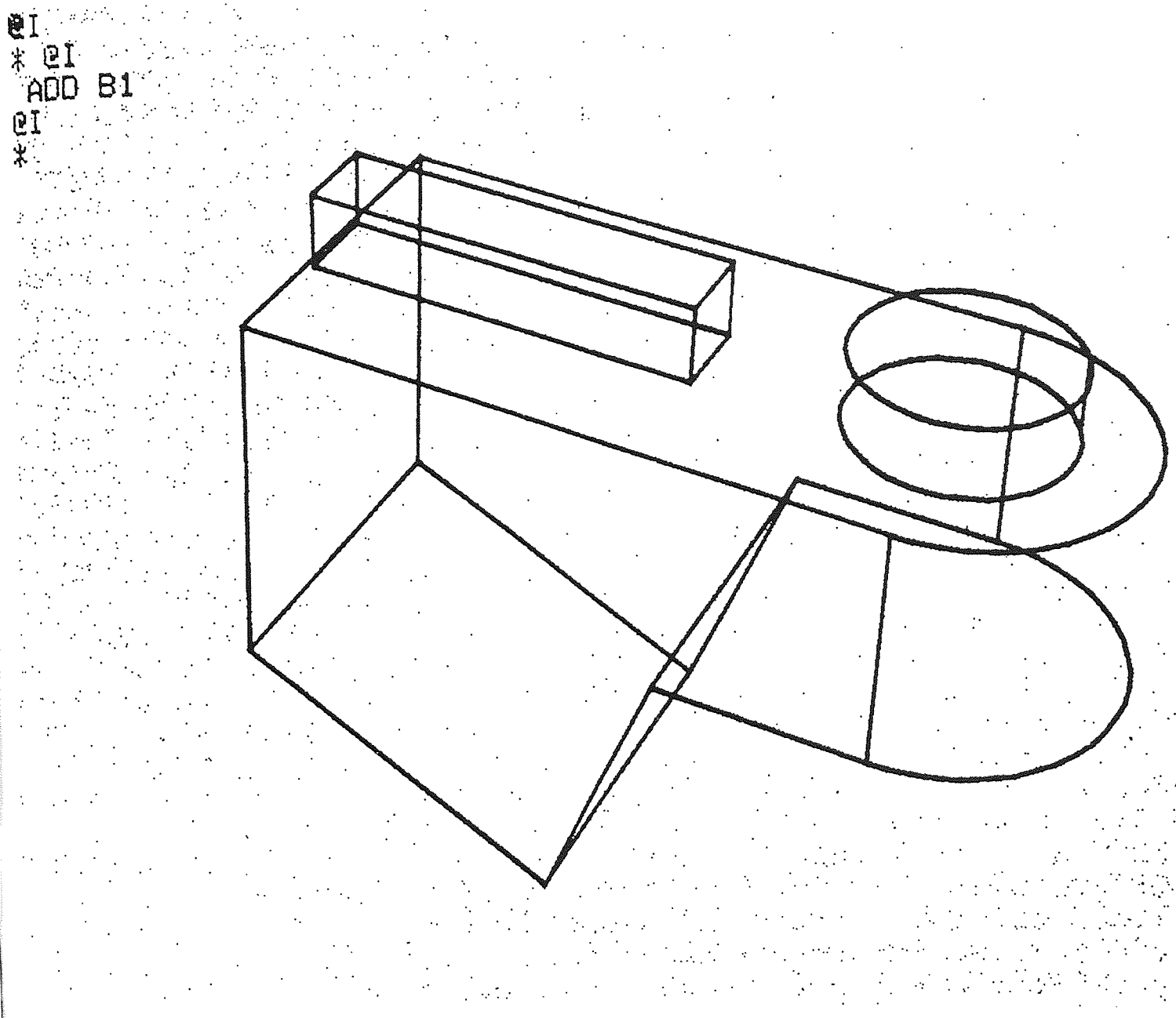


FIGURE 9.11: The completed part

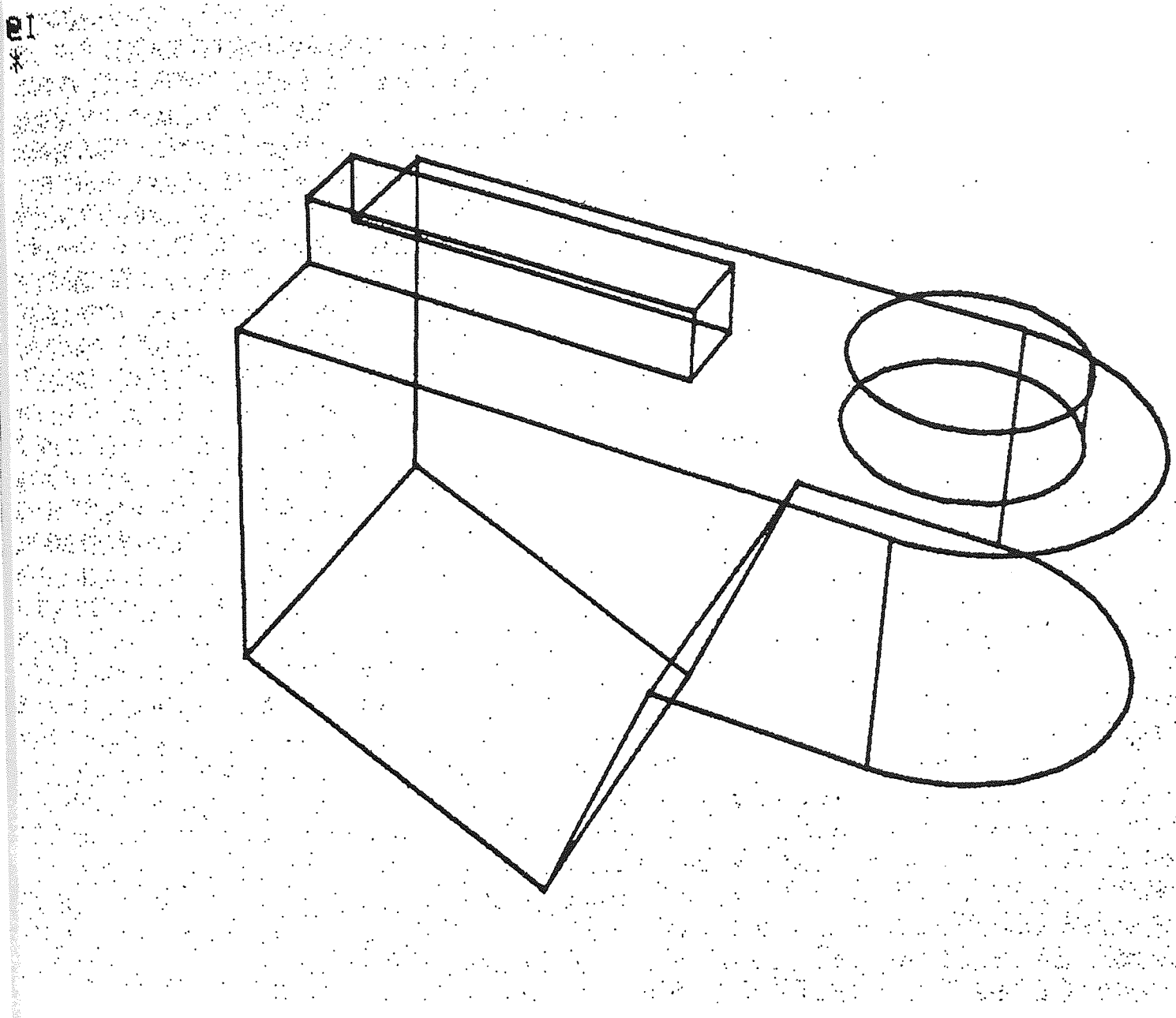


FIGURE 9 12 Hidden line view

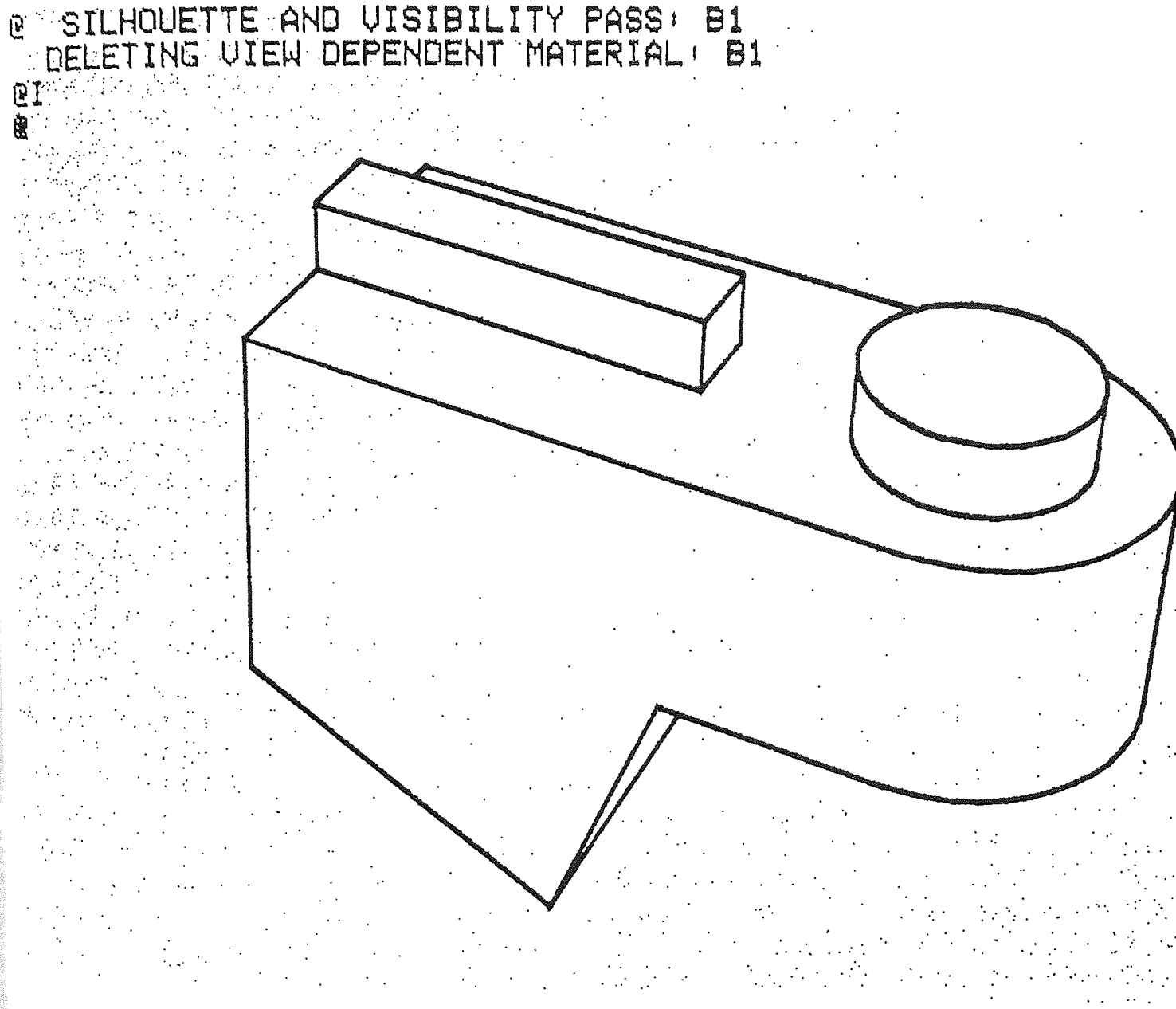


FIGURE 9.13 Orthogonal wire-frame view projections

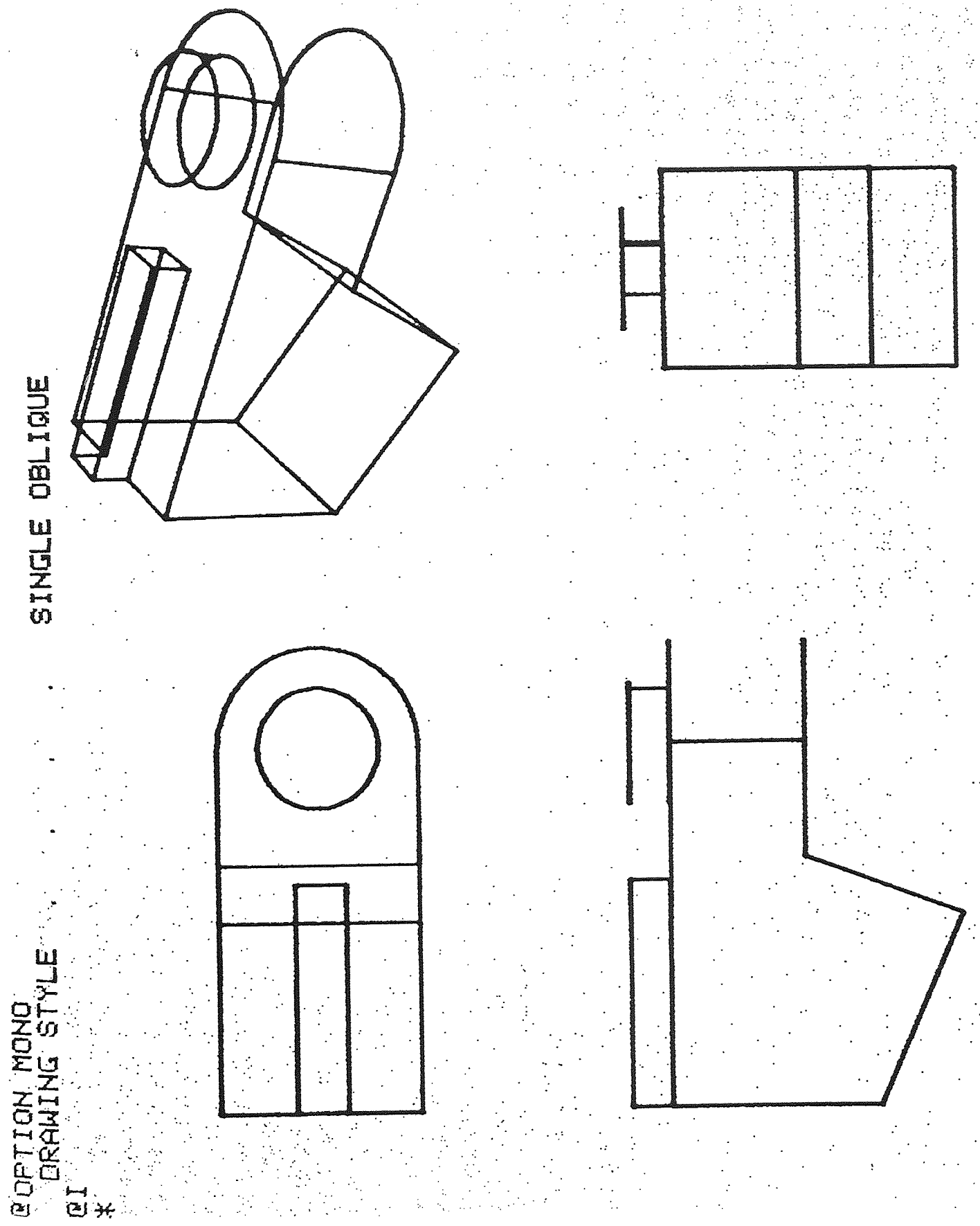


FIGURE 9 14: Silhouette view

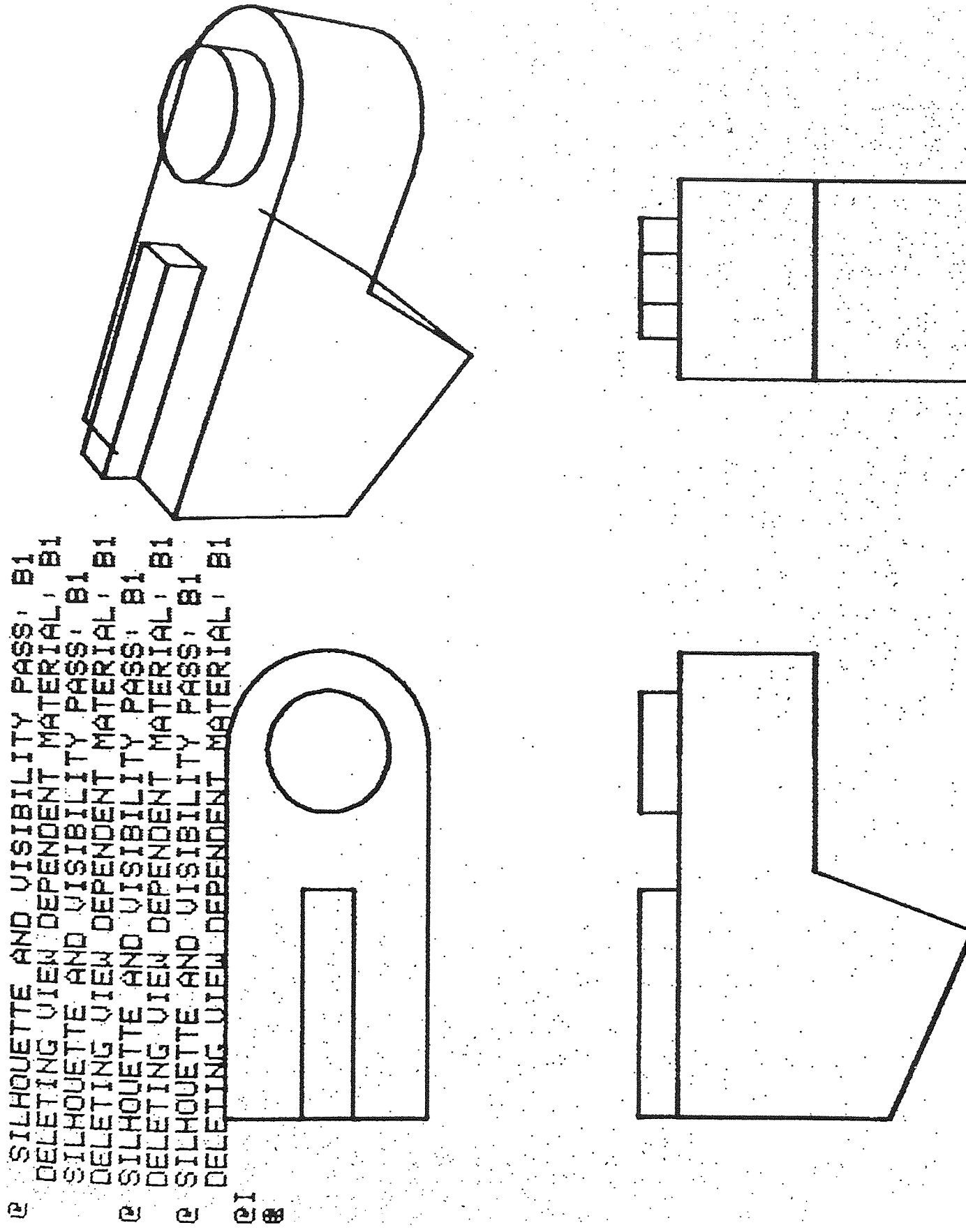


FIGURE 9 15 Procedure for sectioning

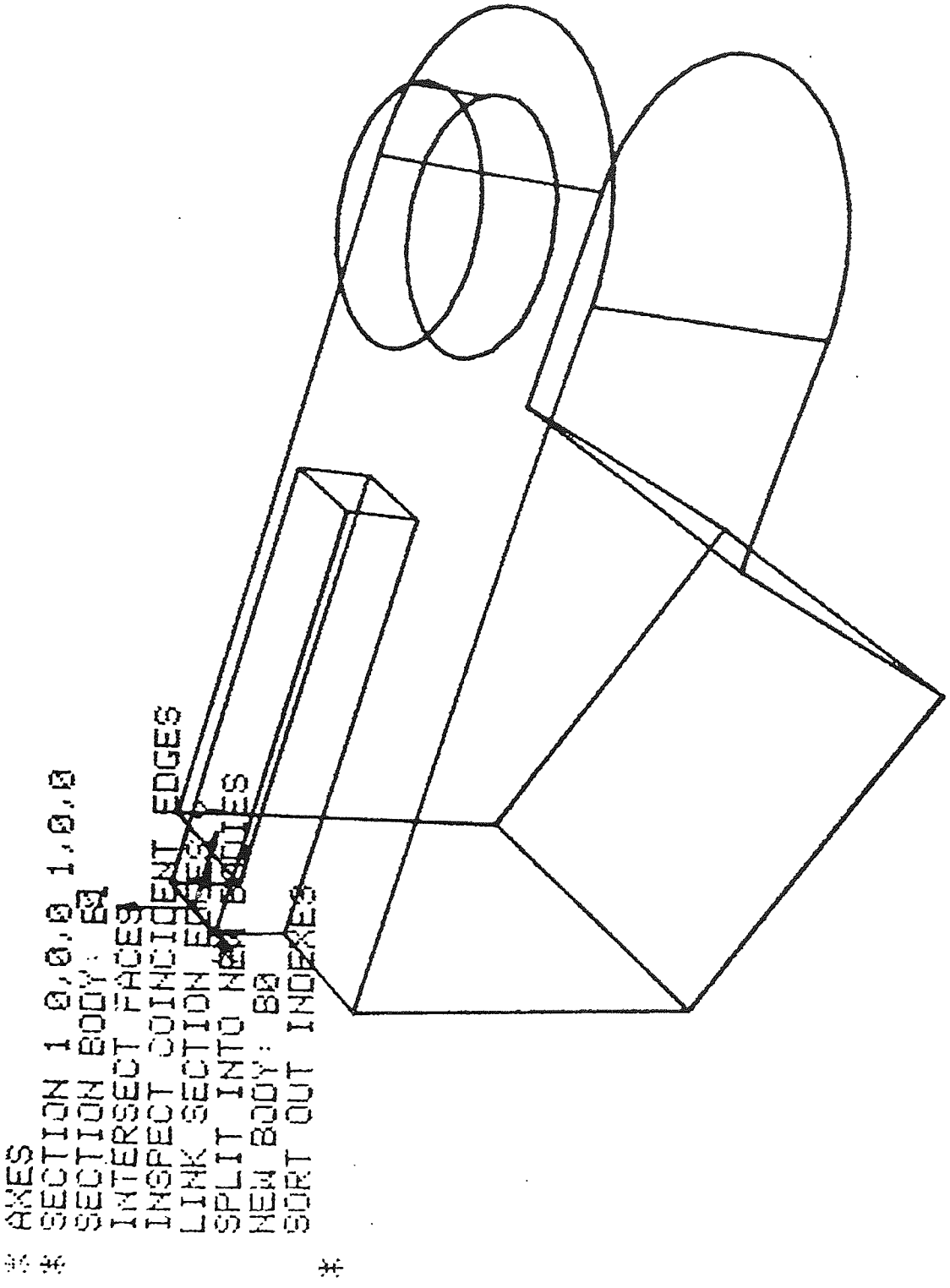


FIGURE 9.16: A view of the cross-section

SELECTED: ^-B0

*

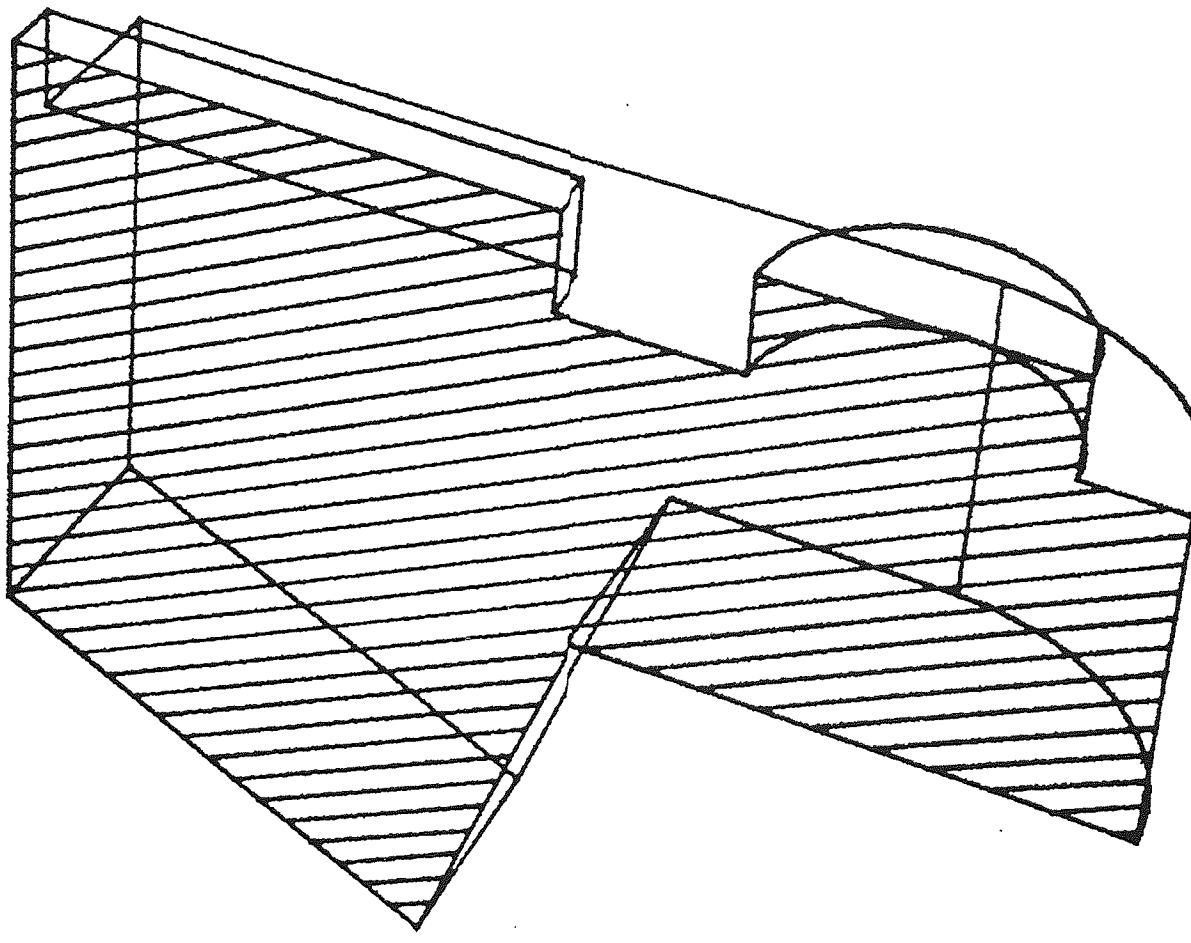
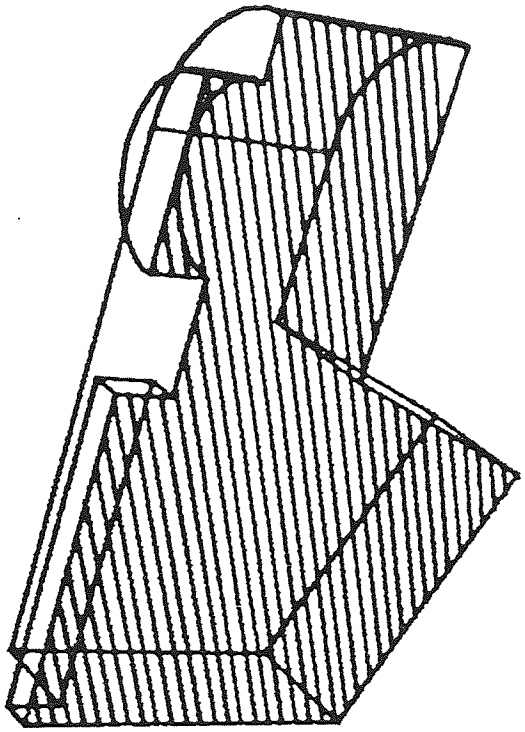
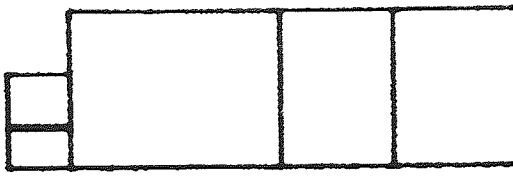
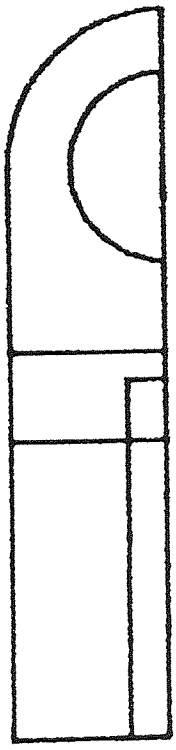
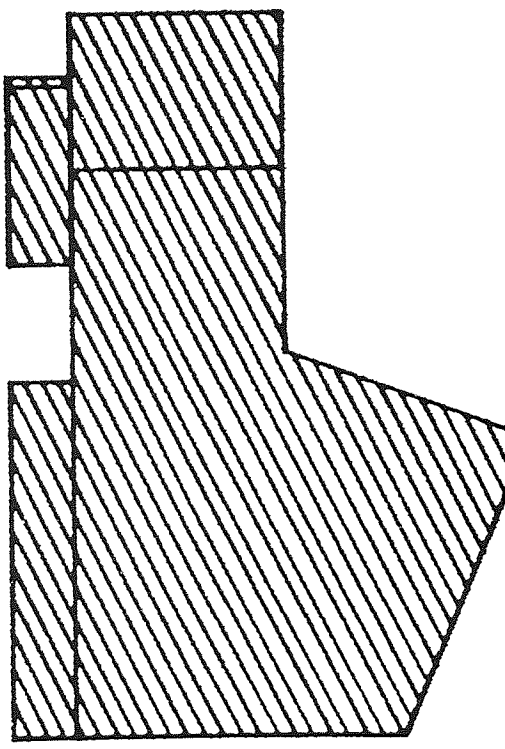


FIGURE 9.17 Orthogonal views of cross-section









SILHOUETTE AND VISIBILITY PASS: B0
DELETING VIEW DEPENDENT MATERIAL: B0
SILHOUETTE AND VISIBILITY PASS: B0
DELETING VIEW DEPENDENT MATERIAL: B0
SILHOUETTE AND VISIBILITY PASS: B0
DELETING VIEW DEPENDENT MATERIAL: B0
SILHOUETTE AND VISIBILITY PASS: B0
DELETING VIEW DEPENDENT MATERIAL: B0

FIGURE 9.18 Labels of faces

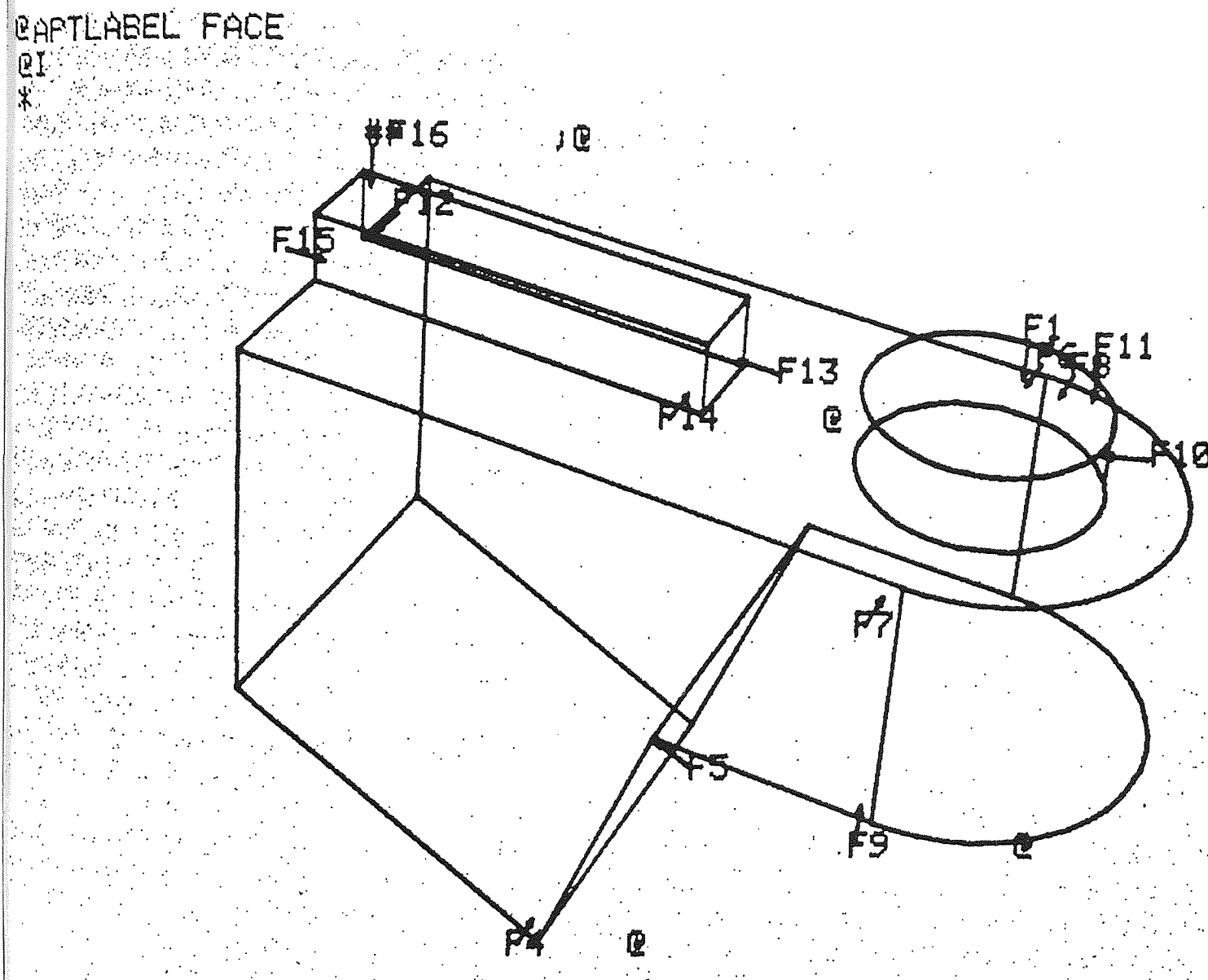


FIGURE 9.19 Labels of points and edges of a top face F1

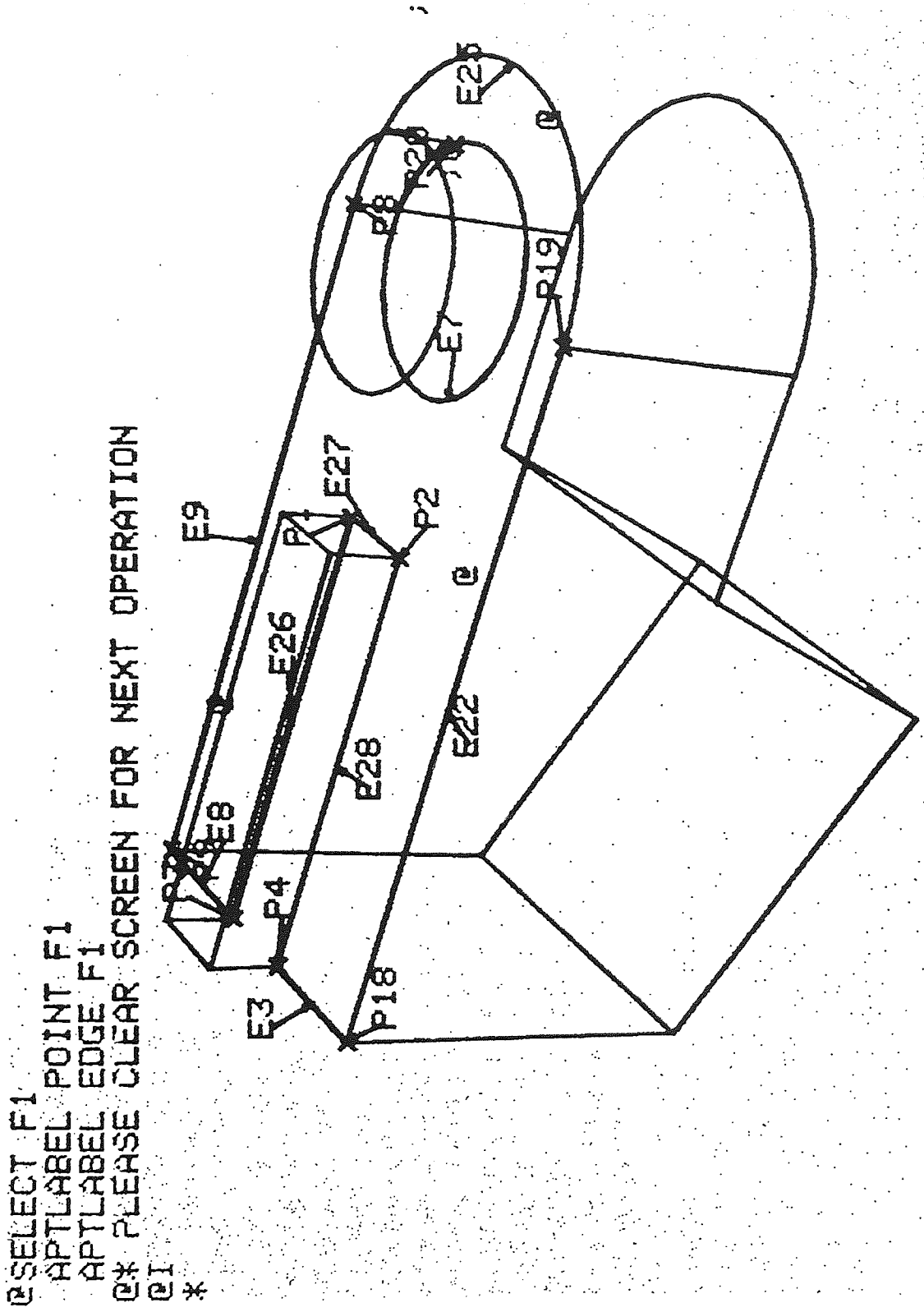


FIGURE 9.20 Automatic generation of APT geometry statements

```
APTE F1
$$ THE FOLLOWING 3 LINES ARE PLANE DEFINITION      FOR   F1
F1P  = POINT/    0.0,0.0,-10.0
F1V  = VECTOR/   0.0,0.0,1.0
F1   = PLANE/    F1P   , PERPTO ,   F1V
APTE POI F1
P8   = POINT/    -25.0,91.0,-10.0
P9   = POINT/    -25.0,0.0,-10.0
P3   = POINT/    -5.25,0.0,-10.0
P1   = POINT/    -5.25,57.0,-10.0
P2   = POINT/    5.24999,57.0,-10.0
P4   = POINT/    5.24999,0.0,-10.0
P18  = POINT/    25.0,0.0,-10.0
P19  = POINT/    25.0,91.0,-10.0
P20  = POINT/    -6.7082,104.416,-10.0
APTE EDG F1
E9   = LINE/     P9   , P8
E8   = LINE/     P9   , P3
E26  = LINE/     P1   , P3
E27  = LINE/     P1   , P2
E28  = LINE/     P4   , P2
E3   = LINE/     P4   , P18
E22  = LINE/     P19  , P18
E25P = POINT/    0.0,91.0,-10.0
E25  = CIRCLE/CENTER,  E25P      , RADIUS ,  25.0
       TO        P8
$$ P19
E7P  = POINT/    0.0,91.0,-10.0
E7   = CIRCLE/CENTER,  E7P       , RADIUS ,  15.0
       TO        P20
$$ P20
```

FIGURE 9.21a: A blank block (the work piece)

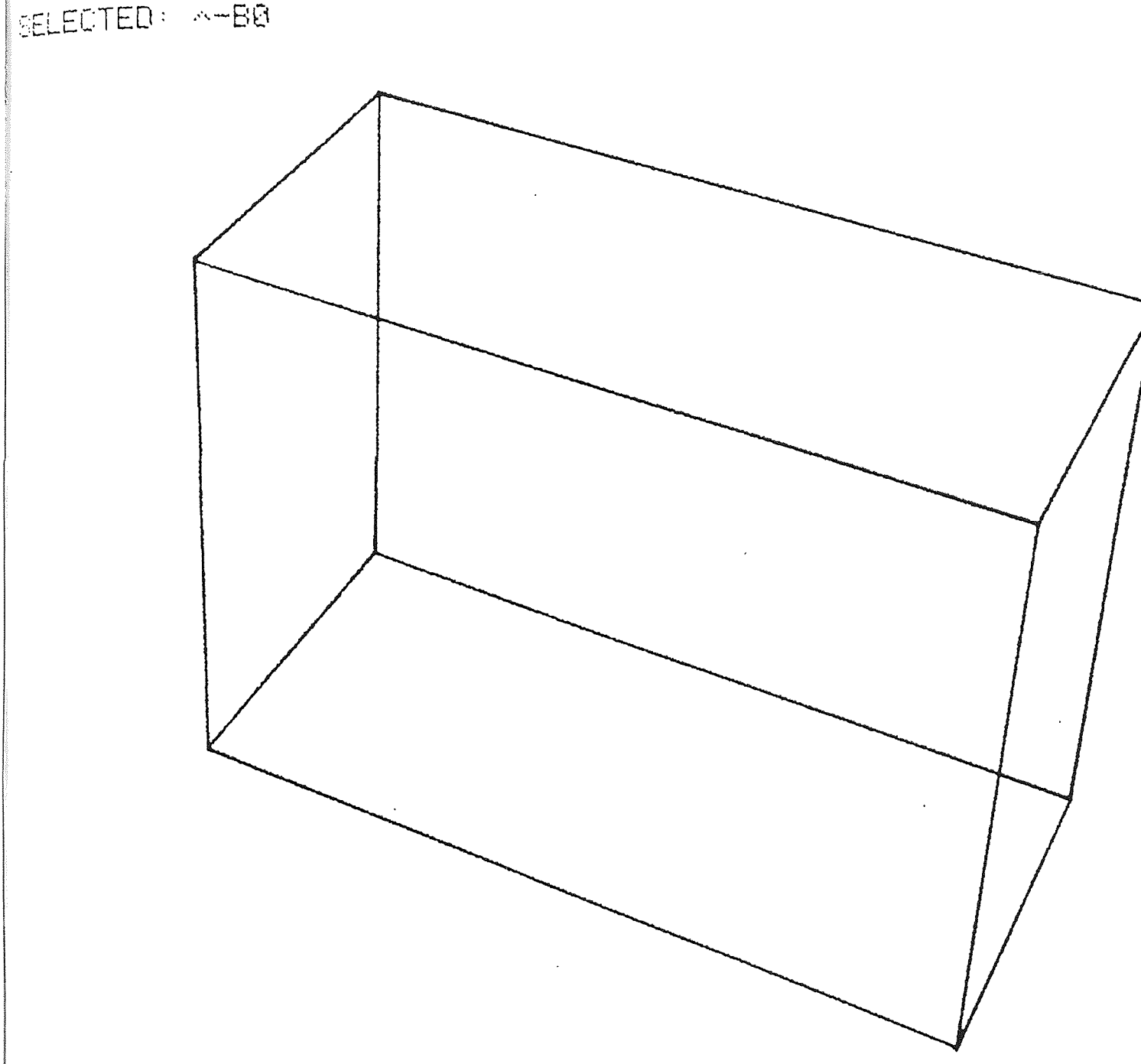


FIGURE 9.21b CINTIMATIC 3VT Milling Machine with Acramatic 5 Controller

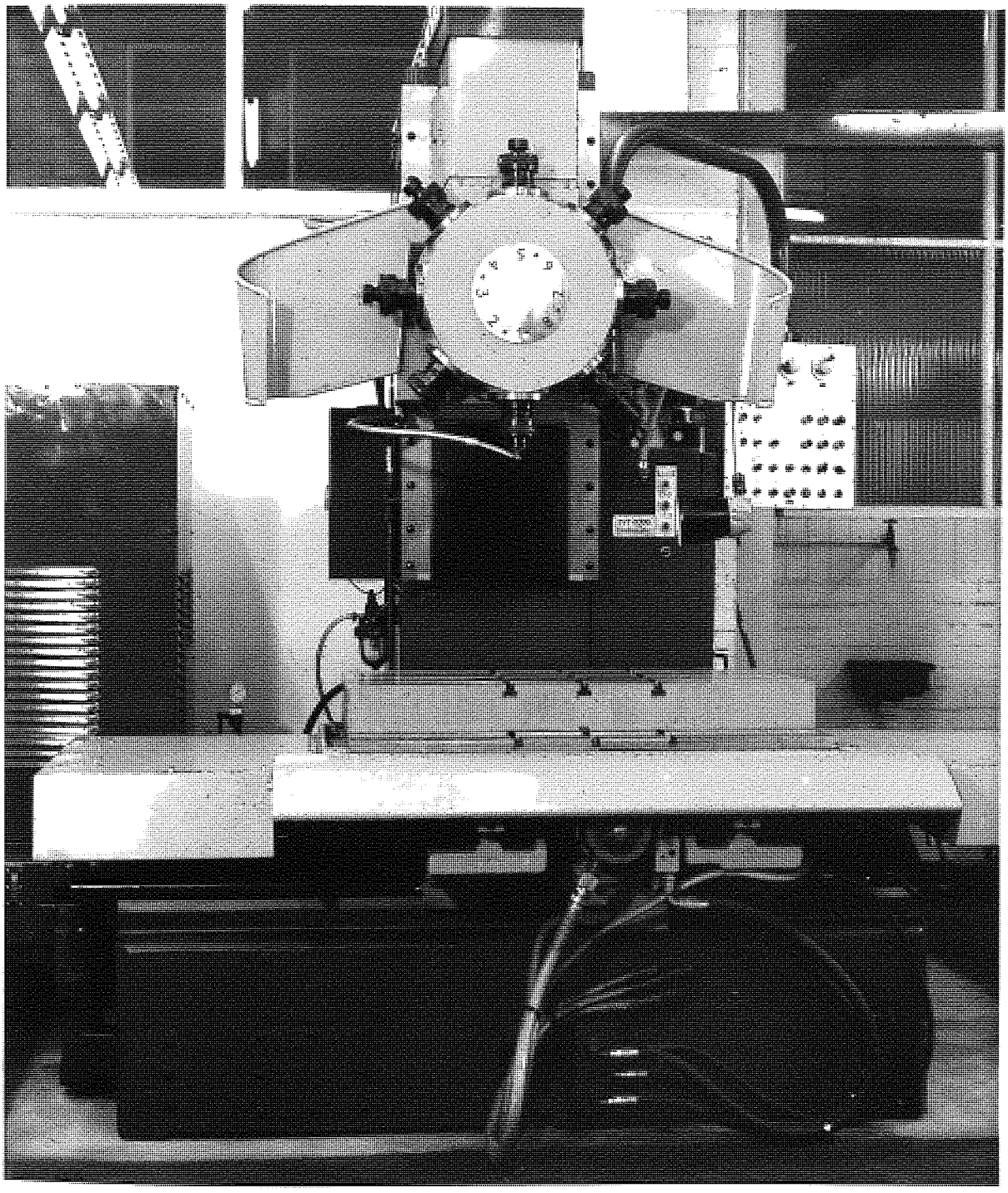


FIGURE 9.22 The blank block and the finished part

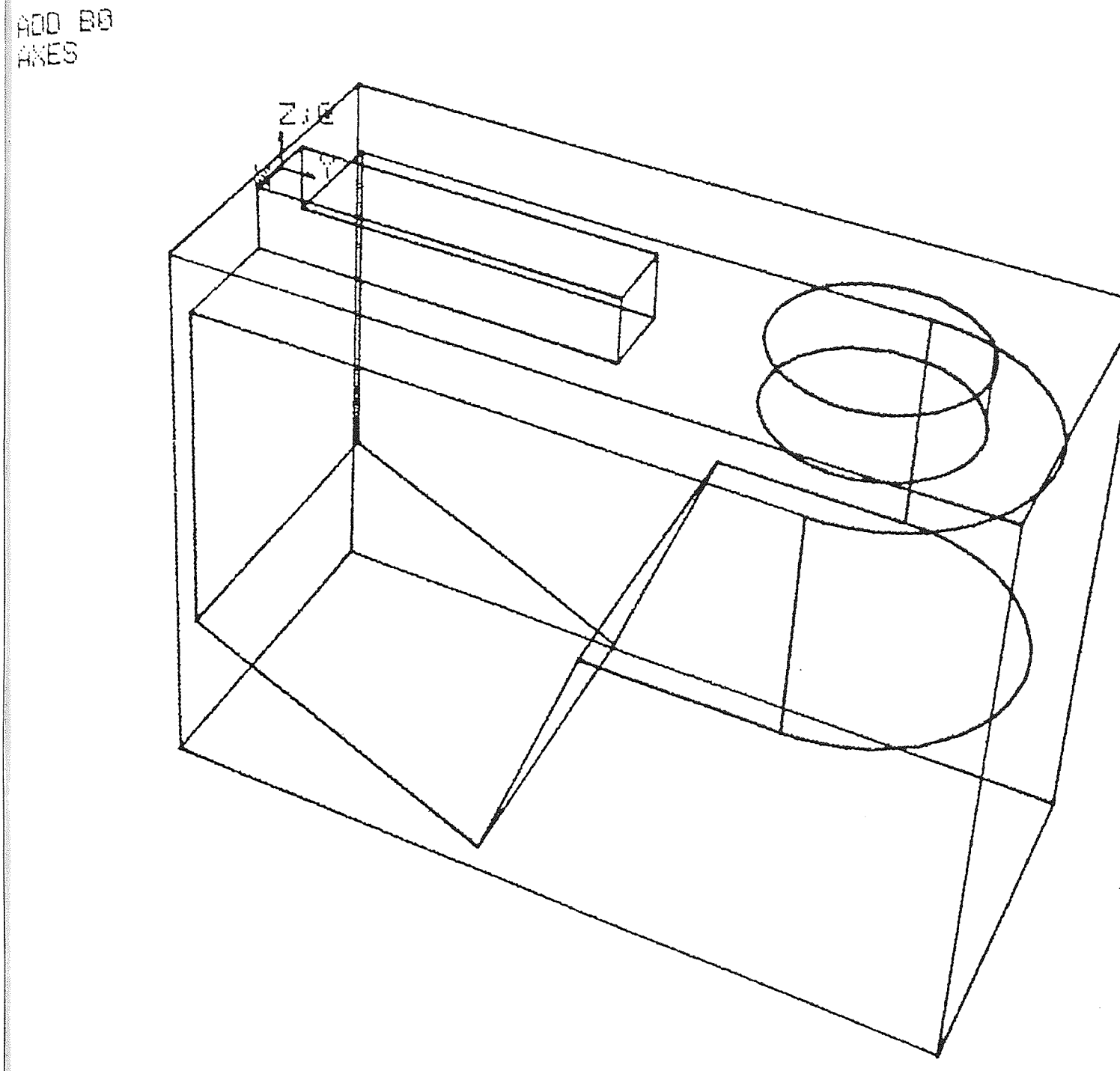


FIGURE 9 23 The outer profile

EL F53
*

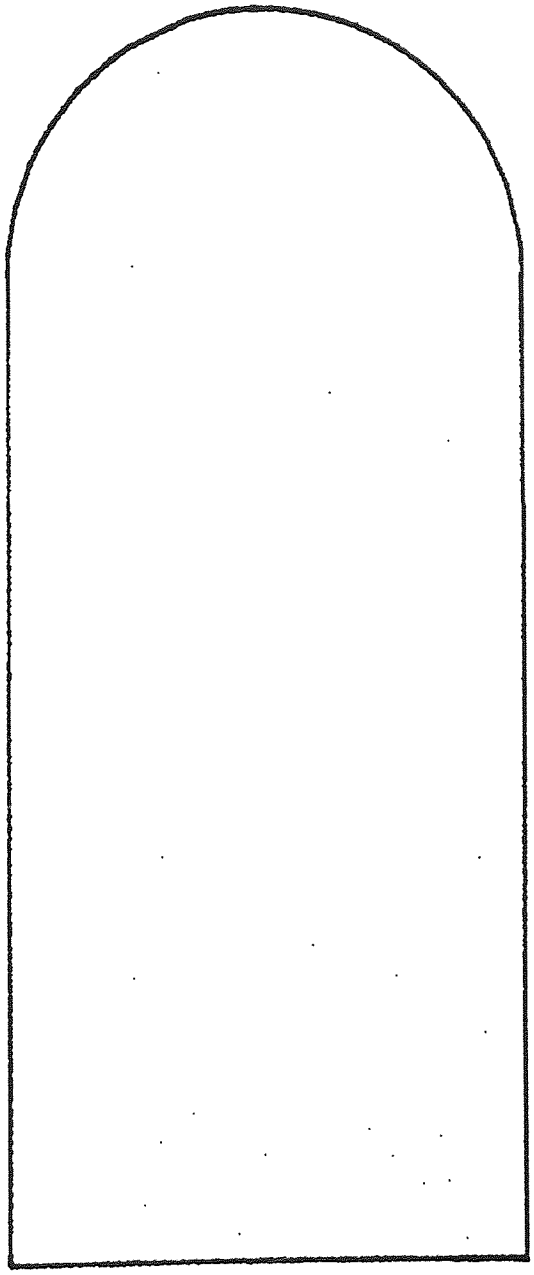


FIGURE 9 24 The inner profile

SEL F1
*

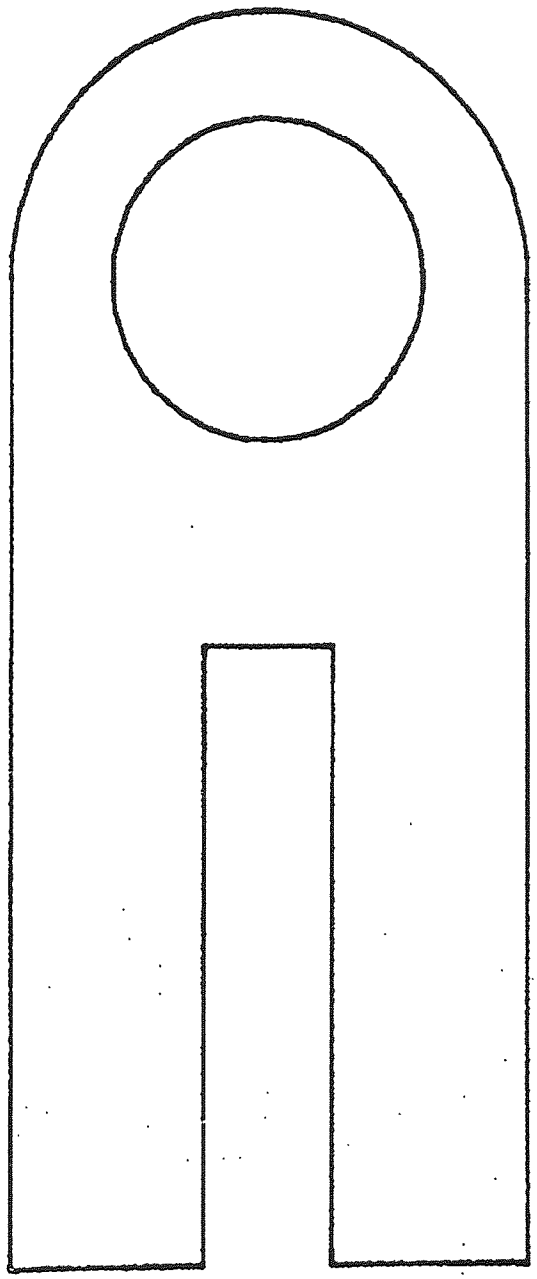


FIGURE 9.25 Procedure for generating parallel horizontal planes

```
SILHOUETTE AND VISIBILITY PASS: B1
DELETING VIEW DEPENDENT MATERIAL: B1
* CNC CUT FLAT 22
*    22.00000
** SELECT F6
** CNC PLANE CUR
VEC BEFORE = : 121.103,29.5419
VEC AFTER  = : 64.3496,-0.000021,-113.589
VEC BEFORE = : 121.103,54.68
VEC AFTER  = : 64.3496,-0.000011,-61.3249
          52.26376
          52
NL=
*
```

FIGURE 9 26 Side view of planes generated

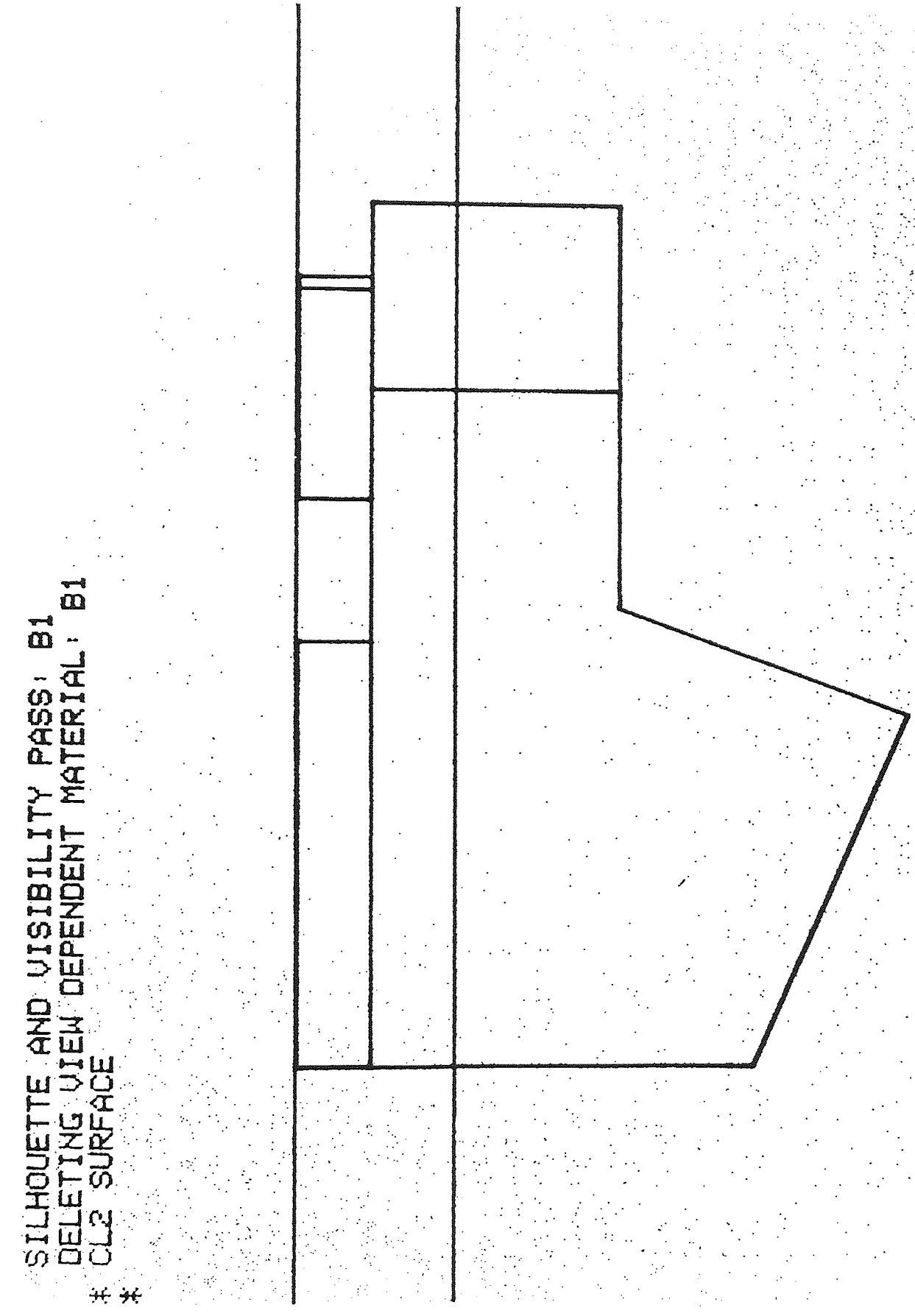


FIGURE 9.27: A perspective view of planes generated

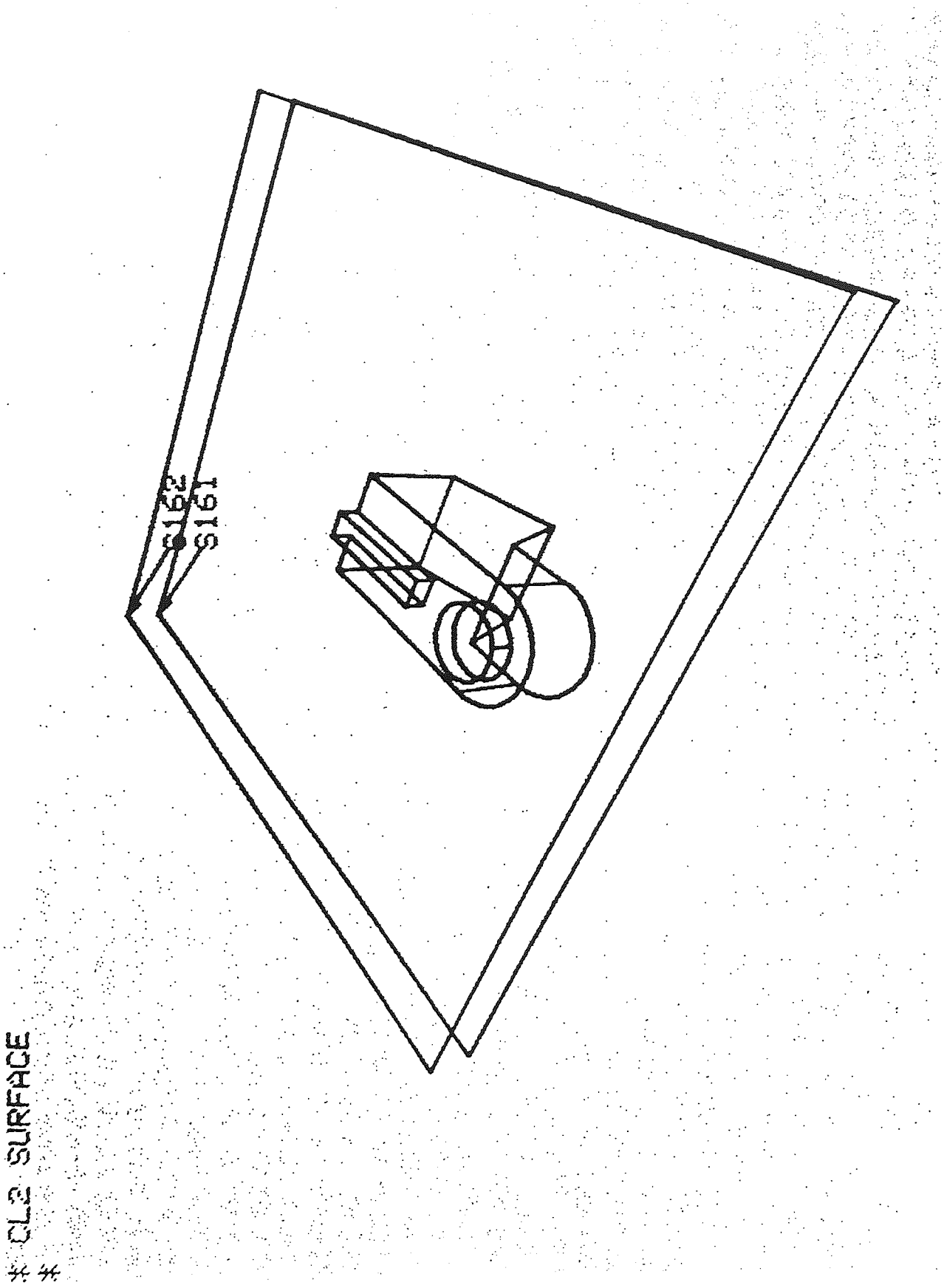


FIGURE 9.28 The outer profile

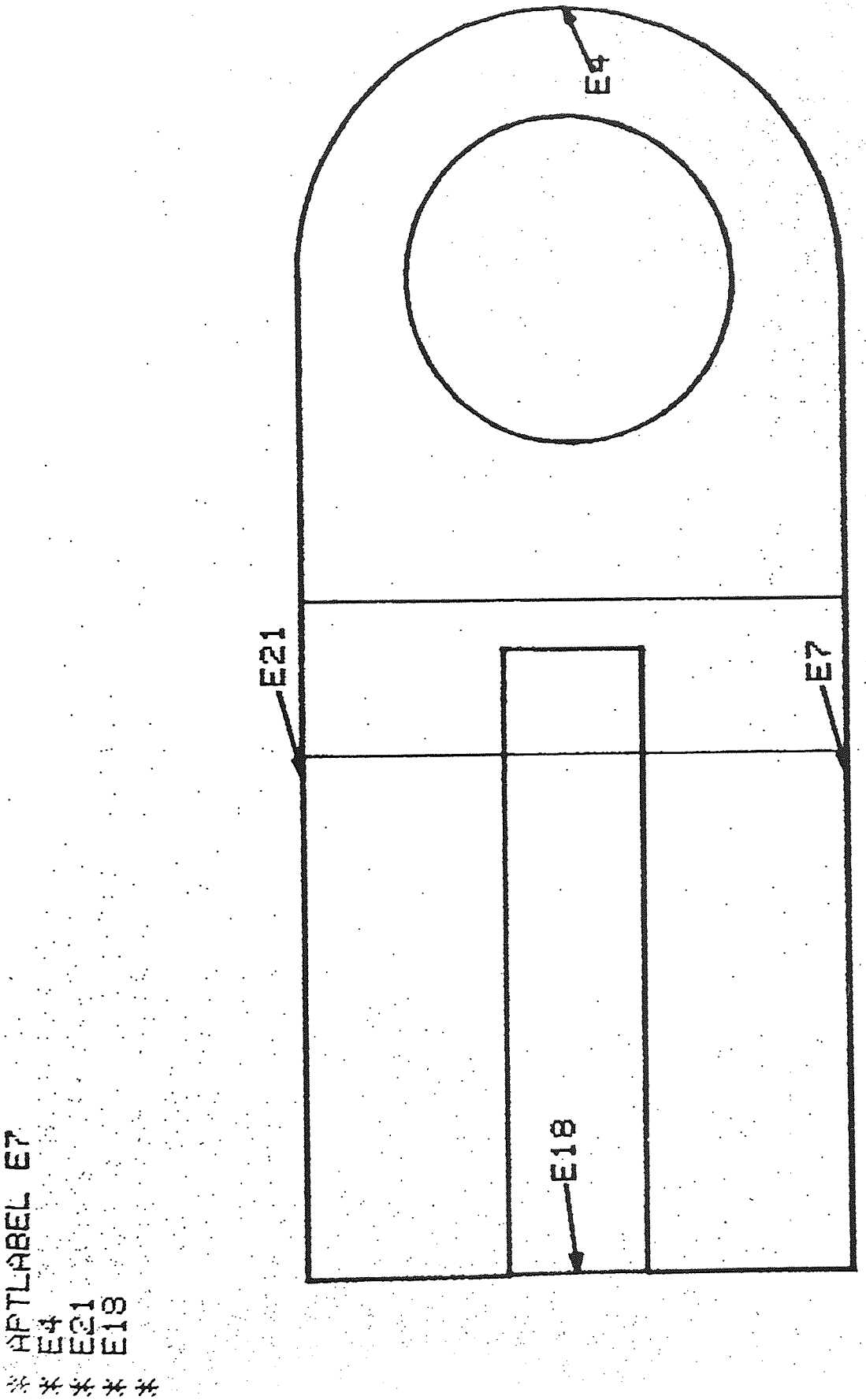


FIGURE 9 29 The cutter path for the outer profile for a cutter radius of 4.0 units

```
* PROFILE E7 E4 E21 E18 FLAT 4.0 ZAXIS -93
  PROF FACE : F8
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F9
  PROF INTRK : 1
  PROF INTRK : 1
  PROF CIRCLE : $277
  PROF FACE : F7
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F3
  PROF INTRK : 0
  PROF INTRK : 0
VULNERABLE WORKSPACE AT SLOT 429, POINTER 443
VULNERABLE WORKSPACE AT SLOT 443, POINTER 457
* PROFILE E7 E4 E21 E18 FLAT 4.0 ZAXIS -115
  PROF FACE : F8
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F9
  PROF INTRK : 1
  PROF INTRK : 1
  PROF CIRCLE : $277
  PROF FACE : F7
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F3
  PROF INTRK : 0
  PROF INTRK : 0
VULNERABLE WORKSPACE AT SLOT 429, POINTER 443
VULNERABLE WORKSPACE AT SLOT 443, POINTER 457
*
```

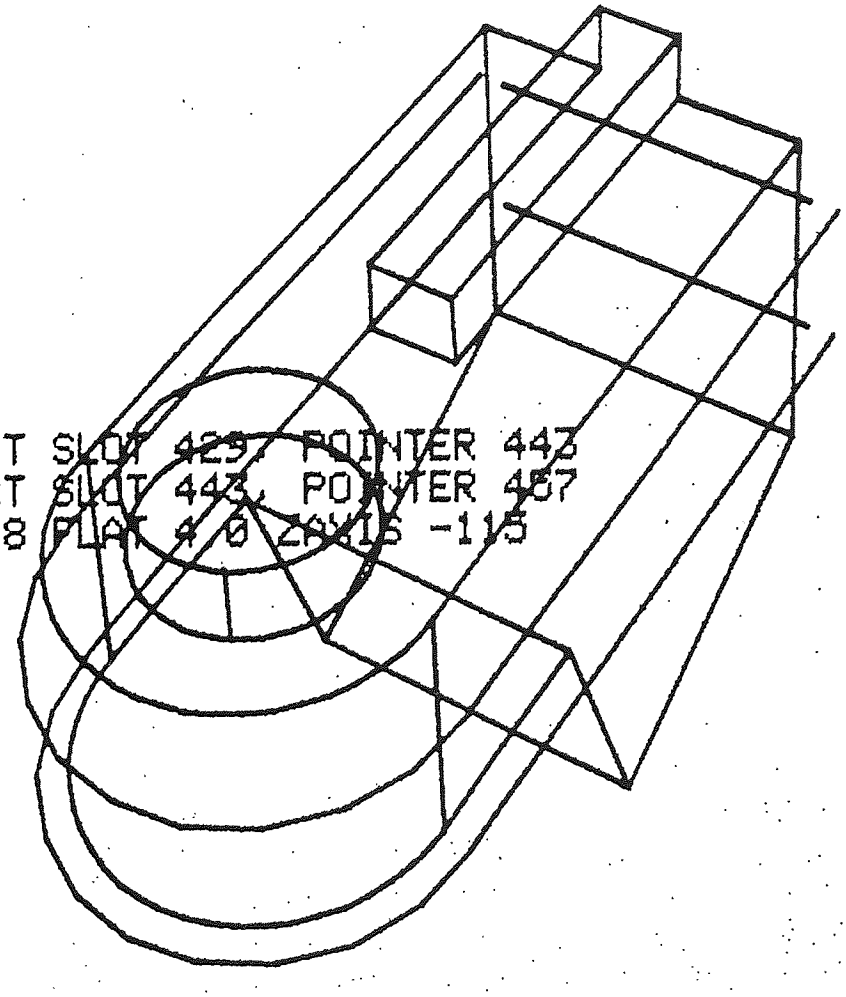


FIGURE 9.30 The cutter path for the outer profile for a cutter radius of 19.05 units

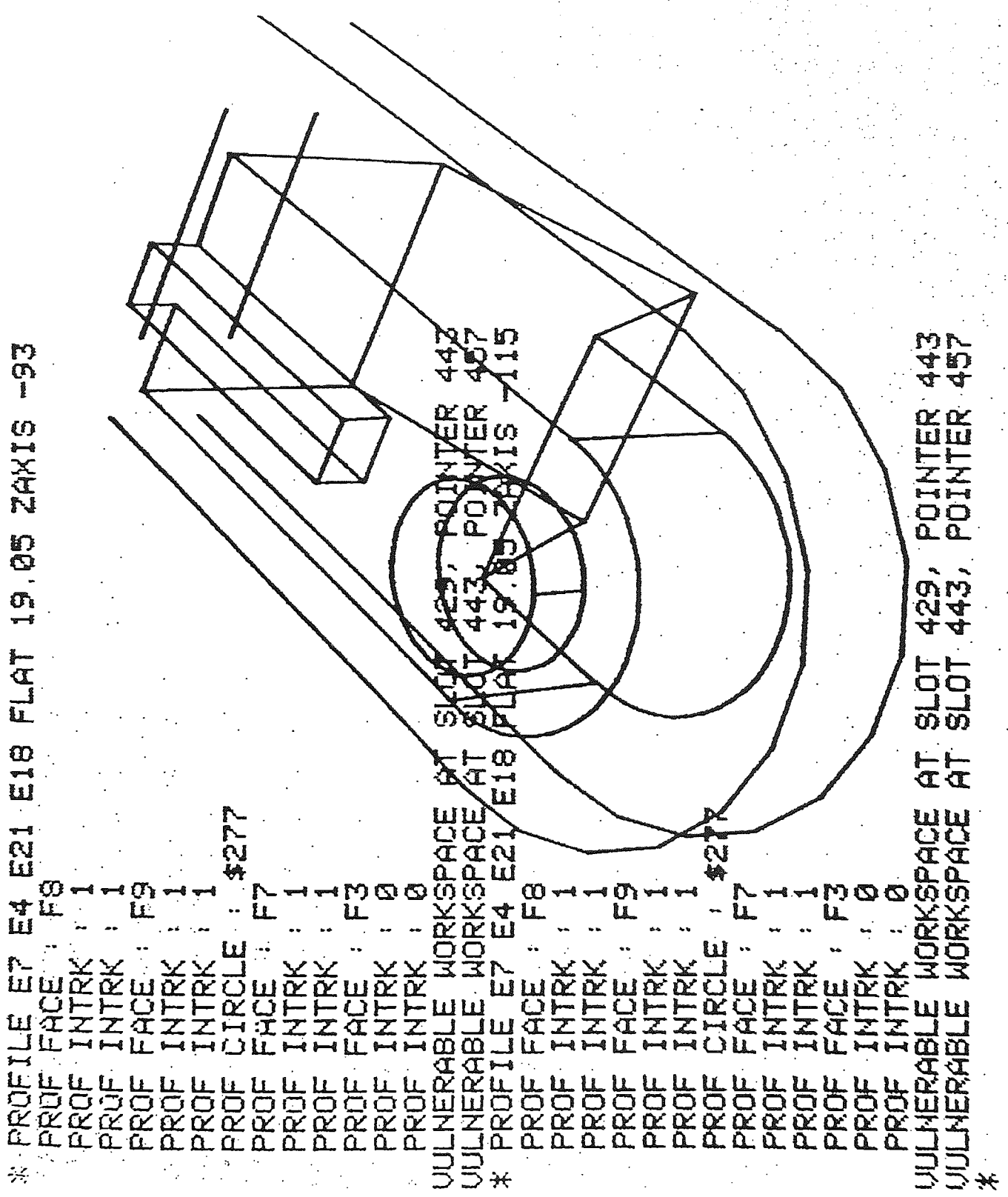


FIGURE 9.31 The points and edges of the outer profile

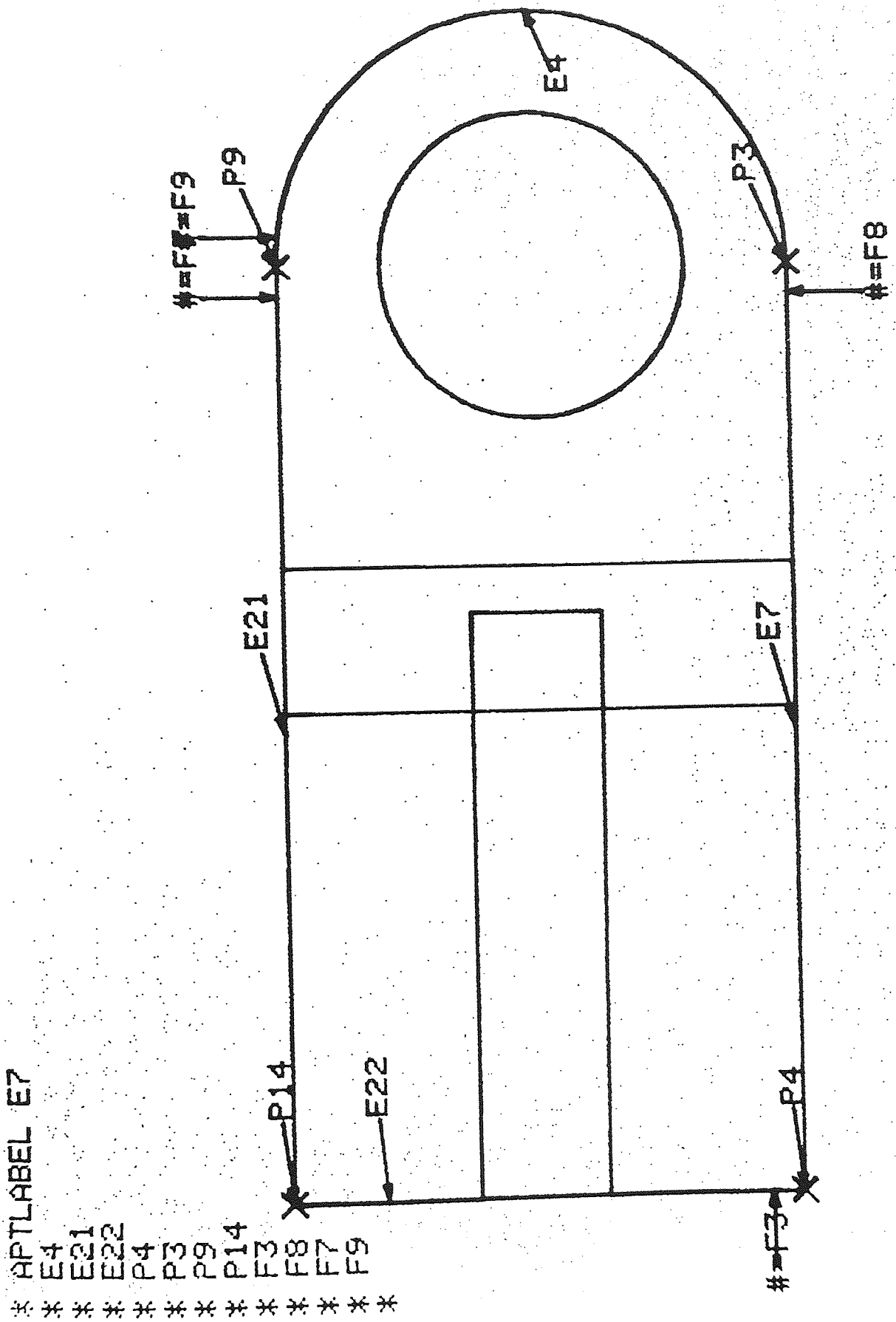


FIGURE 9 32 The procedure for manual generation of data structure required for the outer profile

```
CNC LOOP CLOSE OUT MANUAL
 INPUT SENSE, GLOBAL RAISE FLAG, NO. OF ITEMS =
1 1 4
 INPUT MIN. DISTANCE, MIN. RADIUS =
1.0,1.0
 INPUT POINTS   4
 (POINT-NAME) ?
+ P4 P3 P9 P14
 INPUT EDGES   4
 (EDGE-NAME) ?
+ E7 E4 E21 E22
 INPUT RAISE FLAG   4
1 1 1 1
 INPUT RAISE FACE   4
 (FACE-NAME) ?
+ F8 F9 F7 F3
 INPUT CONNECTING FLAG   4
1 2 3 1
 INPUT CONNECTING TYPE   4
3 2 2 3
 INPUT DOUBLE FLAG   4
1 1 1 1
 INPUT OMIT FLAG   4
0 0 0 0
 INPUT CONCAVE FLAG   4
0 0 0 0
 INPUT REVERSE RAISE FLAG   4
0 0 0 0
```

FIGURE 9.33 The profile data structure generated

```
  $17 $14 $204 $5 $311
  $10 $272 $204 $318 $311
  $76 $193 $204 $65 $311
  $171 $248 $204 $186 $311
   1    1    4    1    0    0    0    0    0    0
   0    0    1    0    1    3    1    0    0    0
   0    0    1    0    2    2    1    0    0    0
   0    0    1    0    3    2    1    0    0    0
   0    0    1    0    1    3    1    0    0    0
VULNERABLE WORKSPACE AT SLOT 412, POINTER 427
*
```

FIGURE 9.34: The APT motion statments generated for the profile

```
CNC APTLOOP CURRENT
 PSURF =     1
 INPUT STARTING POINT
-10,-35,-10
  $$  APT  LOOP :    1
  FROM /     -10.0,-35.0,-10.0
 LOOP =                1              1
 LOOP =         -14664         -14669
  GO / TO,    PS          ,TO ,    P10          ,TO ,    P3
    2    1    1    1
 IT =       1
  INDIRP /    P3
  TLRGT , GOFWD /    E7          , TANTO ,    E4
 IT =       3
 LOOP =                0
  INDIRP /     116.0,0.000282,-80.0
  TLRGT , GOFWD /    E4          , TANTO ,    E21
 IT =       1
  INDIRP /    P14
  TLRGT , GOFWD /    E21         , PAST ,     E22
 IT =       1
  INDIRP /    P4
  TLRGT , GOFWD /    E22         , PAST ,     E7
  GODLTA /    0.0
  GO TO /     -10.0,-35.0,-10.0
*
```

FIGURE 9.35 The generation of header, tolerance, cutter definition and end statements.

```
CNC HEADER
   PARTNO    B0
   CLPRNT
   MACHIN / PLOTPP, 1
   UNITS / INCHES
   INTOL /    0.05
   OUTTOL /    0.05
* CNC MACHINE
   $$ MACHINING START
   LOADTL / 1
   PPLOT / ON, XYZ, 1
   CUTTER /    8.0
* CNC END
   END
   FINI
*
```

FIGURE 9 36 The cutter path verification (plane view) of the outer profile

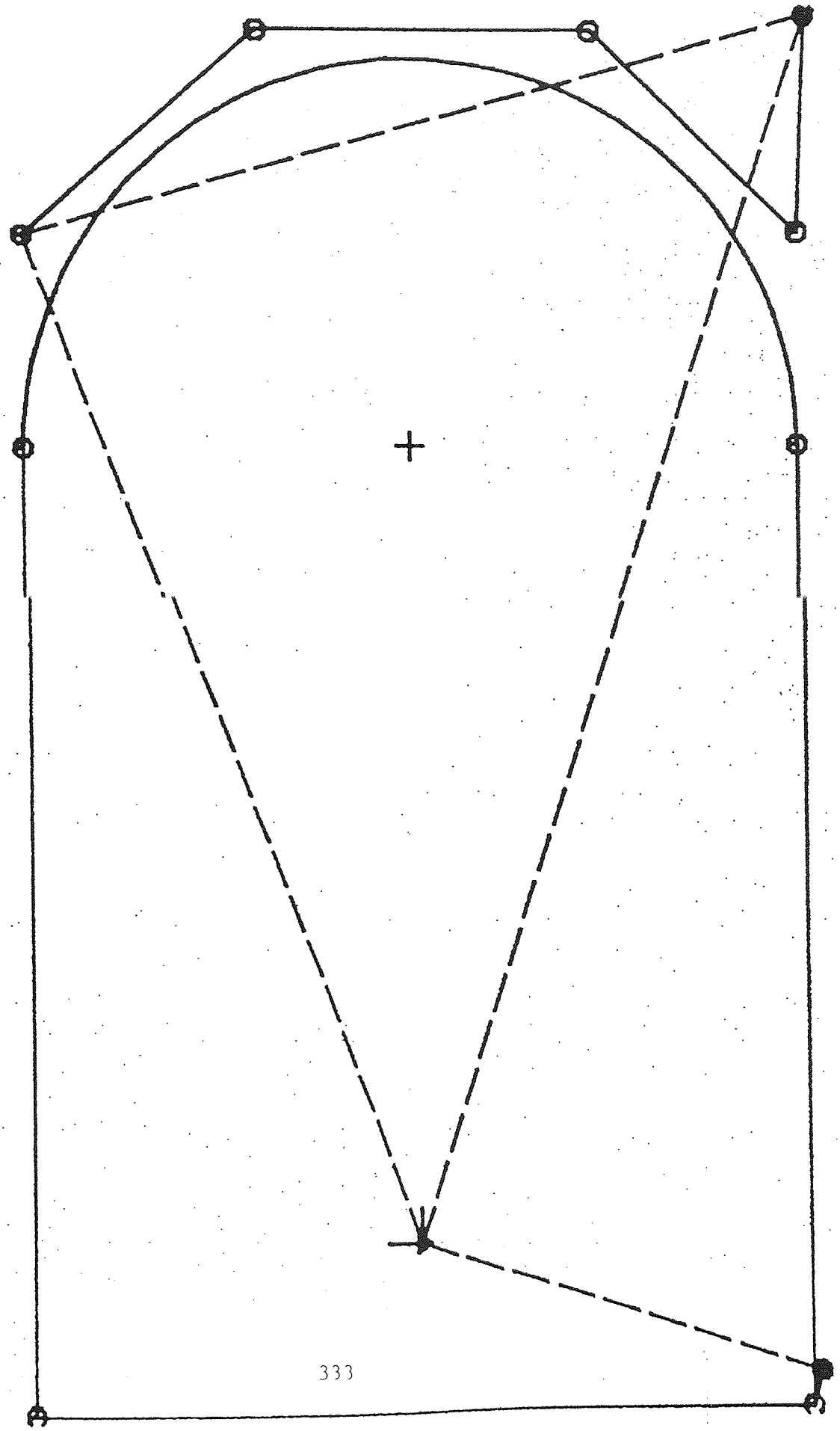


FIGURE 9.37 The cutter path verification (perspective view) of the outer profile.

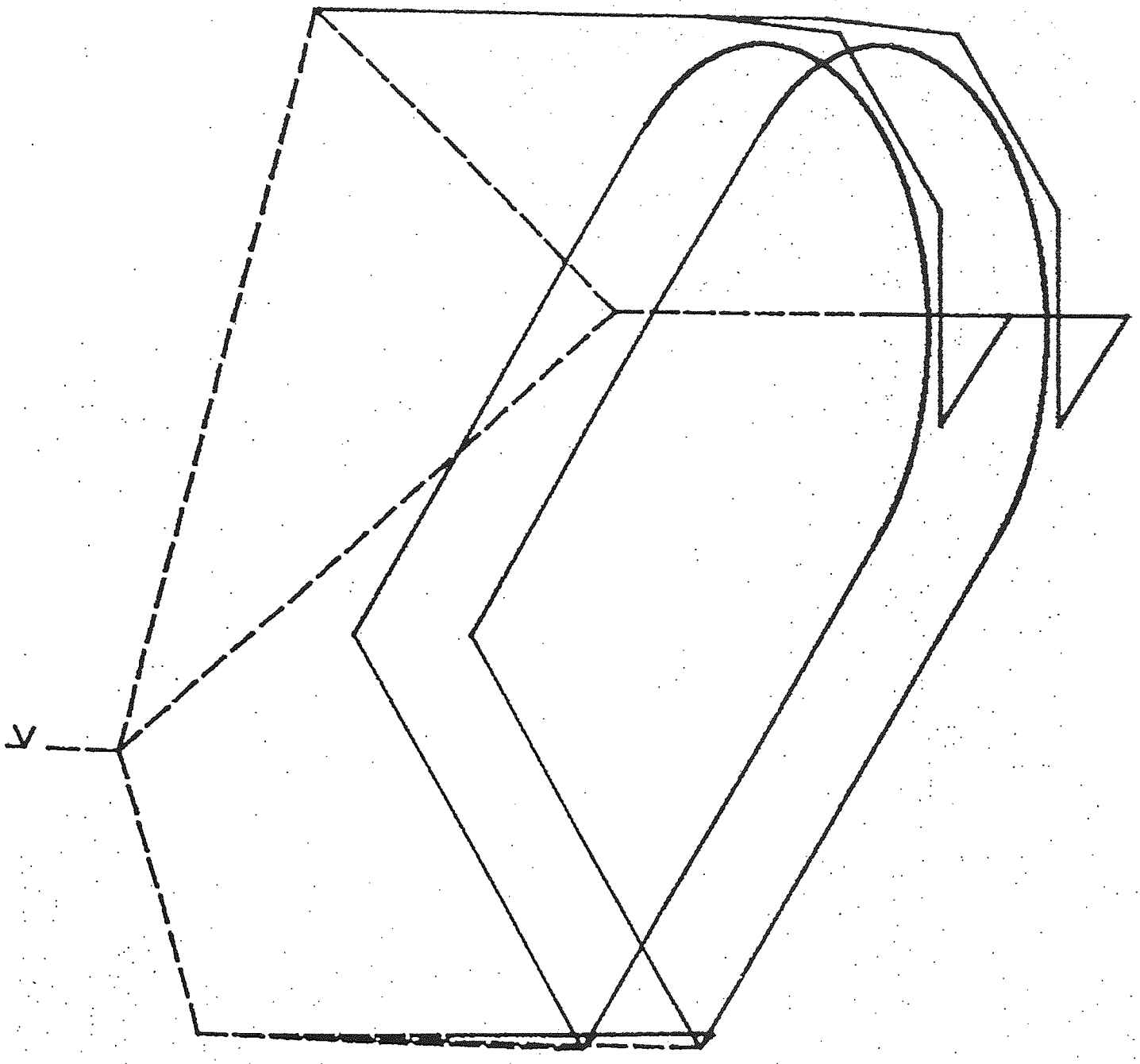


FIGURE 9.38: The points and edges of the inner profile

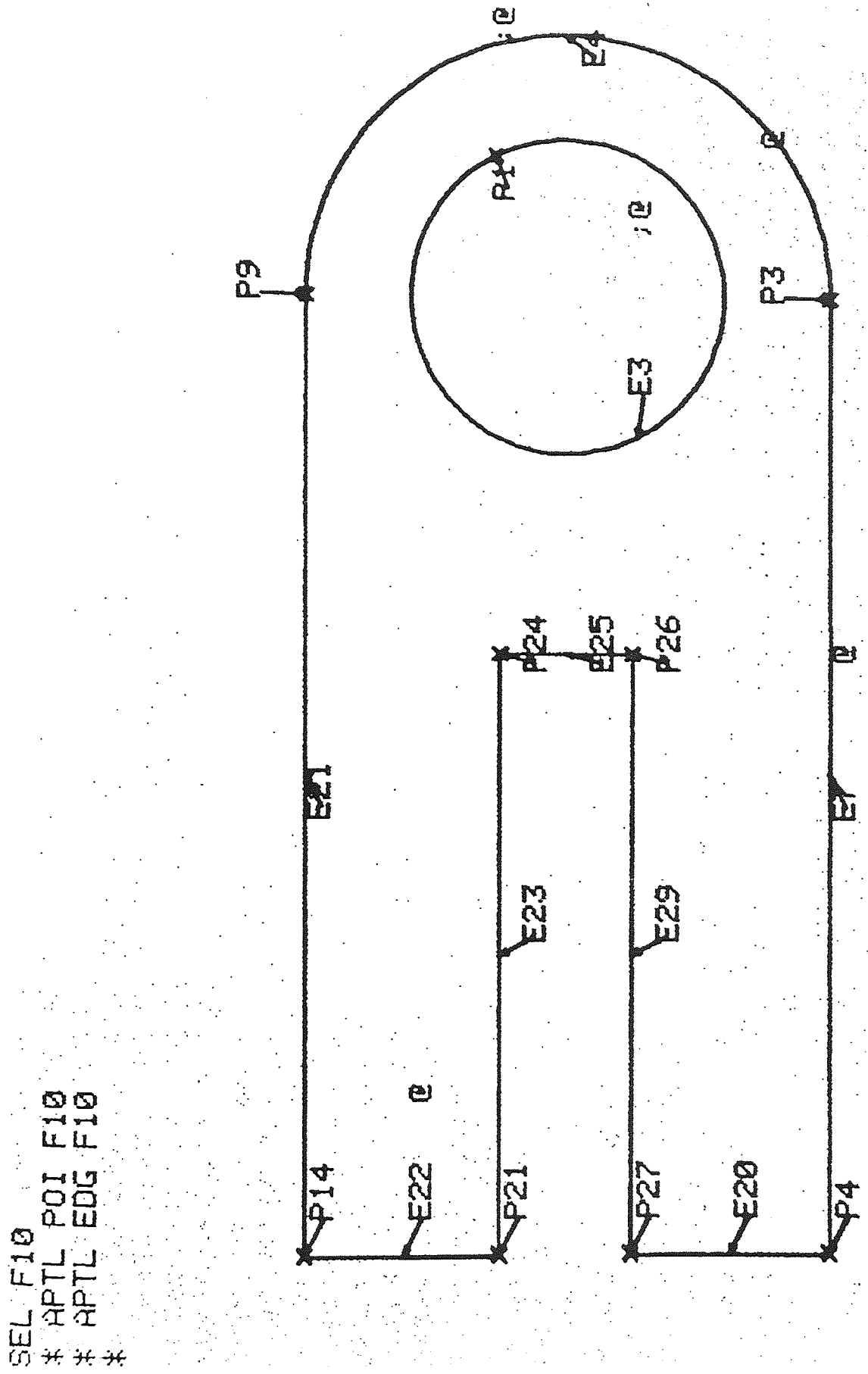


FIGURE 9.39 The cutter path generated for the inner profile

```
* APTLABEL E7
* SELECT F10
* CNC CUT FLAT 6.35
              6.35000
* PROFILE E7 LOOP FLAT 6.35 CUR
  PROF FACE : F8
  PROFBY EDGE : : $21
*
```

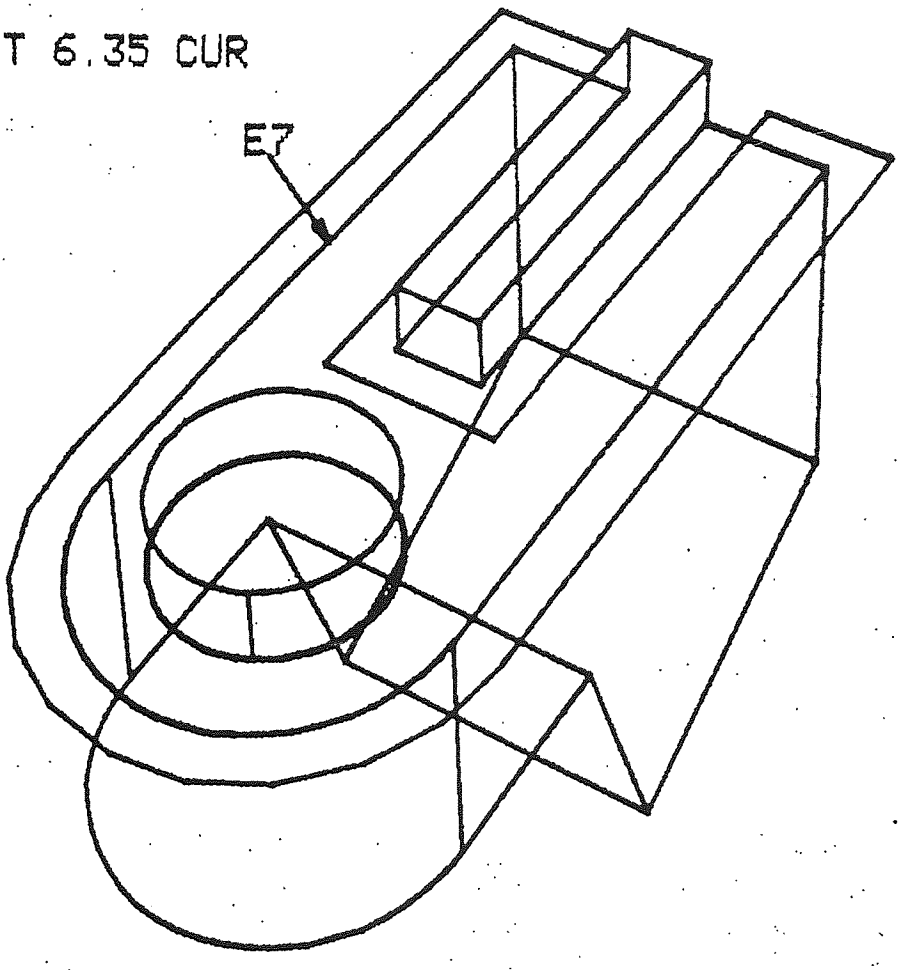


FIGURE 9 40 Automatic estimation of cutter size

```
@I
* @I
 CNC DISTANCE CURSOR
  VEC BEFORE = : 181.356,175.311
  VEC AFTER = : -0.34981,76.0211
  VEC BEFORE = : 181.712,137.617
  VEC AFTER = : -0.174853,57.4751
  DISTANCE =: 18.5468
  MAX. RADIUS =: 9.27341
@I
*
```

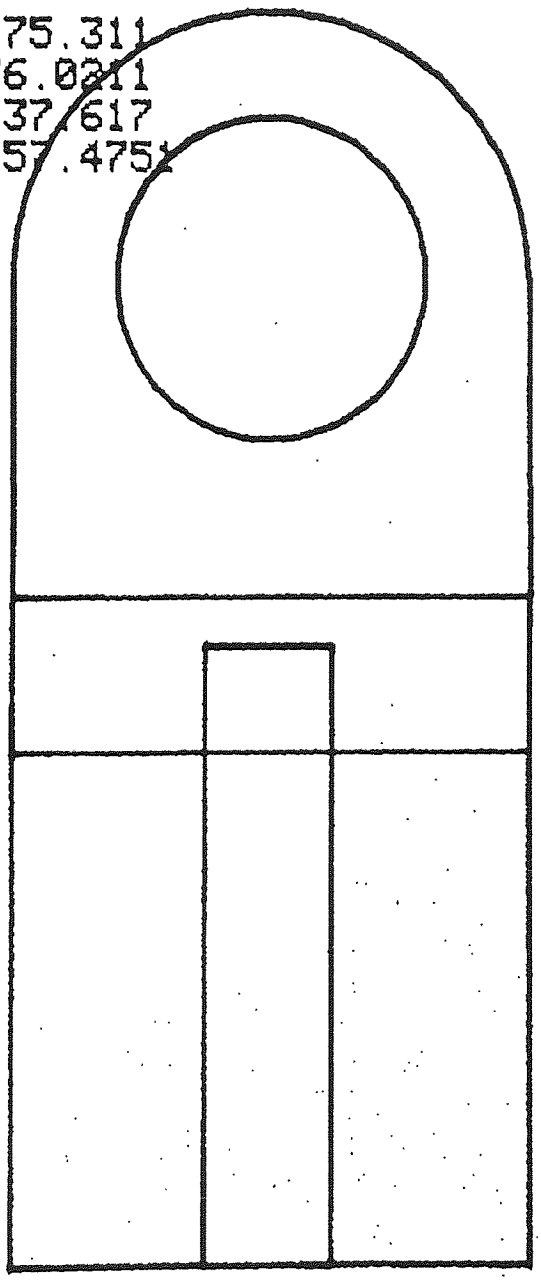


FIGURE 9.41 The procedure for the automatic generation of the data structure required for the inner profile

```
* APTL E7
* P4
* P3
* CNC CUT FLAT 6.35
                6.35000
* CNC LOOP CLOSE OUT AUTOMATIC
 ZZ =     1.000
 INPUT GLOBAL RAISE FLAG
0
 INPUT DIRECTION (2 POINTS)
AUTOMATIC ?
 (POINT-NAME) ?
+ P4 P3
```

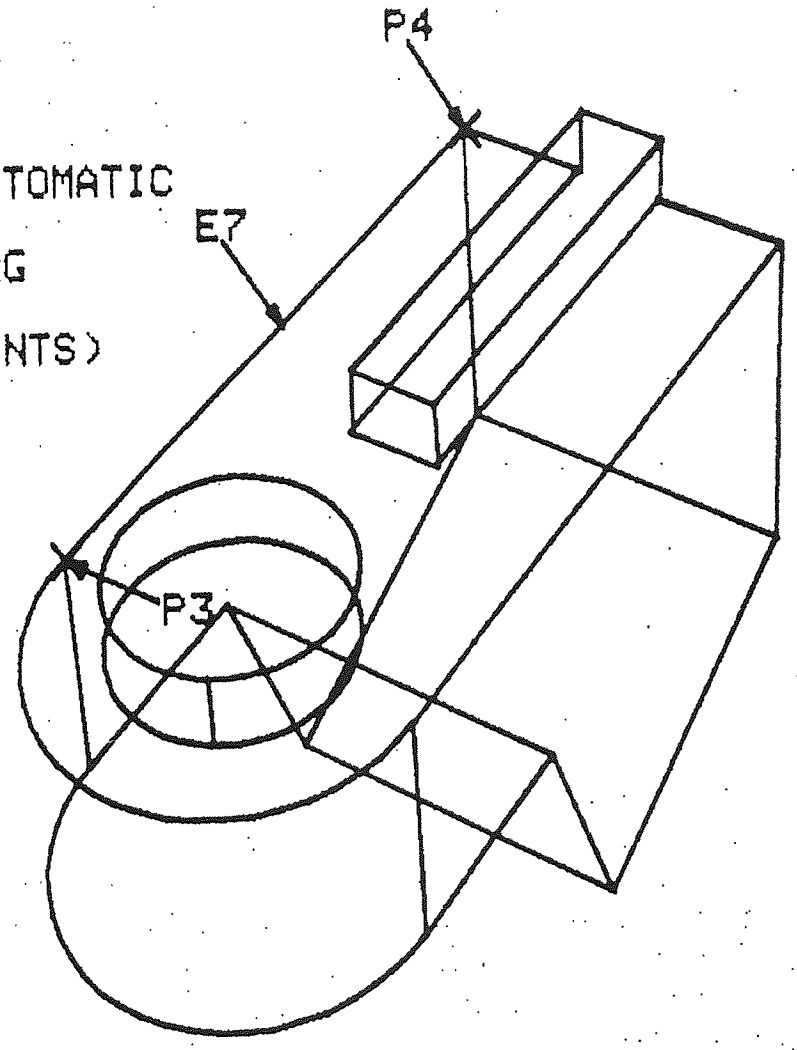


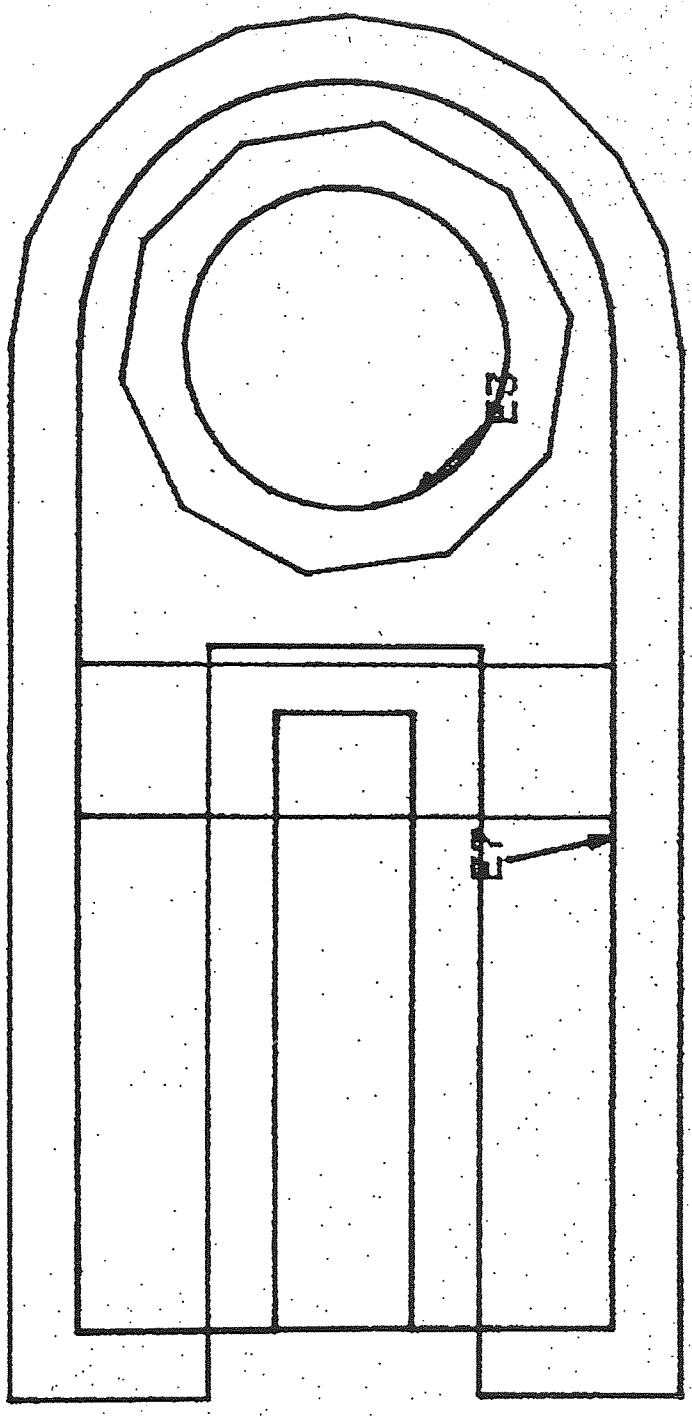
FIGURE 9.42 The data structure generated automatically for the inner profile

```
  0     0     0     0
 $17 $14 $204 $5 $311
 $10 $272 $204 $318 $311
 $76 $193 $204 $65 $311
 $171 $248 $204 $186 $311
 $341 $334 $204 $520 $311
 $232 $189 $204 $201 $311
 $297 $339 $204 $309 $311
 $91 $180 $204 $186 $311
  1   0   8   1   0   0   0   0   0   0
  0   0   0   0   1   3   1   0   0   0
  0   0   0   0   2   2   1   0   0   0
  0   0   0   0   3   2   1   0   0   0
  0   0   0   0   1   3   1   0   0   0
  0   0   1   0   1   3   1   0   0   0
  0   0   1   0   1   4   1   0   0   1
  0   0   1   0   1   4   1   0   0   1
  0   0   0   0   1   3   1   0   0   0
VULNERABLE WORKSPACE AT SLOT 412, POINTER 427
VULNERABLE WORKSPACE AT SLOT 427, POINTER 441
VULNERABLE WORKSPACE AT SLOT 441, POINTER 459
VULNERABLE WORKSPACE AT SLOT 459, POINTER 467
VULNERABLE WORKSPACE AT SLOT 467, POINTER 481
VULNERABLE WORKSPACE AT SLOT 481, POINTER 495
VULNERABLE WORKSPACE AT SLOT 495, POINTER 509
VULNERABLE WORKSPACE AT SLOT 509, POINTER 523
VULNERABLE WORKSPACE AT SLOT 523, POINTER 541
VULNERABLE WORKSPACE AT SLOT 541, POINTER 555
*
```

FIGURE 9 43 The automatic generation of APT motion statements for the inner profile

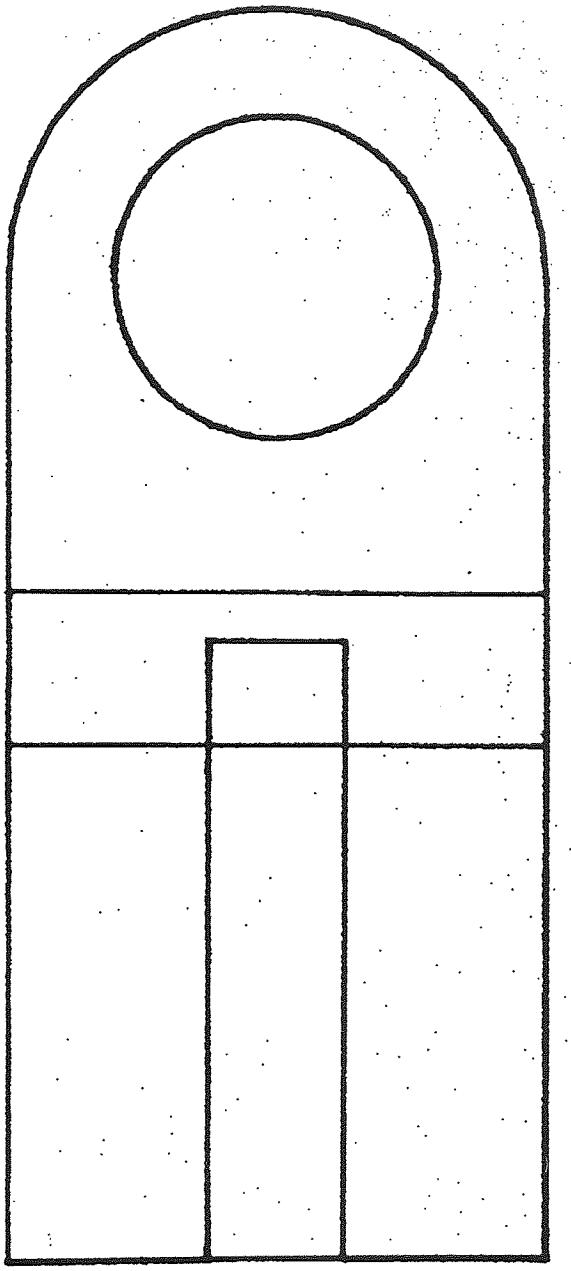
```
CNC APTL CUR                                GODLTA /    0.0
  PSURF = 1                                 GO TO /     -10.0,-35.0,-10.0
    INPUT STARTING POINT
-10,-35,-10                      *
     $$  APT  LOOP : 1
    FROM /    -10.0,-35.0,-10.0
LOOP =          0          0
LOOP =       -14921     -14926
   GO / ON,   E7    ,TO ,   F10          ,ON ,    E20
     2   1    2   1
IT =   1
  INDIRP /    P3          E7        , TANTO ,    E4
  TLON, GOFWD /
IT =   3
LOOP =         0
  INDIRP /    116.0,0.000282,-80.0
  TLON, GOFWD /    E4          , TANTO ,    E21
IT =   1
  INDIRP /    P14          , ON ,    E22
  TLON, GOFWD /    E21
IT =   1
  INDIRP /    P21          , TO ,    E23
  TLON, GOFWD /    E22
IT =   1
  INDIRP /    P24          , PAST ,    E25
  TLLFT, GOFWD /    E23
IT =   1
  INDIRP /    P26          , PAST ,    E29
  TLLFT, GOFWD /    E25
IT =   1
  INDIRP /    P27          , ON ,    E20
  TLLFT, GOFWD /    E29
IT =   1
  INDIRP /    P4          , ON ,    E7
  TLON, GOFWD /    E20
```

FIGURE 9.44 The cutter path generated for the inner profile



```
* APTL E7
  E3
* * * CNC CUT FLAT 6.35
          6.35000
  *  PROFILE E7 LOOP FLAT 6.35 CUR
       PROF FACE : F8
       PROFBY EDGE : $21
  *  PROFILE E3 FLAT 6.35 CUR
       PROF FACE : F1
  *  *
```

FIGURE 9.45 Procedure for area clearance



```
* CNC CUT F 10
* CNC CLEAR CURSOR 10.00000
```

FIGURE 9.46a: The cutter path generated for the area clearance

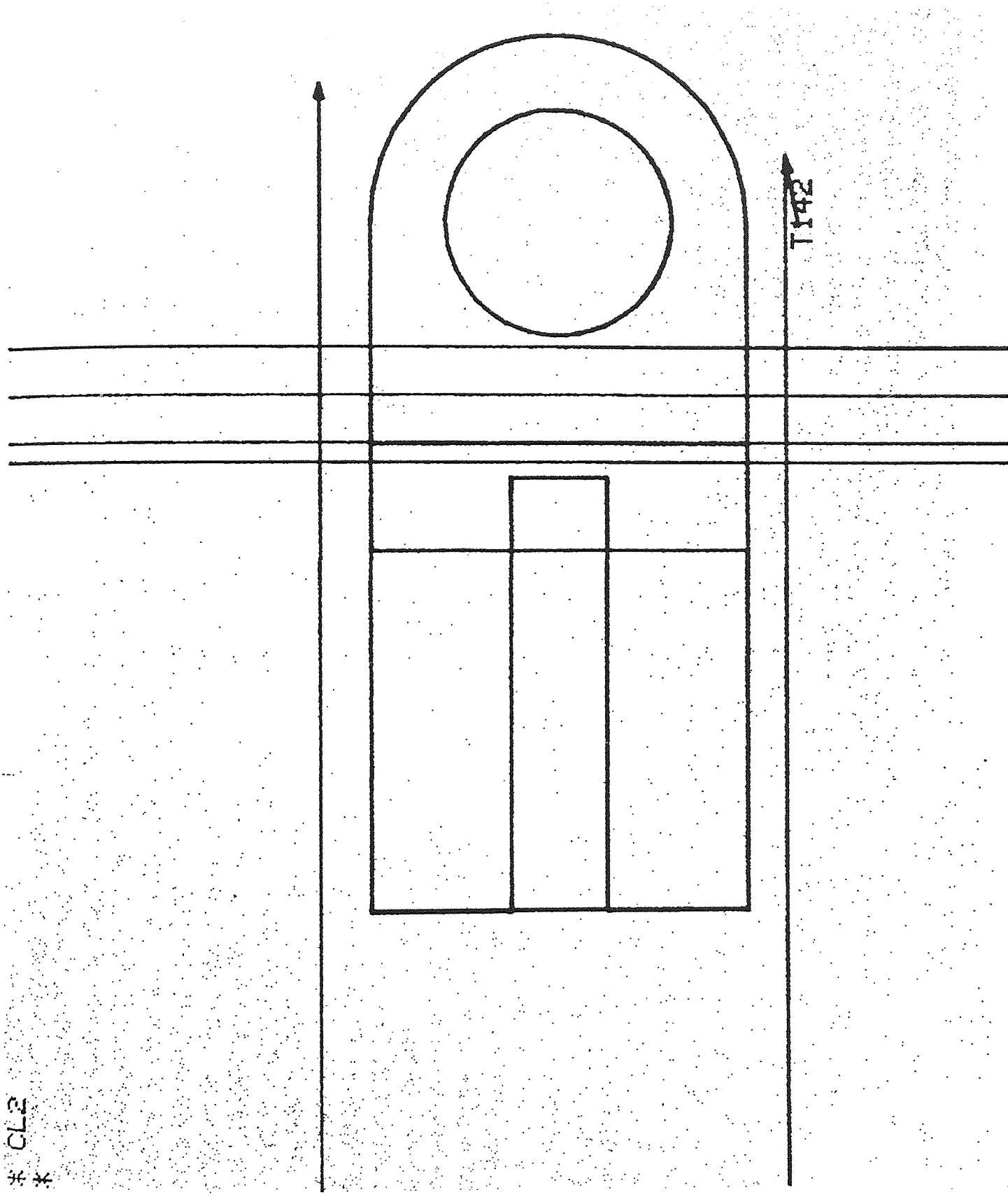


FIGURE 9 46b The complete cutter path for the top face F1

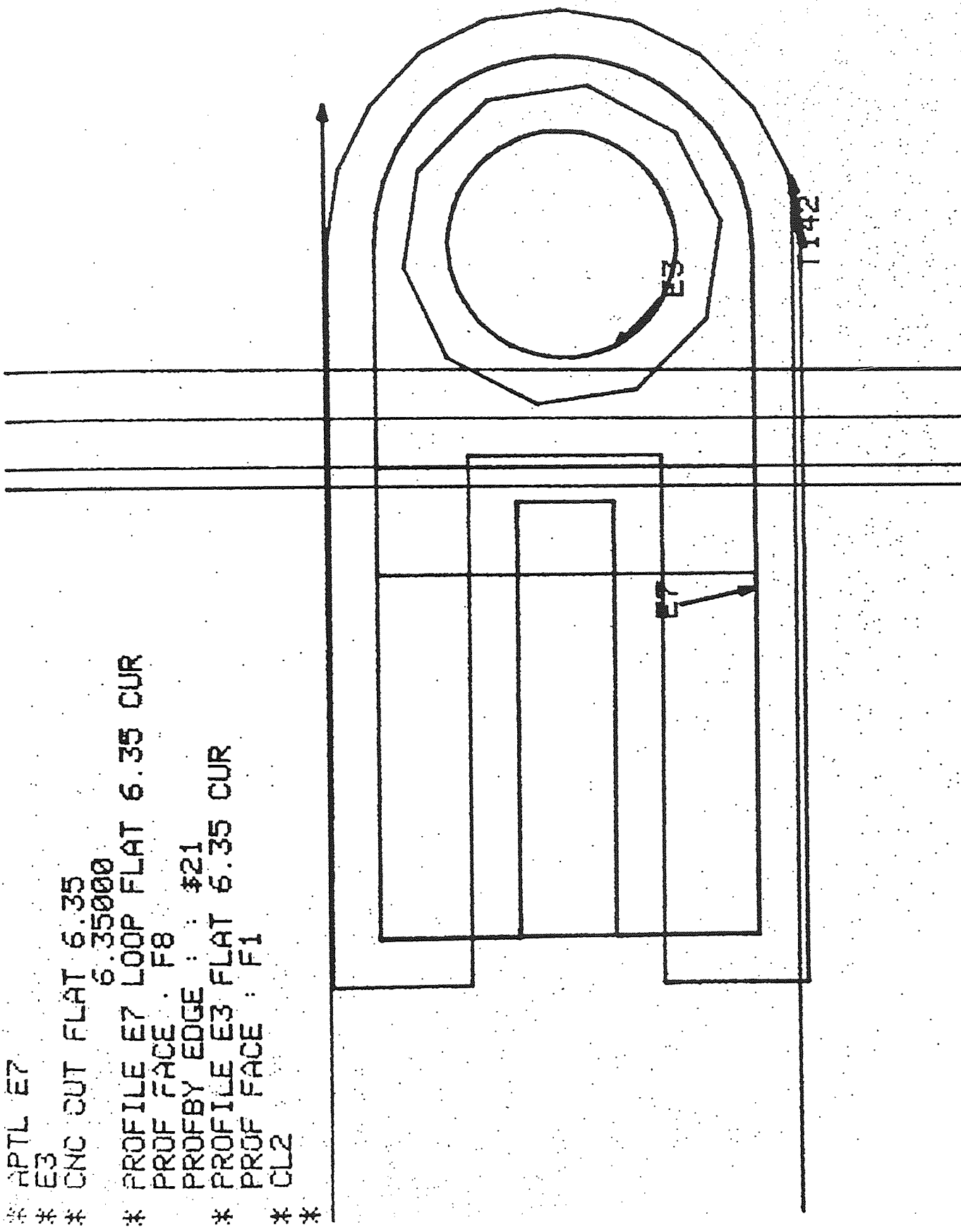


FIGURE 9.47 The APT motion statements generated for the area clearance

```
$$ AREA CLEARANCE    F10
FROM /     66.0,1.67,-5.0
T161       =LINE/   72.9458,26.9322,-80.0   $
             ,      72.9458,-27.6427,-80.0
T162       =LINE/   72.9458,-27.6427,-80.0   $
             ,      60.2406,-27.6427,-80.0
T163       =LINE/   60.2406,-27.6427,-80.0   $
             ,      60.2406,26.9322,-80.0
T164       =LINE/   60.2406,26.9322,-80.0   $
             ,      72.9458,26.9322,-80.0
GO / TO ,   T164        , TO ,   F10        , TO ,   T161
TLRGT, GORGT /   T161       , TO ,   T162
$$  LAST CUT OF AREA
IC =    0
TLRGT, GORGT /   T162       , TO ,   T163
TLRGT, GORGT /   T163       , TO ,   T164
GO TO /    66.0,1.67,-5.0
*
```

FIGURE 9.48 The cutter path verification (plane view) for the whole inner profile

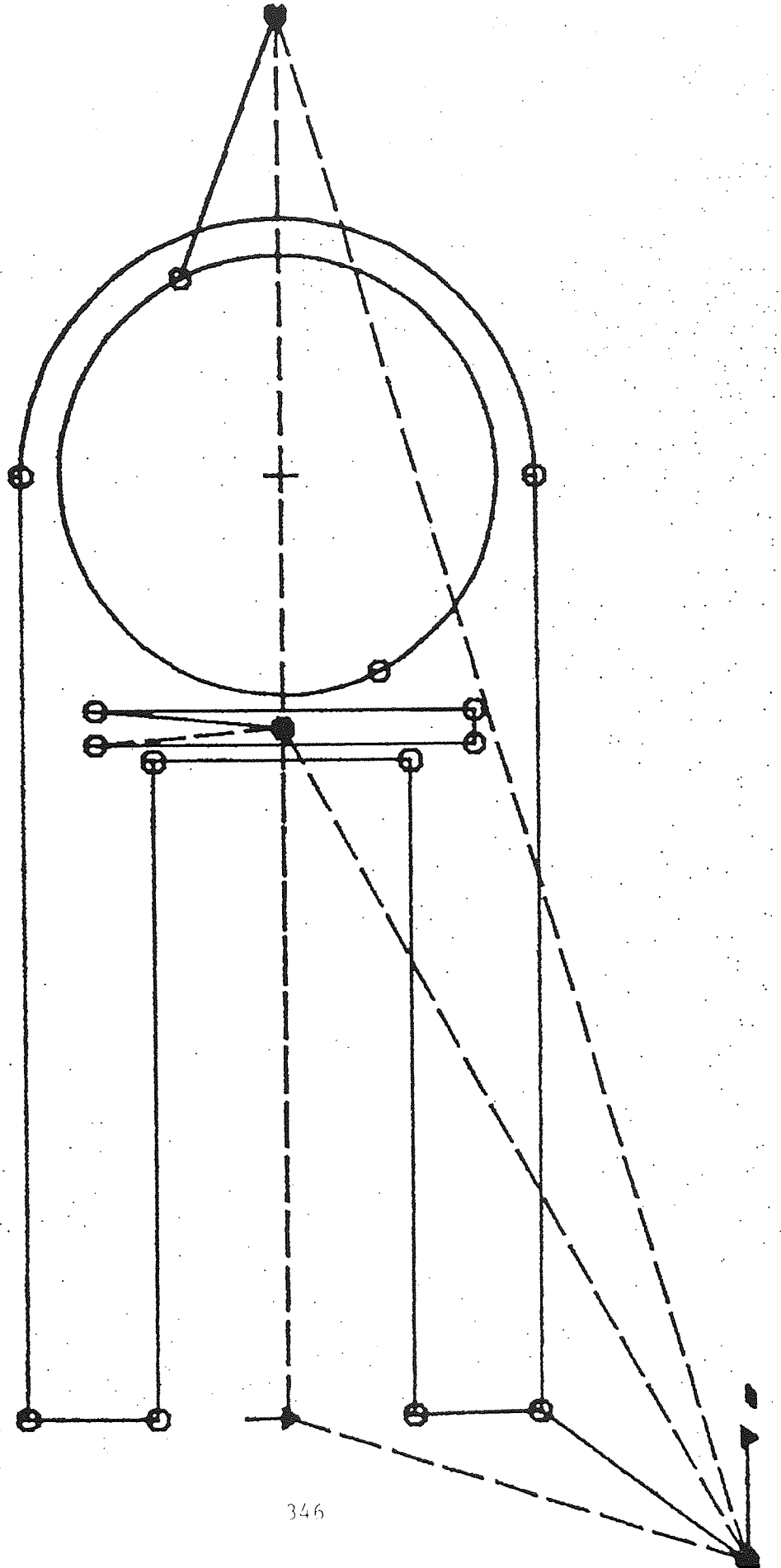


FIGURE 9.49: The cutter path verification (perspective view)
for the whole inner profile

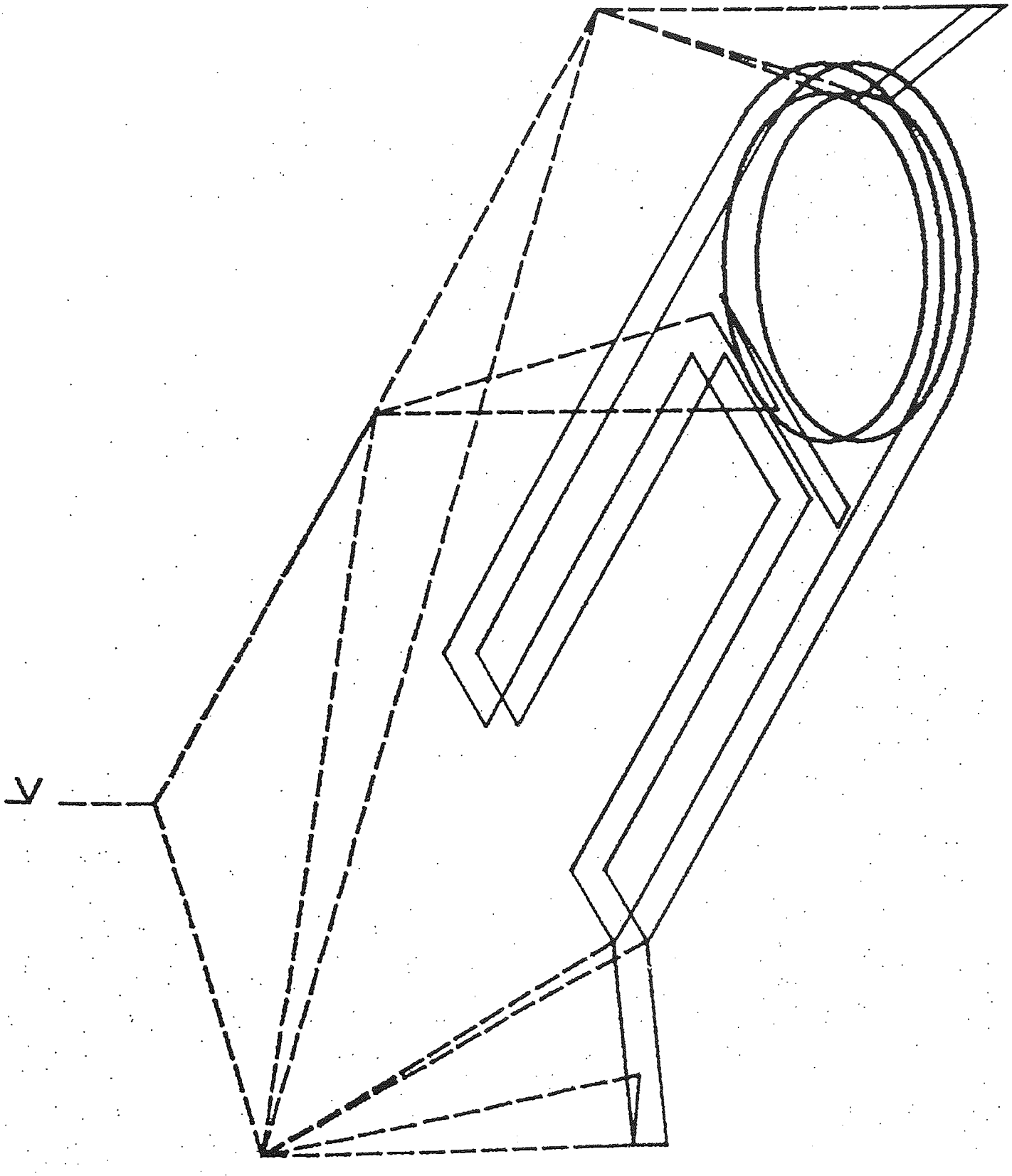


FIGURE 9 50· The workpiece is rotated 180 degree.

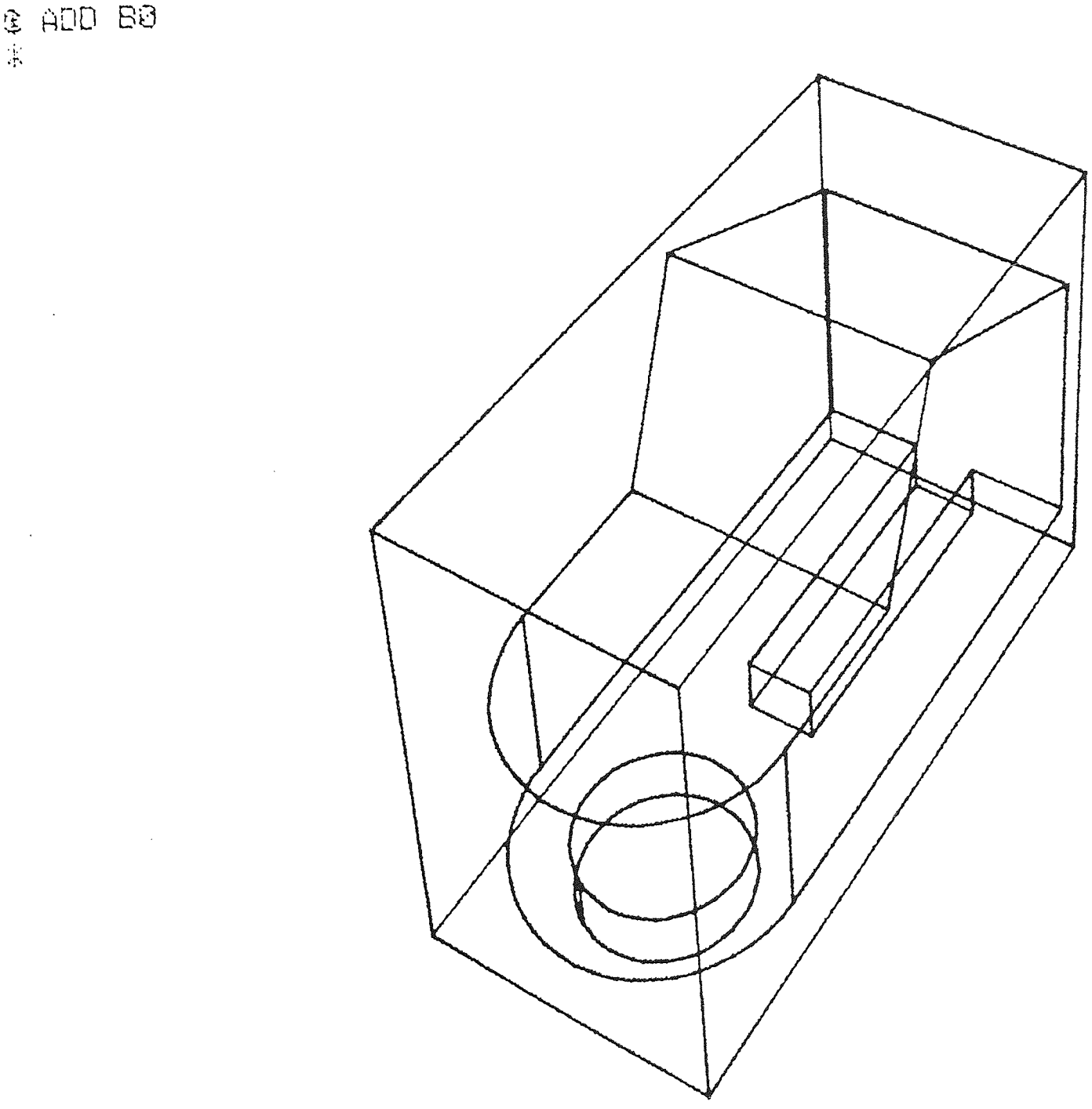


FIGURE 9.51: Procedure for generating parallel horizontal planes

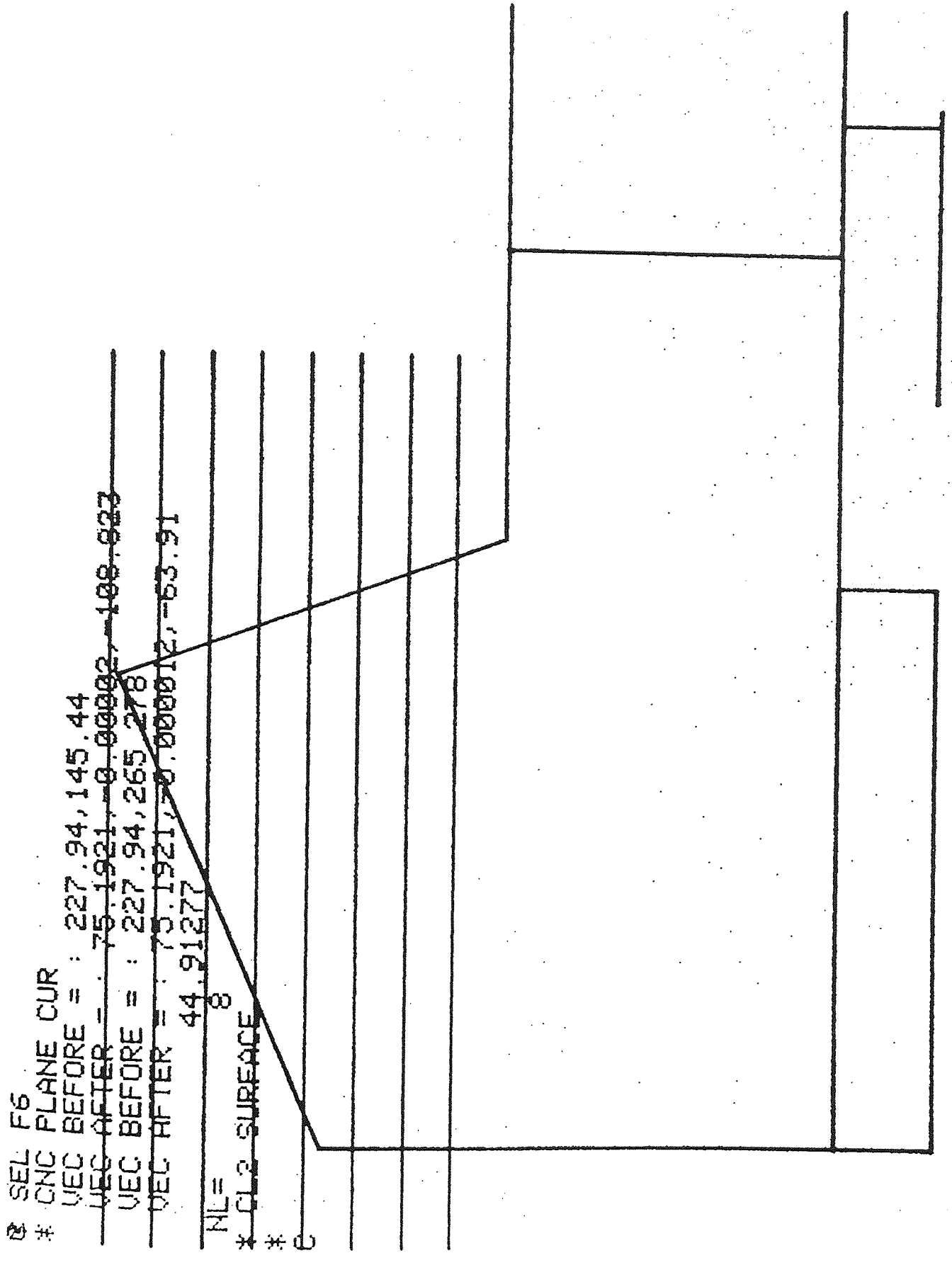


FIGURE 9.52 Planes generated

FIGURE 9.53 Perspective view of planes generated

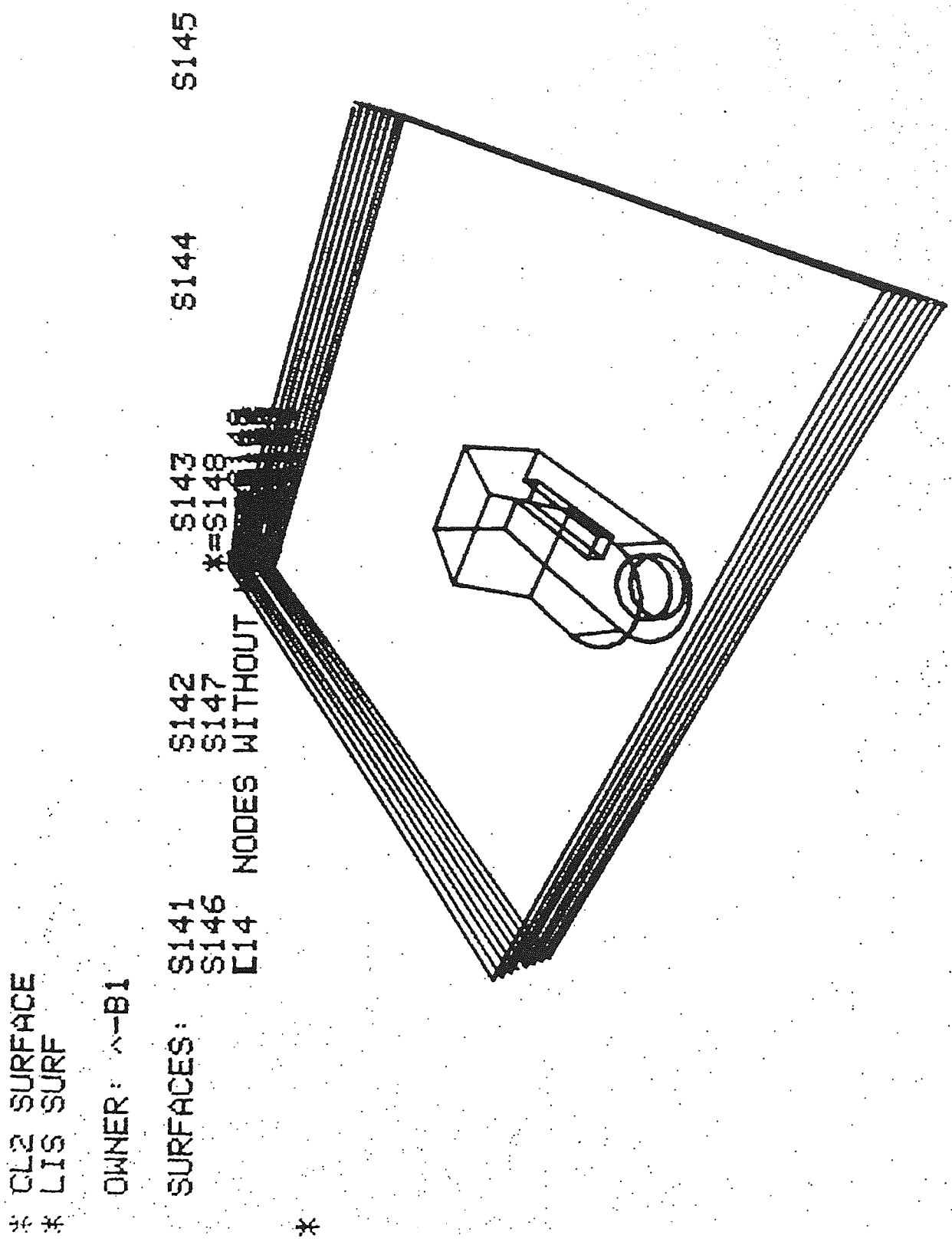


FIGURE 9 54: Procedure for layer clearance

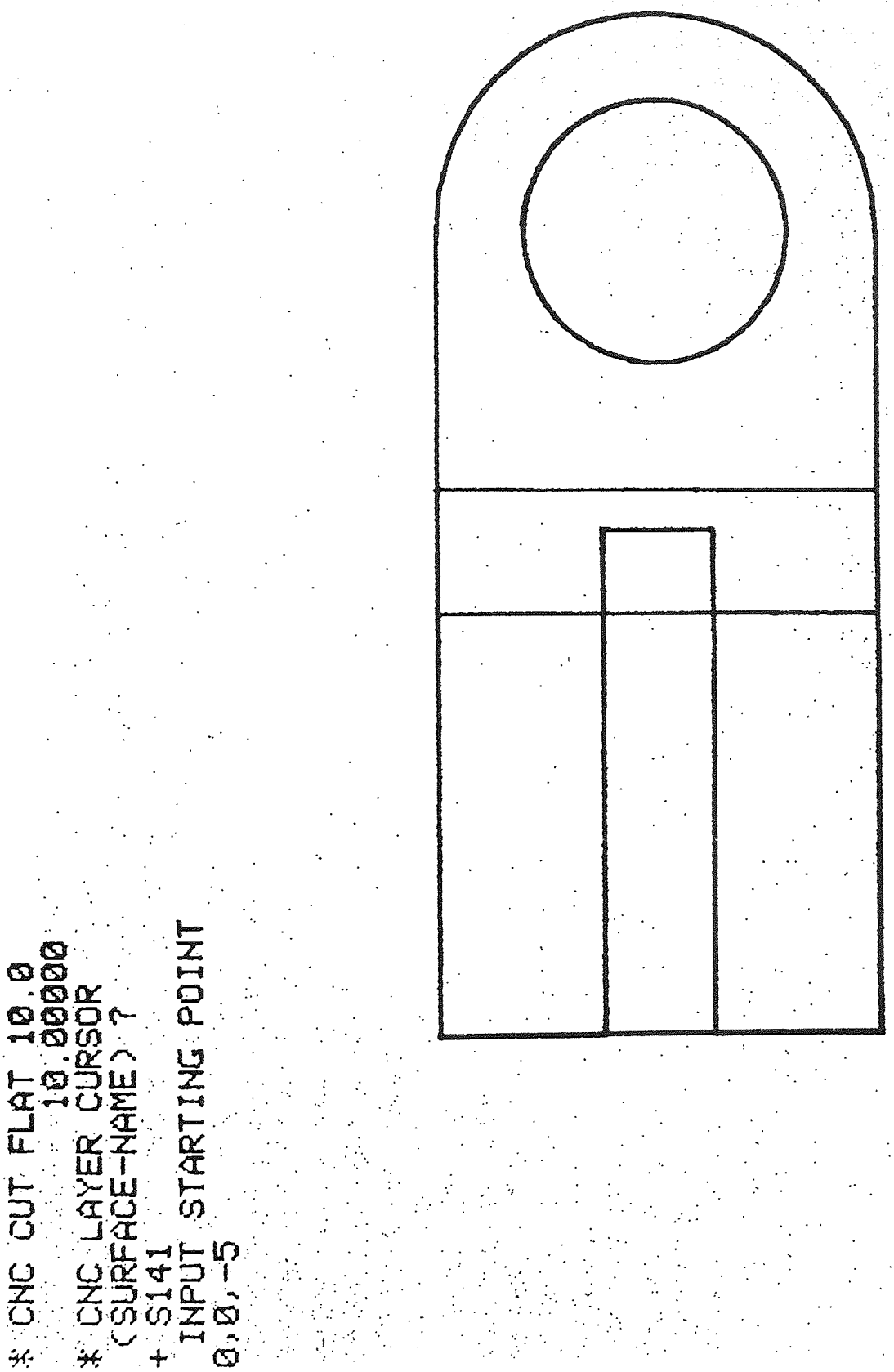


FIGURE 9.55 The cutter path pattern is generated for the layer clearance

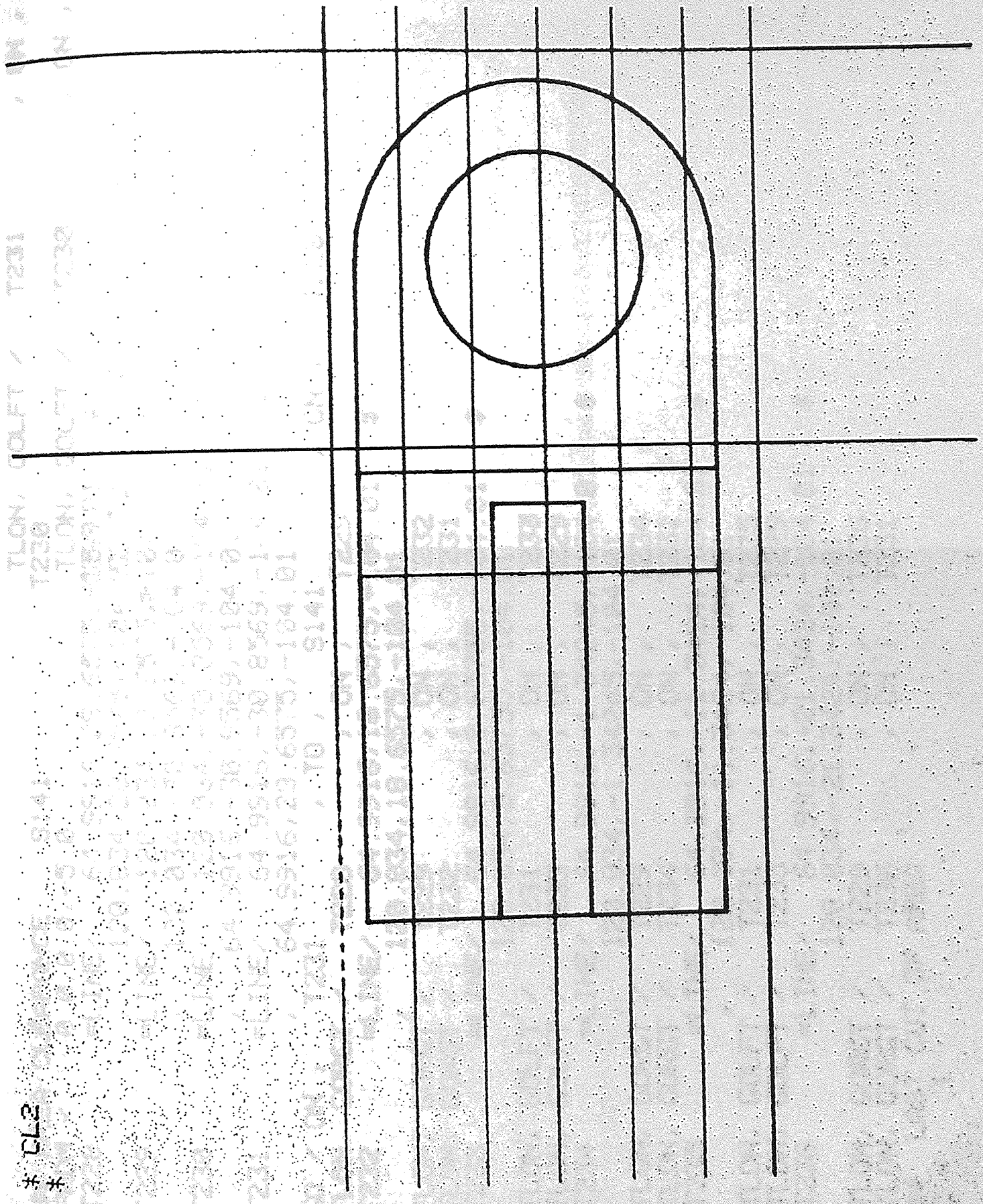


FIGURE 9.56 The corresponding APT motion statements is generated for the layer clearance

```
                                    TLON, GOLFT /   T231          , ON ,
                          T230      TLON, GOLFT /   T230          , ON ,
$$ AREA CLEARANCE    S141
FROM /  0.0,0.0,-5.0                                $    0.0,0.0,-5.0
T228     =LINE/   64.9916,28.6575,-104.01
              ,   120.034,28.6575,-104.01  GO TO /   $
T229     =LINE/   120.034,28.6575,-104.01
              ,   120.034,-30.8569,-104.01
T230     =LINE/   120.034,-30.8569,-104.01    $
              ,   64.9916,-30.8569,-104.01
T231     =LINE/   64.9916,-30.8569,-104.01    $
              ,   64.9916,28.6575,-104.01
GO / ON , T231    , TO , S141      , ON ,   T228
TLON, GORGT /   T228      , ON ,   T229
T232     =LINE/   64.9916,18.6575,-104.01    $
              ,   120.034,18.6575,-104.01
TLON, GORGT /   T229      , ON ,   T232
TLON, GORGT /   T232      , ON ,   T231
T233     =LINE/   64.9916,8.65755,-104.01    $
              ,   120.034,8.65755,-104.01
TLON, GOLFT /   T231      , ON ,   T233
TLON, GOLFT /   T233      , ON ,   T229
T234     =LINE/   64.9916,-1.34245,-104.01   $
              ,   120.034,-1.34245,-104.01
TLON, GORGT /   T229      , ON ,   T234
TLON, GORGT /   T234      , ON ,   T231
T235     =LINE/   64.9916,-11.3424,-104.01   $
              ,   120.034,-11.3424,-104.01
TLON, GOLFT /   T231      , ON ,   T235
TLON, GOLFT /   T235      , ON ,   T229
T236     =LINE/   64.9916,-21.3424,-104.01   $
              ,   120.034,-21.3424,-104.01
TLON, GORGT /   T229      , ON ,   T236
TLON, GORGT /   T236      , ON ,   T231
$$  LAST CUT OF AREA
IC =  1
```

FIGURE 9.57 The cutter path verification (plane view) of the workpiece

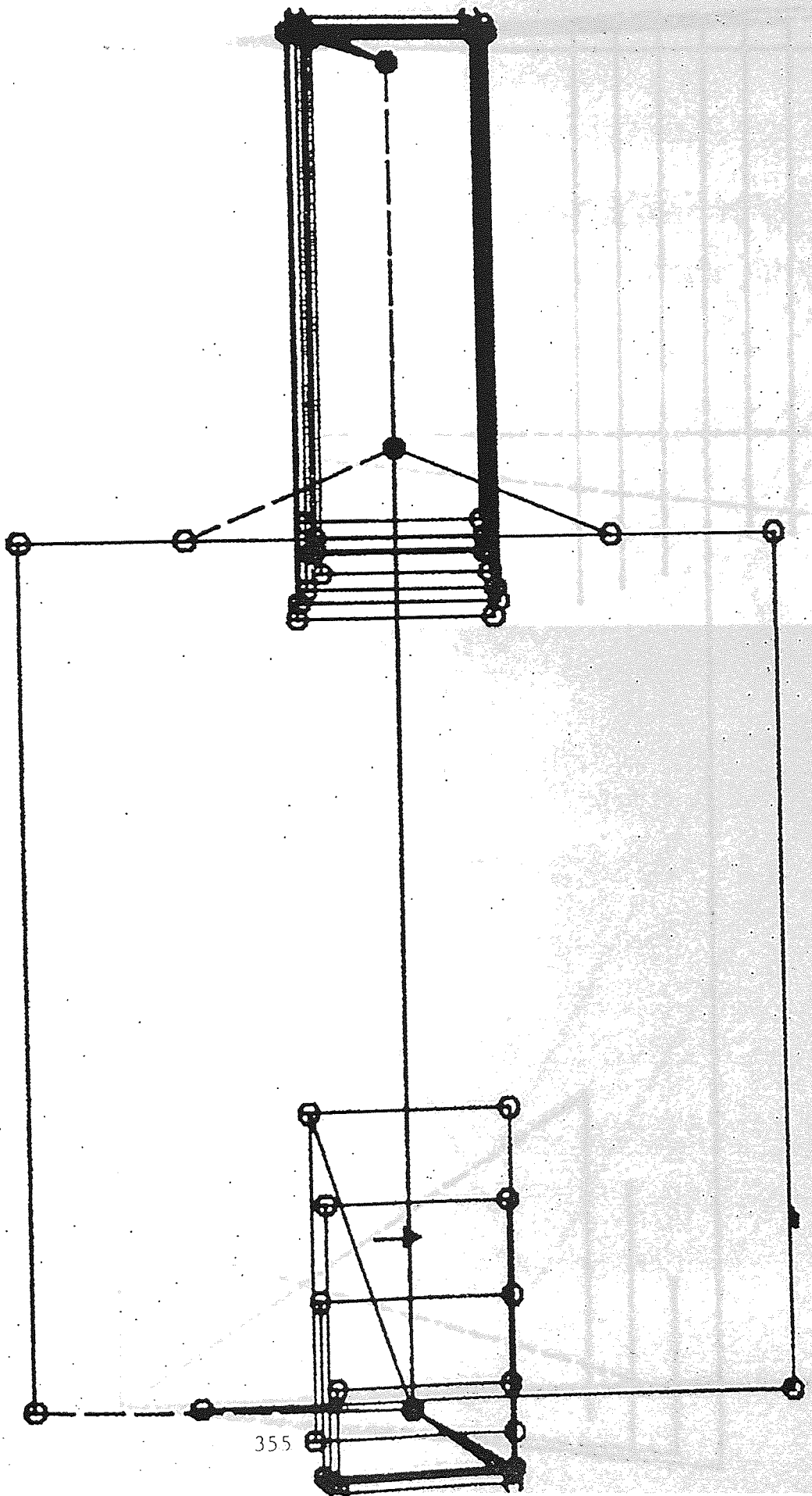


FIGURE 9.58 The cutter path verification (side view) of the workpiece

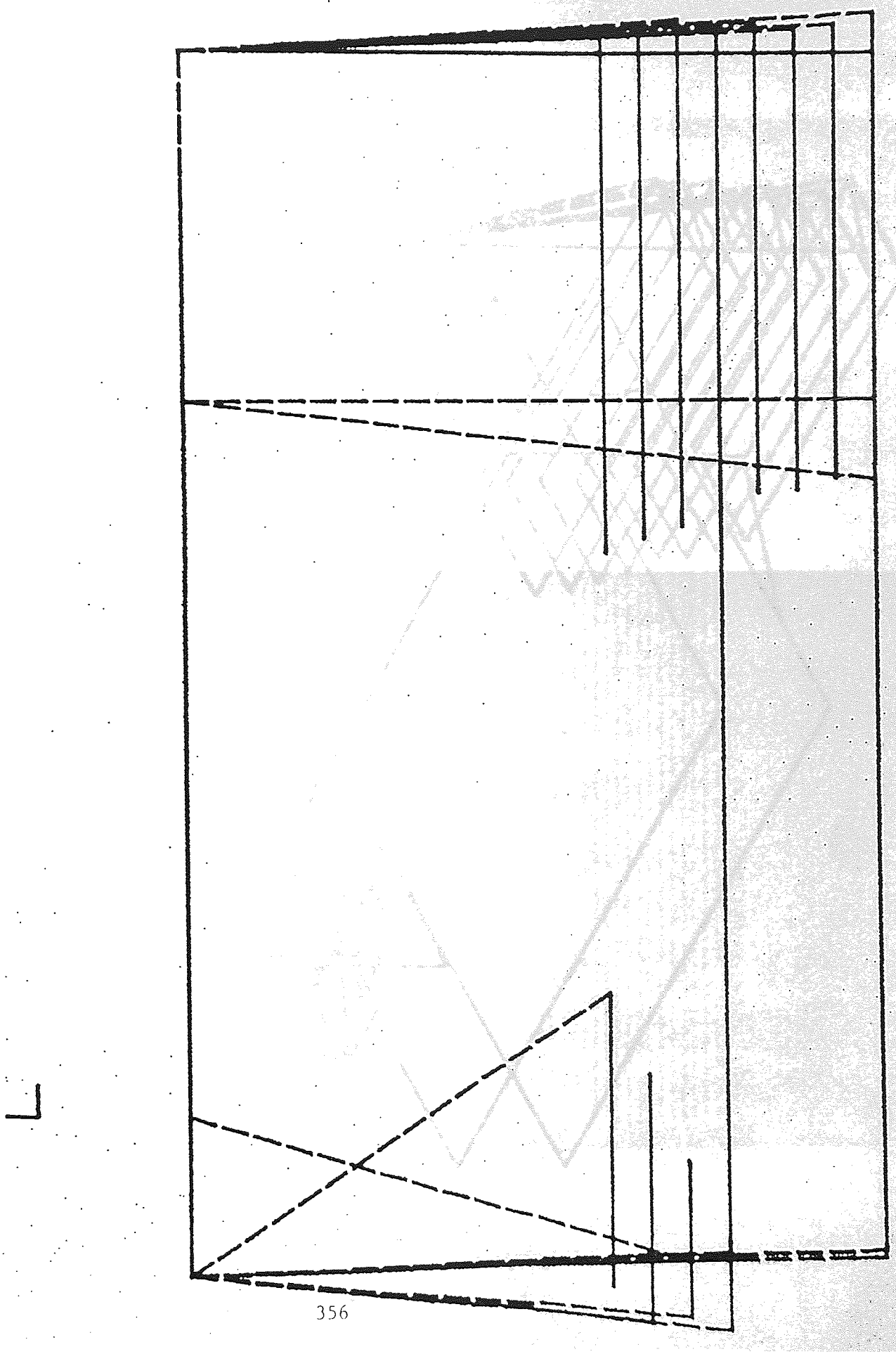


FIGURE 9.59: The cutter path verification (perspective view) of the workpiece

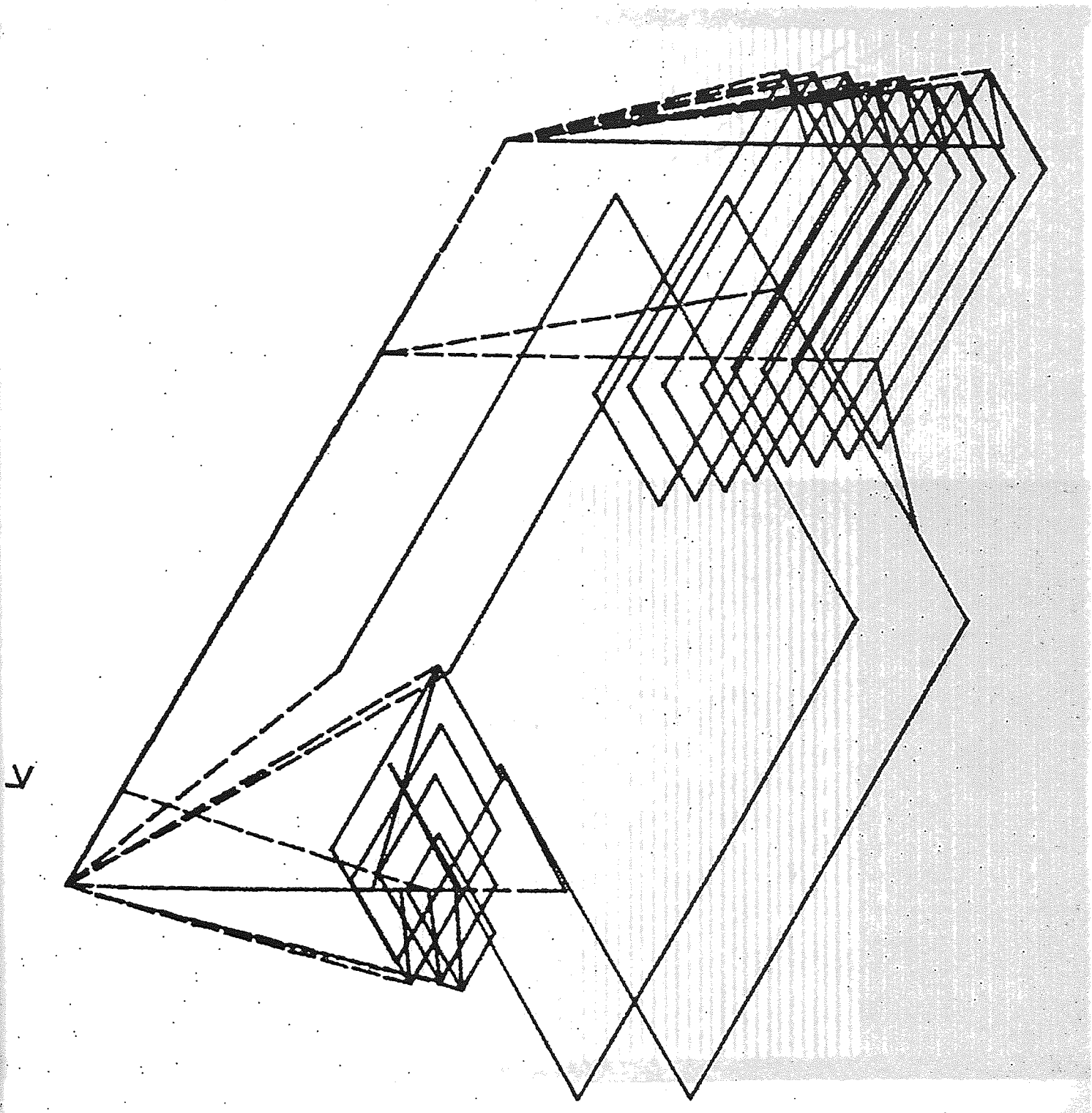


FIGURE 9.60 The cutter path verification (plane view) of the slanting planes

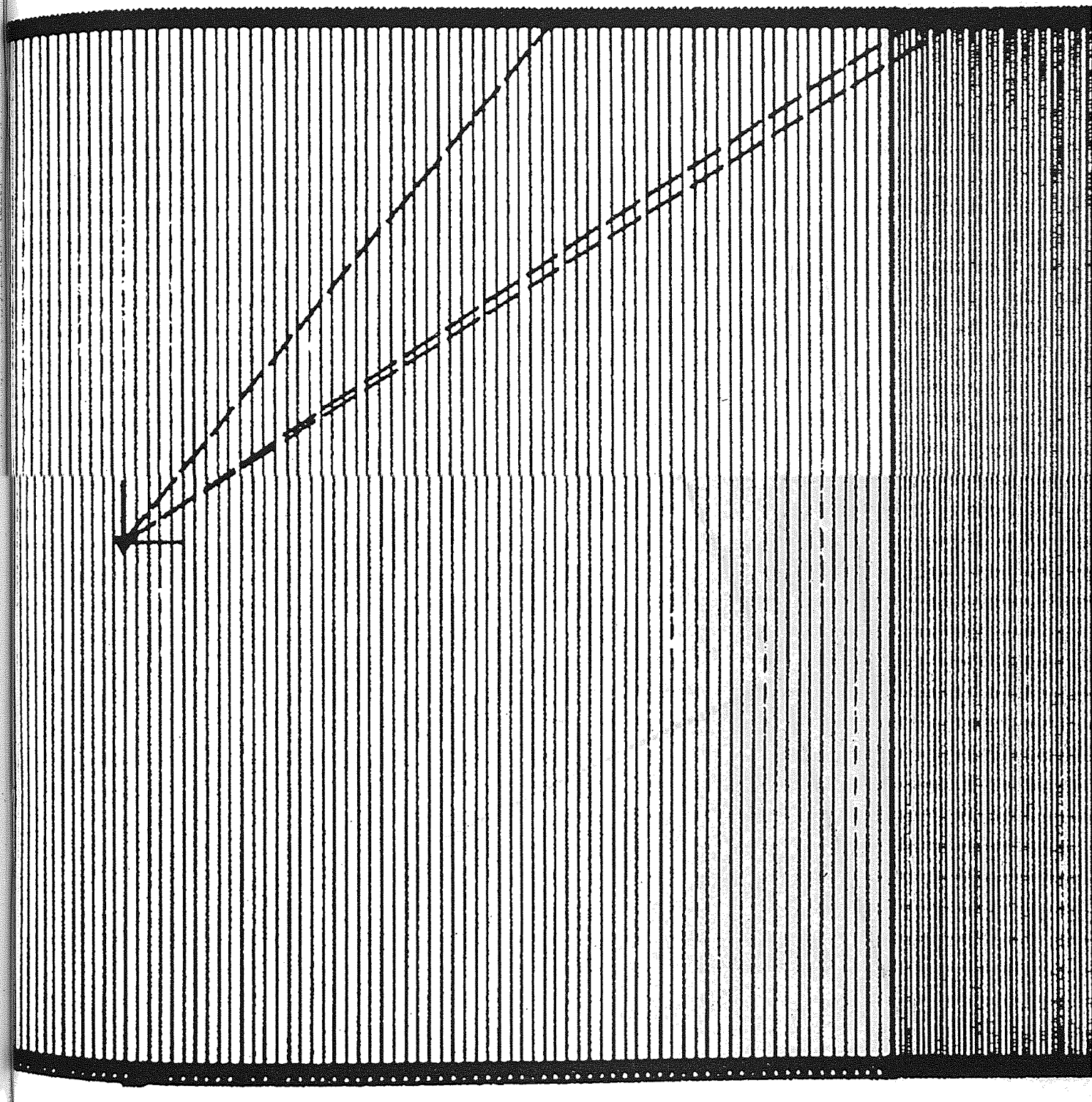


FIGURE 9.61 The cutter path verification (side view) of the slanting planes

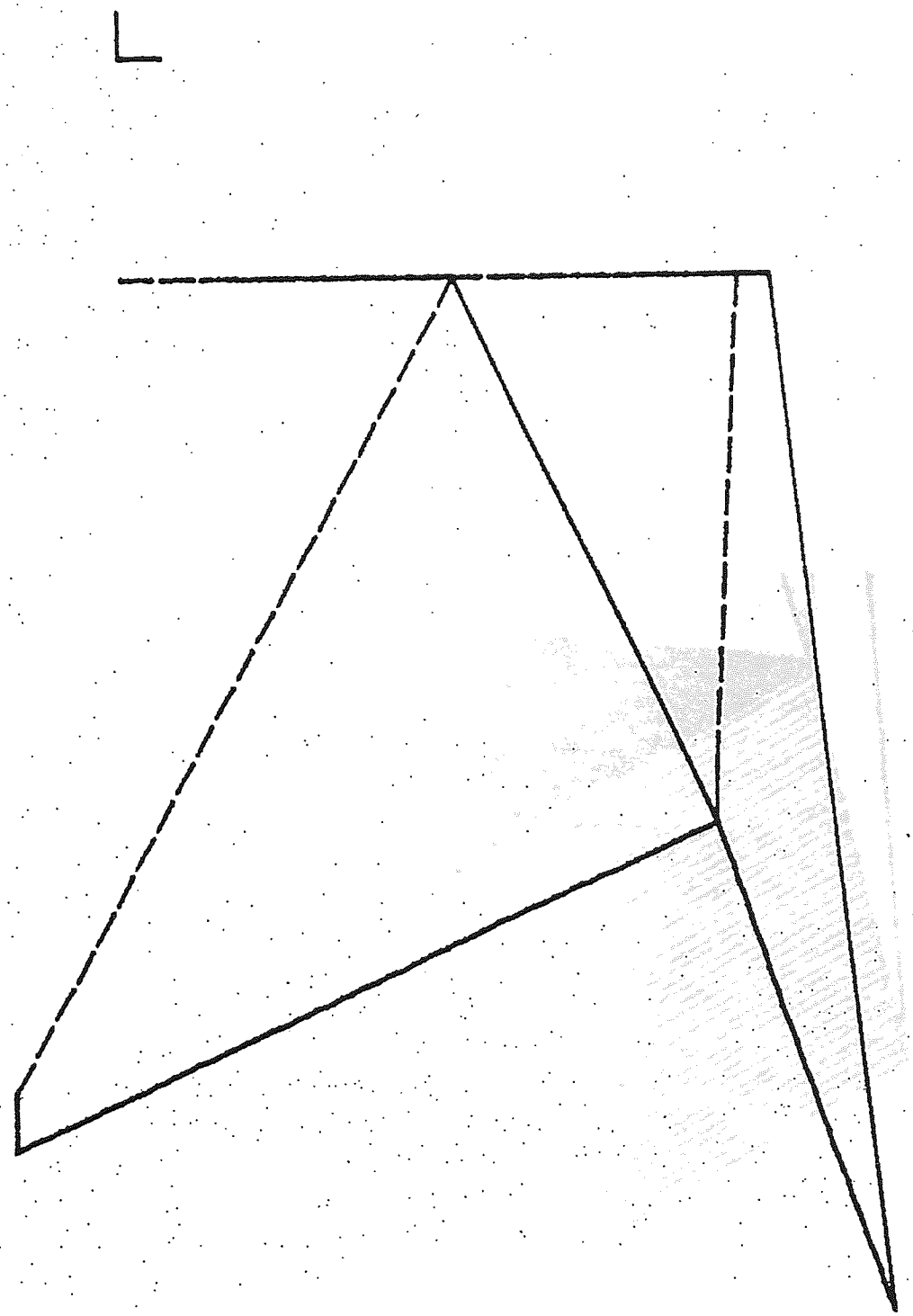


FIGURE 9.62 The cutter path verification (perspective view) of the slanting planes.

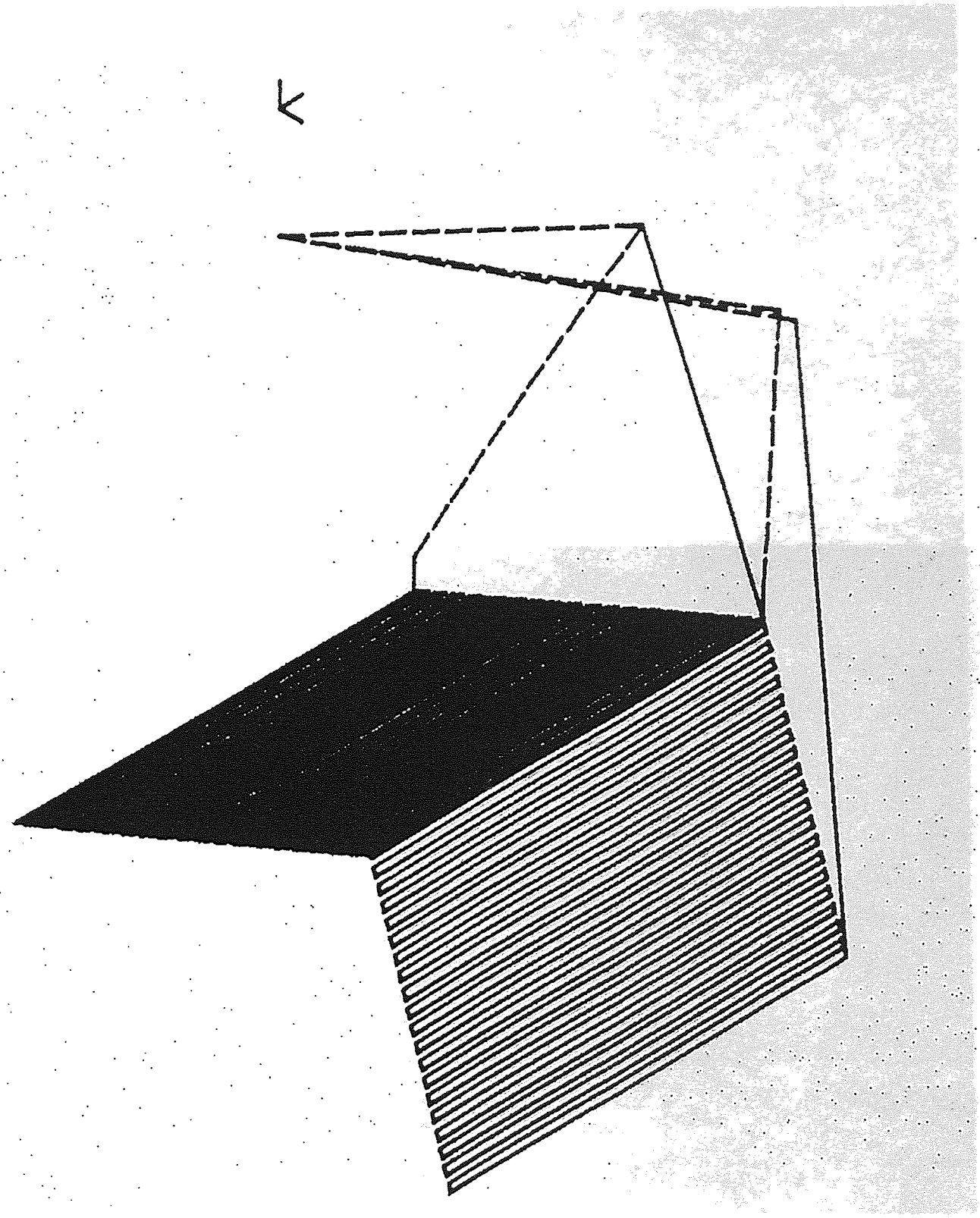


FIGURE 9.63 The finished part

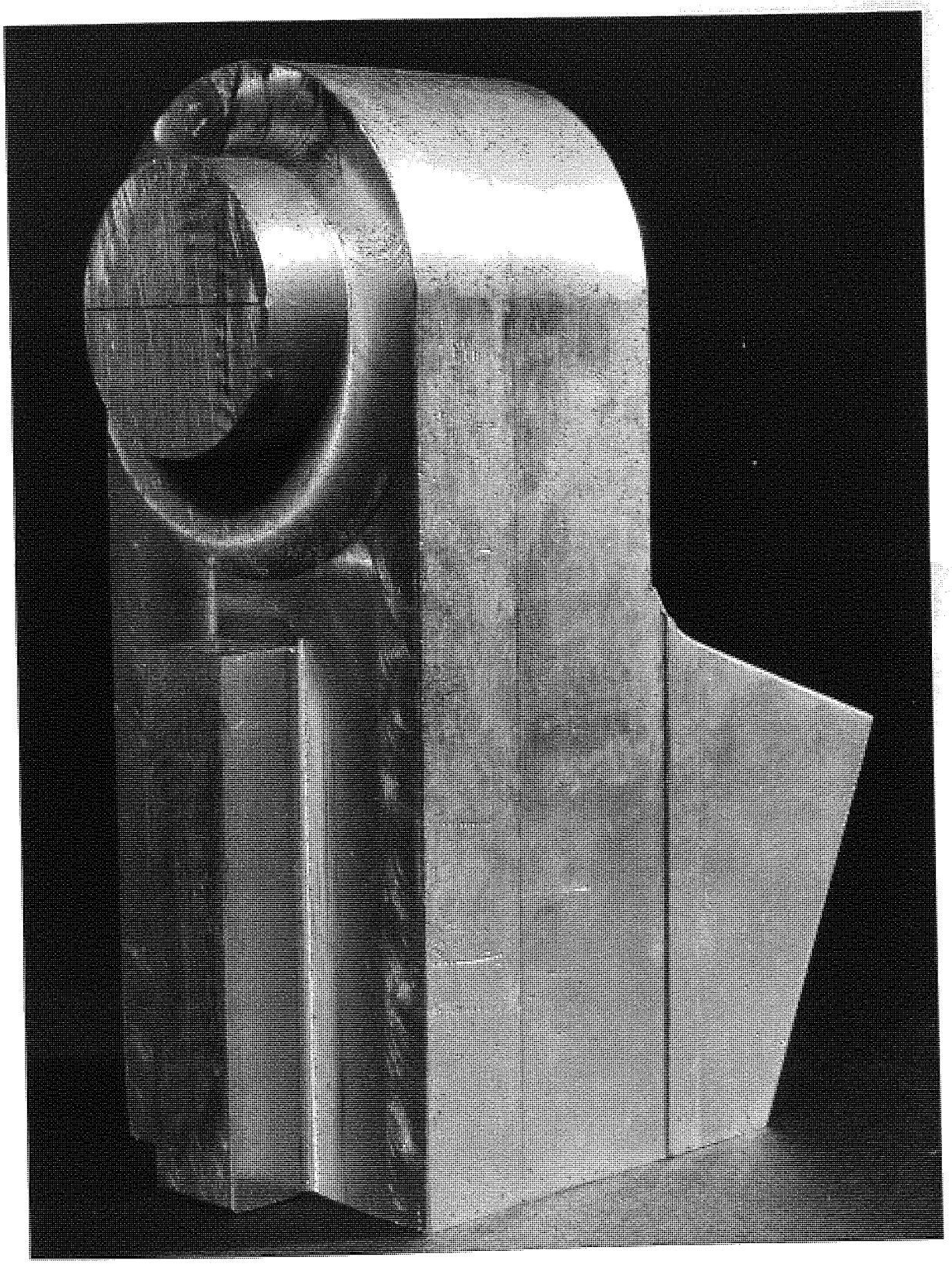
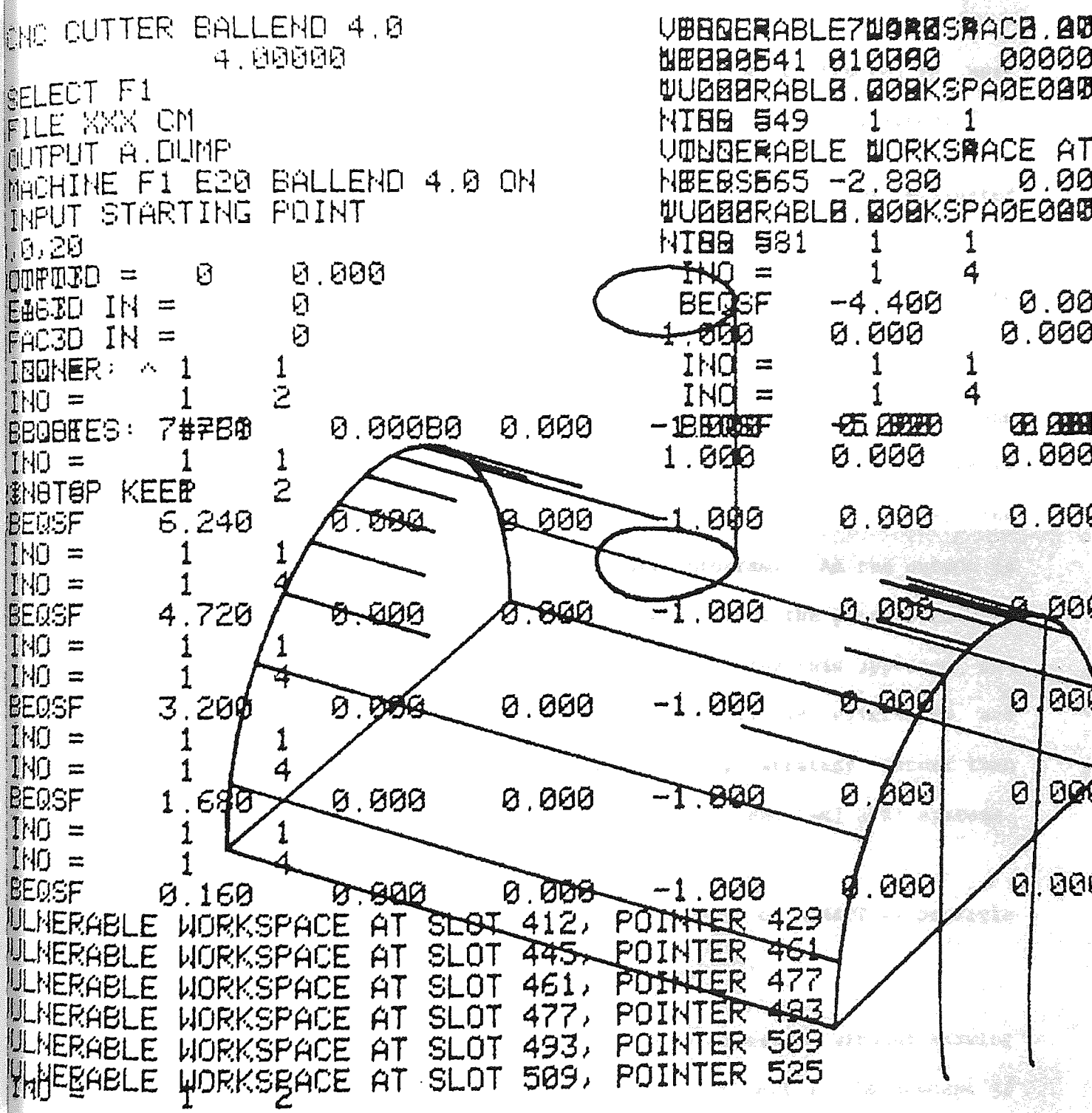


FIGURE 9.65 The cutter path generated automatically for the cylindrical face



10. CONCLUSION

This thesis investigates the link between geometric models and NC. As a result, a number of conclusions can be drawn.

1) The data structure of the geometric model can be constructed to suit the (conflicting) requirements of data storage and computing speed.

2) Comparing with other geometry representation schemes, the evaluated boundary representation geometric models, with all adjacent entities stored, are more suited for NC applications. As NC applications often require direct access to faces, edges and their adjacencies.

3) To allow for the diversity of possible manufacturing processes and machining strategies that may be used, the ROMAPT (stage 1) system is designed to output APT geometry statements and the user supplies the motion statements to complete the part program. As the output is readable to the NC programmer, he can check and edit the program to suit the task using his experience and expertise. Using this approach, the error-prone process of decoding geometry manually is eliminated and allows the user to concentrate on machining strategy rather than geometry. ROMAPT is compatible with APT-AC, APT4 and CAMI SSR1 systems.

4) For portability, a modeller-independent version of ROMAPT is possible using the CAMI Application Interface (i.e. AIAPT).

5) Automation of NC for geometric models is not feasible without knowing whether the model is machinable or not. Consequently, the concept of machinable features and models is introduced to define whether a model is machinable or not. The classification of milling operation into profiling, pocketing and facing (and their associated features) has been found practical.

6) To provide more automation for NC, a set of cutter offset functions have been introduced. These functions have been successfully applied to produce NC tape for 2 1/2 D and limited 3D parts. For general 3D parts, although the algorithms should be applicable, it would require a development effort far exceeding the scope of this work. A side product of this effort is a NC cutter path graphic generator for 2 1/2D parts.

7) The above leads to the conclusion to carry on with the work on ROMAPT to generate a complete APT part program. Hence, a facility to generate APT cutter motion statements is required. To generate APT cutter motion statements automatically, a simple expert system type approach is used in ROMAPT (stage 2). To enable more efficient computing, various data models (i.e. LOOP, FACE and OBJECT) are introduced. The validity of this approach is verified by the successful machining of a real engineering part in aluminium.

It has been demonstrated above that both the theory and algorithms developed by the author for computer aided manufacturing of 3D solid geometric models via NC are both valid and applicable.

10.1 DISCUSSION

ROAMPT is a first step in linking a geometric modelling system to a NC system directly in an integrated manner. Indeed, the full benefits of CAD/CAM can only be realized if all different components of CAD and CAM can be totally integrated (i.e. computer integrated manufacturing CIM). The justification of CAD/CAM system should not be piecemeal automation but total integration.

10.1.1 ECONOMIC

In traditional NC programming, approximately 20 percent of errors (due to ambiguous engineering drawing, outdated drawing or misinterpretaion, typing error etc) are attributed to geometry processing particlarly for complex parts. Using the automatic APT

geometry source statement facility in ROMAPT, most of these geometry errors can be eliminated. This facility is beneficial to both experienced and inexperienced NC programmers.

Using the automatic machining facilities in ROMAPT, a relatively inexperience APT NC programmer can produce working APT part programm quicker.

The main advantages of using ROMAPT is really apparent during the prototype making stages where lead time can be improved considerably. For certain new designs, the lead time can be shortened from 4 - 6 weeks to 2 - 3 weeks. In some circumstances, that is where a contract can be won or lost.

With further development of ROMAPT in machining technology, ROMAPT may eventually be used for tooling purposes.

10.2 FUTURE WORK

The future possible development of ROMAPT is suggested in the following.

10.2.1 COMPACT II

In NC machining, aparting from APT, COMPACT II is another popular NC controlling language.

Based on the theory and algorithms developed in ROMAPT, A COMPACT II version of ROMAPT can be developed. Hence, the two most popular NC languages APT and COMPACT II can both be handled by ROMAPT.

The algorithm for scanning the geometric model would be similar to that of APTALL. APTOPTIMAL and APTENQUIRE (see Secions 4.3.8.2 and 8.6) except corresponding COMPACT II language would be output instead.

10.2.2 LATHE APPLICATION

It can be seen clearly that by using the above mentioned functions, an automatic NC cutter path generator can be developed for any type of machining: drilling, turning and milling.

In lathe applications, it is usually a combination of roughing and profiling operations. It is expected that theory and technique developed in ROMAPT for milling, especially the automatic profiling (PROFILE) and regional milling (MACHINE) can be applied to turning as well.

The volume generated by subtracting the finished part from the stock (i.e. feature volume, see Section 6.2) is the material

needed to be removed. A sectioning (along the axis of symmetry) on the volume reveals that a combination of roughing cuts and a finishing profiling cut are required.

The MACHINE command can be modified to perform roughing for a turning part and the PROFILE or APTLOOP cammand can be modified for finishing cuts of contouring (see Sections 7.3.2, 7.3.3, 8.6 and 8.8).

10.2.3 5-AXIS MACHINING

The theory and algorithms in ROMAPT can be further developed for 5-axis machining applications.

For analytical part surfaces, the 5-axis machining application is envisaged to be developed along the following steps.

1) The APT geometry of the part surface can be generated by using the ROMAPT facility (see Section 4.3.8):

```
APTENQUIRE face
```

2) The surface normal at a given point can be calculated using a ROMULUS subroutine NORMFA (42):

```
NORMFA (ANORM, FA, CO)
```

where,

ANORM: normal to face.

FA: face name.

CO: co-ordinates of point at which normal is computed.

3) The cutter axis can be oriented along a surface normal v as in the following APT statement (29):

TLAXIS / [vector v/ a, b, c]

where,

v: is the vector defining the surface normal.

a, b, c: is the co-ordinate of a vector.

4) To develop an algorithm to use the information provided by steps (1) to (3) to machine a continuous 3D path on a 5-axis machine.

10.2.4 LINKING TO COMPUTER AIDED PROCESS PLANNING (CAPP) SYSTEM

The capability of the ROMAPT system can be further extended by

linking it to a computer aided process planning (CAPP) system to provide ROMAPT with most of the manufacturing and machining technology required.

An integrated CAPP and ROMAPT system would have the capability to make process planning decision such as tool selection, sequence planning and machinability (see Section 4.2).

Thus, a more practical NC machining strategy function Si and the linking function Li (see Section 6.5) can then be developed to provide further automation for NC applications.

10.2.5 REPLACEMENT OF APT ARELEM BY A GEOMETRIC MODELLER

Most of the ambiguities and unreliabilities of APT processing are caused by the arithmetic element (ARELEM) proceesor. The replacement of ARELEM by a bounded 3D solid modelling processor would resolve the above problem.

All subroutine calls to the ARELEM processor would be replaced by calls to the geometric modelling routines instead. This can perhaps be implemented via call to the Application Interface (see Section 2.6).

This system is then a truly bounded NC processor.

10.2.6 RULE BASED SYSTEM

The intelligent data processing capability in ROMAPT (see Section 8.4.2) can perhaps be extended or replaced by formal rule based artifical intelligent processor (LIST or PROLOG) to further improve its intelligence.

The intelligent data processing subroutines in ROMAPT may be replaced by routines written in LIST or PROLOG which handles all rule based calculations.

This extension will certainly strengthen the possible link between ROMAPT and a CAPP system.

10.2.7 ROBOTIC APPLICATIONS

A combination of the theoretic ground work of ROMAPT (see Section 8.2.6) and an artifical intelligence processor may eventually led to development of more intelligent and precise positional and directional controller for robots (perhaps as intelligent tool or part handlers feeding a flexible machining system (FMS)).

10.3 EXTENSION OF ROMAPT TO SCULPTURED SURFACE

The current trend of development in geometric modelling is such that sculptured surface will be incorporated into solid modeller very soon. It is therefore relevant to describe briefly how ROMAPT can be extended to handle sculptured surfaces.

10.3.1 REPRESENTATION OF SCULPTURED SURFACE

As discussed in Section 2.2, the representation of sculptured surface could be any of the three major representations: Coons, Bezier or B-spline method.

Perhaps, the analytical surfaces of an object can be modelled via the usual solid modelling technique discussed in Section 2.5. A face (or a set of faces) of the object is then modified or replaced to become a sculptured surface.

10.3.2 PARAMETRIC MACHINING OF SCULPTURED SURFACE

The usual method of machining sculptured surface of bi-cubic parametric representation (F(s,t)) is by parametric machining. The cutter path is calculated by the cutter (ball-ended) offset of a general point on the surface defined by specific values of the two parameters s and t i.e. $0 < s < 1$, $0 < t < 1$.

The main disadvantage of this method is the difficulty of controlling machining accuracy. Very often it results in over-cutting in one region and under-cutting in another region.

10.3.3 MACHINING OF SCULPTURED SURFACE IN APT

The following discussion of machining of sculptured surface in APT is based on the work of CADAPT (48) (see Section 3.2).

To machine the part defined as sculptured surface, the APT system needs to know the minimum distance along a given direction from the cutter to the surface (i.e. minimum directive distance).

10.3.4 CALCULATION OF MINIMUM DIRECTIVE DISTANCE

With reference to Figure 10.1, let

r0: position vector of cutter.

n: direction vector of cutter.

k: scalar.

r1 (u, v): any point on the patch such that

$0 < u < 1$

$0 < v < 1$

Let the vector distance between the cutter and a general point on the patch to be defined as:

$r = r0 + kn$

The distance (d) between r and r1 is given by:

$r - r1 = r0 + kn - r1\ (u, v)$

The modulus is

$d = / r0 + kn - r1\ (u, v) /$

The square of the modulus is:

$d \times d = / r0 + kn - r1\ (u, v) /^2$

Let $f = d \times d$

It can be seen that if $d = 0$, then $r = r1$, i.e.

$r0 + kn = r1\ (u, v)$

Hence k is the minimum directive distance required.

It is a minimization of f with respect to u, v, k so that f tends to 0 and d tends to 0.

A minimization is reached if :

$\delta f / \delta u = 0$, $\delta f / \delta v = 0$, and $\delta f / \delta k = 0$

subject to the constraints:

$0 < u < 1$

$0 < v < 1$

$k > 0$

Let

$$X = \begin{bmatrix} u \\ v \\ k \end{bmatrix}$$

then the above equation can be re-written in matrix form:

$$\delta f / \delta X = 0$$

Using Taylor's series:

$$f(X^*) = f(X) + f'(X)(X^*-X) + f''(X)(X^*-X)(X^*-X)/2! + \ldots\ldots$$

Differentiate the above equation w.r.t. X (and ignoring higher order terms):

$$0 = f'(X) + f''(X)(X^*-X) - f'(X)(X^*-X)(X^*-X)$$

Taking the limit for a minimum:

$f'(X)$ tends to 0

X^* tends to 0

$X^* - X$ tends to 0

Hence the term $f'(X)(X^*-X)$ can be ignored, i.e.

$$0 = f'(X) + f''(X)(X^*-X)$$

$$X^* = X - f'(X)/ f''(X)$$

i.e. $X^* = X - g(X) / G(X)$

where,

$$g(X) = \begin{bmatrix} \partial f / \partial u \\ \partial f / \partial u \\ \partial f / \partial k \end{bmatrix}$$

is the Gradiant vector.

$$G(X) = \begin{bmatrix} \partial f / \partial^2 u & \partial f / \partial u \, \partial v & \partial f / \partial u \, \partial k \\ \partial f / \partial u \, \partial v & \partial f / \partial^2 v & \partial f / \partial v \, \partial k \\ \partial f / \partial u \, \partial k & \partial f / \partial v \, \partial k & \partial f / \partial^2 k \end{bmatrix}$$

is the Hessian matrix.

A feasible initial value for X can be:

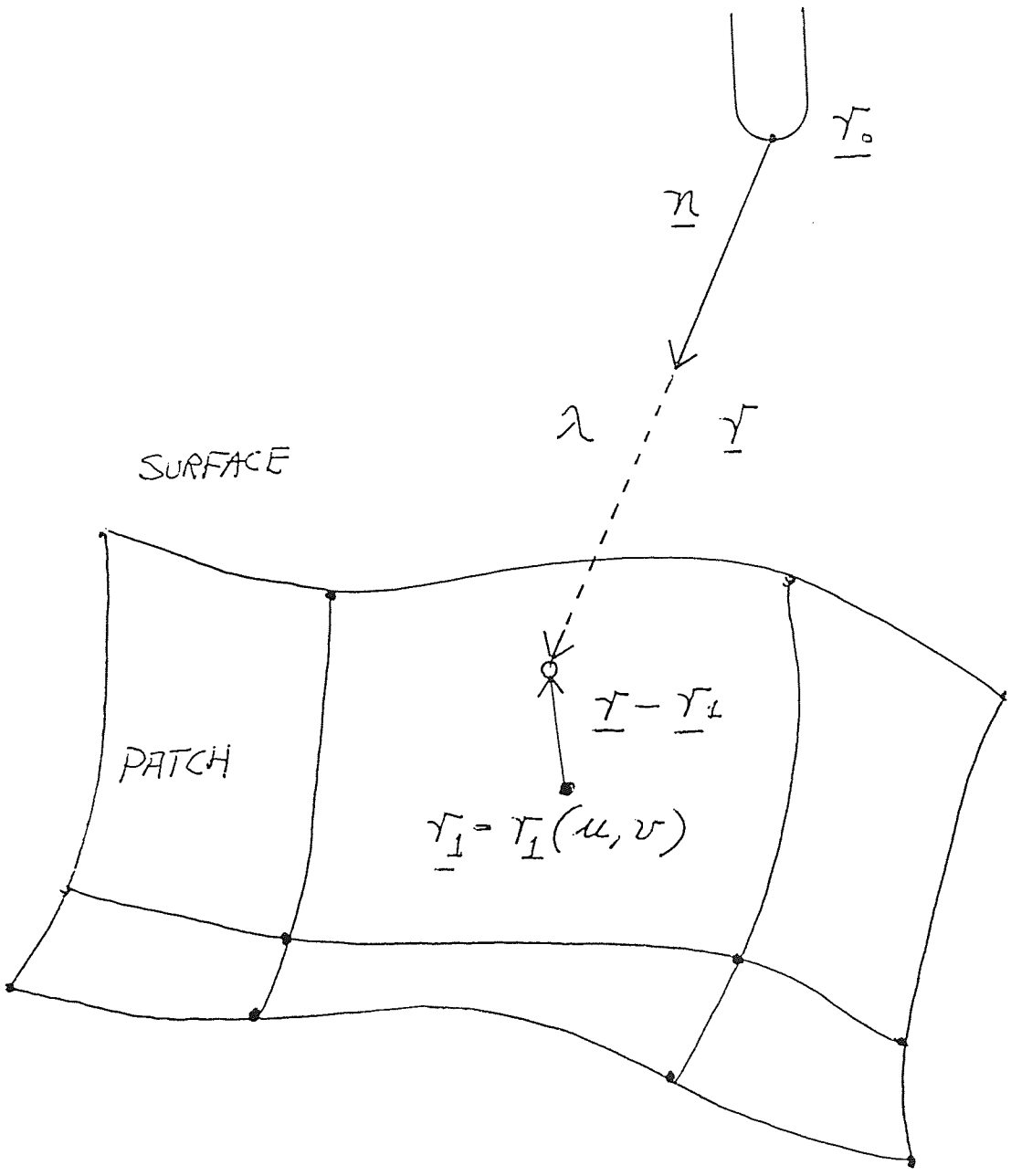
$$X = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

Then the equation $X^* = X - g(X)/ G(X)$ can be used to iterate to the required solution X^*.

10.3.5 INTERFACE TO SSR1 REGIONAL MILLING CAPABILITIES

As a possible alternative, the machining of sculptured surface can be accomplished by output regional milling statements of SSR1 (31) via ROMAPT's extended pocketing facilities (see Section 4.3.11)

FIGURE 10.1: Minimum directive distance



LUCAS GROUP CAD/CAM SYSTEMS

HARDWARE	AEROSPACE SHAFTMR LANE	AEROSPACE HUYTON	AEROSPACE HEMEL	AEROSPACE FORDHOUSES 1	AEROSPACE FORDHOUSES 2	AEROSPACE BURNLEY	AEROSPACE BRADFORD
CPU	Computervision CGP 200	Computervision CGP 200	Computervision CGP 200 (X)	Computervision CGP 200 (X)	Computervision CGP 200 (X)	Computervision CGP 200 (X)	Computervision
Console (Add'l)					2 x CCT 100		
VDU	CCT 100	CCT 100					
Printer	Centronics 353	Centronics 353			Centronics 353		
Papertape: Reader/ Punch	P	P	Facit R/P	P	P	Facit	
Plotters	Calcomp 960 Versatec 36"	Versatec 36"	CV 1000 Versatec V80	Calcomp 960	-	Versatec 36"	
Workstations: Instaview (+HCU)	1 5 + 1 HCU	4	6	1 4	3	6	
Storage							
IBM 3251 Vector Refresh							
Adage 4250 Vector Refresh							
Communications							
Hasp to Shirley Remote Number	P.w. 109	P.w. 110	(Dial) 141		(Dial)	(Dial) 142	
Applications							
Product Design/Drafting	CADDS4 202E	CADDS4 202E	CADDS4 202E	CADDS3 110E	CADDS4 202E	CADDS4 202E	
Tool Design Drafting	x	x	x	x	x	x	
NC Part Programming	x	x	x		x	x	
PCB Design			x			x	
PCB Artwork							
Integrated Circuit Design							
Phototool Tape Generation			x				
Electrical Schematic						x	
Circuit Simulation			x				
Plant Layout	x	x		MICLASS MIPLAN	MICLASS MIPLAN	x	
Process Planning			x			x	
Data Extract	x	x	x				
Finite Element Preproc	x	x					
Finite Element Analysis							

HARDWARE	CAV GILLINGHAM	CAV SUDBURY	ELECTRICAL GKS E & D	ELECTRICAL GKS TDO	ELECTRICAL CANNOCK	LUCAS ELECTRICAL E & S CRICKLEWOOD
CPU	Computervision CGP 100/200	Computervision CGP 100/200	Computervision CGP 100	Computervision Designer M	Computervision Designer M	Computervision CGP 100
Console (Add'l)		Lear Siegler	Decwriter			Lear Siegler
VDU						
Printer						Daisy Wheel
Papertape: Reader/ Punch						P
Plotters	Calcomp 1051 Benson 9336	Benson 5342	CVP 1000	Benson 1302	Benson 1302 (Calcomp 1037)	Benson 5342
Workstations: Instaview (+HCU)	6	5	6	2	2	5
Storage						
IBM 3251 Vector Refresh						
Adage 4250 Vector Refresh						
Communications						
Hasp to Shirley			Dial	P.w	Dial	
Remote Number			71	144	146	
Applications						
Product Design/Drafting	CADDS3 1103A	CADDS3 1103A	CADDS3 1104A		CADDS3 1101H	CADDS3 101H
Tool Design Drafting	x	x		CADDS3 110H		
NC Part Programming				x		x
PCB Design			x			
PCB Artwork			x			
Integrated Circuit Design						
Phototool Tape Generation						
Electrical Schematic						
Circuit Simulation						
Plant Layout		x				
Process Planning						
Data Extract	x	x	x			
Finite Element Preproc						
Finite Element Analysis						

HARDWARE	LIS LUCAS LOGIC	LUCAS RESEARCH	AEROSPACE YORK ROAD	ELECTRICAL MERE GREEN	AEROSPACE BURNLEY
CPU	DIAD turnkey system.Computer Automation 4/30 and 4/90	PRIME 550 Mk 2 general computing facility	CALMA Data General Nova 2	PRIME 250 system for electronics CAD only	RACAL REDAC MINI Design system
Console (Add'l)					
VDU		x			
Printer		x			Decwriter 2
Papertape: Reader/ Punch		x	x		
Plotters	Benson 1302	Benson 1323	Calcomp 936		Calcomp 1039
Workstations: Instaview (+HCU) Storage	1 Turnkey	2	2 Tek 4014 + Tablet Tek 4014 + Digitiser	2 colour AED (from DICOL) 1 Tek.4014	
IBM 3251 Vector Refresh Adage 4250 Vector Refresh					
Communications					
Hasp to Shirley		P.w	Mag tape to PDP11, to Ferranti Photo plotter.		
Applications					
Product Design/Drafting	DIAD	MEDUSA			
Tool Design Drafting					
NC Part Programming			CALMA		
PCB Design			x		
PCB Artwork			x	GAELIC	
Integrated Circuit Design			x	x "	
Phototool Tape Generation			x		x
Electrical Schematic	x		x	SPICE	
Circuit Simulation					
Plant Layout					
Process Planning					
Data Extract	x				
Finite Element Preproc					
Finite Element Analysis					x
Factory Facilities Planning					

HARDWARE	GIRLING TYSELEY	GIRLING CWMBRAN	GIRLING KOBLENZ
CPU	IBM 4331 Model II	IBM 4341	IBM 4341 Model II
Console (Add'l)			
VDU			
Printer			
Papertape: Reader/ Punch			
Plotters	2 x Versatec 36"	Versatec 24" and V80	Versatec 36"
Workstations:			
Instaview (+HCU)			
Storage			
IBM 3251 Vector Refresh	8	8	
Adage 4250 Vector Refresh			4
Communications			
Hasp to Shirley	(Dial)	P.w.	(Dial)
Applications			
Product Design/Drafting	CADAM 19.0		CADAM 19.0
Tool Design Drafting		CADAM 19.0	
NC Part Programming		x	
PCB Design			
PCB Artwork			
Integrated Circuit Design			
Phototool Tape Generation			
Electrical Schematic			
Circuit Simulation			
Plant Layout		x	
Process Planning			
Data Extract	x	x	
Finite Element Preproc			
Finite Element Analysis			

APPENDIX 2

LUCAS RESEARCH REPORT

Report No: Preliminary RAPT3.3

Date: March 1984

ROMAPT VERSION 3.3

COMMAND SUMMARY

BY

B. T F. CHAN

ABSTRACT

UDC No:

Ref:

LUCAS GROUP SERVICES LTD
LUCAS RESEARCH CENTRE
SHIRLEY, SOLIHULL
WEST MIDLANDS B90 4JJ

ROMAPT COMMANDS SUMMARY

A) GRAPHICS

VI2

Function: to view the object on the tektronic screen.

VI2 [WINDOW]

Function: To window on a portion of the screen via two cursor positions.

VI2 [RELEASE / SET]

Function: To release the window or set to the original perspective view.

VI2 [SHIFT] [x, y, z]

Function: To shift the view by a vector in the x-y plane.

SELECT face

SELECT face [LOOP] n

To select a face or a loop within a face for operation.

SELPLOT face [LOOP] n

Function: to select a face or a loop within a face for plotting.

APTLABEL [FACE/ POINT/ EDGE]

Function: To label all faces, or points, or edges.

APTLABEL [POINT/ EDGE] face [LOOP] n

Function: To label all points or edges of the loop n of face specified.

B) FILE HANDLING

FILE name [CM/ MM/ INCH]

Function: To output machining data to file name.

CLOSE name

Function: To close machinine file specified.

OUTPUT name

Function: To output APT statements to file name.

OUTPUT

Function: To close above file and message switches back to screen.

C) GEOMETRY

APTALL body

Function: To output all APT geometry statements of the body.

APTOPTIMAL body

Function: To output APT geoemetry statements (minimized) necessary for the machining of a 2 1/2 D part.

APTENQUIRE [FACE/ POINT/ EDGE/ TRACK/ SURFACE]

Function: To output APT geometry for all faces, or points, or edges, or tracks or surfaces.

APTENQUIRE [face/ point/ edge/ track/ surface]

Function: To output the APT geometry of the face, or point, or edge, or track or surface.

APTENQUIRE [POINT/ EDGE] face [LOOP] n

Function: To output APT geometry of all points or edges of loop n of a face.

APTINFORM

Function: APTINFORM is same as APTENQUIRE except the geometry is pseudo-APT statement. This facility provides APT-like information for users .

GENERATE

Function: To generate all labels in the geometric model in a consecutive manner.

CANCEL [FACE/ POINT/ EDGE/ TRACK/ SURFACE]

Function: To cancel labels of all faces, or points, or edges, or tracks, or surfaces.

CANCEL [face/ point/ edge/ track/ surface]

Function: To cancel label of a face, or point, or edge, or track, or surface.

CONVENTION

Function: To convert all labels in an APT-like convention, i.e.,

L: for straight lines.

C: for circles

CNC EDIT [APT statements]

Function: To input user's APT statements.

D) CONSTRUCTION GEOMETRY

CNC GRID [CURSOR/ vector 1, vector 2]

Function: To define a horizontal grid of straight lines (each separated by the amount specified by the user via the cutter definition) for area clearance operation in NC applications.

CNC PLANE [CURSOR/ height]

Function: To generate a set of parallel planes (each separated by the amount specified by the user via the cutter definition) to a horizontal base plane for roughing application.

CL2 [TRACK/ SURFACE]

Function: To view all tracks or surfaces generated.

CL2 [track/ surface]

Function: To view a track or surface.

E) MACHINING

CNC [CUTTER] [FLAT/ BALL] radius

Function: To define the cutter size.

CNC [HEADER/ MACHINE/ END]

Function: To output default APT statements for the header section, or machine section or end statement.

CNC RETRACT height

Function: To define the retracting height for the cutter.

CNC [TOLERANCE] intol outtol

Function: To define the tolerances intol and outtol for APT processing.

CNC [SCALLOP/ S2/ SOFF]

Function: To activate the scallop calculation, or half the scallop, or switch scallop calculation of.

CNC [POINT/ DISTANCE] [CURSOR]

Function: To find out the co-ordiantes of a point, or a distance between two specified points.

CNC [POCKET] [CURSOR/ CURRENT/ PSURF/ POINT/ ZAXIS/ SET]

Function: To output APT POCKET statements with the following options:

1) CURRENT: pocket on the currently selected face.

2) PSURF: pocket on user-generated planes.

3) POINT: pocket on points given (maximum 20, covex polygon).

4) ZAXIS: pocket on user defined Z-planes.

5) SET: user can redefine all default values of the pocket.

CNC [SET] point vector

Function: To change the point co-ordinate by a vector.

CNC [REGIONAL] [TO, ON, PAST]

Function: To set options for machining TO, ON or PAST the boundary of an area clearance operation.

CNC [REGIONAL] [POINT/ EDGE]

Function: To use either next point or edge as guide for the direction of the next move of the cutter.

CNC [AREA/ CLEAR/ LAYER] [CURSOR/ vector 1, vector 2]

Function: To clear off an area defined by cursor or two vectors with the following options:

1) AREA for trial run only and tracks are not output.

2) CLEAR to clear the area in the face with APT statements generated automatically.

3) LAYER: same as CLEAR but operates on user-generated planes.

PROFILE [edge 1...edge n] [LOOP] [FLAT/ BALL] radius [CURRENT/ ZAXIS/ PSURF]

Function: To generate a cutter path for a profile of a 2 1/2 D part.

CNC [LOOP] [OPEN/ CLOSE] [OUT/ ON/ IN] [MANUAL/ AUTO]

Function: To generate the data necessary for the machining of the profile in APT. The loop can be defined as OPEN or CLOSE. The cutter can be placed OUTside, ON or INside the loop. The data can be generated either MANually or AUTOmatically.

CNC [APTLOOP] [CURRENT/ PSURF/ CONSTRUCT/ ZPLANE]

Function: To output a set of APT motion statements automatically for the machining of the profile with the following options:

1) CURRENT: to operate on a currently selected face.

2) PSURF: to operate on a face.

3) CONSTRUCT: to operate on a user-generated plane.

4) ZPLANE: to operate on a user defined Z-plane.

MACHINE face edge [FLAT/ BALL] radius [ON/ TO/ PAST]

Function: To generate automatically both the cutter path and the corresponding APT statements for the machining of the face along the direction given by edge and cut to the boundary as specified by the options ON, TO or PAST.

ROMAPT: A new link between CAD and CAM

B T F Chan

As systems for CAD and production of mechanical parts have developed, there has arisen a need for techniques for the comprehensive description of the desired part, including its 3D shape. It is desirable for links to be established between these geometric modellers and machining programs.

Currently, unbounded APT and some bounded geometry systems are being widely used in manufacturing industry for machining operations such as: milling, drilling, boring and turning, applied mainly to engineering parts. APT systems, however, are presently only linked to wire-frame drafting systems. The combination of a geometric modeller and APT will provide a powerful manufacturing system for industry from the initial design right through part manufacture using NC machines.

This paper describes a recently developed interface (ROMAPT) between a bounded geometry modeller (ROMULUS) and an unbounded NC processor (APT)

computer-aided design, computer-aided manufacture, geometric modelling

The use of numerically controlled machine tools is now firmly established within the manufacturing industry. Design and manufacturing systems based on an increased armoury of mathematical techniques for manipulating shapes by computer will greatly increase the use and effectiveness of numerically controlled machine tools from now on[1-3].

One aspect of this armoury, namely, the advent of geometric modelling[4-6] techniques for the description and manipulation of 3D shapes should further this effectiveness[7]. It is essential therefore, to establish a link between geometric modelling and NC machine tools to provide an integrated CADCAM system.

However most geometric modelling systems are bounded in the mathematical sense, ie lines and planes are finite[4], whereas the controlling computer language for many NC machine tools, namely APT, is unbounded, ie lines and planes are infinite[8]. The following describes the recently developed software interface (ROMAPT) between a bounded geometric modeller (ROMULUS of Shape Data Limited) and an unbounded NC processor (APT).

DESIGN CONCEPT

The most straightforward approach is to design a bounded NC processor to interface with a bounded geometric modeller. However, the manufacturing industry has invested substantially over the years in APT and similar NC processors. It is very unlikely that the industry would be willing to shed this investment instantly. The alternative approach is to design a link between a bounded geometric modeller and an unbounded NC processor like APT.

There are two possible approaches to the design of the interface.

Low level (CLFILE)

Assuming a particular machining strategy and manufacturing process are decided, the output of the interface can be an APT compatible CLFILE which can be post-processed to generate NC tapes. However, the development and debugging cost of using this approach on a 3D geometric modeller will be considerable. As the output is a CLFILE which is unreadable to the NC programmer, he cannot check conveniently whether the program has performed correctly or not. It is this difficulty of use that prevents some potentially good systems from being used more widely.

Higher level (APT source)

Alternatively, to allow for the diversity of possible machining strategies and manufacturing processes that may be used, the interface can be designed to output APT geometric source statements. In this case the output is readable to the NC programmer; he can check and edit the APT source to suit the task. Conceptually, this approach is very straightforward and easy to use and offers a much lower development cost in comparison to the first approach. Thus, the second approach was adopted with the following objectives:

- to provide an easy-to-use interface
- to enable the NC programmer to understand the geometric model better
- to provide APT geometric source statements as required

Within Lucas Industries, there is a substantial investment in APT and with it a concomitant pool of NC programming expertise. With due consideration for the above mentioned factors, the ROMAPT system was designed to convert bounded geometric information held in the geometric modeller to standard unbounded APT geometry statements. The experience and expertise of the NC programmer is then used to complete the NC part program. This further allows the NC programmer to concentrate on the machining strategy rather than the geometry.

ROMAPT SYSTEM CAPABILITY

Traditionally, an NC programmer is given a set of engineering drawings of the part he wishes to make. He has to understand and interpret the drawings accurately before he proceeds to write his NC part program. This is where much time is consumed and many errors are possible.

Engineering Department, Engineering Computing, Lucas Research Centre, Shirley, Solihull, West Midlands B90 4JJ, UK

Working with a geometric modeller, in addition to the set of engineering drawings given, will help to lessen these drawbacks. However, the NC programmer needs to be able to query and understand the part given in the form of a geometric model defined by a data structure[10].

Manipulation

Supplementary to the manipulation facilities (ie rotation, viewing) provided by ROMULUS[9,10], further manipulating facilities have been developed to help a user's understanding of the geometric model. All these additional facilities help a user to generate APT geometry statements for a component. These extra commands are described below:

- VIEW: allows viewing of the object via a display screen
- ROTATE: allows rotation of the object at will
- SCALE: allows scaling of the object
- PROJECT: allows views of third angle projections, similar to a conventional engineering drawing
- GENERATE: generates name for unnamed geometric entities (points, edges, faces). The following naming conventions are used: P for points, E for edges and F for faces
- CANCEL: cancels name(s) of geometric entities (point, edge, face)
- LABEL: labels name(s) of required geometric entities (point, edge, face) on a display drawing
- ENQUIRE: outputs required geometric information of the model of the part
- SHIFT: shifts the picture on display in X and Y directions
- WINDOW: provides windowing facilities on current view, so that a more complex geometric model can be handled

Geometry statement generation

The ROMAPT program scans through the data structure held in the geometric model in a systematic manner : first the body is scanned, then the faces, followed by the points and edges (shown in Figure 1). This geometric data (bounded) is then converted into standard APT geometry format (unbounded) (Figure 2 and Figure 3). The APT geometric statements for a component are output according to the hierarchy in Figure 1. The components are processed individually. For each component (ie a body), the APT geometric definition for every face will be output and its associated points and lines will be output accordingly in a systematic manner.

Standard APT-AC[8] geometric definitions are employed, as in the following examples:

- Points:

  P1 = POINT/x, y, z

  The point P1 is given in (x, y, z) co-ordinates.

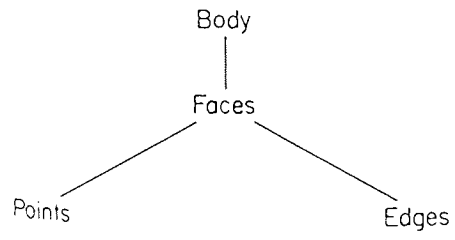


Figure 1. Hierarchy of geometric entities

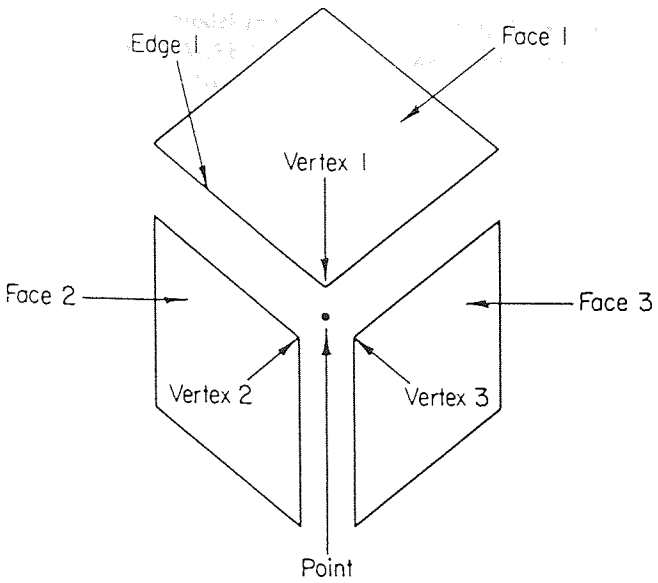


Figure 2. Bounded geometric model

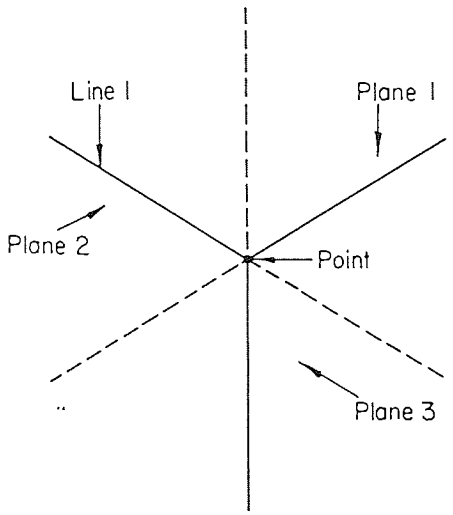


Figure 3. Unbounded geometric model

- Edges:

  E1 = LINE/P1, P2

  E1 is a straight line defined between point P1 and point P2.

  E2P = POINT/x, y, z
  E2 = CIRCLE/CENTER, E2P, RADIUS, t
  $$ P1 TO P2

  E2 is a circular arc or a circle defined between point P1 and point P2 with centre at E2P(x, y, z) and radius of t units.

  E3 = ELLIPS/INTOF, F1, F3
  $$ P3 TO P4

  E3 is an ellipse between point P3 and point P4 and is defined by the intersection of plane F1 and cylinder F3.

- Faces:

  F1P = POINT/x, y, z

  F1V = VECTOR/u, v, w

  F1 = PLANE/F1P, PERPTO, F1V

  F1 is a plane defined by a point on the plane at F1P (x, y, z) and the normal to the plane : vector F1V(u, v, w).

F3P = POINT/x, y, z
F3V = VECTOR/u, v, w
F3 = CYLNDR/F3P, F3V, t

F3 is a cylinder defined by a point F3P(x, y, z) on the cylinder axis and a vector F3V(u, v, w) along the cylinder axis and the radius (t) of the cylinder.

F4P = POINT/x, y, z
F4V = VECTOR/u, v, w
F4 = CONE/F4P, F4V, d

F4 is a cone defined by the vertex point F4P(x, y, z), the axis vector F4V(u, v, w) and the half-angle (d) of the vertex of the cone.

F5P = POINT/x, y, z .
F5 = SPHERE/CENTER, F5P, RADIUS, r

F5 is a sphere defined by the centre F5P(x, y, z) and the radius (r).

F6P = POINT/x, y, z
F6V = VECTOR/u, v, w
F6 = TORUS/F6P, F6V, r1, r2

F6 is a torus defined by the centre point F6P(x, y, z), the axis vector F6V(u, v, w), the major radius (r1) and the minor radius (r2).

The ROMAPT interface consists mainly of 3 commands in 3 different modes of operations:

- APTALL (dump mode): generates APT geometric statements for all geometric entities of the body concerned.
- APTOPTIMAL (optimized mode): generates an optimal set of APT geometric statements (mainly for 2½D parts).
- APTENQUIRE (interactive mode): generates APT statements for requested geometric entities interactively to enable an NC programmer to select the minimum set of APT geometric statements for the component concerned.

SYSTEM IMPLEMENTATION

The ROMAPT system is currently coded in standard Fortran (ANSI X3.9-1966) and implemented on a PRIME 400 computer.

MODES OF OPERATION

An NC programmer can use APTALL to generate a complete definition of the geometric model in standard APT geometry statements for each and every geometric entity of the model.

An optimal set of geometric statements necessary for the NC programming can be obtained by using APTOPTIMAL, mainly for 2½D parts.

APTENQUIRE can be used to pinpoint the particular entities required; thus an even smaller set of geometric statements can be obtained by the NC programmer.

WORKED EXAMPLE

Let us assume an artificial mechanical part GEHAUSE is designed by a design office using ROMULUS. (The GEHAUSE has come to be a standard test part of geometric modellers. It was conceived by Messerschmitt-Bolkow-Blohm for the CAM-i Geometric Modelling Seminar, Bournemouth, UK, 1980 as a means of comparing the capability and performance of developed geometric modellers.)

The following procedure is adopted for the demonstration and the effective use of the ROMAPT system:

- A geometric model and a set of engineering drawings of the part GEHAUSE (as shown in Figures 4(*a*) and 4(*b*)) are given to an NC programmer.

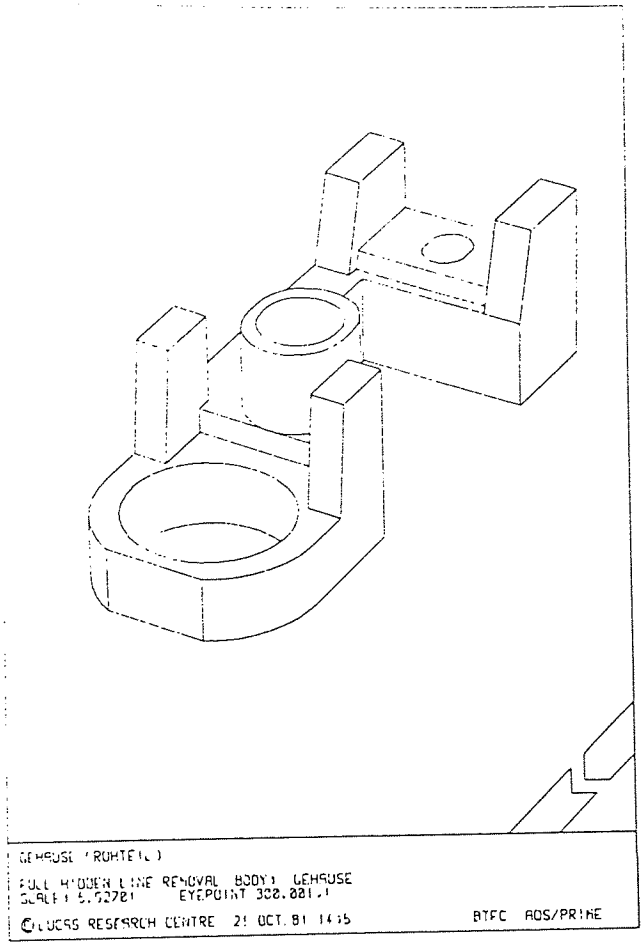


Figure 4(a). GEHAUSE (hidden line view)

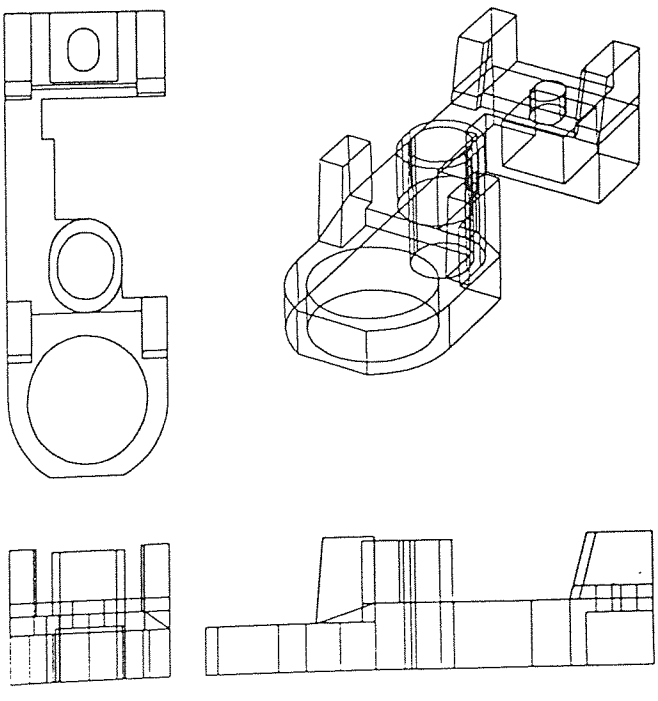


Figure 4(b). GEHAUSE and its projections (orthogonal view)

- The programmer uses a combination of the VIEW/WINDOW/SKETCH/LABEL/ENQUIRE[1] commands of ROMAPT interactively, to understand the geometry of the part. (The computer monitor file in Figure 5(*a*) has been commented comprehensively to provide cross-references between computer operations and corresponding figures.)
- Any part of the model can be windowed and enlarged for viewing. Face labels can be displayed to identify faces of interest (Figure 6(*a*)). (Note that names for POINT/EDGE/FACE are generated by the system automatically.)
- Any face can be pinpointed and selected together with its associated loops, points and edges (Figures 6(*b*), 6(*c*), 7(*a*), 7(*b*)).
- Any face can be windowed and enlarged for viewing so that more complex geometry can be handled (Figure 7(*c*)).
- Repeat the last four steps until the whole geometry of the part is understood by the NC programmer.
- When the geometry of the part is understood by the NC programmer, he can then select the convenient machining angle (default *z*-axis) and output any APT geometry of the part selectively and interactively via the ROMAPT command APTENQUIRE. (The APT geometry output is shown in Figure 5(*b*)).
- Repeat the last step for other faces required for machining.

Alternatively, the programmer can generate an optimal set of APT geometric statements of the part by using the

```
@# MONITOR FILE CREATED  14:21 26 APR 82  BY  BTFC
@#  ROMULUS  VERSION 3.3
 OPTION TERMINAL 4010
@# ACCESS THE FILE CONTAINING THE GEOMETRIC MODEL : GEHAUSE
 GET ROMTEMP>BODY.G
@# LIST BODIES
 LIST
@# GENERATE NAMES FOR POINT/EDGE/FACE OF THE MODEL IF NOT
@# ALREADY DONE SO BY THE PROGRAM
 GENERATE GEHAUSE
@# SKETCH A VIEW OF THE BODY
 SKETCH
@# PLOT A HIDDEN LINE VIEW OF GEHAUSE  FIGURE 4a
 HLPLOT
@# PLOT A 3D ENGINEERING (ORTHOGONAL) VIEW OF GEHAUSE : FIGURE 4b
 OPTION ORTHO
 SKPLOT
@# MONO VIEW
 OPTION MONO
 SKETCH
@# USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
@# PLOT THE WINDOW VIEW : FIGURE 5a
 SKPLOT
@# LABEL ENTITIES INTERESTED
 APTLABEL FACE
@# PLOT THE FACE TOP WITH LOOPS LABELLED : FIGURE 5b
 SEPLOT TOP
 SELECT TOP LOOP
 APTLABEL TOP
@# PLOT THE FACE TOP WITH POINTS/EDGES ON LOOP 1 : FIGURE 5c
 SEPLOT TOP
 APTLABEL POINT TOP LOOP 1
 APTLABEL EDGE TOP LOOP 1
@# RELEASE THE WINDOW
 VIEW RELEASE
 SKETCH
@# SELECT THE BOTTOM FACE BASE
 SELECT BASE
@# PLOT THE FACE BASE WITH LOOPS LABELLED : FIGURE 6a
 SEPLOT BASE
 SELECT BASE LOOP
 APTLABEL BASE
@# PLOT THE FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIGURE 6b
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
@# USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
@# PLOT THE ENLARGED FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIG. 6c
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
@# THE ABOVE PROCEDURE CAN BE REPEATED FOR OTHER FACES
@# UNTIL THE GEOMETRY OF THE PART IS UNDERSTOOD BY THE USER.
@# WHEN THE GEOMETRY IS UNDERSTOOD , THE USER CAN PROCEED TO
@# OUTPUT THE APT GEOMETRY SELECTIVELY AS IN THE FOLLOWING : FIGURE 7b
 OUTPUT APTDATA
 APTENQUIRE BASE
 APTENQUIRE POINT BASE LOOP 1
 APTENQUIRE EDGE BASE LOOP 1
 OUTPUT
@# STOP KEEP
OK,
```

Figure 5(a). Computer monitor file (recording operations on ROMAPT)

```
$$  THE FOLLOWING 3 LINES ARE PLANE DEFINITION        FOR   BASE
BASEP       = POINT/   0.0,0.0,0.0
BASEV       = VECTOR/   0.0,0.0,-1.0
BASE        = PLANE/  BASEP       , PERPTO ,   BASEV
P0          = POINT/   0.0,9.1,0.0
P1          = POINT/   0.0,12.8,0.0
P2          = POINT/   6.4,12.8,0.0
P3          = POINT/   6.4,3.0,0.0
P4          = POINT/   9.39999,3.0,-0.0
P5          = POINT/   9.39999,4.0,-0.0
P6          = POINT/   15.4,4.0,0.0
P7          = POINT/   15.4,6.4,0.0
P8          = POINT/   18.3,9.3,0.000002
P9          = POINT/   21.4,9.3,0.0
P10         = POINT/   21.4,12.8,0.0
P11         = POINT/   28.9,12.8,0.000001
P12         = POINT/   34.6,9.31032,0.0
P13         = POINT/   34.6,3.48967,0.0
P14         = POINT/   28.9,-0.000003,0.000001
P15         = POINT/   0.0,0.0,0.0
P16         = POINT/   0.0,3.7,0.0
P17         = POINT/   5.2,3.7,0.0
P18         = POINT/   5.2,9.1,0.0
E0          = LINE/  P0          ,  P1
E1          = LINE/  P1          ,  P2
E2          = LINE/  P2          ,  P3
E3          = LINE/  P3          ,  P4
E4          = LINE/  P4          ,  P5
E5          = LINE/  P5          ,  P6
E6          = LINE/  P6          ,  P7
E7P         = POINT/   18.3,6.4,0.000002
E7          = CIRCLE/CENTER,  E7P        , RADIUS ,  2.9
  $$   P7          TO   P8
E8          = LINE/  P8          ,  P9
E9          = LINE/  P9          ,  P10
E10         = LINE/  P10         ,  P11
E11P        = POINT/   28.9,6.4,0.000001
E11         = CIRCLE/CENTER,  E11P       , RADIUS ,  6.4
  $$   P11         TO   P12
E12         = LINE/  P12         ,  P13
E13P        = POINT/   28.9,6.4,0.000001
E13         = CIRCLE/CENTER,  E13P       , RADIUS ,  6.4
  $$   P13         TO   P14
E14         = LINE/  P14         ,  P15
E15         = LINE/  P16         ,  P15
E16         = LINE/  P16         ,  P17
E17         = LINE/  P17         ,  P18
E18         = LINE/  P18         ,  P0
OK,
```

Figure 5(b). APT geometry statements

APTOPTIMAL command of the system. The NC programmer completes the NC program by adding the necessary APT machining instructions to the APT statements generated. The NC part program can then be processed by APT and a postprocessor to generate NC tapes.

The machining strategy adopted for this worked example consists essentially of 3 steps:

- area clearance
- profile machining
- machining of holes

Figures 8(*a*) and 8(*b*) illustrate the cutter path generated for the GEHAUSE part.

DISCUSSION

The following is a discussion on some of the features of the system.

Names

Names of POINT/EDGE/FACE are the usual means of communication between the user and the system. These names are generated automatically by ROMULUS (though not consecutively). There are also facilities for cancelling and re-generating (consecutively) of names in ROMAPT. These names can be replaced by user defined names via the renaming facility.

Machining strategy

Currently, the NC programmer has complete freedom in deciding the machining strategy and in the choosing of every separate tool path.

Accuracy and rounding errors

The accuracy of the system is limited inherently by the computer hardware used. Rounding errors may occur as a

result of a tradeoff between computing time and resolution. In testing for approximate equality, ROMULUS makes use of two quantities: REABS for large values and RENOR for normalized quantities. On entry to ROMULUS, REABS and RENOR are set to be 0.002 and 0.00002 respectively for

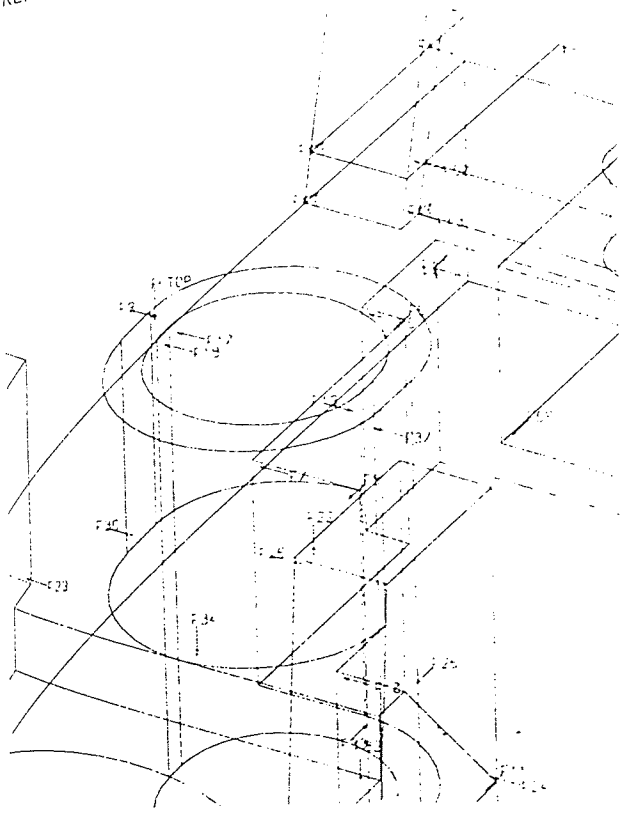


Figure 6(a). Enlarged details of the GEHAUSE (via window)

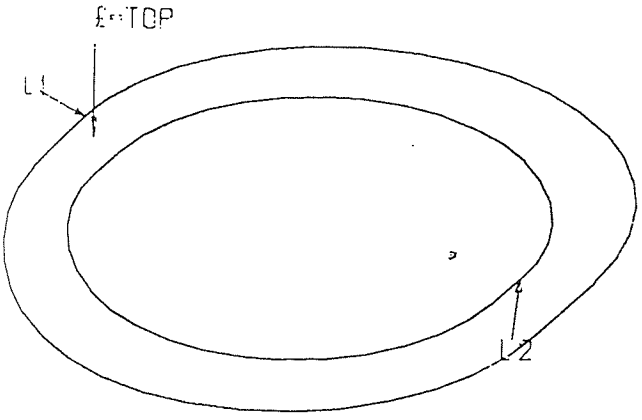


Figure 6(b). Loops of face: TOP of the GEHAUSE

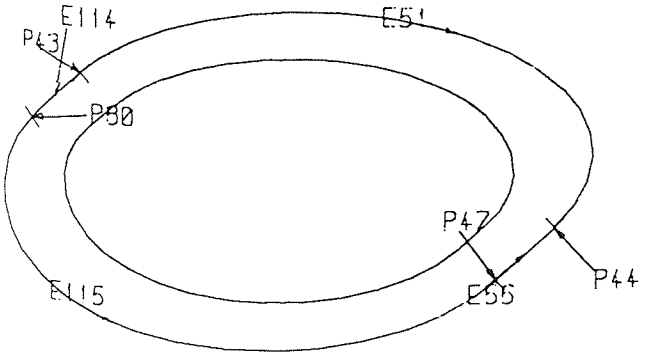


Figure 6(c). Points and edges of face: TOP of the GEHAUSE

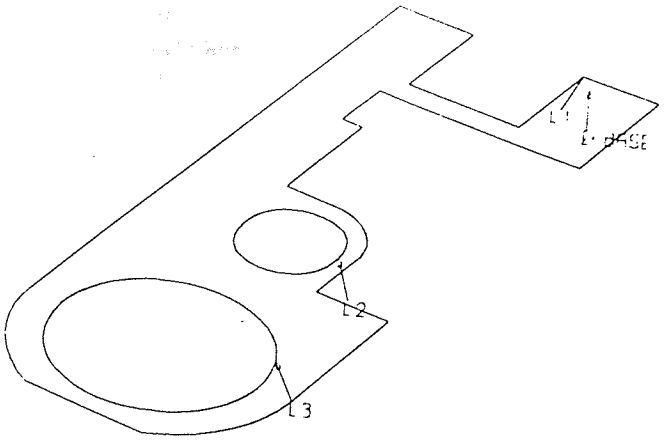


Figure 7(a). Loops of face: BASE of the GEHAUSE

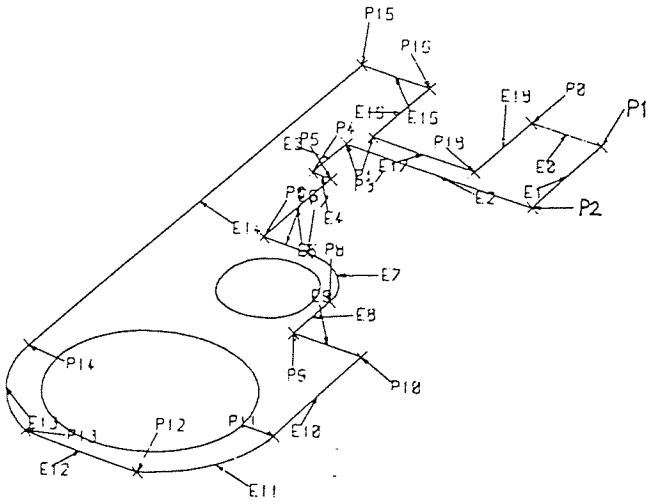


Figure 7(b). Points and edges of face: BASE of the GEHAUSE

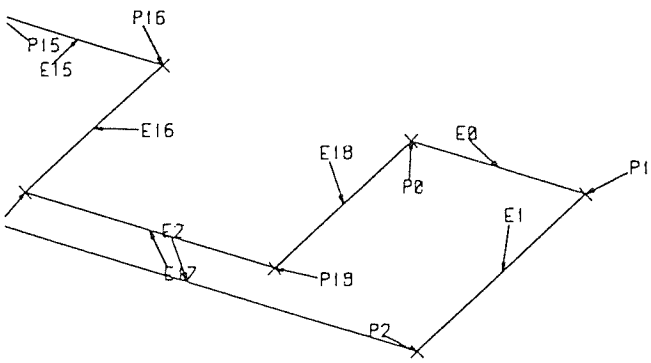


Figure 7(c). Points and edges of face: BASE (enlarged via window)

the PRIME implementation. These factors account for the occurrence of rounding errors in the sixth places after the decimal (Figure 5(*b*)). Fortunately, these rounding errors are usually acceptable for machining.

Circle and ellipse

Both circle and ellipse are stored internally with references to their defining plane and cylinder. If the component is

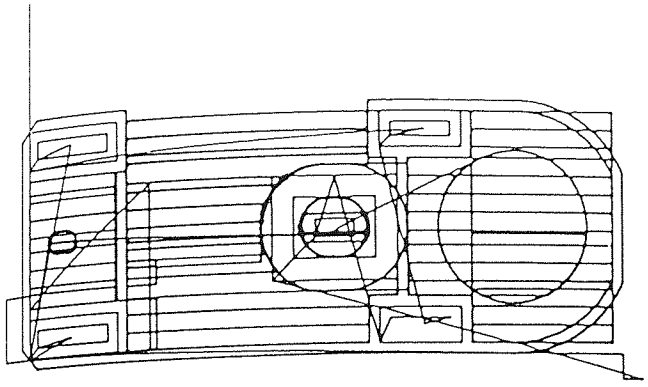


Figure 8(a). Cutter path generated for the GEHAUSE in x-y plane

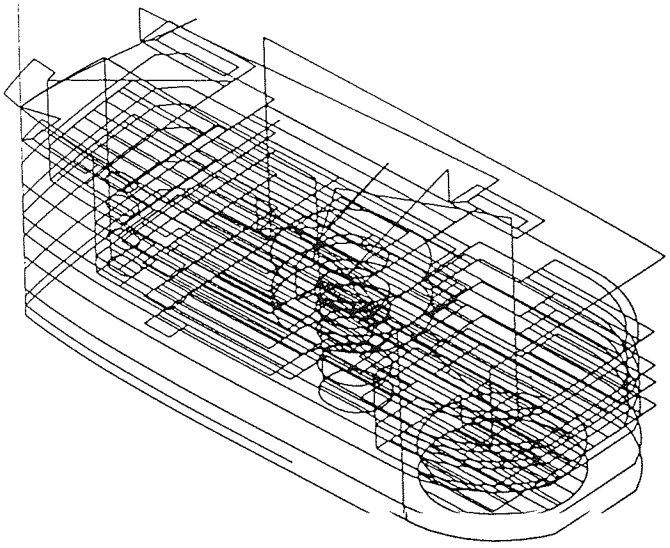


Figure 8(b). Perspective view of cutter path generated for the GEHAUSE

tilted, these references will remain the same and the circle remains unchanged. However, for machining, if the machining angle is not adjusted accordingly when the component is tilted, then the cutter path for a circle becomes an ellipse.

FUTURE WORK

The following facilities are currently under development or are planned:

- to convert labels of names generated by the ROMAPT system to APT conventions:

  P for point
  L for straight line
  C for circle
  PL for plane

- to provide facilities to process construction geometries[9,10]
- to provide further aids for tool path generation

CONCLUSION

The ROMAPT system is easy to use and has been well received.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to the directors of Lucas Industries Ltd in allowing him to publish this paper. He also wishes to thank his colleagues in Lucas for their help in this project.

REFERENCES

1 Bezier, P *Numerical control mathematics and applications* John Wiley and Sons (1970)

2 Chan, Y K and Knight, W A 'MODCON: a system for the CAM of dies and moulds' University of Birmingham, UK (1979)

3 Cox, M D and Snead, J C 'A description of surface generation facilities provided by the SURFSET suite of programs' in *Proc. CAD78* IPC Science and Technology Press, Guildford, UK (1978) pp 387–394

4 Braid, I *Designing with volumes* PhD thesis, Cambridge University, UK (1973)

5 Minutes of Geometric Modelling Interest Group, 6–8 April 1976, m-76-gmig-05, CAM-I (1976)

6 Grayer, A R 'Alternative approaches in geometric modelling' *Comput.-Aided Des.* Vol 12 No 4 (July 1980)

7 'Special Projects' PR-78-ASPP-02, CAM-I (1979)

8 *SYSTEM/370 APT-AC Numerical Control Processor Program Reference Manual* Vol 1, PROGRAM 5740-M53, IBM

9 *Romulus Users' Guide: version 2* Shape Data Limited, Cambridge, UK (July 1979)

10 *Romulus Users' Guide: version 3* Shape Data Limited, Cambridge, UK (1980)

11 Chan, B T F *ROMAPT Version 1.1 User's manual* Lucas Research Centre (1981)

Finite Elements and Related Areas of Computer-Aided Engineering

J.E. MOTTERSHEAD, B.T.F. CHAN and M.P. KELLMAN
Lucas Research Centre, Solihull, England

The past fifteen years have seen an explosion in the use of numerical methods, such as finite elements and turnkey CAD systems, in engineering design centres. Rapid development is currently taking place in second generation CAD systems utilising geometric modellers and sculptured surface packages which can be linked to finite element models and other previously separate areas of computer application such as N.C. machining. The strategy of this paper is to discuss some such systems which are just emerging from the development phase at Lucas Research Centre and to show some typical applications.

In the area of engineering analysis the Lucas Research Centre's FELSET* finite element system[1] is typically used in the stress and vibration analysis of diesel fuel pump and nozzle components, in the analysis of voltage fields in motor car batteries and in magnetostatic field calculations for electrical machines[2]. Recent developments include a finite element formulation for helical springs which is based on the 'exact' differential equations of an infinitesimal element of helical wire[3,4] and allows the investigation of travelling stress waves and static and dynamic buckling of springs.

The FELSET system of computer programs is a combination of in-house software developments and the BERSAFE[5] analysis programs from the Central Electricity Generating Board. The system is deliberately product oriented and this philosophy is shared with other areas of engineering computing.

An early scheme aimed at applying computer aided design and manufacturing (CAD/CAM) techniques to automotive lamp developments resulted in the SURFSET* suite of computer graphics programs[6], which provided facilities for design and manufacture of complex surface shapes, known as sculptured surfaces. A special feature of the suite was the facility for producing a foam model of a component using an N.C milling machine.

The SURFSET programs were based on the use of Bezier patches and have been superseded by the PROTEUS system (after the Greek demigod who had the ability to assume many different shapes) which is based on B-spline mathematics. Complex surfaces are easier to design using B-splines; enabling local surface modifications whilst not affecting the rest of the surface.

Associated programs are capable of generating finite element meshes from sculptured surfaces or calculating cutter path data for N.C machines. The combined facilities of the FELTSET and PROTEUS programs allow several alternative approaches. For example, a sculptured surface, such as that of a motor car panel, may be interactively designed using the PROTEUS system so that it is an aesthetically pleasing or satisfactory aerodynamic shape. Subsequently it can be checked strengthwise using finite elements; furthermore 3-dimensional models of the surface can be obtained using the Lucas N.C model-making milling machine. If the design does not have acceptable mechanical characteristics, suitable design changes can be implemented and assessed quickly. One way of doing this is to retain the surface design but vary the material thicknesses throughout in order to produce a more acceptable stress distribution. Alternatively the sculptured surface could be redesigned and the mechanical properties re-assessed using finite elements.

Systems for computer aided design and production of mechanical parts have a shared interest in the geometry (or shape) of components. When these are composed of a combination of simple shapes (primitives), such as cylinders, spheres and planes, geometric modelling techniques (such as the program ROMULUS[7] from the Shape Data Ltd) may be used successfully for the representation and manipulation of 'solid' components. Geometric modellers such as ROMULUS are able to recognise surfaces and can discriminate between internal volumes and points outside the model, consequently they are often called 'solid modellers'.

Such facilities are of great advantage in connection with N.C machining programs which require information concerning the geometry of the finished part and additional geometrical data concering cutter paths.

The traditional method of producing

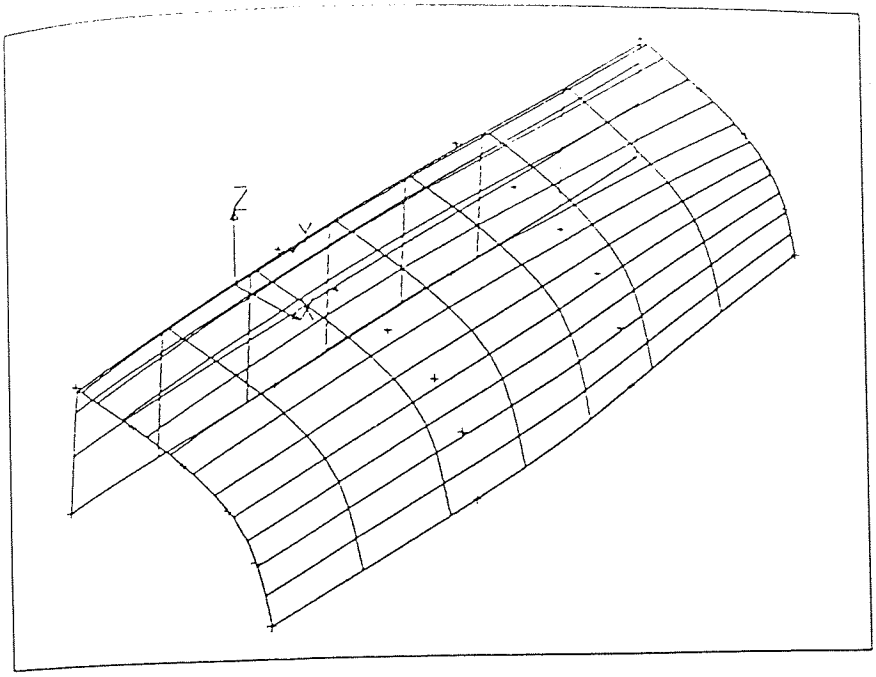


Fig. 1 B-spline Model of a Lamp Surface.

N.C tapes is by means of an N.C program which is hand written from engineering drawings. The N.C program produces standard Cutter Location DATA (CLDATA) using the internationally recognised program, APT (Automatically Programmed Tool), and generates an N.C tape via a postprocessor. The nature of the postprocessor is dependent upon the N.C machine tool since machine tool manufacturers have their own machine languages.

Current turnkey CAD/CAM systems, however, offer links between computer generated (wireframe) drawings and N.C machining. Within Lucas this has been taken further by linking geometric modellers and sculptured surfaces to the APT system. The user may choose to generate a complete N.C program from a geometric model or alternatively the N.C program may be partly hand written and partly generated automatically.

Designing with Sculptured Surfaces

The first step is to produce a set of points which lie on the surface. Curves are then automatically constructed through these points and modifications can occur until an acceptable shape has been created. The surface is derived automatically from these defining curves and all work is conducted at a graphics terminal so that the designer is interacting with a pictorial representation of the shape, not just a numerical one.

The computer surface model can be used to produce:

1) Information for use in a finite element analysis of the surface (automatic mesh generation).
2) Information directly related to manufacture of the surface in the form of cutter paths for an N.C machine.
3) Full engineering drawings via a link with a computer draughting system.

The first example, which is illustrated in Figure 1, shows a lamp surface with the defining points marked with a cross. Figure 2 shows the cutter path generated for this surface with a 1-inch ball-ended cutting tool. A model of the cutter tool can be superimposed at any point and this is used to check for interference problems arising between the cutter path and other parts of the surface being machined.

It is useful to produce a model of the surface in a soft material such as foam or wood as a final aid in verifying the accuracy of the surface definition and the Lucas N.C model-making machine has been used for this purpose. If the foam model is correct an injection moulding tool (for lamps) can be produced in a harder material, such as aluminium or steel, using the same cutter path data. The cutter path can be converted into APT statements or may be used directly, via a postprocessor, with most conventional N.C and C.N.C. machine tools.

The second example shows that the system is not restricted to the design and manufacture of car lamp surfaces but can be applied to almost any other surface form. In Figure 3 the body of the telephone is modelled as a single B-spline surface, the handset as 3 joined surfaces and the dial as a single separate surface. The local modification facility in the B-spline surface definition was used to form the recess for the handset.

Stress Analysis of Sculptured Surfaces

An example is now discussed concerning finite element stress analysis of sculptured surfaces. It was required to design a bottle having an attractive shape, a specified internal volume and

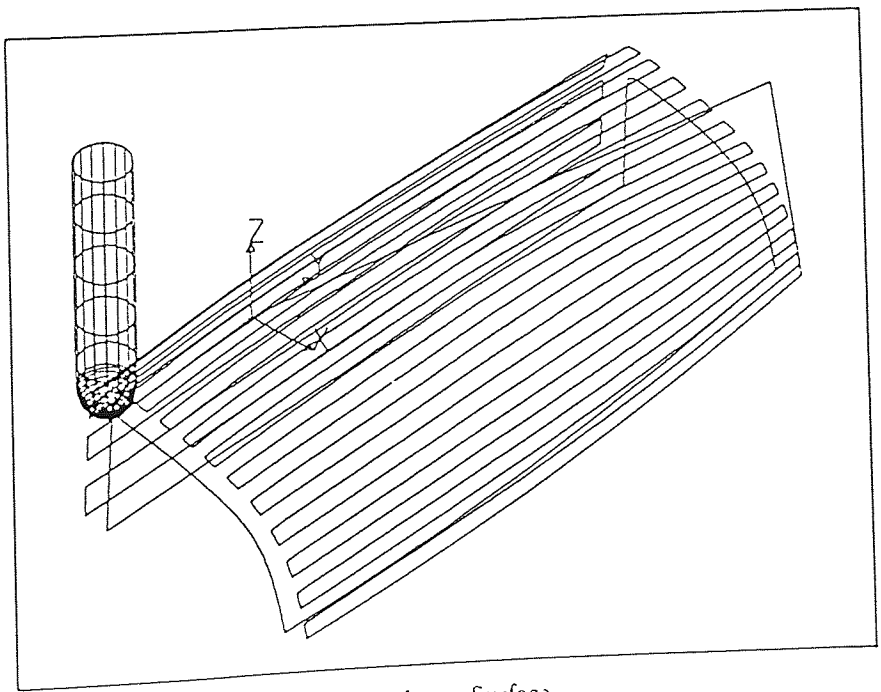


Fig. 2 Machine Cutter Paths for a Lamp Surface.

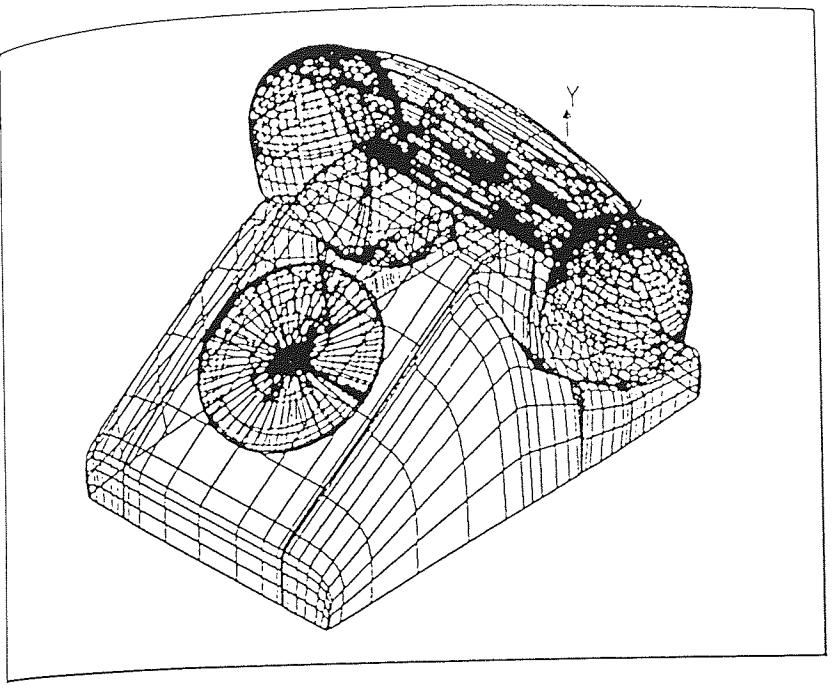


Fig. 3 B-spline Model of a Telephone.

adequate mechanical strength. The bottle was subject to both an internal pressure load and compressive end-stacking loads.

is constructed from Irons semi-loop shell elements and constant thickness was defined for the walls of the bottle.

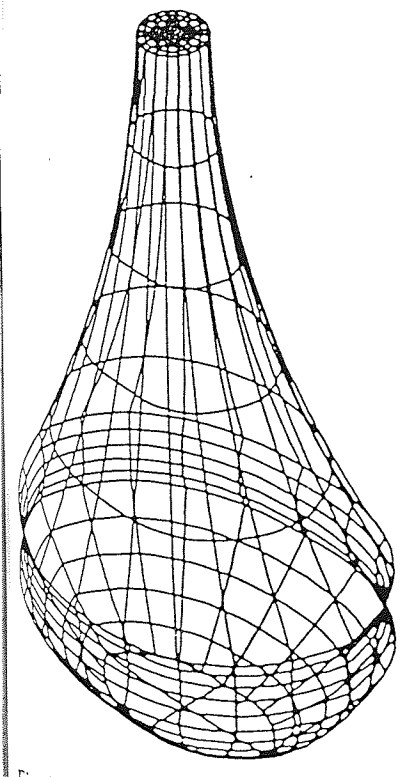


Fig. 4 B-spline Model of a bottle.

It was decided that the container should be symmetrical so that data was only required for a quadrant, further quadrants being constructed by transformation of co-ordinate data. The full computer representation of the sculptured surface is shown in Figure 4 and the automatically generated finite element mesh is illustrated for a quadrant in Figure 5. The mesh

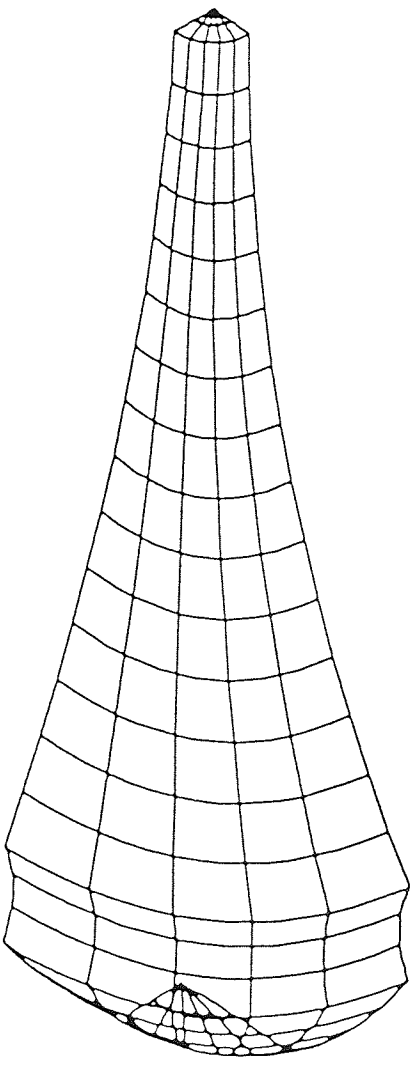


Fig. 5 Finite Element Mesh of a bottle.

Having generated the finite element mesh, all at the graphics terminal, loading and fixing conditions and the material properties were edited into the finite-element dataset. Special interactive graphics programs allow checking of loading and fixing conditions prior to submitting the stress analysis run.

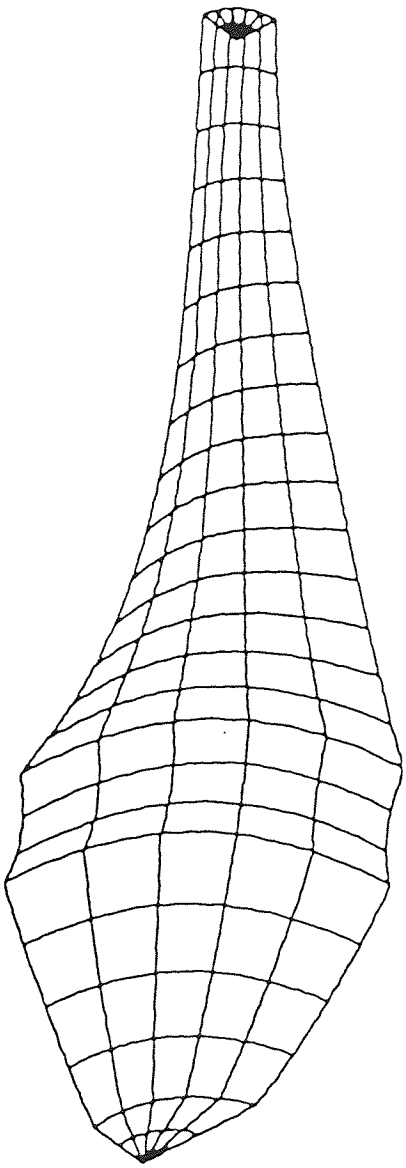


Fig. 6 Deformed Mesh Plot.

The finite element analysis was performed using the BERSAFE stress analysis program (incorporated within the FELSET suite of finite element programs) and the results produced were used to generate deformed mesh plots and stress contour plots, as illustrated in Figures 6 and 7. The results can thus be used for comparison with basic material testing data to assess the mechanical performance of the design or for comparison purposes with existing designs.

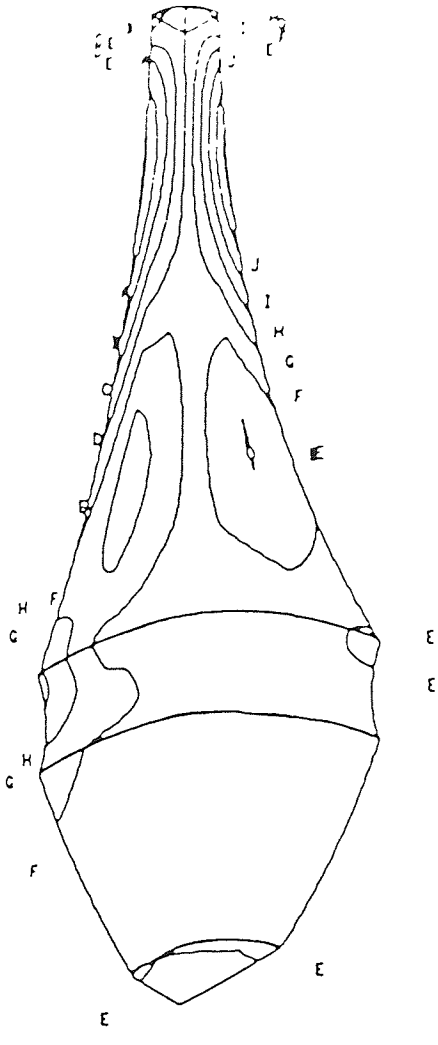


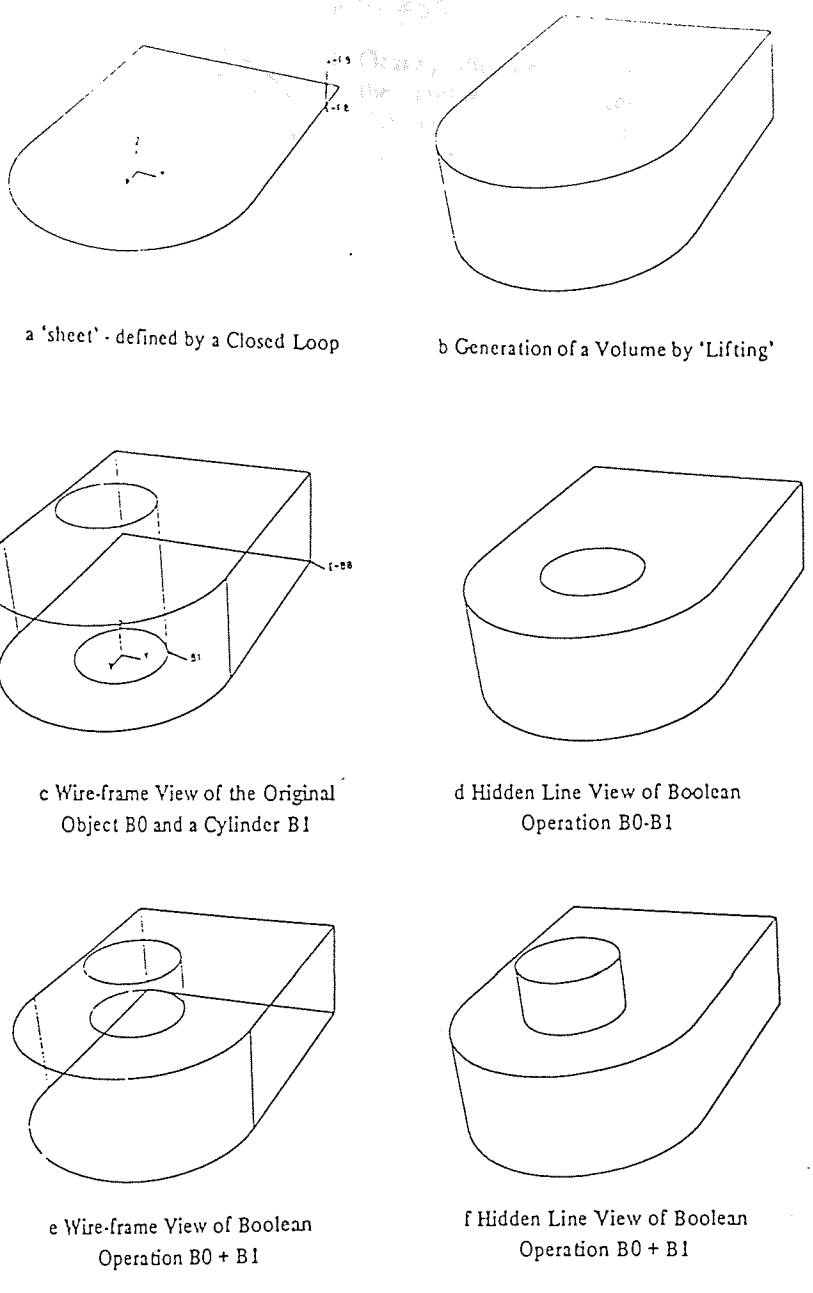
Fig. 7 Stress Contour Plot.

Designing with Geometric Models

The philosophy for the generation of 'solid' objects using the ROMULUS geometric modeller is fundamentally different from that applying to sculptured surfaces. Figure 8 illustrates some of the basic operations of shape manipulation which are available in the program.

At the basic level, various edges (straight to circular) can be constructed, one at a time, to form a closed loop to define a 'sheet' consisting of two faces – this is shown in Figure 8a. In Figure 8b a face has been 'lifted' to provide an object with positive volume.

At a higher level holes and islands can be added using the Boolean operations 'subtract' and 'unite' in order to generate the more sophisticated geometries which are illustrated in Figures 8c – 8f.



a 'sheet' - defined by a Closed Loop

b Generation of a Volume by 'Lifting'

c Wire-frame View of the Original Object B0 and a Cylinder B1

d Hidden Line View of Boolean Operation B0-B1

e Wire-frame View of Boolean Operation B0 + B1

f Hidden Line View of Boolean Operation B0 + B1

Fig. 8 Generation of a 'Solid' Model.

Design and Manufacture of an Alternative End Bracket

Having produced a geometric model of the alternator end bracket isometric

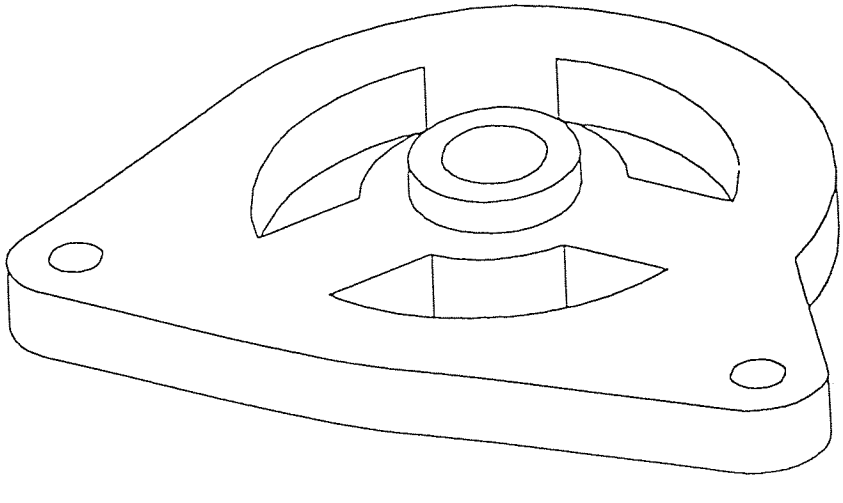


Fig. 9 Isometric View of an Alternator End Bracket.

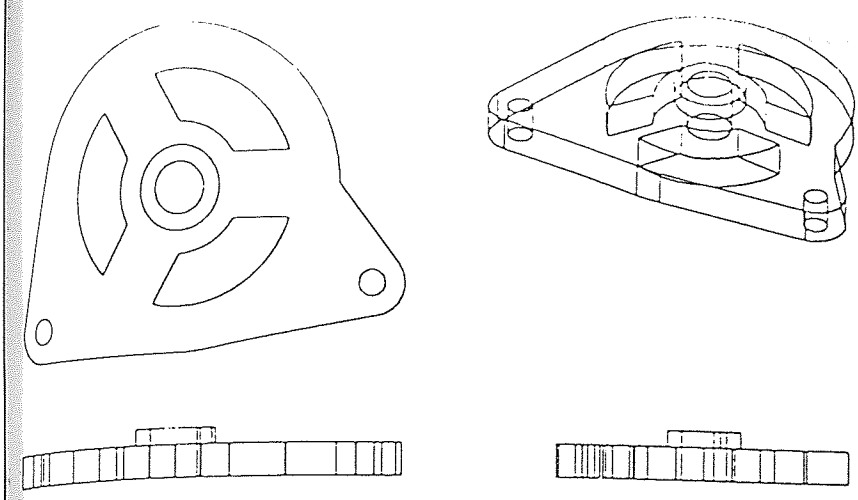


Fig. 10 Four views from the Geometric Model of an Alternator End Bracket.

hidden line plots (Figure 9) and engineering drawings (Figure 10) were obtained automatically from the computer.

Such drawings are useful to the N.C programmer but an interactive program ROMAPT[8] has been developed specially to link geometric models with APT automatically. The program is structured so that the N.C. programmer can extract geometric information at various levels from the geometric model or he can generate APT geometry statements automatically.

An important feature of the ROMAPT system enables labels to be attached to geometric entities which may then be displayed. Figure 11 shows the bottom face of the alternator end bracket which was given the label 'BASE'. Further labels are shown in Figures 12 and 13 which were selected by the N.C programmer in the preparation of N.C tapes for machining. In Figure 12 labels beginning with the letter 'P' indicate points and in Figure 13 letters 'L' and 'C' are associated with labels for straight lines and circles.

The labels help the N.C. programmer to understand the geometry of the component and APT geometry information may be output selectively and interactively in this context. The APT geometry of the perimeter loop of the bottom face (BASE) of the alternator end bracket is shown in Figure 14 where the labels are now associated with geometry in a cartesian system. The N.C programmer may choose to automatically generate the complete set of APT geometry statements or he may use the override facility in ROMAPT and enter some of the APT geometry manually.

The N.C program is completed when the programmer adds the necessary machining instructions to the generated APT geometry statements. Subsequently N.C tapes for machining are produced when the N.C program has been processed by APT and the appropriate postprocessor.

In the case of the alternator end bracket the machining strategy consists of:—

1) Clearing the area and machining the profile of the component. The generated cutter paths are shown in Figure 15.

2) Drilling the holes and machining the various pockets. Figure 16 shows the generated cutter paths.

A further interesting link with the geometric model enables the generation of finite element meshes. In Figure 17 a two-dimensional finite element mesh comprising isoparametric plane-stress quadrilaterals and triangles is shown; and illustrated in Figure 17 is a three-dimensional mesh using isoparametric block and wedge finite elements.

Discussion

The authors have outlined some advances in the area of computer aided geometrical design which create links between apparently separate areas of computer aided engineering.

Stress analysis, an area of design which was affected earliest by the impact of numerical computing, has

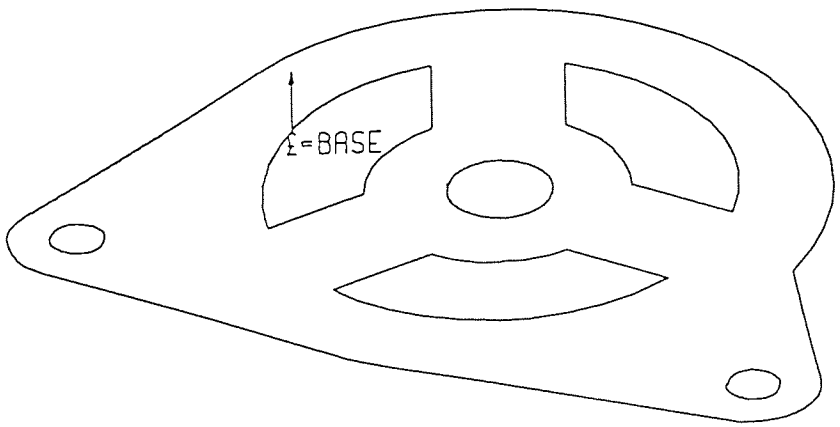


Fig. 11 Bottom Face of an Alternator End Bracket.

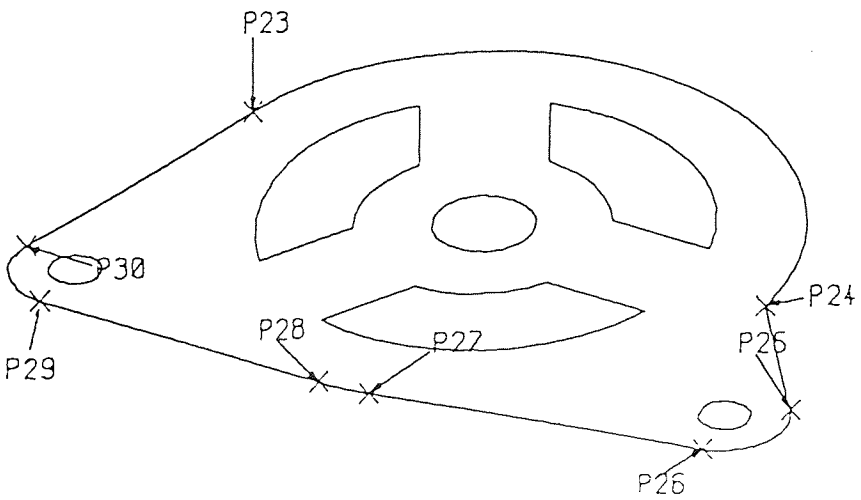


Fig. 12 Labels of Points on the Perimeter Loop of BASE.

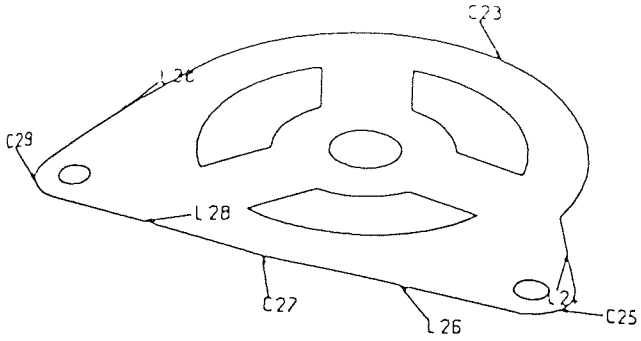


Fig. 13 Labels of Edges on the Perimeter Loop of BASE.

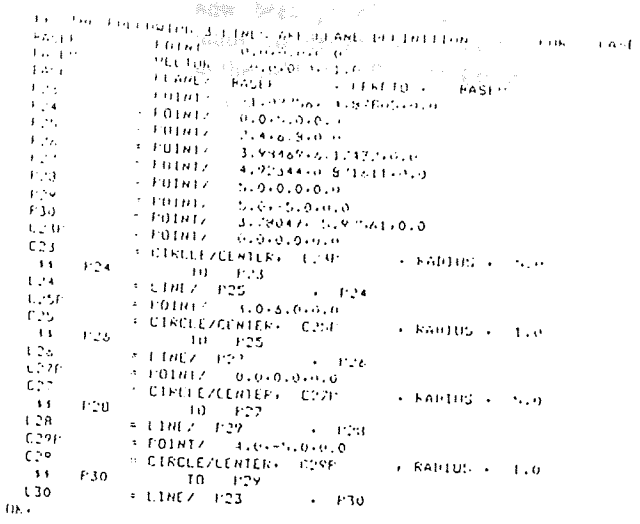


Fig. 14 APT Geometry Statements for the Perimeter Loop of BASE.

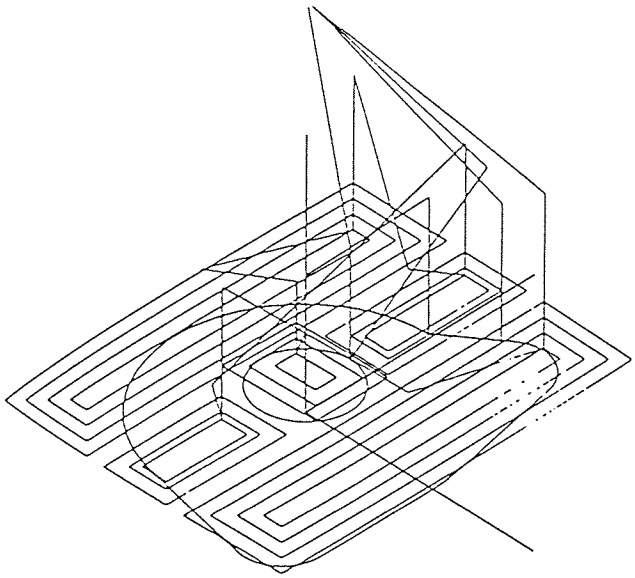


Fig. 15 Cutter Path for Clearing and Profiling.

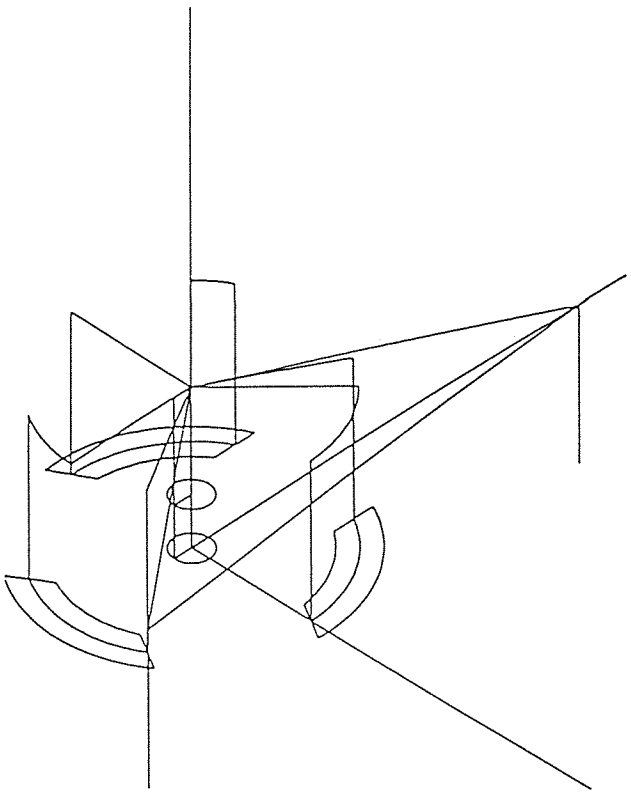


Fig. 16 Cutter Path for Holes and Pockets.

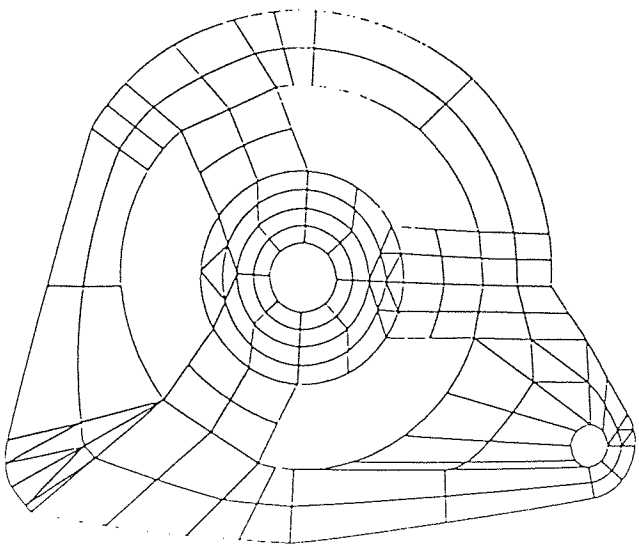


Fig. 17 Two Dimensional Finite Element Mesh.

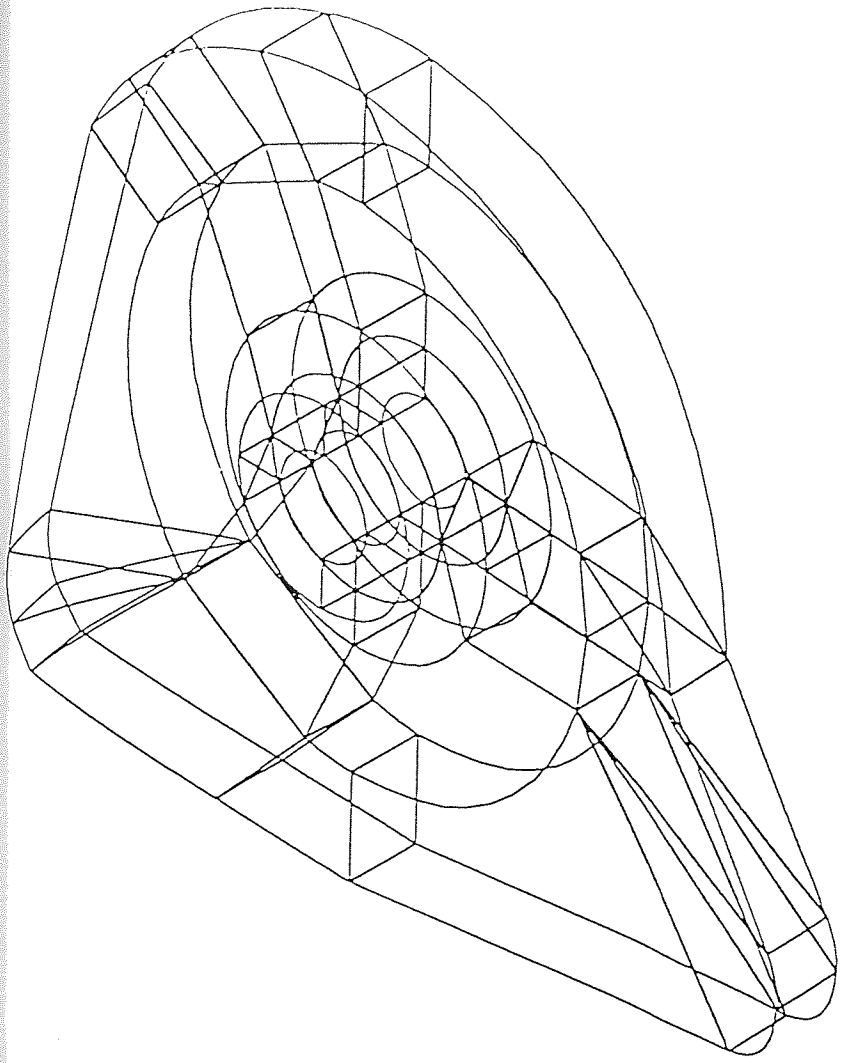


Fig. 18 Three Dimensional Finite Element Mesh.

now been joined by all aspects of industrial design and manufacturing in the field of CAD/CAM. Consequently system architectures involving finite elements, sculptured surfaces, N.C machining and other methods for design and manufacture are being developed rapidly as industry and science recognise the economic benefits which are bound to follow.

References

1. P.R. Wilson. 'Engineering Analysis System FELSET Phase 2 User Manual', Private Communication.
2. D.B. Greenow. 'A computer package for the solution of electromagnetic field problems', Advances in Engng. Software, 1, 125-130 (1979).
3) J.E. Mottershead. 'Finite elements for dynamical analysis of helical rods', Int. J. Mech. Sci., 22, 267-283 (1980).
4) J.E. Mottershead. 'The large displacements and dynamic stability of springs using helical finite elements', Int. J. Mech. Sci., To be published.
5) T.K. Hellen and S.J. Prothero, 'The BERSAFE finite element system', CAD, 6, 15-24 (1974).
6) J.C. Snead. 'Recent developments and applications of the SURFSET suite for computer aided design and manufacture of sculptured surfaces'. Proc. 2nd Int. Conf. on Engng. Software, Imperial College, London. 247-261, (1981).
7) P. Veeman, 'ROMULUS – The design of a geometric modeller', CAM-I Geometric Modelling Seminar, Palace Court Hotel, Bournemouth, November (1979).
8) B.T.F. Chan. 'ROMAPT: A new link between CAD and CAM', CAD, To be published.

**FELSET and SURFSET are registered trademarks of Lucas Industries p.l.c.*

CAM-I COMPUTER AIDED MANUFACTURING - INTERNATIONAL, INC.

SECOND

CAM-I

GEOMETRIC MODELLING SEMINAR

ROBINSON COLLEGE,
CAMBRIDGE, ENGLAND.

12TH-14TH DECEMBER 1983

P-83-GM-01

VOL.1

Editor: Dr. I.D. Faux
Director of European Operations
CAM-I

COMPUTER AIDED MANUFACTURING INTERNATIONAL INC.

NC MACHINING DEVELOPMENTS AT LUCAS BASED ON A GEOMETRIC PART MODEL

© Benny T. F. Chan

Lucas Research Centre,

Lucas Group Services Ltd.

ABSTRACT

This paper describes a recently developed CAD/CAM interface system (ROMAPT) in the Lucas Research Centre.

The ROMAPT system provides a vital link between a bounded 3D geometric modeller (ROMULUS) and an unbounded numerical control (NC) processor (APT).

ROMAPT can now provide a fast NC tape generating facility for engineering part prototypes. It is a system for three dimensional NC applications.

It is envisaged that the 3D shape of the part will be designed by using the ROMULUS geometric modelling system. The APT part program for the NC machining of the component can then be generated by using ROMAPT, and finally the NC tape can be generated by processing the NC part program in the APT or SSR system.

1. INTRODUCTION

A new generation of industrial geometry systems is just emerging. Unlike most of the current commercially available systems, the new systems are based on unambiguous 3D solid geometric modellers and they are potentially capable of supporting current and future applications automatically.

ROMULUS is a commercially available 3D geometric modeller. It is installed in the Lucas Research Centre as a means to evaluate the above concept.

The use of numerically controlled machine tools is now firmly established within manufacturing industry. Design and manufacturing systems based on an increased armoury of mathematical techniques for manipulating shapes by computer will greatly increase the use and effectiveness of numerically controlled machine tools from now on.

One aspect of this armoury, namely the advent of geometric modelling techniques for the description and manipulation of 3-D shapes, should further this effectiveness. It is essential therefore, to establish a link between geometric modelling and NC machine tools to provide an integrated computer-aided-design and manufacturing system.

However most geometric modelling systems are "bounded" in a mathematical sense - i.e. lines and planes are finite, whereas the controlling computer language for many NC machine tools, namely APT, is "unbounded" - i.e. lines and planes are infinite. APT systems, however, are presently only linked to wire-frame drafting systems. The combination of a geometric modeller and APT will provide a powerful manufacturing system for industry from the initial design right through part manufacture using NC machines. This research work investigates the theory, develops the concept and algorithms, and the design of an experimental software interface (ROMAPT) between a bounded geometric modeller (ROMULUS of Shape Data Limited) and an unbounded NC processor (APT) (1,2).

APT is used because of its popularity in industry. ROMULUS is used because it represents the state of the art of solid modelling. (ROMULUS is based on the considerable research work of Dr. Ian Braid's BUILD system at Cambridge University.) (3,4,5)

2. ROMAPT DESIGN CONCEPT

The most straight-forward approach is to design a bounded NC processor to interface with a bounded geometric modeller. However, the manufacturing industry has invested substantially over the years in APT and similar NC processors. It is very unlikely that the industry would be willing to shed this investment instantly. The American National Stardard Institue, ANSI, has set 1983 as the target date for the launch of a revised APT language standard to

cater for the additional requirement of intelligent CNC, robot, and inspection machines etc. It, therefore, appears that the revised APT language will become even more widely used.

Moreover, the current state of development of geometric modelling is such that a universally accepted bounded geometric modeller standard does not yet exist. Hence,a universal bounded NC processor has yet to be developed. Currently, CAM-I is addressing this problem.

The investment needed to develop N bounded NC processors for N bounded geometric modellers would be excessive. The alternative approach is to design a link between a state of the art bounded geometric modeller and a popular unbounded NC processor like APT. Using this approach, the theory, concept and algorithms required for the link can be readily tested, verified and refined. The theory and algorithms developed can then be used in future to develop a modeller-independent link between solid modelling and APT. The geometric modeller will be used to provide data for NC calculation and to resolve ambiguities in APT processing, i.e. to provide APT with bounded geometry intelligence to overcome some unreliability problems in APT.

This amounts to developing a technique of binding the unbounded APT system to the bounded ROMULUS solid modeller.

3. APPROACHES TO NC MACHINING DEVELOPMENT

The following NC machining development approaches based on a geometric part model have been investigated.

Automatic APT Geometry Generation

1) ROMAPT (Stage 1)

This is a software link between ROMULUS and APT to generate APT geometry statements automatically.

2) AIAPT

This is a modeller-independent implementaion of ROMAPT (stage 1) using the CAM-I Application Interface subroutine specifications (6).

Automatic NC Cutter Path Generation

3) Bounded Cutter Path Generator

This is an automatic cutter location generator for a geometric

model.

4) ROMAPT (Stage 2)

This is an extension of ROMAPT (stage 1) to provide automatic generation of APT motion (cutter path) statements.

4. ROMAPT (Stage 1)

To allow for the diversity of possible manufacturing processes and machining strategies that may be used, an interface can be designed to output APT geometric source statements. In this case the output is readable to the NC programmer; he can check and edit, if required, the APT source to suit the task. Conceptually, this approach is very straight-forward and easy to use and was adopted with the following objectives:

1. Ease of use.

2. Enabling the NC programmer to understand the geometric model better.

3. Providing APT geometric source statements as required.

Within manufacturing industries, there is a substantial investment in APT and with it a concomitant pool of NC programming expertise. With due consideration for the above mentioned factors, the ROMAPT system was designed to convert bounded geometric information held in the geometric modeller to standard unbounded APT geometry statements. The experience and expertise of the NC programmer is then used to complete the NC part program. This further allows the NC programmer to concentrate on the machining strategy rather than the geometry.

When this approach (stage 1) had been proven, then more automation (stage 2) in the form of automatic cutter path generation was developed for the system.

4.1 ROMAPT SYSTEM CAPABILITY (STAGE 1)

Traditionally, a NC programmer is given a set of engineering drawings of the part he wishes to make. He has to understand and interpret the drawings accurately before he proceeds to write his NC part program. This is where much time is consumed and many errors are possible.

Working with a geometric modeller, in addition to the set of engineering drawings given, will help to reduce the above mentioned drawbacks. However, the NC programmer would need to have some facilities to enable him to query and understand the part given in the form of a geometric model defined by a data structure (Figure

1).

4.2 GRAPHIC MANIPULATION

Supplementary to the manipulation facilities (i.e. rotation , viewing) provided by ROMULUS, further manipulating facilities have been developed to help a user's understanding of the geometric model. All these additional facilities will help a user to generate APT geometry statements for a component. These facilities are described in the following:

COMMAND	FUNCTION
1. VIEW	Facility to see the object via a display screen.
2. ROTATE	Facility to rotate the object at will.
3. SCALE	Facility to scale the object.
4. PROJECT	Facility to have views of third angle projections similar to a conventional engineering drawing
5. GENERATE	Generates name for unnamed geometric entities (points, edges, faces). naming conventions: P : point E : edge F : face
6. CANCEL	Cancels name(s) of geometric entities (point, edge, face).
7. LABEL	Labels name(s) of required geometric entities (point, edge, face) on a display drawing.
8. ENQUIRE	Outputs required geometric information of the model of the part.
9. SHIFT	Shifts the picture on display in X and Y direction.
10. WINDOW	Provides windowing facilities on current view. So that a more complex geometric model can be handled.

4.3 APT GEOMETRY STATEMENT GENERATION

The ROMAPT program scans through the data structure held in the geometric model in a systematic manner: firstly the body, then the faces followed by the points and edges. This geometric data (bounded) is then converted into standard APT geometry format (unbounded). The APT geometric statements for a component are

output accordingly. The components are processed individually. For each component (i.e. a body), the APT geometric definition for every face will be output and its associated points and lines will be output accordingly in a systematic manner.

Standard APT-AC/APT4/SSR1 (7,8) geometric definitions are employed as in the following.

1) POINT

P1 = POINT/ x, y, z

The co-ordinates of point P1 are given in (x, y, z).

2) EDGE

E1 = LINE/ P1 , P2

E1 is a straight line defined between point P1 and point P2.

E2P = POINT / x , y , z

E2 = CIRCLE/ CENTER , E2P , RADIUS , t

$$ P1 TO P2

E2 is a circular arc / circle defined between point P1 and point P2 with centre at E2P (x, y, z) and radius of t units.

E3 = ELLIPS/ INTOF , F1 , F3

$$ P3 TO P4

E3 is an ellipse between point P3 and point P4 and is defined by the intersection of plane F1 and cylinder F3.

3) FACE

F1P = POINT/ x , y , z

F1V = VECTOR/ u , v , w

F1 = PLANE/ F1P , PERPTO , F1V

F1 is a plane defined by a point on the plane at F1P (x, y, z) and the normal to the plane: vector F1V (u, v, w).

F3P = POINT/ x , y , z

F3V = VECTOR/ u , v , w

F3 = CYLNDR/ F3P , F3V , t

F3 is a cylinder defined by a point F3P (x, y, z) on the cylinder axis and a vector F3V (u, v, w) along the cylinder axis and the radius (t) of the cylinder.

F4P = POINT/ x , y , z

F4V = VECTOR/ u , v , w

F4 = CONE/ F4P , F4V , d

F4 is a cone defined by the vertex point F4P (x, y, z), the axis vector F4V (u, v, w) and the half-angle (d) of the vertex of the cone.

F5P = POINT/ x , y , z

F5 = SPHERE/ CENTER , F5P , RADIUS , r

F5 is a sphere defined by the center F5P (x, y, z) and the radius (r).

F6P = POINT/ x , y , z

F6V = VECTOR/ u , v , w

F6 = TORUS/ F6P , F6V , r1 , r2

F6 is a torus defined by the center point F6P (x, y, z), the axis vector F6V (u, v, w), the major radius (r1) and the minor radius (r2).

As there is no formal definition for torus in APT4/SSR1, this has to be defined indirectly as a surface of revolution.

Only the most elementary forms of geometry definitions (i.e. the lowest common demoninator) of the APT family is used to allow maximum transportability among many different APT processors.

4.4 ROMAPT (STAGE 1): MODE OF OPERATION

The ROMAPT interface consists mainly of 3 commands in 3 different modes of operations.

I) APTALL (dump mode)

Generates APT geometric statements for all geometric entities of the body concerned.

II) APTOPTIMAL (optimised mode)

Generates an optimal set of APT geometric statements (mainly for 2 1/2 D parts).

III) APTENQUIRE (interactive mode)

Generates APT statements for requested geometric entities interactively to enable a NC programmer to select the minimum set of APT geometric statements for the component concerned.

APTENQUIRE can be used to pinpoint the particular entities required, thus an even smaller set of geometric statements can be obtained by the NC programmer.

4.5 WORKED EXAMPLE

To prove the validity of the above concept, a worked example of a NC part program was carried out using the APT geometry generated by ROMAPT on an artifical mechanical part Gehause (Figures 2a, 2b) designed by using ROMULUS.

The generation of the APT geometry for the Gehause is shown in Figures 3a, 3b.

After the geometry of the part has been generated, the NC programmer completes the NC program by adding the necessary APT machining instructions. The APT part program is then processed by APT and a post-processor to generate a NC tape. The cutter path is shown in Figure 4, and the part has been machined successfully on a Lucas Model Making Machine.

Thus, it can be concluded that the foundamental concept behind ROMAPT i.e. the binding the unbounded geometry of APT via a bounded 3D solid geometric modeller (ROMULUS) is valid.

5. AIAPT

Although ROMAPT is a working system, it suffers the inconvenience of only functioning within the ROMULUS modeller. In order to have a modeller-independent version of ROMAPT, the concept of AIAPT is described in the following.

The Geometric Modelling Project of CAM-I have developed specifications for a set of FORTRAN subroutines to act as an Application Interface (AI) to geometric modellers. This specification is available as CAM-I Report R-80-GM-04 and Addenda. The interface effectively shields application programs from an underlying geometric modeller. AIAPT will perform the same functions as ROMAPT but instead of calling ROMULUS routines calls AI routines instead, thus freeing the program from any modeller dependancies. It is possible for AIAPT to work with any geometric modeller that supports a boundary description provided that there is an implementation of the AI available for the modeller.

6. NC AUTOMATION (ROMAPT STAGE 2)

The previous sections have established a valid link between a geometric modeller and APT at the geometry level. The following describes how this link has been extended to provide automatic generation of NC cutter path.

6.1 CONCEPT OF MACHINING MODEL

As geometric modelling was first developed, it was charged with the objective to represent the geometrical aspect of the design intent of the designer. Little or no provision was made for the fact that the geometric model thus created has to be manufactured using certain manufacturing process. (Some may argue that this should be the function of application software for manufacturing.) Perhaps this is one of the reasons why 3D solid modelling, inspite of its enormous potentials, has not been much widely used in industry for production.

With the current state of the art of geometric modelling, the modeller would be unable to inform the designer that a certain design or construct is not manufacturable (for certain manufacturing process) or machinable (for certain machining conditions or tools). It is the task of the designer/production engineer to ensure that a certain geometric model is suitable for manufacture. Hence, the concept of designing a machining model that is machinable (i.e. design for manufacture).

6.2 MACHINING FEATURES

The following attempts to identify whether a model or part of it is machinable or not.

The machining model, MM, is defined as a geometric model made up of a number of machining features. A machining feature is defined as an entity of interest for numerical control. Various machining features are described in the following:

1) POINT feature

The point feature relates to point-to-point operation of NC machining usually connected with hole drilling or tapping.

2) PROFILE feature

The profile feature relates to the boundary of a face, which is usually made up of a number of edges bounded in an ordered manner (i.e. a loop entity with a bounded face). This feature usually associated with the profiling operation of NC machining.

3) FACE feature

This feature relates to a bounded region (e.g. a face) of a surface. This feature usually relates to the pocketing facility of NC machining.

A machinable model can then be defined as a geometric model made up of a number of machinable features, Fm.

$$MM = \sum_j Fmi$$

6.3 THEORY OF AUTOMATIC CUTTER PATH GENERATION

As formal theory for computer aided automatic machining of 3D solid geometric model has not yet existed and as reported work in this area is nearly non-existing, the author has no choice but to propose his own theory. The theory has been developed for boundary representation solid modeller. The current trend is such that constructive solid modeller (CSG) will provide boundary representation, so that theory developed here will be applicable for both types of geometric modellers.

Details of the above theory are given in another report (B. T. F. CHAN, 'Computer Aided Machining of Geometric Models', Oct. 1983. (9)).

The next section describes a practical implementation based on this theory.

7. BOUNDED NC CUTTER PATH GENERATOR

The following describes the concept of a bounded NC cutter path generator based on a geometric part model.

7.1. BOUNDED VS UNBOUNDED NC PROCESSORS

Existing APT based NC systems are based on the philosophy that the direction and the position of the tool is known and the problem is to 'find' the part. With bounded geometry description of the part, the problem of the NC processor becomes reversed: the description of the part is known and the problem is one of finding the position of the tool and its direction to generate the tool path required for the part and process involved.

7.2. CONCEPT OF OFFSET

On a closer examination, the cutter path required for the bounded NC cutter path generator is the cutter offset of the part concerned. In the case of a profile feature, the cutter path is the cutter offset of a loop of a face. For a face feature, the cutter path is the cutter offset of the face.

As for milling operations, the machining requirement usually consisted of a combination of profiling and pocket milling.

7.3 AUTOMATIC CUTTER PATH GENERATOR FOR 2D AND 2 1/2D PARTS

Using the above concept, the automatic cutter path generator for 2D and 2 1/2D parts has been developed. To demonstrate the applicability of the above theory, an engineering part has been created by using ROMULUS and then the geometric model is machined by using ROMAPT to generate the cutter path automatically.

Figures 5, 6 show the geometric model of the part.

The demonstration of the automatic generation of the cutter path is described in the following.

7.3.1 MACHINING OF A LOOP (PROFILE FEATURE)

The outer profile of the part can be considered to be made of 4 edges: E7, E4, E21 and E18 as shown in Figure 7.

The cutter path for the outer profile can be generated automatically by the command:

PROFILE [edge1, edge2...] [LOOP] [tool] [radius] [options]

where,

[edge]: name of edges.

[LOOP]: options for the whole loop.

[tool]: options for either flat-end or ball-end cutter.

[radius]: radius of the cutter.

[options]: either the current selected face or user-defined Z-planes.

The cutter path for cutter radius of 4.0 mm is shown in Figure 8.

7.3.2 MACHINING OF FACE (FACE FEATURE)

The cutter path for the machining of the top surface, F1 of the part can be generated by the command:

MACHINE [face] [edge] [tool] [radius] [options]

where,

[face]: name of face.

[edge]: name of the edge indicating the cutting direction.

[tool]: options for flat-end or ball-end cutter.

[radius]: radius of the cutter.

[options]: options for the cutter to cut TO, ON, PAST.

The cutter path generated for F1 is shown in Figure 9.

The output of the above automatic cutter path generator is a cutter location file containing the centre of the cutter in x, y, z co-ordinates (Figure 10). This can be further processed to output NC tape:

1) The cutter location file can be post-processed to output a NC tape for a particular NC machine. In this case, the Lucas Model Making Machine is used.

2) The cutter location file can also be translated into an APT part program consisting of a set of simple APT point-to-point statements:

GO TO / x, y, z

This part program can then be processed in the usual manner to produce a NC tape.

Although the above approach works effectively for 2D and 2 1/2 D applications, for 3D applications, it is likely to have the following disadvantages:

1) As the output is a cutter location file, it is difficult to understand and debug. The NC programmer cannot check conveniently whether the program has performed correctly or not.

2) The part program thus generated tends to be much larger than that generated by manual APT programming.

3) To develope the system to a production grade system for tooling would require a substantial investment especially in machining technology and post-processing development.

For 2D and 3D applications, the following approach was adopted.

8. ROMAPT (STAGE 2)

This is an extension of the ROMAPT (stage 1) concept. Instead of outputing the cutter locations, the corresponding APT motion statements (for cutter path generation) are output.

Together with the APT geometry generation capability and some auxiliary facilities, a complete APT part program can be generated using ROMAPT.

8.1 IMPLEMENTATION

It can be seen clearly that by using the above mentioned concepts, an automatic NC cutter path generator can be developed for any type of machining: drilling, turning and milling.

This research addresses the problem of automatic milling of 3D geometric models. It is expected that theories and techniques developed for milling, especially the automatic profiling and regional milling can be applied to turning as well. It also assumes that much of the technology for cutting is provided via a proccess planning module situated between the geometric modeller and the automatic NC generator.

To ease the task of software implementation, the algorithms were initially developed for 2 1/2D applications where a large number of engineering part can be represented. When the agorithms had been proven in 2 1/2D parts, then they were extended for the full 3D applications.

8.2 DEVELOPMENT AND RESULTS

The main effort of this research lies on the development of the above mentioned theory and the implementation of the software (ROMAPT).

The following describes the automatic profiling and pocketing facilities in ROMAPT.

8.2.1 GENERATION OF APT MOTION STATEMENTS

The relationship (topology) of various geometric entities: points, edges and faces of the outer profile of the part is shown in Figure 11.

The data structure required for the generation of APT motion statements can be set up either manually or automatically:

CNC [LOOP] [option 1] [option 2] [option 3]

where,

[LOOP]: is the key word for the generation of the data structure for the loop.

[option 1]: indicates whether the loop is OPEN or CLOSE.

[option 2]: indicates whether the cutter is situated IN, ON, OUT of the loop.

[option 3]: indicates whether the MANUAL or AUTOMATIC mode is required.

The manual method allows more flexibility (to suit the technological requirement of the machining). The manual method is shown in Figure 12. The data structure generated is shown in Figure 13. The automatic generation of the required APT motion statement is effected by the command:

CNC [APTLOOP] [options]

where,

[APTLOOP]: is a keyword for the generation of the APT motion statment for the loop.

[option]: define either the currently selected face or user-defined Z-planes.

The corresponding APT motion statements is shown in Figure 14. (This assumes the required geometry statements have been generated previously by using the APTENQUIRE command (10).)

The cutter path verification for the outer profile is shown in Figures 15a and 15b.

8.3 MACHINING OF THE INNER PROFILE

The relationship (topology) of various geometrical entities: points, edges and faces of the inner profile is shown in Figure 16a. The cutter path for the inner profile is generated (for a cutter radius of 6.35 mm) as shown in Figure 16b.

ROMAPT can also recommended a maximum cutter size for certain critical areas (i.e. passing the gap between loops) as shown in Figure 17.

8.3.1 GENERATION OF APT MOTION STATEMENTS

The data structure (intelligent data) required for the generation

of APT motion statement for the whole loop is generated automatically as shown in Figures 18 and 19. The APT motion statements are generated as shown in Figure 20. Similarly, the APT motion statement can be generated for the cylinder. The cutter path is shown in Figure 21. Finally, the task is completed by an area clearance operation.

CNC CLEAR CURSOR

This is followed by the definition of the area by 2 cursor positions (as shown in Figure 22). The cutter path and the APT motion statement generated are shown in Figures 23 and 24.

The cutter path verification for the whole inner profile is shown in Figures 25 and 26.

Note that the machining is in 2 steps each cut to a depth of 5 mm.

8.3.2 AUXILIARY FACILITIES

The header, tolerance, cutter definition and end statements can be generated by ROMAPT as shown in Figure 27.

Finally, post-processor, coolant, spindle and feedrate statments can be inserted to complete the NC program. (Note that the spindle speed and feed rate can be obtained from workshop machining standard tables recommended by NC/tool manufacturers.)

Figure 28 shows the whole cutter path of the part generated automatically by using ROMAPT.

Figure 29 shows a complete APT part program generated by ROMAPT.

9.0 CONCLUSION

It has been demonstrated above that both the theory and algorithms developed by the author for the development of NC automation in 3D solid modelling are both valid and applicable.

ROMAPT allows the potential of a solid geometric modeller (ROMULUS) to be further exploited for NC applications without requiring major investment in new NC processor. ROMAPT supports output in APT-AC, APT4 and the CAM-I SSR languages. Further languages, such as EXAPT, may be easily added, thus providing NC processor independence. Similarly geometric modeller independence has been demonstrated by the use in AIAPT of the CAM-I Application Interface.

REFERENCE:

1. SHAPE DATA LIMITED, Romulus Version 4 Users´ Reference Manual, Shape Data Limited, Cambridge, U.K. (1982).

2. IITRI, APT Computer System Manual, IIT Research Institute, RM-79-APT-01, CAM-I, U.S.A. (1979).

3. BRAID, I. Designing with Volumes, PhD thesis, Cambridge University, U.K. (1973).

4. BRAID, I. ´New Direction in Geometric Modelling´, CAD Group Document No. 98, Computer Laboratory, University of Cambridge, U.K. (March 1978).

5. BRAID, HILLYARD, STROUD, ´Stepwise Construction of Polyhedra in Geometric Modelling´, CAD Group Document No. 100, Computer Laboratory, University of Cambridge, U.K. (October 1978).

6. CAM-I, CAMI Report R-80-GM-04, (1980).

7. IBM, SYSTEM/370 APT-AC Numerical Control Processor Program Reference Manual Vol 1, PROGRAM 5740-M53, IBM.

8. CAM-I, Documentation for Scultured Surfaces Release 1, SSR1, PS-81-SS-01, CAM-I, U.S.A. (1981).

9. B. T. F. CHAN, ´Computer Aided Machining of Geometric Models´, PhD First Year Report, Production Technology & Production Management Department, University of Aston, Birmingham, England. (October 1983).

10. B. T. F. CHAN, ´ROMAPT: A new link between CAD and CAM´, CAD Journal, Vol. 14 No. 5, pp 261 - 266, (September 1982).

FIGURE 1: A boundary object data model

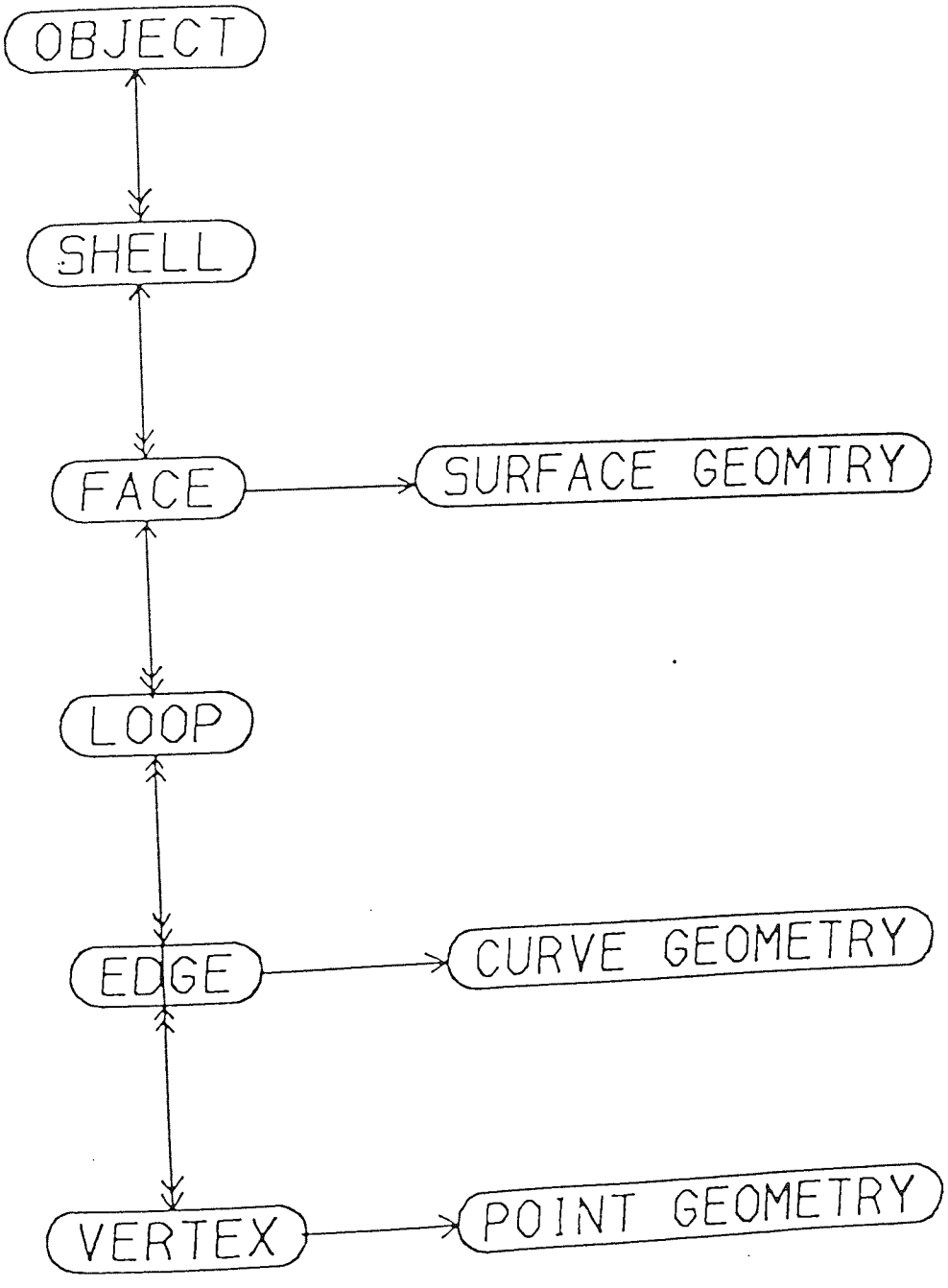


FIGURE 2a: GEHAUSE (hidden line view)

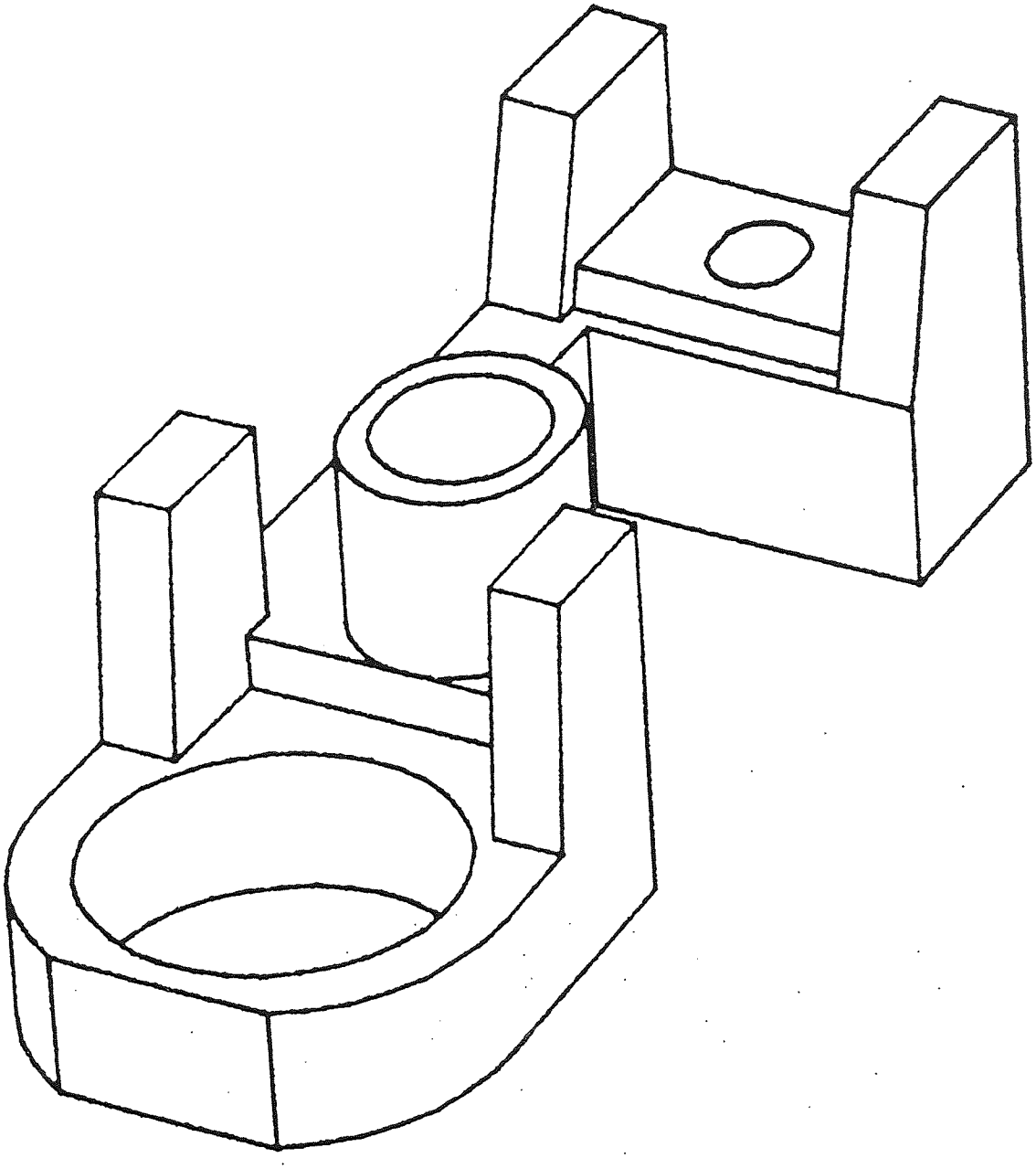


FIGURE 2b: GEHAUSE (orthogonal views)

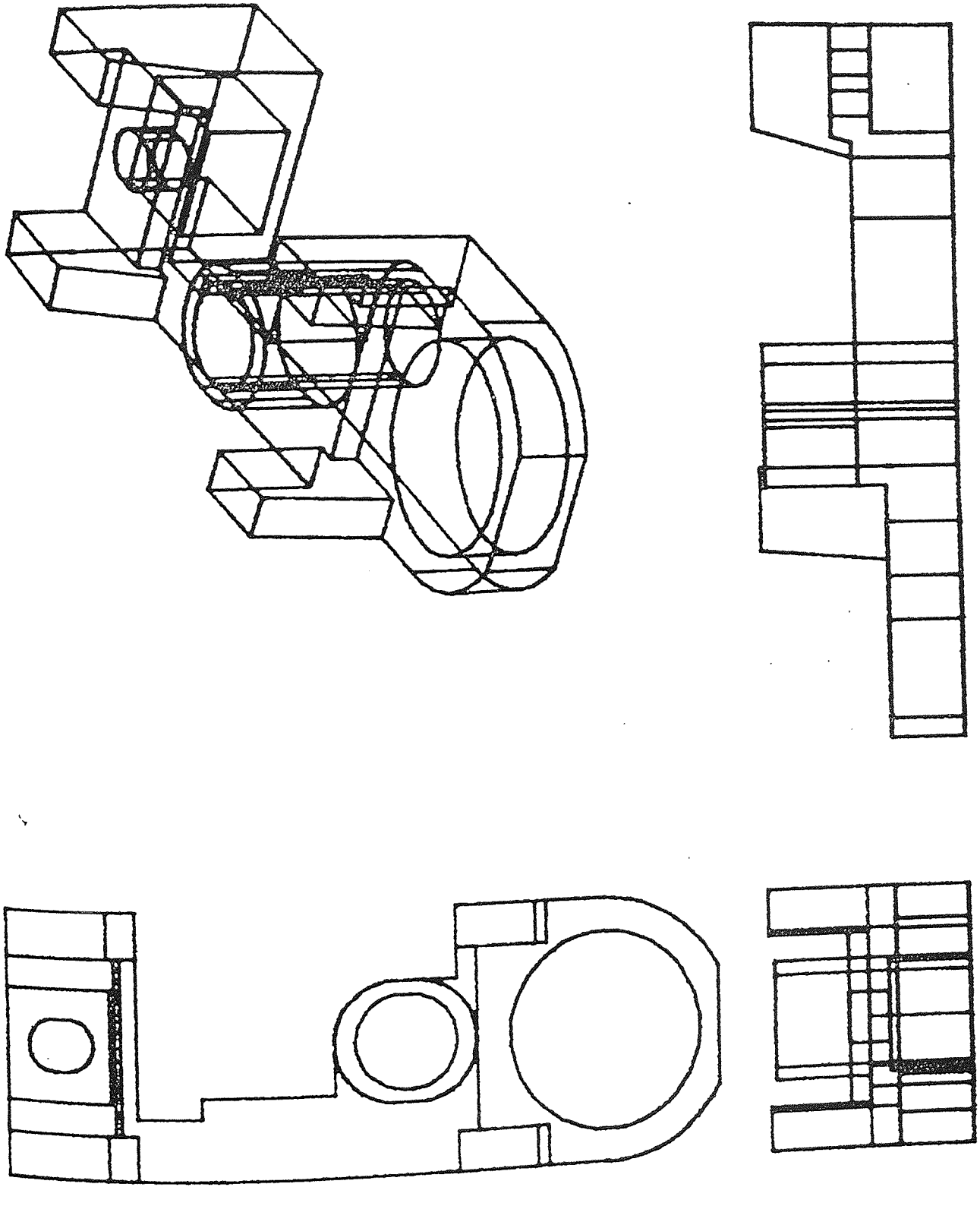


FIGURE 3a. ROMAPT operations

```
$$ MONITOR FILE CREATED  14:21 26 APR 82  BY  PTFC
$$  ROMULUS  VERSION 3.3
 OPTION TERMINAL 4010
$$ ACCESS THE FILE CONTAINING THE GEOMETRIC MODEL : GEHAUSE
 GET ROMTEMP>MODT.G
$$ LIST BODIES
 LIST
$$ GENERATE NAMES FOR POINT/EDGE/FACE OF THE MODEL IF NOT
$$ ALREADY DONE SO BY THE PROGRAM
 GENERATE GEHAUSE
$$ SKETCH A VIEW OF THE BODY
 SKETCH
$$ PLOT A HIDDEN LINE VIEW OF GEHAUSE  FIGURE 4a
 HLPLOT
$$ PLOT A 3D ENGINEERING (ORTHOGONAL) VIEW OF GEHAUSE : FIGURE 4b
 OPTION ORTHO
 SKPLOT
$$ MONO VIEW
 OPTION MONO
 SKETCH
$$ USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
$$ PLOT THE WINDOW VIEW : FIGURE 5a
 SKPLOT
$$ LABEL ENTITIES INTERESTED
 APTLABEL FACE
$$ PLOT THE FACE TOP WITH LOOPS LABELLED : FIGURE 5b
 SEPLOT TOP
 SELECT TOP LOOP
 APTLABEL TOP
$$ PLOT THE FACE TOP WITH POINTS/EDGES ON LOOP 1 : FIGURE 5c
 SEPLOT TOP
 APTLABEL POINT TOP LOOP 1
 APTLABEL EDGE TOP LOOP 1
$$ RELEASE THE WINDOW
 VIEW RELEASE
 SKETCH
$$ SELECT THE BOTTOM FACE BASE
 SELECT BASE
$$ PLOT THE FACE BASE WITH LOOPS LABELLED : FIGURE 6a
 SEPLOT BASE
 SELECT BASE LOOP
 APTLABEL BASE
$$ PLOT THE FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIGURE 6b
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
$$ USE WINDOW TO ENLARGE THE VIEW
 VIEW WINDOW
 SKETCH
$$ PLOT THE ENLARGED FACE BASE WITH POINTS/EDGES ON LOOP 1 : FIG. 6c
 SEPLOT BASE
 APTLABEL POINT BASE LOOP 1
 APTLABEL EDGE BASE LOOP 1
$$ THE ABOVE PROCEDURE CAN BE REPEATED FOR OTHER FACES
$$ UNTIL THE GEOMETRY OF THE PART IS UNDERSTOOD BY THE USER.
$$ WHEN THE GEOMETRY IS UNDERSTOOD , THE USER CAN PROCEED TO.
$$ OUTPUT THE APT GEOMETRY SELECTIVELY AS IN THE FOLLOWING : FIGURE 7b
 OUTPUT APTDATA
 APTENQUIRE BASE
 APTENQUIRE POINT BASE LOOP 1
 APTENQUIRE EDGE BASE LOOP 1
 OUTPUT
$$ STOP KEEP
 OK.
```

FIGURE 3b: ROMAPT geometry statements

```
$$  THE FOLLOWING 3 LINES ARE FRAME DEFINITION       FOR   BASE     ,  ,
BASEP       = POINT/   0.0,0.0,0.0
BASEV       = VECTOR/  0.0,0.0,-1.0
BASE        = PLANE/  BASEP     , PERPTO ,   BASEV
P0          = POINT/   0.0,9.1,0.0
P1          = POINT/   0.0,12.8,0.0
P2          = POINT/   6.4,12.8,0.0
P3          = POINT/   6.4,3.0,0.0
P4          = POINT/   9.39999,3.0,-0.0
P5          = POINT/   9.39999,4.0,-0.0
P6          = POINT/   15.4,4.0,0.0
P7          = POINT/   15.4,6.4,0.0
P8          = POINT/   18.3,9.3,0.000002
P9          = POINT/   21.4,9.3,0.0
P10         = POINT/   21.4,12.8,0.0
P11         = POINT/   28.9,12.8,0.000001
P12         = POINT/   34.6,9.31032,0.0
P13         = POINT/   34.6,3.48967,0.0
P14         = POINT/   28.9,-0.000003,0.000001
P15         = POINT/   0.0,0.0,0.0
P16         = POINT/   0.0,3.7,0.0
P17         = POINT/   5.2,3.7,0.0
P18         = POINT/   5.2,9.1,0.0
E0          = LINE/  P0         ,  P1
E1          = LINE/  P1         ,  P2
E2          = LINE/  P2         ,  P3
E3          = LINE/  P3         ,  P4
E4          = LINE/  P4         ,  P5
E5          = LINE/  P5         ,  P6
E6          = LINE/  P6         ,  P7
E7P         = POINT/   18.3,6.4,0.000002
E7          = CIRCLE/CENTER,  E7P       , RADIUS ,  2.9
  $$   P7          TO    P8
E8          = LINE/  P8         ,  P9
E9          = LINE/  P9         ,  P10
E10         = LINE/  P10        ,  P11
E11P        = POINT/   28.9,6.4,0.000001
E11         = CIRCLE/CENTER,  E11P      , RADIUS ,  6.4
  $$   P11         TO    P12
E12         = LINE/  P12        ,  P13
E13P        = POINT/   28.9,6.4,0.000001
E13         = CIRCLE/CENTER,  E13P      , RADIUS ,  6.4
  $$   P13         TO    P14
E14         = LINE/  P14        ,  P15
E15         = LINE/  P16        ,  P15
E16         = LINE/  P16        ,  P17
E17         = LINE/  P17        ,  P18
E18         = LINE/  P18        ,  P0
OK,
```

FIGURE 4: Perspective view of cutter path generated for the GEHAUSE

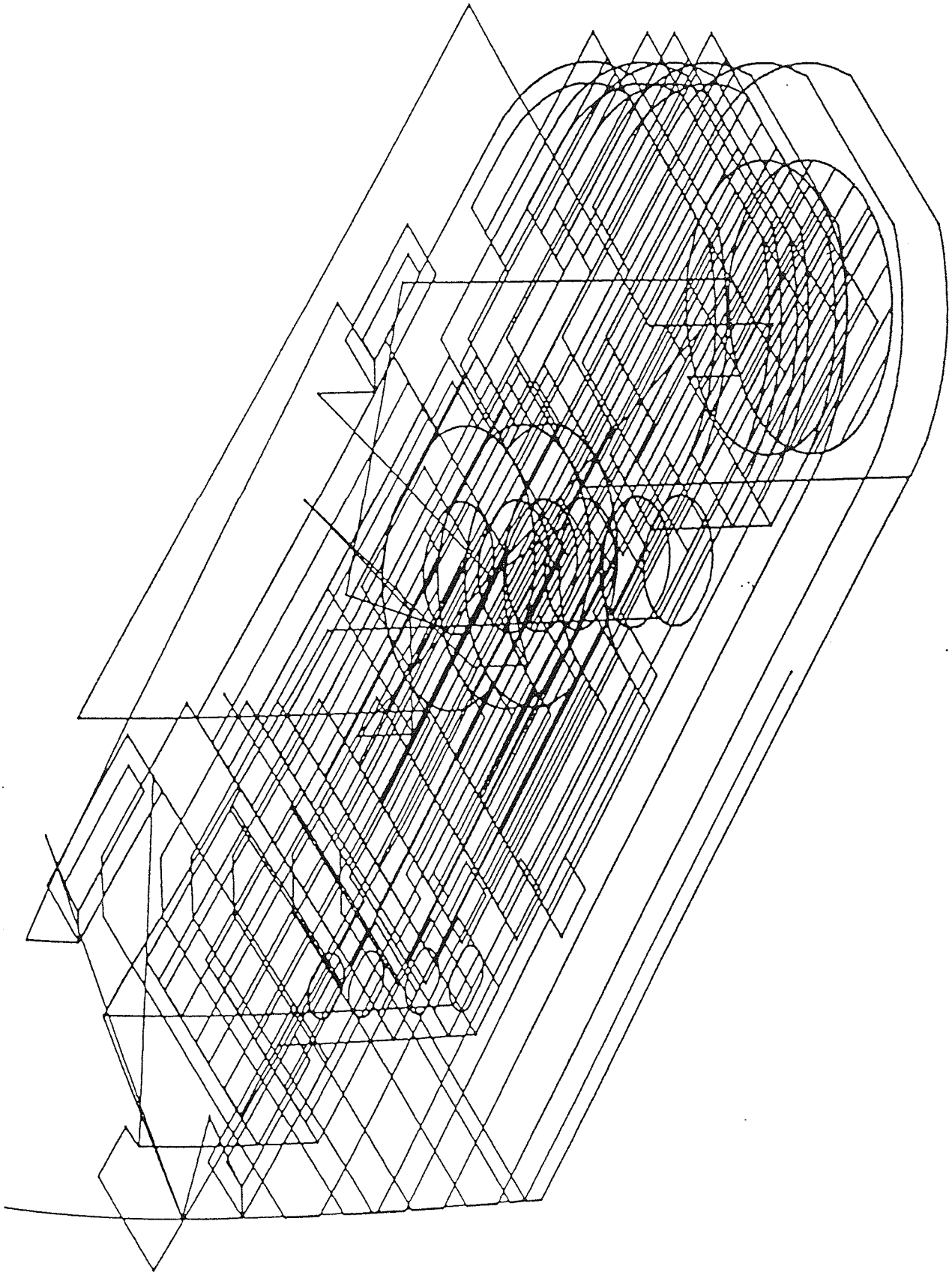


FIGURE 5: Hidden line view of the engineering part

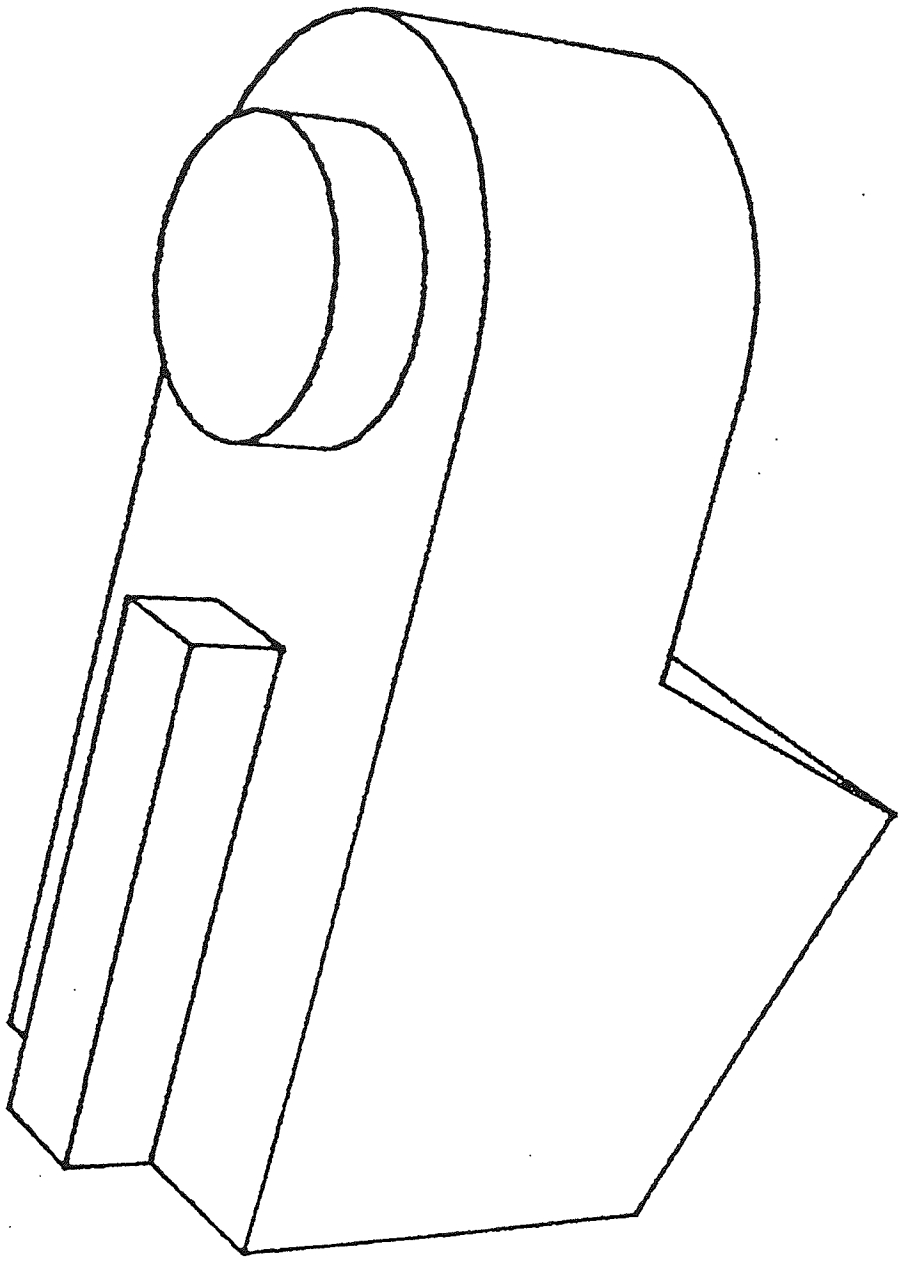


FIGURE 6 Orthogonal wire-frame view projections

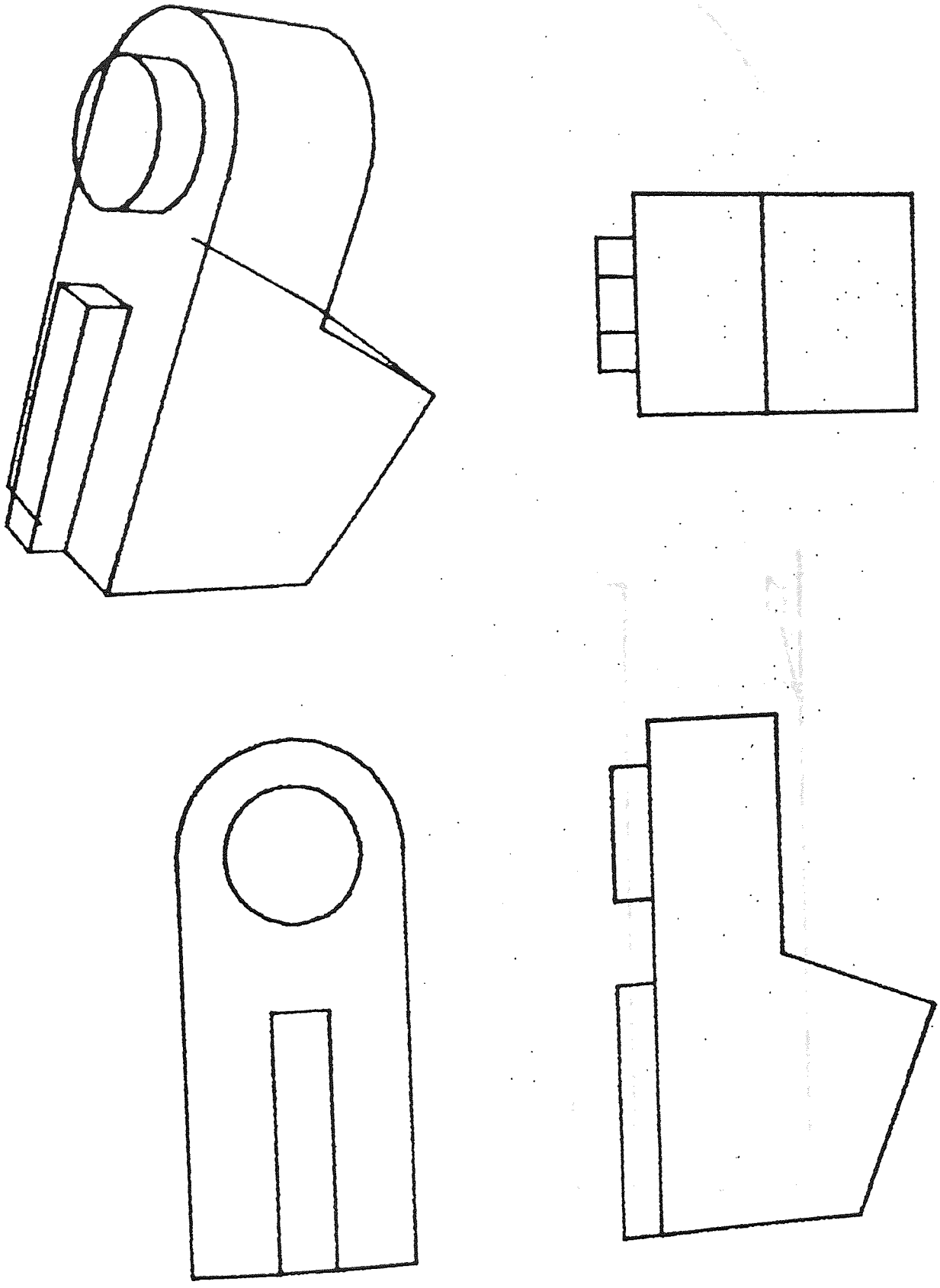
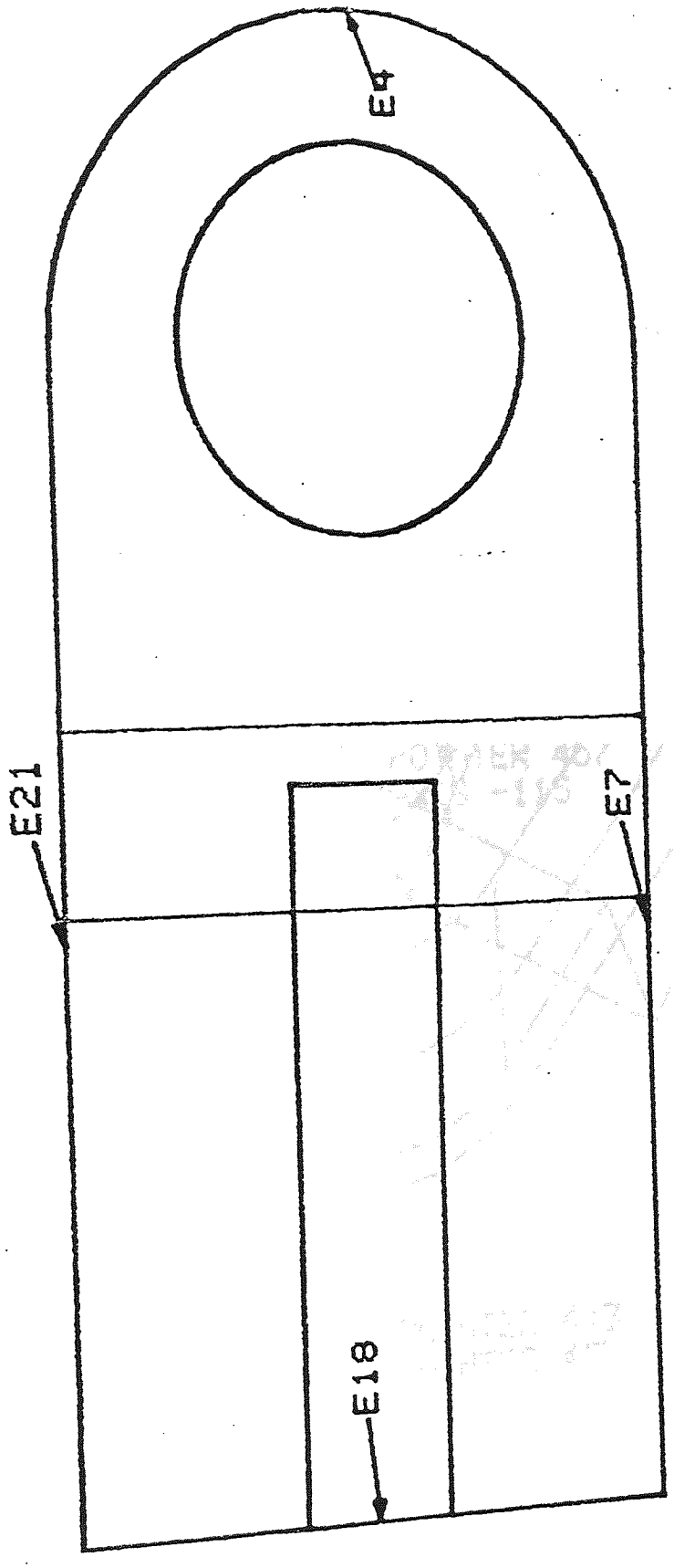


FIGURE 7: The outer profile




```
*APTLABEL E7
*E4
*E21
*E18
*****
```

FIGURE 8 The cutter path for the outer profile for a cutter radius of 4.0 units

```
* PROFILE E7 E4 E21 E18 FLAT 4.0 ZAXIS -93
  PROF FACE : F8
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F9
  PROF INTRK : 1
  PROF INTRK : 1
  PROF CIRCLE : $277
  PROF FACE : F7
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F3
  PROF INTRK : 0
  PROF INTRK : 0
VULNERABLE WORKSPACE AT SLOT 429, POINTER 443
VULNERABLE WORKSPACE AT SLOT 443, POINTER 457
* PROFILE E7 E4 E21 E18 FLAT 4.0 ZAXIS -115
  PROF FACE : F8
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F9
  PROF INTRK : 1
  PROF INTRK : 1
  PROF CIRCLE : $277
  PROF FACE : F7
  PROF INTRK : 1
  PROF INTRK : 1
  PROF FACE : F3
  PROF INTRK : 0
  PROF INTRK : 0
VULNERABLE WORKSPACE AT SLOT 429, POINTER 443
VULNERABLE WORKSPACE AT SLOT 443, POINTER 457
*
```

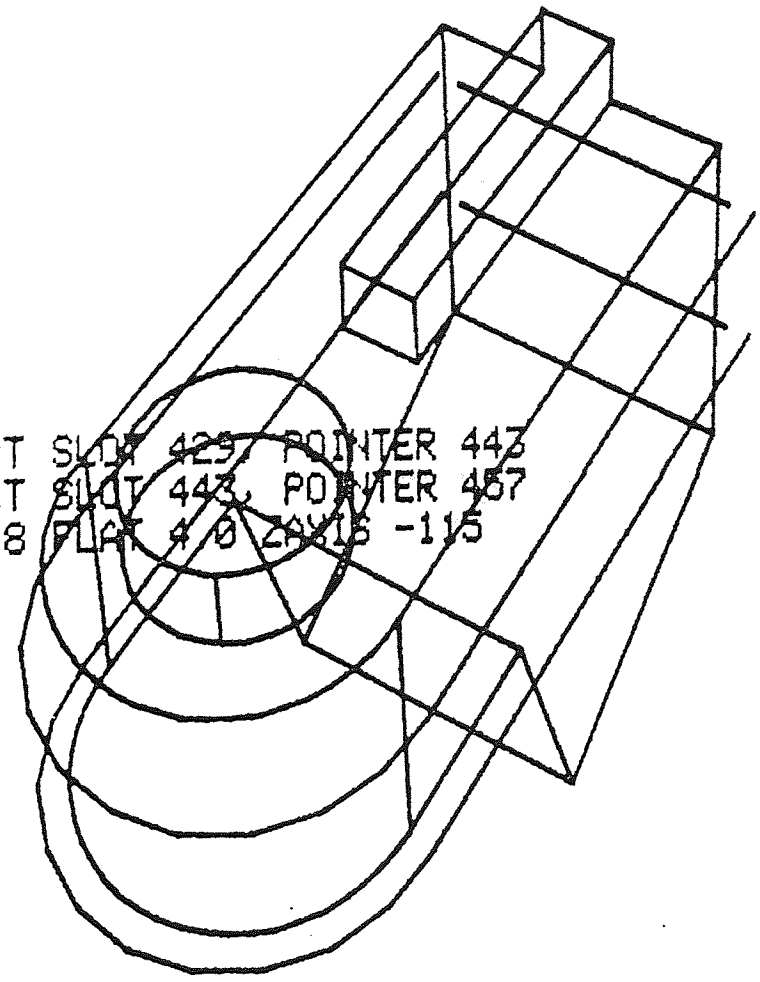


FIGURE 9: The cutter path generated for face: F1


```
* output a.dump
* mach f1 s22 f 4.0 on
  INPUT STARTING POINT
0.0,20
  CUTT3D .   0    0.000
  FAC3D IN .  2
  ISUM .      2
  ISUM .      2
  ISUM .      2
Vulnerable workspace at slot 437, pointer 453
Vulnerable workspace at slot 469, pointer 485
Vulnerable workspace at slot 485, pointer 499
Vulnerable workspace at slot 499, pointer 513
Vulnerable workspace at slot 513, pointer 527
Vulnerable workspace at slot 527, pointer 541
Vulnerable workspace at slot 541, pointer 557
Vulnerable workspace at slot 557, pointer 573
Vulnerable workspace at slot 573, pointer 587
Vulnerable workspace at slot 587, pointer 603
*
```

FIGURE 10: The cutter location co-ordinates

```
MACHINE  CM
FLATEND   0.400000E 01 0.000000E 00 0.000000E 00 0.400000E 01
P  0.2900000E 02  0.9100000E 02 -0.1000000E 02
L  0.2900000E 02 -0.4000000E 01 -0.1000000E 02
L  0.1024999E 02 -0.4000000E 01 -0.1000000E 02
L  0.1024999E 02  0.6099995E 02 -0.1000000E 02
L -0.1025000E 02  0.6099998E 02 -0.1000000E 02
L -0.1025000E 02 -0.4000000E 01 -0.1000000E 02
L -0.2900000E 02 -0.4000000E 01 -0.1000000E 02
L -0.2900000E 02  0.9099995E 02 -0.1000000E 02
L -0.2758068E 02  0.9996135E 02 -0.1000000E 02
L -0.2346157E 02  0.1080457E 03 -0.1000000E 02
L -0.1704586E 02  0.1144614E 03 -0.1000000E 02
L -0.8961578E 01  0.1185806E 03 -0.1000000E 02
L -0.7602492E-04  0.1200000E 03 -0.1000000E 02
L  0.8961432E 01  0.1185807E 03 -0.1000000E 02
L  0.1704573E 02  0.1144615E 03 -0.1000000E 02
L  0.2346147E 02  0.1080458E 03 -0.1000000E 02
L  0.2758064E 02  0.9996150E 02 -0.1000000E 02
L  0.2900000E 02  0.9100000E 02 -0.1000000E 02
L  0.2900000E 02  0.9100000E 02 -0.1000000E 02
 E
```

FIGURE 11: The points and edges of the outer profile

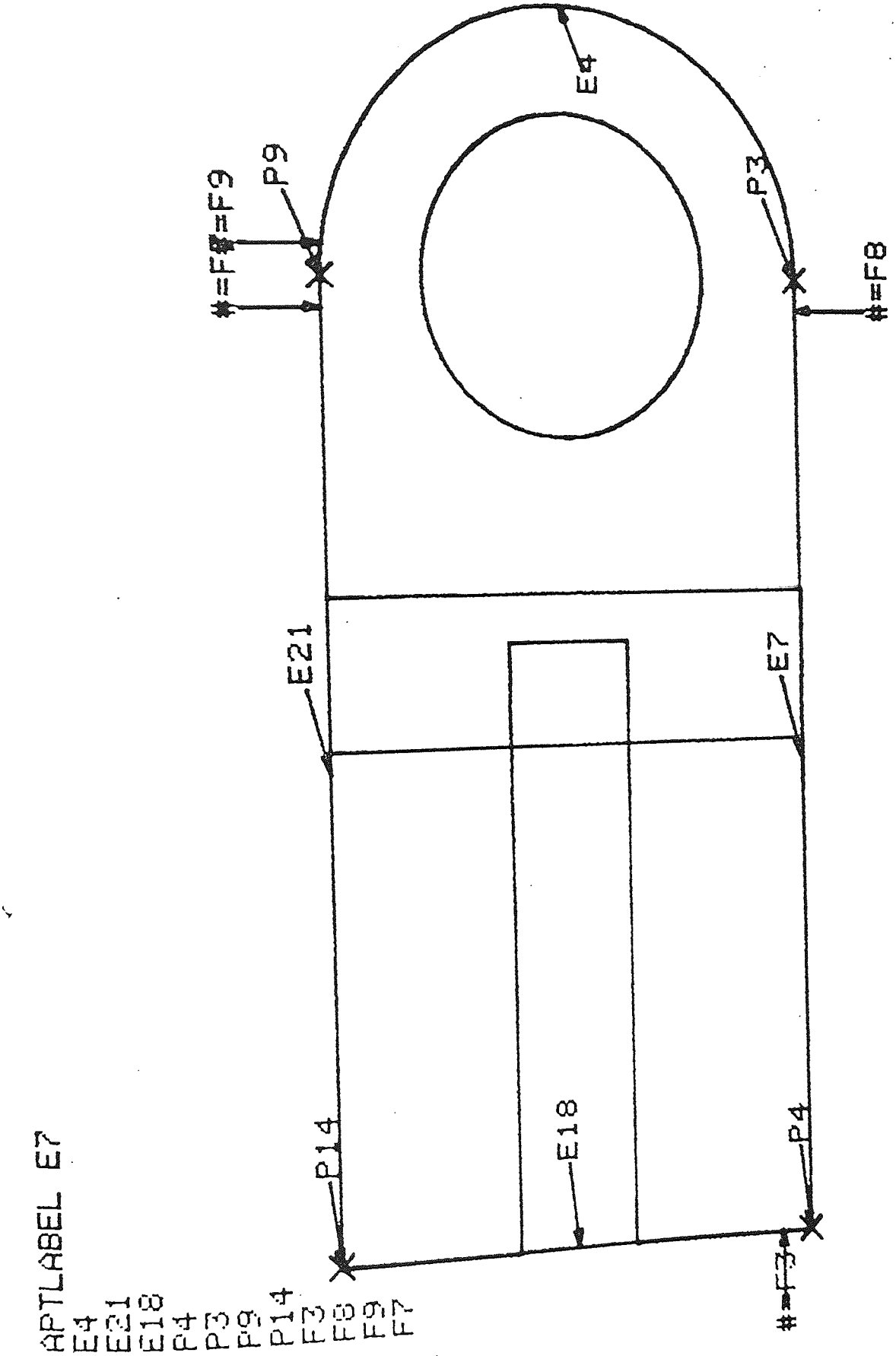


FIGURE 12: The procedure for manual generation of data structure required for the outer profile

```
CNC LOOP CLOSE OUT MANUAL
 INPUT SENSE, GLOBAL RAISE FLAG, NO. OF ITEMS =
1 1 4
 INPUT MIN. DISTANCE, MIN. RADIUS =
1.0,1.0
 INPUT POINTS    4
 (POINT-NAME) ?
+ P4 P3 P9 P14
 INPUT EDGES    4
 (EDGE-NAME) ?
+ E7 E4 E21 E22
 INPUT RAISE FLAG    4
1 1 1 1
 INPUT RAISE FACE    4
 (FACE-NAME) ?
+ F8 F9 F7 F3
 INPUT CONNECTING FLAG    4
1 2 3 1
 INPUT CONNECTING TYPE    4
3 2 2 3
 INPUT DOUBLE FLAG    4
1 1 1 1
 INPUT OMIT FLAG    4
0 0 0 0
 INPUT CONCAVE FLAG    4
0 0 0 0
 INPUT REVERSE RAISE FLAG    4
0 0 0 0
```

FIGURE 13: The profile data structure generated

1	1	4	1	0	0	0	0	0	0
0	0	1	0	1	3	1	0	0	0
0	0	1	0	2	2	1	0	0	0
0	0	1	0	3	2	1	0	0	0
0	0	1	0	1	3	1	0	0	0

FIGURE 14: The APT motion statments generated for the profile

```
CNC APTLOOP CURRENT
 PSURF =     1
 INPUT STARTING POINT
-10,-35,-10
  $$  APT  LOOP :    1
  FROM /     -10.0,-35.0,-10.0
  GO / TO,    F8          ,TO ,   F10        ,TO ,   F3
  INDIRP /    P3
  TLRGT , GOFWD /    E7          , TANTO ,   E4
  INDIRP /     116.0,0.000282,-80.0
  TLRGT , GOFWD /    E4          , TANTO ,   E21
  INDIRP /    P14
  TLRGT , GOFWD /    E21         , PAST ,    E22
  INDIRP /    P4
  TLRGT , GOFWD /    E22         , PAST ,    E7
  GODLTA /    0.0
  GO TO /     -10.0,-35.0,-10.0
 *
```

FIGURE 15a: The cutter path verification (plane view) of the outer profile

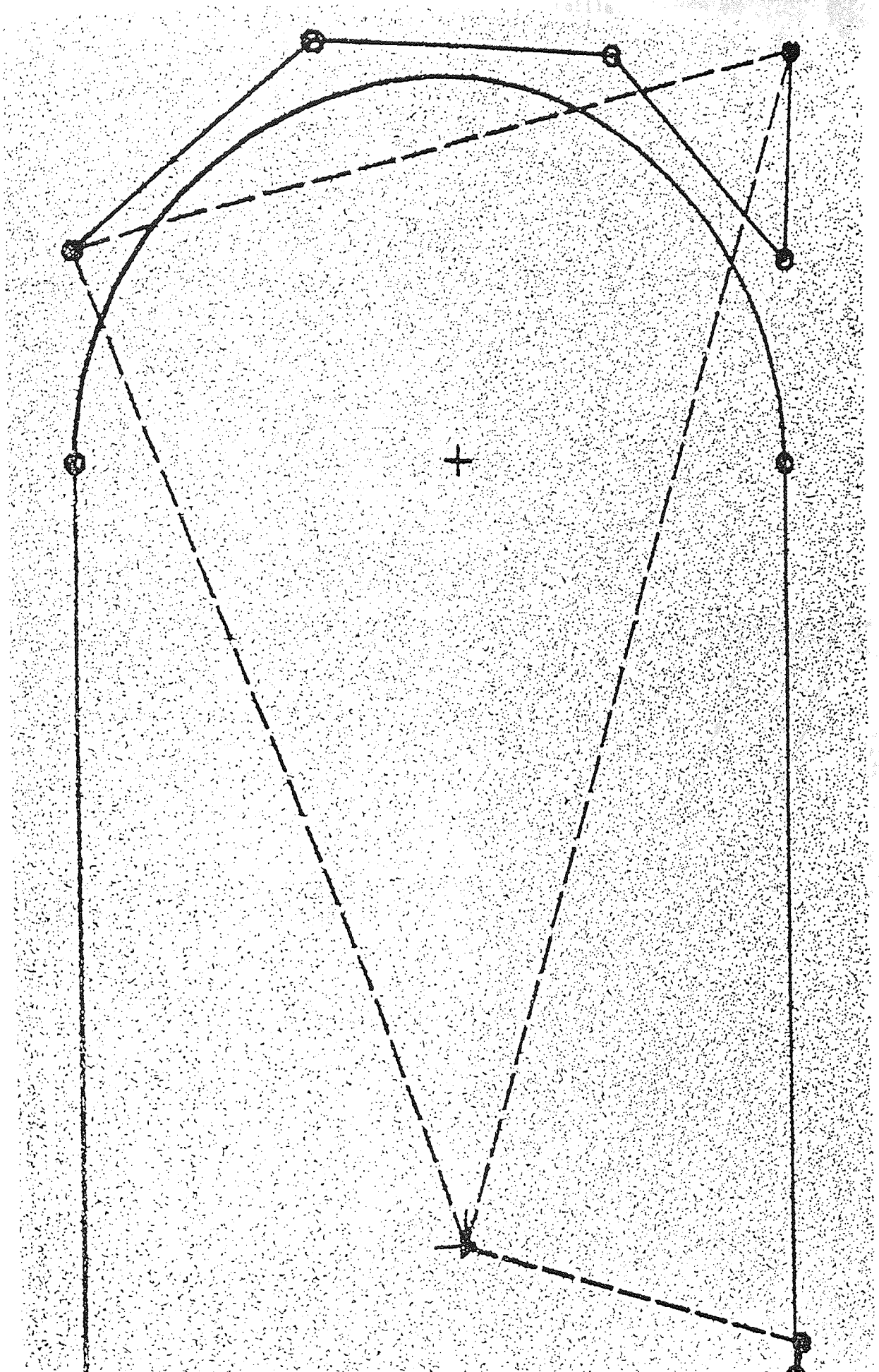


FIGURE 15b: The cutter path verification (perspective view) of the outer profile.

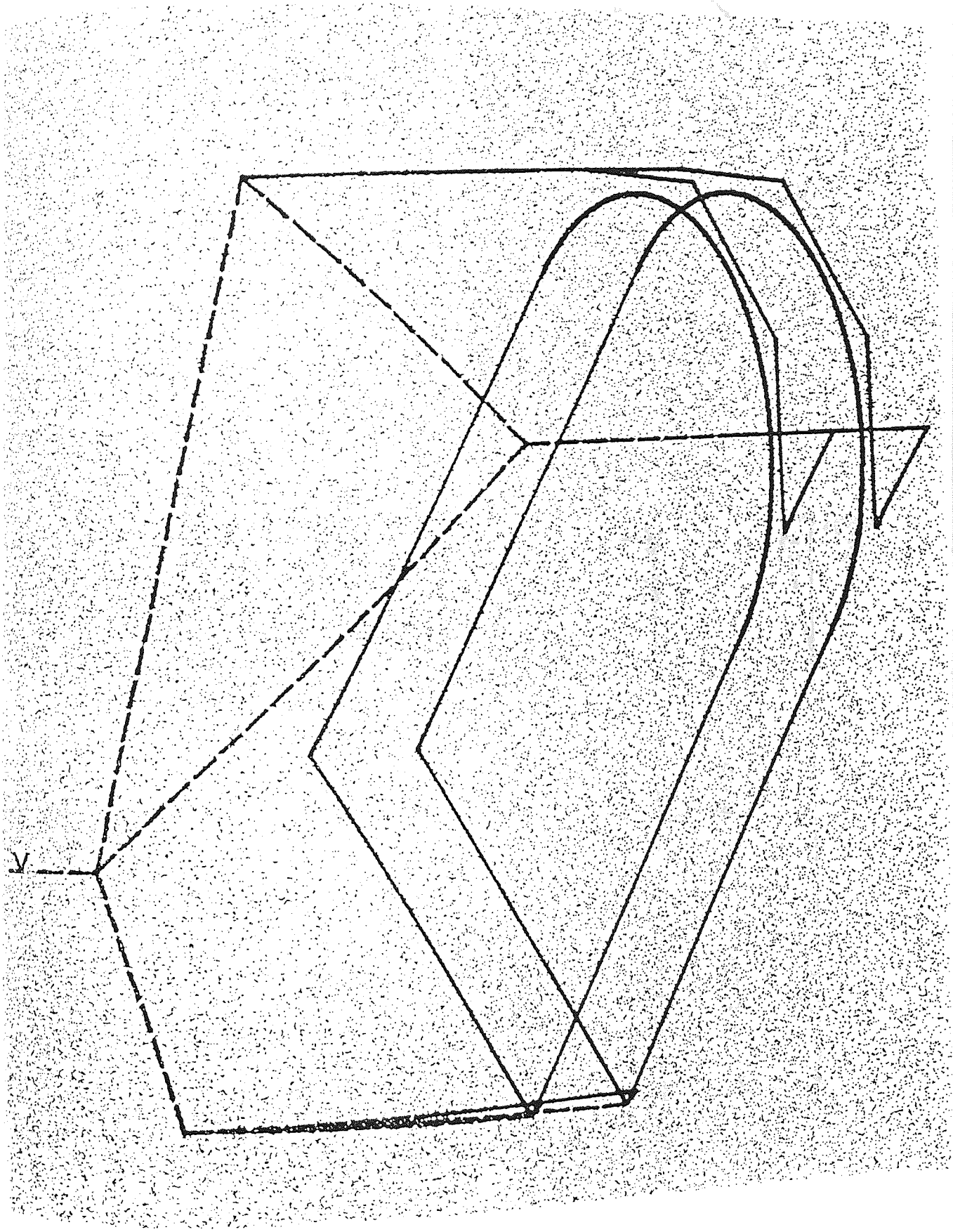
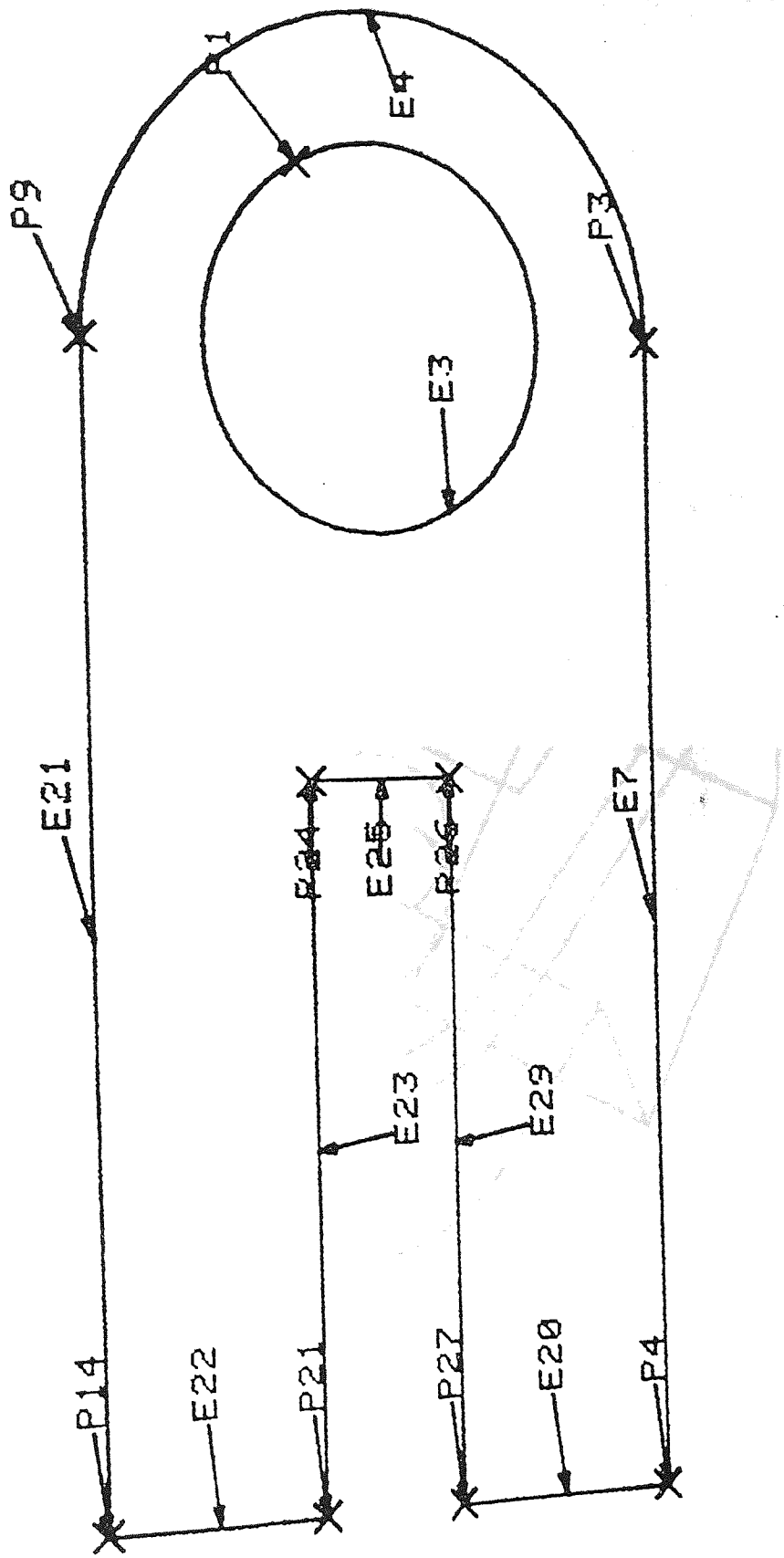


FIGURE 16a: The points and edges of the inner profile



SELECT F10 APTLABEL POINT F10
**** APTLABEL EDGE F10

FIGURE 16b: The cutter path generated for the inner profile

```
$ APTLABEL E7
$ SELECT F10
$ CNC CUT FLAT 6.35
                6.35000
$ PROFILE E7 LOOP FLAT 6.35 CURRENT
$
```

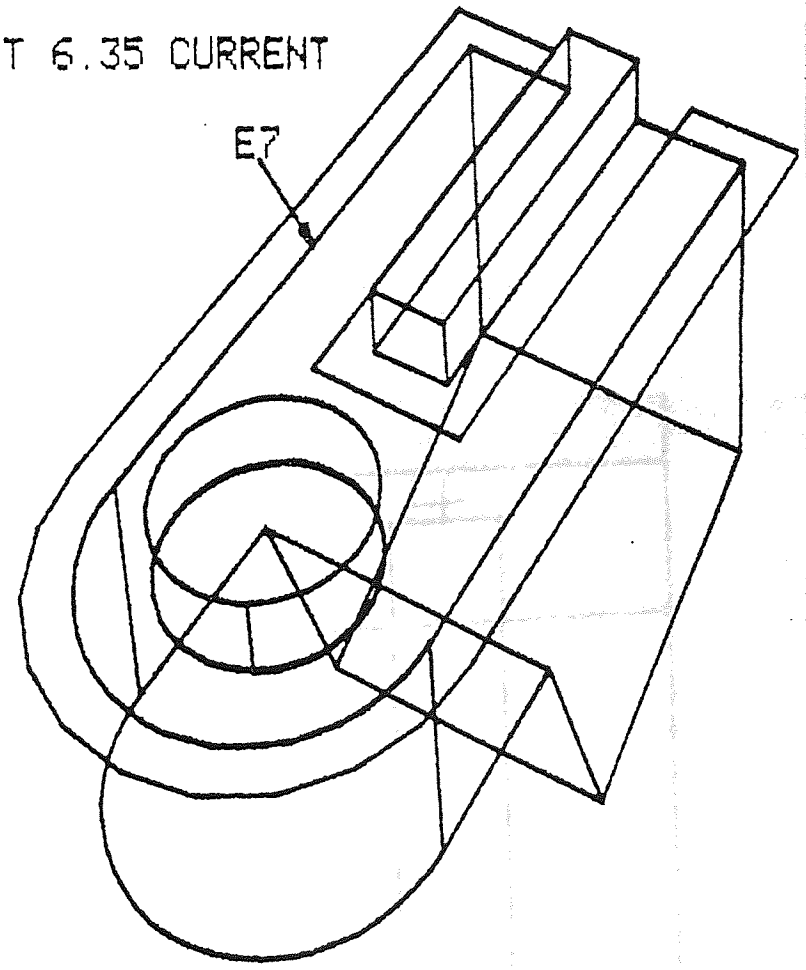


FIGURE 17: Automatic estimation of cutter size

```
CNC DISTANCE CURSOR
VEC BEFORE =  :  93.2129,90.644
VEC AFTER  =  :  -0.751141,76.2699
VEC BEFORE =  :  93.2129,71.0106
VEC AFTER  =  :  -0.751141,57.5329
DISTANCE =:  18.737
MAX. RADIUS =:  9.36848
```

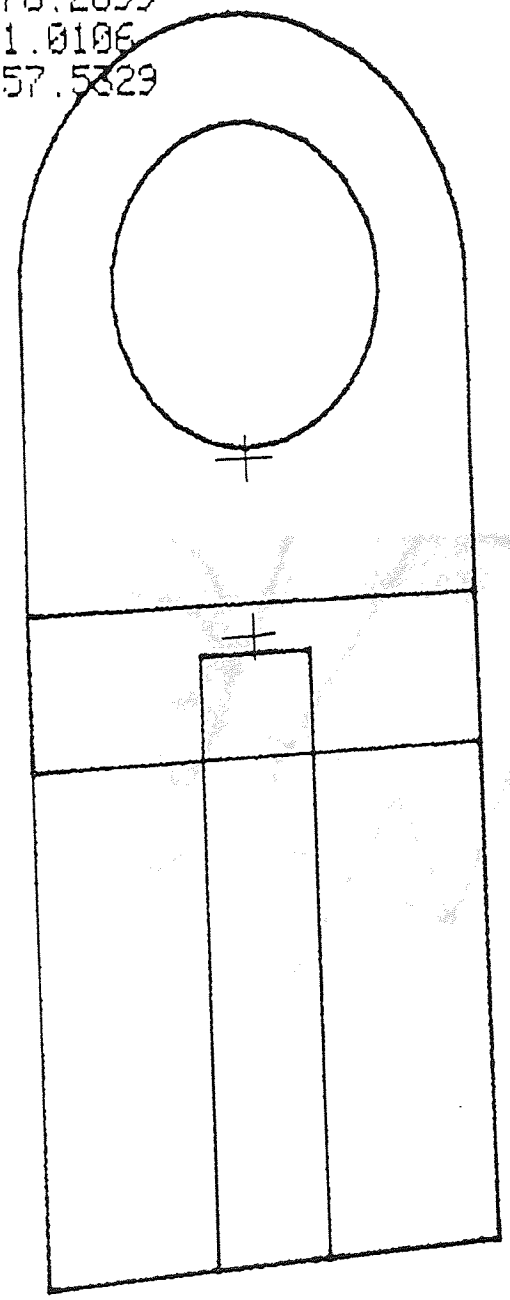


FIGURE 18: The procedure for the automatic generation of the data structure required for the inner profile

```
* SELECT F10
* APTLABEL E7
* P4
* P3
* CNC CUT FLAT 6.35
              6.35000
* CNC LOOP CLOSE OUT AUTOMATIC E7
 ZZ =     1.000
 INPUT GLOBAL RAISE FLAG
0
 INPUT DIRECTION (2 POINTS)
AUTOMATIC ?
 (POINT-NAME) ?
+ P4 P3
```

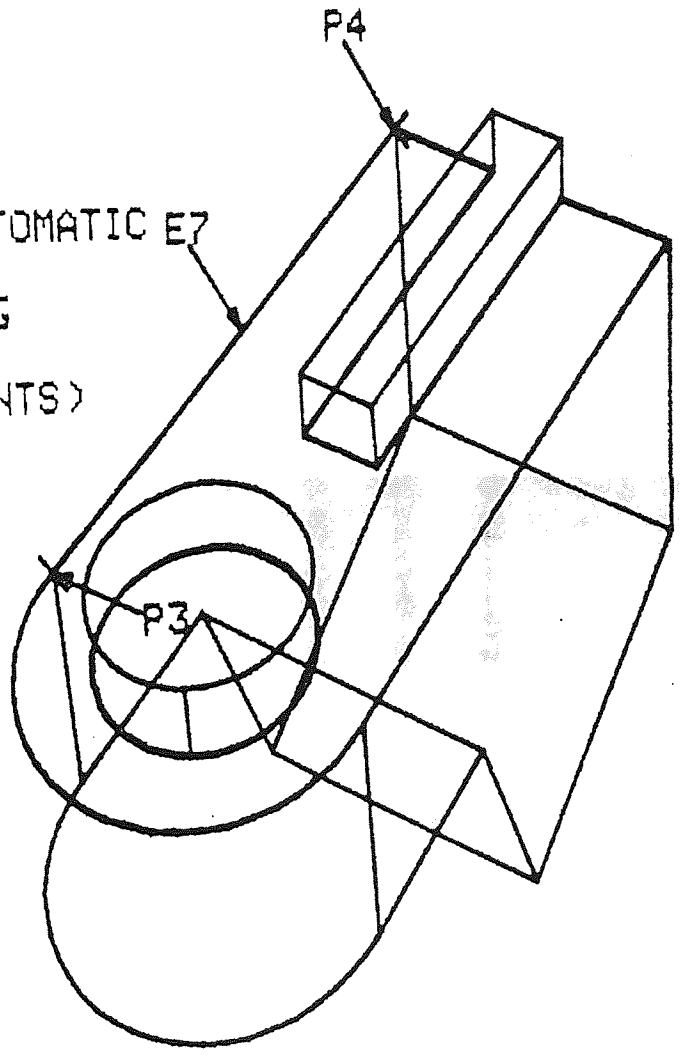


FIGURE 19. The data structure generated automatically for the inner profile

```
$17 $14 $204 $5 $311
$10 $272 $204 $318 $311
$76 $193 $204 $65 $311
$171 $248 $204 $186 $311
$341 $334 $204 $520 $311
$232 $189 $204 $201 $311
$297 $339 $204 $309 $311
$91 $180 $204 $186 $311
 1   0   8   1   0   0   0   0   0   0
 0   0   0   0   1   3   1   0   0   0
 0   0   0   0   2   2   1   0   0   0
 0   0   0   0   3   2   1   0   0   0
 0   0   0   0   1   3   1   0   0   0
 0   0   1   0   1   3   1   0   0   0
 0   0   1   0   1   4   1   0   0   1
 0   0   1   0   1   4   1   0   0   1
 0   0   0   0   1   3   1   0   0   0
```

FIGURE 20 The automatic generation of APT motion statements for the inner profile

```
CNC APTLOOP CURRENT
 PSURF =     1
 INPUT STARTING POINT
-10,-35,-10
  $$  APT  LOOP :    1
  FROM /     -10.0,-35.0,-10.0
  GO / ON,    E7          ,TO ,   F10       ,ON ,   E20
  INDIRP /    P3
  TLON, GOFWD /    E7          , TANTO ,   E4
  INDIRP /     116.0,0.000282,-80.0
  TLON, GOFWD /    E4          , TANTO ,   E21
  INDIRP /    P14
  TLON, GOFWD /    E21         , ON ,   E22
  INDIRP /    P21
  TLON, GOFWD /    E22         , TO ,   E23
  INDIRP /    P24
  TLLFT, GOFWD /    E23        , PAST ,   E25
  INDIRP /    P26
  TLLFT, GOFWD /    E25        , PAST ,   E29
  INDIRP /    P27
  TLLFT, GOFWD /    E29        , ON ,   E20
  INDIRP /    P4
  TLON, GOFWD /    E20         , ON ,   E7
  GODLTA /    0.0
  GO TO /     -10.0,-35.0,-10.0
 $
```

FIGURE 21: The cutter path generated for the inner profile

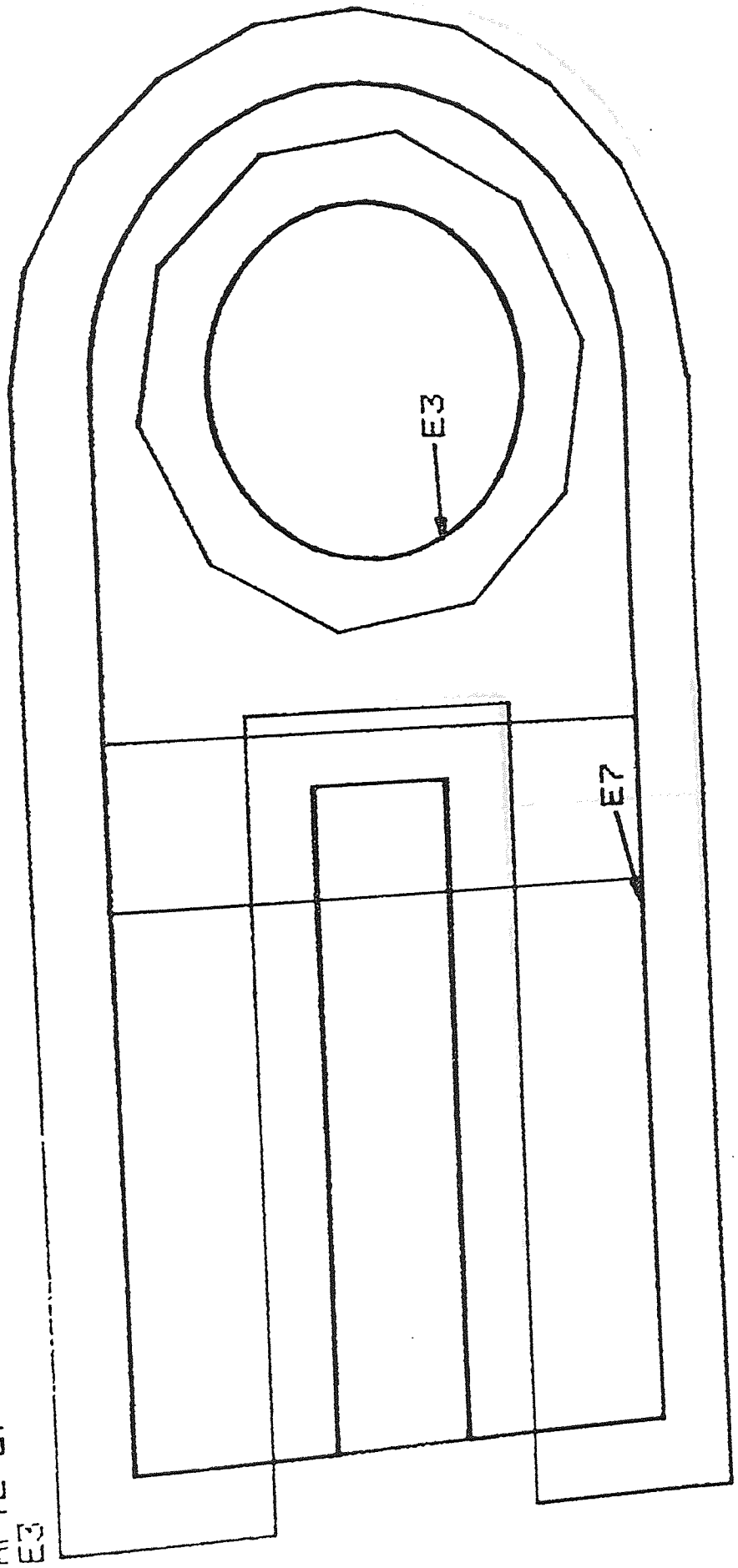
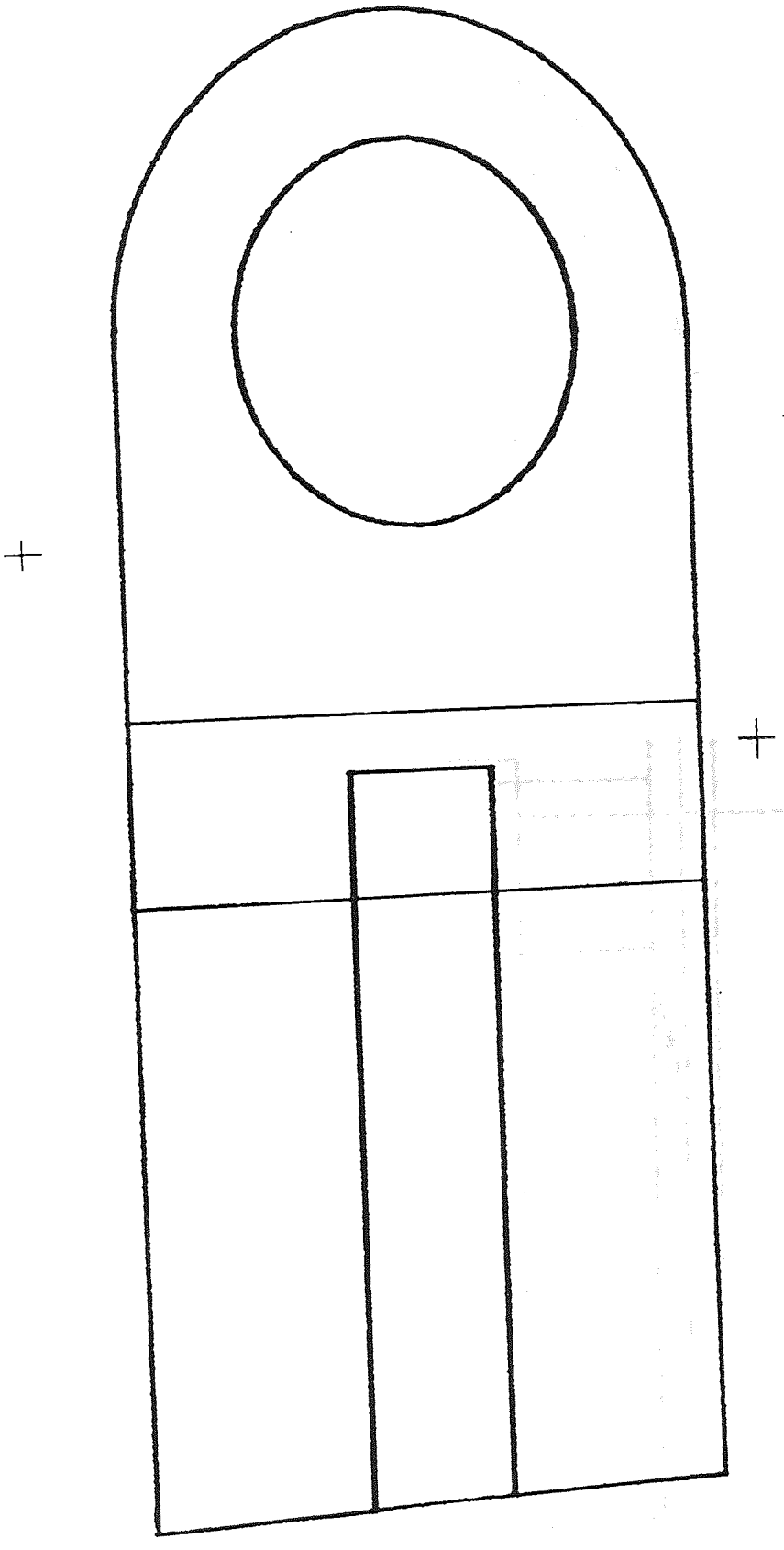
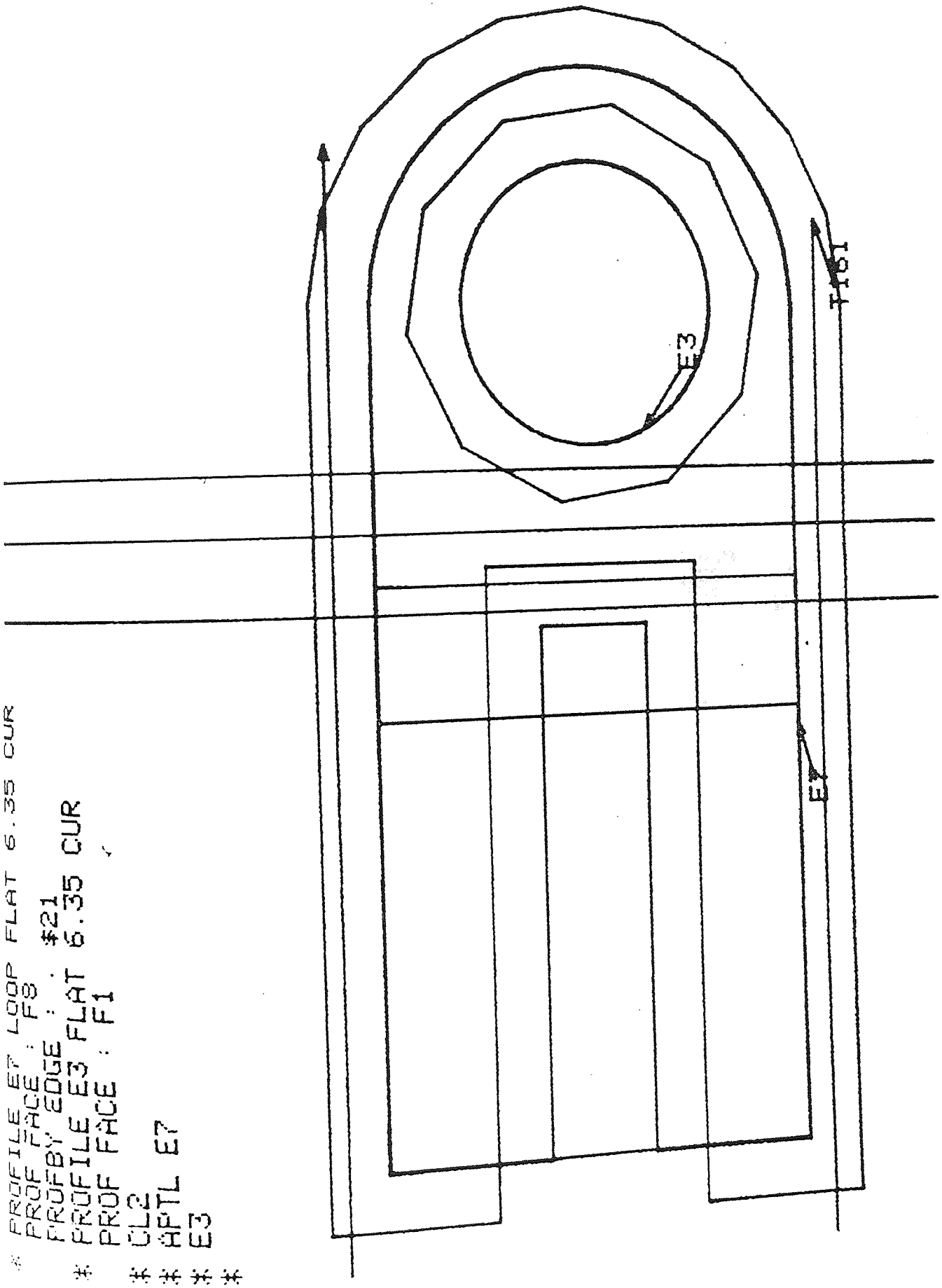


FIGURE 22: Procedure for area clearance



```
* CNC CUT FLAT 10
        10.000000
* CNC CLEAR CURSOR
```

FIGURE 23: The cutter path generated for the area clearance




```
* PROFILE EF LOOP FLAT 6.35 CUR
  PROF FACE : F8
  PROFBY EDGE : : #21
* PROFILE E3 FLAT 6.35 CUR
  PROF FACE : F1
  CL2
  APTL E7
  E3
  *****
```

FIGURE 24: The APT motion statements generated for the area clearance

```
$$ AREA CLEARANCE     F10
FROM /      66.0,1.67,-5.0
T149        =LINE/    59.6929,26.4174,-80.0    $
              ,       73.1457,26.4174,-80.0
T150        =LINE/    73.1457,26.4174,-80.0    $
              ,       73.1457,-29.9248,-80.0
T151        =LINE/    73.1457,-29.9248,-80.0    $
              ,       59.6929,-29.9248,-80.0
T152        =LINE/    59.6929,-29.9248,-80.0    $
              ,       59.6929,26.4174,-80.0
GO / TO ,    T152          , TO ,    F10          , TO ,    T149
TLRGT, GORGT /    T149          , TO ,    T150
T154        =LINE/    59.6929,6.4174,-80.0    $
              ,       73.1457,6.4174,-80.0
TLRGT, GORGT /    T150          , ON ,    T154
TLON, GORGT /     T154          , TO ,    T152
T155        =LINE/    59.6929,-3.5826,-80.0    $
              ,       73.1457,-3.5826,-80.0
TLLFT, GOLFT /    T152          , ON ,    T155
TLON, GOLFT /     T155          , TO ,    T150
$$  LAST CUT OF AREA
IC =      0
TLRGT, GORGT /    T150          , TO ,    T151
TLRGT, GORGT /    T151          , TO ,    T152
GO TO /     66.0,1.67,-5.0
*
```

FIGURE 25. The cutter path verification (plane view) for the whole inner profile

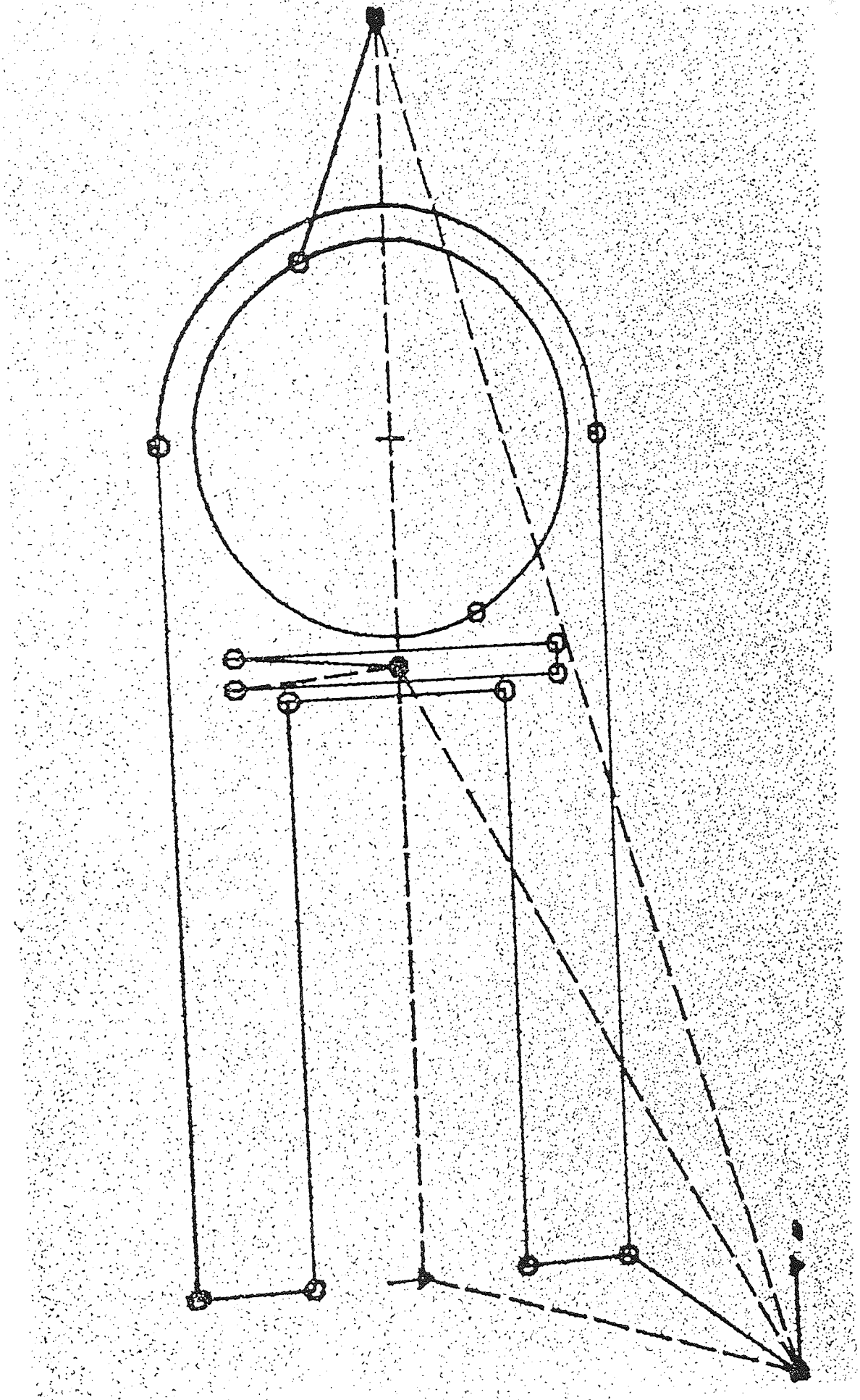


FIGURE 26: The cutter path verification (perspective view) for the whole inner profile

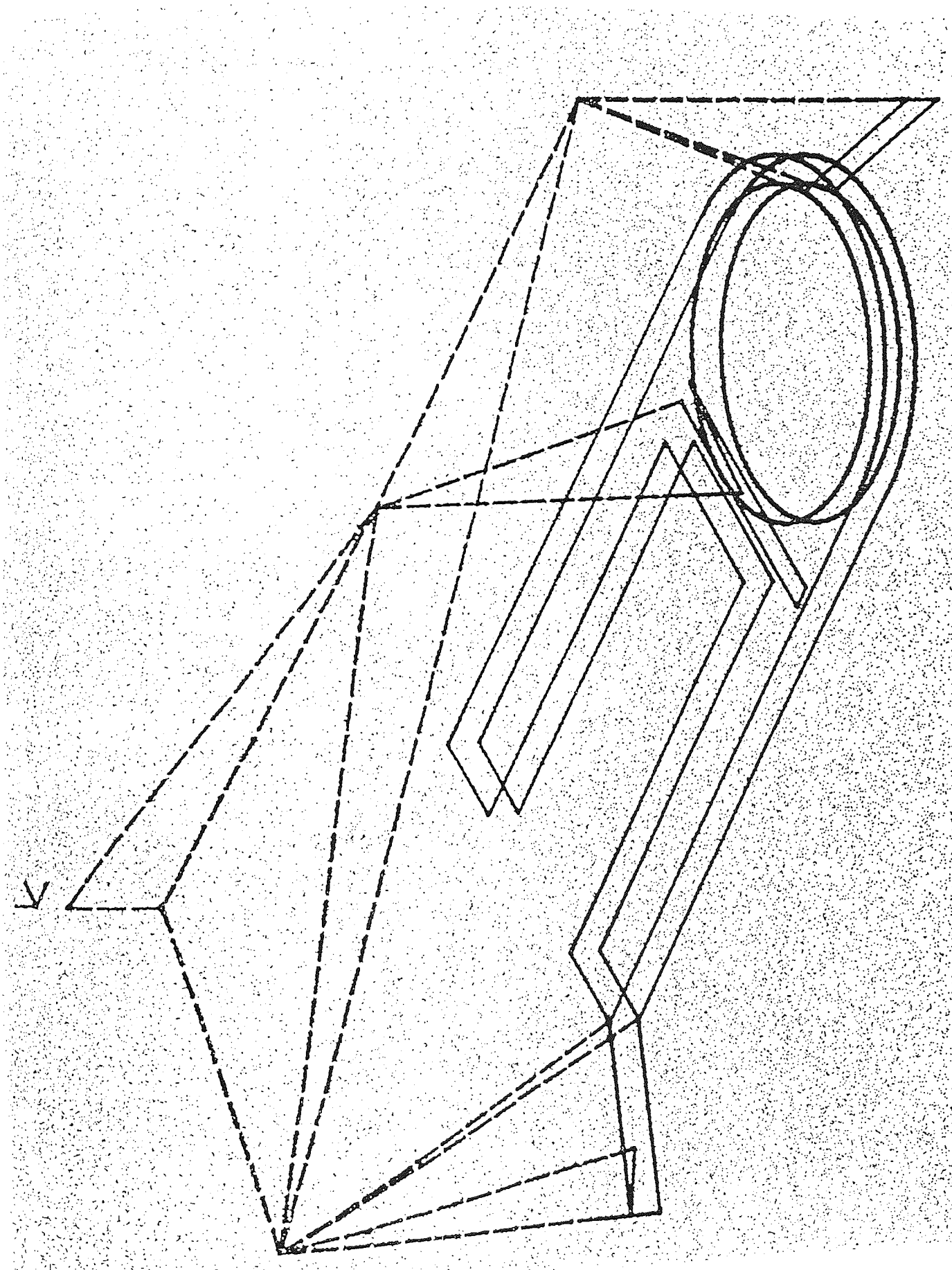


FIGURE 27: The generation of header, tolerance, cutter definition and end statements.

```
cnc header
  PARTNO   B1
  CLPRNT
  MACHIN / PLOTPP, 1
  UNITS / INCHES
* apte f1
  $$  THE FOLLOWING 3 LINES ARE CYLINDER DEFINITION   FOR   F1
  F1P          = POINT/   0.0,10.0,0.0
  F1V          = VECTOR/   0.0,-1.0,0.0
  F1           = CYLNDR/  F1P       ,  F1V       ,  8.0
* apte poi f1
  P12          = POINT/   -8.0,-10.0,0.0
  P14          = POINT/   8.0,-10.0,0.0
  P30          = POINT/   8.0,10.0,0.0
  P8           = POINT/   -8.0,10.0,0.0
  P1           = POINT/   -0.89431,1.78915,7.94985
* ?apte edg f1
  E2P          = POINT/   0.0,-10.0,0.000011
  E2           = LINE/  P14       ,  P12
  $$ LINEARIZED NON X-Y PLANE CIRCLE   E2
  E20          = LINE/  P14       ,  P30
  E3P          = POINT/   0.0,10.0,0.0
  E3           = LINE/  P8        ,  P30
  $$ LINEARIZED NON X-Y PLANE CIRCLE   E3
  E18          = LINE/  P8        ,  P12
  $$   E4          IS THE INTERSECTION OF FACE   F4        & FACE   F1
   $$   P1          TO   P1
* cnc mach
  $$ MACHINING START
  LOADTL / 1
  PPLOT / ON, XYZ, 1
  CUTTER /   4.0   ,   2.0
* cnc aptl cur
 PSURF =    1
 INPUT STARTING POINT
0,0,20
  $$  APT  LOOP :   1
  FROM /    0.0,0.0,20.0
 LOOP =          1              1
 LOOP =        -14924        -14668
                          -1.97789,-0.295607,7.75223
  E4M          = POINT /                    E4M
  E4L          = LINE /   P1          ,
                                        ,TO ,   F4
  GO / ON,   E4L         ,TO ,   F1          ,   E4L
  TLLFT, GOLFT /   F4              , ON ,
  GOFWD /   F4           , ON ,   E4L
  GODLTA /   0.0
  GO TO /    0.0,0.0,20.0
* cnc end
  END
  FINI
*
```

FIGURE 28 The whole cutter path of the part

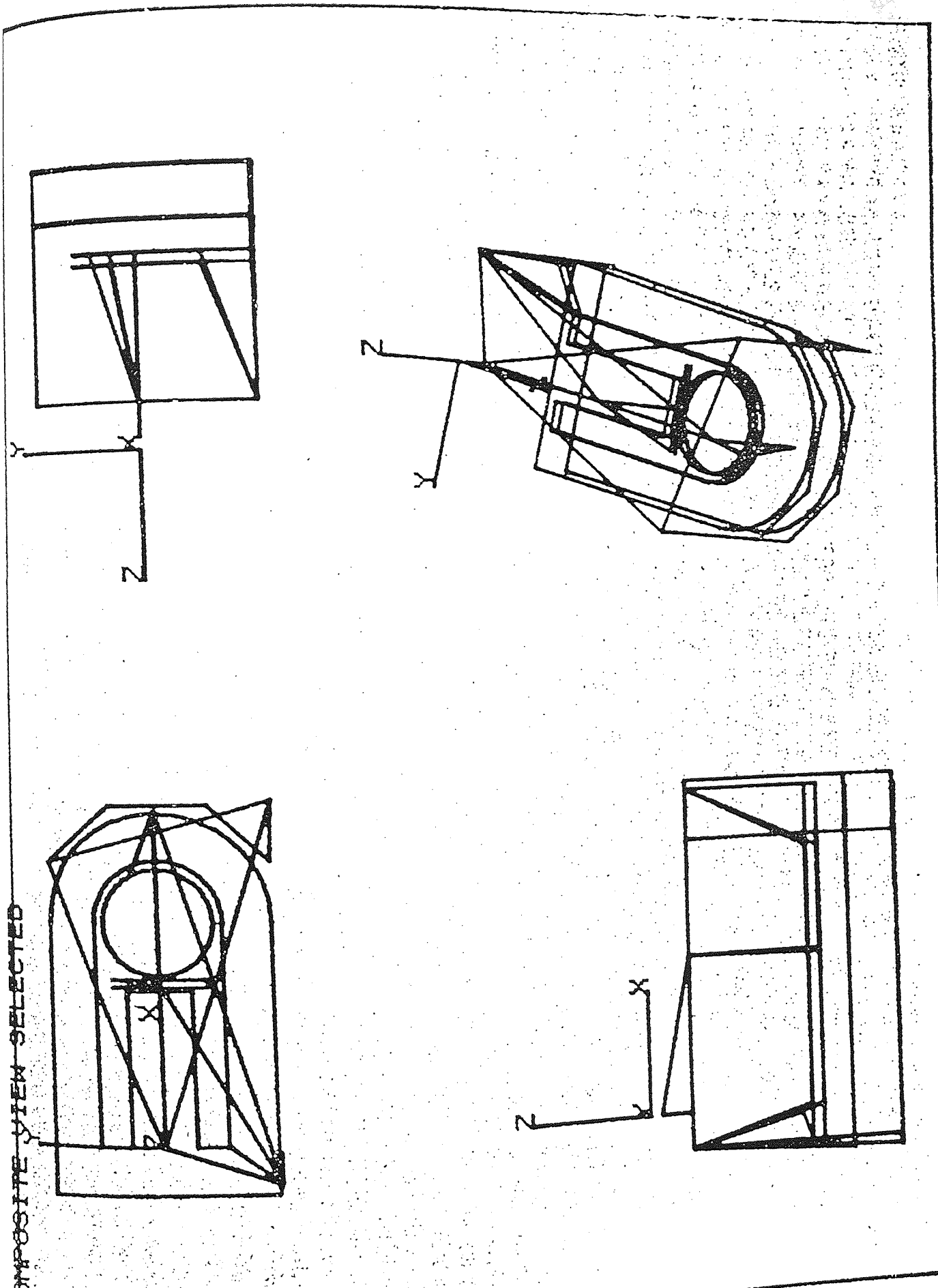


FIGURE 29 A complete APT part program generated by ROMAPT

```
PARTNO    B1
CLPRNT
MACHIN / PLOTPP, 1
UNITS / INCHES
INTOL /    0.05
OUTTOL /    0.05
$$  THE FOLLOWING 3 LINES ARE PLANE DEFINITION       FOR   F10
F10P         = POINT/    -0.0,-25.0,-80.0
F10V         = VECTOR/    -0.0,0.000001,1.0
F10          = PLANE/  F10P         , PERPTO ,    F10V
P4           = POINT/    -0.0,-25.0,-80.0
P3           = POINT/    91.0,-24.9998,-80.0
P9           = POINT/    90.9999,25.0002,-80.0001
P14          = POINT/    -0.000122,25.0,-80.0001
E7           = LINE/  P3          ,  P4
E4P          = POINT/    90.9999,0.000237,-80.0
E4           = CIRCLE/CENTER,  E4P        , RADIUS ,  25.0
 $$    P3           TO    P9
E21          = LINE/  P14         ,  P9
E22          = LINE/  P21         ,  P14
$$  THE FOLLOWING 3 LINES ARE PLANE DEFINITION       FOR   F8
F8P          = POINT/    0.0,-25.0,-95.0
F8V          = VECTOR/    0.000002,-1.0,0.000002
F8           = PLANE/  F8P          , PERPTO ,    F8V
$$  THE FOLLOWING 3 LINES ARE PLANE DEFINITION       FOR   F3
F3P          = POINT/    0.0,-25.0,-95.0
F3V          = VECTOR/    -1.0,-0.000002,0.0
F3           = PLANE/  F3P          , PERPTO ,    F3V
$$ MACHINING START
LOADTL / 1
PPLOT / ON, XYZ, 1
CUTTER /    8.0
$$  APT  LOOP :    2
FROM /     -10.0,-35.0,-10.0
GO / TO,    F8          ,TO ,    F10        ,TO ,    F3
$$ INDIRP /    P3
TLLFT , GORGT /    E7          , TANTO ,    E4
$$ INDIRP /     116.0,0.000282,-80.0
TLLFT , GOFWD /    E4          , TANTO ,    E21
$$ INDIRP /    P14
TLLFT , GOFWD /    E21         , PAST ,    E22
$$ INDIRP /    P4
TLLFT , GOLFT /    E22         , PAST ,    E7
GODLTA /    0.0
GO TO /     -10.0,-35.0,-10.0
END
FINI
```

REFERENCES

(1) ACARD, "Computer Aided Design and Manufacture", Advisory Council for Applied Research and Development, Cabinet Office, HMSO, U.K., (1980).

(2) BRAID I., "Designing with Volumes", PhD Thesis, Cambridge University, U.K., (1973).

(3) BRAID I., "New Direction in Geometric Modelling", CAD Group Document No. 98, Computer Laboratory, University of Cambridge, U.K., (March 1978).

(4) JARED G., STROUD I., "Local Operators in the BUILD System", Advances in CAD/CAM, Ellis & Semenkov, ed, North-Holland, pp. 55-65 (1983).

(5) VEENMAN P., "ROMULUS - The Design of a Geometric Modeller", Geometric Modelling Seminar, W.A. Carter, ed, P-80-GM-01 CAM-I, Bournemouth, U.K., pp. 127-152 (Nov. 1979).

(6) BAER, EASTMAN, HENRION, "Geometric Modelling: A Survey", CAD, Vol.II No.5 , pp.253-272 (Sept. 1979).

(7) OKINO N. et al, "TIPS-1", Institute of Precision Engineering, Hokkaido University, Sapporo, Japan (1978).

(8) "PADL System Documentation", Production Automation Product, University of Rochester, New York (1977).

(9) SHAPE DATA LIMITED, "Romulus Version 4.0 Users' Reference Manual", Shape Data Limited, Cambridge, U.K. (1982).

(10) NBS, "Initial Graphics Exchange Specification Version 1.0", National Bureau of Standards, (Jan. 1980).

(11) CAM-I, "CAM-1 Geometric Modelling, Boundary File Design (XBF-2)", CAM-I, Report R-81-GM-02.1, (Oct. 1982).

(12) CAM-I, "An Interface Between Geometric Modelling and Application Programs", CAM-I Report R-80-GM-04, (Dec. 1980).

(13) FERGUSON J.C., "Multi-Variable Curve Interpretation", The Boeing Company, No. D2-22504, (1963).

(14) COONS S.A., "Surface for Computer-aided Design of Space Forms", Project MAC, MIT, (1967).

(15) BEZIER P., "Numerical Control", Mathematics and Applications, Wiley (1972).

(16) GORDON W.J., RIESENFELD R.F., "B-Sphere Curves and Surfaces" in Computer Aided Geometric Design, Barnhill & Riesenfeld, ed, Academic Press (1974).

(17) CIS, "Meduser Drafting System User Guide", MED 1001/2/0, Cambridge Interactive System Limited, Cambridge (1981).

(18) BAUMGART B.G., "Winged Edge Polyhedron Representation", AIM-79, STAN-CS-320, Stanford University (1972).

(19) EASTMAN C. and HENRION, "GLIDE: A Language for Design Information Systems", Computer Graphics (Siggraph 1977 Proc), Vol.II No.2, pp. 24-33 (July, 1977).

(20) CAM-I, Proceedings of CAMI's 5th Annual Meeting of Members, Dallas, Texas, U.S.A., P-77-MM-01, (Nov. 1976).

(21) HOSAKA, KIMURA, KAKISHITA, "A Unified Method for Processing Polyhedra", Information Processing, North-Holland Publishing Company, Amsterdam (1974).

(22) SPUR G, "Status and Further Development of the Geometric Modelling System COMPAC", Proc. Geometric Modelling Project Meeting, P-78-GM-0, CAMI, St. Louis (Mar. 1978).

(23) SEIFERT H., BARGELE W., FRITSCHE B., "Different Ways to Design 3D Representations of Engineering Parts with PRORGN2", Proc. Conf. Interactive Techniques Computer Aided Design, Bologna, Italy (Sept. 1978).

(24) BERNASCON T.J., BRUN J.M., "Automated Aids for the Design of Mechanical Parts", Technical Paper MS75-508, SME, (1975).

(25) ENGALI M.E., "A Language for 3D Graphic Application in International Computing Symposium", Grunther, ed, North-Holland, (1973).

(26) SABIN M.A. (Ed), "Programming Techniques in Computer Aided Design", NCC, England, pp. 283 (1974).

(27) ANSI , "Digital Representation of Physical Object Shapes", ANSI Y14.26 Subcommittee Technical Report (Jan. 1976).

(28) NBS , "Initial Graphics Exchange Specification Version 2.0", U.S. Department of Commerce, National Bureau of Standards (Feb. 1983).

(29) IBM, SYSTEM/370 APT-AC Numerical Control Program Reference Manual Vol. I, PROGRAM S-740-M53, IBM (1972).

(30) IITRI, APT4 Computer System Manual, IIT Research Institute, RM-79-APT-01, CAM-I (1979).

(31) CAM-I, "Sculptured Surfaces Documentation", Vol I, II, III, CAM-I Report PS-80-SS-01, (Sept. 1980).

(32) WILSON P.R., "Felset User Manual", Lucas Research Centre, CPM 060, (1976).

(33) SNEAD J.C., "The SURFEST Suit - an Overview", Lucas Research Centre, CPM116 (1978).

(34) LUCAS, "Implementation and Testing of the CAM-I GMP Boundary Representation for Solid Objects", Lucas CAM-I Contract Report, R83-GM-01 (1983).

(35) FAUX I.D., "User Requirements for Geometric Modelling", Cranfield I.T., (Oct. 1979).

(36) CAM-I, Minutes of Advanced Technical Planning Committee Meeting, M-78-ATPC-0, Royal Bath Hotel, Bournemouth, U.K., (Mar. 1978).

(37) GRAYER A., "The Automatic Production of Machined Components Starting From a Stored Geometric Description", PROLAMAT, (1976).

(38) CAM-I, "Design of an Advanced Numerical Control Processor", CAM-I Report Vol. I & II R-82-ANC-01 (1982).

(39) ROGERS D.F., ADAMS J.A., "Mathematical Elements for Computer Graphics", McGraw Hill, (1976).

(40) WOO T.C., "Computer Aided Recognition of Volumetric Designs", Vol. I, PROLAMAT, Stirling, Scotland, U.K., (1976).

(41) CHOI B., BARASH M., ANDERSON D., "Automatic Recognition of Machined Surfaces from a 3D Solid Model", CAD Vol.16 No.2, (March 1984).

(42) ROMULUS Version 4 Programmer's Manual, Shape Data Limited, U.K.,

(May 1982).

(43) CHAN B.T.F., "ROMAPT: A New Link Between CAD and CAM", CAD Journal, Vol.14 No.5, pp. 261-266, (Sept. 1982).

(44) CHAN B.T.F., MOTTERSHEAD J.E., KELLMAN M.P., "Finite Elements and Related Areas of Computer Aided Engineering", Materials & Design, Vol.3 No.4, pp. 495-501, (Aug. 1982).

(45) CHAN B.T.F., "NC Machining Developments at Lucas based on a Geometric Part Model", 2nd CAM-I Geometric Modelling Seminar, Robinson College, Cambridge, England, pp. 31-79A, (Dec. 1983).

(46) CHAN B.T.F., "Building the Unbounded: ROMAPT", Research Horizon, Lucas Research Centre, (Jan. 1983).

(47) CHAN B.T.F., "A New Link for CAD and CAM", Industrial Automation Reporter, (Feb. 1983).

(48) CHAN B.T.F., "CADAPT Project: Mathematical Technique", Lucas Research Centre, (Oct. 1978).