AN APPROACH TO
THE LOW COST COMPUTER AIDED DESIGN
AND MANUFACTURE OF COMPONENTS
EMBODYING FREE FORM SURFACES

A Thesis Submitted to

THE VICTORIA UNIVERSITY OF MANCHESTER

for the Degree of

DOCTOR OF PHILOSOPHY

by

XIAO-YU GAO

Manufacturing and Machine Tools Division
Department of Mechanical Engineering
University of Manchester
Institute of Science and Technology

June 1989

DECLARATION

No part of the work described in this thesis has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Professor B. J. Davies for his direction and supervision throughout the project, and to Dr. O. H. Narayanan, Dr D. K. Harrison and Mr. N. Clayton for their help in using the CAD/CAM facilities in the UMIST CAE Unit.

The author also wishes to express his sincere thanks to Mr. J. E. Boon and Mr. K. Dabbs for their help and contribution in the preparation for machining the bottle mould and the scaled cab roof model on the UMIST Takisawa Machining Centre.

The author also wishes to express his sincere thanks to Mr. Z. R. Li for his suggestions and help during the project.

ABSTRACT

The modelling and numerical control (NC) machining of free form surfaces is usually accomplished utilizing minicomputer or mainframe computer aided design and manufacture (CAD/CAM) systems. The resulting NC data are suitable for either three-axis of five-axis machining dependent upon machine tool type/capability and CAM package complexity. However, the cost of such packages and systems precludes the use of these in smaller companies with an occasional requirement for components embodying free form surfaces. The increasing availability of powerful yet reasonably priced personal computers (PC) coupled with improving software provides such companies with a far cheaper alternative approach. However, the PC based CAD systems are not sufficient for this work and such CAM systems are limited to 2-1/2-axis machining for all but special applications such as three-axis canted Z-plane milling of four-axis wire electro discharge machining (EDM).

This thesis presents a method of modelling and manufacturing true three dimensional free form surfaces using PC based CAD/CAM systems. The method is particularly suitable for manufacturers who make use of moulds, such as are found in the plastic processing industries. The modelling of a telephone handset from digitized data and a plastic bottle from 2 dimensional engineering drawings and the manufacture of a plastic bottle mould and a scaled cab roof model are presented as typical examples.

PREFACE

Computer Aided Design and Manufacture (CAD/CAM) has been widely used in aircraft, space, military, ship building and motor car industries. Increasingly, smaller manufacturers manufacturing a wide range of products have started using this new technology. The use of low cost Personal Computer (PC) based CAD/CAM systems in relatively small companies, such as the plastic mould making industry, will increase markedly in the coming decades.

This thesis introduces a method of modelling and NC manufacturing of components embodying free form surfaces with PC based CAD/CAM systems. The background of CAD/CAM is first reviewed in Chapter One and the general concepts of CAD/CAM systems and applications are introduced in Chapter Two. Chapter Three outlines the scope of the project, the software and hardware used, and the general procedure for designing and manufacturing a component composed of complex surfaces. The generation of surface models from digitized data and 2 dimensional engineering drawings are described in Chapter Four and Chapter Five respectively. Chapter Six details the preparation of toolpath geometry for NC machining and Chapter Seven describes the automatic part programming and NC machining of two examples (a bottle mould and a scaled cab roof model). Finally, the general conclusions and recommendations for further work are given in Chapter Eight.

The figures, tables, pictures, User Programming Language (UPL) programs and some typical NC Part Programs and post-processed Part Programms are given in the appendices at the end of the thesis.

CONTENTS

Chapter One

The Background of CAD/CAM

1.1 Summary

This chapter reviews the development of Computer-Aided Design and Manufacture (CAD/CAM), from the first appearance of digital computers to the current mini and micro computers. The development of the CAD/CAM software, Numerical Control (NC) technology and the theory of CAD/CAM - numerical geometry, are also detailed, together with the growth of the CAD/CAM industry and the international competition in both general-purpose digital computers and CAD/CAM systems.

1.2 The development of digital computers

Modern digital computers first appeared in the 1940s[1]. They were huge, electromechanical machines that used clicking relays to perform computations. The first and largest of this generation of computers was Aiken's 5-ton Automatic Sequence Controlled Calculator, or MARK I, built in cooperation with the IBM Corporation in 1944[2].

Soon relays and other mechanical moving parts were replaced by vacuum tubes. The first of these vacuum-tube computers was ENIAC (Electronic Numerical Integrator and Calculator) developed for the U.S. Army in 1945 by J. Presper Eckert and John W. Mauchly of the Moore School of Engineering of the University of Pennsylvania[3] [4]. Subsequently, many other vacuum-tube computers emerged in rapid succession in the late 1940s. One of the most powerful of these was IBM's Naval Ordnance Research Calculator.

Transistors replaced vacuum tubes in computer circuits in the late 1950s, creating a new generation of faster, more compact equipment. The most widely used computers of the second generation were IBM1620, IBM1401 and IBM7094.

Third-generation computers are characterized by the use of hybrid or integrated circuits, These computers began to be delivered in 1965 and the most common system was IBM system/360. Computer size and cost were further reduced and computing speeds increased in the 1970s with integrated circuitry. Because of integrated circuit technology, computers today are much more compact, faster, and less expensive than their earlier counterparts. Furthermore, minicomputers about the size of a desk now handle complex tasks that only a few years ago required a mainframe, and a type-writer-size desktop computer contains the computing power of a bulky system that would have filled a room years ago.

1.3 The Appearance of mini- and Personal Computers

Minicomputers first appeared about the beginning of the 1960s[5]. Their progress followed closely the development of digital circuit technology. First transistors, then integrated circuits and now Medium Scale Integration (MSI) and Large Scale Integration (LSI) circuits have been used in their construction. To illustrate the early rate of expansion, in 1969 about 8,000 machines were installed worldwide. In 1975 this figure reached 40,000 installations[6].

The Personal Computer (PC) is an adaptation of microcomputer technology to 'Personal' type applications. And the introduction of the Itel 8008, an 8-bit microprocessor, in the early 1970's made the PC commercially possible[7]. The first use of microprocessors for PCs was in the form of kits for the hobby or educational market. The Scelbi 8H, using the Intel 8008, introduced in 1973, was perhaps the first personal microcomputer kit, but it was not widely marketed. Until January 1975, MITS, Inc. of Albuquerque, New Mexico, developed an 8008 system, called the Altair 8800, and became an overnight success. The PC industry was born.

In 1976 and 1977 there were over a dozen small companies manufacturing PCs and MITS and IMSAI were market share leaders. Radio Shack entered the market in 1978 with its TRS-80 computer and with its substantial production and distribution capability, It soon completely overwhelmed the PC market.

The IBM PC was announced in August 1981 and marks the first entry of IBM into the low cost, Personal Computer market.

1.4 The birth and growth of CAD

The first use of Computer Aided Design (CAD) was during the 1950s[8], in the aerospace and military organizations of the United States. By the mid-1950s the SAGE Air Defence System was using a light pen to display numerical information, but there was still no grasp of dynamic interaction[9]. Throughout the 1960s, the Aerospace Industry Association was a major source of funding for work on CAD and from the early 1960s, there was increased interest in CAD in the U.S. motor industry.

The first important step forward in computer graphics was at Massachusetts Institute of Technology (MIT) in 1963 when a 2 dimensional (2D) interactive CAD system, called SKETCHPAD - A man-machine communication system, was demonstrated[10] [11]. This system consisted of a cathode-ray oscilloscope driven by a Lincoln TX2 computer whereby graphical information was displayed on the screen[12]. Soon afterwards, Johnson[13], devised a 3D drawing system, SKETCHPAD III. At this time, the mathematical description of surfaces was being studied by the Lochkeed-Georgia Corporation[14]. They announced a research program in 1965, which later became the commercial CADCAM system now leased by IBM. However, the interactive graphics systems developed in the 1960s were restricted mostly to very large companies such as General Motors and Boeing because of their high costs.

By the close of 1960s there was a great expansion in commercial involvement in computer graphics. This was a spur to the next important development - the arrival of raster based displays and plotters. These were the results, in the early 1970's, of the computer graphics industry looking to t.v. technology to provide a new type of graphic device which would be inexpensive, and for the first time exploit the full potential of colour.

In 1971, the first practical surface design system (UNISURF) was introduced by Bezier[15]. This system enables sections of curves and surfaces to be freely designed by an operator who needs no advanced mathematical training and who works entirely in terms of elementary geometrical concepts. And by the early 1970s, interactive graphics could be performed on less-expensive minicomputers and then on the then recently developed 16-bit microcomputers. This made the systems economically practical for a much wider range of companies in general industry.

Near the end of the 1970s, powerful multi-user mini systems such as the PRIME 750 and VAX 11/780 allowed a number of such users to operate on a single system thus lowering the price per user slot[16]. At the beginning of the 1980s, 32-bit microcomputers became available and were quickly adopted. Now, 2D drawing, 3D wireframe, surface modelling and solid modelling are available, all of which may be run on microcomputers.

1.5 The birth and growth of NC technology and CAM

As improvements were being achieved in CAD, Numerical Control (NC) technology was being developed in parallel. Numerical Control began around 1947 when John Parsons of the Parsons Corporation tried to generate continuous curve data to control machine-tool motions[17]. He had been working on a project for developing equipment that would machine flat templates for inspecting the contour of helicopter blades. A proposal covering a machine to prepare these templates was presented by the Parsons Corporation to the U.S. Air Force and resulted in a development contract in 1948[18]. In 1949 Parsons was joined by MIT as a major subcontractor on the project. Development work continued throughout 1950s[19]. This resulted in the successful demonstration of a three-motion milling machine and the APT (Automatically Programmed Tools) language for programming NC machine tools[20].

After studying the applicability of numerical control machining for high-speed aircraft structures, the Aerospace Industries Association, an organization comprised of aerospace manufacturers, recommended to the Air Force that forthcoming machines be equipped with numerical controls and in 1955 the Air Force began awarding some 35 million dollars for the manufacture of approximately 100 numerically controlled milling machines. It was not until approximately 1960 that numerical control began to appear on a reasonably wide commercial scale.

In the 1960s, the automatic generation of NC tapes for 2D milling operations was being studied[14]. And the growth in the number of numerically controlled machines, has been accelerating rapidly.

From 1970, the development in NC can be tracked against the development of the memory element, the Dynamic Random Access Memory[21]. The first generation of NC used a hardwired computer comprising memory elements up to 1K bits per device and the predecessor of the microprocessor, the Arithmetic Logic Unit. By 1974 the early 8 bit mini computer was on the scene and this was developed into an NC system by most NC suppliers. By 1976 the first generation 8 bit microprocessors were being used and memory size had quadrupled by use of 16K devices. Today NC uses several microprocessors (typically 3) and in most cases 16 bit devices are employed. Computing power and memory size are advancing on a logarithmic scale with respect to time and already discussions are taking place regarding the use of 32 bit processors for products late this decade.

From mid 1970s, the early NC controls were replaced with Computer Numerical Controls (CNC) so that part program buffer storage, tape editing facilities and full cutter diameter compensation are available[22]. Three axis and five axis machining packages are now available to generate NC data from components with complex shapes designed on CAD systems and the process from design to manufacture of components using CAD/CAM methods is well established, although the mechanism still relies on the input of know-how from the draughtsman, the planner, the part- programmer and others. Some machining packages can be run on Personal Computers (PC's), but these tend to be limited to two and half (2-1/2) axis machining, which cannot address the manufacture of 3 dimensional free form surfaces[13].

1.6 The principal developments in numerical geometry

Numerical geometry first assumed practical importance during the Second World War when production pressures, particularly in the aircraft industry, stimulated the development of new methods of design[24]. The new methods were based upon analytical curves, conics[25] and the design process had been mainly graphical, using the techniques given in books such as that by Wellman[26]. Much of the manual draughting work formerly undertaken was avoided, but a considerable element of computation was introduced.

With the advent of electronic computer, graphical methods were further developed. For instance, surfaces had been previously represented by the construction of a number of longitudinal curves to blend a set of cross-sections, a process called LOFTING. Surface can not be defined completely, but only a system of curves lying on the surface. However, the new methods treat the longitudinal and the cross-sectional curves in an equal manner and regard them as dividing the surface into quadrilateral patches, each of which may be represented by a mathematical formula.

In 1963, Ferguson[27] developed his patch system which was one of the earliest systems and has since become standard usage because of the use of parametric rather than Cartesian coordinates in its curve and surface definitions. The major advantage of the parametric coordinates is that shape is independent of the frame of reference[28], and so the parametric approach may be regarded as an entirely natural development. Moreover, it enables twisted curves in three dimensions to be given a simple mathematical representation and coordinate transformations, projections, perspectives and so on for graphics displays to be performed very simply. It also avoids certain problems which can arise in representing closed curves and curves with vertical tangents in a fixed coordinate system.

A number of important developments were made after the first surface-defining systems became operational. First, the properties of spline curves and surfaces were much exploited and it has been shown that any spline curve or surface can be expressed in terms of fundamental splines or B–splines[29], which enables local modifications to be made without the need for its numerical representation to be recomputed from scratch each time. Secondly, it was shown by Coons[30] in 1967 how four arbitrary boundary curves can be blended into a single smooth patch and demonstrated how inter-patch continuity of gradient and curvature may be achieved. However, the theory can not be easily understood by an operator who may not be an expert in mathematics, because of its generality. The third major mathematical developments was the UNISURF system introduced by Bezier in 1971[15], which enables curves and surfaces to be freely designed by an operator who needs no advanced mathematical training, and a similar system proposed by Gordon and Riesenfeld[31] [32] which makes use of the B-splines and exploits their properties to provide the facility for local modification of curves and surfaces.

An increasing interest is now being shown in the 'fairness' of curves. Since a curve can be very smooth in the mathematical sense, but a great many undulations may still occur. S chweikert[33] found a Spline in Tension and Mehlum[34] used a nonlinear spline in the AUTOKON system. Those spline functions are less oscillatory than the earlier splines.

1.7 Growth of the Industry

In the early 1970s there were only about 200 CAD/CAM workstations installed in the world[1], and almost all were in-house systems in large aerospace and automotive companies that could afford them.

With the advent of the minicomputer in the mid-1970s, the cost of these systems decreased dramatically and the use of CAD/CAM took off. Since then, sales have quadrupled. By 1976, it was generally agreed that the total computer graphics market was about a billion dollars[35]. As 1980 began, computer graphics has grown to a 2 billion dollar industry with some 180,000 graphics devices in place, with a user community of perhaps 250,000 and with some 40,000 people employed. Sales volume in 1984 alone was estimated to be 8000 million dollars[36]. CAD/CAM ranks among the fastest growing sectors of the economy.

Now over 90 percent of all mechanical drafting is done by CAD, and about 30 percent of all manufacturers use some form of CAM, the number of CAD/CAM installations is growing by more than 40 percent a year.

1.8 World competition in general-purpose digital computers and CAD/CAM systems

Among manufacturers of large- and medium-scale general-purpose digital computers, IBM Corp. is the top contender[37]. In 1979, IBM had an international sales volume of 18 billion dollars in both hardware and software - about 50% of the world's computer business.

During the 1960's, high-priced, IBM system 360-series computers dominated the international market with about 85% of the computer business. But during the first half of 1970's, the more advanced IBM system 370-series main-frame computers had to compete for the first time with the MSI-based minicomputers; and the Digital Equipment Corporation (DEC) took the lead in international medium- and small-scale computer operations. During the second half of 1970's, the lower-cost minicomputers and currently low-cost microcomputers qualitatively and quantitatively expanded the computer business a hundredfold. IBM has oriented itself to lower-cost, highly powerful main-frame computers such as 4300-series and a special model 8100. For medium-and small- business applications, IBM turned its emphasis from its System 3-series computers to the Superminis, such as 4331 and 4341, in order to compete with the minicomputer manufacturers on a more or less equal footing.

In the 1980's, the major contenders in the general-purpose minicomputer area are DEC, Hewlett-Packard Co., Data General Corp. and Honeywell Information Systems.

Prime Computervision (CV) is currently probably the world leader in the number of minicomputer CAD/CAM systems installed. Intergraph, Applicon and McAuto are also major international minicomputer CAD/CAM system suppliers[38]. They provide powerful software packages which run on minicomputers, including 2D draughting, 3D draughting, 3D modelling, FEA (Finite Element Analysis), NC programming, robotics, simulation, etc.

IBM PC and micro VAX-based systems are leading products in the field of PC based CAD/CAM systems, with many other host computers, e.g. SUN, Apollo, IBM 'look-alike', etc., rapidly developing the market. These systems are capable of running simulation applications, high- resolution graphics and the UNIX operating system, all at a relative modest price.

Chapter Two

CAD/CAM Systems and Applications

2.1 Summary

In the previous chapter, the historic background of CAD/CAM has been reviewed. In this chapter, the general hardware and software of current CAD/CAM systems will be introduced, including input devices, output devices, elements of a workstation, the types of computer configurations, interfaces, databases, NC technology and computer assisted part programming. Typical application software is also briefly described, such as three dimensional geometric modelling, finite element analysis, computer aided drafting, process planning and industrial robotics.

2.2 Hardware for CAD/CAM

CAD/CAM involves input hardware, processing hardware and output hardware. Because design information is frequently encapsulated in the form of drawings, input and output hardware capable of accepting information from drawings, or able to present information pictorially, is of special interest in CAD/CAM applications. CAD/CAM is by definition an interactive process and, consequently, the hardware used must support man/machine interaction[39].

2.2.1 The workstation and its input devices

Most of the equipment of a CAD/CAM system is grouped in an arrangement called a WORKSTATION. This is the physical cockpit where the user controls and manipulates the vast amount of unseen data and software residing in the computer. It is the engineer's vehicle for communication with the CAD/CAM system. A workstation consists of input and output devices. The output devices, however, can be shared by several workstations in the same CAD/CAM system.

The central item in a workstation is the GRAPHICS TERMINAL. This has a TV-like display that may be either Refresh, Raster, or Storage Screen depending on the system. It is both an input and an output device as instructions generated by the user may result in some change to the displayed image. The Cathode Ray Tube (*CRT*) is the basic element around which most display terminals are constructed[40].

A KEYBOARD is also an integral part of the graphics terminal. This is a type-writer-like alphanumeric arrangement for entering commands, text, and parameter values. Other input devices in the workstation are as follows:

Tracking devices provide a means of graphical input and are off-screen devices which depend on visual feedback for their effectiveness. Types of tracking devices are THUMB WHEELs, JOYSTICK, TRACKER BALL and TABLET. Feedback is achieved by the display of a cursor, normally in the form of a cross on the screen. They can be used either for positioning geometric entities or for selecting them prior to some action being performed on them.

The TABLET is a convenient way for the user to enter X and Y locator positions. The tablet stylus is held like a pencil and moved over the tablet surface. Downward pressure on the stylus closes a microswitch and interrupts the computer. Hence the tablet can serve as a sampled locator with a single event-causing button[41].

The TOUCH-SENSITIVE PANEL and LIGHT PEN are attached to a visual display, and the user registers a position by touching the screen surface directly. Unlike the touch-screen, light pen and tablet, the MOUSE does not normally require a special surface. It signals how far it has moved from its last position, rather than where it is in relation to some fixed surface. However, the optical mouse is used in conjunction with a special surface.

The DIGITIZING BOARD is a common graphics input device in a computer aided draughting system, which enables the direct input to a CAD system, of data from an outline drawing in the form of digital co-ordinates. Not all systems will support this activity, therefore if it is to be an important feature of the proposed system the purchaser should check this point with the supplier.

2.2.2 The output devices

The PLOTTER is a graphics output device and is the means of obtaining a high quality hard copy representation of geometry stored in the CAD system's database. It is a necessary piece of equipment for a draughting system, since such a system's main purpose is the production of engineering drawings. PRINTERs and COLOUR PRINTERs are also important output devices used for producing hard copy of the image on the screen.

2.2.3 The computer hardware

Computers for CAD purposes can range in size from the large, general purpose, mainframes down to modem microcomputers. They are generally identified by their word length, i.e. 8-bit, 16-bit, 32-bit etc. The word length represents the size of the data bus or the number of bits that can be transferred around the computer system in parallel. The longer the word length, the more complex the CAD system that can be operated. Until a few years ago, microcomputers had 8-bit, minicomputers had 16-bit and mainframes had 32-bit word lengths. However, the trend is changing as 16-bit micros, and 32-bit minicomputers are now commonplace. Recently, 32-bit microcomputers are available and in use, such as the CV386. This means that computer aided design tasks which traditionally had to be carried out on expensive mainframes can now be run on cheaper mini- and microcomputers[42].

There are a number of basic ways in which the computer hardware can be configured for CAD use. Three of them will be identified.

i) CENTRALIZED SYSTEM: A centralized system would be based on one of the larger mainframes capable of supporting twenty or more simultaneous users. as shown in Figure 2.1. Following the initial capital costs, additional terminals may be added on at a relatively low cost. However, if too many users require simultaneous access to the computer, the response time may become irritatingly slow.

ii) MULTIPLE HOST SYSTEM: This configuration is usually suitable for minicomputers. A number of minicomputers each have limited number of terminals connected to them. The minicomputers are also interconnected so that the individual user appears to have the same facility as a centralized system (Figure 2.2).

The distinct advantage of this type of system is that if one computer were to go down then only a small number of terminals would be affected. However, The capital outlay for a multiple host system could be greater then for a centralized system.

iii) DISTRIBUTED SYSTEM: A distributed system is one in which each terminal has its own computer. The computers are connected by a network arrangement to form a integrated system. The computers could be micros, minis, super-minis or even mainframes, and users of the network system can share common peripheral devices such as disc drives , plotters and printers as shown in Figure 2.3

2.3 The database

For an engineering company with any computer-aided design equipment, a database is a desirable tool, particularly if the intention is to eventually integrate the total CAD/CAM activities.

A database is a generalized integrated collection of data which is structured on natural data relationships so that it provides all necessary access paths to each unit of data in order to fulfil the differing needs of all users of the system. In a database system the data is stored once only. The different application programs (such as modelling, draughting, numerical control) can access the same database via a set of controlling programs known as the database management system.

Figure 2.4 shows how both design and manufacturing activities can share a common database.

2.4 Interfaces and inter-system communication

All CAD/CAM systems require INTERFACEs which link the computer with other elements of the workstation. An interface has two elements, a hardware element, i.e. electrical components, and software elements to control the information and detect errors. There are two basic methods for interfacing the computer to peripherals, a SERIAL connection using RS232 or RS422 interfacing link, or a PARALLEL connection for data interfacing using a parallel interface. The latter gives very high transmission speeds since the information for one character is sent down a number of parallel wires simultaneously.

In order to enable geometric data to be readily exchanged between different CAD systems in digital form, a data interchange specification is used. The Initial Graphics Exchange Specification (IGES)[43] has been designed to present a standardized form for draughting and geometry entities. It is a project of the U.S. Air Force Integrated Computer Aided Manufacture (ICAM) project. The IGES format includes entities describing geometry (point, line, arc, spline, etc.), annotation (dimensions, drawing notes, etc.), and structure (properties, associations, groups, etc.). The transfer of a CAD database is a two step process. First, the sending system uses its IGES preprocessor to translate the entities comprising its database into the neutral or canonical IGES format. Then the receiving system uses its IGES postprocessor to translate the IGES entities into its own internal format.

The Association Europeenne des Constructeurs de Materiel Aerospatial (AECMA) has produced a recommendation[44] for an interface between geometry systems. The AECMA format enables the receiving system to do everything with a curved surface except modify it.

2.5 NC, CNC and DNC

The most mature of the CAM technology is Numerical Control (NC). This is the technique of controlling machine tools with prerecorded, coded numerical information[45].

In the most basic NC system the programmed instructions are stored on punched paper tapes and interpreted by electromechanical tape readers connected to the machine tool.

More advanced systems use Computer Numerical Control (CNC), in which the machine is controlled by a dedicated computer with the NC instructions stored in its memory. The stored instructions can be easily and quickly modified providing a high order of machine versatility. CNC systems, which usually have feedback control, have many advantages over conventional NC systems, namely, greater reliability and flexibility of operation, provision of management information (i.e. machine status, downtime, cutting time). Tool positioning and cutter operating regimes can be tightly controlled in order to minimize tool wear, and to optimize machining. A CNC machine tool which can perform a range of activities is referred to as a MACHINING CENTRE.

More sophisticated systems may use Direct Numerical Control (DNC), which may also be referred to as Distributed Numerical Control when the machine control is distributed among a network of computers. In these systems the individual manufacturing units are connected through communication lines to a central computer that is in command. The communication lines also provide for the feedback of production and machine tool status from the shop floor. Some systems use a hierarchical arrangement of computers connected to the central computer, while other systems have eliminated the intervening mini- or microcomputers and have direct interfaces between the mainframe and machine tools.

A major advantage of the DNC system is the increase in efficiency, since the processor's input/output equipment is located in a nearby area.

2.6 Computer-Assisted Part Programming (CAPP)

In the early days of NC and CNC, the parts to be machined were of two-dimensional configurations requiring simple calculations to describe the tool paths.

With the increased use of NC and CNC systems and the growth in the complexity of parts to be machined, the part-program is becoming more difficult to write and it is less easy for the part programmer to calculate the tool paths efficiently.

These limitations can be overcome by using the computer to assist with the part programming process. Computers can perform the required mathematical calculations quickly and accurately, and the computation errors which arise with manual calculations are reduced with Computer-Aided Part Programming (CAPP).

There are a wide variety of computer-assisted part programming languages available, such as APT, ADAPT, UNIAPT, SPLIT and CINTURN II. These languages relieve the programmer from manually entering any geometry data. APT (Automatically Programmed Tools) is the original and more universal language. It has become the standard of the NC industry.

In a typical computer assisted system, when the NC program is selected, a menu appears on the screen, and the user designates the machine which is to cut the part. When the machine is selected, another menu appears, allowing selection of:

i) Cutter size

ii) Type of material

iii) Feeds and speeds

iv) Material thickness

v) Clearance above which the tool must rise when not cutting.

Then the screen displays a geometric model of the part, on which the programmer defines the machining operations using various types of lines. For example, a solid line might indicate a rough cut and a dashed line a finish cut. From these inputs the system automatically generates an NC tape to machine the part.

Most of the NC programming packages also simulate the tool paths so that the program can be checked. The tool motion can be animated to allow the programmer to observe the tool as it moves on the part. These features allow a verification that the program properly guides the cutter.

After the programming is completed, the computer translates or postprocesses the tool path description into coded instructions tailored to a specific machine. These instructions are stored on punched paper tape or magnetic tape for input to the machine tool. The instructions may also be transferred directly to a memory in the machine tool control system.

Beyond computer assisted .programming, the software may use the geometric model created in CAD system as a basis for producing the NC instructions. The geometric information is accessed through a shared database in the computer, thus the programmer does not have to enter data manually. The computer may also prompt the programmer to respond to questions displayed on a terminal screen.

These NC programming packages can also display the simulated tool paths so that the programmer can check the program. The system allows the programmer to create NC instructions graphically, without requiring a detailed knowledge of programming languages.

2.7 CAD/CAM functions and applications

A user at a CAD graphics terminal can design a part, analyse stress and deflections, study mechanical action, and produce engineering drawings automatically. From the geometric description provided by CAD, production people may produce NC instructions, generate process plans, program robots, and manage plant operations with CAM systems.

These two technologies and their associated capabilities are being combined into integrated CAD/CAM systems. In other words, a design is created and the manufacturing process is controlled and executed with a single computer system. Generally, CAD functions are grouped in four categories: geometric modelling, engineering analysis, kinematics, and automated drafting. And present activity in CAM technology centers around four main areas: numerical control, process planning, robotics, and factory management. Some of the CAD/CAM features will be detailed below.[1]

2.7.1 Three-dimensional geometric modelling

The designer constructs a geometric model on the CAD/CAM terminal screen to describe the shape of a structure to the computer. The computer then converts this pictorial representation into a mathematical model which it stores in a database for later use. The model may be recalled and refined by the engineer at any point in the design process. And it may be used as an input for virtually all other CAD/CAM functions.

Three dimensional objects can be modelled using one of the three methods: WIREFRAME MODELLING, SURFACE MODELLING and SOLID MODELLING.

A wireframe model of an object is made up of a set of three- dimensional co-ordinates, which define the end points of lines in space. Information about the type of line is also held (i.e. the curvature of the line.). This type of model, because it is described by its edges and vertices, can only provide partial information. The model cannot be used to perceive any information about the volume of the object or indeed which lines represent the inside or the outside of the component.

Surface models hold the description of an object in terms of points, edges and faces between the edges. Surface modelling systems are able to calculate surface intersections and areas and some systems are capable of producing shaded images and removing hidden lines automatically. Surface models are ideally suited to applications where a model of a complex surface is required, such as aerospace and car body styling. From the geometry of a surface model, numerical control path data can be generated. All machining operations, including multi-axis profiling can be catered for, allowing complex surfaces to be machined.

A solid modeller is probably the most powerful of the three dimensional modelling techniques, as it provides the user with complete information about the outline, surface, volume and mass properties of the model. A solid modeller holds a complete description of an object, in terms of the space which it occupies. Solid modelling systems can produce very impressive visual displays of objects, using perspective, colour shading and high-lighting to create images of photographic quality.

2.7.2 Engineering analysis

Most CAD/CAM systems permit the user to move directly from the geometric model to analytical functions. For example, single keyboard instructions may command the computer to calculate weight, volume, surface area, or centre of gravity of a part.

The most powerful method of analysing a structure on a computer is probably the FINITE-ELEMENT method. In this technique, the structure is represented by a network of simple elements that the computer uses to determine stresses, deflections, and other structural characteristics.

In integrated CAD/CAM systems, the user can call up the geometric model of the part and create a finite element model quickly and easily using automatic node-generation and element- generation routines. Once a part is modelled, the user specifies loads and other parameters. Then the model may be analyzed with programs such as NASTRAN, STRUDL, ANSYS or SAP.

If the finite-element analysis reveals excessive stress or deflection, the computer model may be modified and reanalyzed. In this manner, the designer can evaluate structural behavior before the product is actually built, and the design can be appropriately modified without building costly physical models and prototypes.

2.7.3 Computer-aided drafting

Computer-aided drafting features automatically produce detailed engineering drawings on command from the geometric model database or from inputs entered by the user at the graphics terminal.

These systems have many features that automate a range of drafting tasks to speed the production of drawings. Most systems have automatic scaling and dimensioning features, and changes made to one view may be automatically added to other multiple views. Moreover, function menus permit the user to specify points, locate lines, enter text, and produce cross-hatching at the push of a button. These automatic features coupled with the high speed of computer-driven plotters enable users to produce new drawings up to five times faster than with manual drafting methods.

2.7.4 Process planning

Numerical control is concerned with controlling the operation of a single machine. However, process planning is a much broader function that considers the detailed sequence of production steps required to fabricate an assembly from start to finish.

One important aspect of process planning systems is group technology. This concept organizes similar parts into families to enable fabrication steps to be standardized.

Most process planning systems now use retrieval techniques. In this approach, process plans are developed for part families from existing databases on standard tooling and fabrication processes. One of the first of these systems was the CAM-I Computer Automated Process Planning (CAPP) system.

Retrieval systems are far less sophisticated than the future generative process planning systems which many experts predict. Generative process planning systems currently under development by CAM-I would produce process plans directly from the geometric model database with almost no human assistance. The process planner would receive the input from design engineering on a CAD/CAM terminal and then enter this data into the computer system, which would automatically generate complete plans.

2.7.5 Robotics

Robots are automated manipulator arms that perform a variety of material-handling tasks in a CAD/CAM system. Robots may select and position tools and workpieces for NC machine tools, or they may carry equipment or parts between various locations on the shop floor. They also may use their mechanical hands to grasp and operate drills, welders, and other tools.

Most robots presently are programmed in a so-called teach mode. In this approach, a user physically leads the robot through the individual steps of an operation. This type of manual teach-programming is time consuming and error prone. Also program changes usually require the entire sequence of steps to be repeated.

To overcome these difficulties and extend robot capabilities beyond the traditional teach mode, present research in robotics is directed at developing advanced programming languages with which robot instructions may be issued through a computer. One of these advanced languages presently under development is the IBM Automated Parts Assembly System (AUTOPASS).

Chapter Three

Introduction to the Project

3.1 Summary

In the previous chapters, the background of CAD/CAM has been reviewed and typical CAD/CAM systems and their general applications have been introduced. In this chapter, the project of this thesis which is aimed at the development of a method of modelling and NC machining of free form surfaces using a PC based CAD/CAM system will be detailed.

This chapter starts with an introduction to the current problems with the modelling and manufacturing of components with free form surfaces and the importance of the project, followed by the description of the objectives of the project and the equipment used throughout the project. The capabilities and limitations of the software used and the programming language are discussed, e.g. microCADDS Geometric Constructing and Detailing (GCD) and Surfaces, Personal Machinist (PM) and User Programming Language (UPL). Finally, the general procedure of modelling a component with microCADDS and NC part programming and machining of the geometric model with Personal Machinist is introduced and illustrated by two flowcharts. The additional programs developed during the project are also pointed out where they are used in the flowcharts.

3.2 Introduction

Sculptured surfaces are made up of arbitrary, nonanalytical contours that may not obey mathematical laws. These surfaces are found in a wide range of components, including those used for aircraft, automobiles, construction and agricultural equipment, machine tools, and domestic appliances. The modelling and NC machining of these surfaces has been a major difficulty in CAD/CAM applications.

Up to now, the modelling and NC machining of free form surfaces is mainly accomplished on mainframe Computer Aided Design and Manufacturing (CAD/CAM) systems such as those supplied by Computervision (CV). Recent development in computer hardware and software resulted in the use of 32-bit super minicomputers for this purpose. Such as Prime, DEC-VAX, Apollo, Data General, ICL Perq, Sun, Control Data, IBM and Whitechapel. Product geometry (or the computer model) can be manipulated, modified and refined with powerful surface modelling packages such as CADDS 4X advanced surface modelling software supplied by the Computervision[46] [47] [48] and DUCT (Design Using Computer Techniques) supplied by Delta Computer Aided Engineering Limited[49] [50]. NC data can be obtained from these systems via 3-axis or 5-axis machining packages[51]. However, the cost of such packages and systems precludes the use of these in smaller companies.

With the appearance and development of 16-bit and 32-bit micro-computers (e.g. IBM PC AT and CV386), CAD/CAM functions are being run on these low cost systems, such as Computervision MicroCADDS GCD[52] and Surfaces[53] and Personal Machinist[54]. However, the CAD packages on personal systems are currently not sufficient to model and manipulate complex surfaces. The NC machining packages are generally limited to 2-1/2 axis machining for all but special applications such as 4-axis wire Electro Discharge Machining (EDM)[55].

Crippa and Faltoni have done a lot of surface modelling work on PC based CAD systems for the plastics industries[56], and Takeuchi, Morita and Sakamoto[57] developed a Personal CAD/CAM system named MCSYS. They used wire frame models to represent objects so that toolpaths can be easily generated. The cutting direction, however, is limited to one and the stepovers are constant. This may result in low accuracy when the component is composed of complex surfaces.

3.3 The objectives of the project

Since the surface modelling facilities on PC based CAD systems are not powerful enough, i.e. lack of smoothing, filletting and matching facilities and the CAM systems are limitied to 2-1/2 axis machining, this project is aimed at the development of additional programs which make it possible to model and NC machine 3D free form surfaces with Personal Computer (PC) based CAD/CAM softwares currently available. The methods have been tested by several examples, i.e., the modelling of a telephone handset from digitized coordinate data of a master part and a plastics bottle modelled from a 2 dimensional engineering drawing, The bottle mould and a scaled cab roof model have been NC machined.

This successful experiment is of particular importance to smaller sized plastics mould design and manufacture industries such as, injection moulding and blown moulding. To satisfy this particular usage, components can be changed in shape and volume quickly with the programs developed in this project.

3.4 The equipment used in this project

The computers used in this project are 16-bit IBM PC-ATs and 32-bit CV386s with colour graphics terminals, the operating system is the Disc Operating System (DOS). Generally, all the work can be carried out on 16-bit microcomputers. The use of 32-bit computers increases the running speed and capacity, and this is the future of CAD/CAM. The computers used are arranged in a distributed configuration with HP7475A pen plotters and EPSON FX-105 line printers connected to the network. The CNC machine used is a Takisawa Machining

Centre with FANUC MAC-V3 controller and a RENISHAW probe system. The NC data is transferred from a workstation to the CNC machine controller via a DNC (Distributed Numerical Control) link. This is done by a dedicated Personal Computer and a cable which links the PC and the controller of the machining centre by RS232 connectors.

Usually a Coordinate Measuring Machine (CMM) is required to digitize master parts for input information into the CAD system. In this project, the RENISHAW probe system installed on the Takisawa Machining Centre was used for this purpose.

Figure 3.1 shows the network of the microcomputer CAD/CAM system. Geometric and NC data can be directly transferred between workstations via the V8 PC CLUSTER controller or floppy discs. Geometric information can also be transferred to the PC CAD/CAM system either from the Computervision CADDS 4X system by the binary converter or from other systems by IGES (Initial Graphics Exchange Specification).

3.5 The CAD package used and its limitations

The design software used is Computervision Personal Designer, which consists of two fully integrated packages, namely, microCADDS Geometric Construction and Detailing (GCD) and microCADDS Surfaces. MicroCADDS GCD is at the core of every Computervision Personal Designer system. It provides extensive construction and editing tools for creating three-dimensional models of complex geometry and two-dimensional layouts and drawings. It can be used for design studies or for creating engineering layouts quickly and easily.

MicroCADDS Surfaces option for the Personal Designer is an integrated extension to the wireframe design and drafting capabilities provided by microCADDS GCD. 2 dimensional and 3 dimensional curves and surfaces can be easily created and both planar section and the intersection of complex surfaces can be calculated.

For curve and surface representation, microCADDS Surfaces uses Bezier mathematics, an approach that was refined and performed in the most advanced Computervision systems. The powerful command set insulates the user from the underlying mathematical complexity while providing an easy-to-use, flexible, and efficient mechanism for defining and manipulating the geometry. There are four types of surface definitions i.e. Bezier surfaces, Ruled surfaces, Coon's surfaces and surfaces of revolution.

However, surface offsetting, surface filleting, surface smoothing and surface matching are not available in Personal Designer. These are very important features for free form surface modelling. The definitions of surfaces are also limited, some advanced surfaces are not available, such as, B-spline surfaces, Rational surfaces and Non-Uniform B-spline surfaces. Personal Designer is not well integrated with Personal CAM packages, and NC toolpaths can not be easily created from the geometric model.

3.6 The User Programming Language (UPL)

User Programming Language (UPL) is a compiled modular programming language specifically designed to run and interact with microCADDS GCD and Surfaces or Advanced Architectural Drafting software[58]. The language has all of the most commonly used features of other high-level languages, such as FORTRAN or Ada. Several additional features are incorporated which are oriented towards interacting with GCD or Advanced Architectural Drafting, both at the command input level and directly with the drawing data base. Several built-in procedures and functions are included specifically for CAD applications.

The limitations of UPL are: information related to the curves and surfaces can not be obtained directly, curves and surfaces can not be inserted and created by running programs written in UPL.

3.7 The Personal Machinist and its limitations

The Personal Machinist (PM) system joins CAD and CAM together on a single system. Because it utilizes inexpensive Personal Computer technology, the PM system is available at a price which individual departments and small job shops can afford.

With the PM system, the engineer can design three-dimensional geometry, extract key geometry information, and build the NC manufacturing programs required to produce parts, tooling, and fixtures. The PM is compatible not just with a selected group of machine tools, but with a wide range of popular machine tools, including milling machines, lathes, punch presses, electrical discharge machines, and flame/laser cutting equipment.

3.7.1 The NC Processor (NCP)

The Personal Machinist NC Processor (NCP) is a two and half (2-1/2) axis NC tape preparation system designed to use microCADDS model data base information as input geometry, and to use the NCP for toolpath graphics display and verification.

Computer Assistant Part Programming (CAPP) technology is used the NCP so that the part programs are generated automatically from the defined or transferred geometry. This frees the part programmer from tedious manual part programming and geometric calculation work. The NC system is designed to generate NC machining data from a microCADDS model. However, the NCP has its own geometric construction capabilities, and it is not mandatory to use a microCADDS model.

The limitations of the NC Processor are: only points, lines and arcs can be defined in the NCP or transferred from a microCADDS model. Curves, strings and surfaces can not be converted from the microCADDS model, or constructed in the NCP. Arbitrary 3 dimensional profiling is not available in NCP. 3D machining is restricted to the case where the horizontal cross-sections of the side surfaces are different in size, but similar in shape. see Figure 3.2. In most cases, the machining is limited to two and half axis, which allows the tool to move freely and simultaneously in two dimensions while keeping the third dimension fixed for each operation. However, the toolpath translation, rotation and mirroring features of the NCP can be used to decrease the shortage of machining axes.

3.7.2 NC Postprocessor and NC Postprocessor Generator (PGEN)

The Postprocessor Generator (PGEN) is a flexible tool included with the PM NC system that allows users to use question/answer prompts to create machine specific postprocessors to generate an NC tape image from an NC part program file (with a .NC1 extension) for a particular NC machine-control combination. Since each NC/CNC machine has its own particular characteristics, a tape for one machine will almost always be different from a tape for another machine. The postprocessor is the software that formats output information for an individual machine's requirements.

Developing a postprocessor is a three step process executed at the DOS level.

Step 1: Generate the postprocessor source file.

The postprocessor source file is an ASCII text file, which can be generated by running PGEN, the postprocessor generator program, to generate a machine specific postprocessor. PGEN prompts the user with a series of questions about the NC machine and subsequently generates a source postprocessor with a .NC2 extension that can be compiled and used to post-process a part program.

Step 2: Edit the postprocessor.

This is an optional step that we might want to take if we have a postprocessor that requires additional changes or enhancements. Because we edit the ASCII text file directly, knowledge of the basic programming language is required.

Step 3: Compile the postprocessor.

The final step is compiling the ASCII text file version of the postprocessor (a .NC2 file) into a binary format (a .NC9 file)

post .NC2 ⟶ PC ⟶ post.NC9 (compiler)

In this way a postprocessor is created to postprocess a part program in the NC Processor to generate an NC tape file. A tape file stores the NC program output in a format that can be understood by the machine controller.

part program .NC1 ⟶ post .NC9 ⟶ tape file .NC2

Figure 3.3 illustrates the three steps in developing a postprocessor at DOS level. Figure 3.4 shows how postprocessing is accomplished in the NCP processor, from an NC part program into an NC image ready to drive a machine tool.

3.8 The general procedure for designing and manufacturing a component embodying free form surfaces

The general procedure necessary to model and manufacture a component with complex surfaces, such as a mould, via Personal Computer CAD/CAM system is set out in Figure 3.5 and Figure 3.6. The modelling and NC part program generation of a required part will be discussed separately.

3.8.1 The creation of geometric model of a component to be machined

The geometric model of a component can either be originated from digitized information taken from a master part by Coordinate Measuring Machine (CMM) or modelled using 2 dimensional (2D) engineering drawings with microCADDS. It may also be transferred to microCADDS by either the binary converter from the Computervision CADDS 4X system or via IGES from other systems.

If the computer model is created from a master part with microCADDS, the following steps are taken. (See Figure 3.5).

i) Measuring the master part on an CMM or an CNC machining centre with a *RENISHAW Probe* system. The number of points to be digitized should be pre-decided and the positions of these points must be planned. Otherwise the resulted surface patches may be badly arranged and consequently, further manipulations of the surface patches will be very difficult.

ii) Input the measurements manually into microCADDS and display the points on the computer screen. This could be done automatically with a prepared interface between the Coordinate Measuring Machine or the CNC machining centre and the CAD system.

iii) Generate a set of curves from the points. The curves will not necessarily pass through every point , since the measurements have errors. The degrees of the curves should be carefully selected. Usually a number of low degree composite curves are used.

iv) Generate surface patches from the curves. The number of the patches and the orders of the surface patches should be carefully selected. Usually low order composite surface patches are used.

v) Smooth and refine the surfaces. Special methods have been developed to achieve this because of the shortage of these facilities in microCADDS. The program developed for smoothing surfaces is "SMOSPL.UPL".

If the computer model is created from a 2D engineering drawing supplied by a customer, the following steps are necessary:

a) Draw the three main views (TOP, RIGHT and FRONT views) of the part in the drawing using microCADDS GCD. Put them on different layers. (Layers are used by microCADDS to separate drawings). The datum of the elements in the three views must be the same.

b) Draw any other section views of the part in the engineering drawing and put them on different layers. The orientation of the section views in space must correspond to the main views.

c) Project geometric entities (lines, points, arcs and curves) in the TOP view so that the orientation and position of these entities are correct in space. This is done by first projecting the entities in the TOP view until they are corresponding to their RIGHT view in space and then projecting them until they are corresponding to their FRONT view in space.

d) Use the projected entities in the TOP view (which are orientationally and positionaly correct in space) and the entities of any other section views to build a wireframe model of the part. Additional entities may be created by the designer to complete the wireframe model. However, the entities in the RIGHT and FRONT views are not used for building the wireframe model.

e) Put surface patches on the wireframe model. The number of patches and the types of surface patches should be carefully considered.

f) Smooth and refine the surface model of the part. The method of smoothing and refining is the same as that already mentioned.

3.8.2 The procedure for NC machining the component using the computer surface model

Once the model has been built in microCADDS or transferred from other systems, the geometry is separated for machining and this is typically achieved by layering (See Figure 3.6). The surface types are then identified and using the program "BEZOFF.UPL" developed in this project, the offset points are generated to produce the offset surfaces. These are offset from the original shape by an amount equal to the radius of the ball-ended milling cutter to be used for cutting the mould. The tool size is determined by the curvature of the surface to be cut. The program "RADIUS.UPL" was used to calculate the minimum radius of curvature of a concave surface.

The offset surface is then sectioned using the program "CUSPH.UPL" at the given step sizes which generates curves for toolpath geometry. Since these curves can not be transferred to Personal Machinist they have to be exploded to form line segments. This is done by running another program which has been developed in this project, "LINPATH.UPL" to generate a limited number of lines along the curves, the maximum deviation between the curve and each line segment being controlled. Machining groups are then formed from each chain of line segments.

Once the above tasks have been completed within the microCADDS part (.DRW), the Personal Designer to Personal Machinist interface (*PDPM*) UPL program can be run to capture relevant information about the conversion process for use by the batch file *CONVERT.BAT*. The user then exits from microCADDS back to DOS level where CONVERT.BAT is actioned to perform some housekeeping tasks and to execute the PDPM utility which undertakes the actual conversion. Finally, CONVERT.BAT copies the converted file (.NC5) into the Personal Machinist directory and takes the user into the NC Processor (NCP).

Once in the part program (.NC1) the first step is to merge in the converted geometry (.NC5). The machining groups can also be defined at this stage. Currently, within PM all merged geometry is displayed in the XY plane. If it is necessary to machine in the YZ or ZX plane then translation and rotation are carried out at this stage. Next the tool motion statements are added, together with speeds, feeds, tool types, etc. to perform the machining. It should be noted here that by generating an offset surface at the outset the ball-ended tool diameter is zero as the compensation is already catered for.

During the build up of the tool motion statements the results can be monitored via a graphical post processor. Once this has proved satisfactory the part program is processed through TAKMET to produce the NC code (.NC2). TAKMET is the UMIST post processor for the Takisawa Machining Centre which is fitted with a Fanuc 6MB Controller. This NC code part program is then transferred via the Local Area Network (LAN) to the machine tool controller using the DNC command which uses a machine parameter file (MPF) to perform the handshaking between the computer and machine tool controller.

The Post-Processor TAKMET was generated in house using the PM Post-processor Generator (PGEN) and then further adjusted using the ED editor. PGEN creates a question and answer file (TAKMET.NC3) and a post processor written in NC BASIC (TAKMET.NC2) which may be edited via ED. To create a workable post processor the (.NC2) file has to be compiled via the post compiler (PC) to give the binary (.NC9) post processor, which is accessed via the NC Processor (NCP).

Chapter Four

Generating Surface Models from Digitized Data

4.1 Summary

In the last chapter, the general procedure of modelling and machining free form surfaces has been briefly introduced. In this chapter, the method is described in detail. The mathematics of the surface definitions in microCADDS will be first discussed, followed by the introduction of the approach to modelling surfaces from digitized data from measuring of a master part to the refining of the resulting surface model. A new method of interpolating digitized data with Bezier surface patches and the smoothing of composite surfaces will be described. The chapter concludes with an assessment of the major advantages and drawbacks of this method.

4.2 Introduction

The information used for constructing surface models in an CAD system usually comes from two sources, i.e. digitized data from a master part or prototype or from two dimensional orthographic engineering drawings. In this section, the approach to modelling free form surfaces using digitized data will be described in detail, leaving the second approach to the next chapter.

The general procedure of manufacturing components with complex surfaces such as ship hulls, car bodies or aircraft fuselages is, first designing the shapes of the desired components by experienced loftmen manually or by using modern computer packages such as CONSURF as used by British Aircraft Corporation (Warton)[62] [63] [64]; then building prototypes of the surface made of wood or wax by skilled people; when the prototypes satisfy all the constrains, they are digitized by a Coordinate Measuring Machine (CMM) or a digitizer. The digitized data is then transferred to a CAD system and the surface models are constructed. When the surface models are refined, NC machining cutter paths and part programs are created from the surface model with an integrated CAM package. Finally, the components are manufactured on a CNC machine using the NC programs generated by the CAM package.

Since the purpose of this project is to find a method of modelling and NC machining of free form surfaces with PC based software, the complete procedure described above was not followed. Instead, a telephone handset was digitized on the CNC machining centre and a surface model was built using the digitized data using a Personal Computer.

4.3 The surface definitions in the microCADDS

It has been mentioned in the last chapter that the microCADDS Surfaces package is based upon Bezier mathematics, which is a mature approach used by most CAD packages. Although most advanced CAD packages, particularly those running on mainframe or minicomputers, have more advanced and sophisticated surface representations such as Rational B-spline surfaces[59] and Rational Bezier surfaces[60] [61], Bezier curves and surfaces are still their basic curve and surface definitions. MicroCADDS also contains other types of basic surfaces such as ruled surfaces, Coons surface patches and surfaces of revolution.

4.3.1 Bezier curves

A Bezier curve can be defined by a set of control points called Bezier points or vertices. This approach defines a curve which only approximates to these given control points, rather than passing through them. For example, the Bezier cubic curve is determined by two end point vectors $\mathbf{V}_0$, $\mathbf{V}_3$ and two control point vectors $\mathbf{V}_1$, $\mathbf{V}_2$, which do not lie on the curve. (See Figure 4.1). A Bezier curve of degree n is determined by a set of control points $\mathbf{V}_i (i = 0, 1, 2, ..., n)$, which form a characteristic polygon called the Bezier polygon. It can be generally expressed as[38]:

$$\mathbf{r} = \sum_{i=0}^{n} B_{n,i}(u)\ \mathbf{V}_i \tag{4.1}$$

where $B_{n,i}(u)$ is the Bernstein function,

$$B_{n,i}(u) = C(n,i)\ u^i (1-u)^{n-i} \tag{4.2}$$

$C(n,i)$ is the binomial coefficient,

$$C(n,i) = \frac{n!}{i!\ (n-i)!} \tag{4.3}$$

u is the parameter of the curve and $0 \le u \le 1$.

A cubic Bezier curve can be derived from equation 4.1, 4.2 and 4.3 when n=3,

$$\mathbf{r}(u) = (1-u)^3\ \mathbf{V}_0 + 3\ u\ (1-u)^2\ \mathbf{V}_1 + 3\ u^2(1-u)\ \mathbf{V}_2 + u^3\ \mathbf{V}_3 \tag{4.4a}$$

In matrix form, it can be expressed as:

$$\mathbf{r}(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{bmatrix} \tag{4.4b}$$

or in simplified form:

$$\mathbf{r}(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{bmatrix} \tag{4.5}$$

where

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

The command used in microCADDS to generate Bezier curves is "INSert CPOle", followed by modifiers. CPOLEs are names of curves given by Computervision. Modifiers are used to specify the features and properties of a geometric entity.

4.3.2 Bezier surfaces.

The definition of a general Bezier surface is shown by the illustration of the formation of a bi-cubic surface. First, 16 control vertices $\mathbf{V}_{i,j}$ (i=0, 1, 2, 3; j=0, 1, 2, 3) are selected which form a (4 × 4) matrix which defines a characteristic polyhedron in a geometric space. (See Figure 4.2).

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{00} & \mathbf{V}_{01} & \mathbf{V}_{02} & \mathbf{V}_{03} \\ \mathbf{V}_{10} & \mathbf{V}_{11} & \mathbf{V}_{12} & \mathbf{V}_{13} \\ \mathbf{V}_{20} & \mathbf{V}_{21} & \mathbf{V}_{22} & \mathbf{V}_{23} \\ \mathbf{V}_{30} & \mathbf{V}_{31} & \mathbf{V}_{32} & \mathbf{V}_{33} \end{bmatrix}$$

Using four vertices, which are matrix elements in the same column, to define a Bezier cubic curve, the matrix representation may be obtained as follows.

$$S_0(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i0}$$

where $B_{3,i}(u)$ is the Bernstein function. The other three curves defined in this way are $S_1(u)$, $S_2(u)$, and $S_3(u)$.

$$S_1(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i1}$$

$$S_2(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i2}$$

$$S_3(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i3}$$

Choosing any value of the parameter u between zero and one, i.e., $u = u_1$, we can regard $S_0(u_1)$, $S_1(u_1)$, $S_2(u_1)$ and $S_3(u_1)$ as four vertices of the characteristic polygon and construct a Bezier cubic curve with parameter v.

$$Q(v) = \sum_{j=0}^{3} B_{3,j}(v)\ S_j(u_1) \qquad (4.6)$$

Where $0 \le v \le 1$. When the parameter u_1 varies between 0 and 1, the curve $Q(v)$ forms a bi-cubic surface and can thus be derived from equation 4.6[38].

$$r(u,v) = \begin{bmatrix} S_0(u) & S_1(u) & S_2(u) & S_3(u) \end{bmatrix} \begin{bmatrix} B_{3,0}(v) \\ B_{3,1}(v) \\ B_{3,2}(v) \\ B_{3,3}(v) \end{bmatrix}$$

or

$$r(u,v) = \begin{bmatrix} B_{3,0}(u) & B_{3,1}(u) & B_{3,2}(u) & B_{3,3}(u) \end{bmatrix} V \begin{bmatrix} B_{3,0}(v) \\ B_{3,1}(v) \\ B_{3,2}(v) \\ B_{3,3}(v) \end{bmatrix}$$

or in simplified form:

$$\mathbf{r}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \ \mathbf{V} \ M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{4.7}$$

where M^T is the transpose of M.

This principle can be generalized to a surface of $(n+1) \times (m+1)$ order. The positional vectors $\mathbf{V}_{ij}$ $(i = 0, 1, 2, ..., n;\ j = 0, 1, 2, ..., m)$ of the vertices of the characteristic polyhedron form the following vertex information matrix:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{00} & \mathbf{V}_{01} & \cdots & \mathbf{V}_{0m} \\ \mathbf{V}_{10} & \mathbf{V}_{11} & \cdots & \mathbf{V}_{1m} \\ \cdot & \cdot & \cdots & \cdot \\ \mathbf{V}_{n0} & \mathbf{V}_{n1} & \cdots & \mathbf{V}_{nm} \end{bmatrix}$$

and the surface is defined as

$$\mathbf{r}(u,v) = \begin{bmatrix} B_{n,0}(u) & B_{n,1}(u) & \cdots & B_{n,n}(u) \end{bmatrix} \mathbf{V} \begin{bmatrix} B_{m,0}(v) \\ B_{m,1}(v) \\ \cdot \\ \cdot \\ \cdot \\ B_{m,m}(v) \end{bmatrix} \tag{4.8}$$

or

$$\mathbf{r}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{n,i}(u) \ B_{m,j}(v) \ \mathbf{V}_{ij} \tag{4.9}$$

where u, v are two parameters and $0 \le u \le 1$, $0 \le v \le 1$.

The command used in microCADDS to generate Bezier surfaces is "INSert SPOle", followed by modifiers. SPOLEs are names of surfaces given by Computervision.

4.3.3 Ruled surfaces

A ruled surface is a three-dimensional surface entity defined by two curves that can be thought of as consisting of a number of straight lines, each going through a separate point on one curve to a separate or corresponding point on the other curve. The command used in micro-CADDS to generate a ruled surface is "INSert RSURF", followed by modifiers. A ruled surface takes the form of[24]

$$\mathbf{r}(u,v) = \mathbf{r}_0(u) + v \left[\mathbf{r}_1(u) - \mathbf{r}_0(u) \right] \tag{4.10}$$

or

$$\mathbf{r}(u,v) = (1-v)\, \mathbf{r}_0(u) + v\, \mathbf{r}_1(u) \tag{4.11}$$

Where u, v are parameters and $0 \le u \le 1$, $0 \le v \le 1$, $\mathbf{r}_0(u)$ and $\mathbf{r}_1(u)$ are two Bezier curves of degree n and m respectively.

$$\mathbf{r}_0(u) = \sum_{i=0}^{n} B_{n,i}(u)\, \mathbf{V}_{i0}$$

$$\mathbf{r}_1(u) = \sum_{i=0}^{m} B_{m,i}(u)\, \mathbf{V}_{i1}$$

Figure 4.3 shows a typical ruled surface generated from two cubic Bezier curves.

4.3.4 Surfaces of revolution

A surface of revolution may be described by the rotation of a plane curve about the axis of symmetry. If we take this axis to be the Z-axis, and let the surface intersect the OXZ plane in the plane curve

$$\mathbf{r} = \mathbf{r}_c(u) = p(u)\, \vec{i} + z(u)\, \vec{k}$$

then Figure 4.4 shows that the surface of revolution has the equation[24]

$$\mathbf{r} = p(u) \cos\theta\, \vec{i} + p(u) \sin\theta\, \vec{j} + z(u)\, \vec{k} \tag{4.12}$$

The command used in microCADDS to generate a surface of revolution is "INSert SREV", followed by modifiers.

4.3.5 Coons surface patches

Coons patches are used in microCADDS to generate surfaces from four boundary curves such as those shown in Figure 4.5. The definition of an Coons patch is given in matrix form as[24]:

$$\mathbf{r}(u,v) = \begin{bmatrix} (1-u) & u \end{bmatrix} \begin{bmatrix} \mathbf{r}(0,v) \\ \mathbf{r}(1,v) \end{bmatrix} + \begin{bmatrix} \mathbf{r}(u,0) & \mathbf{r}(u,1) \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

$$- \begin{bmatrix} (1-u) & u \end{bmatrix} \begin{bmatrix} \mathbf{r}(0,0) & \mathbf{r}(0,1) \\ \mathbf{r}(1,0) & \mathbf{r}(1,1) \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix} \tag{4.13}$$

Where $\mathbf{r}(u,0)$, $\mathbf{r}(u,1)$, $\mathbf{r}(0,v)$, and $\mathbf{r}(1,v)$ are the boundary curves of the surface patch and u, v are two parameters, $0 \le u \le 1, 0 \le v \le 1$.

The command used in microCADDS to create an Coons patch is "SMOoth SPOle", followed by modifiers.

It should be noted that when an Coons patch is created in microCADDS, it is transmitted into Bezier form automatically by the software and the user then uses the properties of Bezier surfaces to manipulate it.

4.4 Digitizing a telephone handset on the CNC machining centre

A telephone handset was selected as an example since it is a typical plastics product widely used and frequently changed in shape and size, and hence its design and manufacture is the suitable subject for the use of PC based CAD/CAM systems.

The Renishaw Probe system on the Takisawa Machining centre at UMIST was used for digitizing. The diameter of the probe is 6 mm and it should be noted that the data obtained is the coordinates of the probe centre. To obtain the size of the measured part, the surface model constructed directly from the digitized data should be offset using the method described in chapter 6. The offset amount is equal to the radius of the probe used and the offset direction is normal to the surface. The set up and the coordinate system is shown in Figure 4.6.

Before the measurement is carried out, the amount of data to be collected and the digitizing procedure should be planned carefully. Usually one axis is fixed (in this case the Y axis), and the other two axes (in this case the X, Z axes) are moved simultaneously for each measuring operation (in this case for each cross section). The Y values were pre-determined manually by experience depending upon the complexity of the surface shape along the Y direction of the surface under consideration[65]. (See Table 4.1). The Y values determine the positions of the cross sections in the Y direction, and the differences between two adjacent Y values determine the gaps between two adjacent cross sections.

For each cross section (corresponding to each Y value), the number of points to be digitized (or the X values) were determined manually by experience depending upon the complexity of the surface under consideration. For instance, For a simple surface shown in Figure 4.7 (a), 3×3 points (totally 9 points) may be sufficient. However, for a more complex surface shown in Figure 4.7 (b), 5×5 points (totally 25 points) are required.

If the number of points to be digitized and the positions of these points on the surface are not properly planned, first the shape of the handset may not be sufficiently described, secondly it may be difficult to build and refine the surface model, since the sub-patches of the model may be badly arranged.

Once the above has been done, the handset was measured as follows:

i) Set the pre-determined Y value of the probe manually.

ii) Set the pre-determined X value manually, move the probe along the Z axis until the probe touches the measured surface and write down the Z value. Repeat this for the same cross section.

iii) Repeat the above two steps (i and ii) for each Y value, the coordinates of the points at each cross section were obtained.

The results is shown in Table 4.1. It should be noted that the points at the edge of the measured surface and some additional points in special areas should be taken, since these points are very important. The measurement was done manually, since the probe has not been programmed for this purpose.

4.5 Generating cross sectional curves from the digitized data

The digitized data obtained was then input manually into microCADDS, since the interface used for transferring digitized data from the CNC machining centre to the CAD/CAM system has not been developed. The input points are shown in Figure 4.8.

To generate cross sectional curves using the points obtained, two aspects must be taken into account, namely, *interpolation* and curve fitting.

There are different mathematical expressions (called interpolants) used to interpolate a set of data, such as polynomial, rational, spline[66] and parametric curves (Ferguson, Bezier and Coons, etc.)[24]. Each of them has its own advantages and disadvantages. Generally, high degree interpolants may exhibit an oscillation problem. Therefore, in practice, composite curves are constructed by fitting successive low-degree curves to successive groups of data points. The positional and gradient continuity of the resulting piecewise functions at the joint points is the major task of curve fitting.

In microCADDS, Bezier's method is used to approximate data. Since a Bezier curve does not pass through all the given points except the two end points, a special computational algorithm is used, i.e., the data is interpolated by computing the control polygon to make the curve pass through the given points. This is the inverse algorithm of Bezier curves[60].

For $(n+1)$ points provided, i.e. $\mathbf{r}_i$, $(i = 0, 1, 2, ..., n)$, a Bezier curve of degree n is created by two steps:

i) Using the given points $\mathbf{r}_i$ to calculate the vertices $\mathbf{V}_i$. This can be achieved by resolving the following equation group:

$$\mathbf{r}_0 = \mathbf{V}_0$$

$$\mathbf{r}_j = \sum_{i=0}^{n} B_{n,i}(u_j)\ \mathbf{V}_i \tag{4.14}$$

$$\mathbf{r}_n = \mathbf{V}_n$$

The above equation group is derived directly from equation 4.1 and where j= 1, 2, ..., n-1. The parameter u_j which is corresponding to $\mathbf{r}_j$ is determined by[38]

$$u_j = \begin{cases} 0 & j = 0 \\ \sum_{k=1}^{j} l_k \ / \sum_{k=1}^{n} l_k & j == 1, 2, ..., n-1 \\ 1 & j = n \end{cases}$$

Where l_k is the chord length and

$$l_k = |\ \mathbf{r}_k - \mathbf{r}_{k-1}\ |$$

ii) Using the obtained vertices $\mathbf{V}_i$ to construct an Bezier curve of degree n.

$$\mathbf{r} = \sum_{i=0}^{n} B_{n,i}(u)\ \mathbf{V}_i$$

The command used in microCADDS is "SMOth CPOle" followed by modifiers. It should be noted that the degree of the curve could be less than n when n+1 points are given, since it is desired to use a low degree curve to approximate as many points as possible within the required tolerance. In the case of the telephone handset, two Bezier curves of degree 3 or 4 were created for each cross section.

The two curves for one cross section form a composite curve and the positional and gradient continuity at the joint point can be achieved when the following conditions are satisfied (see Figure 4.9).

a) $V_3^{(1)} = V_0^{(2)}$, where $V_3^{(1)}$ is the third vertex of the first curve, and $V_0^{(2)}$ is the first vertex of the second curve.

b) $V_2^{(1)}$, $V_3^{(1)} = V_0^{(2)}$ and $V_1^{(2)}$ are collinear. This can be easily derived by resolving the following equations:

$$\begin{aligned} \mathbf{r}^{(2)}(0) &= \mathbf{r}^{(1)}(1) \\ \dot{\mathbf{r}}^{(2)}(0) &= k_1\, \dot{\mathbf{r}}^{(1)}(1) \\ \ddot{\mathbf{r}}^{(2)}(0) &= k_2\, \ddot{\mathbf{r}}^{(1)}(1) \end{aligned} \tag{4.16}$$

Where k_1 and k_2 are constants and $\dot{\mathbf{r}}^{(1)}$, $\dot{\mathbf{r}}^{(2)}$ are differentiations of $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ with respect to u; $\ddot{\mathbf{r}}^{(1)}$ and $\ddot{\mathbf{r}}^{(2)}$ are differentiations of $\dot{\mathbf{r}}^{(1)}$ and $\dot{\mathbf{r}}^{(2)}$ with respect to u.

The command used in microCADDS to achieve positional and gradient continuity of two curves is "MATch CURve" followed by modifiers. It can also be achieved manually. The generated and refined curves are shown in Figure 4.10.

4.6 Generating surfaces from cross sectional curves

There is only one approach to generate surface patches in microCADDS, this is to use Coons patches which were discussed in section 4.2.5. It uses four boundary curves to create a "flattest" surface patch. In practice, this is far from satisfactory. First, the interior of an Coons patch is determined by the mathematical representation, instead of by the designer who may need extra points or lines within the patch to describe the shape. Secondly, the sizes of the patches must be small enough to reach the required accuracy, this results in a large number of patches used to describe a given component, and more design time; Thirdly, one surface patch should cover as much area as possible within the required tolerance, otherwise further manipulations of the surfaces and NC data will be difficult. The Coons patches are unsuitable for this purpose.

Bezier surfaces cannot be used directly to interpolate points or curves, since the given points which define the surface are control vertices and do not lie on the surface except those at the four corners of the patch. However, using the inverse algorithm of Bezier surfaces which is similar to the inverse algorithm of Bezier curves, a fairly large surface patch can be created from a curve mesh. The comparison of Coons patches and Beziers patches for interpolation is shown in Figure 4.11. The principle and procedure of this method is discussed below.

In section 4.2.2 the formation of an bi-cubic Bezier surface was discussed (see Figure 4.2). The inverse procedure is:

i) Four curves $Q_0(v)$, $Q_1(v)$, $Q_2(v)$, and $Q_3(v)$ which lie on the expected surface are given, or points used to generate the four curves are given. (See Figure 4.12 a). Since

$$Q_0(v) = \sum_{j=0}^{3} B_{3,j}(v)\ S_j(u_0)$$

$$Q_1(v) = \sum_{j=0}^{3} B_{3,j}(v)\ S_j(u_1)$$

$$Q_2(v) = \sum_{j=0}^{3} B_{3,j}(v)\ S_j(u_2)$$

$$Q_3(v) = \sum_{j=0}^{3} B_{3,j}(v)\ S_j(u_3)$$

it is clear that the vertices of the control polygons of $Q_0(v)$, $Q_1(v)$, $Q_2(v)$ *and* $Q_3(v)$ *are the points lying on the curves* $S_0(u)$, $S_1(u)$, $S_2(u)$ *and* $S_3(u)$.

ii) Using the vertices of the control polygons of the four given curves to construct four curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$ (see Figure 4.12 b). Where

$$S_0(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i0}$$

$$S_1(u) = \sum_{i=0}^{3} B_{3,i}(u)\ V_{i1}$$

$$S_2(u) = \sum_{i=0}^{3} B_{3,i}(u)\, V_{i2}$$

$$S_3(u) = \sum_{i=0}^{3} B_{3,i}(u)\, V_{i3}$$

It is clear that the vertices of the control polygons of the curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$ are the vertices of the control polyhedron of the expected Bezier surface patch. Where

$$V = \begin{bmatrix} V_{00} & V_{01} & V_{02} & V_{03} \\ V_{10} & V_{11} & V_{12} & V_{13} \\ V_{20} & V_{21} & V_{22} & V_{23} \\ V_{30} & V_{31} & V_{32} & V_{33} \end{bmatrix}$$

iii) Using the vertices of the control polyhedron to generate an Bezier surface patch $r(u,v)$. (See Figure 4.13).

$$r(u,v) = \begin{bmatrix} B_{3,0}(u) & B_{3,1}(u) & B_{3,2}(u) & B_{3,3}(u) \end{bmatrix} V \begin{bmatrix} B_{3,0}(v) \\ B_{3,1}(v) \\ B_{3,2}(v) \\ B_{3,3}(v) \end{bmatrix}$$

The surface generated will, of course, pass through the four given curves $Q_0(v)$, $Q_1(v)$, $Q_2(v)$ and $Q_3(v)$ and this principle can be generalized for surfaces of $(n + 1) \times (m + 1)$ order.

The surfaces created from the cross sectional curves of the telephone handset using the inverse algorithm of Bezier surfaces are shown in Figure 4.14

4.7 The principle of smoothing adjacent Bezier surfaces

The smoothing of adjacent surfaces is very important in CAD systems. Unfortunately, there is no such facility in microCADDS. (Nearly all the current CAD packages run on Personal Computers do not supply this facility)

For the packages running on mini or mainframe computers, there are several methods of smoothing surfaces such as Clark[67], Barsky[68] and Boehm[69], these methods smooth the surface everywhere, which would be inconvenient for representing objects containing scalloped edges such as an umbrella, or a boat's hull, etc. A system like Unisurf developed at the Regie Renault in France smoothes parametric surface patches by varying degrees[70], but the number of control points increases with the freedom necessary to meet the mathematical constraints. Beeker[71] developed a method which solved the above restriction, however, his method requires a change of data base and is not suitable for Personal Computers.

The method used in this project has the following advantages:

i) Bezier surfaces can be smoothed locally without changing the whole surface patch. So that complex surfaces can be smoothed.

ii) The degrees of the surface is not changed and the calculation is simple and quick.

iii) Data base is not changed. This will save the computer memory and the time for the re-design of the data base.

iv) It is interactive and suitable for PC based systems.

The major drawback is that this method is only semi-automatic. The program will calculate the new control polyhedra of the surfaces by changing the related control vertices. The user, however, needs to insert the new surface patches using the modified control polyhedra.

The mathematics used is as follow:

For two adjacent cubic Bezier patches depicted in Figure 4.15,

$$\mathbf{r}^{(1)}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \; \mathbf{V}^{(1)} \; M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

and

$$\mathbf{r}^{(2)}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \ \mathbf{V}^{(2)} \ M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

Positional continuity across the boundary will result if[38]

$$\mathbf{r}^{(1)}(1,v) = \mathbf{r}^{(2)}(0,v) \tag{4.17}$$

for all v such that $0 \leq v \leq 1$. That is:

$$\mathbf{V}_{3i}^{(1)} = \mathbf{V}_{0i}^{(2)} \ , \quad i = 0,\ 1,\ 2,\ 3 \tag{4.18}$$

These imply, reasonably enough, that a common boundary curve between the two patches requires a common boundary polygon between the two characteristic polyhedra (see Figure 4.16).

For gradient continuity across the boundary, the tangent plane of patch 1 on u=1 must coincide with that of patch 2 on u=0, for all v such that $0 \leq v \leq 1$. Then the direction of the surface normal will be continuous across the boundary, and the condition

$$\mathbf{r}_u^{(2)}(0,v) \times \mathbf{r}_v^{(2)}(0,v) = \lambda(v)\ \mathbf{r}_u^{(1)}(1,v) \times \mathbf{r}_v^{(1)}(1,v) \tag{4.19}$$

must apply. Where $\mathbf{r}_u^{(2)}(0,v)$, $\mathbf{r}_v^{(2)}(0,v)$, $\mathbf{r}_u^{(1)}(1,v)$ and $\mathbf{r}_v^{(1)}(1,v)$ are cross-boundary slopes and $\lambda(v)$ is a positive valued scalar function, which provides a "freedom" for the difference in the magnitude of the surface normal vector.

Since

$$\mathbf{r}_v^{(2)} = \mathbf{r}_v^{(1)}(1,v)$$

then equation 4.19 yields

$$\mathbf{r}_u^{(2)}(0,v) = \lambda(v)\ \mathbf{r}_u^{(1)}(1,v) \tag{4.20}$$

This implies that all curves with parameter u only in the composite surface will have continuity of gradient direction. Setting $\lambda(v) = \lambda$, a positive constant, and the four equations can be obtained,

$$(\mathbf{V}_{1i}^{(2)} - \mathbf{V}_{0i}^{(2)}) = \lambda (\mathbf{V}_{3i}^{(1)} - \mathbf{V}_{2i}^{(1)}) , \quad i = 0, 1, 2, 3 \tag{4.21}$$

This shows that the four pairs of polyhedron edges which meet at the boundary, must be collinear (see Figure 4.16). And the cross-boundary tangent magnitude ratio λ must be constant along the common boundary.

In practice, the restriction imposed by the constancy of tangent ratio are severe. Consider, for example, the construction of the smooth composite surface shown in Figure 4.17, starting with patch A and continuing with patches B, C and D, in that order. The sixteen vertices of the control polyhedron of patch A can be chosen freely. Then the tangent ratio across the boundary between patches A and B must be specified. Once this has been done, no less than 8 vertices of the control polyhedron of B are fixed by the conditions for positional and gradient continuity, leaving only 8 free to be chosen. Similarly, once the tangent ratio between patches A and C is specified, only 8 vertices left to choose in defining patch C. With patch D, the tangent ratios between B and D and between C and D are already fixed, and it is found that only four of the sixteen vertices which define this patch are free to be chosen. The gradient continuity between B and D and between C and D near the corner at which the four surfaces meet can never be achieved.

The collinearity of the pairs of polyhedron edges which meet at the boundary is also too restrictive in the cases such as that shown in Figure 4.18.

To obtain more freedom in constructing composite surfaces, equation 4.19 can be satisfied by[24]

$$\mathbf{r}_u^{(2)}(0,v) = \lambda(v)\, \mathbf{r}_u^{(1)}(1,v) + \mu(v)\, \mathbf{r}_v^{(1)}(1,v) \tag{4.22}$$

in which $\mu(v)$ is another scalar function of v. This simply requires $\mathbf{r}_u^{(2)}(0,v)$ to lie in the same plane as $\mathbf{r}_u^{(1)}(1,v)$ and $\mathbf{r}_v^{(1)}(1,v)$, that is in the tangent plane of patch 1 at the boundary point concerned. Much more scope is now available.

It is important to realize that when equation 4.22 is used as the smoothness condition, cross boundary gradient vectors are no longer continuous in the direction across patch boundaries. The collinearity condition on polyhedron edges meeting at a boundary may therefore be discarded. Using this more general criterion for gradient continuity, a smooth continuous surface can be built up. The composite patch boundaries, in both the u and v direction, will have positional continuity but gradient discontinuity at all patch corners. It can be inferred from equation 4.22, applied at a mesh intersection, that the tangent directions of all four patch boundaries meeting at an intersection must be coplanar.

The gradient continuity condition equation 4.22, can be expressed in terms of characteristic polyhedra. For two cubic Bezier surfaces (see Figure 4.18), The following equations must apply:

$$\mathbf{V}_{10}^{(2)} - \mathbf{V}_{00}^{(2)} = \lambda \, (\mathbf{V}_{30}^{(1)} - \mathbf{V}_{20}^{(1)}) + \mu_0 \, (\mathbf{V}_{31}^{(1)} - \mathbf{V}_{30}^{(1)}$$

$$\mathbf{V}_{11}^{(2)} - \mathbf{V}_{01}^{(2)} = \lambda \, (\mathbf{V}_{31}^{(1)} - \mathbf{V}_{21}^{(1)}) + \frac{1}{3} \mu_0 \, (2 \, \mathbf{V}_{32}^{(1)} - \mathbf{V}_{31}^{(1)} - \mathbf{V}_{30}^{(1)}) + \frac{1}{3} \mu_1 \, (\mathbf{V}_{31}^{(1)} - \mathbf{V}_{31}^{(1)})$$

$$\mathbf{V}_{12}^{(2)} - \mathbf{V}_{02}^{(2)} = \lambda \, (\mathbf{V}_{32}^{(1)} - \mathbf{V}_{22}^{(1)}) + \frac{1}{3} \mu_0 \, (\mathbf{V}_{33}^{(1)} + \mathbf{V}_{32}^{(1)} - 2 \, \mathbf{V}_{31}^{(1)}) + \frac{1}{3} \mu_1 \, (\mathbf{V}_{32}^{(1)} - \mathbf{V}_{31}^{(1)})$$

$$\mathbf{V}_{13}^{(2)} - \mathbf{V}_{03}^{(2)} = \lambda \, (\mathbf{V}_{33}^{(1)} - \mathbf{V}_{23}^{(1)}) + (\mu_0 + \mu_1) \, (\mathbf{V}_{33}^{(1)} - \mathbf{V}_{32}^{(1)})$$

The first equation shows that the three vectors $(\mathbf{V}_{10}^{(2)} - \mathbf{V}_{00}^{(2)})$, $(\mathbf{V}_{30}^{(1)} - \mathbf{V}_{20}^{(1)})$ and $(\mathbf{V}_{31}^{(1)} - \mathbf{V}_{30}^{(1)})$ must be coplanar. Similarly, the fourth equation shows that the three vectors $\mathbf{r}_u^{(2)}(0,1)$, $\mathbf{r}_u^{(1)}(1,1)$ and $\mathbf{r}_v^{(1)}(1,1)$ must be coplanar. (See Figure 4.18).

4.8 The procedure of smoothing

The two criteria for positional and gradient continuity described in the last section have their own advantages and disadvantages. The major advantage of the first criterion is that the smoothing is easy and not limited by the degrees of the surface patches. The major disadvantage is that the design freedom is restricted. On the other hand, the second criterion has more freedom in design, however, the mathematical representation of the gradient continuity is very complicated and becomes more complicated with the increase of the degrees of the surface patches.

The method used in this project is based on the first criterion (equation 4.21), while taking advantage of the second criterion (equation 4.22) for the corners at which several patches meet. The procedure for smoothing a series of surface patches such as the telephone handset shown in Figure 4.14 is as follows:

i) Select one principal direction. In this direction, equation 4.21 is satisfied, that is, the smoothness of the surfaces in the principal direction is a priority. In the case of the telephone handset, the longitudinal direction was selected as the principal direction.

ii) Run the developed program "SMOSPL.UPL" which is based on equation 4.21 to recalculate the related vertices of the polyhedra of the joint surface patches in the principle direction. This will result in the smoothness of the surfaces in this direction.

iii) Run the same program to re-calculate the related vertices of the polyhedra of the joint surface patches in the other directions (in the case of the telephone handset, the cross sectional directions). The vertices which affect the smoothness in the principle direction should not be changed. The result of this is that in the directions except the principle direction, surfaces are smooth everywhere except those near to the corners.

iv) For the corners, the second criterion, i.e., equation 4.22 is used, since this will give enough freedom to achieve the smoothness in these areas. The program "SMOSPL.UPL" also allows the user to do this.

Once the above four steps have been finished, the surfaces are visually verified by shading. If the shaded picture of the surfaces is not satisfactory, the surfaces are refined locally by changing the vertices manually which may affect the defective areas using the command "MOVe POLe" in microCADDS.

The smoothed and shaded shape of the telephone handset is shown in Picture 1, Appendix three.

4.9 The program used for smoothing - "SMOSPL.UPL"

The program "SMOSPL.UPL" is used for smoothing surfaces by changing the related vertices of the polyhedra of the surface patches. Equation 4.21 is used to achieve colinearity and equation 4.22 is used to achieve coplanarity of the vertices.

For three vertices $A(x_a, y_a, z_a)$, $M(x_m, y_m, z_m)$ and $B(x_b, y_b, z_b)$ shown in Figure 4.19, the tangent ratio λ is determined by

$$\lambda = \frac{\overline{AM}}{\overline{MB}} \tag{4.23}$$

Where $\overline{AM}$ and $\overline{MB}$ are the lengths of lines AM and MB.

The value of λ can be given by the user or calculated using equation 4.23 for three given vertices. The vertex which is changed to achieve colinearity may be any of the three vertices under consideration. The default point is the intermediate one, e.g. M. Let $M'(x', y', z')$ be the new position of M, since

$$\lambda = \frac{x' - x_a}{x_b - x'} = \frac{y' - y_a}{y_b - y'} = \frac{z' - z_a}{z_b - z'}$$

It can be obtained:

$$x' = \frac{\lambda\, x_b + x_a}{1 + \lambda} \tag{4.24}$$

$$y' = \frac{\lambda\, y_b + y_a}{1 + \lambda}$$

$$z' = \frac{\lambda\, z_b + z_a}{1 + \lambda}$$

The coplanarity of four vertices E, F, G, H is achieved as follows (see Figure 4.20):

Assume that E, F, G are fixed and H' is the new position of H, such that E, F, G and H' are coplanar (H' is also the projection of H in the plane EFG).The vectors $\vec{EF}$, $\vec{EG}$, $\vec{EH}$, $\vec{EH'}$ and $\vec{HH'}$ are denoted by $\vec{a}$, $\vec{b}$, $\vec{c}$, $\vec{d}$ and $\vec{h}$ respectively.

Since the unit vector ($\vec{N}$) of the normal to the plane EFG is given by[72]

$$\vec{N} = \frac{\vec{a} \times \vec{b}}{|\vec{a} \times \vec{b}|} \tag{4.25}$$

and

$$\vec{N} = \pm \frac{\vec{h}}{|\vec{h}|},$$

The vector $\vec{h}$ can be obtained

$$\vec{h} = \pm \frac{|\vec{h}|}{|\vec{a} \times \vec{b}|} \vec{a} \times \vec{b} \tag{4.26}$$

In the above equation, only $|\vec{h}|$ is unknown. It can be derived by first calculating the angle ϕ between $\vec{d}$ and $(\vec{a} \times \vec{b})$:

$$\vec{d} \cdot (\vec{a} \times \vec{b}) = |\vec{d}| \, |\vec{a} \times \vec{b}| \cos \phi \tag{4.27}$$

Where $0 \le \phi \le \pi$. For the triangle EHH',

$$|\vec{h}| = |\vec{d}| \, |\cos \phi| \tag{4.28}$$

Combining equation 4.26, 4.27 and 4.28, the vector $\vec{h}$ can be expressed as:

$$\vec{h} = \pm \frac{|\vec{d} \cdot (\vec{a} \times \vec{b})|}{|\vec{a} \times \vec{b}|^2} \vec{a} \times \vec{b} \tag{4.29}$$

Equation 4.29 produces two different results. The expected point should meet the condition for coplanar vectors[72]:

$$\vec{a} \cdot (\vec{b} \times \vec{c}) = 0 \tag{4.30}$$

In the program, the condition should be checked before the calculation, if it is met, i.e., the four vertices E, F, G and H are already coplanar, and further calculation is stopped. Since the real value is not exactly zero, the above equation is written as

$$| \vec{a} \cdot (\vec{b} \times \vec{c}) | \le \varepsilon \tag{4.30a}$$

Where ε is a small positive value and $0 < \varepsilon \ll 1.0$.

The flowchart of the program is shown in Figure 4.21.

4.10 Contributions and comments

In this chapter, a method of generating surface models from digitized data has been introduced. This method is particularly suitable for use on the PC based CAD/CAM system. The suggested use of this method is in relative small companies such as in the plastics mould industry which produces products with complex surfaces and which can afford to purchase such systems.

The major improvements made in this method are

i) A improved approach to the interpolation of digitized data with Bezier representation. This method is able to represent the interior of a surface patch and generate a fairly large low order surface patch to approximate a fairly large amount of data. The method is also easy to use.

ii) A program for the smoothing of adjacent surface patches. This program can smooth the surface patches locally without changing the whole shape of the original surfaces. The degrees of the surfaces are not necessary to be changed and the data base of the system is not required to be rebuilt.

Both of the above approaches are integrated with microCADDS and are interactive to the user. The only drawback is that these methods are not fully automatic. The user is required to prepare, or modify the characteristic polyhedra for the expected surface patches. Sometimes, the control polyhedra of Bezier surfaces may be very complicated, or distorted if the digitized data is irregularly distributed.

Chapter Five

Generating Surface Models from 2D Drawings and Changing the Shape and Volume of Surface Models

5.1 Summary

In chapter four, the approach to the construction of surface models using digitized data from a master part was described. In this chapter, the method of constructing surface models from two dimensional (2D) orthographic engineering drawings will be detailed. The principle and program used for changing surface models in shape and size and the calculation of the volume of components embodying free form surfaces are also introduced. Finally, the advantages and disadvantages of the procedures and programs are described.

5.2 Using 2D orthographic engineering drawings to construct surface models

Engineering drawings are currently the most important technological documentation scheme for design, manufacture and storage. They are also a kind of international language for technical exchanges between engineering and technical personnel in different countries and so constructing surface models from 2D orthographic engineering drawings is very important in CAD/CAM applications.

5.2.1 Introduction to the problem

Efforts have been made in automating the procedure of constructing 3D computer models from 2D drawings. However, those methods are not sufficiently general and are limited to polyhedral parts[73] [74]. Further investigation aimed at extending shape-modelling possibilities was made by Aldefeld[75] and a new approach which has the potential for dealing with a variety of non-polyhedral shapes was presented. This method, however, requires the 2D representation of an object to be complete in three views and the input data to include the projections of all its edges and boundaries. In practice, most engineering drawings omit certain geometrical details that the user will later supply on the basis of known conventions. To solve this problem and to further generalize the method, Ho Bin[76] presented a program which can deal not only with drawings with three normal complete views, but also with those given by two views, or with incomplete views, or with conventional parts of revolution, or with parts drawn by incomplete projections, or with cross sections. The types of the primitives provided are cuboid, pyramids, frustums of pyramids and cones. General revolutional volumes and swept objects can be easily added to the program.

All the above methods are suitable for constructing solid models which can be easily classified into different types of primitives such as cylinder, cone, sphere and pyramid, etc. The mathematical representations of those primitives are well known. However, the construction of surface models is much more difficult for the following reasons:

i) The drawings can not easily be classified into different separate surface patches. Since the mechanical engineer who made the drawings does not know the shape, size and number of the sub- surface patches which will be used to construct the surface model on the computer, when he designed the component.

ii) To build a surface model, however, it is necessary to divide the whole surface of a component into sub- surface patches. The user must decide the number of sub-patches to be used, but the definitions of the surface patches are not complete in the drawings. Only outlines of the surfaces and their edges are defined by lines and arcs.

iii) The shape of the interior of the surface patches is not described, or specified by some cross sections in the drawing.

iv) The mathematical representations of the surfaces patches are unknown. For a solid model, if the type of a primitive is given, that is, the mathematical representation is known, the user can get some information from the 2D drawings and the solid model will be created directly by the computer. The construction of a surface model is more difficult because of the unknown mathematical formula of the surface patches.

Richards[77] tried to automate this approach and started with the Bezier representation. This method, however, requires the construction of the control polyhedron of the surface patch from the drawing views and the complete description of the edges of the surface patches. In practice, components may be complex and the definitions of surfaces in the engineering drawings may be incomplete. This method therefore has only limited application.

In this section, a method of constructing surface models from 2D drawings will be described. The method can deal with any component composed of complex surfaces, especially those frequently changed in size and shape as in the plastics industry. The problems described above will be solved and the method is particularly suitable for PC based systems.

A typical engineering drawing of a plastics bottle provided by McBride (UK) was used to construct a surface model. The procedure is detailed below.

5.2.2 Input 2D information into the computer

Basically, an engineering drawing consists of three PRINCIPAL views, i.e. the Top, Front and Right view and some additional views such as section views. Local areas which are small in size may be scaled. The geometry of the drawings is mainly composed of lines and arcs. Figure 5.1 shows the basic elements of the drawing supplied by McBride (UK). The datum and the X, Y and Z axes were added and the dimensions were omitted.

To construct a surface model of the bottle, the first step is to input the 2D drawing information into microCADDS. The geometry in the Top view lies in the XOY plane, the geometry in the Front view lies in the XOZ plane and the geometry in the Right view lies in the YOZ plane. The geometry of the three views must be constructed using the same datum and stored in different layers. Layers are used to separate geometric elements in microCADDS.

For the section views C-C and D-D in Figure 5.1, two extra views should be defined in microCADDS, since they are not the standard views supplied by the software. The lines $\overline{CC}$ and $\overline{DD}$ are used as datum lines and the geometry in these two views is positionally and orientationally correct in the space OXYZ. Different layers should be used to store the elements in section views C-C and D-D.

5.2.3 Interpret 2D information into 3D geometry

Once the elements in the 2D drawing have been input into microCADDS, the next step is to transform each element (line, arc, or spline) into 3D geometry. We start with the elements in the Top view, which are the projections of the required 3D elements in the Top view. then repeat for the Front view or the Right view and move the elements in the Top view until they overlie on their counterparts in the Front view or the Right view. This is shown by a single curve in Figure 5.2.

i) For a single element (1) in the Top view (see Figure 5.2a), sufficient points are generated from the curve using the command "GENerate POInt" in microCADDS. These points are used later for the generation of the required curve.

ii) Then activate the Front view. In the Front view, element 1 becomes a horizontal line lying on the X axis. Element 1 and its counterpart - element 2 in the Front view are shown in Figure 5.2b.

iii) In the Front view, move the points generated from element 1 until they overlie on element 2 by changing their Z values only. The X and Y values of these points must be kept unchanged. (see Figure 5.2c). The points are now lying on the required 3D curve.

iv) Finally, using the moved points to create a curve - element 3 in the *Isometric* view (see Figure 5.2d). This curve is the expected 3D curve.

Alternatively, the 3D curve can be obtained by using the Top view and the Right view, since a general space element can be sufficiently described by two orthographic views, that is, any two of the three principle views. However, for elements which lie in the planes parallel to the XOZ plane, the Front view must be used; for elements which lie in the planes parallel to the YOZ plane, the Right view must be used. That is why three basic views are necessary.

The three dimensional elements of the bottle transformed in this way from the two dimensional drawing are shown in Figure 5.3.

5.2.4 Construct the wireframe model

Before the surface model is built, a wireframe model should be constructed. It is not straightforward to construct a wireframe model from the interpreted 3D elements obtained in the last section for the following reasons:

a) The number of elements is not enough to build a wireframe model. The wireframe model of a surface model consists of all the boundary curves of the sub-patches of the surface model. As discussed in section 5.2.1, the mechanical engineer who designed the component in terms of 2D drawings does not know the sub-patches which are going to be used to build the surface model. Therefore, he does not draw the boundary curves on his drawings.

b) The shape of the bottle is complex, therefore the sizes and shapes of the sub-patches used to form the model must be carefully selected. There are no fixed rules for this. The size and arrangement of the sub-patches must be done by experience.

c) The interior of the surface is not sufficiently described in the orthographic drawing, therefore additional cross sectional curves must be designed.

The first step is to determine the sizes, shape and positions of the sub-patches to be used. For the bottle considered, surface patches are radial from the handle hole. Incorrect selection of the sizes, shapes or positions of the surface patches will cause distortion of the bottle faces and make it difficult to further manipulate and refine the surfaces. It will also affect the generation of NC machining data from the surfaces.

The second step is to create extra elements such as boundary curves, to give four or three boundary curves for each surface patch selected. Extra cross sectional curves may be necessary to describe the shape of the bottle completely. All the extra elements designed by the CAD user should be verified using two or three views of the original 2D information. The interpreted 3D elements and the additional elements form the wireframe model shown in Figure 5.4.

5.2.5 To construct the surface model from the wireframe model

Once the wireframe model has been built, surface patches are then created from the wireframe model using the command "SMOothe SPOle". This produces Coons Patches which were discussed in section 4.6. Once this has been done, The surface patches are smoothed using the developed program "SMOSPL.UPL" which was discussed in sections 4.7, 4.8 and 4.9. The shape of the interior of the surface patches can be modified and verified by the three views of the original 2D information input from the drawing. Finally, the surface model is visually verified by a shaded picture. The satisfactory surface model is shown in Figure 5.5 and the shaded picture is shown in Picture 2, Appendix three.

5.3 Changing shape and volume of components embodying free form surfaces

In today's market, domestic products such as plastics bottles, bodies of television sets, computers and cars, etc. vary widely in shape and size. The shape and size of a product also changes frequently to meet different customers requirements. Taking a bottle which contains liquid product as an example, the requirements may be

i) Keep the basic shape of the bottle for recognition, increase the volume by, say, 15% to give customers an extra amount free of charge as a sales promotion.

ii) Keep the basic shape, produce different size bottles, such as, half of a litre, one litre, two litres, etc. to meet different needs.

iii) Keep the basic shape, make the bottle longer or fatter for reasons of shelf space etc.

iv) Change the shape of local areas, e.g. the handle, the bottom or the shoulder of the bottle.

It is very difficult to make the above changes without CAD. The changes in shape will result in the re-design of the bottle in terms of 2D drawings. The changes in volume will cause more difficulties. Any accurate estimation of the volume is impossible in 2D engineering drawings, since the shape of the bottle in the drawings is not completely defined.

With Computer Aided Design, the above tasks can be done easily. Personal Computer based CAD/CAM systems are good choices for this purpose because of their low cost and good increasing capabilities. However, the current commercial packages running on micro-computers have not sufficient facilities to deal with this matter.

In this section, a method of changing the shape and volume of components easily and rapidly will be described. Another bottle model which was transformed from the Computervision CADDS 4X system will be taken as a case study. The model is shown in Figure 5.6 and Picture 3. More examples will be given later in this chapter.

5.3.1 The principle of changing shape of free form surfaces

The bottle shown in Figure 5.6 consists of sub-patches such as Bezier surfaces, Ruled surfaces and surfaces of revolution. Changes in the X, Y or Z direction can be achieved by scaling the sub-patches in the X, Y or Z direction. The scalers for the X, Y and Z axes are denoted by S_x, S_y and S_z respectively. The overall scaler is denoted by S and

$$S = S_x \; S_y \; S_z \tag{5.1}$$

For the bi-cubic Bezier surface given by equation 4.7

$$\mathbf{r}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \;\; \mathbf{V} \;\; M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

since S_x, S_y and S_z are constants, it can be obtained

$$S_x \; \mathbf{r}_x(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \;\; (S_x \; \mathbf{V}_x) \;\; M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{5.2}$$

and

$$S_x \; \mathbf{V}_x = \begin{bmatrix} S_x\mathbf{V}_{x00} & S_x\mathbf{V}_{x01} & S_x\mathbf{V}_{x02} & S_x\mathbf{V}_{x03} \\ S_x\mathbf{V}_{x10} & S_x\mathbf{V}_{x11} & S_x\mathbf{V}_{x12} & S_x\mathbf{V}_{x13} \\ S_x\mathbf{V}_{x20} & S_x\mathbf{V}_{x21} & S_x\mathbf{V}_{x22} & S_x\mathbf{V}_{x23} \\ S_x\mathbf{V}_{x30} & S_x\mathbf{V}_{x31} & S_x\mathbf{V}_{x32} & S_x\mathbf{V}_{x33} \end{bmatrix}$$

similarly

$$S_y \; \mathbf{r}_y(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \;\; (S_y \; \mathbf{V}_y) \;\; M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{5.3}$$

$$S_z\ \mathbf{r}_z(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M\ (S_z\ \mathbf{V}_z)\ M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \tag{5.4}$$

and

$$S_y\ \mathbf{V}_y = \begin{bmatrix} S_y \mathbf{V}_{y00} & S_y \mathbf{V}_{y01} & S_y \mathbf{V}_{y02} & S_y \mathbf{V}_{y03} \\ S_y \mathbf{V}_{y10} & S_y \mathbf{V}_{y11} & S_y \mathbf{V}_{y12} & S_y \mathbf{V}_{y13} \\ S_y \mathbf{V}_{y20} & S_y \mathbf{V}_{y21} & S_y \mathbf{V}_{y22} & S_y \mathbf{V}_{y23} \\ S_y \mathbf{V}_{y30} & S_y \mathbf{V}_{y31} & S_y \mathbf{V}_{y32} & S_y \mathbf{V}_{y33} \end{bmatrix}$$

$$S_z\ \mathbf{V}_z = \begin{bmatrix} S_z \mathbf{V}_{z00} & S_z \mathbf{V}_{z01} & S_z \mathbf{V}_{z02} & S_z \mathbf{V}_{z03} \\ S_z \mathbf{V}_{z10} & S_z \mathbf{V}_{z11} & S_z \mathbf{V}_{z12} & S_z \mathbf{V}_{z13} \\ S_z \mathbf{V}_{z20} & S_z \mathbf{V}_{z21} & S_z \mathbf{V}_{z22} & S_z \mathbf{V}_{z23} \\ S_z \mathbf{V}_{z30} & S_z \mathbf{V}_{z31} & S_z \mathbf{V}_{z32} & S_z \mathbf{V}_{z33} \end{bmatrix}$$

The above equations show that scaling a Bezier surface patch in the X, Y and Z directions can be achieved by scaling each vertex of the control polyhedron of the surface patch in the X, Y and Z directions by the scalers S_x, S_y and S_z, respectively. This principle applies to general Bezier surfaces of order $(n+1) \times (m+1)$.

For the ruled surface given by equation 4.11

$$\mathbf{r}(u,v) = (1-v)\ \mathbf{r}_0(u) + v\ \mathbf{r}_1(u)$$

The scaled surface in the X direction is given by

$$S_x \mathbf{r}_x(u,v) = (1-v)\ S_x\ \mathbf{r}_{0x}(u) + v\ S_x\ \mathbf{r}_{1x}(u) \tag{5.5}$$

Assume that $\mathbf{r}_0(u)$ and $\mathbf{r}_1(u)$ are two cubic Bezier curves, thus

$$S_x \mathbf{r}_{0x}(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \begin{bmatrix} S_x \mathbf{V}^0_{x0} \\ S_x \mathbf{V}^0_{x1} \\ S_x \mathbf{V}^0_{x2} \\ S_x \mathbf{V}^0_{x3} \end{bmatrix} \tag{5.6}$$

$$S_x \mathbf{r}_{1x}(u) = \left[\begin{array}{cccc} 1 & u & u^2 & u^3 \end{array} \right] M \left[\begin{array}{c} S_x \mathbf{V}_{x0}^1 \\ S_x \mathbf{V}_{x1}^1 \\ S_x \mathbf{V}_{x2}^1 \\ S_x \mathbf{V}_{x3}^1 \end{array} \right] \tag{5.7}$$

Where $\mathbf{V}_{xj}^0$ $(j = 0, 1, 2, 3)$ are the X values of the vertices of the curve $\mathbf{r}_0(u)$ and $\mathbf{V}_{xj}^1$ $(j = 0, 1, 2, 3)$ are the X values of the vertices of the curve $\mathbf{r}_1(u)$.

Equations 5.5, 5.6 and 5.7 show that scaling a ruled surface in the X direction can be achieved by scaling each vertex of the control polyhedron of the surface in the X direction by the same scaler S_x. Similarly, it can be deduced that scaling a ruled surface in the Y and X directions can be achieved by scaling each vertex of the control polyhedron of the surface in the Y and Z directions by the same scalers S_y and S_z respectively.

Since surfaces of revolution have been transformed into Bezier representations in micro-CADDS, the principle of scaling Bezier surfaces also applies.

5.3.2 The scaling program and its capabilities

Generally, for a closed space in the Cartesian Coordinate system, the volume enclosed can be expressed as

$$V = \iiint_{\omega} dv = \iiint_{\omega} dx\ dy\ dz \tag{5.8}$$

Where ω is the whole surface which encloses the space. It could be composed of several sub-patches ω_i $(i = 1, 2, ..., n)$, therefore,

$$\omega = \sum_{i=1}^{n} \omega_i$$

If the space, or the enclosing surface is scaled by the scalers S_x, S_y and S_z, that is

$$x' = S_x\ x$$

$$y' = S_y \; y$$
$$z' = S_z \; z$$

Where x', y' and z' are the scaled variables of x, y and z. The volume of the scaled surface (Ω) becomes

$$V' = \iiint_{\Omega} dx' \; dy' \; dz' = \iiint_{\omega} S_x \; S_y \; S_z \; dx \; dydz \tag{5.9}$$

or

$$V' = S_x \; S_y \; S_z \iiint_{\omega} dx \; dy \; dz = S \; V \tag{5.9a}$$

Equation 5.9 implies that the overall scaler S is the volume scaler. A program "SCLSPL.UPL" has been developed based on the above principles and it offers the following choices:

a) Change the volume of a component such as the bottle by an amount, say, 15 %, keeping the shape unchanged. By definition, the volume scaler is

$$S = 1 + 15\% = 1.15$$

The surface can be scaled in the X, Y and Z axes by the same value which is

$$S_x = S_y = S_z = S^{\frac{1}{3}} = 1.0477$$

b) Change the volume by scaling the surface in the X, Y and Z axes using different scalers. Assume that the volume scaler is still 1.15, then the designer can select two scalers freely, e.g. $S_x = 1.0$, $S_y = 1.05$ and the third scaler can determined by

$$S_z = \frac{S}{S_x \; S_y} = 1.0952$$

c) Change the volume by scaling special parts of the surface. Assume that the volume of the component is composed of several parts V_i $(i = 1, 2, ..., n)$, thus

$$V = \sum_{i=1}^{n} V_i$$

The designer can change any part of the component by scaling the volume V_i. It should be noted that the total volume and the volume of the part to be changed must be known. The calculation of the component with free form surfaces will be discussed in the next section.

5.4 Calculating the volume of components with free form surfaces

The calculation of the volume of a component may be required, especially if they are used as containers. Only the volume of 3D models on the CAD system can be accurately calculated except those composed of simple elements such as cylinders, spheres and pyramids, etc. Most CAD systems have this facility, however, PC based software usually does not. The reason is that the accurate calculation of the volume of the component with complex surfaces is not a simple matter. The method used in this project is not fully automatic by comparison with those used in large systems, however, a result accurate and quick enough for most engineering purposes can be obtained. In this section, the general principle will be first described, then the equations for special cases will be discussed, and finally, the program used to calculate the volume of a component will be introduced.

5.4.1 The general principle

For a surface patch **r** = **r**(u,v), the appropriate surface ELEMENT is formed by the two triangles ABC and BCD. (See Figure 5.7). (An element is a small area used for integral). For large δu and δv, these triangles are not coplanar. However, for a small element, these triangles can be taken as coplanar and the vector area of the element ABCD is approximately

$$ABCD = \left[\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right] \delta u \ \delta v \tag{5.10}$$

Since the volume of a pyramid is equal to one third of the base area times the perpendicular height, we see that the volume subtended by ABCD at the origin O is

$$OABCD = \frac{1}{3}\, \mathbf{r} \cdot \left[\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right] \delta u \; \delta v \tag{5.11}$$

so that the total volume V subtended by a region R of the u, v plane is approximately

$$V = \frac{1}{3} \iint_R \mathbf{r} \cdot \left[\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right] du \; dv \tag{5.12}$$

This equation is very difficult to work out, since the integral

$$V = \frac{1}{3} \iint_R \begin{vmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z \\ \frac{\partial \mathbf{r}_x}{\partial u} & \frac{\partial \mathbf{r}_y}{\partial u} & \frac{\partial \mathbf{r}_z}{\partial u} \\ \frac{\partial \mathbf{r}_x}{\partial v} & \frac{\partial \mathbf{r}_y}{\partial v} & \frac{\partial \mathbf{r}_z}{\partial v} \end{vmatrix} du \; dv$$

requires a large amount of mathematical calculation. Alternatively, we use the idea of integral and divide the surface patch into small sub areas and calculate the approximate volume of each sub area, the sum will be the approximate volume subtended by the surface patch. The maximum error of the volume of each sub area will be calculated and the sum forms the maximum possible error. If the error is too big, the surface can be divided into more sub areas and re-calculated until the error is acceptable.

For the element shown in Figure 5.7, its area is the sum of the two small triangles ABC and BCD, thus[78]

$$\begin{aligned} A_{ij} &= \sqrt{p_1 (p_1 - \overline{AB})(p_1 - \overline{BC})(p_1 - \overline{AC})} \\ &+ \sqrt{p_2 (p_2 - \overline{BD})(p_2 - \overline{DC})(p_2 - \overline{BC})} \end{aligned} \tag{5.13}$$

Where

$$p_1 = \frac{1}{2} (\overline{AB} + \overline{BC} + \overline{AC})$$

$$p_2 = \frac{1}{2} (\overline{BD} + \overline{DC} + \overline{BC})$$

and A_{ij} is the area of the element corresponding to $\mathbf{r}_{ij}$. For the pyramid OABCD, we assume that the height is $|\mathbf{r}(u,v)|$. Then the volume of the pyramid is

$$V_{ij} = \frac{1}{3} \; |\mathbf{r}| \; A_{ij} \tag{5.14}$$

The total volume subtended by the surface patch is given by

$$V = \sum_{i=1}^{k} \sum_{j=1}^{l} V_{ij} \tag{5.15}$$

Where $k \times l$ is the number of elements divided from the surface patch. The resulting error will be discussed in section 5.4.3

5.4.2 Special cases

The above method applies to general cases. In practice, there are many components which possess a NATURAL AXIS, about which they generally have some degree of symmetry. These products are usually designed by defining a number of cross sectional curves. Those cross sectional curves are perpendicular to this axis. A typical example is the bottle shown in Figure 5.6. For this particular case, the calculation of volume becomes easier. Figure 5.8 shows a small element ABCD of a surface patch and its axis OO'.

We assume that the element volume is a regular polyhedron such that the volume of this element is approximately

$$V_{ij} = \frac{1}{2} \; |\mathbf{r}| \; A_{ij} \tag{5.16}$$

Where $|\mathbf{r}|$ is the line $\overline{FA}$ and A_{ij} is the area of ABCD.

The sum of the volume of the elements is the volume of the surface patch.

$$V = \sum_{i=1}^{k} \sum_{j=1}^{l} V_{ij}$$

In practice, there are also many components which are designed symmetrically, or to some degree of symmetry, about a plane. A typical example of this type is the bottle with the handle shown in Figure 5.5.

Figure 5.9 shows an element taken from this type of component where the plane of symmetry is the XOY plane. When this element is small enough, it can be assumed to be a cuboid. Therefore, the volume of this element is approximately

$$V_{ij} = | \mathbf{r}_z | A_{ij} \tag{5.17}$$

Where $\mathbf{r}_z$ is the Z value of the vector $\mathbf{r}$ and A_{ij} is the sum of the areas of the two triangles $A'B'C'$ and $B'C'D'$. Thus

$$A_{ij} = \sqrt{p_1 (p_1 - \overline{A'B'}) (p_1 - \overline{B'C'}) (p_1 - \overline{A'C'})} + \sqrt{p_2 (p_2 - \overline{B'D'}) (p_2 - \overline{D'C'}) (p_2 - \overline{B'C'})} \tag{5.18}$$

Where

$$p_1 = \frac{1}{2} (\overline{A'B'} + \overline{B'C'} + \overline{A'C'}) \tag{5.19}$$

$$p_2 = \frac{1}{2} (\overline{B'D'} + \overline{D'C'} + \overline{B'C'}) \tag{5.20}$$

The sum of the volume of the elements is the volume of the surface patch.

$$V = \sum_{i=1}^{k} \sum_{j=1}^{l} V_{ij}$$

5.4.3 The program used for volume calculation

In practice, the volume of products which are used as containers or covers are usually required to be calculated in the design stage. In this section, a program "VOLUME.UPL" used for this purpose will be discussed. The program is based on the principles described previously, so that it can deal with different types of surface models.

Before running the program, the surface model of a component should be analysed. That is, the surface model must be classified into the following cases:

Case 1: The whole surface model has no axis of symmetry, however, the volume can be easily divided into elements such as that shown in Figure 5.7. In this case, the equations listed in section 5.4.1 can be used to calculate the sum of these volume elements.

Case 2: The whole model is designed about an axis, so that the volume can be divided into the volume elements such as that shown in Figure 5.8. A typical example is the bottle shown in Figure 5.6

Case 3: The surface model has a reference plane, about which it has some degree of symmetry. Therefore, the volume can be divided into volume elements such as shown in Figure 5.9. A typical example is the bottle shown in Figure 5.5.

Sometimes a surface model may be very complex. It can not be simply classified into any one of the cases listed above. However, it can always be divided into different parts, each belonging to one of the three basic cases. Then the program will calculate these parts separately.

Once the surface has been classified, the program "VOLUME.UPL" can be run several times to calculate each part of the surface model. A number of surface patches of the same model can be calculated at once. The control points of each surface patch must be digitized when running the program.

The number of elements for each surface patch is decided by the relative error, which is defined by

$$\delta = \frac{\Delta V}{V} \times 100\ \% \qquad (5.21)$$

Where δ is the relative error, ΔV is the error in volume and V is the volume. The program first calculates the relative error of one boundary curve, then changes the number of elements until the overall error is acceptable. This restricts the overall error to an acceptable range.

5.5 Examples showing the features of the developed programs.

The two programs VOLUME.UPL and SCLSPL.UPL have been tested by three examples. The bottle model shown in Figure 5.6 was used for this purpose. The first example shows the features of VOLUME.UPL and the other two examples show the features of SCLSPL.UPL. The results are detailed in Table 5.1.

Example 1: To calculate the volume of the surface model shown in Figure 5.6 by running VOLUME.UPL. It took about 4 hours to calculate the volume of the bottle. The total volume is 1.124 pints (0.639 litres) and the average relative error is 0.011 %.

Example 2: Scaling the bottle in the X and Z direction to increase the volume by 50 %. Therefore the volume scaler $S = 1.15$. Let $S_y = 1.0$ and $S_x = S_z$, thus

$$S_x = S_z = \sqrt{S} = 1.225$$

The running time is about one hour. The scaled bottle is shown in Figure 5.10 and Picture 4.

Example 3: Scaling the bottle locally to decrease the volume by 22.8 %. The middle part of the bottle was scaled in the Y direction only. Since the volume of the part to be scaled is 0.5128 pint, the volume scaler of the part is given by

$$S_l = \frac{0.5128 - 1.124 \times 22.8\%}{0.5128} = 0.5$$

Since $S_{lx} = S_{lz} = 1.0$, thus $S_{ly} = S_l = 0.5$. Where S_l, S_{lx}, S_{ly} and S_{lz} are the local scalers. The running time is only 20 minutes. The locally scaled bottle is shown in Figure 5.11 and Picture 5.

5.6 Contributions and comments

The main work described in this chapter is the construction of surface models from two dimensional orthographic drawings and the change of any surface model in shape and size. The calculation of the volume of components embodying free form surfaces is also described. The following results have been achieved:

i) A method has been described for interpreting 2D engineering drawings into 3D geometric elements and constructing surface models from the interpreted elements has been tested on a typical plastics bottle supplied by McBride (UK). The method can deal components of any complexity including those incompletely defined in the 2D drawings.

ii) A program "SCLSPL.UPL" has been described for re-constructing existing surface models. The surface models can be scaled in the X, Y and Z directions respectively. The surface models can also be modified locally in shape and volume.

iii) A program "VOLUME.UPL" has been described for calculating the volume of components composed of free form surfaces. This program is suitable not only for general surface models but also for those with symmetric axes or planes.

The above methods are particularly suitable for *PC* based *CAD* systems. The expert who is working on more sophisticated systems such as Computervision CADDS 4X systems will find that these methods require too much "manual" work. However, results have shown that the methods are quick and accurate enough for PC users.

Chapter Six

Preparing the Toolpath Geometry for NC Machining

6.1 Summary

In the previous chapters, the construction of surface models has been discussed, in this chapter, the method of preparing toolpath geometry for two and half-axis NC machining will be detailed. Some important concepts in machining free form surfaces will be first introduced, then the principle of the method is described. The determination of the sizes of ball-ended cutters, the generation of offset surfaces to the part surfaces, the selection of machining directions, the determination of toolpath intervals and the calculation of the maximum cusp height and deviation are also discussed in detail. Two examples are given to test the developed programs, and finally, the advantages and disadvantages of the procedures and the programs are described. The NC part programming and NC machining of the two models will be discussed in the next chapter.

6.2 Introduction

Since 5-axis and 3-axis machining packages are unavailable at present in the PC environment, the major aim of this work was to find a method of machining free form surfaces using PC-based CAD and 2-1/2-axis CAM software. Before describing the method, some important concepts are introduced as follows.

6.2.1 Multi-axis machining

There are currently various NC machining packages run on mainframe systems and super mini-computer based systems. The machining axes are from 2-1/2- to 5-axes, such as the Computervision 2-1/2-axis milling, drilling and machining centre programming facilities of CVNC-M2, the 3-axis surface machining capabilities of CVNC-M3 and the 5-axis NC programming software of CVNC-M5. However, multi-axis machining is essential for the manufacture of free form surfaces, and 3-axis machining has been available and used satisfactorily for this purpose for many years, although now there is an increasing move towards 5-axis machining in order to reduce manufacturing times. Table 6.1 compares the 3-axis and 5-axis machining of a commercial vehicle cab roof model and demonstrates the advantages of each method. The model and toolpaths are shown in Figure 6.1 and Figure 6.2. From this it may be seen that 5-axis machining is very efficient for the manufacture of large convex surfaces. Indeed, high-quality finishes are possible requiring very little hand finish since the flat-ended milling cutter is kept normal to the surface as highlighted in Figure 6.3. The volume of data is also very much reduced because of the large stepover which is possible, thereby speeding up the processing time. However, intricate and concave surfaces generally require 3-axis machining with a ball-ended cutter and consequently, a mixture of the two is usual for achieving optimum performance within companies with 5-axis capability.

6.2.2 Preliminary considerations

Having received surface data suitable for machining , the following topics need consideration before programming begins.

With this method, ball-ended tools must be used for 3-axis and 2-1/2-axis work, but there are shortcomings related to the cutting conditions found when the tool is normal to the surface as shown in Figure 6.4. Therefore speeds and feeds need to be considered with due cognizance of the effective radius of the cutter. The tools to be used must be decided on at an early stage. This should not only enable optimum material removal but also must be made with due reference to the application of the CAM routines which will be used and the minimum concave radius of the surface which determines the maximum tool radius allowable. It is essential at the earliest possible stage to establish a practical geometric relationship between the part orientation and the component setup on the machine tool. This should take account of component geometry , machine tool axis configuration, working envelope, datum locations and clamping arrangements. A considerable amount of machining time can be saved by the application of maximum material removal routines, normally with different cutters and toolpaths from those used for finish machining. Roughing is not easily automated and the creation of approximate roughing surfaces has proved the most efficient way to produce the rough toolpaths.

6.2.3 Some important aspects in machining sculptured surfaces

In machining sculptured surfaces the following aspects should be addressed, namely[79] [80],

i) Toolpath planning.

ii) Stepovers along the toolpaths determined by the chordal deviation tolerance.

iii) Path intervals (i.e. the distance between adjacent paths) determined by scallop height tolerance.

iv) Gouge avoidance (i.e. anti-interference machining).

Toolpath planning means determining methods of guiding (or driving) a cutting tool to generate the desired surface from a solid block or a preformed raw stock. The commonly used tool path planning methods are

a) To introduce explicitly tool guiding surfaces (driving surfaces).

b) To plan toolpaths on the XY-plane of a Cartesian coordinate system.

c) To plan toolpaths on the parametric space.

The first approach is used in APT III and is called the APT-based toolpath generation method. In APT, the surface to be generated is called the PART SURFACE, and the cutter passes are defined by means of a series of intermediate drive surfaces. The basic idea is that the cutter is moved in one direction while maintaining contacts with both the part surface and the drive surface. At each stepping point iterative numerical searches are made to locate the cutter position within a specified tolerance limit. The main disadvantage of this method is that the iterative computation is time consuming and unstable. This toolpath planning method is not suitable for use in machining complex surfaces because there is no guarantee of convergence for irregularly curved sculptured surfaces[79].

The second approach is sometimes called the Cartesian machining method, and the third approach is called the parametric machining method. The parametric toolpath planning method can only be used for parametric surfaces of the form $\mathbf{r} = \mathbf{r}(u,v)$. In the Cartesian toolpath planning method, it is a common practice to plan the toolpaths to be parallel straight lines on the XY-plane. By projecting the straight lines back to the surface, actual toolpaths on the part surface are obtained. It is equivalent to finding intersection curves between the part surface and the vertical planes.

Since the CAM software used in this project is 2-1/2-axis machining and the cutter paths must be plane curves, the Cartesian toolpath planning method was used, although the surface definitions in microCADDS are parametric.

There are two possible choices in the Cartesian machining approach: the part surface intersected by vertical planes and an offset surface of the part surface intersected by vertical planes. The former is called the Cutter Contact (CC)-Cartesian method and the latter is called the Cutter Location (CL)-Cartesian method. In this project, the CL-Cartesian method was used. The reason for this will be discussed in Section 6.3.

The determination of stepovers and path intervals and the gouge problem will be discussed later in this chapter.

6.3 The principle of the method

Since the tool used must be ball-ended for 3-axis and 2-1/2-axis machining, to reach a point on the surface, the centre point of the cutter radius should be on the normal to the surface at the point, and at a distance equal to the tool radius. In other words, the centre point of the cutter should be on an offset surface of the surface to be cut as shown in Figure 6.5.

Surfaces are usually defined parametrically in CAD systems as this facilitates easy design manipulation. A number of offset points are calculated for each small change in the parameters which define the surface for the generation of NC data in most CAD/CAM systems. This is a considerable computing task because the amount of change in the parameters must be small enough to reduce the chordal deviation and the scallop cusp height between two adjacent cuts.

For 2-1/2-axis machining packages, the offset points of one cut must be on a plane normal to one axis (the X, Y or Z). To calculate these points from a parametrically defined surface requires more computing time and also introduces some approximation due to numerical methods.

To reduce the amount of calculation so that the PC based systems can run the program, a limited number of offset points are calculated and then an offset surface is generated using these points. The offset surface is then intersected by a set of planes normal to an axis. Toolpath geometry (consisting of line segments) is then generated along the intersections. These may then be joined to form a lace pattern toolpath which is only limited in size by the number of data blocks the machine tool controller can handle.

The method can be summed up as:

i) Selecting tool size, or offset amount. The tool radius should be less than the minimum radius of curvature of the concave part of the part surface.

ii) Calculating offset points normal to the part surface. The number of points are determined manually by the complexity of the part surface. The number of points required to adequately describe a surface was discussed in section 4.4 and shown in Figure 4.7(a) and (b).

iii) Generating an offset surface from the offset points. The offset surface is continuous if the part surface is continuous. The offset surface will also remain smooth within a tolerable amount if the part surface is smooth.

iv) Cutting the offset surface with a set of parallel planes normal to any of the three axes X, Y and Z, this extends the 2-1/2-axis machining to 3-axis machining. MicroCADDS offers this facility, however, the number of surfaces and planes are limited and the computer is often crashed when multi-planes and surfaces intersect each other. Moreover, the intersections are splines and the mathematical representations of these splines are unknown. This restricts the generation of line segments for the toolpath geometry. In this project, a program has been developed to automatically create intersection curves quickly and without crashing the computer.

v) The intersection curves are used for toolpath generation, i.e., for the generation of small line segments within given deviation tolerance.

It should be noted that the machining directions must be carefully selected and the scallop cusp height resulting from the path intervals should be calculated and limited within the given tolerance.

6.4 Selecting tool size and avoiding the gouging problem

Generally, large sizes of ball-ended tools are used for efficient cutting and finish machining, since the cusp height left on the part surface decreases as the tool size increases. Figure 6.6 (a) and (b) compares a bigger tool and a smaller tool for the same path interval. However, when the tool radius exceeds the minimum radius of curvature of the concave part of a surface, a gouging problem arises (see Figure 6.6 (c)). Therefore, the radius of a ball-ended tool must be less than the minimum radius of curvature of the concave part of a surface to be machined.

It can be seen here that the gouging problem can be avoided at an early stage by using the Cutter Location Cartesian toolpath planning method discussed in section 6.2.4. Another advantage of using CL-Cartesian method is that the computing time is significantly decreased by calculating a limited number of offset points to build an offset surface, instead of calculating many offset points for each small change in the parameters which define the surface. This was discussed in the last section.

6.4.1 The principal curvatures of a surface

For a surface $\mathbf{r} = \mathbf{r}(u,v)$ and an arbitrary curve $\mathbf{r}(t)$ on the surface, the curve equation may be written as

$$\mathbf{r} = \mathbf{r}(t) = \mathbf{r}(u(t),v(t)) \tag{6.1}$$

The normal curvature of the surface is defined as the curvature of the intersection curve between the surface and the plane containing the surface normal $\mathbf{n}$ and the tangent vector. The curvature of the curve $\mathbf{r}(t)$ is defined as the normal curvature of the surface in the direction of the curve (see Figure 6.7). It is given by[38]:

$$k_n = \frac{\begin{bmatrix} \dfrac{du}{dt} & \dfrac{dv}{dt} \end{bmatrix} D \begin{bmatrix} \dfrac{du}{dt} \\ \dfrac{dv}{dt} \end{bmatrix}}{\begin{bmatrix} \dfrac{du}{dt} & \dfrac{dv}{dt} \end{bmatrix} G \begin{bmatrix} \dfrac{du}{dt} \\ \dfrac{dv}{dt} \end{bmatrix}} \tag{6.2}$$

where k_n is the normal curvature of the surface in the direction of the curve $\mathbf{r}(t)$. G is the first fundamental matrix of the surface and it is given by

$$G = \begin{bmatrix} \dfrac{\partial \mathbf{r}}{\partial u} \cdot \dfrac{\partial \mathbf{r}}{\partial u} & \dfrac{\partial \mathbf{r}}{\partial u} \cdot \dfrac{\partial \mathbf{r}}{\partial v} \\ \dfrac{\partial \mathbf{r}}{\partial u} \cdot \dfrac{\partial \mathbf{r}}{\partial v} & \dfrac{\partial \mathbf{r}}{\partial v} \cdot \dfrac{\partial \mathbf{r}}{\partial v} \end{bmatrix} \tag{6.3}$$

and D is the second fundamental matrix of the surface and it is given by

$$D = \begin{bmatrix} \mathbf{n} \cdot \dfrac{\partial^2 \mathbf{r}}{\partial u^2} & \mathbf{n} \cdot \dfrac{\partial^2 \mathbf{r}}{\partial u \partial v} \\ \mathbf{n} \cdot \dfrac{\partial^2 \mathbf{r}}{\partial u \partial v} & \mathbf{n} \cdot \dfrac{\partial^2 \mathbf{r}}{\partial v^2} \end{bmatrix} \tag{6.4}$$

where $\mathbf{n}$ is the unit vector of the normal of the surface and it is given by

$$\mathbf{n} = \pm \frac{\dfrac{\partial \mathbf{r}}{\partial u} \times \dfrac{\partial \mathbf{r}}{\partial v}}{\left| \dfrac{\partial \mathbf{r}}{\partial u} \times \dfrac{\partial \mathbf{r}}{\partial v} \right|} \tag{6.5}$$

Since the curve $\mathbf{r}(t)$ is arbitrary, the normal curvature for the same point on the surface varies as the curve which passes through this point changes. The maximum and minimum curvatures of the surface at the point are known as PRINCIPAL CURVATUREs, and we may deduce from equation 6.2 that these occur when

$$(D - k_n G) \begin{bmatrix} \dfrac{du}{dt} \\ \dfrac{dv}{dt} \end{bmatrix} = 0$$

or

$$| G | k_n^2 - (g_{11}d_{22} + d_{11}g_{22} - 2g_{12}d_{12}) k_n + | D | = 0 \tag{6.6}$$

where

$$| G | = \begin{vmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{vmatrix} = g_{11}g_{22} - g_{21}g_{12}$$

$$| D | = \begin{vmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{vmatrix} = d_{11}d_{22} - d_{21}d_{12}$$

where d_{ij} and g_{ij} are the elements of the matrix G and D respectively.

The two roots of equation 6.6 are the principle curvatures and it can be seen that the principle curvatures are determined by the surface $\mathbf{r}(u,v)$ itself.

6.4.2 The program used for the calculation of the maximum tool radius

A program RADIUS.UPL has been developed to calculate the maximum tool radius used for machining a free form surface without interference (gouging). The program allows the user to select a number of points on a surface and then the program calculates the principle curvatures at these points. The program then calculates the maximum principle curvatures of all the points and prints out the maximum tool radius allowed, which is

$$R_{\max} = \frac{1}{k_{n,max}} \tag{6.7}$$

It should be noted that the directions of the curvatures of a surface show the concave or convex properties with respect to the normal to the surface. The directions of the principle curvatures at different points, or even at the same point, may not be the same. The program will calculate these separately.

The curvatures at the edges of a surface do not exist. (Therefore, the program ignores these points on the boundaries which are selected by the user). This can be easily shown as follows.

For a general surface patch given by equation 4.9

$$\mathbf{r}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{n,i}(u)\ B_{m,j}(v)\ \mathbf{V}_{ij}$$

The partial differential of **r** with respect to u is given by

$$\frac{\partial \mathbf{r}}{\partial u} = \sum_{i=0}^{n} \sum_{j=0}^{m} \frac{d(B_{n,i}(u))}{du} B_{m,j}(v) \mathbf{V}_{ij} \tag{6.8}$$

where

$$\frac{d(B_{n,i}(u))}{du} = \frac{n!}{i!(n-i)!} (i\, u^{i-1} (1-u)^{n-i} - (n-i)\, u^{i} (1-u)^{n-i-1})$$

When the points are on the boundaries of the surface, the value of $\frac{\partial \mathbf{r}}{\partial u}$ does not exist since

$$u^{i-1} = 0^{-1} \ \textit{when}\ i = 0 \text{ and } u = 0$$

and

$$(1-u)^{n-i-1} = 0^{-1} \ \textit{when}\ i = n \text{ and } u = 1$$

Similarly, it can be shown that when the points are on the boundaries of the surface, the value of $\frac{\partial \mathbf{r}}{\partial v}$ does not exist since

$$v^{j-1} = 0^{-1} \ \textit{when}\ j = 0 \text{ and } v = 0$$

and

$$(1-v)^{m-i-1} = 0^{-1} \ \textit{when}\ j = m \text{ and } v = 1$$

The higher order of partial differential of **r** with respect to u contained in equation 6.8 is given by

$$\frac{\partial^2 \mathbf{r}}{\partial u^2} = \sum_{i=0}^{n} \sum_{j=0}^{m} \frac{d^2(B_{n,i}(u))}{du^2} B_{m,j}(v) \mathbf{V}_{ij} \tag{6.9}$$

where

$$\frac{d^2(B_{n,i}(u))}{du^2} = \frac{n!}{i!(n-i)!} (i(i-1)u^{i-2} (1-u)^{n-i}$$

$$- i(n-i)u^{i-1}(1-u)^{n-i-1} - i(n-i)u^{i-1}(1-u)^{n-i-1}$$

$$- i(i-1)(n-i)u^{i-2}(1-u)^{n-i-1} + i(n-i)(n-i-1)u^{i-1}(1-u)^{n-i-2})$$

which contains the elements of u^{i-1}, u^{i-2}, $(1-u)^{n-i-1}$ and $(1-u)^{n-i-2}$. These elements do not exist when the points are on the boundaries of the surface. This also applies to other higher order of partial differentials of **r** with respect to u and v in equation 6.6.

6.5 Generating offset surfaces to the surfaces to be machined

It has been discussed that the CL-Cartesian toolpath planning method was used in this project and therefore offset surfaces to the surfaces to be cut must be prepared after the tool sizes are selected. Since different tools may be used for different parts of a component consisting of a number of surface patches, the offset amount may be different and therefore the offset surfaces of the whole component may not be continuous.

6.5.1 Generating offset points to the part surfaces

For a Bezier surface $\mathbf{r} = \mathbf{r}(u,v)$, the unit normal at a point is given by equation 6.5,

$$\mathbf{n} = \pm \frac{\dfrac{\partial \mathbf{r}}{\partial u} \times \dfrac{\partial \mathbf{r}}{\partial v}}{\left| \dfrac{\partial \mathbf{r}}{\partial u} \times \dfrac{\partial \mathbf{r}}{\partial v} \right|} \tag{6.5}$$

The offset point at a distance equal to the tool radius from the point is given by

$$P = \mathbf{r}(u,v) \pm h\,\mathbf{n} \tag{6.10}$$

where h is the offset amount or the tool radius, P is the offset point.

$$\frac{\partial \mathbf{r}}{\partial u} = \sum_{i=0}^{n} \sum_{j=0}^{m} \frac{d(B_{n,i}(u))}{du} B_{m,j}(v)\ \mathbf{V}_{ij}$$

$$\frac{\partial \mathbf{r}}{\partial v} = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{n,i}(u) \frac{d(B_{m,j}(v))}{dv} \mathbf{V}_{ij}$$

and

$$\left[\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right] = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \left[\frac{\partial \mathbf{r}}{\partial u}\right]_x & \left[\frac{\partial \mathbf{r}}{\partial u}\right]_y & \left[\frac{\partial \mathbf{r}}{\partial u}\right]_z \\ \left[\frac{\partial \mathbf{r}}{\partial v}\right]_x & \left[\frac{\partial \mathbf{r}}{\partial v}\right]_y & \left[\frac{\partial \mathbf{r}}{\partial v}\right]_z \end{vmatrix} \tag{6.11}$$

A program BEZOFF.UPL has been written based on the above equations. It was mentioned in section 6.4.2 that the partial differentials $\frac{\partial \mathbf{r}}{\partial u}$ and $\frac{\partial \mathbf{r}}{\partial v}$ do not exist at the boundaries of the surface patches. To solve this problem, the unit normal vectors at the boundaries were replaced by those which were very near to the boundaries. This only approximates the directions of the unit normals. For instance, the unit normal $\mathbf{n}$ in equation 6.10 at $\mathbf{r}(0,v)$ can be replaced by the unit normal vector at $\mathbf{r}(0.001,v)$ and the offset point is calculated by

$$P_{0,v} = \mathbf{r}(0,v) \pm h\,\mathbf{n}_{0.001}$$

where $\mathbf{n}_{0.001}$ is the unit normal at the position near to the point $\mathbf{r}(0,v)$, and $P_{0,v}$ is the offset point at $\mathbf{r}(0,v)$.

The number of offset points required to build an offset surface was discussed in section 4.4 and shown in Figure 4.7 (a) and (b), the maximum number of offset points is 8×8.

6.5.2 Generating offset surfaces to the part surfaces

Generating offset surfaces with the calculated offset points is an inverse procedure of the formation of a Bezier surface which was discussed in section 4.2.2 and section 4.6 and shown in Figure 4.2 and Figure 4.14. One example is given below.

For 4×4 offset points obtained, a 3×3 order of Bezier surface patch can be generated (see Figure 6.8). The first step is to create four curves using the points at four rows, these curves are $Q_0(v)$, $Q_1(v)$, $Q_2(v)$ and $Q_3(v)$ in Figure 4.2 and the vertices of the control polygons of these four curves are the points lying on the curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$. Therefore, the second step is to construct another four curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$ in the column directions using the vertices of the control polygons of the curves $Q_0(v)$, $Q_1(v)$, $Q_2(v)$ and $Q_3(v)$, the vertices of the control polygons of the curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$ are the vertices of the control polyhedron of the offset surface. The final step is to construct the offset surface using the vertices of the control polygons of the curves $S_0(u)$, $S_1(u)$, $S_2(u)$ and $S_3(u)$. The generated offset surface with the part surface and offset points is shown in Figure 6.9.

6.5.3 Generating offset surfaces to a ruled surfaces

A ruled surface (see Figure 4.3) takes the form of

$$\mathbf{r}(u,v) = (1-v)\,\mathbf{r}_0(u) + v\,\mathbf{r}_1(u)$$

Where u, v are parameters and $0 \le u \le 1$, $0 \le v \le 1$, $\mathbf{r}_0(u)$ and $\mathbf{r}_1(u)$ are two Bezier curves of degree n.

$$\mathbf{r}_0(u) = \sum_{i=0}^{n} B_{n,i}(u)\,\mathbf{V}_{i0}$$

$$\mathbf{r}_1(u) = \sum_{i=0}^{m} B_{m,i}(u)\,\mathbf{V}_{i1}$$

A program has been written to offset a ruled surface based on these relationships. The procedure for generating an offset surface is the same as that for generating an offset surface of a Bezier surface except in some very simple cases where the offset surface is also a ruled surface. A ruled surface can be taken as a special case of a Bezier surface. Thus the following conclusion may be drawn (see Figure 6.10).

Using the vertices of a ruled surface to form a Bezier surface:

$$\mathbf{r}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{n,i}(u)\; B_{m,j}(v)\; \mathbf{V}_{ij} \tag{6.12}$$

Let the two curves $\mathbf{r}_0(u)$ and $\mathbf{r}_1(u)$ which form the ruled surface be of degree 3, that is $n = 3$, then

$$\mathbf{r}(u,v) = \begin{bmatrix} B_{3,0}(u) & B_{3,1}(u) & B_{3,2}(u) & B_{3,3}(u) \end{bmatrix} \mathbf{V} \begin{bmatrix} B_{1,0}(v) \\ B_{1,1}(v) \end{bmatrix} \quad (6.13)$$

Where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{00} & \mathbf{V}_{01} \\ \mathbf{V}_{10} & \mathbf{V}_{11} \\ \mathbf{V}_{20} & \mathbf{V}_{21} \\ \mathbf{V}_{30} & \mathbf{V}_{31} \end{bmatrix}$$

since

$$B_{1,0}(v) = \frac{1!}{0!\,(1-0)!}\, v^0(1-v)^{1-0} = (1-v) \quad (6.14)$$

$$B_{1,1}(v) = \frac{1!}{1!\,(1-1)!}\, v^1(1-v)^{1-1} = v \quad (6.15)$$

substituting equations (6.13), (6.14) and (6.15) in equation (6.12) and re-arranging

$$\mathbf{r}(u,v) = (1-v) \sum_{i=0}^{n} B_{n,i}(u)\ \mathbf{V}_{i0} + v \sum_{i=0}^{n} B_{n,i}(u)\ \mathbf{V}_{i1}$$

or

$$\mathbf{r}(u,v) = (1-v)\ \mathbf{r}_0(u) + v\ \mathbf{r}_1(u)$$

The above is sufficient to show that a ruled surface is a special case of a Bezier surface of order $n \times m$ when $m = 1$. Therefore, the program *BEZOFF.UPL* is also suitable for offsetting a ruled surface.

6.6 Selecting machining directions

Before creating toolpath curves (or line segments) the machining directions must be carefully selected. Since the Personal Machinist (PM) is a 2 - 1/2-axis machining package, the toolpath geometry must be in the planes parallel to the XOY, YOZ and XOZ planes. For the same component, different machining directions may be selected for different parts of the component. Machining in an incorrectly selected direction may result in severe uneven path intervals and therefore causes severe uneven cusp height on the finished surface.

Figure 6.11 compares three machining directions for the same surface patch when the gaps between the cutting planes keeps constant. It is obvious that for this surface patch, the cutting planes should be parallel to the plane YOZ shown in Figure 6.11c.

In the next section, a program used to solve the uneven cusp height problem will be introduced. However, properly selected machining directions can still significantly decrease the toolpath planning time and in some cases it becomes the vital choice. For instance, the cutting planes for a "flat" surface which is shown in Figure 6.11d can be parallel either to the XOY plane or to the YOZ plane. However, if the cutting planes were parallel to the XOZ plane, the intersection curves between the planes and the surface would be very irregular (see Figure 6.11d), and therefore, the cusp height would be very uneven.

6.7 Calculating the gap between two adjacent cuts

Once the machining directions have been selected, the next step is to calculate the gap between two adjacent cuts, i.e., the gap between two adjacent cutting planes. The gap is determined by the scallop cusp height left on the machined surface.

6.7.1 The gap between two adjacent cuts for a flat surface

For a flat surface shown in Figure 6.12, the gap between two adjacent cuts can be calculated as follows.

By Pythagoras' theorem, it can be shown that

$$(R - h)^2 = x\ (2R - x) \tag{6.16}$$

where h is the cusp height and R is the radius of the cutter. It can be re-written as

$$x^2 - 2Rx + (R - h) = 0$$

The two roots of the equation are

$$x_1 = R + \sqrt{R^2 - (R - h)^2}$$

$$x_2 = R - \sqrt{R^2 - (R - h)^2}$$

Since $x \leq R$, the second root applies, that is

$$x = x_2 = R - \sqrt{R^2 - (R - h)^2} \tag{6.17}$$

It is seen in equation 6.12 that

$$L = 2R - 2x \tag{6.18}$$

where L is the gap between two cuts. Thus,

$$L = 2\sqrt{2Rh - h^2}$$

In practice, h^2 can be ignored since the cusp height is very small, thus

$$L = 2\sqrt{2Rh} \tag{6.19}$$

6.7.2 The gap between two adjacent cuts for a general surface

Equation 6.19 applies to plane cutting. However, for the offset surface of a part surface shown in Figure 6.13, the determination of the gap between two planes normal to the Y axis which cut the surface is much more complicated. Suppose C_1 is the curve formed from the intersection of a plane and the offset surface and the plane is expressed as

$$y = y_0 \tag{6.20}$$

The surface can be represented as

$$z = z(x, y) \tag{6.21}$$

or

$$\mathbf{r} = \mathbf{r}(u, v) \tag{6.22}$$

In Figure 6.13, P is the tangent plane of the surface at point P_1 on the offset surface. **T** is the tangent vector of curve C_1 at point P_1, **N** is the normal vector of the surface at point P_1 and **CL** is a vector in the tangent plane P and thus

$$\mathbf{CL} = \mathbf{N} \times \mathbf{T} \tag{6.23}$$

$$\mathbf{N} = \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \tag{6.24}$$

$$\mathbf{T} = \frac{\partial z}{\partial x} \tag{6.25}$$

Since $y = y_0$,

$$\mathbf{T} = \frac{\partial z}{\partial x} = \begin{bmatrix} 1 & 0 & \frac{dz}{dx} \end{bmatrix} \tag{6.26}$$

To obtain $\frac{\partial z}{\partial x}$ in equation 6.26,

$$\frac{dz}{dx} = \frac{(\partial z/\partial u) \,.\, du + (\partial z/\partial v) \,.\, dv}{(\partial x/\partial u) \,.\, du + (\partial x/\partial v) \,.\, dv} \tag{6.27}$$

Since y is a constant for the curve C_1,

$$dy = \frac{\partial y}{\partial u} \, . \, du + \frac{\partial y}{\partial v} \, . \, dv = 0$$

or

$$\frac{du}{dv} = - \frac{\frac{\partial y}{\partial v}}{\frac{\partial y}{\partial u}} \tag{6.28}$$

Substitute equation 6.28 into equation 6.27,

$$\frac{dz}{dx} = \frac{(\partial z/\partial v) \, . \, (\partial y/\partial u) - (\partial z/\partial u) \, . \, (\partial y/\partial v)}{(\partial x/\partial v) \, . \, (\partial y/\partial u) - (\partial x/\partial u) \, . \, (\partial y/\partial v)} \tag{6.29}$$

For approximation, (CL/ |CL|) . L is treated as the vector with the absolute value of L in equation 6.19, that is

$$\mathbf{L} = \frac{\mathbf{CL}}{|\mathbf{CL}|} \, . \, L \tag{6.30}$$

Substitute equation 6.19 into equation 6.30,

$$\mathbf{L} = 2\sqrt{2 \, R \, h} \, \frac{\mathbf{CL}}{|\mathbf{CL}|} = [\; l_x \;\; l_y \;\; l_z \;] \tag{6.31}$$

Where l_x, l_y, l_z are the components of the vector L in the X, Y, and Z axes respectively. The gap between two cutting planes is thus

$$G = |l_y| \tag{6.32}$$

A program CUSPH.UPL has been developed based on the above equations. The user can select any number of points on the offset surface to calculate the maximum path intervals at those positions and then use equal or unequal intervals for the surface patch.

It should be noted that the surface used for cusp height calculation is the offset surface to a part surface, since it can be shown that the result obtained is the same as that obtained by calculating the part surface.

Figure 6.14 is a simplified diagram of Figure 6.13 with tool and the part surface. It can be seen that **L** = **L**′ and therefore,

$$G = |l_y| = |l'_y|$$

Where

$$\mathbf{L}' = [\, l'_x \;\; l'_y \;\; l'_z \,]$$

This means that the gap between two adjacent cuts can be obtained either by calculating the part surface or by calculating the offset surface.

6.8 Generating plane curves on the offset surface for toolpath geometry

Once the cutting directions and the path intervals have been selected, the next step is to construct plane curves on the offset surface for toolpath geometry. This could be done by intersecting the offset surface by a set of parallel planes using a command "CUT SPO" in microCADDS. However, there are several problems with this approach, namely,

i) The number of planes and surfaces to be intersected is limited. When a large number of planes and surfaces are processed, errors often appear and the computer may be crashed.

ii) The intersections are splines. The mathematical representations of the splines are unknown, and therefore, line segments are not easy to be created from the spline. Usually, Bezier curves are required to be generated manually from the splines and then line segments are created from the curves.

iii) It is difficult to automate the toolpath planning procedure.

To solve the above problem, a program "BEZTP" has been developed to calculate the intersections of the cutting planes and the offset surface. Bezier curves are generated automatically and this makes it possible to automate the whole procedure for toolpath planning.

6.8.1 The principle of intersecting surfaces by planes

A plane defined by its normal from the origin and a point on the plane takes the form[81] (see Figure 6.15).

$$(\mathbf{P} - \mathbf{N}) \,.\, \mathbf{N} = 0 \tag{6.33}$$

Where **P** is a point on the plane, **N** is the normal of the plane from the origin. For a curve $\mathbf{r} = \mathbf{r}(u)$, the intersection between the curve and the plane is given by (see Figure 6.16)

$$(\mathbf{r}(u) - \mathbf{N}) \,.\, \mathbf{N} = 0 \tag{6.34}$$

For a surface $\mathbf{r} = \mathbf{r}(u,v)$, the intersection between the surface and the plane is expressed as:

$$(\mathbf{r}(u,v) - \mathbf{N}) \,.\, \mathbf{N} = 0 \tag{6.35}$$

To obtain the intersection curve , the first step is to calculate a number of intersection points, then create a curve from the points. This is achieved by selecting a set of isoparametric curves on the surface and then calculating the intersecting points between the isoparametric curves and the plane. That is

$$(\mathbf{r}(u_i,v) - \mathbf{N}) \,.\, \mathbf{N} = 0 \tag{6.36}$$

Where $u_i = u_0, u_1, u_2, \ldots, u_n$ and $0 \le u_i \le 1$, $\mathbf{r}(u_i,v)$ are isoparametric curves on the surface $\mathbf{r}(u,v)$. If the plane is parallel to the XOZ plane (see Figure 6.17) then

$$\mathbf{N} = [\ 0 \quad k_N \quad 0\ ] \tag{6.37}$$

Where k_N is a real constant. Since

$$\mathbf{r}(u_i,v) - \mathbf{N} = [\ \mathbf{r}_x(u_i,v) \quad \mathbf{r}_y(u_i,v) - k_N \quad \mathbf{r}_z(u_i,v)\ ]$$

Thus equation 6.36 becomes

$$(r_y(u_i,v) - k_N) \,.\, k_N = 0 \tag{6.38}$$

When $k_N \neq 0$, the root of the above equation is

$$r_y(u_i,v) - k_N = 0$$

or

$$r_y(u_i,v) = k_N \tag{6.39}$$

For a bi-cubic Bezier surface given by equation 4.7

$$\mathbf{r}(u,v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \ \mathbf{V} \ M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

When the parameter u is selected and

$$\mathbf{r}_y(u_i,v) = \begin{bmatrix} 1 & u_i & u_i^2 & u_i^3 \end{bmatrix} M \ \mathbf{V}_y \ M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} = k_N \tag{6.40}$$

Where

$$\mathbf{V}_y = \begin{bmatrix} \mathbf{V}_{y00} & \mathbf{V}_{y01} & \mathbf{V}_{y02} & \mathbf{V}_{y03} \\ \mathbf{V}_{y10} & \mathbf{V}_{y11} & \mathbf{V}_{y12} & \mathbf{V}_{y13} \\ \mathbf{V}_{y20} & \mathbf{V}_{y21} & \mathbf{V}_{y22} & \mathbf{V}_{y23} \\ \mathbf{V}_{y30} & \mathbf{V}_{y31} & \mathbf{V}_{y32} & \mathbf{V}_{y33} \end{bmatrix} \tag{6.41}$$

Equation 6.41 can be classified into the polynomial form:

$$a + bv + cv^2 + dv^3 = 0 \tag{6.42}$$

Therefore, the value of v corresponding to u_i can be calculated by numerical analysis methods described in the following section. Then the point $\mathbf{r}(u_i,v_i)$ can be calculated.

Similarly, when the cutting plane is parallel to the XOY or the YOZ plane, the value of

v corresponding to u_i can be obtained by solving the equations

$$\mathbf{r}_z(u_i, v) = k_N \tag{6.43}$$

or

$$\mathbf{r}_x(u_i, v) = k_N \tag{6.44}$$

and then the intersection points can be obtained.

6.8.2 Numerical methods of solving polynomial equations

For low degree polynomial equations such as quadratic and cubic equations, the DIRECT solution method is suitable[82]. For a quadratic equation

$$x^2 + a_1 x + a_0 = 0 \tag{6.45}$$

The two roots are

$$x = \frac{1}{2}\left(-a_1 \pm \sqrt{a_1^2 - 4a_0}\right)$$

For a cubic equation

$$x^3 + a_2 x^2 + a_1 x + a_0 = 0 \tag{6.46}$$

Substitute

$$x = y - \frac{a_2}{3}$$

the equation becomes

$$y^3 + b_1 y + b_0 = 0 \tag{6.47}$$

where

$$b_1 = \frac{(3a_1 - a_2^2)}{3}$$

$$b_0 = \frac{(2a_2^3 - 9a_2a_1 + 27a_0)}{27}$$

A discriminant d^2 can be calculated by

$$d^2 = \left[\frac{b_1}{3}\right]^3 + \left[\frac{b_0}{2}\right]^2$$

If $d^2 > 0$, there will be one real root and two conjugate roots. If $d^2 = 0$, there will be three real roots and at least two equal roots. If $d^2 < 0$, there will be three unique real roots. Then the three roots of equation 6.47 are given by

$$y_1 = A + B$$

$$y_2 = -\frac{(A + B)}{2} + (A - B)\sqrt{-\frac{3}{2}}$$

$$y_3 = -\frac{(A + B)}{2} - (A - B)\sqrt{-\frac{3}{2}}$$

where

$$A = (-\frac{b_0}{2} + d)^{\frac{1}{3}}$$

$$B = (-\frac{b_0}{2} - d)^{\frac{1}{3}}$$

and therefore, the roots of equation 6.46 can be found.

Since the direct method becomes very complicated when the degrees of the equations are high, for high degrees of polynomial equations, especially for our problems, the BRACKETING method is a better choice[83]. The method is introduced as follows.

For a polynomial equation of any degree

$$y = f(x) \quad (0 \leq x \leq 1) \tag{6.48}$$

Assuming that $f(x)$ is always a concave or convex curve in the range [0, 1], then if $f(0)$ and $f(1)$ have the same sign, there is no real root. (see Figure 6.18a and b). If $f(0)$ and $f(1)$ have opposite signs, there is a unique root. (see Figure 6.18c)

In Figure 6.18c, select x_1 in [0, 1], if $f(0)$ and $f(x_1)$ have opposite signs, the range [0, 1] is replaced by $[0, x_1]$, else if $f(x_1)$ and $f(1)$ have opposite signs, the range [0, 1] is replaced by $[x_1, 1]$. Repeat this procedure to find a x_i which makes

$$|x_i - x_{i-1}| \leq \varepsilon \tag{6.49}$$

Where ε is the acceptable error. In practice, the middle point of the range is selected. e.g.,

$$x_1 = \frac{0+1}{2}.$$

$$x_2 = \frac{0+0.5}{2} \quad or \quad x_2 = \frac{0.5+1}{2}, \quad etc.$$

The error is predictable,

$$e_i = \frac{1}{2}, \left[\frac{1}{2}\right]^2, \left[\frac{1}{2}\right]^3, \ldots, \left[\frac{1}{2}\right]^n \tag{6.50}$$

Where e_i is the error in x for the ith iteration and $i = 1, 2, 3, \ldots, n$

All the assumptions made for this method are acceptable since the intersections of the planes are unique.

6.8.3 The program used for calculating intersections between planes and surfaces

A program BEZTP.UPL has been developed based on the above equations. The user first inputs the control polyhedron of the offset surface, the gap between two adjacent cuts and the tolerance. The user should also tell the computer which axis the cutting planes are parallel to. Then the program will calculate the intersection points and create a Bezier surface from the points automatically. The program has the following features:

i) The program applies to general surfaces of order $n \times m$.

ii) The error is predictable and therefore, high accuracy can be achieved.

iii) No failure during running the program.

iv) The intersection curves are automatically created and they are Bezier curves.

v) Cutting planes are not used. all the operations are performed on the offset surface itself. This is very similar to the direct toolpath generation along isoparameters of the surface performed on large *CAD/CAM* systems.

vi) Since only a limited number of offset points are calculated to form a intersection curve, running time is saved.

6.8.4 Blending toolpath curves to form complete cutting profiles

The toolpath curves created by the program BEZTP.UPL are for one patch. The component could be machined patch by patch (see Figure 6.19a). However, it is more efficient to machine a number of surface patches in a single operation. (See Figure 6.19b). The toolpath for a single cut is called a cutting PROFILE. If the path intervals of two or more adjacent surface patches are the same, the toolpath curves on those patches can be blended to form larger cutting profiles. This is achieved by blending the curves to form composite curves. The command used in microCADDS is "MATch CPO".

Since the toolpath curves are plane curves and the surfaces are arbitrary, the toolpath curves at some corners of the surface patch may not be complete. The program BEZTP.UPL does not create those incomplete curves, however, it produces offset points for those incomplete toolpath curves (see Figure 6.20a). Because the surface is continuous, the adjacent surface patch will also have incomplete toolpaths (see Figure 6.20b). Toolpath path curves can be generated manually from those points and form complete toolpath curves (see Figure 6.20c).

6.9 Calculating the line segments and the deviations

In the last section, the generation of toolpath curves and cutting profiles was described. However, these curves can not be transferred to Personal Machinist (PM) for further processing. A set of small line segments have to be created from each curve. Then the line segments are transferred to PM as toolpath geometry.

A program LINPATH.UPL has been developed to generate line segments from toolpath curves automatically. The deviations are calculated and the maximum deviation is limited to an acceptable tolerance. The mathematics is described below.

For a curve with its offset curve shown in Figure 6.21, the deviations given by

$$dev = (\rho - R)\,(1 - \cos\frac{\theta}{2}) \tag{6.51}$$

Where dev is the deviation, ρ is the radius of curvature of the offset curve and R is the radius of the ball-ended tool. The angle θ can be obtained by

$$\frac{L}{2} = \rho \cdot \sin\frac{\theta}{2} \tag{6.52}$$

Where L is the chord length of the offset curve. The radius of curvature of the offset curve is given by

$$\rho = \frac{1}{k} \tag{6.53}$$

Where k is the curvature of the offset curve. Since the toolpath curves are plane curves, the curvature can be calculated by

$$k = \frac{\left| \dfrac{d^2y}{dx^2} \right|}{\left[1 + \left[\dfrac{dy}{dx} \right]^2 \right]^{\frac{2}{3}}} \tag{6.54}$$

Since the toolpath curves are Bezier curves of the form $\mathbf{r} = \mathbf{r}(u)$,

$$\frac{dy}{dx} = \frac{\dfrac{dy}{du}}{\dfrac{dx}{du}} = \frac{\dfrac{d\mathbf{r}_y}{du}}{\dfrac{d\mathbf{r}_x}{du}} = \frac{\dot{\mathbf{r}}_y(u)}{\dot{\mathbf{r}}_x(u)} \tag{6.55}$$

$$\frac{d^2y}{dx^2} = \frac{\dfrac{d\left[\dfrac{\dot{\mathbf{r}}_y(u)}{\dot{\mathbf{r}}_x(u)} \right]}{du}}{\dot{\mathbf{r}}_x(u)}$$

or

$$\frac{d^2y}{dx^2} = \frac{1}{\dot{\mathbf{r}}_x^3(u)} \left(\ddot{\mathbf{r}}_y(u) \,.\, \dot{\mathbf{r}}_x(u) - \dot{\mathbf{r}}_y(u) \,.\, \ddot{\mathbf{r}}_x(u) \right) \tag{6.56}$$

Where

$$\dot{\mathbf{r}}_x(u) = \frac{d\mathbf{r}_x}{du}$$

$$\dot{\mathbf{r}}_y(u) = \frac{d\mathbf{r}_y}{du}$$

$$\ddot{\mathbf{r}}_x(u) = \frac{d^2\mathbf{r}_x}{du^2}$$

$$\ddot{\mathbf{r}}_y(u) = \frac{d^2\mathbf{r}_y}{du^2}$$

Substitute equations 6.55 and 6.56 into equation 6.54,

$$\rho = \frac{1}{k} = \frac{|\dot{\mathbf{r}}_x^3(u)| \cdot \left[1 + \left[\frac{\dot{\mathbf{r}}_y(u)}{\dot{\mathbf{r}}_x(u)} \right]^2 \right]^{\frac{2}{3}}}{|\ddot{\mathbf{r}}_y(u) \cdot \dot{\mathbf{r}}_x(u) - \dot{\mathbf{r}}_y(u) \cdot \ddot{\mathbf{r}}_x(u)|}$$

6.10 Examples

The method of preparing the toolpath geometry for free form surfaces has been introduced above, two examples have been done to test the method, and the programs and the procedure is described below.

6.10.1 Example 1: generating toolpath geometry for a bottle mould

The bottle model shown in Figure 5.6 was chosen as an example. To machine a mould (a female tool) of the model, the following steps were taken.

i) Select tool size. Choose several surface patches which are most curved (except the fillets at the bottom and the shoulder of the bottle) and calculate the minimum radius of curvature of the patches by running RADIUS.UPL. The minimum radius is 8.85 mm. The radii of the fillets are 4 mm. Then the actual tool diameter was selected as 10 mm. The tool radius should not be equal to or near to the minimum radius of curvature, otherwise the toolpath would become or near to a point for the concave part of the female mould. The 10 mm tool was used for the body of the bottle mould. Different tools were used to machine the bottom, shoulder and the neck of the bottle mould. This will be discussed in the next chapter.

ii) Generate the offset surface to the part surface by running BEZOFF.UPL. The offset amount is equal to the tool radius, i.e. 5 mm. The offset surface is shown in Figure 6.22.

iii) Select cutting direction. The cutting direction must be carefully selected. In this case, the cross sectional direction was selected. Picture 6 shows two male models of the bottle machined in a wrong direction and with incorrect tool sizes. It can be seen that the cusp height is irregular.

iv) Select path intervals. This was achieved by running *CUSPH.UPL*. The maximum cusp height was limited to 0.12 mm when the path intervals were 2 mm.

v) Generating toolpath curves by running *BEZTP.UPL* and create line segments by running *LINPATH.UPL*. The maximum chordal deviation was limited to 0.05 mm. The toolpath geometry is shown in Figure 6.23. The total time used to do the above work using a IBM PC AT system is 9 hours.

6.10.2 Example 2: generating toolpath geometry for a cab roof model

A cab roof model supplied by LEYLAND-DAF was scaled by 18 % and machined at UMIST. The model is shown in Figure 6.24 and Picture 7. The procedure for the generation of toolpath geometry is the same as described above. The tool size is not limited by gouging criteria, since the part to be machined is a male model. However, if the tool was too big, the offset surface will be too large and the defects and errors with the model would be increased. By experience, a 25 mm ball ended cutter was selected to machine the main part of the cab roof model and a 15 mm ball ended cutter was selected to machine the two corners of the model. The offset surfaces to the surfaces of the cab roof is shown in Figure 6.25. Three directions were selected to machine different parts of the model. For the 25 mm tool, the path intervals were 4 mm and the maximum cusp height and deviation were limited to 0.16 and 0.05 respectively; For the 15 mm tool, the path intervals were 3 mm and the maximum cusp height and deviation were limited to 0.15 mm and 0.05 mm respectively. The toolpath geometry is shown in Figure 6.26. The results of the two examples are listed in Table 6.2.

6.11 Contributions and comments

In this chapter, a method of generating the toolpath geometry for machining free form surfaces with 2 - 1/2-axis machining package has been introduced. The major features of the method and programs are:

i) The method is suitable for the machining of surfaces of any complexity. The principle applies to any 2 - 1/2-axis machining system.

ii) The method is suitable for male or female tool manufacture, particularly for mould manufacture in the plastics industry.

iii) The toolpath geometry is created automatically. The maximum scallop cusp height and chordal deviation can be controlled.

iv) The cutting direction is extended from 1 to 3, and different sizes of tools can be used to machine different parts of the component.

v) Gouging problem can be prevented by calculating the minimum radius of curvature of concave surfaces and selecting proper tools.

The only drawback of the method is that the toolpath curves near to the edges of the surface patches may need to be built manually. A common data base is needed to save the information of the model to be machined so that the user does not have to input the information each time he runs the program.

Chapter Seven

Automatic Part Programming and NC Machining

7.1 Summary

In the last chapter, the preparation of the toolpath geometry for the machining of the bottle mould and the scaled cab roof model was discussed. In this chapter, the procedure of NC machining the bottle mould and the cab roof model, including converting and transferring the toolpath geometry from microCADDS to the Personal Machinist, NC part programming, post processing and NC machining on the Takisawa Machining Centre, will be detailed.

7.2 Converting the toolpath geometry and transferring it to the Personal Machinist

Once the toolpath geometry (line segments) has been prepared in microCADDS, the next step is to convert the line segments by a User Programming Language (UPL) program called "PDPM.UCD", so that it can be accepted by the Personal Machinist Numerical Control Processor (NCP). A "MERGE" command is then used to transfer the converted geometry (.NC5) into a part program (.NC1) where the geometry may be viewed graphically and is available for subsequent processing. The merged geometry appears in the part program as the definations of a group of lines which are used as a cutting group. The starting and parking positions of the tool can be specified by digitizing two points on the screen.

In doing this, two points must be emphasized, namely

i) The line segments for each cut should be grouped using a command "GROUP ENT" in microCADDS. The definitions of these lines in the part program (.NC1) can therefore be achieved automatically. Otherwise, each line segment must be digitized manually in the Personal Machinist. This is very difficult since the line segments are very small in length.

ii) Since the merged geometry in NCP corresponds to the geometry in the Top view in microCADDS, if the prepared toolpath geometry is not parallel to the XOY plane in the Top view in microCADDS, it must be rotated to satisfy this requirement. The rotated geometry can be translated back in NCP using a command "TRANSLATE" followed by the specifications of the angles and distances about the X, Y and Z axis. Both of the two examples shown in Figure 6.23 and Figure 6.26 need to be rotated in microCADDS and translated in NCP.

7.3 NC processing and post-processing

Once the toolpath geometry has been merged in NCP as the definition part of the part program, the NC processor then asks for specifications of tool size, feed rate, speed and rapid moves. For the finish machining, the toolpath geometry is the path of the centre of the ball ended tool, therefore, the tool radius is defined as zero. (Tool size definitions are used to specify the offset amount to the left or the right in NCP).

Then the cutting subroutines are defined using the merged toolpath geometry. If the toolpath geometry is rotated in microCADDS, a "TRANSLATE" command must be included in the subroutine to translate the geometry back to the original position and orientation.

All the above is done by "answering questions" in NCP and the part programs which apply to a general machine tool are created automatically by NCP. A typical part program (BD1.NC1) used to machine the bottle mould in cross sectional direction and three typical part programs (RFNC20.NC1, RFNC50.NC1 and RFNC60.NC1) used to machine the cab roof model in three directions are given in Appendix five.

The generated part programs can be verified graphically by displaying the toolpaths on the screen, rapid moves are displayed by dashed lines. Different views can be selected to show the toolpaths. Figure 7.1 and Figure 7.2 show the toolpaths used to machine the bottle mould and the scaled cab roof model.

Having checked the graphical representation, the toolpaths or part programs must be post processed into a suitable format for a specific machine tool. This is achieved by a running a post processor "TAKMET" and the NC code (.NC2) is the ASCII format. TAKMET is a post processor used to convert the part programs into the format suitable for the UMIST Takisawa Machining Centre. The NC code (.NC2) can also be edited if necessary. The post processed part programs in Appendix five are given in Appendix six.

7.4 Rough machining

So far we have considered the toolpaths and part programs for finish machining. In practice, the block used to machine the required part should be rough machined prior to the finishing operations. The tools used for roughing are usually flat ended and the sizes of the tools should be as large as possible. The toolpaths for the roughing are less accurate than those for the finishing. Some amount of stock material must be left on the block for the finish machining.

The machining direction of the roughing operations is usually horizontal and the toolpath geometry can be achieved by cutting the part surface with a set of parallel planes. A number of lines are created near to the intersections between the cutting planes and the part surface. Those lines are then converted and transferred to NCP for part programming. One typical part program (BR1.NC1) used for the roughing of the bottle mould is given in Appendix five and the post processed part program (BR1.NC2) is given in Appendix six.

The toolpaths for the roughing of the bottle mould are shown in Figure 7.3. The size of the flat ended tool is 15 mm in diameter and the depth of each cut is 5 mm. The size of the wooden block used for the bottle mould is $300\ mm \times 115\ mm \times 45\ mm$. Since the material of the block used for the cab roof model is wax and the amount of material to be removed is not large, the roughing operation is very simple. The toolpaths for roughing consist of several straight lines, therefore, the roughing operation for the cab roof model will not be discussed in detail. The flat ended tool used for the roughing of the cab roof model is 10 mm in diameter.

7.5 The finish machining of the bottle mould and the cab roof model

Once the NC code (.NC2) was prepared, it was transferred to the controller of the NC machine via the Distribute Numerical Control (DNC) link. The toolpaths can be further verified graphically on the computer screen of the Takisawa Machining Centre prior to the machining of the parts. The Takisawa Machining Centre is shown in Picture 8. The wooden blocks which show the roughing operations and the result of the finished bottle mould are shown in Picture 9, Appendix three. The wax block of the machined cab roof model is shown in Picture 10, Appendix three.

The time used to prepare the NC programs for the bottle mould on IBM PC AT is about 24 hours and the machining time is 4 hours. The time used to prepare the NC programs for the cab roof model is 8 hours on the CV386 system and the machining time is 4 hours. The results are given in Table 7.1.

In practice, the moulds or models after finish machining must be hand finished or refined by Electro Discharge Machining (EDM). Hand finishing is relatively simple because a controlled cusp height is produced.

Chapter Eight

Conclusions and Recommendations

8.1 Conclusions

A method of modelling and NC machining components embodying free form surfaces has been developed. The method and the programs developed are suitable for Personal Computer (PC) based CAD/CAM systems. The suggested use of the method is in relatively small companies such as in the plastics mould making industry which produces products with complex surfaces such as bottle moulds and which can afford to purchase such relatively low cost systems.

Surface models of components can be specified using the digitized data from a master part. An improved approach to the interpolation of digitized data with Bezier surface patches has been developed, which can represent the interior of the surface patches.

Surface models of components can also be specified using 2 dimensional orthographic engineering drawings. The method can deal with components of any complexity including those incompletely defined in the 2D drawings.

The generated surface models can be smoothed by running the program "SMOSPL.UPL". The program can smooth the surface models locally without changing the whole shape of the original surfaces. The degrees of the surfaces do not need to be changed and the data base of the system does not require to be re-built.

The volume of components, such as of those used as containers, can be calculated by running the program "VOLUME.UPL". The volume can also be increased or decreased by scaling the surface models in any direction by running the program "SCLSPL.UPL". The surface models can also be modified locally in shape and volume.

The 2-1/2-axis machining package Personal Machinist has been extended to 3-axis machining. The toolpath geometry can be created automatically and the maximum scallop cusp height between two adjacent cuts and the maximum chordal deviation can be controlled. The cutting direction is extended from 1 to 3, and different sizes of ball ended tools can be used to machine different parts of the component. Gouging problems can be prevented by calculating the minimum radius of curvature of concave surfaces and selecting proper tools. The method is suitable for male and female tool manufacture, particularly for mould manufacture in the plastics industry.

A bottle mould and a scaled cab roof model have been machined using the UMIST Takisawa milling machine. The results are satisfactory. The machining operations were efficient, and the shape of the machined components is correct. Hand finishing is relatively simple, because a controlled cusp height is produced.

All the programs developed in the project are fully integrated with the Computervision microCADDS GCD and Surfaces and the Personal Machinist. However, the principles apply to any CAD/CAM system.

8.2 Recommendations for the use of the developed software

The developed software is recommended to be used by small die making shops which manufacture components with free form surfaces, especially plastic products such as bottles. The basic equipment required includes a CNC milling machine, a 32-bit or 16-bit PC CAD/CAM system with the Computervision Personal Designer and Personal Machinist installed. One or two fresh graduates in engineering can operate the system after 4 week training.

The software is also suitable for those who do not actually manufacture the moulds, but design the models of components for the manufacturers. The shape of the products can be designed freely and changed to meet the customers requirements. Then toolpaths and NC programs can be generated using the models stored in the computer. The designing company then sent the models and the NC codes to the manufacturer for machining directly.

8.3 Recommendations for further work

Since all the vertices of the control polyhedron of a surface patch must be digitized when running each program, a common data file is required to be built to save all the information related to the surface models, so that the data of any part of the surface models can be obtained directly.

The programs written in User Programming Language (UPL) cannot generate surfaces directly, because this function is not supplied by UPL. This limits the automation of the whole procedure for toolpath generation. The programs should be modified when more functions are developed for UPL, e.g. inserting surfaces, getting information related to curves and surfaces, etc. More detailed instructions should be displayed when running the programs.

Special CAD/CAM systems should be built for different uses, such as a system used for designing and manufacturing bottle moulds. Then the programs can be linked to form a fully automatic system.

REFERENCES

1. John, K. Krouse, *Computer-Aided Design and Computer-Aided Manufacturing*, Marcel Dekker, INC., 1982. p1-4 & p19-20

2. Bert F., Green J., *Digital computers in research*, McGraw-Hill Book Company, Inc., 1963. p4-6

3. Davis Gordon B., *Introduction to electronic computers*, McGraw-Hill Book Company, Inc., 1965. p7-13

4. Booth Taylor L., *Introduction to Computer Engineering*, John Wiley & Sons Inc., 1984. p2

5. Paker, Y., *Minicomputers - A Reference Book*, Abcaus Press, 1981. p5-17

6. Paker, Y., *Industrial Applications of Small Computers*, Conference Proceedings, Birniehill Institute, East Kilbride, Nov. 1970

7. McGlynn, Daniel, R., *Personal Computing*, John Wiley & Sons, 1982. p28-63

8. Arnold, E., *Computer-Aided Design in Europe*, 1984 Sussex European Research Centre.

9. Harris, D., *Computer Graphics and Applications*, Chapman and Hall Ltd., 1984, pvi-ix.

10. Besant C. B., *Computer Aided Design and Manufacture*, Ellis Horwood Ltd, 1980. p11.

11. Sutherland I. E., *SKETCHPAD, a man-machine graphic communication system*, 1963 Spring Joint Computer Conference, AFIPS Conf. Proc., vol. 23, Detroit, Michigan. p326-346.

12. Walker B. S., Gurd J. R. and Drawneek K. A., *Interactive Computer Graphics* , Crane, Russak & Company, Inc., 1975, p1-7

13. Johnson T. E., *SKETCHPAD III - A computer program for drawing in three dimensions*, 1963 Spring Joint Computer Conference, AFIPS Conf. Proc. vol. 23, Detroit, Michigan, p347-353.

14. Chasen S., *The introduction of man-computer graphics into the aerospace industry*, AFIPS Fall Joint Computer Conference 1965, Pt. 1, p883-892.

15. Bezier P., *Example of an Existing System in the Motor Industry: the UNISURF System*, Proc. Roy. Soc. Lond. 1971. A321, 207-218.

16. Hopgood F. R. A., *Single user workstations*, EUROGRAPHICS'83, Proceedings of the International Conference and Exhibition, Published by ELSEVIER SCIENCE PUBLISHERS B.V., 1983

17. Luggen William W., *Introduction to NC/CNC Operation*, John Willy & Sons. 1986. p1-17

18 Childs James J., *Principles of Numerical Control*, The Machinery Publishing Co. Ltd., 1965 p2-5.

19. *Air Force Computer Aided Manufacturing (AFCAM) Master Plan*, AFML-TR-74-104, July 1974.

20. *APT Part Programming*, ITT Research Institute, McGraw-Hill, 1967.

21. Taylor I. K. and Ford M. J., *The relationship between numerical control systems, programmable controllers and overall computer control in the development of a cell controller*, Proceedings of the 1st International Conference on Machine Control Systems, Edited by Prof. D. S. Ross, 23-25 October 1984, Brighton, UK. pp81-82.

22. Ogden H., *Computer Integrated Manufacture of Avionic Equipment*, Proceedings of the 1st International Conference on Machine Control Systems, Edited by Prof. D. S. Ross, 23-25 October 1984, Brighton, UK.

23. Gao X., *The Capabilities and Limitations of Modelling and Machining an Aerofoil Surface in a microcomputer Environment*, M.Sc Dissertation, 1986, UMIST, UK.

24. Faux I. D. & Pratt M. J., *Computational Geometry for Design and Manufacture*, Ellis Horwood Limited, 1979. p13-16.

25. Liming R. A., *Practical Analytical Geometry with Applications to Aircraft*, Macmillan, New York, 1944.

26. Wellman B. L., *Technical Descriptive Geometry*, McGraw-Hill, 1957.

27. Ferguson J. C., *Multivariable Curve Interpolation*, Report No. D2-22504, The Boeing Co., Seattle, Washington. 1963.

28. Forrest A. R., *Mathematical Principles for Curve and Surface Representation: In Curved Surface in Engineering*, Proc. Conference at Churchill College, Cambridge, 1972. IPC Science and Technology Press. Ltd.

29. Curry H. B. & Schoenberg I. J., *On Polya Frequency Functions IV: The Fundamental Spline Functions and Their Limits*, J. Analyse Math. 1966. p17, p71-107.

30. Coons S. A., *Surfaces for Computer Aided Design of Space Forms*, Report MAC-TR-41, Project MAC, M. T. T., 1967

31. Gordon W. J. & Riesenfeld R. F., *B-spline Curves and Surfaces, In Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, eds., Academic Press, 1974.

32. Gordon W. J. & Riesenfeld R. F., *Bernfein-Bezier Methods for the Computer Aided Design of Free Form Curves and Surfaces*, Journal ACM 21, 2, 293-310, 1974.

33 Schweikert D. G., *An Interpolation Curve Using a Spline in Tension*, J. Math. & Phys. 45, 312-317. 1966.

34. Mehlum E., *Curve and Surface Fitting based on a Variational Criterion for Smoothness*, Central

Institute of Industrial Research, Oslo. 1969.

35. Machover Carl, *What's ahead for computer graphics*, Computer Graphics'80, Proceedings of International Conference, London, 1980.

36. Machover Carl, *Has computer Graphics fulfilled its potential?* Computer Graphics'84, Proceedings of International Conference, London, 1984.

37. Guthikonda V. Rao, *Microprocessors and Microcomputer Systems*, Van Nostrand Reinhold Company, 1982. p8-10

38. Ding Q. L. & Davies B. J., *Surface Engineering Geometry for Computer Aided Design and Manufacture*, Ellis Horwood Limited, 1987. p312.

39. Scrivener S. A. R. (editor), *Computer-Aided Design and Manufacture, State of the Art Report*, Pergamon Infotech, 1985. p103-117.

40. Smith W. A. (editor), *A Guide to CAD/CAM*, The Institution of Production Engineers, 1983. p6-17

41. Foley J. D. and Dam A. V., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Company, 1984. p93-135

42. Haigh M. J., *An Introduction to Computer-Aided Design and Manufacture*, Blackwell Scientific Publications, 1985. p6-10

43. *Initial Graphics Exchange Specification Version 1.0*, National Bureau of Standards Publication NBSIR 80-1978 (R) 1981.

44. *AECMA Recommendation for Geometry Data Exchange*, October, 1980.

45. Hordeski M. F., *CAD/CAM Techniques*, Reston Publishing Company, Inc., 1986. p508-512.

46. *CADDS 4X Bezier Geometry Reference, Software Revision 5.0*, Computervision Corporation, 1983.

47. *CADDS 4X Surface Reference, Software Revision 5.0*, Computervision corporation, 1983.

48. *CADDS 4X Advanced Surface Design Reference, Software Revision 5.0*, Computervision corporation, 1983.

49. *DUCT (Design Using Computer Techniques), Version 4.4*, Delta Computer Aided Engineering Limited., 1987

50. Lambourne E. B. & Bandura P., *Computer Graphics as an Aid to the Planning and Estimating of Surface Machining Operations*, Proceedings of Twenty-fourth International Machine Tool Design and Research Conference, 31 August-1st September 1983 in Manchester, U.K.

51. *CADDS 4X NC Vision Reference, Software Revision 5.0*, Computervision corporation, 1984.

52. *MicroCADDS Geometric Construction and Detailing User Guide, Revision 3.0*, Computervision corporation, 1987

53. *MicroCADDS Surfaces User Guide, Revision 3.0*, Computervision corporation, 1987.

54. *NC Processor User Guide, Revision 1.0*, Computervision corporation, 1986

55. Gao X., Davies B. J. and Harrison D. K., *The Machining of Free Form Surfaces with PC Based Software*, Proceedings of the Computervision European User's Conference, Rome, 15-18 March, 1988.

56. Crippa G. & Faltoni G., *Design of Objects Made with Plastics: The Use of CAE Techniques on a*

Personal Computer, Conference Papers of Computervision European Users Conference in Rome, 15-18 March, 1988.

57. Takeuchi Y., Morita K. & Sakamoto M., *Development of a Personal CAD/CAM System for Metal Moulds*, Proceedings of Twenty-sixth International Machine Tool Design and Research Conference, 17-18 September, 1986, Manchester, U.K.

58. *User Programming Language Reference, Revision 2.0*, Computervision Corporation, 1985.

59. Tiller W., *Rational B-splines for curve and surface representation*, IEEE Computer Grapgics and Applications. September, 1983.

60. Piegl L., *Hermite- and Coons-like interpolants using rational Bezier approximation form with infinite control points*, Computer Aided Design, Volume 20, Number 1, Jan/Feb 1988.

61. Piegl L., *Interactive Data Interpolation by Rational Bezier Curves*, IEEE Computer Grapgics and Applications, Volume 7, Number 4. April 1987.

62. Ball A. A., *CONSURF Part one: introduction of the conic lofting tile*, Computer Aided Design, Volume 6, Number 4. October 1974.

63. Ball A. A., *CONSURF Part two: description of the algorithms*, Computer Aided Design, Volume 7, Number 4. October 1975

64. Ball A. A., *CONSURF Part three: How the program is used*, Computer Aided Design, Volume 9, Number 1. January 1977.

65. Gao X., Harrison D. K. & Davies B. J., *An approach to low cost Computer Aided Manufacture of free form surfaces*, Proceedings of Institution of Mechanical Engineers.1989.

66. Hayes J. G. (editor), *Numerical Approximation to Functions and Data*, Institute of mathematics

and its applications, 1970

67. Clark J. H., *Designing Surfaces in 3D*, Communications of the ACM, Volume 19, Number 8, August 1976, p454-460

68. Barsky B. and Greenberg D., *Interactive Surface Representation System using a B-spline formulation with interpolation capability*, Computer Aided Design, Volume 14, Number 4, July 1982. p187-194

69. Boehm W., *Generating the Bezier points of B-spline curves and surfaces*, Computer Aided Design, Volume 13, Number 6 November 1981. p365-366

70. Bezier P., *Mathematical and practical possibilities of UNISURF*, Computer Aided Geometric Design, 1975.

71. Beeker E., *Smoothing of shapes designed with free-form surfaces*, Computer Aided Design, Volume 18, Number 4, May 1986. p224-231

72. Bourne D. E. & Kendall P. C., *Vector Analysis and Cartesian Tensors*, Published by Thomas Nelson and Sons Ltd., 1977. p33, p43-44.

73. Idesawa M., *A system to generate a solid figure from three view*, Bull. JSME volume 16, Number 92. February, 1973, p216-225.

74. Idesawa M. & Shibata S., *Automatic input of line drawing and generation of solid figure from three-view data*, Proc. Int. Comput. Symposium 1975. Volume 2 p304-311.

75. Aldefeld B., *On automatic recognition of 3D structures from 2D representations*, Computer Aided Design, Volume 15, number 2, March 1983. p59-64.

76. Ho Bin, *Inputing constructive solid geometry representations directly from 2D orthographic*

engineering drawings, Computer Aided Design, Volume 18, Number 3, April 1986. p147-155.

77. Richards T. H. & Onwubolu G. C., *Automatic interpretation of engineering drawings for 3D surface representation in CAD*, Computer Aided Design, Volume 18, Number 3, April 1986. p156-160.

78. Wang L. X. (editor), *Handbook of mathematics*, Higher education publishing Ltd. Dec. 1977. p60-78

79. Choi B. K., Lee C. S., Hwang J. S. & Jun C. S., *Compound surface modelling and machining*, Computer Aided Design, Volume 20, Number 3, April 1988. p127-136.

80. Loney G. C. & Ozsoy T. M., *NC machining of free form surfaces*, Computer Aided Design, Volume 19, Number 2, March 1987. p85-90.

81. Mortenson M. E., *Geometric Modelling*, John Wiley & Sons, 1985.

82. Melvin J. M., *Numerical Analysis: A practical approach*, Macmillan Publishing Co., Inc. 1982 p60.

83. Lafara R. L., *Computer methods for science and engineering*, International Textbook Company Ltd., 1973.

Appendix One

FIGURES

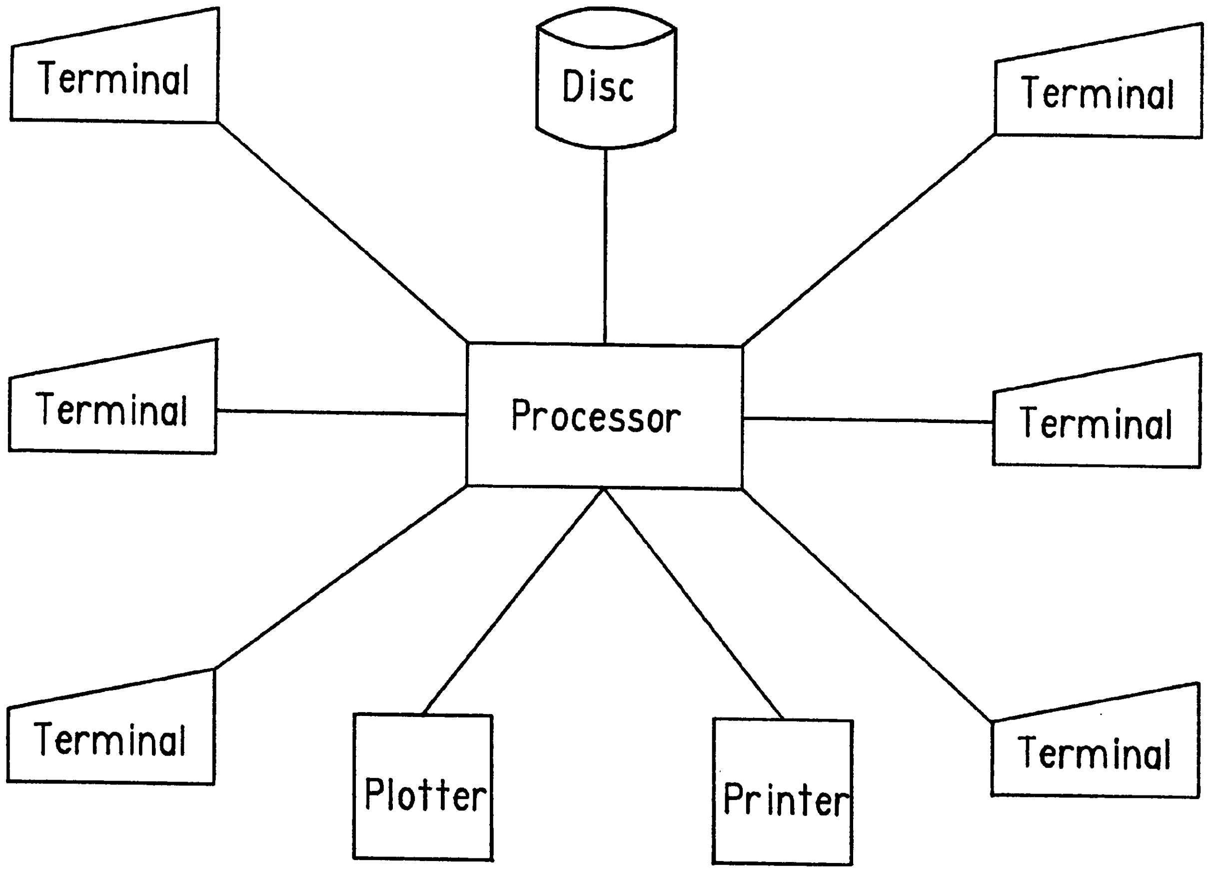


Figure 2.1 Centralized computer configuration.

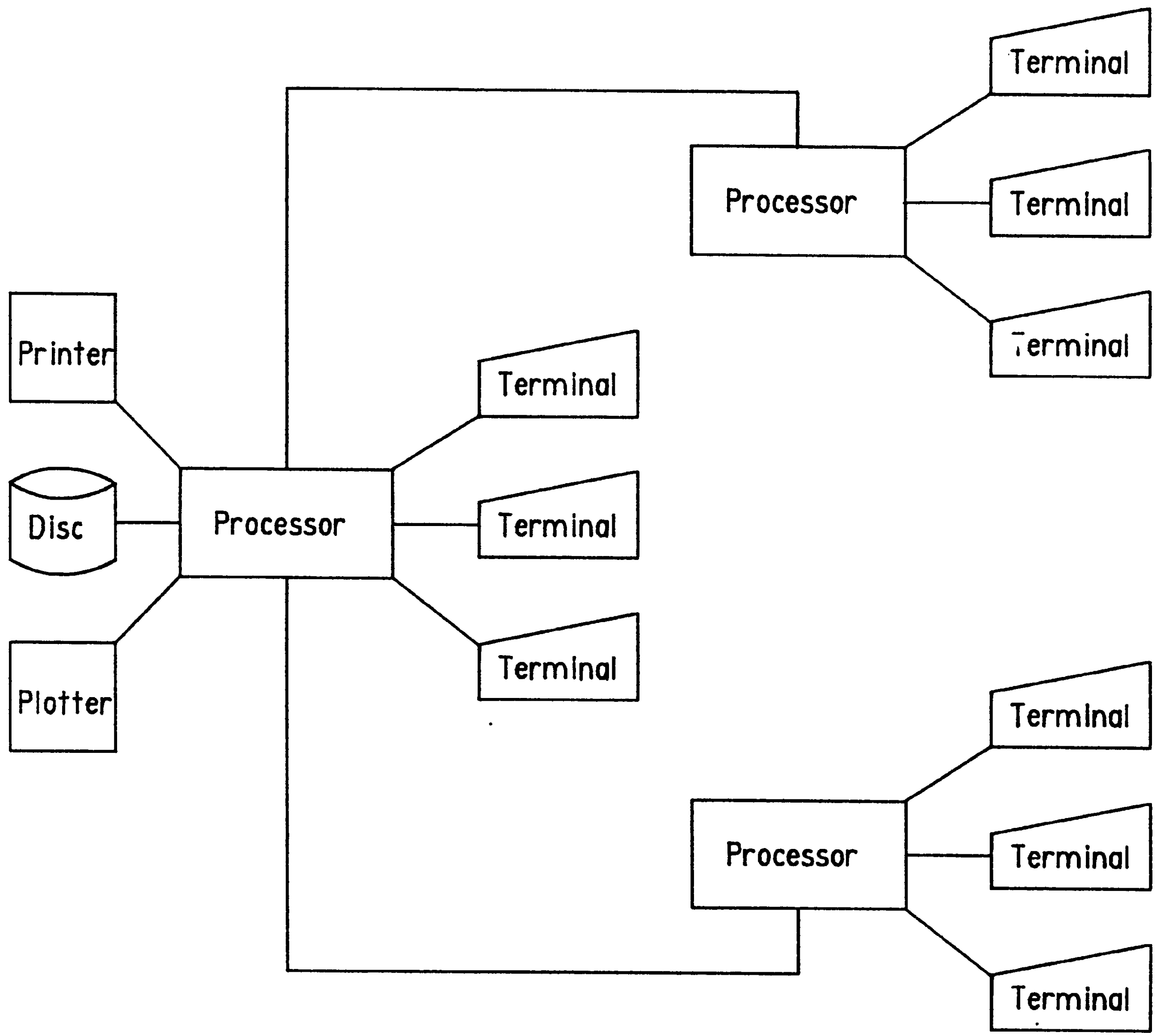


Figure 2.2 Multiple host computer configuration.

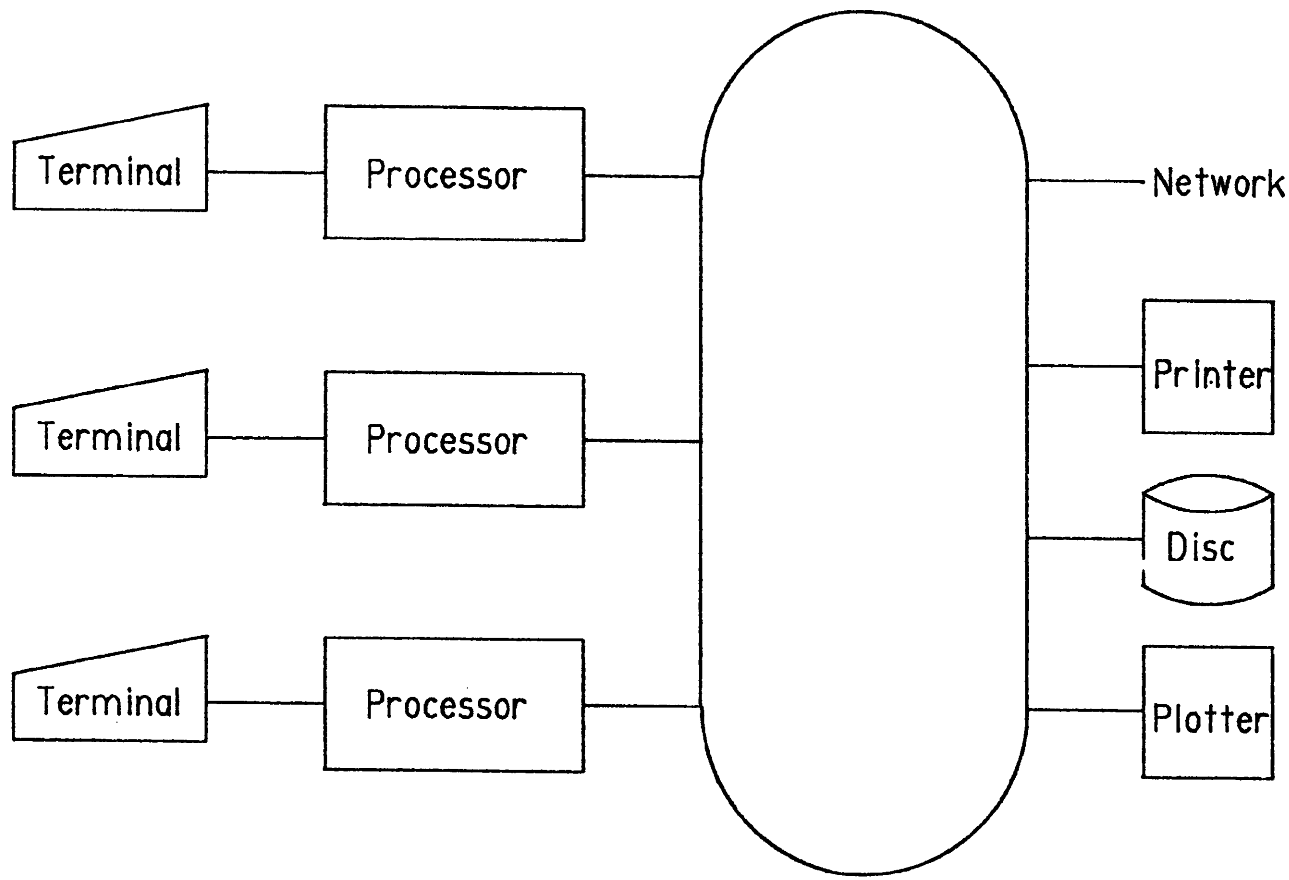


Figure 2.3 Distributed computer system

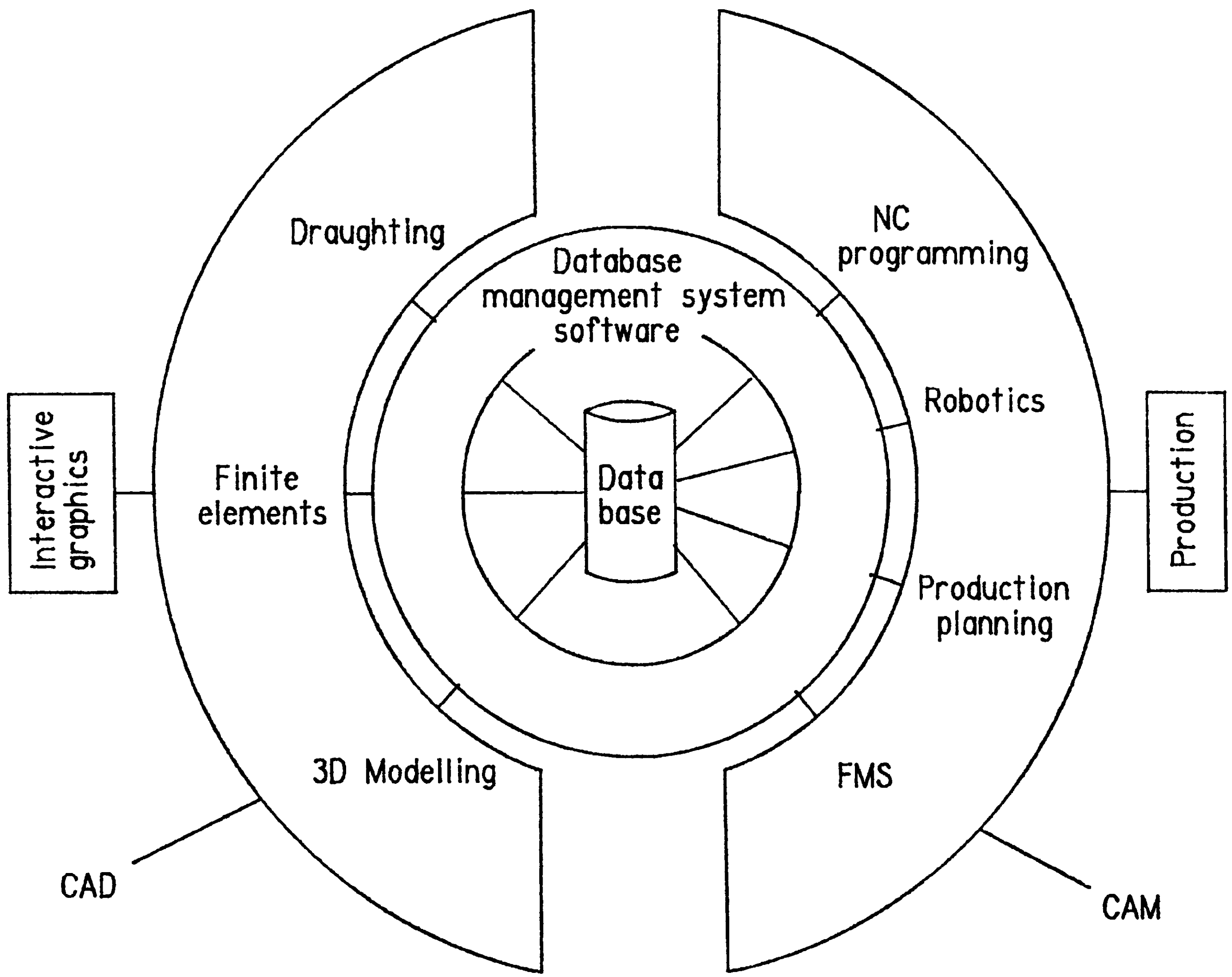


Figure 2.4 CAD/CAM activities sharing a common database

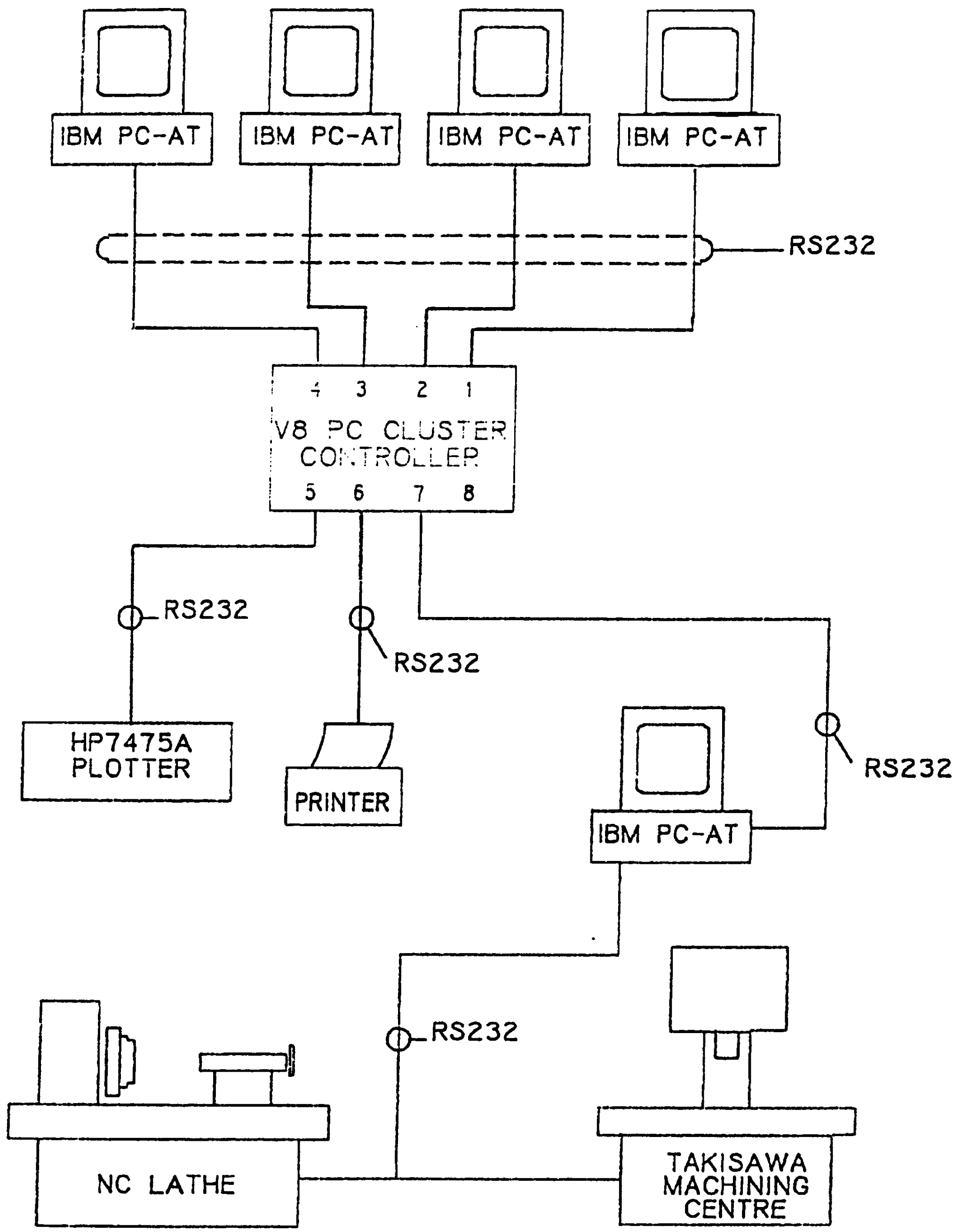


Figure 3.1 The network of the PC CAD/CAM system at UMIST

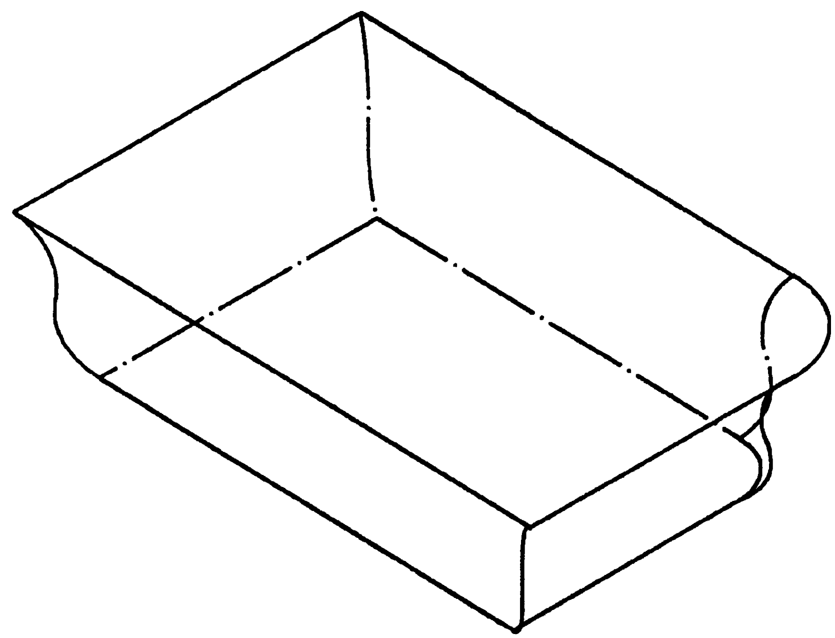


Figure 3.2 A typical 3D pocket which can be machined with Personal Machinist

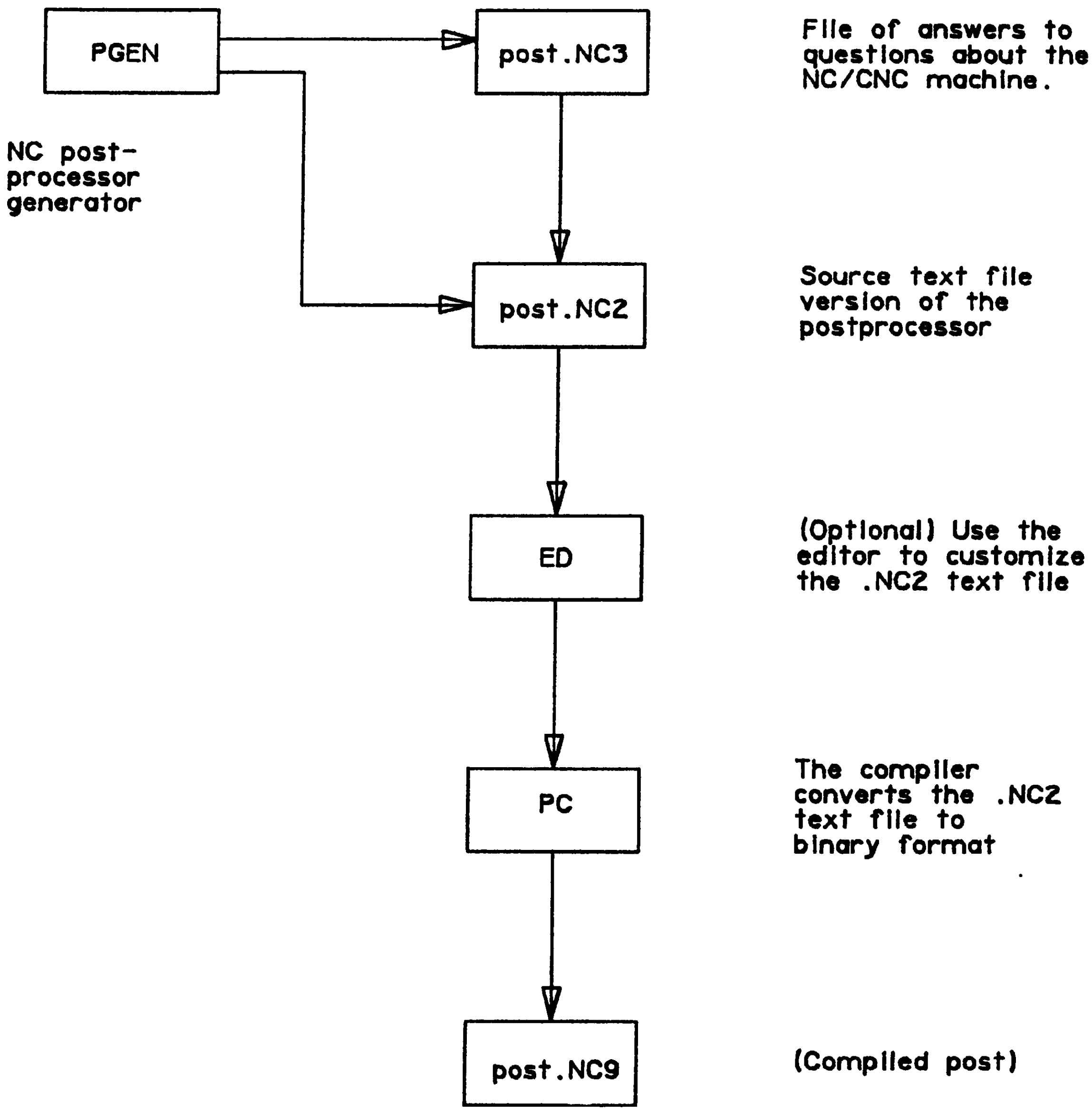


Figure 3.3 The steps of developing a postprocessor at DOS level

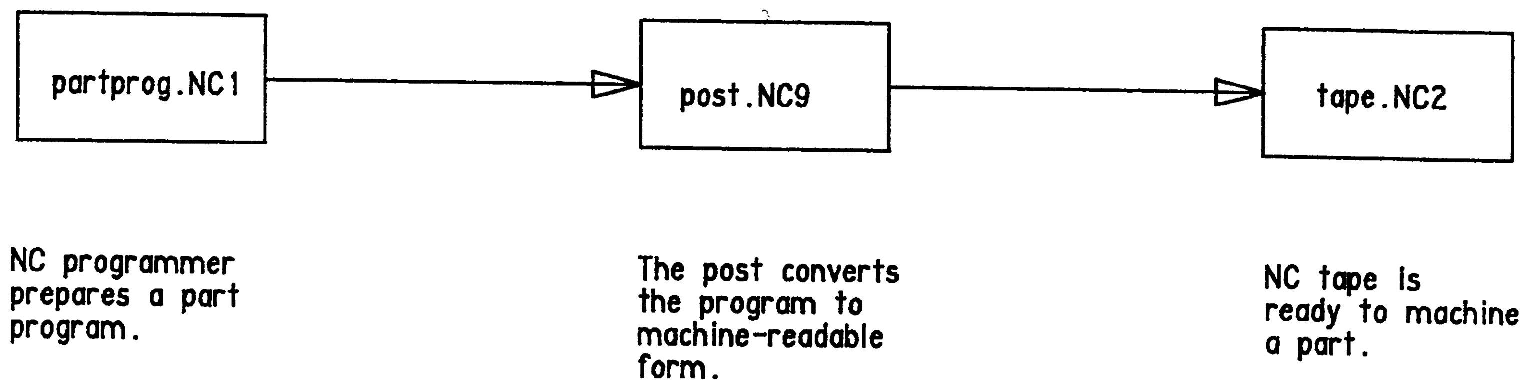


Figure 3.4 The post processing of part programs.

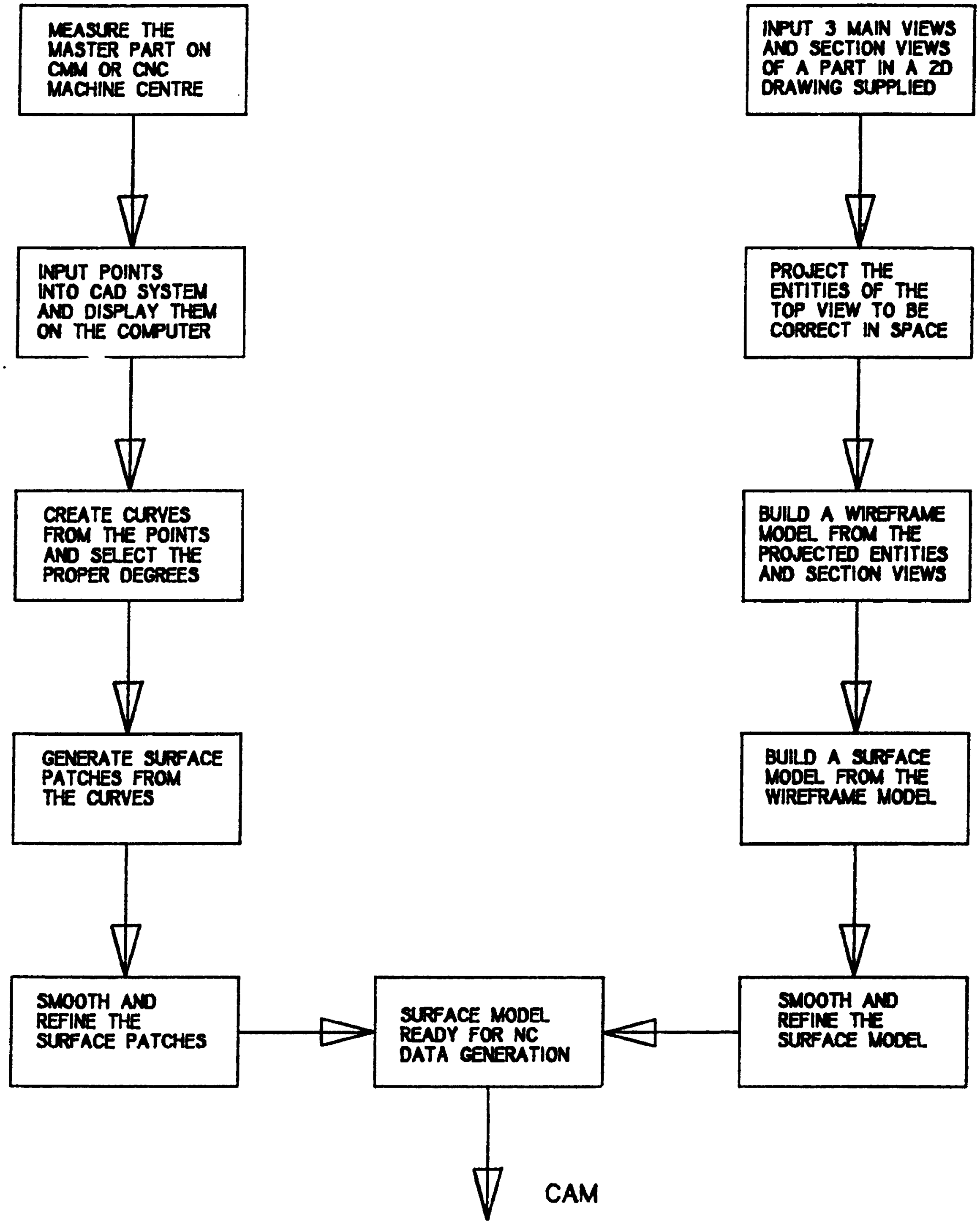


Figure 3.5 The procedure of modelling a component with microCADDS

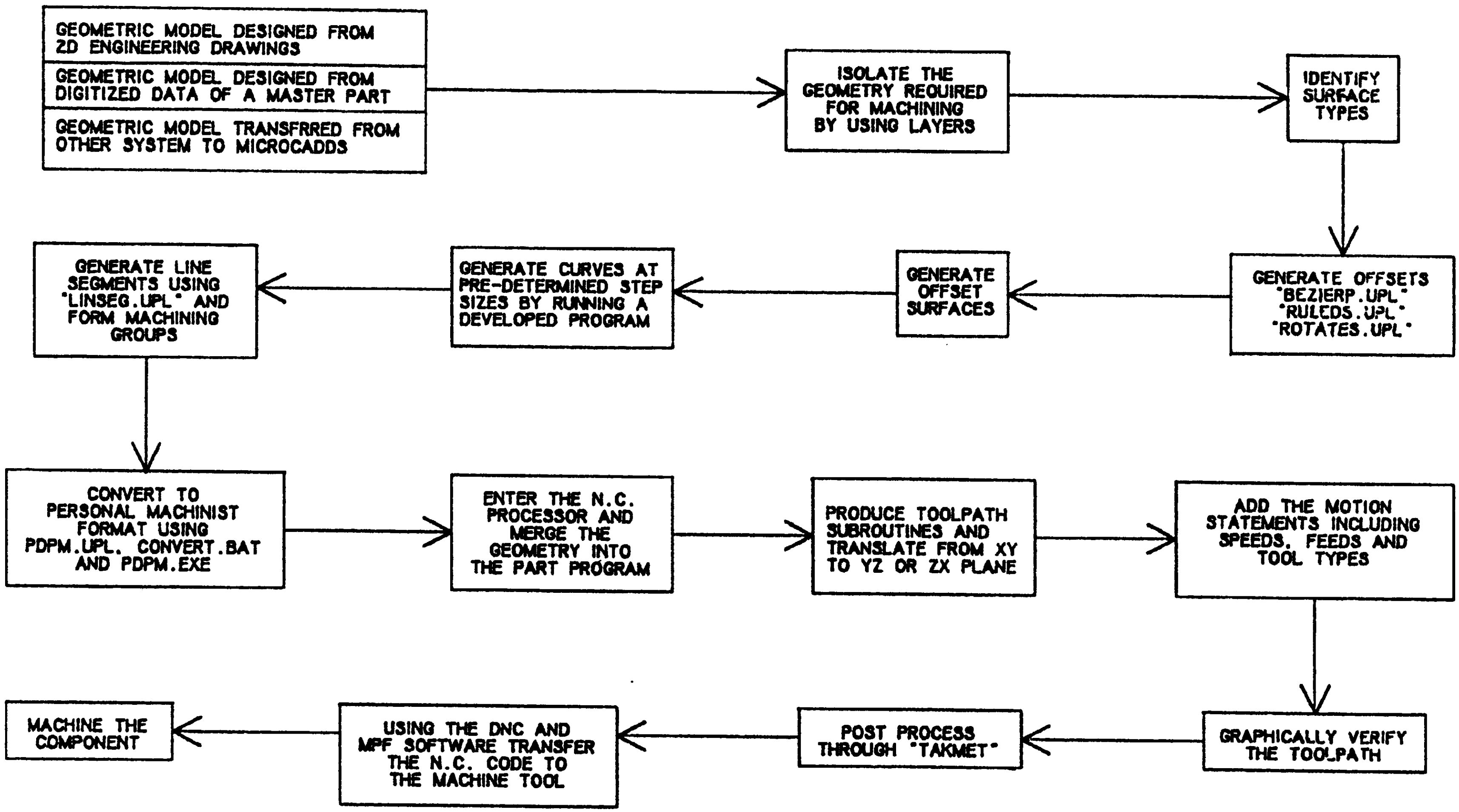


Figure 3.6 Flowchart of the general procedure from model to machined part

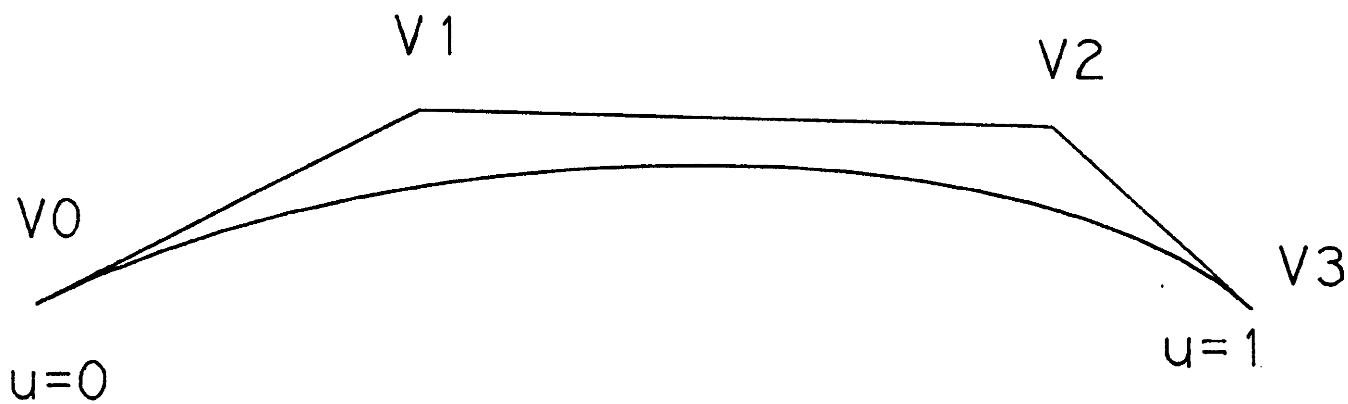


Figure 4.1 A Bezier cubic curve

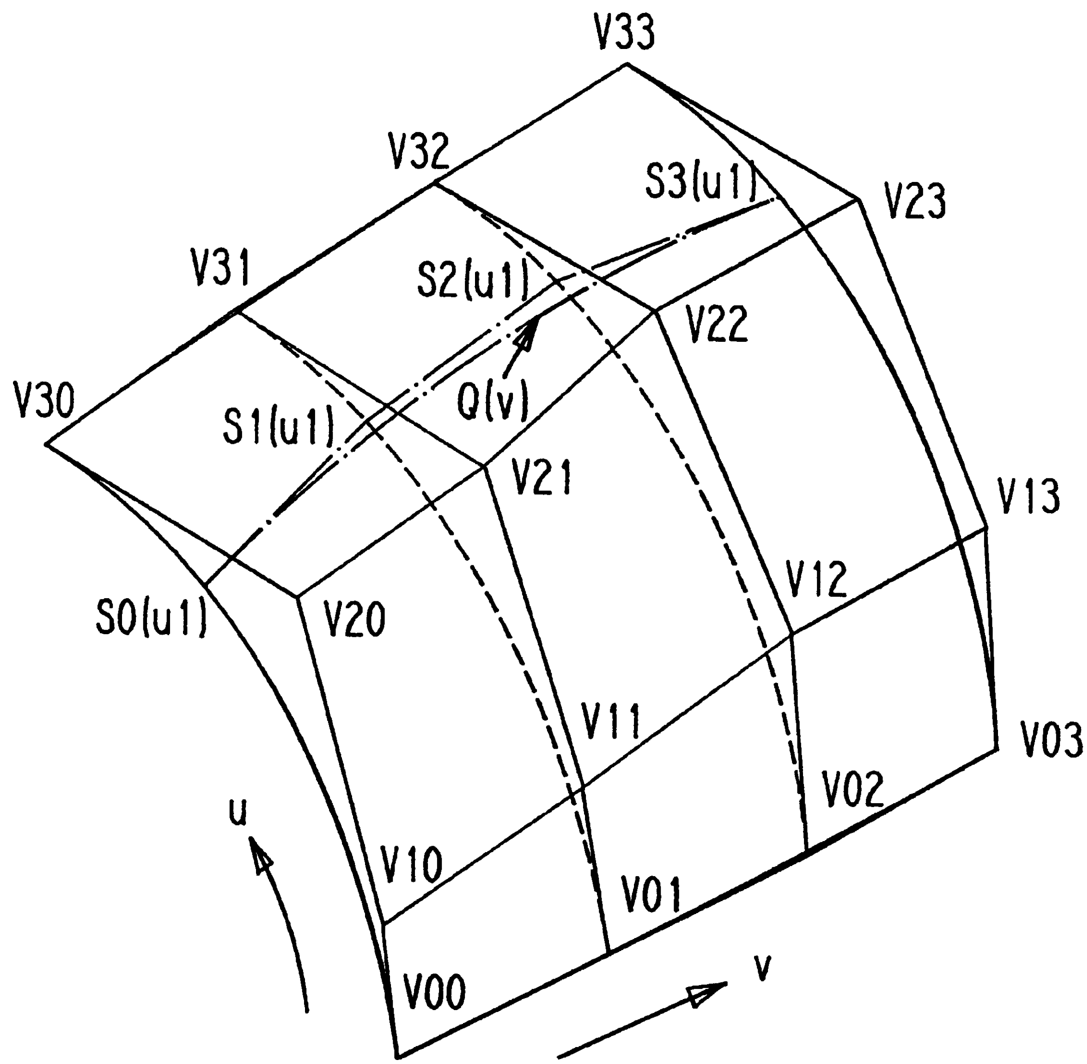


Figure 4.2 The formation of a bi-cubic surface

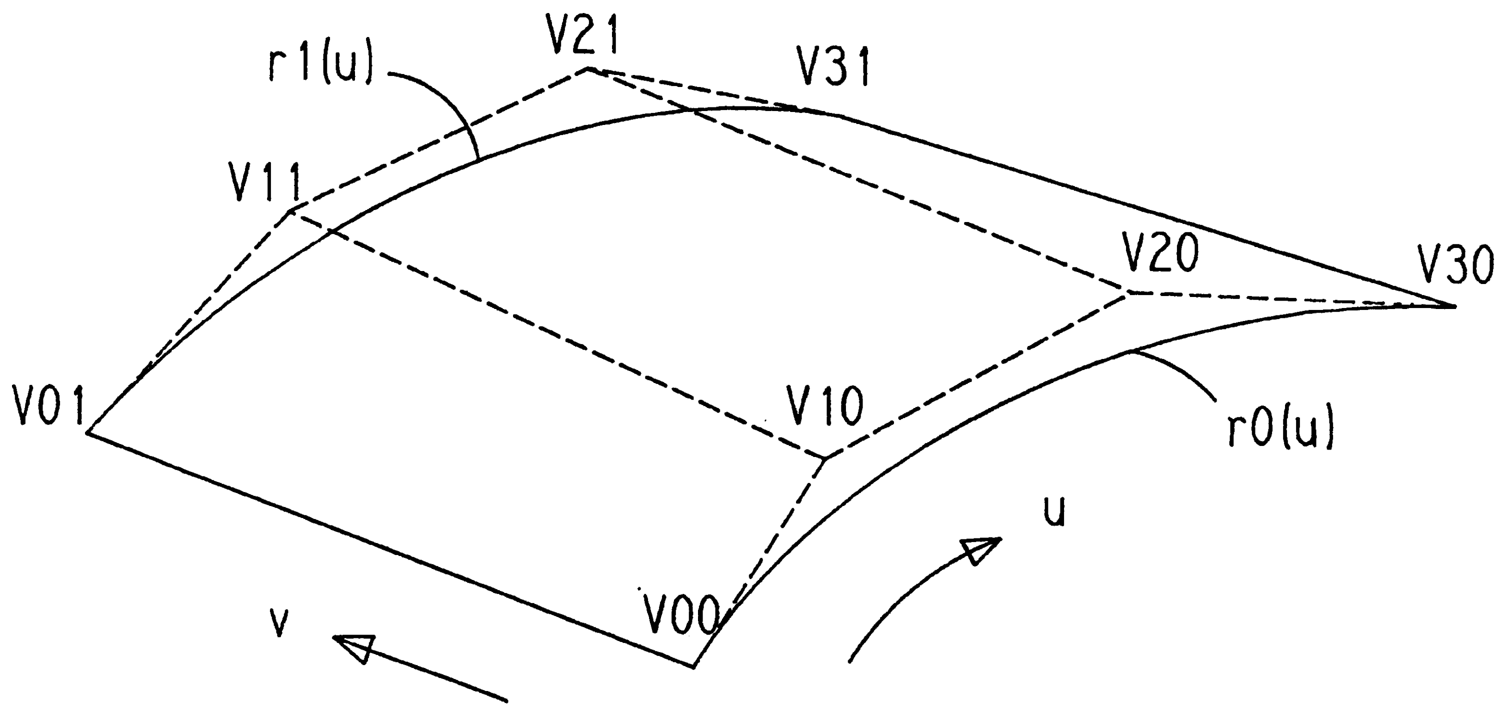


Figure 4.3 A ruled surface

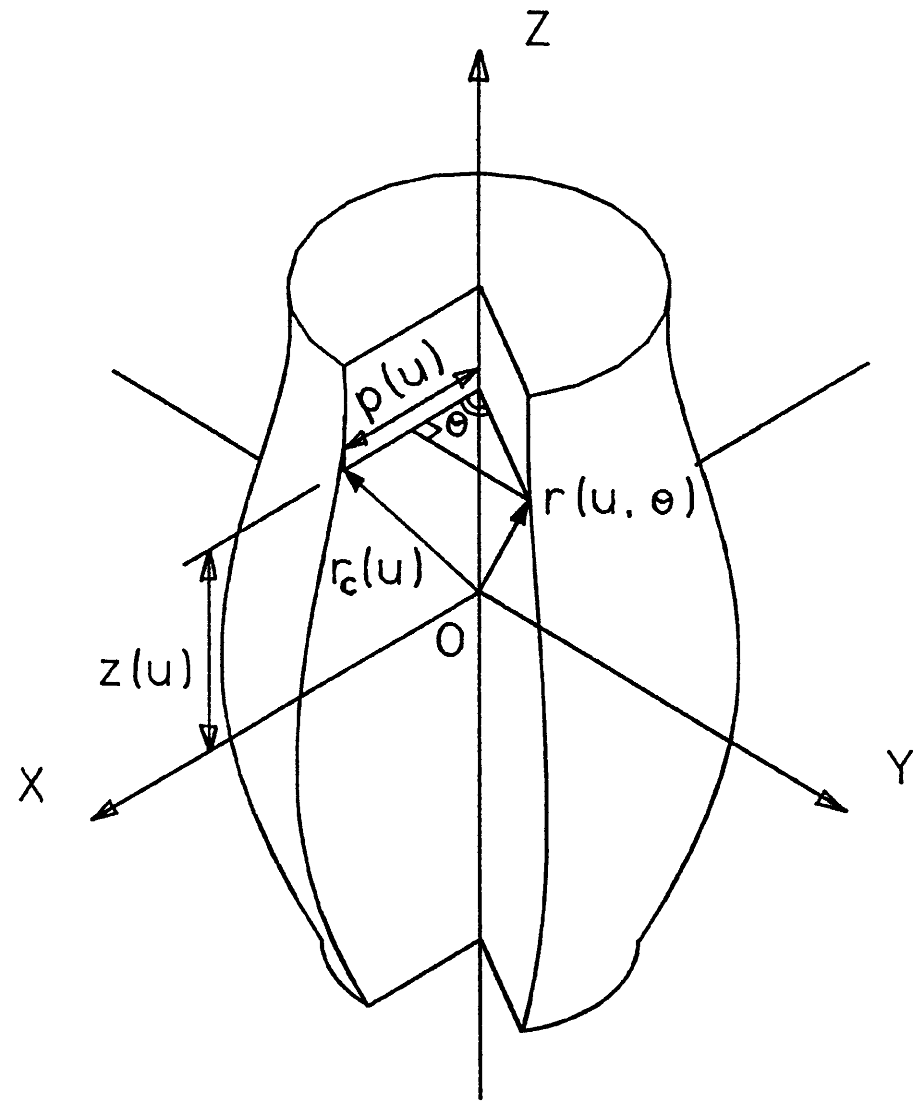


Figure 4.4 The formation of a surface of revolution

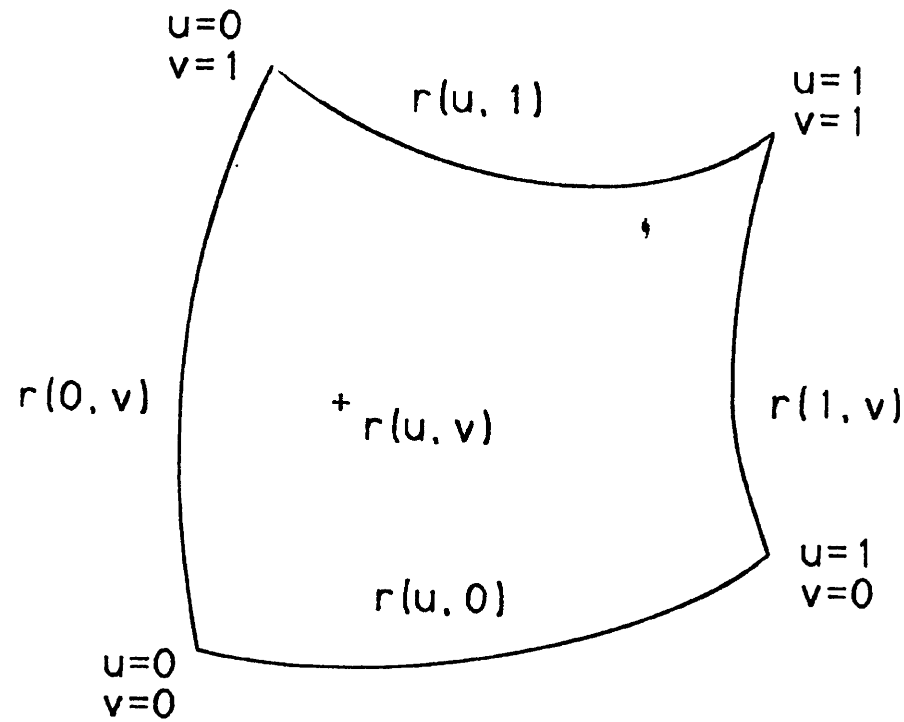


Figure 4.5 A Coons parametric patch

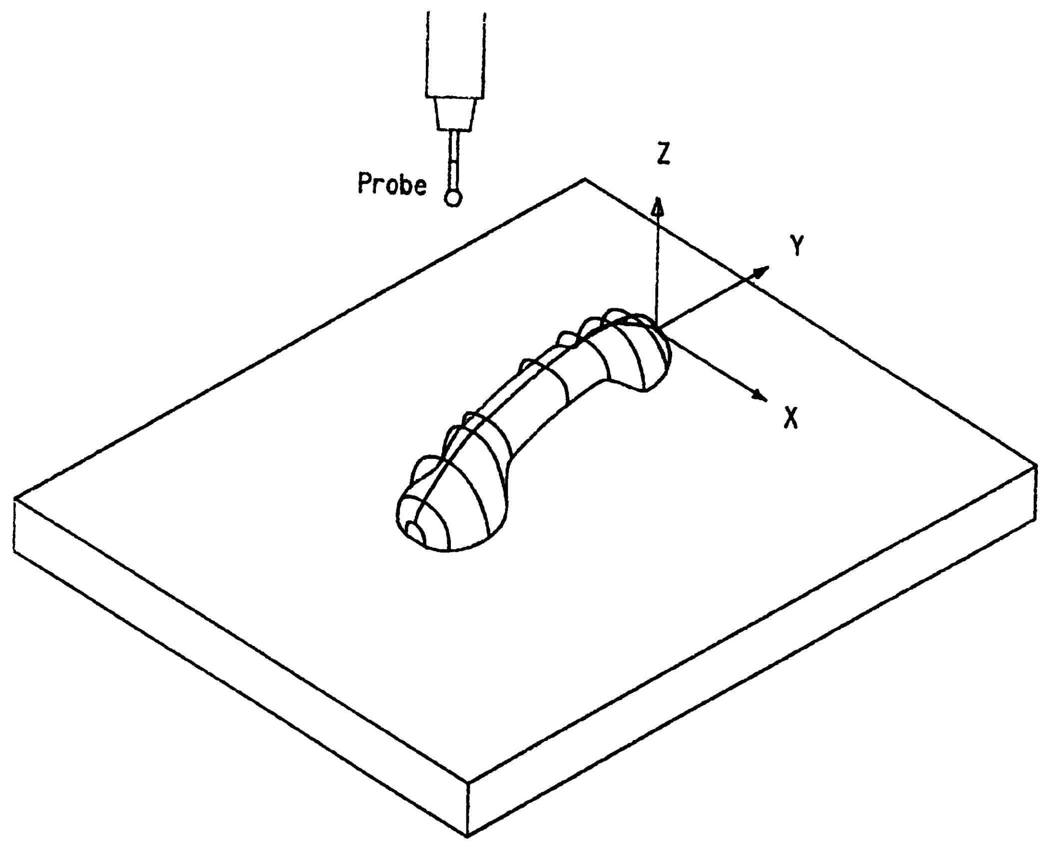


Figure 4.6 The Illustration of the set up of the telephone on CNC machining centre

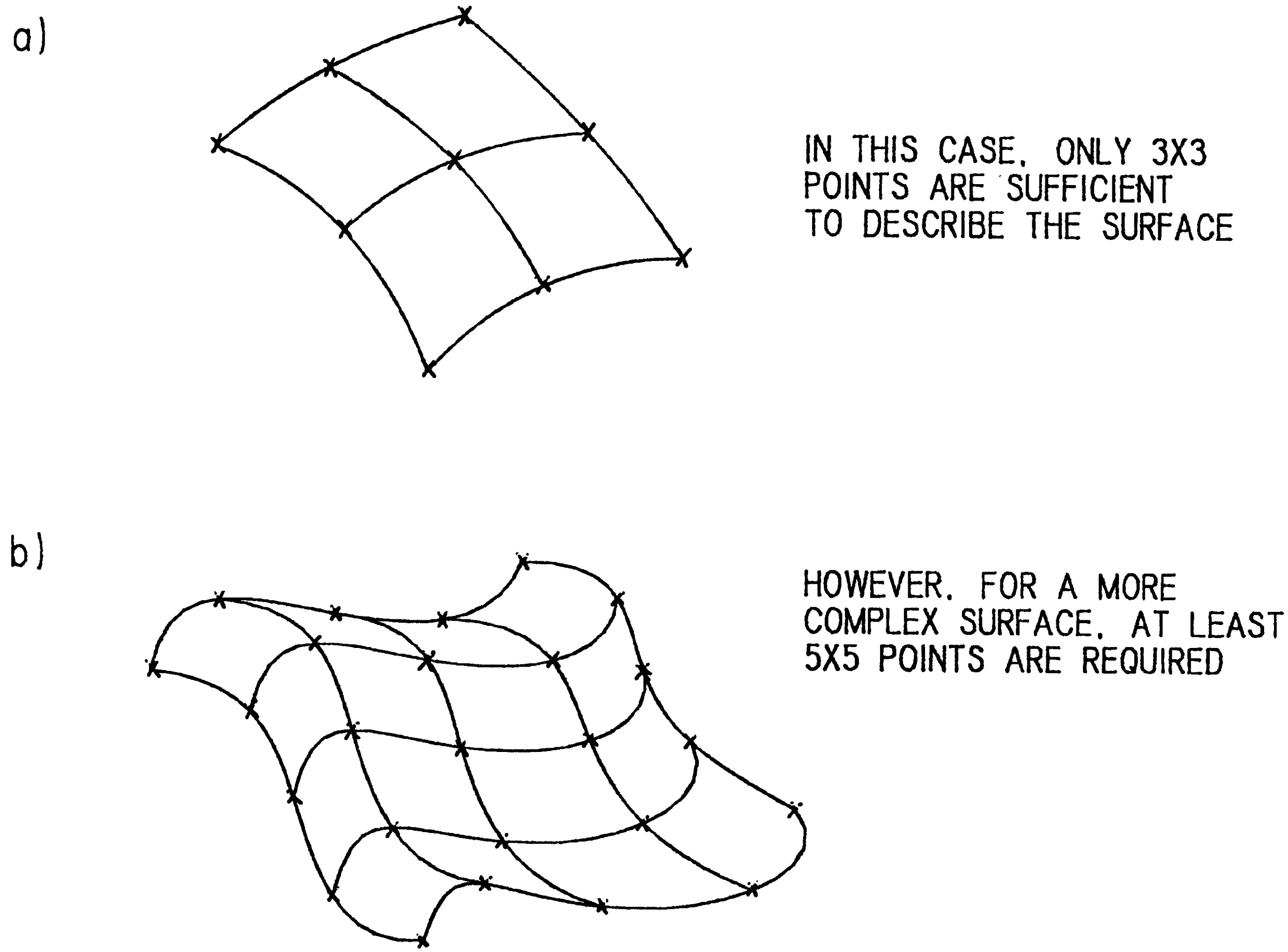


Figure 4.7 The complexity of a surface determines the number of points required to define it

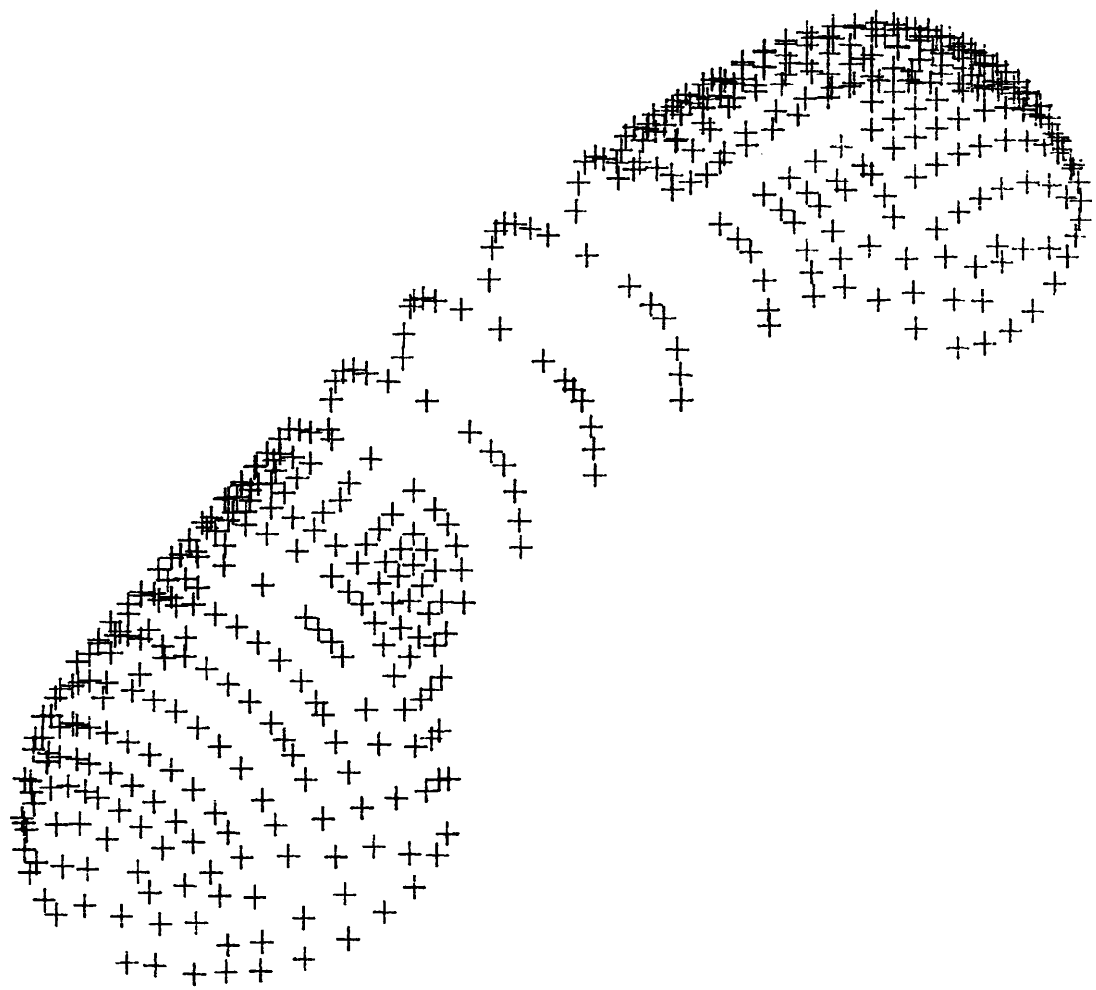


Figure 4.8 The digitized points from a telephone handset

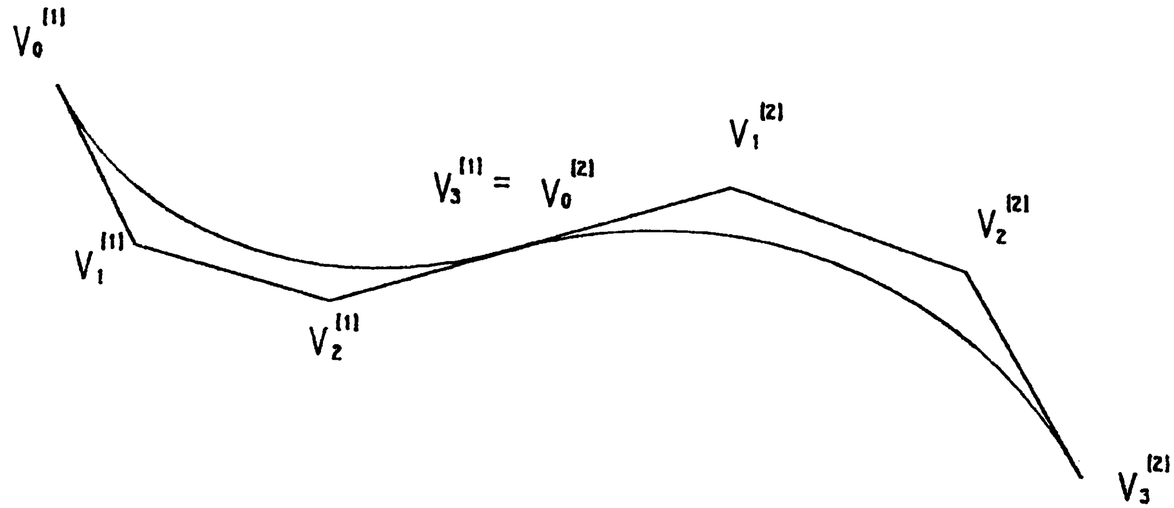


Figure 4.9 Composite curves

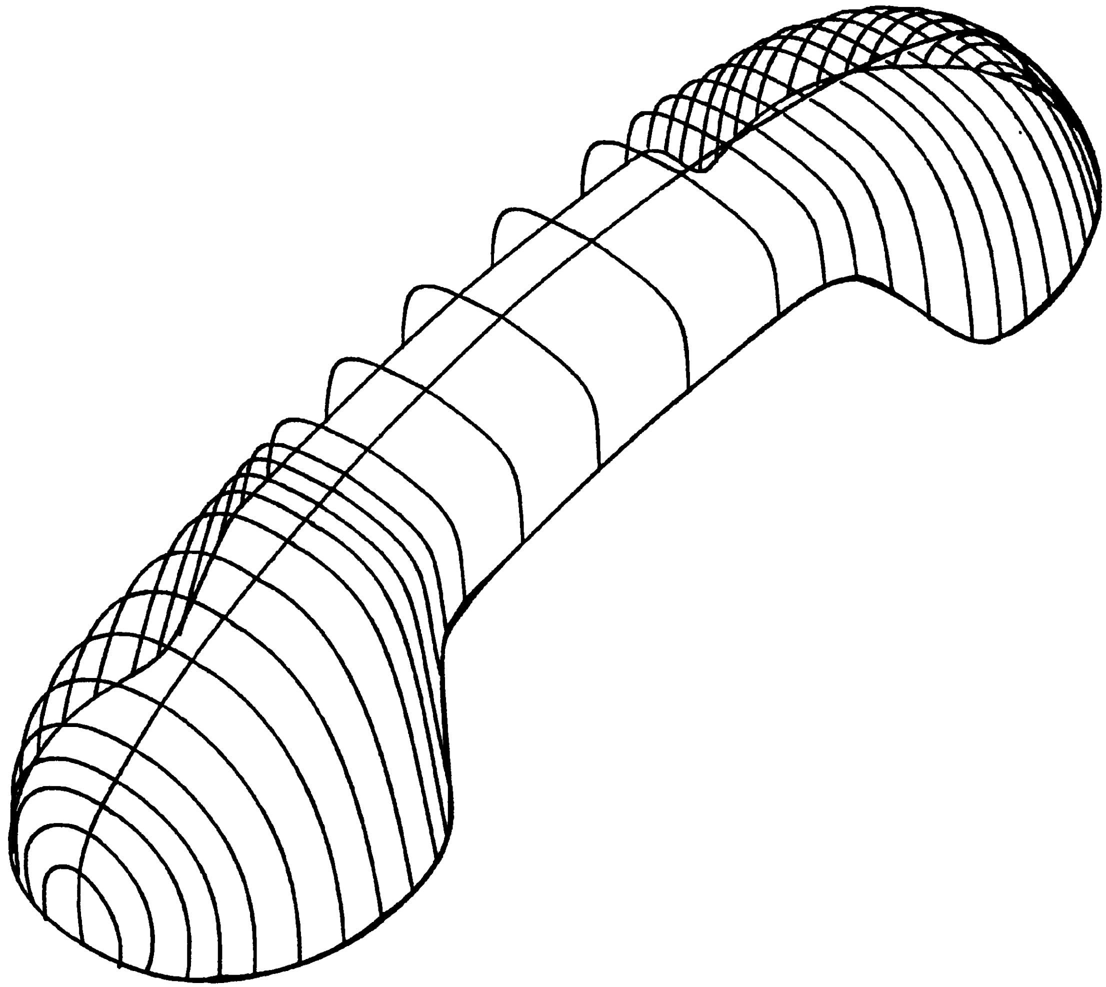
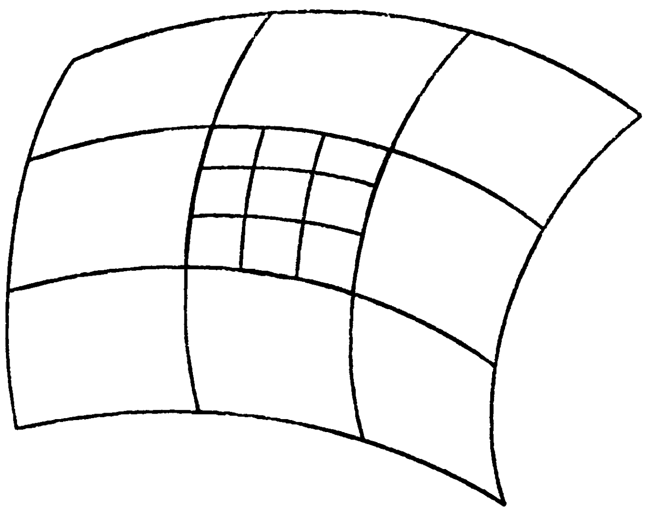
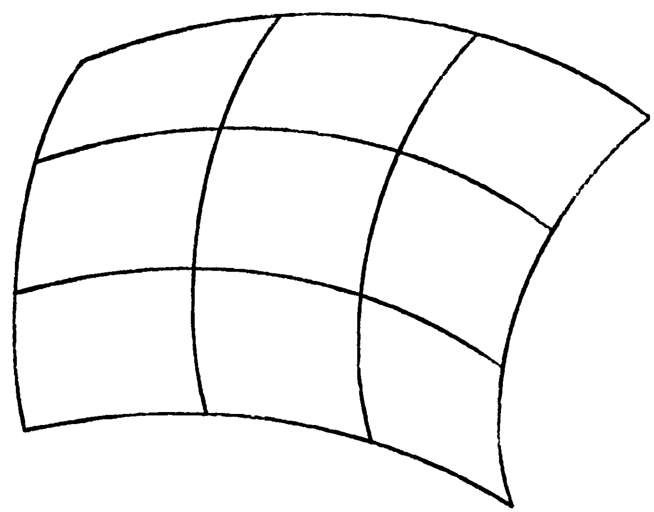


Figure 4.10 The curves created from digitized points



a) 9 Coons patches



b) a single Bezier patch

Figure 4.11 The defference between the Coons and Beziers method in the number of patches needed to cover the same area

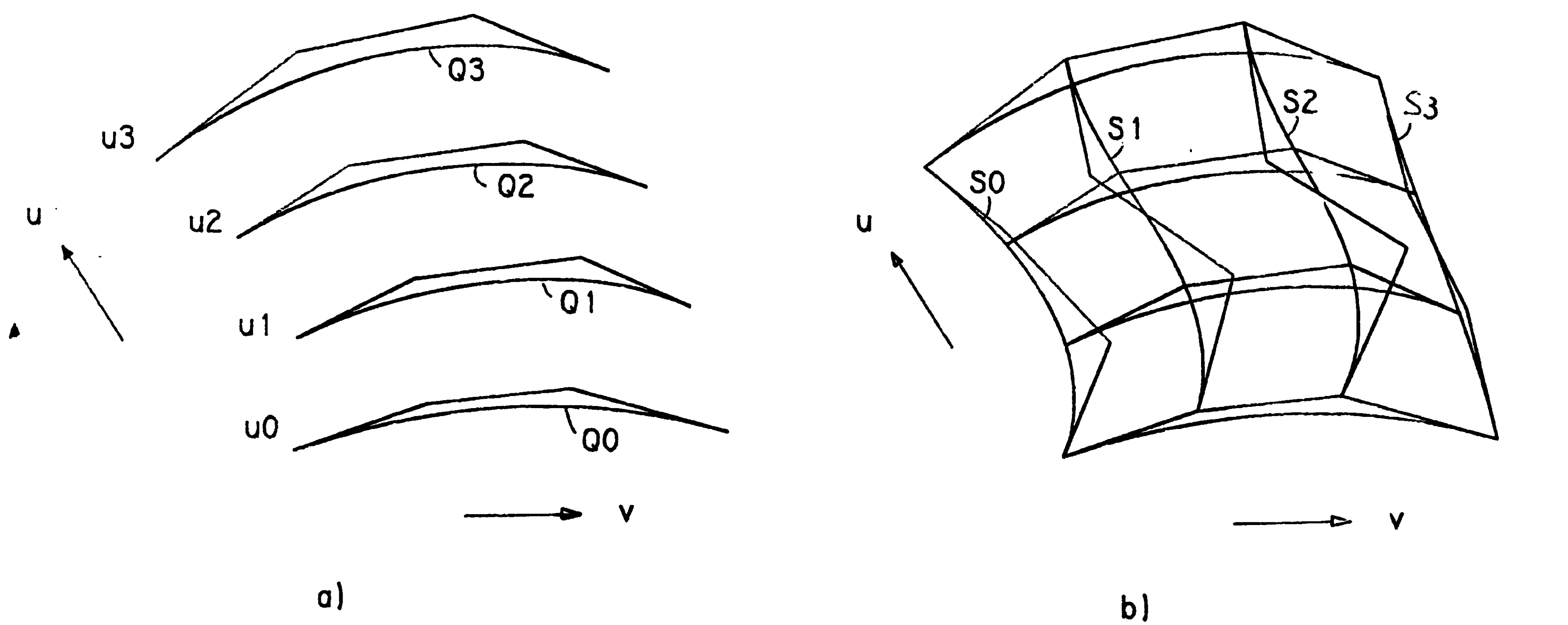


Figure 4.12 The inverse algorithm of Bezier surfaces

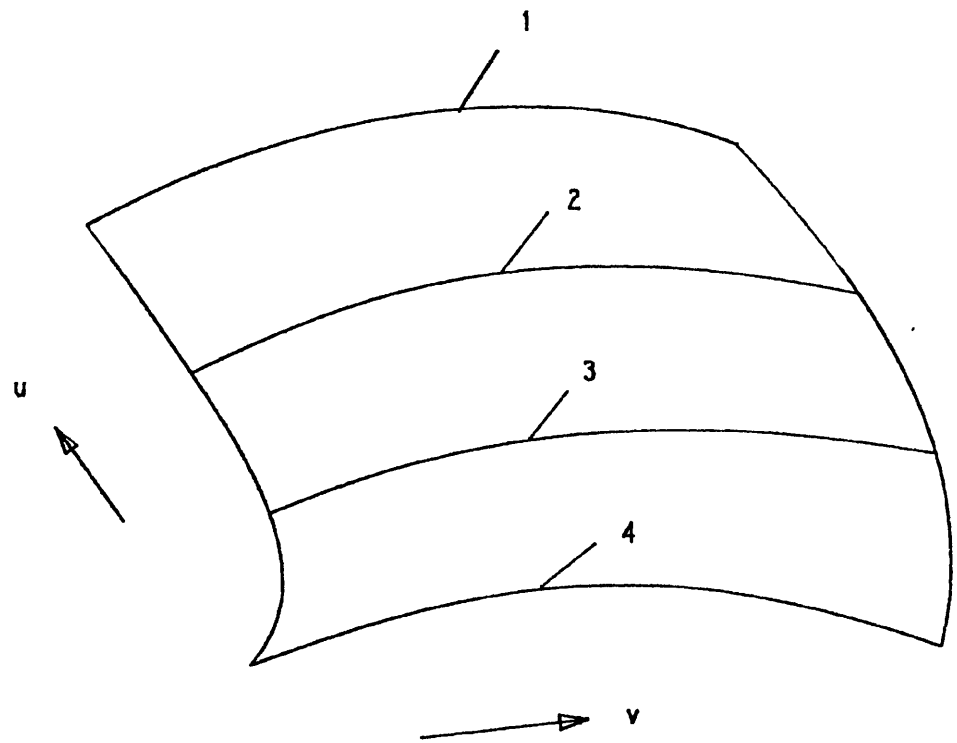


Figure 4.13 The surface patch passing through 4 given curves

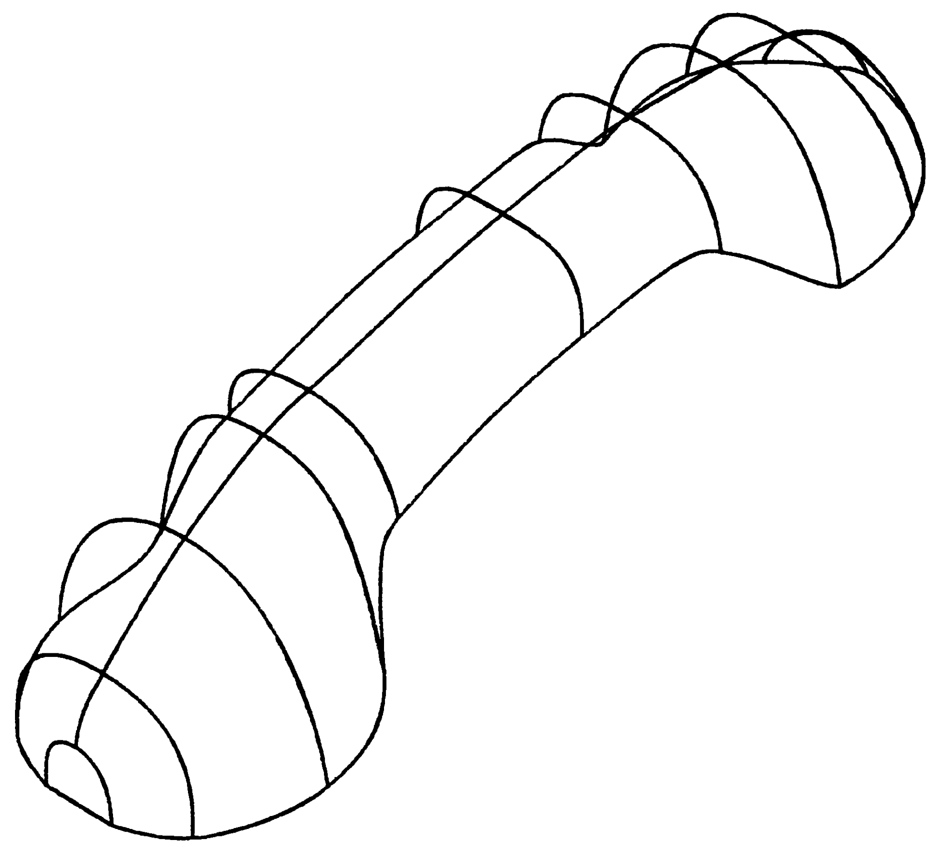


Figure 4.14 The surfaces generated from cross sectional curves

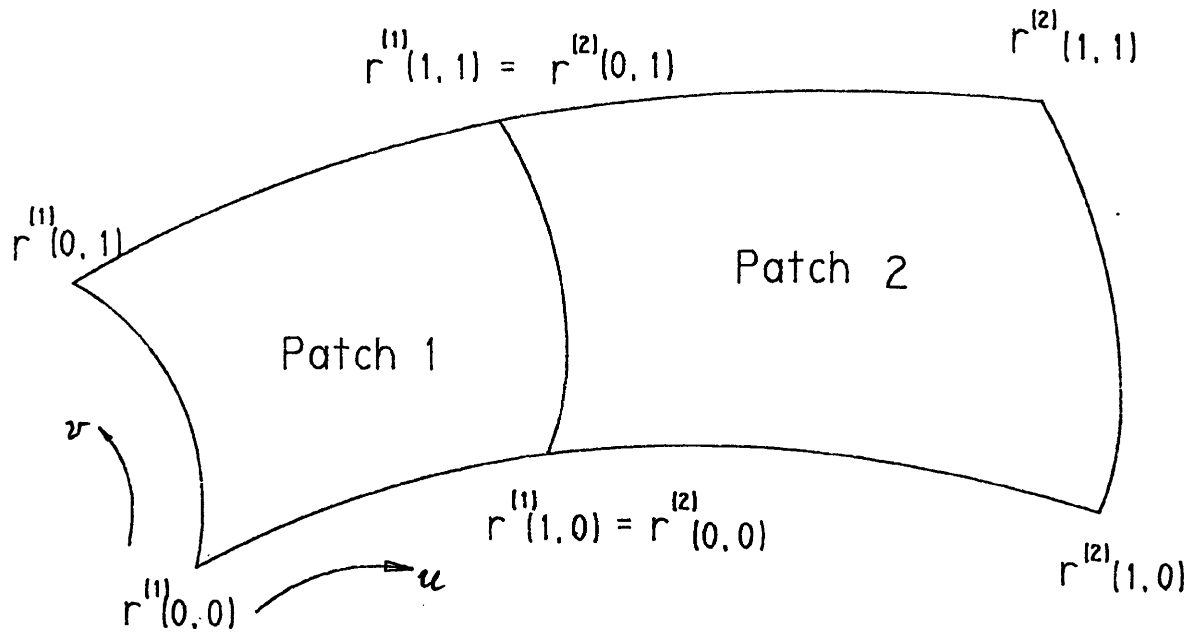


Figure 4.15 The positional continuity of composite surfaces

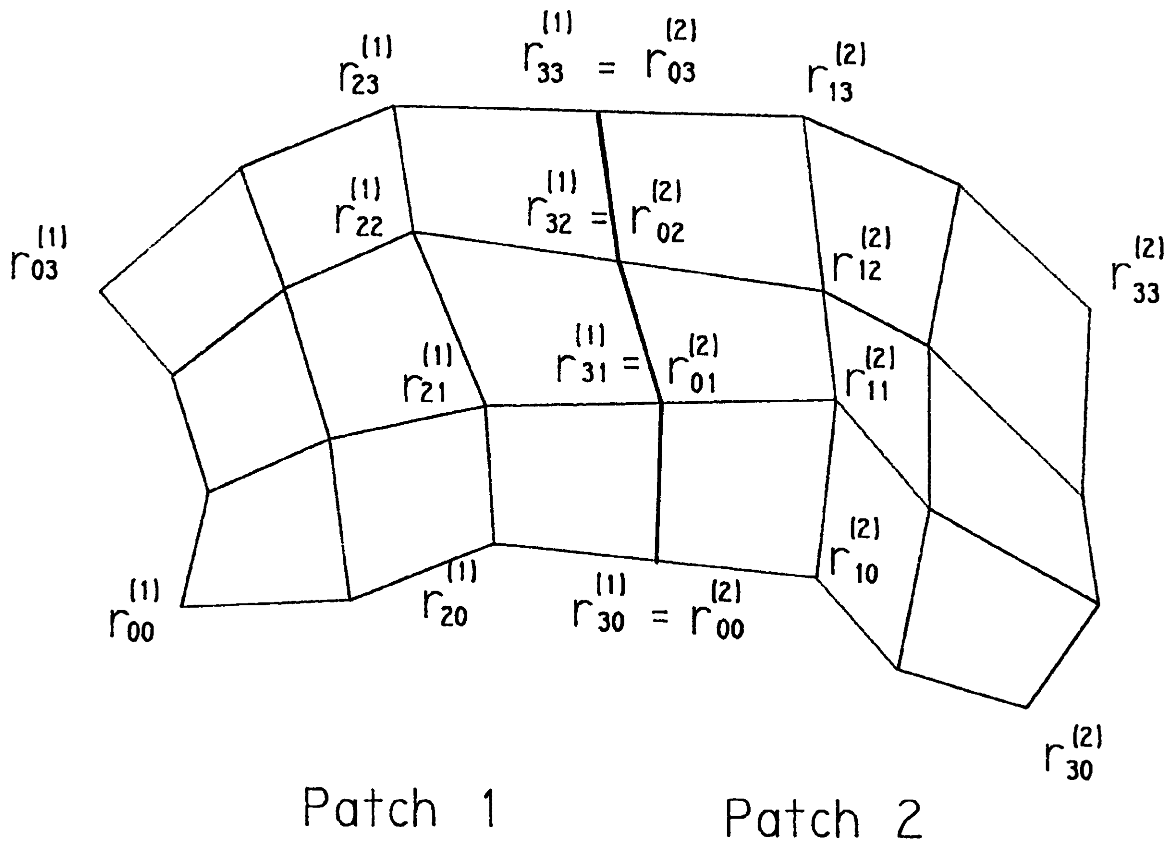


Figure 4.16 The polyhedra giving positional and gradient continuity between cubic Bezier patches

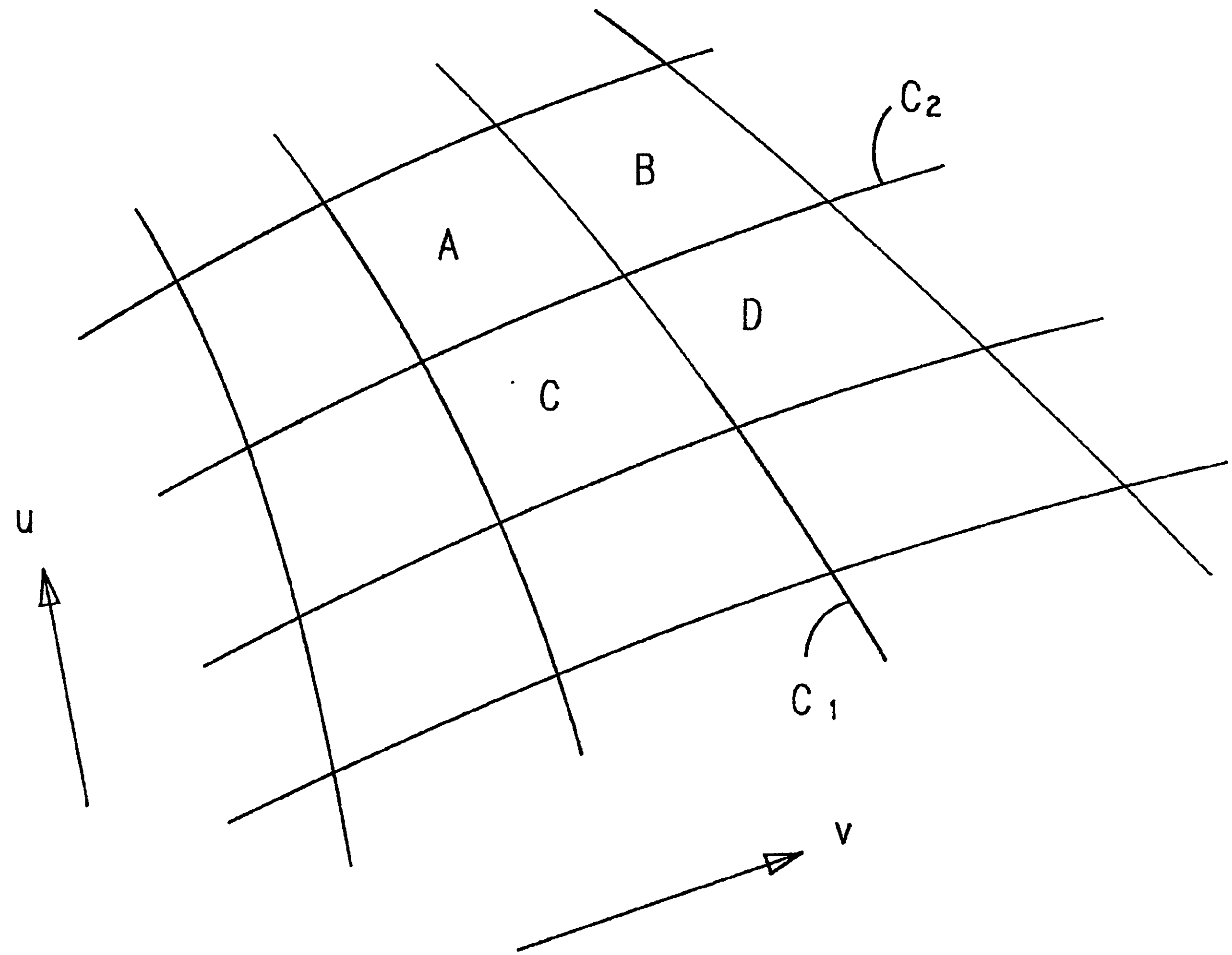


Figure 4.17 Constant tangent magnitude ratios across composite patch boundary curves C_1 and C_2

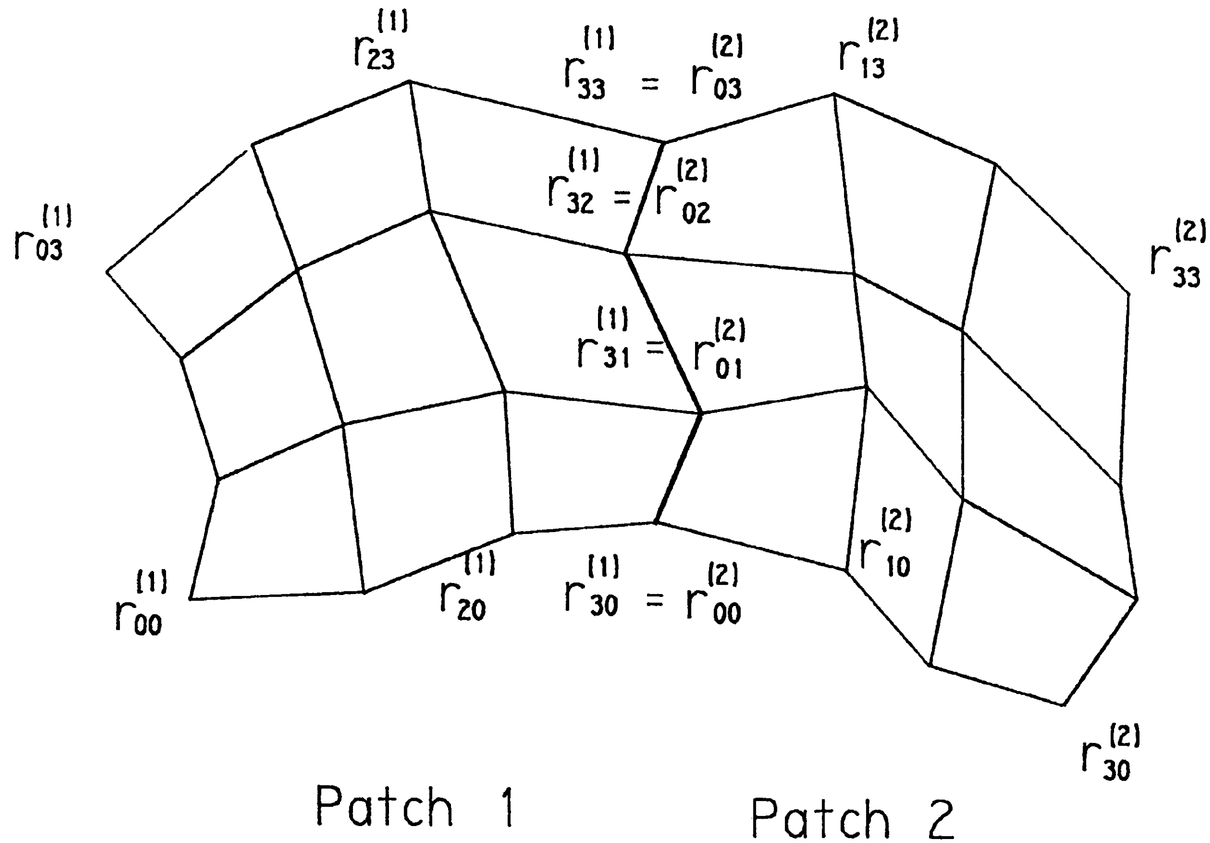


Figure 4.18 The polyhedra giving positional and gradient continuity between cubic curves

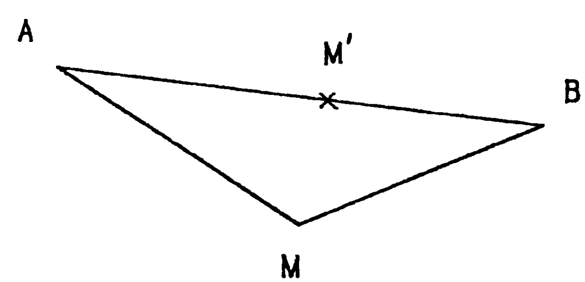


Figure 4.19 The calculation of colinearity

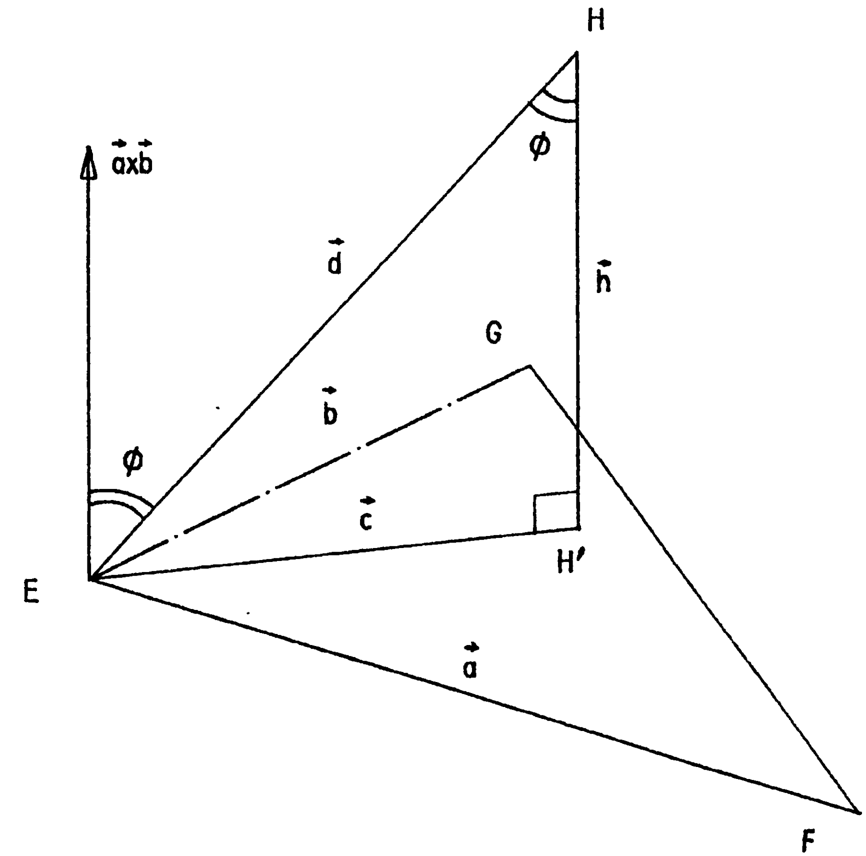


Figure 4.20 The calculation of coplanarity

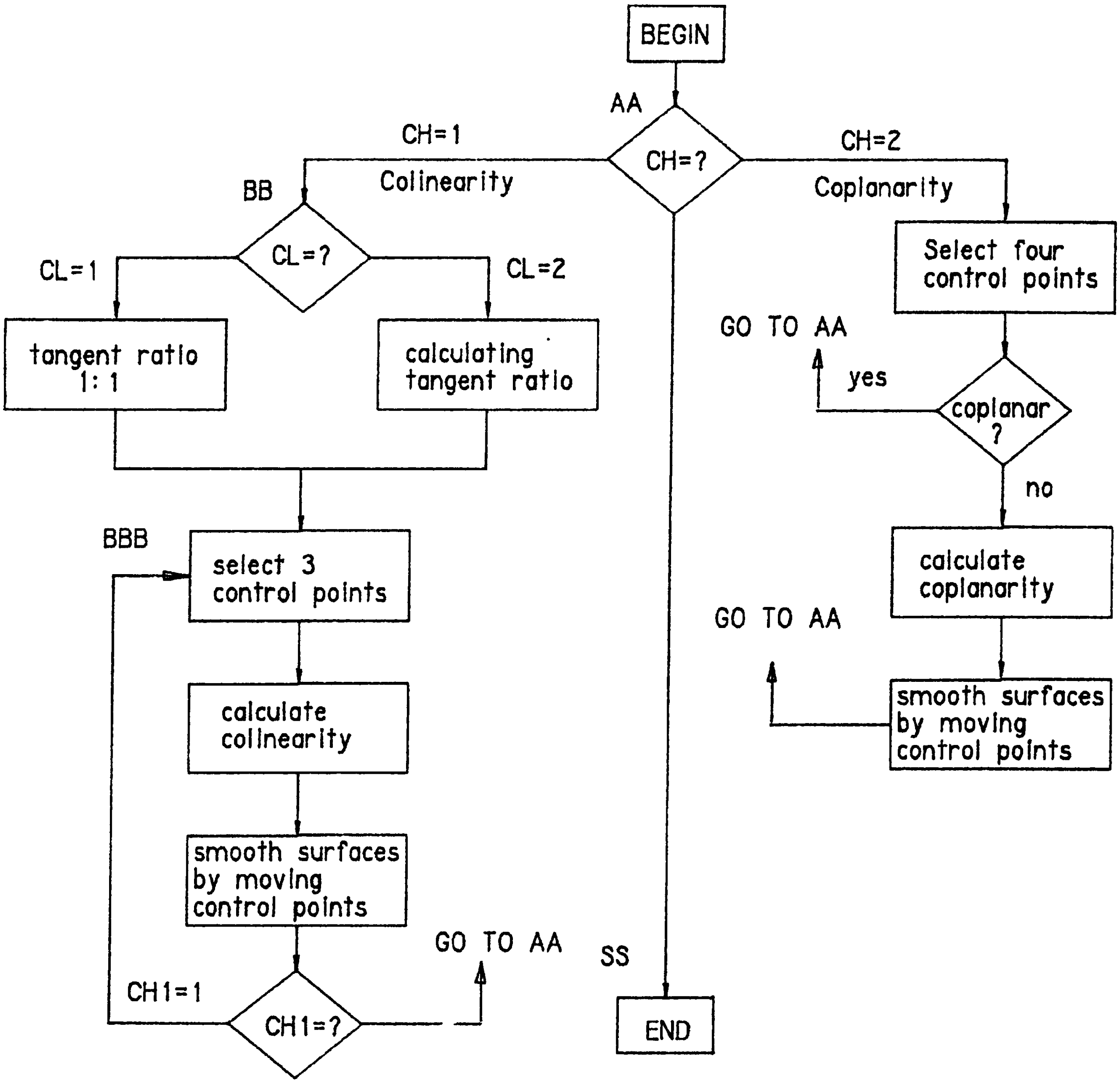


Figure 4.21 The flowchart of the smoothing program - SMOSPL.UPL

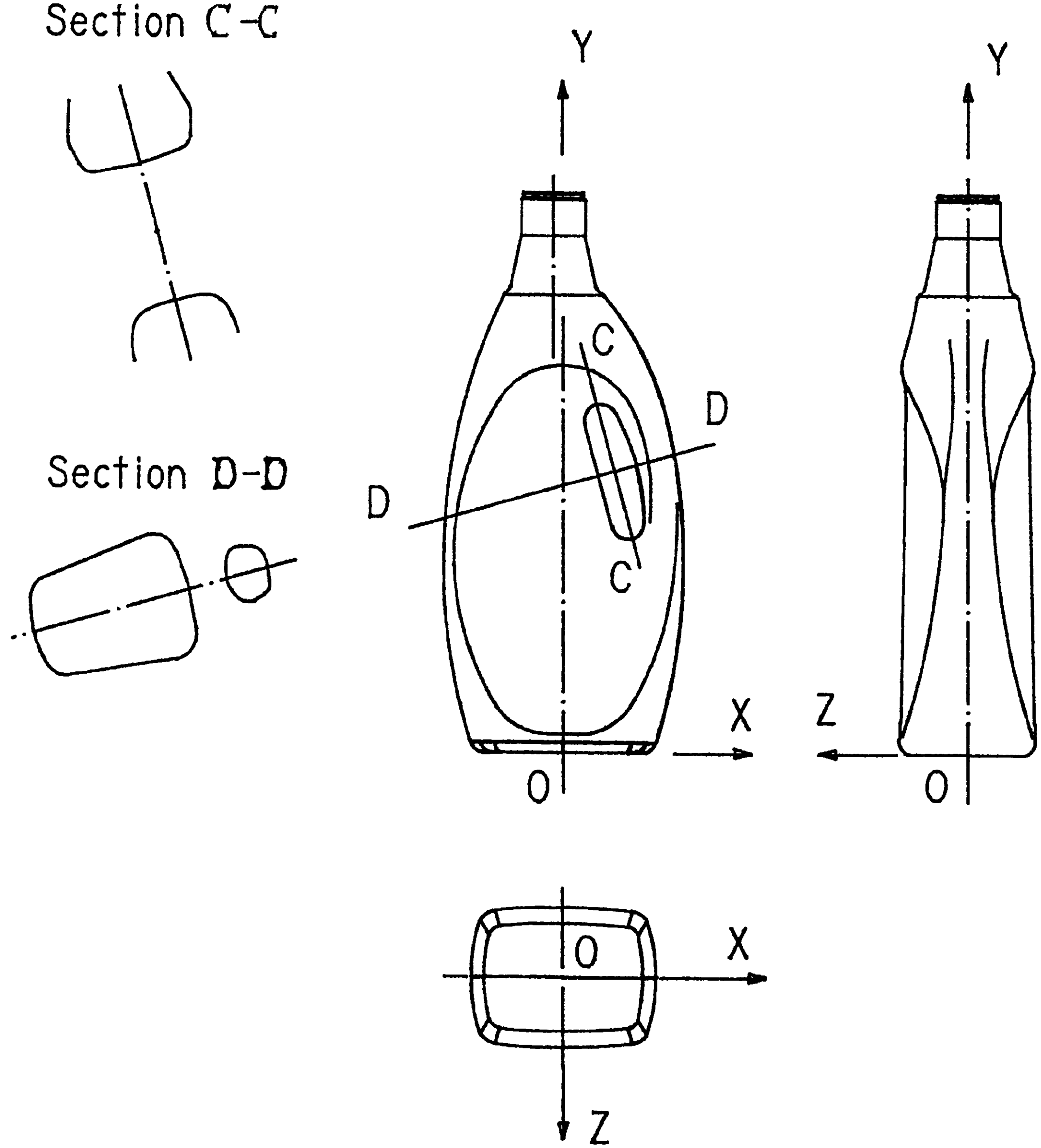


Figure 5.1 The geometric elements of the bottle in the 2D drawing

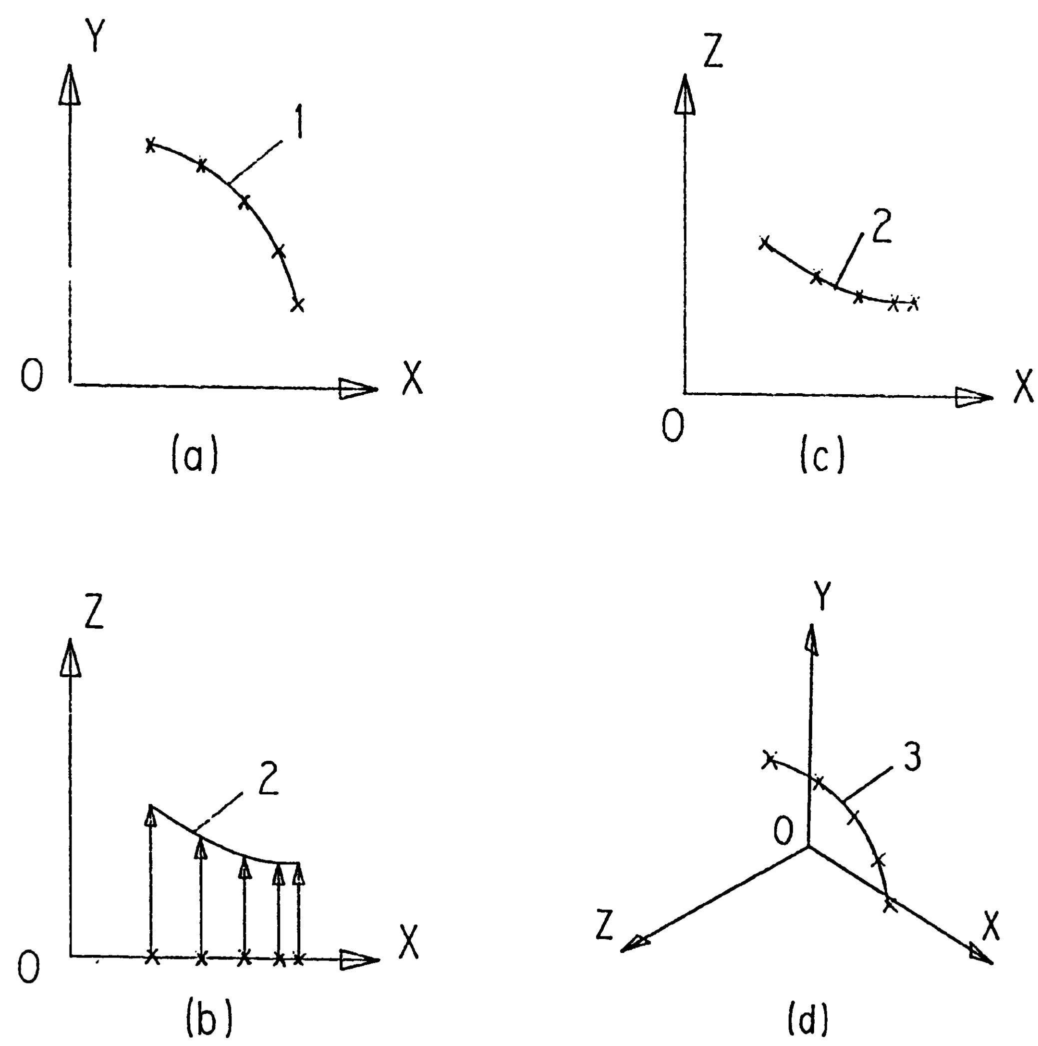


Figure 5.2 The principle of interpreting 2D informatiom into 3D information

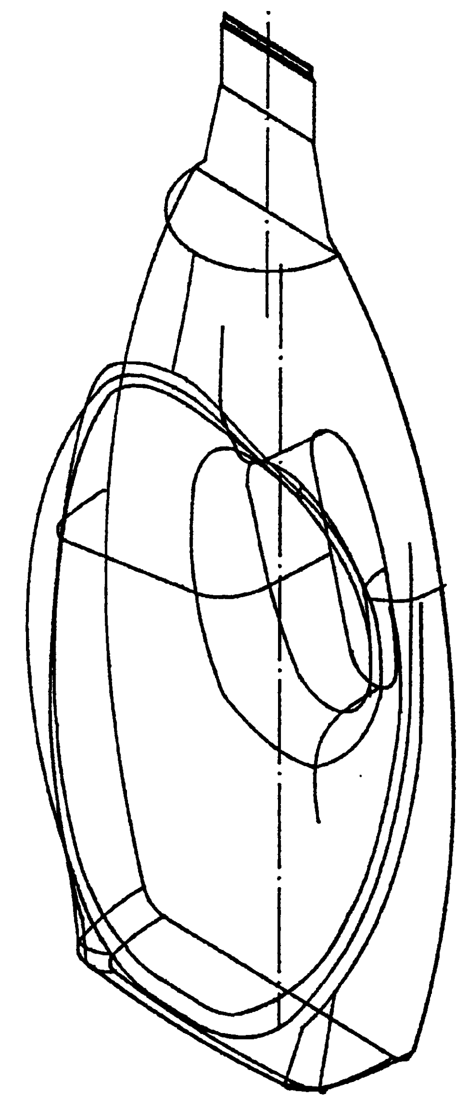


Figure 5.3 The 3D elements of the bottle (1/2) transformed from the 2D drawing

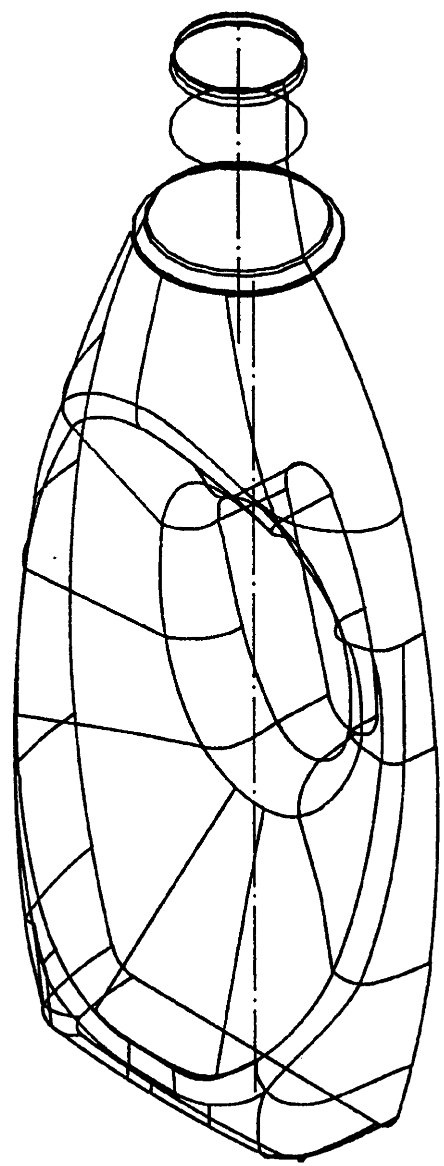


Figure 5.4 The wireframe model of the bottle (1/2)

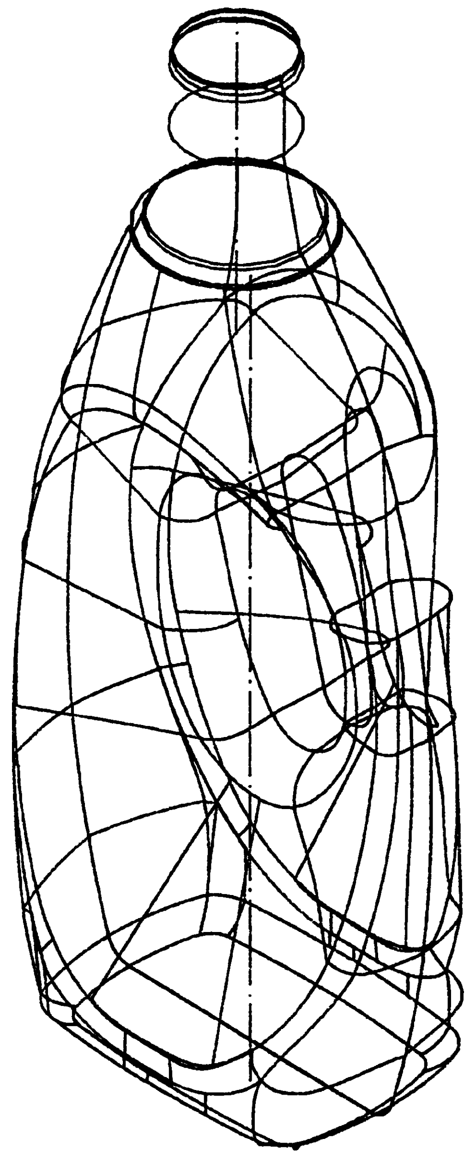


Figure 5.5 The refined surface model of the bottle

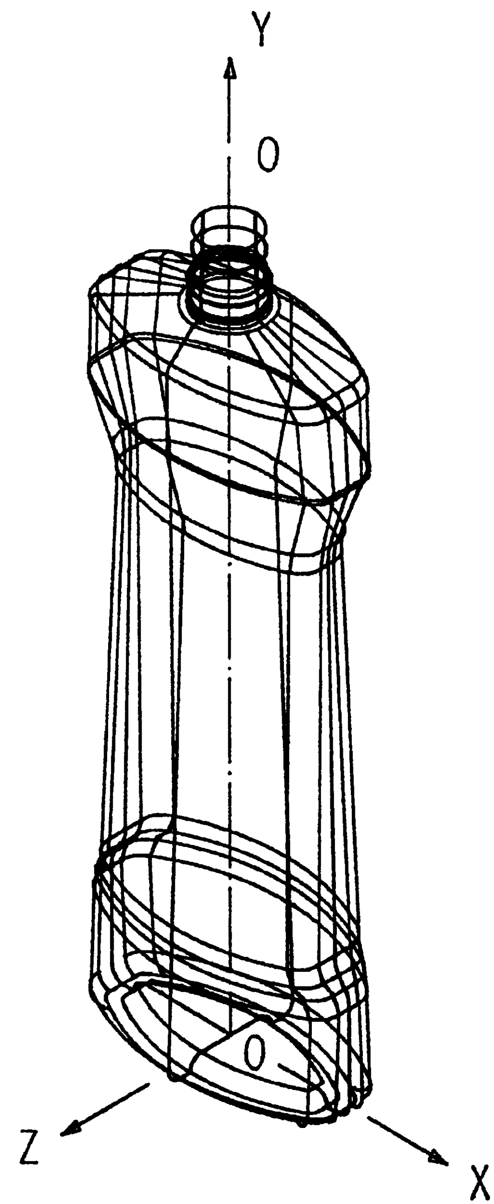


Figure 5.6 The surface model of a bottle designed by cross sections

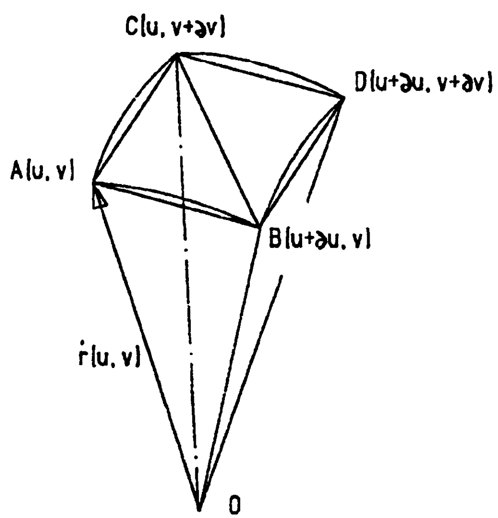


Figure 5.7

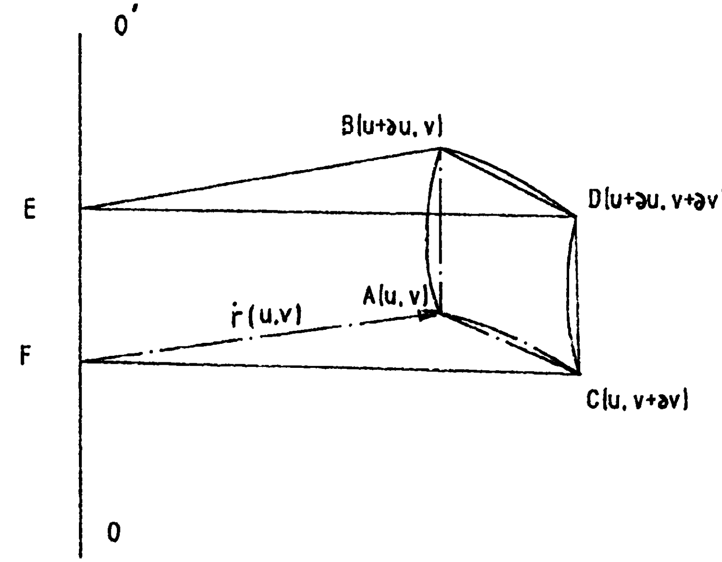


Figure 5.8

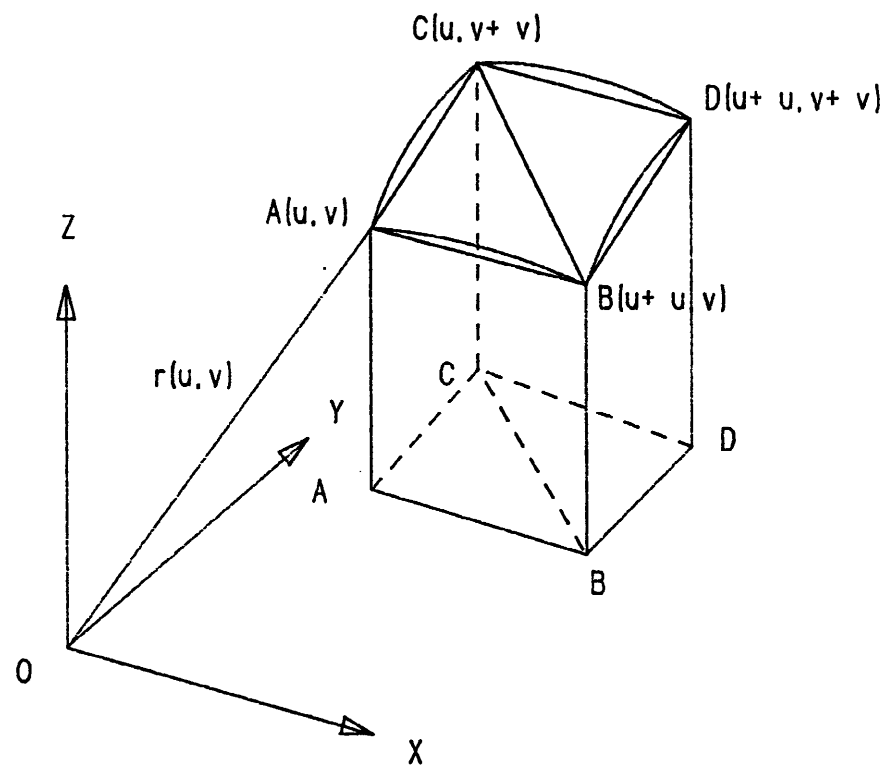


Figure 5.9

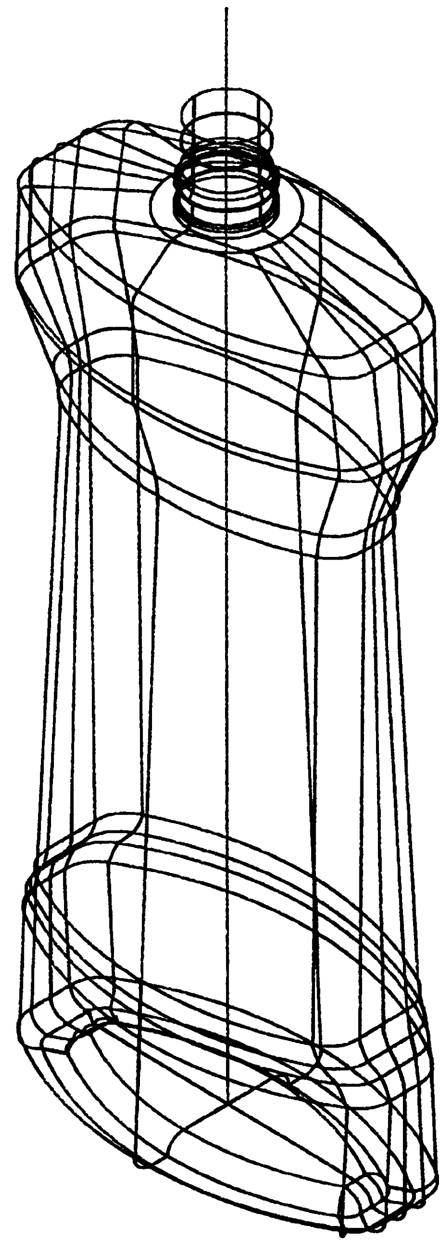


Figure 5.10 The scaled bottle which volume is increased by 50%

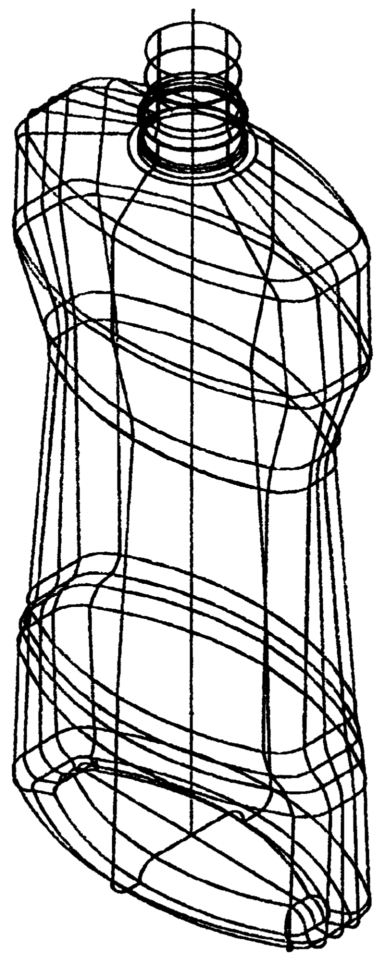


Figure 5.11 The locally scaled bottle which volume is decreased by 22.8%

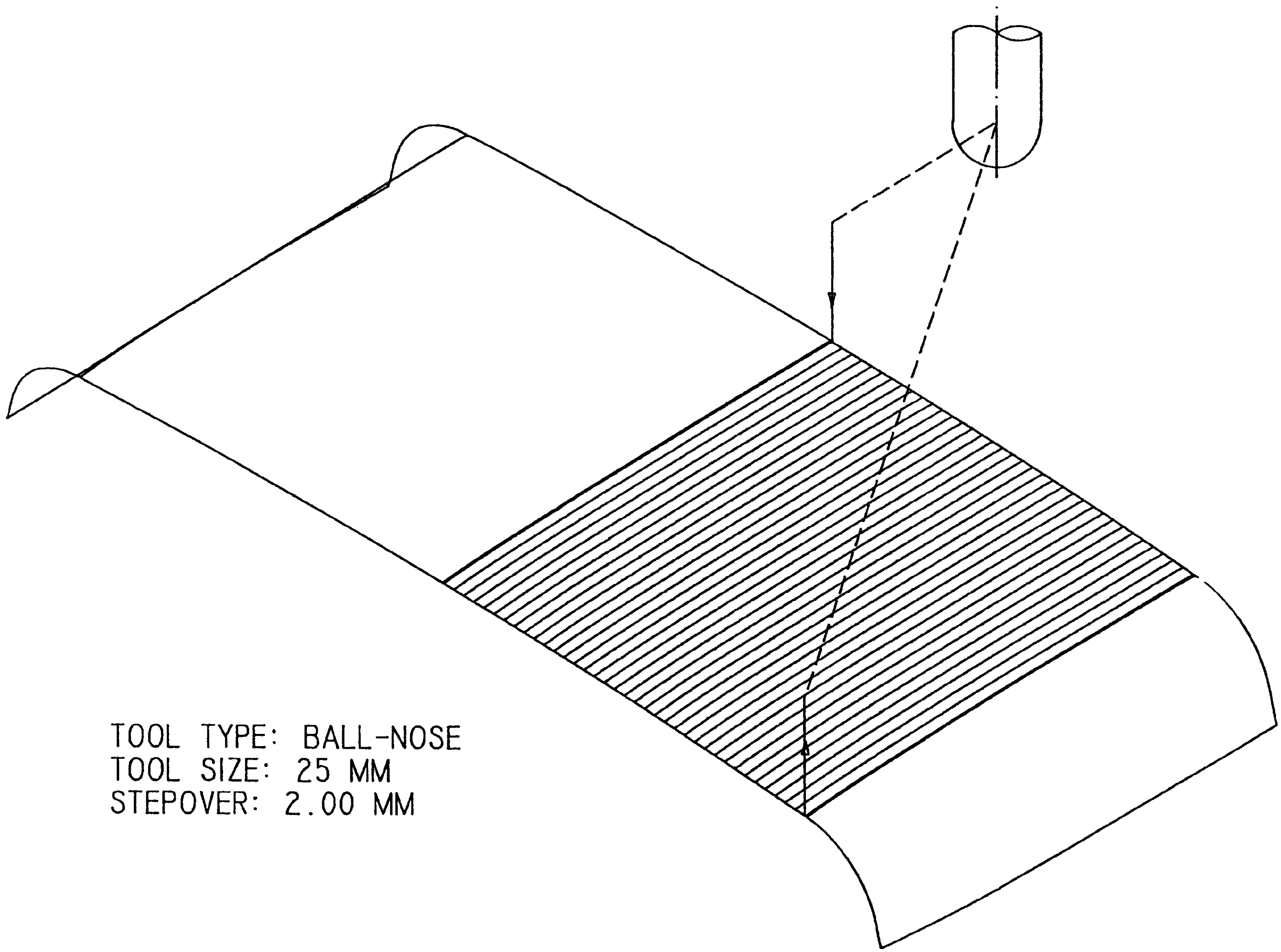


Figure 6.1 The toolpaths used for the 3-axis machining of a commercial vehicle cab roof model

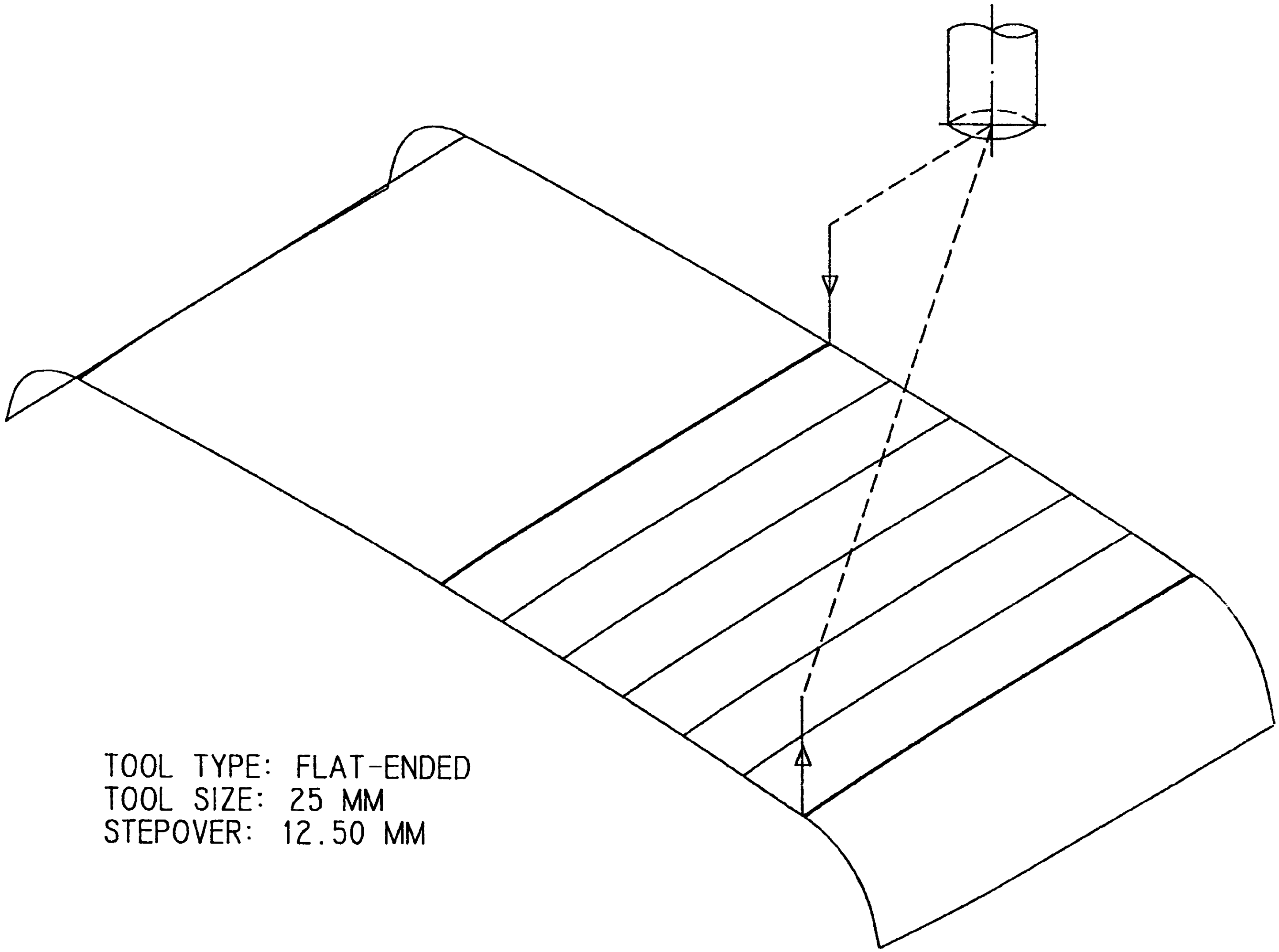


Figure 6.2 The toolpaths used for the 5-axis machining of a commercial vehicle cab roof model

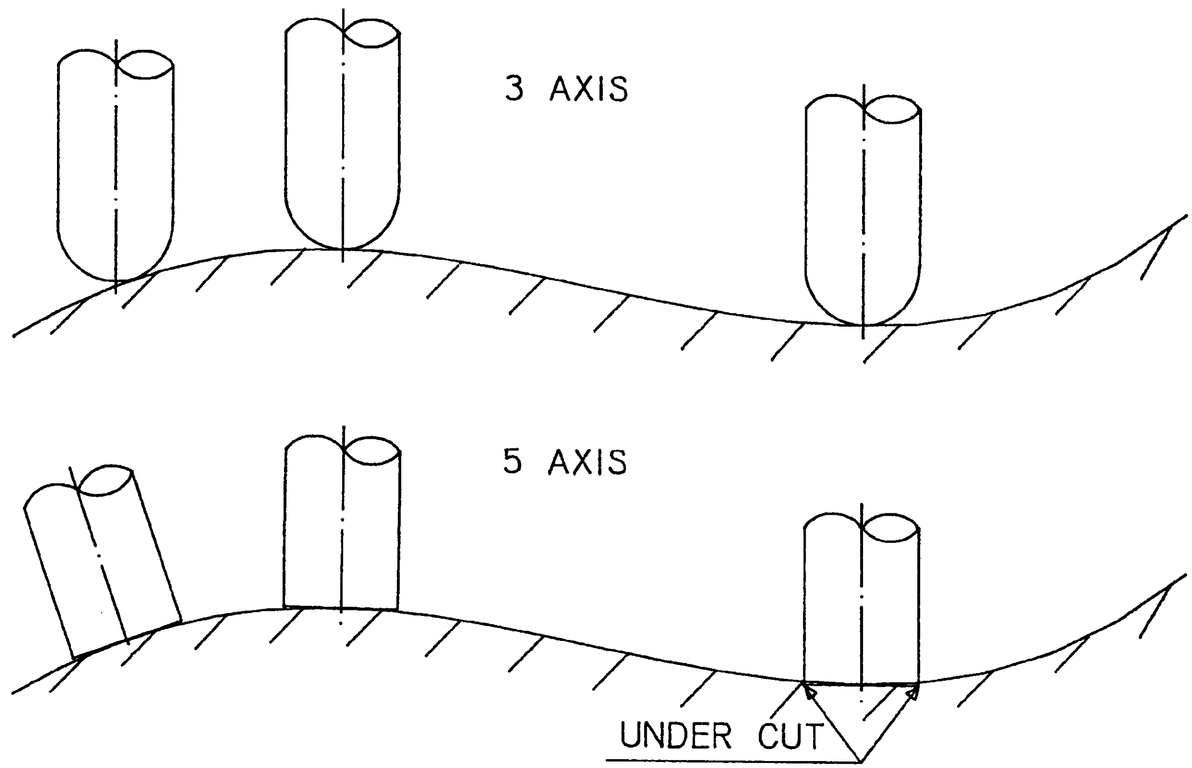


Figure 6.3 A comparison of 3- and 5-axis machining highlighting the foul condition when using non-canted 5-axis machining on concave surfacec

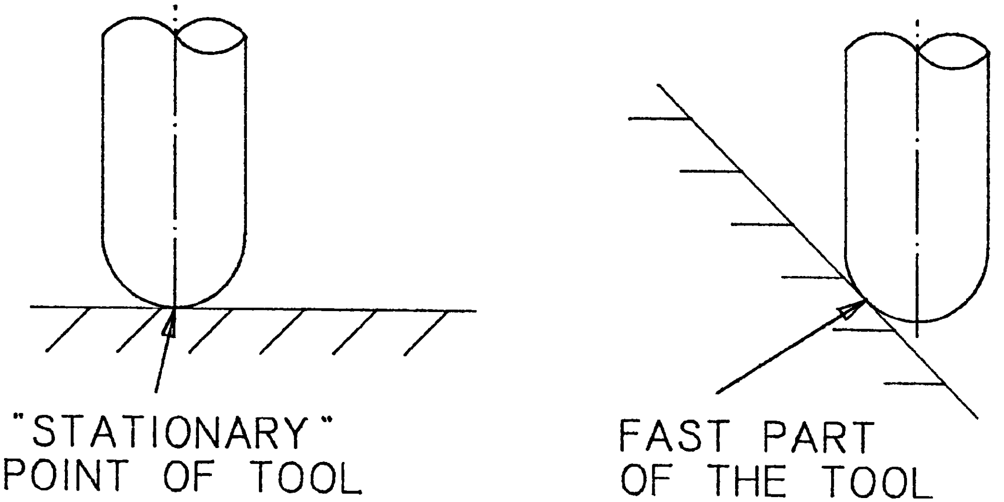


Figure 6.4 An indication of the variable cutting conditions experienced by a ball-ended tool when used for 3-axis machining

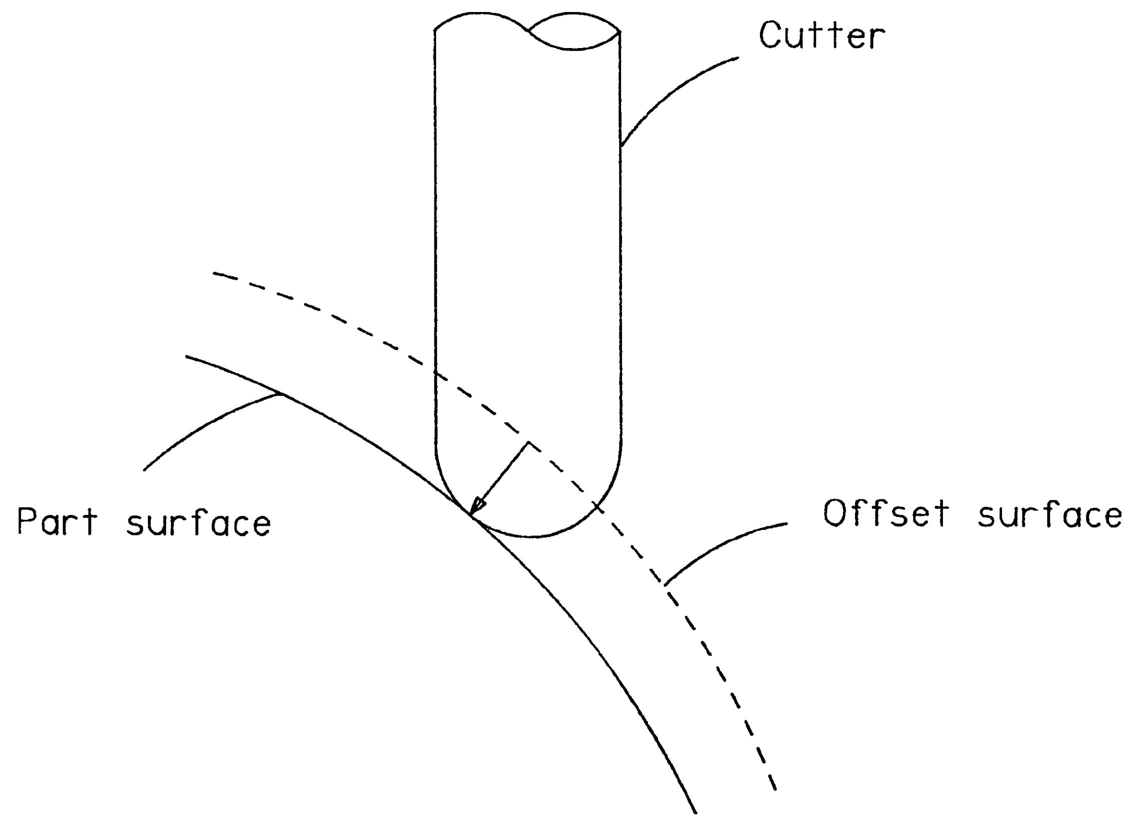


Figure 6.5 A diagram to indicate the relationship of the offset surface to the part surface with respect to the motion of the milling cutter

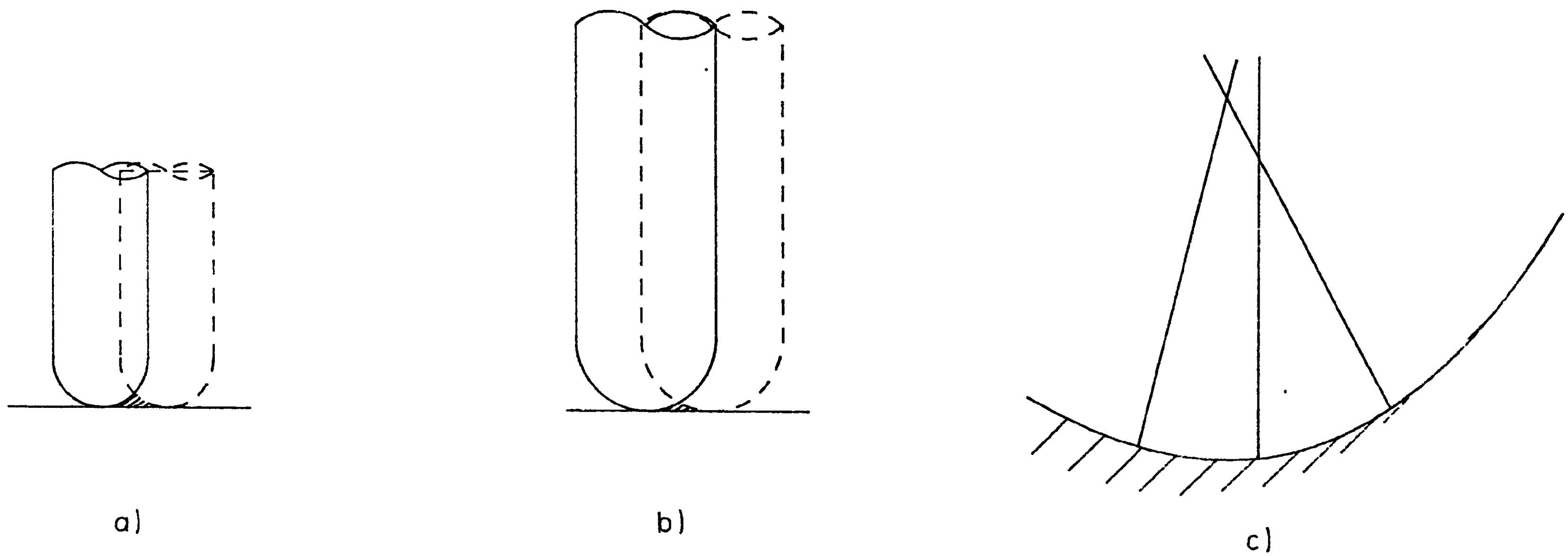


Figure 6.6 Tool size determination

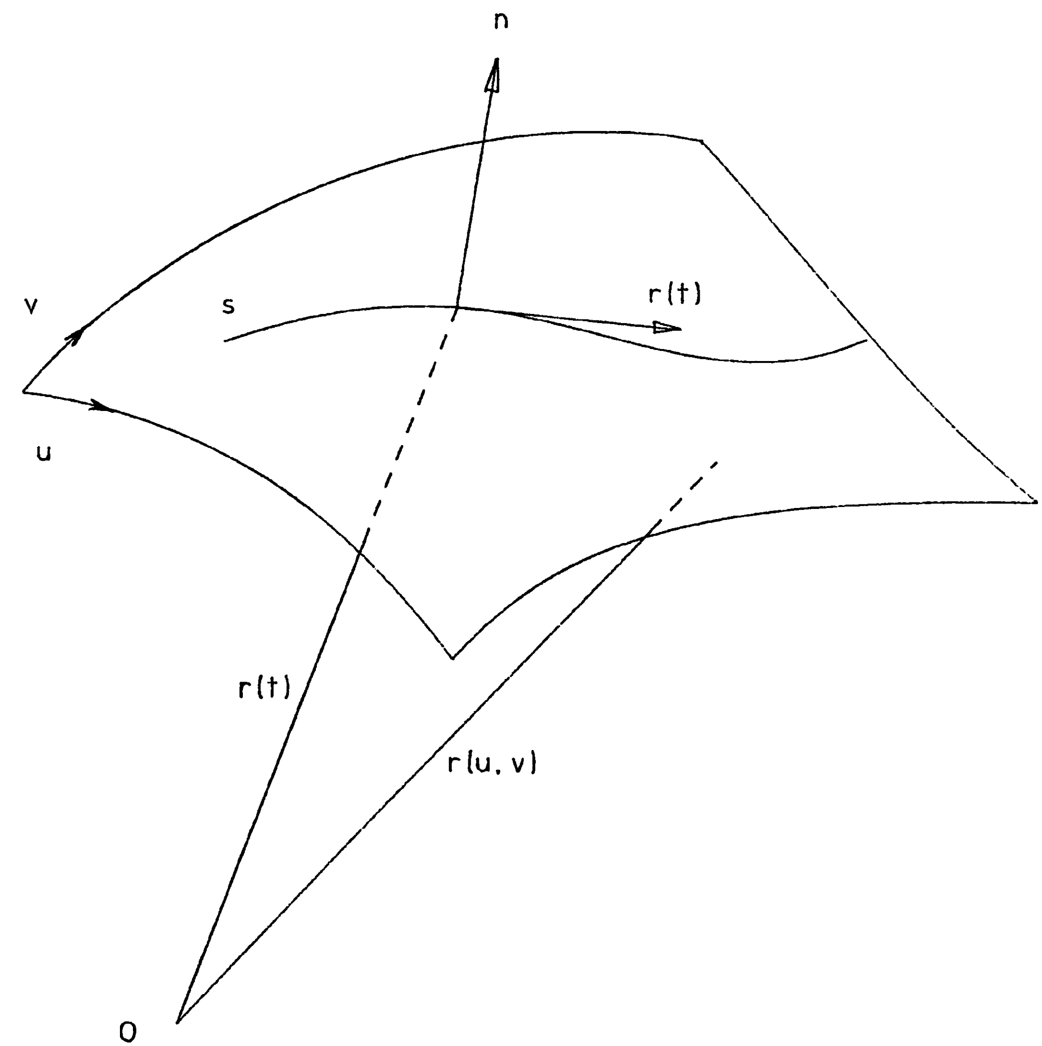


Figure 6.7 A arbitrary curve on a surface

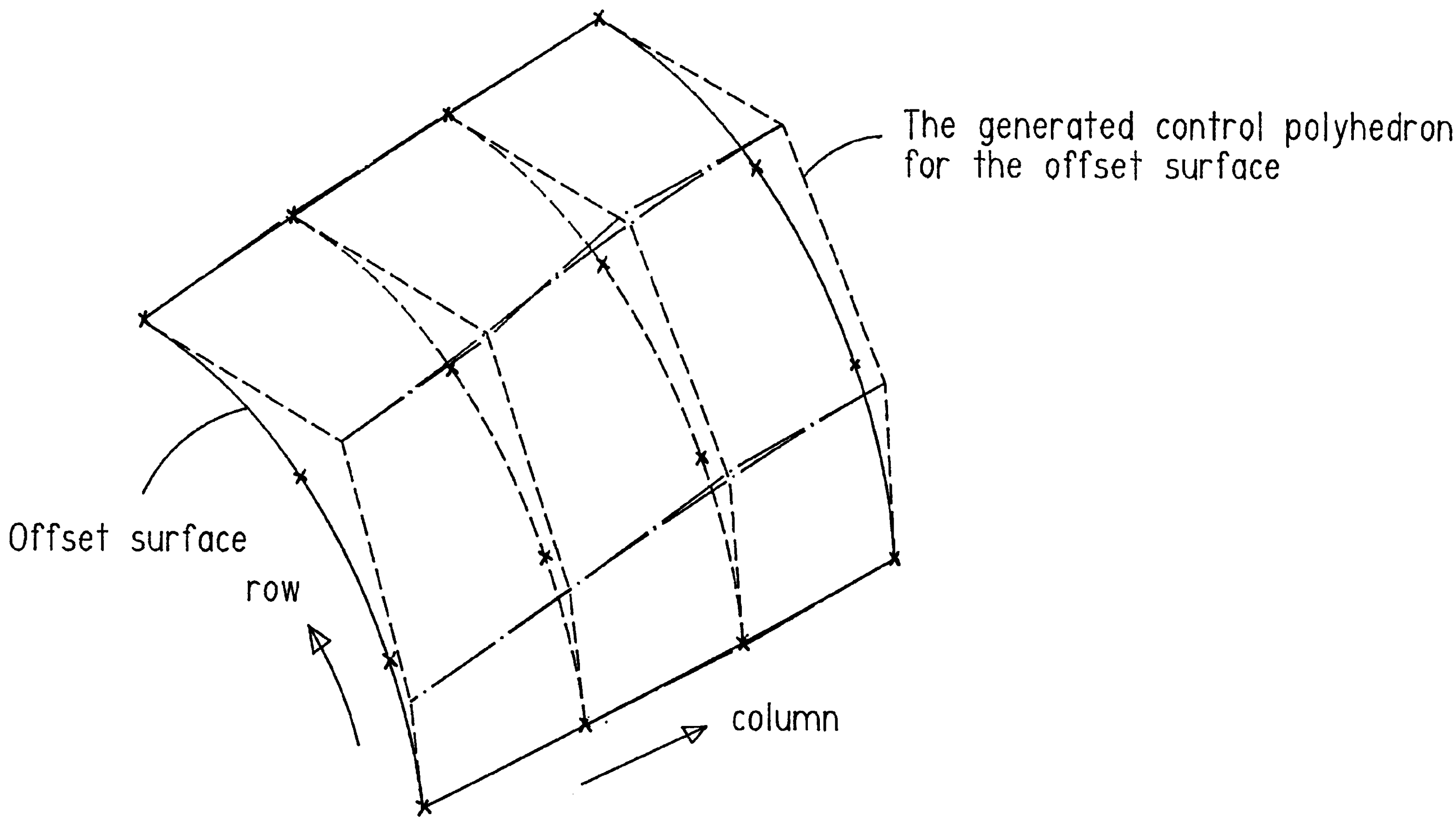


Figure 6.8 The resulting offset surface created from the sixteen offset points

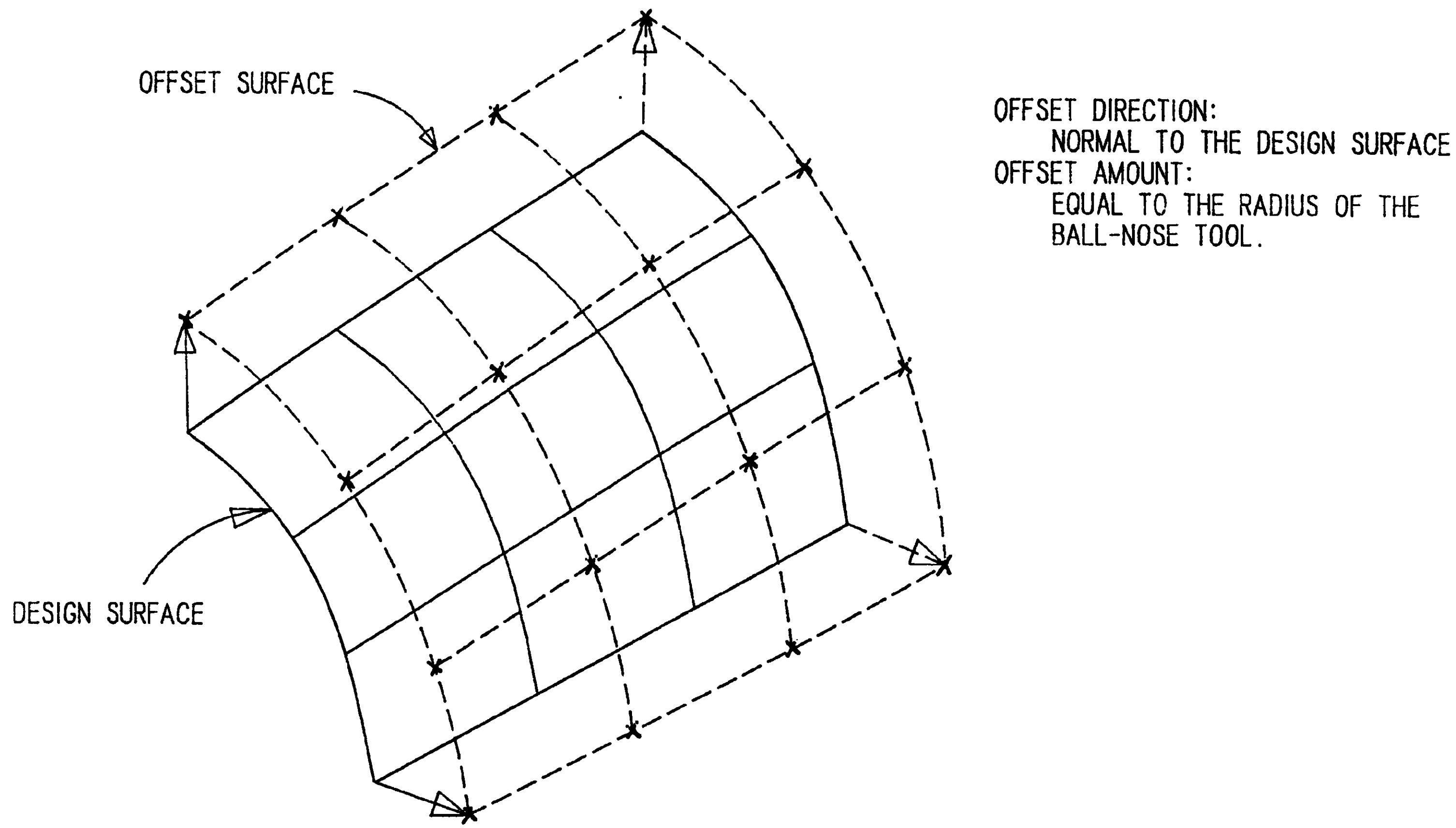


Figure 6.9 A part surface shown with the generated offset points and the offset surface fitting these points

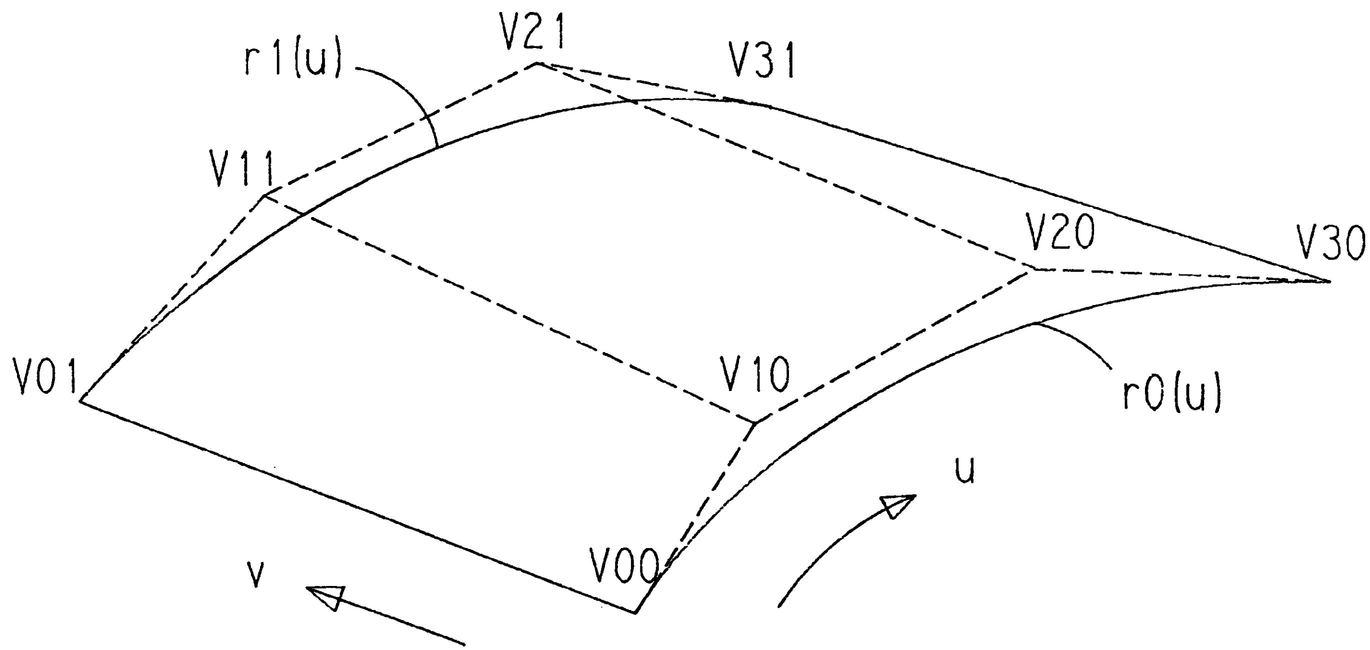
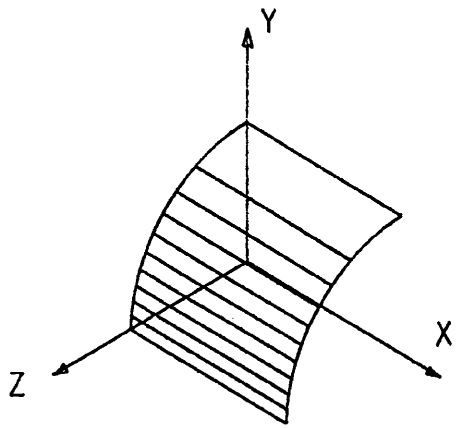
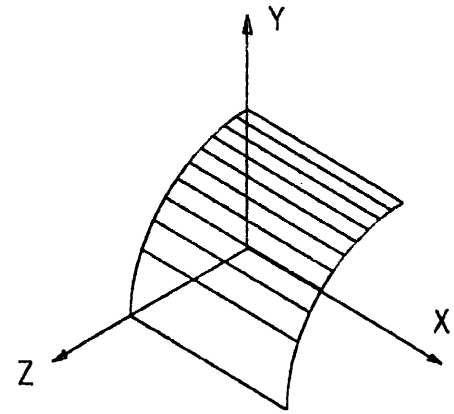


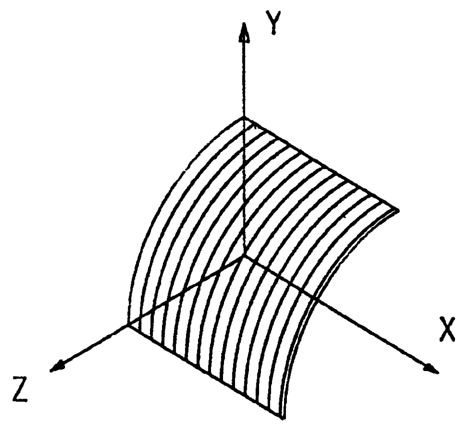
Figure 6.10 The defination of a ruled surface



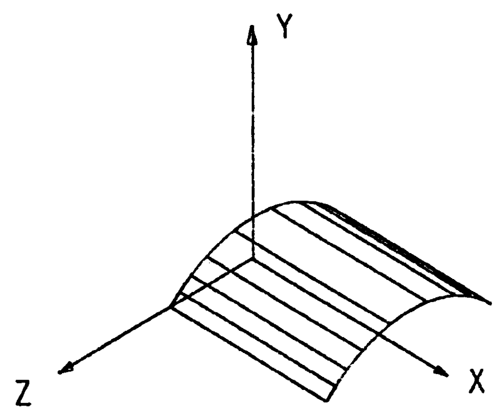
(a) Cutter parallel to Z axis (cutting along surface lines with constant Y increments).



(b) Cutter parallel to Y axis.



(c) Cutter parallel to Y or Z axis.



(d) Cutter parallel to Y axis. A wrong direction is selected to machine a flat surface. Direction should be as (c).

Figure 6.11 Selecting correct machining directions

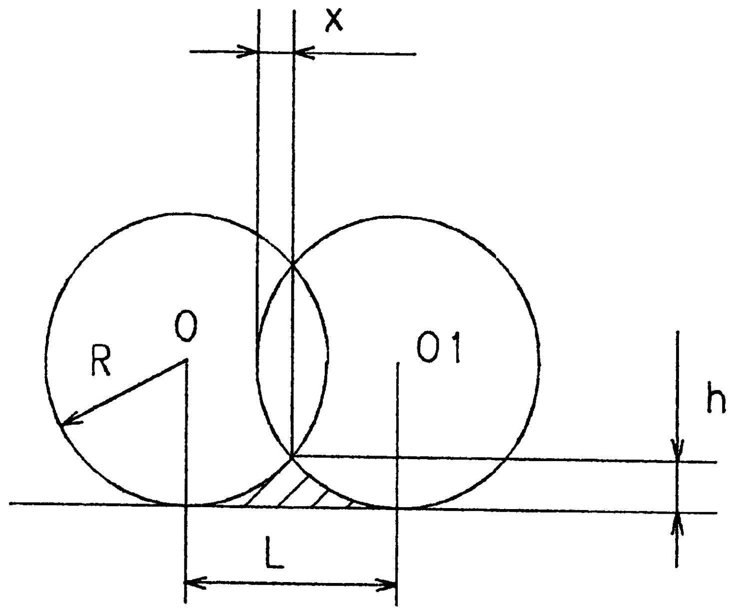


Figure 6.12 The calculation of cusp height for machining flat surfaces with ball-ended tools

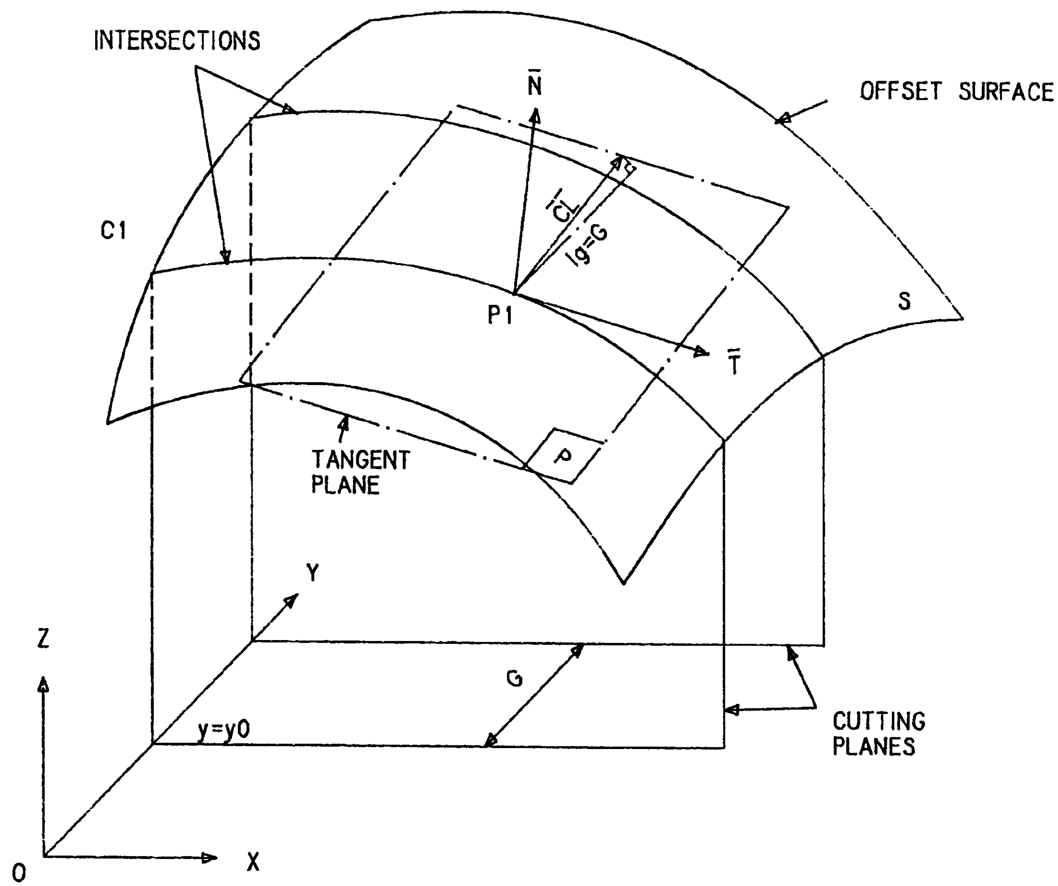


Figure 6.13 A diagram to illustrate the principles by which the cusp height may be determined

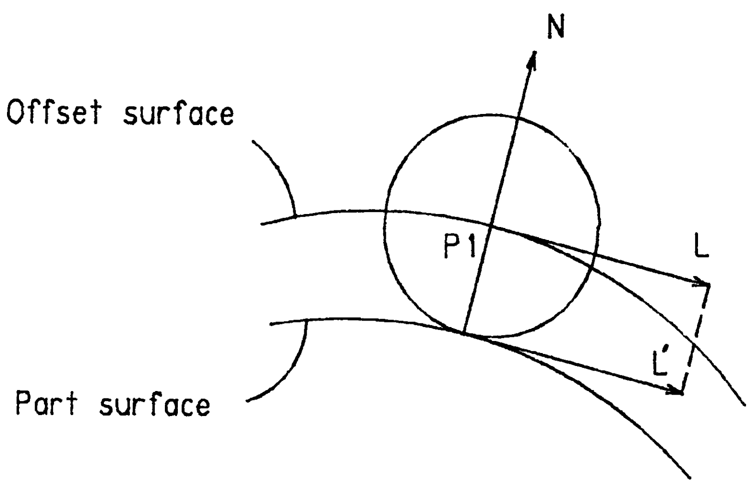


Figure 6.14

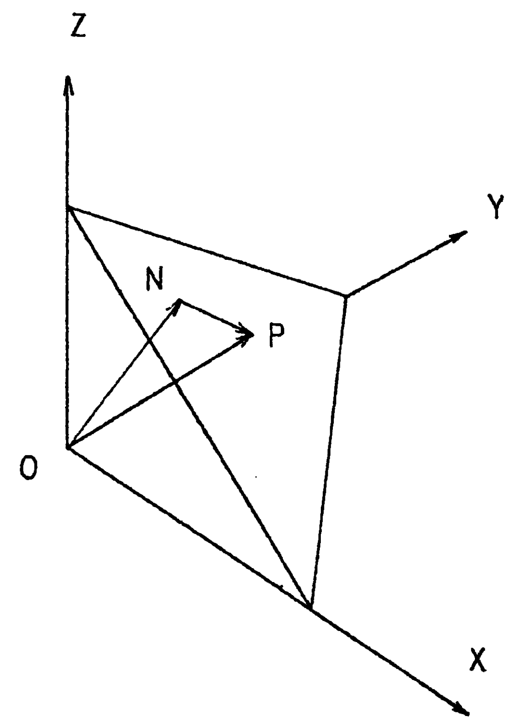


Figure 6.15 The definition of a plane

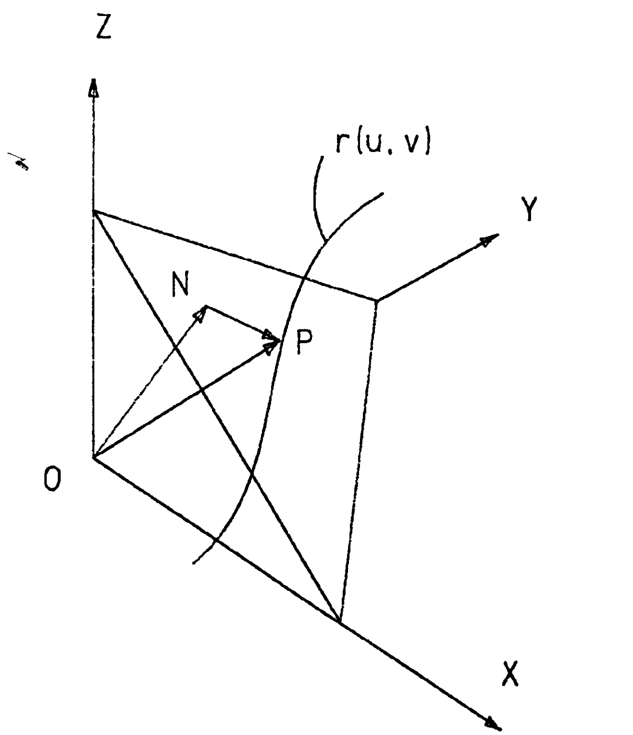


Figure 6.16 The intersection between a plane and a curve

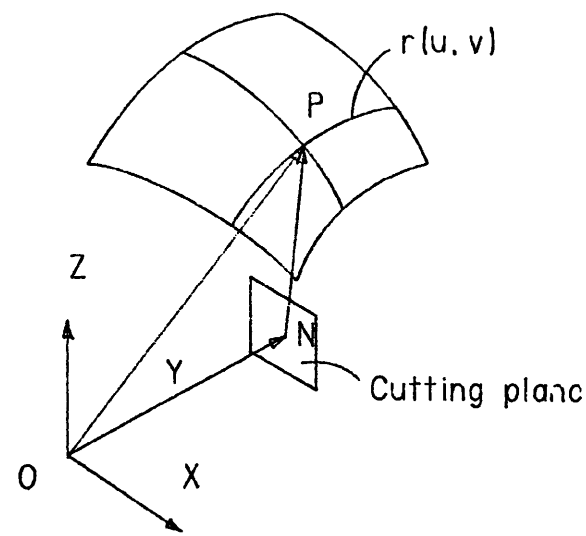


Figure 6.17 The intersection between a plane and a surface

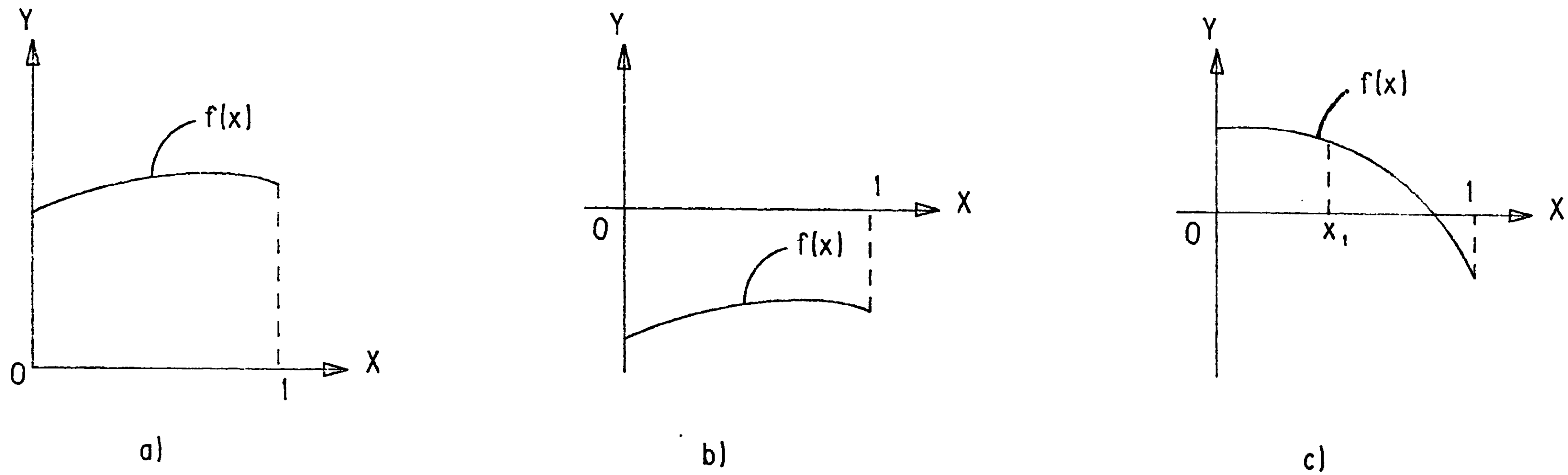


Figure 6.18 The bracketing method of solving polynomial equations

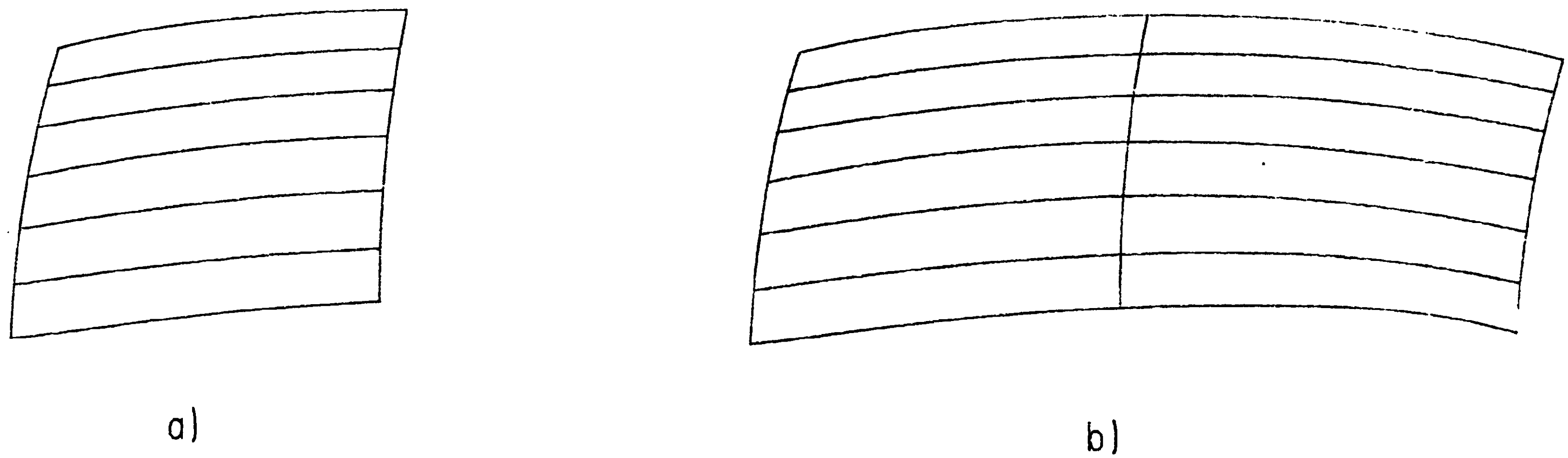


Figure 6.19 Two surface patches can be machined either patch by patch or together

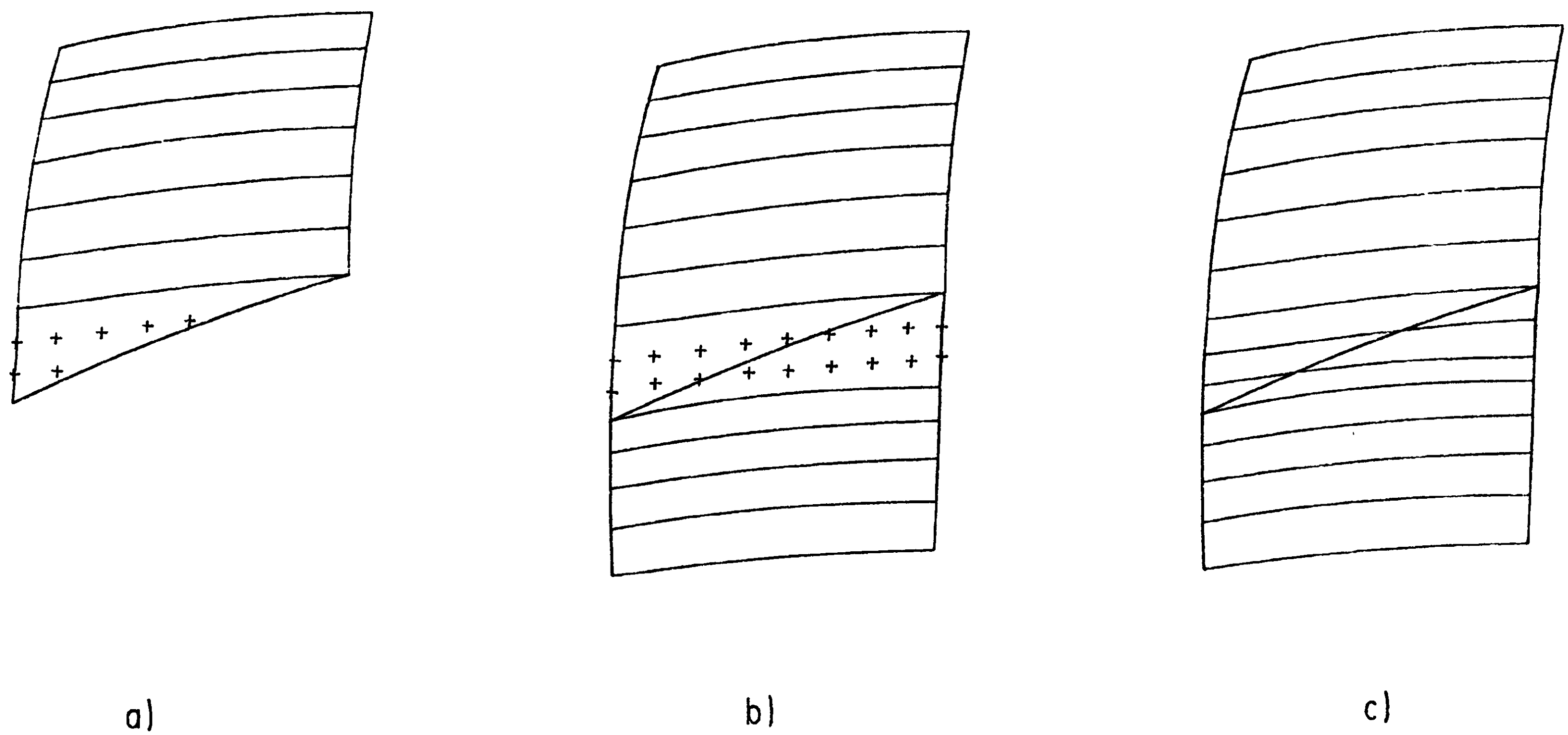


Figure 6.20 Some toolpath curves are required to be created manually

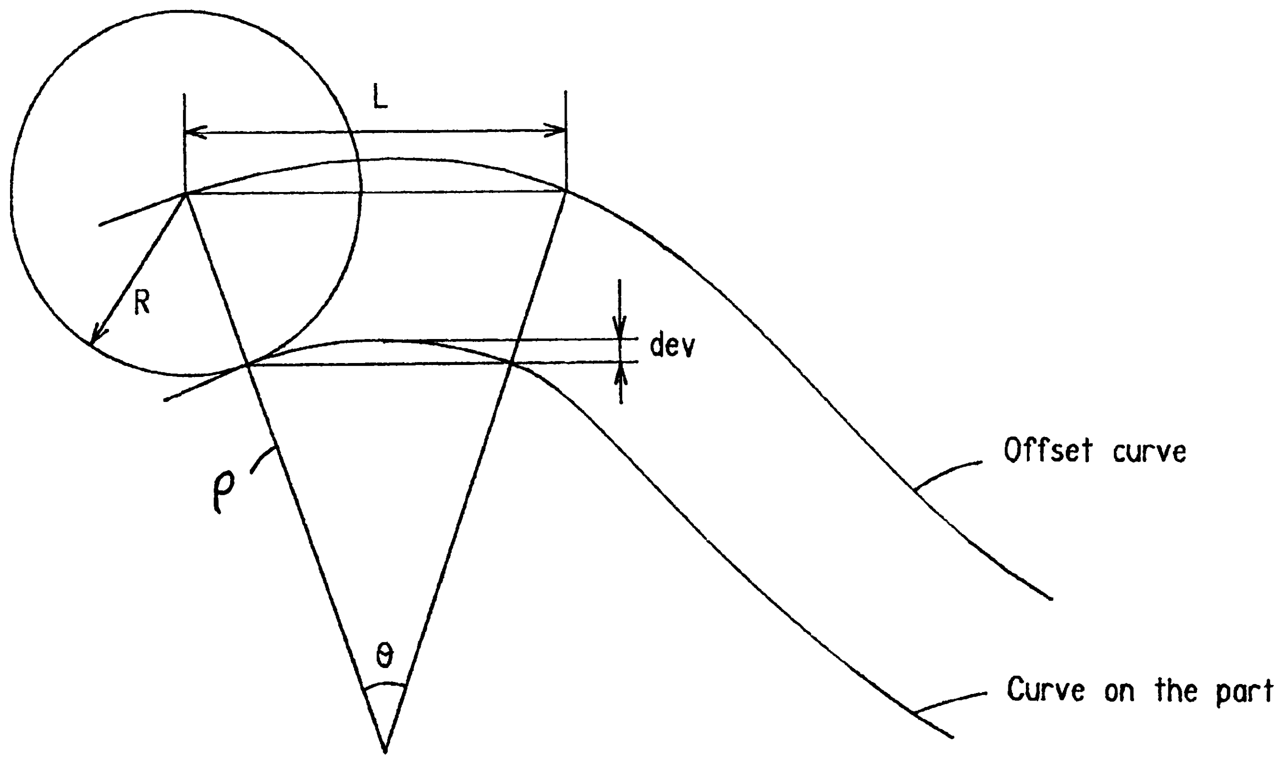


Figure 6.21 The calculation of the chordal deviation

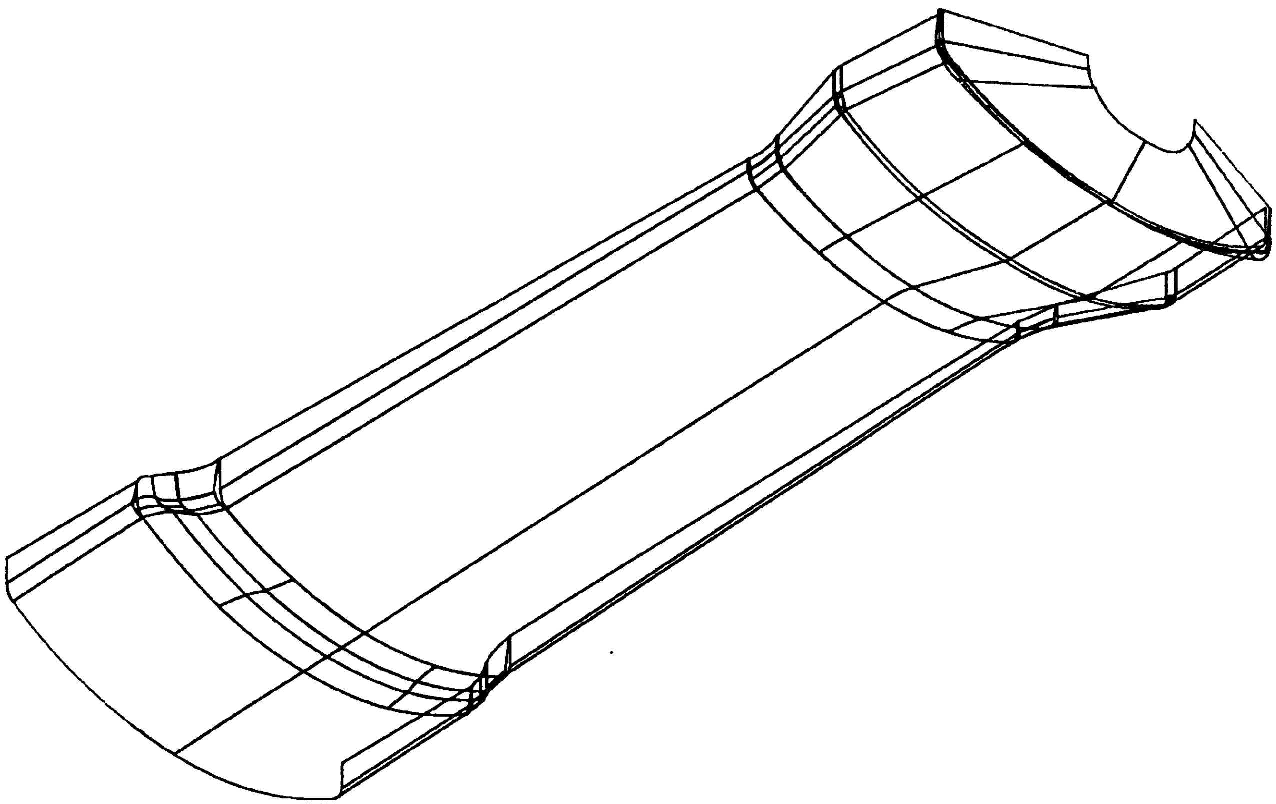


Figure 6.22 The offset surfaces necessary for the manufacture of one-half of the mould required to produce the bottle shown in Figure 5.6

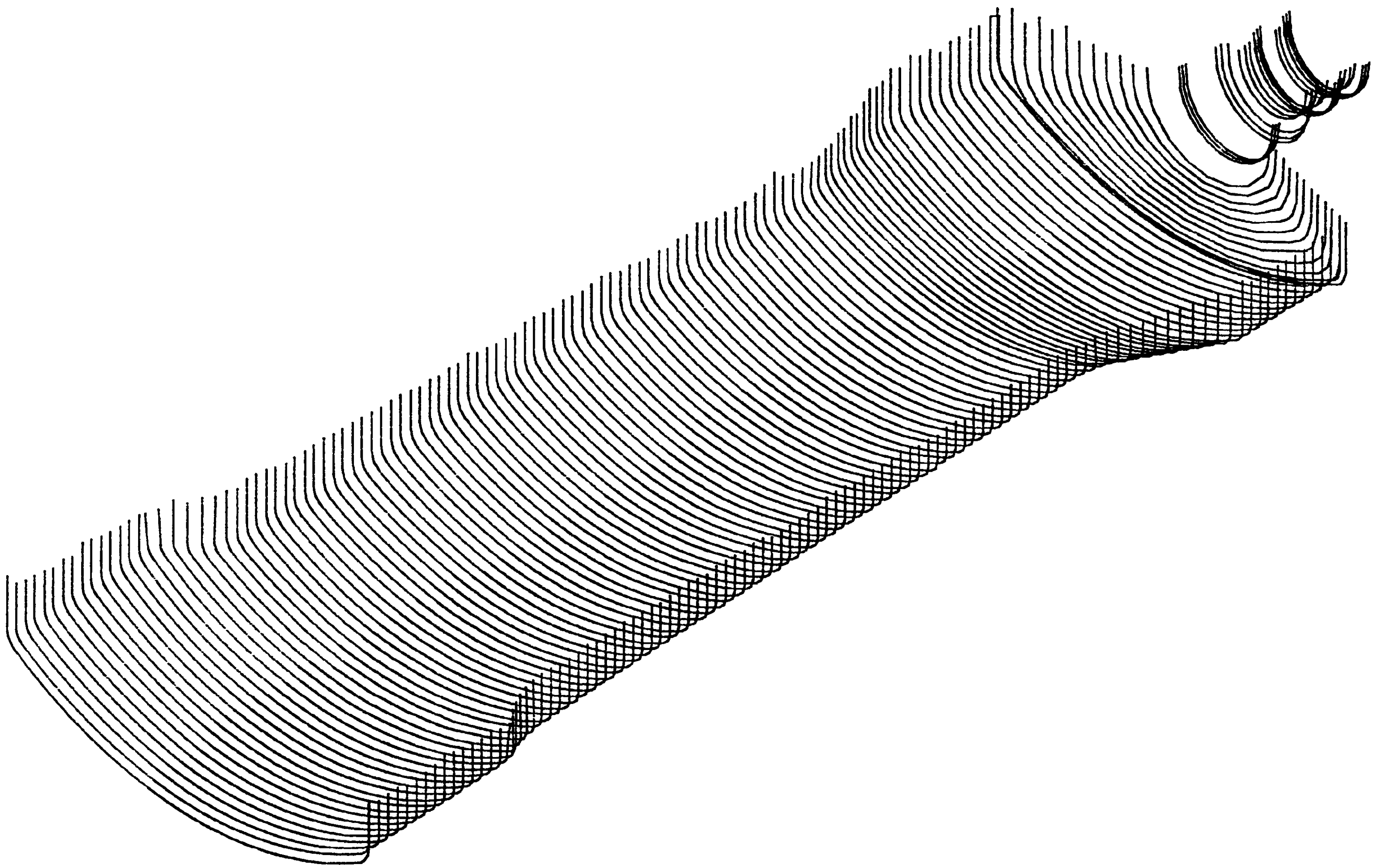


Figure 6.23 The toolpath geometry used for the machining of the bottle mould

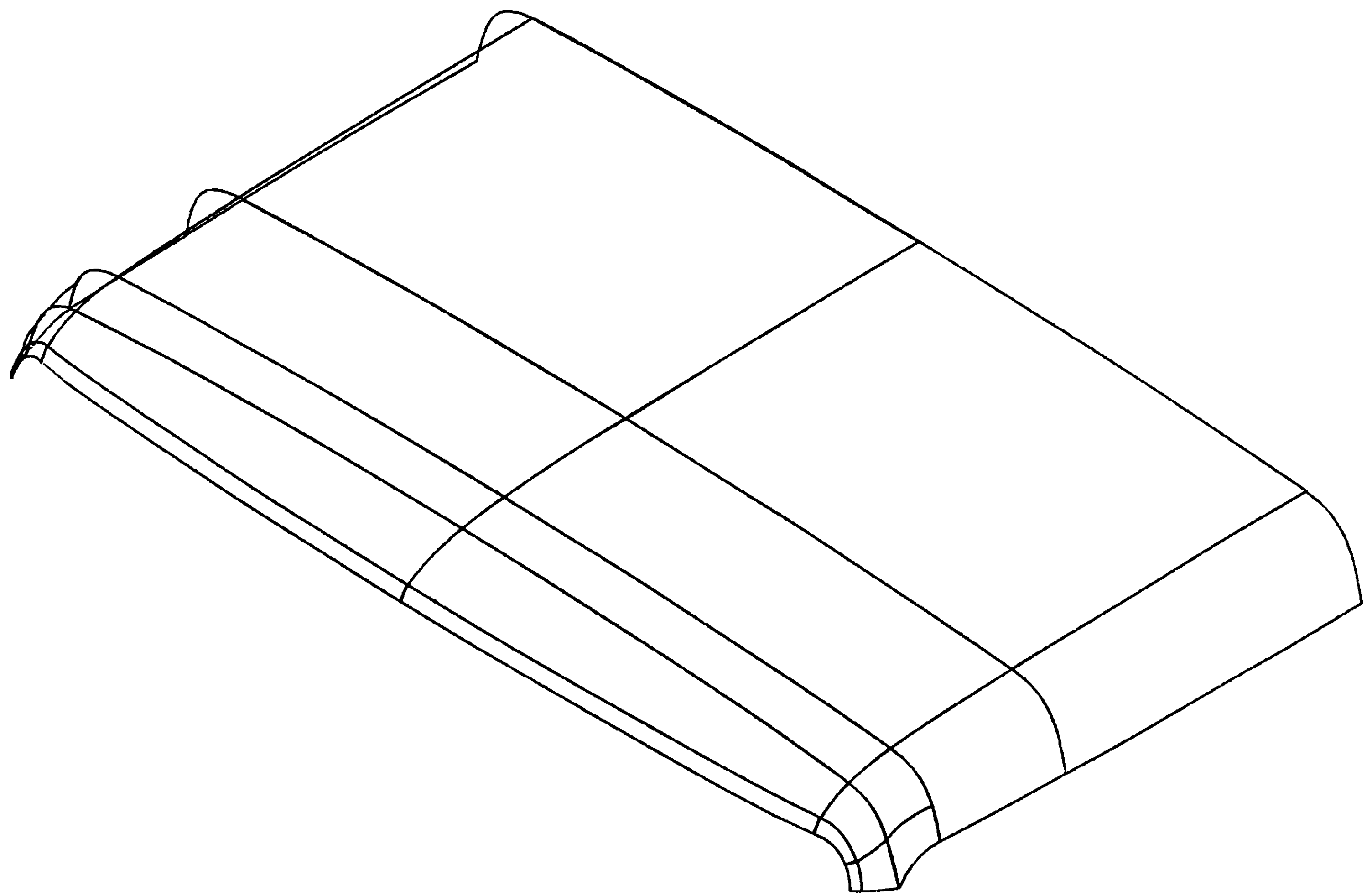


Figure 6.24 The scaled cab roof model

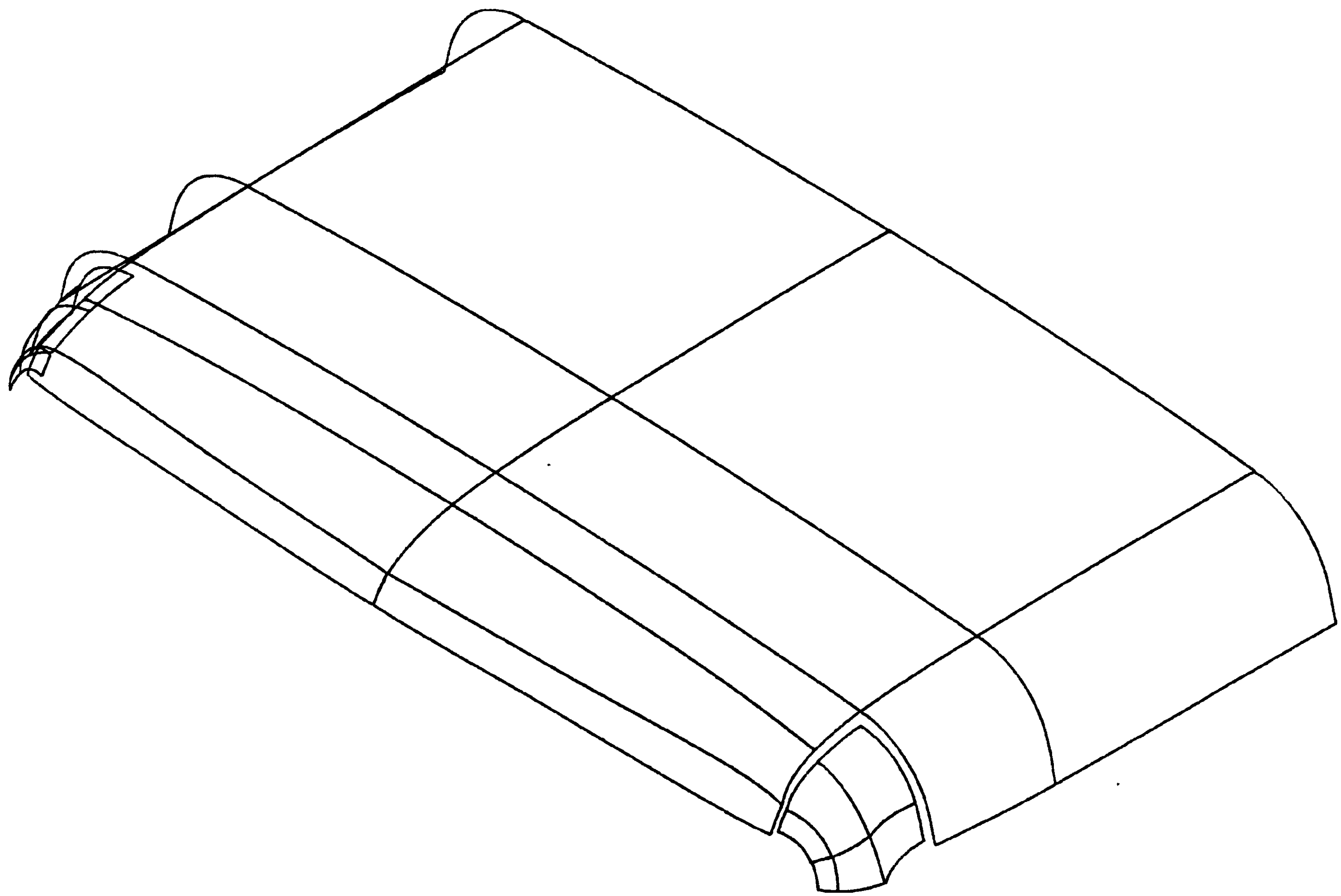


Figure 6.25 The offset surfaces to the part surfaces of the cab roof model

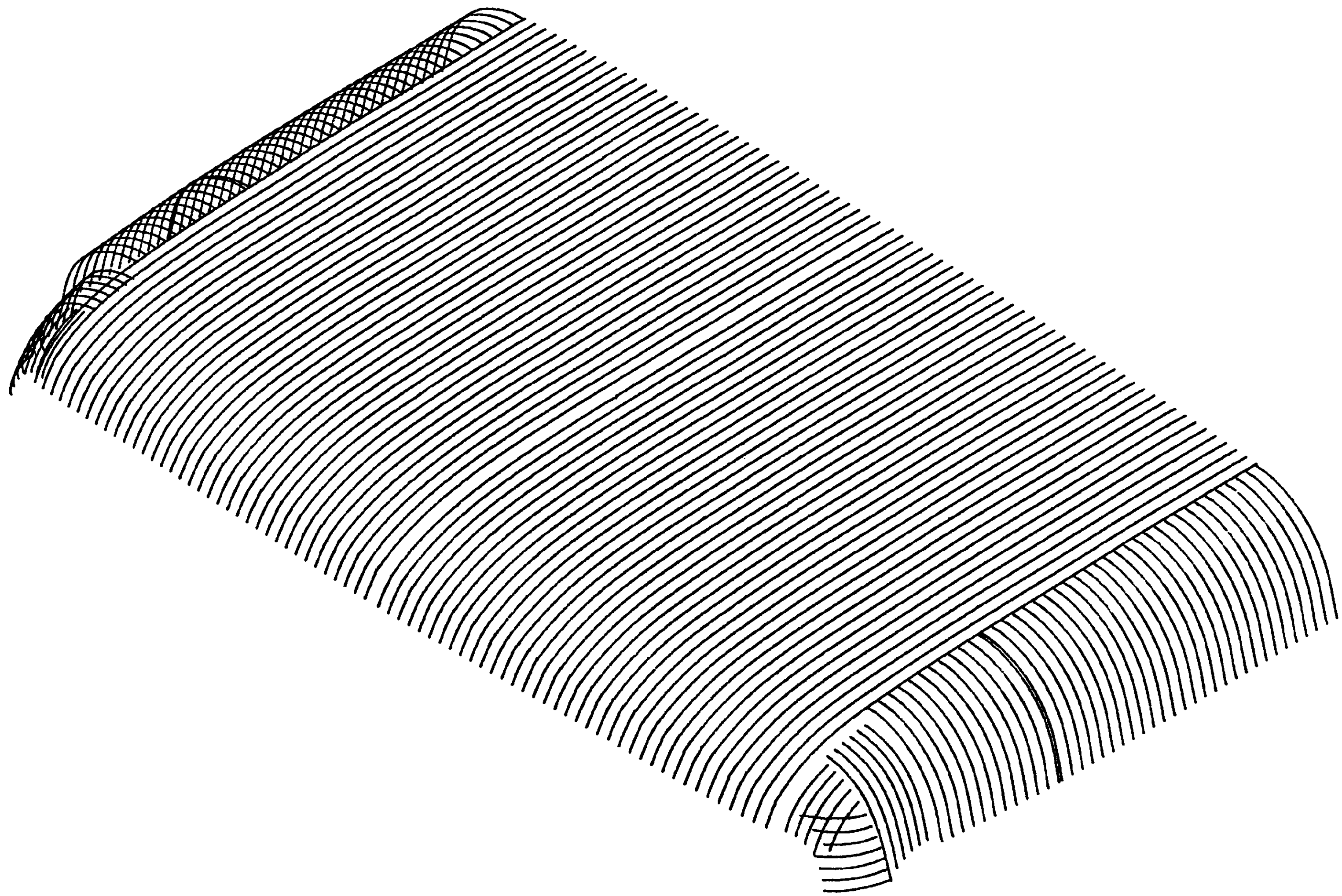


Figure 6.26 The toolpath geometry used for the machining of the roof model

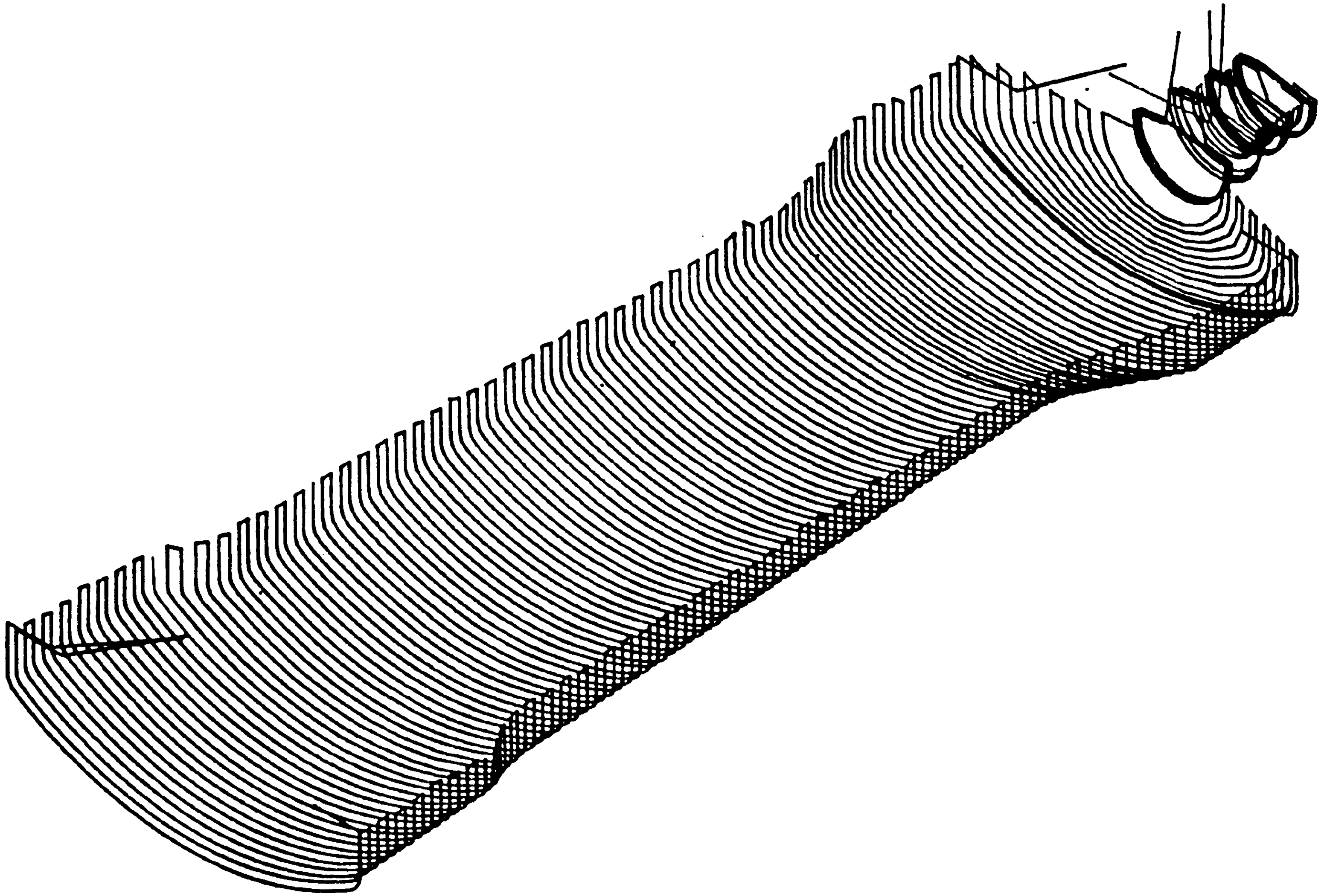


Figure 7.1 The toolpaths for the finish machining of the bottle mould

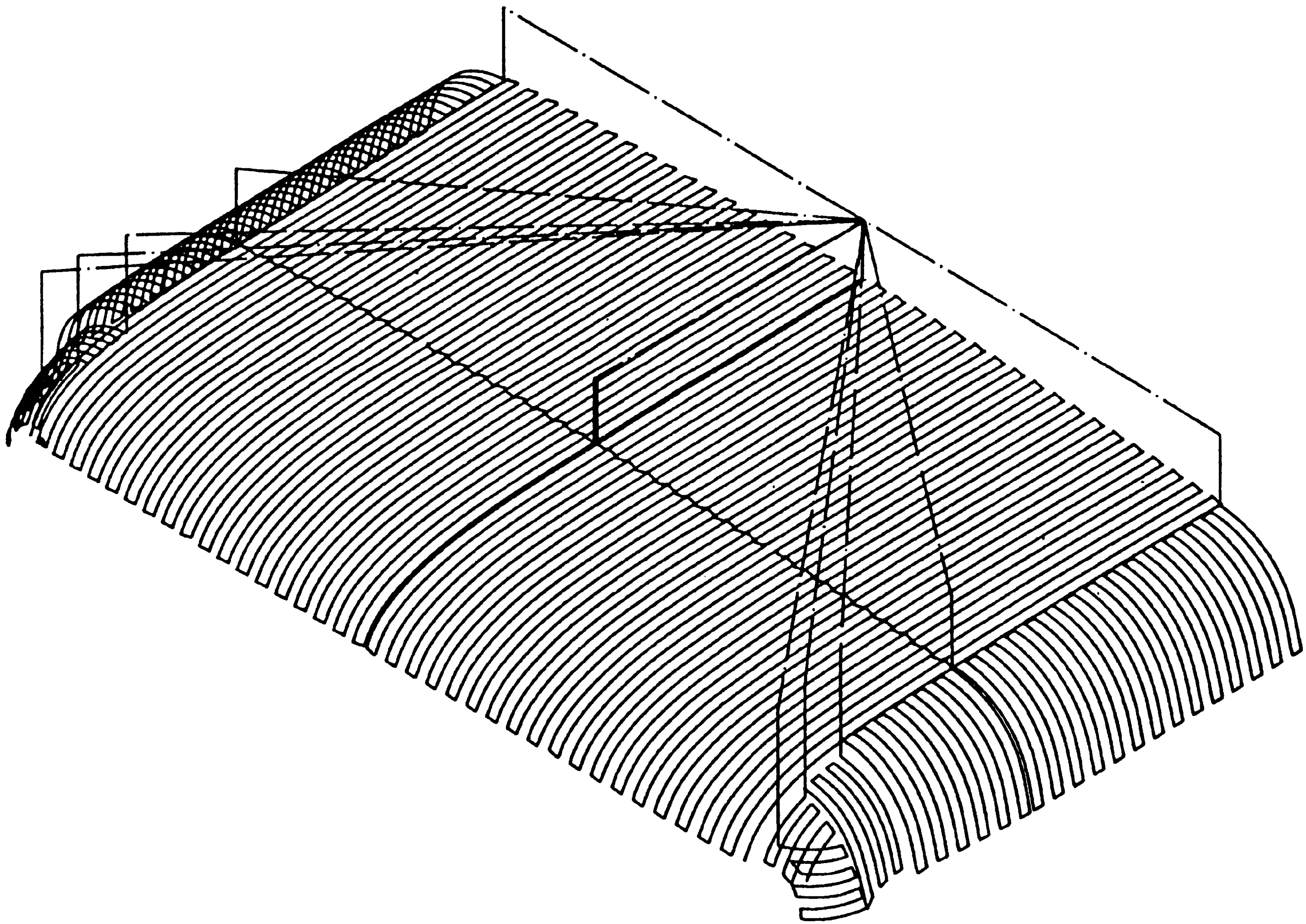


Figure 7.2 The toolpaths for the finish machining of the cab roof model

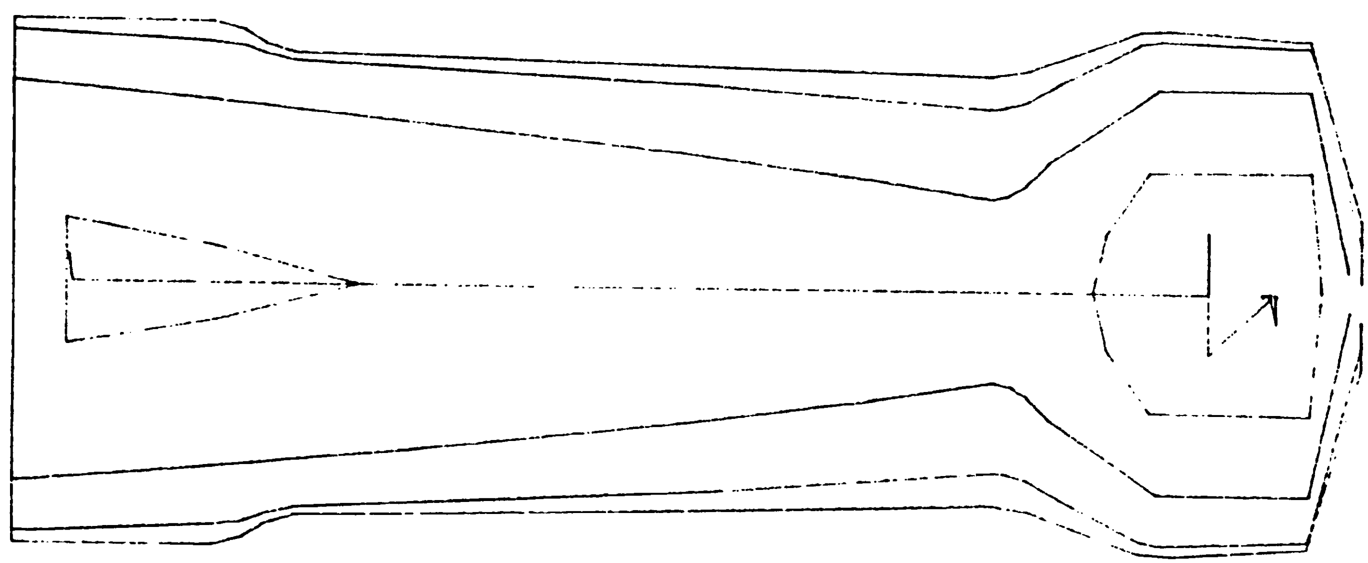


Figure 7.3 The toolpaths for the roughing of the bottle mould

Appendix Two

TABLES

TABLE 4.1: The digitized data from the top half of a telephone receiver on Takisawa Machining Centre.

The machine datum: X-318.3 Y-123.155 Z-202.553
The probe diameter: 6.00mm

Y	(X,Z)				
0.	(0.,0.)				
-1.	(-9.7,0.4)	(-5.,1.9)	(0.,2.6)	(5.,1.6)	(7.7,-0.4)
-2.	(-13.1,-0.7)	(-10.,2.5)	(-5.,4.0)	(0.,4.2)	(5.,4.2)
	(10.,0.6)	(11.2,-1.1)			
-4.	(-17.7,-1.8)	(-15.,2.9)	(-10.,6.8)	(-5.,8.5)	(0.,6.8)
	(5.,8.)	(10.,5.4)	(15.,-0.3)	(15.5,-1.5)	
-6.	(-20.8,-2.)	(-20.,0.2)	(-15.,6.8)	(-10.,10.)	(-5.,11.5)
	(0.,11.7)	(5.,11.)	(10.,8.8)	(15.,4.5)	(18.7,-1.7)
-9.	(24.5,-2.4)	(-20.,6.2)	(-15.,10.9)	(-10.,13.5)	(-5.,14.7)
	(0.,14.9)	(5.,14.3)	(10.,12.6)	(15.,9.2)	(20.,3.1)
	(25.,-2.)				
-12.	(-27.3,-2.9)	(-25.,3.)	(-20.,10.)	(-15.,13.9)	(-10.,16.2)
	(-5.,17.3)	(0.,17.4)	(5.,16.9)	(10.,15.3)	(15.,12.5)
	(20.,7.6)	(25.,-2.8)			
	(-29.4,-3.4)	(-25.,7.1)	(-20.,12.7)	(-15.,16.2)	(-10.,18.)
-15.	(-5.,18.9)	(0.,19.)	(5.,18.6)	(10.,17.5)	(15.,14.9)
	(20.,10.6)	(25.,3.)	(27.1,-3.5)		
	(-31.6,-4.7)	(-30.,1.4)	(-25.,10.3)	(-20.,15.4)	(-15.,18.5)
-19.	(-10.,19.7)	(-5.,20.2)	(0.,20.3)	(5.,19.9)	(10.,19.1)
	(15.,17.3)	(20.,13.5)	(25.,7.3)	(29.2,-4.6)	
	(-33.4,-5.6)	(-32.,0.4)	(-30.,5.5)	(-25.,12.9)	(-20.,17.7)
-24.	(-15.,20.)	(-10.,21.)	(-5.,21.5)	(0.,21.6)	(5.,21.3)
	(10.,20.5)	(15.,19.2)	(20.,16.)	(25.,10.3)	(30.,0.)
	(31.2,-6.4)				
	(-34.,-7.4)	(-32.,3.1)	(-30.,7.6)	(-25.,14.7)	(-20.,19.4)
-30.	(-15.,21.3)	(-10.,22.2)	(-5.,22.8)	(0.,22.8)	(5.,22.5)
	(10.,20.5)	(15.,19.2)	(20.,17.7)	(25.,12.1)	(30.,2.8)
	(32.2,-8.4)				
	(-34.3,-8.8)	(-32.,3.1)	(-30.,7.8)	(-25.,15.5)	(-20.,20.3)
-35.	(-15.,22.)	(-10.,23.)	(-5.,23.5)	(0.,23.6)	(5.,23.2)
	(10.,22.4)	(15.,21.2)	(20.,18.6)	(25.,12.6)	(29.,5.3)
	(30.,2.6)	(32.,-9.8)			
	(-33.4,-10.)	(-32.4,-.8)	(-30.,6.9)	(-25.,15.5)	(-20.,20.8)
-40.	(-15.,22.5)	(-10.,23.5)	(-5.,24.)	(0.,24.1)	(5.,23.7)
	(10.,22.9)	(15.,21.8)	(20.,19.)	(25.,12.4)	(29.,3.7)
	(30.5,-2.3)	(31.1,-10.3)			

To be continued.

TABLE 4.1 continued.

Y	(X,Z)				
-45.	(-32.,-8.3)	(-31.5,-1.9)	(-29.,7.6)	(-25.,15.3)	(-20.,21.1)
	(-15.,22.8)	(-10.,23.8)	(0.,24.4)	(10.,23.2)	(15. 22.1)
	(20.,19.1)	(25.,11.6)	(27.,7.1)	(29.,-0.4)	(29.8,-9.3)
-52.	(-29.4,-2.2)	(-28.5,4.9)	(-25.,14.)	(-22.,19.2)	(-20.,21.1)
	(-17.,22.4)	(-15.,22.9)	(-10.,23.9)	(0.,24.4)	(10.,23.3)
	(15.,22.2)	(17.,21.5)	(20.,18.9)	(22.,15.8)	(25.,9.1)
	(26.5,3.2)	(27.1,-3.5)			
-58.	(-26.8,3.5)	(-25.,12.7)	(-23.1,17.2)	(-22.,19.)	(-20.,21.1)
	(-17.,22.4)	(-15.,22.9)	(-10.,23.9)	(0.,24.5)	(10.,23.4)
	(15.,22.2)	(17.,21.5)	(20.,18.5)	(22.5,13.3)	(24.6,3.7)
-64.	(-24.8,8.6)	(-24.,13.2)	(-22.8,17.)	(-21.5,19.4)	(-18.5,21.8)
	(-15.,22.8)	(-10.,23.7)	(0.,24.4)	(10.,23.2)	(14.,22.3)
	(17.,21.3)	(20.,17.9)	(21.8,13.4)	(22.7,8.3)	
-70.	(-23.8,8.9)	(-23.,14.8)	(-22.,18.)	(-20.,20.7)	(-17.,22.)
	(-15.,22.5)	(-10.,23.5)	(0.,24.1)	(10.,22.9)	(14.,22.)
	(17.,21.)	(20.,17.1)	(21.,13.2)	(21.5,8.8)	
-80.	(-23.6,7.6)	(-23.,13.4)	(-21.5,18.2)	(-19.,20.4)	(-17.,21.1)
	(-14.,21.9)	(-10.,22.6)	(-1.,23.3)	(10.,22.1)	(14.,21.2)
	(17.,20.1)	(20.,16.)	(21.,10.7)	(21.2,7.9)	
-100.	(-23.7,3.9)	(-23.,10.6)	(-22.,14.3)	(-20.,16.8)	(-17.5,18.)
	(-14.,18.9)	(-10.,19.6)	(-1.,20.2)	(9.,19.2)	(14.,18.1)
	(17.,17.)	(20.,12.5)	(20.8,8.1)	(21.,3.2)	
-120.	(-23.8,-1.4)	(-23.5,3.2)	(-22.,9.4)	(-21.,11.1)	(-19.,12.5)
	(-16.,13.5)	(-10.,14.9)	(-1.,15.6)	(9.,14.5)	(14.,13.3)
	(16.,12.6)	(18.,11.4)	(20.,7.3)	(20.6,3.2)	(20.9,-1.8)
-137.	(-24.,-7.2)	(-23.3,-1.)	(-22.3,3.2)	(-20.6,6.)	(-18.,7.5)
	(-15.,8.4)	(-10.,9.4)	(-1.,10.1)	(9.,9.1)	(14.,7.8)
	(17.,6.6)	(19.5,2.7)	(20.5,-2.5)	(20.8,-7.5)	
-150.	(-24.2,-7.2)	(-23.5,-6.6)	(-22.3,-1.8)	(-20.,1.3)	(-17.5,2.5)
	(-15.,3.3)	(-10.,4.4)	(-1.,5.1)	(9.,4.)	(14.,2.8)
	(17.,1.4)	(19.,-1.7)	(20.1,-5.9)	(20.8,-12.)	
-155.	(-25.1,-15.)	(-24.,-10.2)	(-22.5,-4.7)	(-20.,-0.9)	(-17.,0.5)
	(-14.,1.3)	(-10.,2.2)	(-1.,2.9)	(9.,1.8)	(14.,0.5)
	(17.,-1.)	(19.,-4.1)	(20.5,-9.5)	(21.6,-15.1)	
-158.	(-27.,-21.1)	(-25.,-13.6)	(-23.,-7.3)	(-20.,-2.2)	(-15.,-0.3)
	(-10.,0.8)	(0.,1.5)	(9.,0.4)	(14.,-0.9)	(17.,-2.5)
	(19.,-5.6)	(20.5,-10.3)	(22.,-15.7)	(23.4,-21.1)	

To be continued.

TABLE 4.1 continued.

Y	(X,Z)				
	(-32.9,-40.6)	(-30.,-29.4)	(-26.,-16.3)	(-23.,-8.3)	(-20.,-3.7)
-161.	(-16.,-2.)	(-10.,-0.6)	(-1.,0.1)	(9.,-1.1)	(13.5,-2.2)
	(16.5,-3.5)	(19.,-7.1)	(21.3,-13.)	(24.,-21.8)	(26.,-28.9)
	(28.2,-37.1)				
	(-34.8,-44.)	(-32.,-44.)	(-30.,-26.2)	(-25.,-13.6)	(-27.5,-19.5)
-164.	(-18.,-4.1)	(-21.,-6.)	(-15.,-3.2)	(-10.,-2.1)	(-5.,-1.5)
	(0.,-1.4)	(5.,-1.9)	(10.,-2.8)	(15.,-4.2)	(18.,-6.6)
	(20.,-10.4)	(22.,-14.9)	(24.7,-21.9)	(27.,-28.4)	(29.,-35.4)
	(30.9,-44.7)				
	(-36.4,-46.1)	(-34.3,-36.4)	(-32.,-29.4)	(-30.,-24.5)	(-27.,-18.)
-168.	(-25.,-14.1)	(-22.,-9.1)	(-20.,-7.2)	(-16.,-5.6)	(-12.,-4.5)
	(-7.,-3.7)	(0.,-3.5)	(8.,-4.4)	(12.,-5.4)	(15.,-6.3)
	(18.,-8.5)	(20.,-11.7)	(22.5,-16.4)	(25.,-21.8)	(27.5,-27.7)
	(30.,-35.)	(32.5,-46.1)			
	(-38.,-48.)	(-36.,-38.3)	(-34.,-33.1)	(-32.,-28.7)	(-29.,-22.9)
-175.	(-25.,-16.4)	(-22.,-12.6)	(-20.,-11.)	(-16.,-9.3)	(-10.,-8.)
	(-1.,-7.3)	(9.,-8.5)	(12.,-9.3)	(15.,-10.2)	(17.5,-11.8)
	(23.,-19.4)	(26.,-24.6)	(30.,-33.)	(33.1,-42.1)	(34.1,-48.1)
	(-38.1,-49.6)	(-36.7,-40.9)	(-34.,-34.)	(-32.,-30.2)	(-29.,-25.3)
-182.	(-25.,-19.8)	(-22.,-16.6)	(-19.,-14.7)	(-15.,-13.3)	(-10.,-12.2)
	(-5.,-11.6)	(0.,-11.5)	(5.,-12.)	(10.,-12.9)	(15.,-14.4)
	(18.5,-16.8)	(20.,-18.4)	(23.,-22.3)	(26.,-26.6)	(29.,-31.9)
	(31.8,-37.9)	(34.2,-49.4)			
	(-36.7,-51.3)	(-36.,-43.6)	(-34.,-37.4)	(-30.,-30.1)	(-25.,-23.9)
-189.	(-21.,-20.4)	(-18.,-18.9)	(-15.,-17.9)	(-10.,-16.8)	(-5.,-16.2)
	(0.,-16.1)	(5.,-16.6)	(10.,-17.5)	(15.,-19.2)	(18.,-20.9)
	(20.,-22.8)	(23.,-26.1)	(27.,-31.6)	(30.,-37.4)	(32.,-43.7)
	(32.8,-51.9)				
	(-33.6,-53.8)	(-32.8,-42.9)	(-31.,-37.8)	(-28.,-32.6)	(-24.,-28.2)
-196.	(-20.,-25.2)	(-16.,-23.5)	(-10.,-22.)	(-5.,-21.3)	(0.,-21.)
	(5.,-21.7)	(10.,-22.8)	(15.,-24.5)	(20.,-28.1)	(23.,-31.3)
	(26.,-35.8)	(29.5,-46.1)	(29.7,-53.8)		
	(-29.4,-54.6)	(-29.2,-46.1)	(-27.,-39.3)	(-25.,-35.9)	(-21.,-31.8)
-202.	(-17.,-29.3)	(-14.,-28.3)	(-10.,-27.3)	(-5.,-26.6)	(0.,-26.5)
	(5.,-27.)	(10.,-28.2)	(15.,-30.1)	(18.,-32.4)	(21.,-35.7)
	(24.2,-41.8)	(25.8,-49.6)	(25.4,-55.3)		
	(-25.3,-55.7)	(-25.8,-49.9)	(-23.8,-41.8)	(-20.,-36.1)	(-17.,-33.7)
-206.	(-13.,-32.)	(-10.,-31.3)	(-5.,-30.6)	(0.,-30.5)	(5.,-31.1)
	(10.,-32.3)	(14.,-34.2)	(18.,-38.1)	(20.,-41.7)	(21.9,-50.2)
	(21.4,-55.6)				

To be continued.

TABLE 4.1 continued.

Y	(X,Z)				
-209.	(-21.1,-56.5)	(-21.8,-48.6)	(-20.,-42.6)	(-16.,-37.6)	(12.,-35.2)
	(-8.,-34.2)	(-5.,-33.9)	(0.,-33.8)	(5.,-34.4)	(10.,-36.)
	(15.,-40.7)	(17.6,-46.6)	(17.1,-56.7)		
-212.	(-14.7,-57.1)	(-16.5,-50.8)	(-11.8,-41.)	(-6.3,-38.1)	(0.1,-37.8)
	(7.3,-40.5)	(10.,-43.1)	(12.4,-48.6)	(11.1,-56.6)	
-214.	(-8.5,-46.4)	(-2.5,-43.9)	(5.5,-49.)		

Extra points digitized (X,Y,Z):

(-10.9,-213.1,-57.5) (-2.4,-214.8,-50.6) (6.,-213.3,-58.)

Table 5.1 The examples showing the features of the developed programs

Items	Example 1	Example 2	Example 3
Work description	Calculating the volume of the bottle model shown in Figure 5.6	Scaling the bottle in Figure 5.6 in the cross-sectional direction. The scaled bottle is shown in Figure 5.10.	Scaling the middle part of the bottle in Figure 5.6 locally in the longitudal direction. The scaled bottle is shown in Figure 5.11.
The program used	VOLUME.UPL	SCLSPL.UPL	SCLSPL.UPL
Volume (pint)	1.124	1.686	0.868
Volume change	0.0	50 %	-22.8 %
Relative error	0.011 %	0.0	0.0
Operating time	4 hours	1 hour	20 minutes
Scalers	1, 1, 1	1.225, 1, 1.225	1, 0.5, 1
Computer used	CV386	CV386	CV386

Table 6.1 A comparison of the three- and five-axis machining of a commercial vehicle cab roof model

Items	Three-axis	Five-axis
Cutting tool	25 mm ball ended	25 mm flat ended
Stepover (mm)	2.0	12.5
Cusp height (mm)	0.04	0.04
No. of passes	37	7
Finish	Limited hand finishing required	No hand finishing required
Comments	Typically used for intricate and concave surfaces	Typically used for large convex surfaces. Only usable for concave surfaces if the tool is canted.

Table 6.2 Generating toolpaths for a bottle mould and a scaled cab roof model

Items	Example 1	Example 2
Model	Bottle in Figure 5.6	Cab roof in Figure 6.24
The radius of the fillets at the bottom and the shoulder of the bottle	4 mm	-
The tool diameter selected to cut the fillets	8 mm	-
Minimum radius of curvature of the surface patches except fillets	8.85 mm	Not necessary
Tool diameter selected to machine the surfaces except fillets	10 mm	25 and 15 mm
Offset amount	5.0 mm	12.5 and 7.5 mm
Cutting direction	Cross sectional	3 directions
Path intervals	2 mm	4 mm and 3 mm
Maximum cusp height	0.12 mm	0.15 mm
Maximum deviation	0.05 mm	0.05 mm
Time	9 hours	4 hours
System	IBM PC AT	CV386

Table 7.1 The NC machining of the bottle mould and the cab roof model

Items	The bottle mould	The cab roof model
The tools used for roughing	15 mm flat ended	10 mm flat ended
The tools used for finishing	8, 10, 5, 2 mm ball ended	25, 15 mm ball ended
The sizes of the blocks	300 115 45 mm wood	350 230 50 mm wax
Feed rate	500 IPM *	500 IPM
Spindle speed	3000 RPM **	1500 RPM
The time used for NC programming	24 hours	8 hours
Computer system	IBM PC AT	CV386
The time used for NC machining	4 hours	4 hours
Machining Centre	Takisawa	Takisawa

* Inches per minute
** Revolutions per minute

Appendix Three

PICTURES

Picture 1. The top half of a telephone handset modelled from digitized data

Picture 2. A plastics bottle modelled from 2D drawings

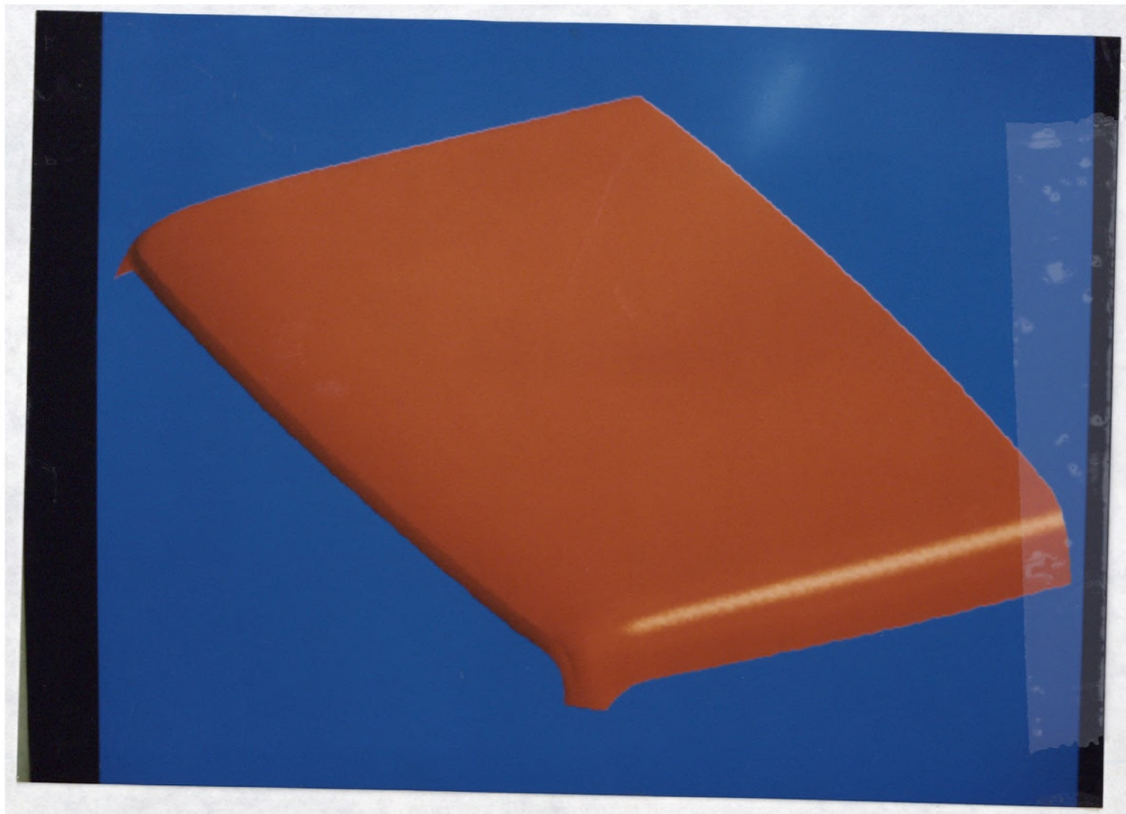
Picture 3. The model of a plastics bottle transferred from CV CADDS 4X system

Picture 4. The model of the scaled bottle shown in Picture 3

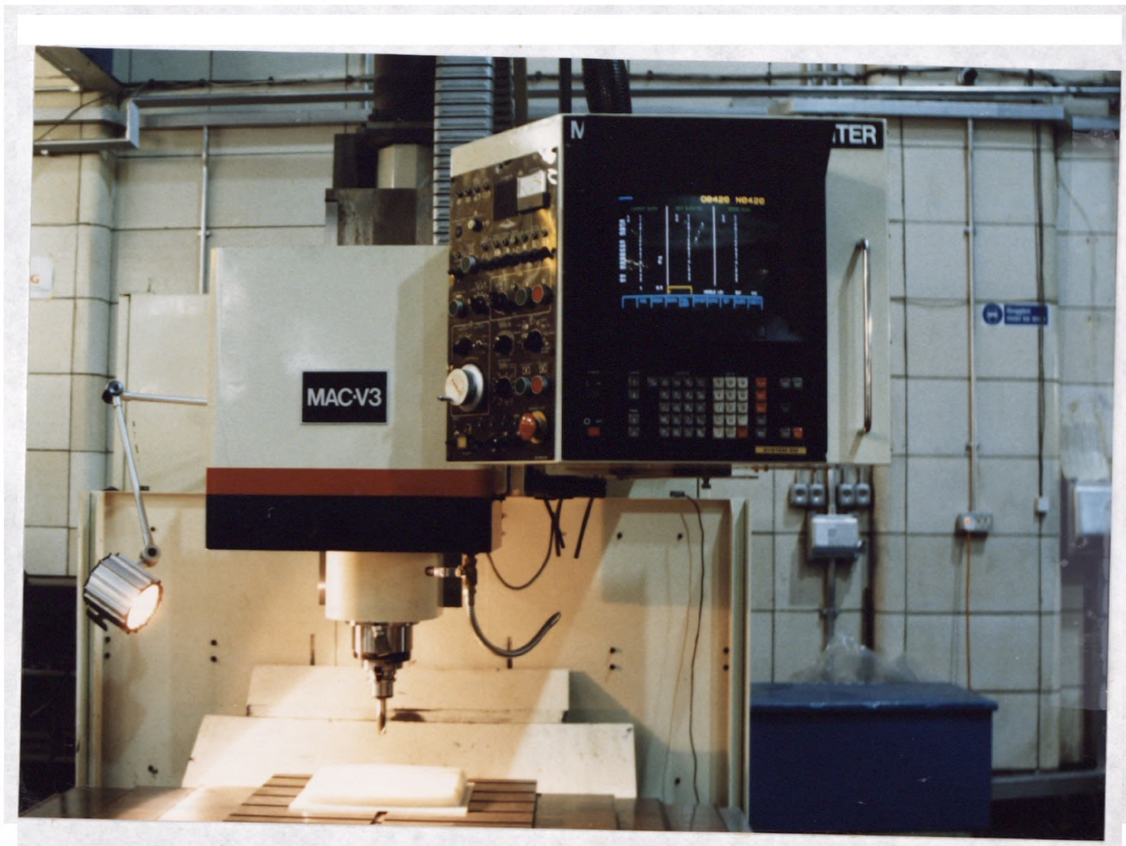
(50% increase in volume, scaled in the cross sectional direction)

Picture 5. The model of the locally scaled bottle shown in Picture 3 (22.8% decrease in volume, scaled in the longitudal direction)

Picture 6 The bottle models machined using incorrect sizes of tools and in a wrong direction



Picture 7 The model of the scaled cab roof supplied by LEYLAND-DAF
(Overall scaler: 0.18)



Picture 8 The UMIST Takisawa Machining centre

Picture 9. The wood blocks showing the results of the roughing and finishing operations for the bottle mould

Picture 10. The machined cab roof model

Appendix Four

THE PROGRAMS WRITTEN

IN

USER PROGRAMMING LANGUAGE (UPL)

SMOSPL.UPL

This program is used to smooth adjacent Bezier surface patches. It is based on the colinearity of the vertices of the control polyhedra of two adjacent surfaces and the coplanarity of the vertices of the control polyhedra of several adjacent surfaces at the joint corner. The program changes some vertices to satisfy the requirement and gives the new points of the vertices. The user is required to move the vertices to be changed to the new positions using microCADDS commands.

PROC MAIN

```
-------------------------------
---  Variable declaritions  ---
-------------------------------
INTEGER CH,CL,CH1        ---------------function selection parameters
INTEGER N1,N2,N3,I,LA,CA1,CA2  -------number of digitizing
REAL LM,LAM,LMB          ---------------tangent ratio
REAL V1,V2,V             ---------------coplanarity (volume=0)
REAL AAB,DAB             ---------------absolute value of vector axb
COORD B(3),B1(3),BM(1),P(4),VEC(4),H,AXB,CP

SEND
SEND 'SEL CPL 1'
SEND
--------------
---  Input ---
--------------
ACCEPT LA PROMPT ('layer number=?__')
ACCEPT CA1 PROMPT ('color number for colinearity=?__')
ACCEPT CA2 PROMPT ('color no. for coplanarity=?__')
-------------------------------------------------------------
AA:ACCEPT CH PROMPT ('1:colinear 2:coplanar 3:finish ?__')

IF CH=2 THEN
   GO_TO CC
ELSE IF CH=1 THEN
   GO_TO BB
ELSE
   GO_TO SS
END_IF

BB:ACCEPT CL PROMPT ('select tangent ratio, 1:default 2:input ?')

IF CL=1 THEN
   LM=1.0
ELSE
   PRINT 'dig three control points'
   N1=0
   GETEND (3,1,N1,B(1))
   PRINT 'finish digitizing'
   LAM=(B(2).X-B(1).X)^2.0+(B(2).Y-B(1).Y)^2.0+(B(2).Z-B(1).Z)^2.0
   LMB=(B(3).X-B(2).X)^2.0+(B(3).Y-B(2).Y)^2.0+(B(3).Z-B(2).Z)^2.0
   LM=(LAM/LMB)^0.5
END_IF

SEND
SEND 'REPAINT'
SEND

BBB:PRINT 'dig 3 control points to be made colinear'

N2=0
```

```
GETEND (3,1,N2,B1(1))

BM(1).X=(LM*B1(3).X+B1(1).X)/(1.0+LM)
BM(1).Y=(LM*B1(3).Y+B1(1).Y)/(1.0+LM)
BM(1).Z=(LM*B1(3).Z+B1(1).Z)/(1.0+LM)

INSERT POINT LOC(BM(1)) COLOR(CA1) LAYER(LA)

ACCEPT CH1 PROMPT ('1:continue smoothing 2:go back ?__')

IF CH1=1 THEN
   GO_TO BBB
ELSE
   SEND
   SEND 'REPAINT'
   SEND
   GO_TO AA
END_IF

CC:PRINT 'dig 4 control points to be made coplanar'
PRINT 'the first 3 will be fixed'

N3=0
GETEND (4,1,N3,P(1))

LOOP I=1 TO 3
      VEC(I).X=P(I+1).X-P(1).X
      VEC(I).Y=P(I+1).Y-P(1).Y
      VEC(I).Z=P(I+1).Z-P(1).Z
END_LOOP

V1=VEC(1).X*VEC(2).Y*VEC(3).Z+VEC(2).X*VEC(3).Y*VEC(1).Z
V1=V1+VEC(1).Y*VEC(2).Z*VEC(3).X
V2=VEC(1).Z*VEC(2).Y*VEC(3).X+VEC(2).X*VEC(3).Z*VEC(1).Y
V2=V2+VEC(1).X*VEC(2).Z*VEC(3).Y
V=ABS(V2-V1)

IF V<0.05 THEN
   PRINT 'coplanarity already achieved'
   GO_TO AA
END_IF

AXB.X=VEC(1).Y*VEC(2).Z-VEC(1).Z*VEC(2).Y
AXB.Y=VEC(1).Z*VEC(2).X-VEC(1).X*VEC(2).Z
AXB.Z=VEC(1).X*VEC(2).Y-VEC(1).Y*VEC(2).X
AAB=(AXB.X)^2.0+(AXB.Y)^2.0+(AXB.Z)^2.0
DAB=VEC(3).X*AXB.X+VEC(3).Y*AXB.Y+VEC(3).Z*AXB.Z

H.X=DAB/AAB*AXB.X
H.Y=DAB/AAB*AXB.Y
H.Z=DAB/AAB*AXB.Z
CP.X=P(4).X-H.X
CP.Y=P(4).Y-H.Y
CP.Z=P(4).Z-H.Z

VEC(4).X=CP.X-P(1).X
VEC(4).Y=CP.Y-P(1).Y
VEC(4).Z=CP.Z-P(1).Z

V1=VEC(1).X*VEC(2).Y*VEC(4).Z+VEC(2).X*VEC(4).Y*VEC(1).Z
V1=V1+VEC(1).Y*VEC(2).Z*VEC(4).X
V2=VEC(1).Z*VEC(2).Y*VEC(4).X+VEC(2).X*VEC(4).Z*VEC(1).Y
V2=V2+VEC(1).X*VEC(2).Z*VEC(4).Y
V=ABS(V1-V2)

IF V>0.05 THEN
```

```
   CP.X=H.X-P(4).X
   CP.Y=H.Y-P(4).Y
   CP.Z=H.Z-P(4).Z
END_IF

INSERT POINT LOC(CP) COLOR(CA2) LAYER(LA)

SEND
SEND 'REPAINT'
SEND

GO_TO AA

SS:ECHO ON

END PROC
```

SCLSPL.UPL

This program is used to scale surface models in the X, Y or/and Z direction. The surface model can also be scaled locally. The increase or decrease in volume can be calculated.

```
                         PROC MAIN

--------------------------------
---  Variable declaritions  ---
--------------------------------
COORD B1(8),B2(8),B3(8),B4(8)              ---Bi(j) are the vertices of
COORD B5(8),B6(8),B7(8),B8(8),B(8,8)       ---control polyhedra
REAL SX,SY,SZ
INTEGER NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
INTEGER NV,NU,I,J,CN,CN1,LNO,YORN,DW

--------Display instructions

WINDOW 5,5,25,64,79

DISP_WIN=5

CLEAR 5
DISPLAY


   This program
 scales a spole
 (Bezier,ruled)
 in x, y and/or
 z direction.

   Other spoles
 are  taken  as
 Bezier surface
 when put their
 polygons on.

  you only need
 to answer  the
 questions comi
 ng up when run
 ning the progr
 am.

 $
ACCEPT DW PROMPT(' GIVE A RETURN TO CONTINUE')

AA:ACCEPT YORN PROMPT('  1: scale surface; 2:finish__')
IF YORN=2 THEN
GO_TO BB
END_IF

NEND1=0; NEND2=0; NEND3=0; NEND4=0
NEND5=0; NEND6=0; NEND7=0; NEND5=0

PRINT 'dig the controlling points row by row'
GETEND(8,1,NEND1,B1(1))
GETEND(8,1,NEND2,B2(1))
GETEND(8,1,NEND3,B3(1))
GETEND(8,1,NEND4,B4(1))
GETEND(8,1,NEND5,B5(1))
```

```
GETEND(8,1,NEND6,B6(1))
GETEND(8,1,NEND7,B7(1))
GETEND(8,1,NEND8,B8(1))
PRINT '  finish dig  '
ACCEPT NV PROMPT('  no of points along v__')
ACCEPT NU PROMPT('  no of points along u__')
ACCEPT SX PROMPT('  scaling coefficient of X')
ACCEPT SY PROMPT('  scaling coeff of Y__')
ACCEPT SZ PROMPT('  scaling coeff of Z__')
ACCEPT LNO PROMPT('  layer no__')
ACCEPT CN PROMPT('  colour no of the controlling points__')
ACCEPT CN1 PROMPT('  colour no of the line__')

LOOP J=1 TO NV
B(1,J)=B1(J); B(2,J)=B2(J); B(3,J)=B3(J); B(4,J)=B4(J)
B(5,J)=B5(J); B(6,J)=B6(J); B(7,J)=B7(J); B(8,J)=B8(J)
END_LOOP

LOOP I=1 TO NU
LOOP J=1 TO NV
B(I,J).X=SX*B(I,J).X
B(I,J).Y=SY*B(I,J).Y
B(I,J).Z=SZ*B(I,J).Z
INSERT POINT COLOR(CN) LAYER(LNO) LOC(B(I,J))
END_LOOP
END_LOOP

LOOP I=1 TO NU
LOOP J=1 TO NV-1
INSERT LINE COLOR(CN1) LAYER(LNO) ENDS(B(I,J),B(I,J+1))
END_LOOP
END_LOOP

GO_TO AA

BB:PRINT 'end running'

END PROC
```

VOLUME.UPL

This program calculates the volume of a surface model and gives the relative error. The volume is calculated patch by patch and the control vertices of each surface patch must be input as follows.

```
-----------------------------------------------------------------
--                                                             --
--                                                             --
--      (column)                                               --
--        I      3  * * * * ---> digitize in this direction    --
--        u ^    2  * * * * --->                               --
--          |    1  * * * * --->                               --
--          |                                                  --
--          |--------> v (row) J                               --
--                                                             --
-----------------------------------------------------------------

-----------------------------------------------------------------
--                GROUP DECLARATIONS                           --
-----------------------------------------------------------------

GROUP

REAL U(41),V(61)                    ----matrix of parameters u,v
REAL CCU(8),CCV(8)                  ----c(n,i)=n!/i!/(n-i)!
COORD R(41,61)                      ----points selected on the surface
COORD B(8,8)                        ----control points
COORD SUM(8)
INTEGER M,N,K

END GROUP

-----------------------------------------------------------------
-- THIS SUBROUTINE CALCULATES PARAMETRIC CONSTANTS             --
-----------------------------------------------------------------

      PROC REDAT1(IN INTEGER NOPU,NOPV)

REAL UINC,VINC
INTEGER NP

UINC=1./(REAL(NOPU)-1.)
VINC=1./(REAL(NOPV)-1.)

U(1)=0.0
U(NOPU)=1.
LOOP NP=2 TO NOPU-1
U(NP)=U(NP-1)+UINC
END_LOOP

V(1)=0.0
V(NOPV)=1.
LOOP NP=2 TO NOPV-1
V(NP)=V(NP-1)+VINC
END_LOOP

END PROC

-----------------------------------------------------------------
-- THIS SUBROUTINE GETS THE CONTROL POINTS OF THE SURFACE      --
-----------------------------------------------------------------

      PROC REDAT(INOUT INTEGER NOBU,NOBV)
```

```
 INTEGER K1,NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
 COORD B1(8),B2(8),B3(8),B4(8),B5(8),B6(8),B7(8),B8(8)

 PRINT 'dig the controlling points row by row'
 PRINT 'give a <CR> for each row'
 NEND1=0;  NEND2=0;  NEND3=0;  NEND4=0
 NEND5=0;  NEND6=0;  NEND7=0;  NEND8=0
 GETEND(8,1,NEND1,B1(1))
 GETEND(8,1,NEND2,B2(1))
 GETEND(8,1,NEND3,B3(1))
 GETEND(8,1,NEND4,B4(1))
 PRINT 'row 5:'
 GETEND(8,1,NEND5,B5(1))
 PRINT 'row 6:'
 GETEND(8,1,NEND6,B6(1))
 PRINT 'row 7:'
 GETEND(8,1,NEND7,B7(1))
 PRINT 'row 8:'
 GETEND(8,1,NEND8,B8(1))

 NOBV=NEND1
 ACCEPT NOBU PROMPT('no. of rows:  ')

 LOOP K1=1 TO NOBV
   B(1,K1)=B1(K1);  B(2,K1)=B2(K1);  B(3,K1)=B3(K1);  B(4,K1)=B4(K1)
   B(5,K1)=B5(K1);  B(6,K1)=B6(K1);  B(7,K1)=B7(K1);  B(8,K1)=B8(K1)
 END_LOOP

 END PROC

 ----------------------------------------------------------
 -- THIS SUBROUTINE FURTHER CALCULATES THE CONSTANT MATRIX   --
 -- c(n,i) and c(m,j) in the general equation              --
 ----------------------------------------------------------

        PROC CCUV(IN INTEGER NOBU,NOBV)

 REAL CN,CNP,CNPB
 INTEGER NP1,NP2,J1

 CN=1.0
 LOOP NP1=2 TO NOBU
   CN=CN*REAL(NP1-1)
 END_LOOP

 CNP=1.0

 LOOP NP1=1 TO NOBU
   IF NP1=1 THEN
     CNP=1.0
   ELSE
     CNP=CNP*REAL(NP1-1)
   END_IF

   J1=NOBU-NP1
   IF NP1=NOBU THEN
     CNPB=1.0
   ELSE
     CNPB=1.0
     LOOP NP2=1 TO J1
       CNPB=CNPB*REAL(NP2)
     END_LOOP
   END_IF

   CCU(NP1)=CN/CNP/CNPB
```

```
END_LOOP

-------------------------------------

CN=1.0

LOOP NP1=2 TO NOBV
  CN=CN*REAL(NP1-1)
END_LOOP

CNP=1.0

LOOP NP1=1 TO NOBV
  IF NP1=1 THEN
    CNP=1.0
  ELSE
    CNP=CNP*REAL(NP1-1)
  END_IF

  J1=NOBV-NP1
  IF NP1=NOBV THEN
    CNPB=1.0
  ELSE
    CNPB=1.0
    LOOP NP2=1 TO J1
      CNPB=CNPB*REAL(NP2)
    END_LOOP
  END_IF

  CCV(NP1)=CN/CNP/CNPB

END_LOOP

END PROC



-----------------------------------------------------------------
-- THIS SUBROUTINE CALCULATES THE POINTS USED                  --
-- FOR VOLUME CALCULATION                                      --
-----------------------------------------------------------------


       PROC UBV(IN INTEGER I,J,NOPU,NOPV,NOBU,NOBV)

REAL UI,VJ,UR(8),VR(8),CCC
INTEGER K1,K2,J1,N1,NP1,NP2
REAL URN1,VRN

UI=U(I)
VJ=V(J)
PRINT '  I=',I,';  J=',J

IF I=1 THEN

  UR(1)=1.0
  LOOP K1=2 TO NOBU
    UR(K1)=0.0
  END_LOOP

ELSE IF I=NOPU THEN

  UR(NOBU)=1.0
  LOOP K1=1 TO NOBU-1
    UR(K1)=0.0
```

```
   END_LOOP

 ELSE

   LOOP NP1=1 TO NOBU

     J1=NOBU-NP1
     CCC=CCU(NP1)
     UR(NP1)=CCC*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1)

   END_LOOP

 END_IF

 -----------------------------------------------

 IF J=1 THEN
   VR(1)=1.0
   LOOP K1=2 TO NOBV
     VR(K1)=0.0
   END_LOOP

 ELSE IF J=NOPV THEN
   VR(NOBV)=1.0
   LOOP K1=1 TO NOBV-1
     VR(K1)=0.0
   END_LOOP

 ELSE

   LOOP NP1=1 TO NOBV

     J1=NOBV-NP1
     CCC=CCV(NP1)
     VR(NP1)=CCC*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1)

   END_LOOP

 END_IF

 LOOP N=1 TO NOBV
   SUM(N)=[0.0,0.0,0.0]
   LOOP N1=1 TO NOBU
     URN1=UR(N1)
     SUM(N).X=URN1*B(N1,N).X+SUM(N).X
     SUM(N).Y=URN1*B(N1,N).Y+SUM(N).Y
     SUM(N).Z=URN1*B(N1,N).Z+SUM(N).Z
   END_LOOP
 END_LOOP

 R(I,J)=[0.0,0.0,0.0]
 LOOP N=1 TO NOBV
   VRN=VR(N)
   R(I,J).X=SUM(N).X*VRN+R(I,J).X
   R(I,J).Y=SUM(N).Y*VRN+R(I,J).Y
   R(I,J).Z=SUM(N).Z*VRN+R(I,J).Z
 END_LOOP

 END PROC


 ---------------------------------------------------------------
 -- THIS IS THE MAIN PROGRAM WHICH GIVES CHOICES AND OUTPUT  --
 ---------------------------------------------------------------

                 PROC MAIN
```

```
INTEGER TYPE,TYPE1,I,J,NC1,NC2,VNO,VNO1
INTEGER NOPU,NOPV,NOBU,NOBV
REAL P1,P2,AIJ,VOL1
REAL VOL(40),VSUM,ASUM,AJSUM,AD,AJ,RER
REAL ABB,AC,BD,DC,BCC,OA,FA
COORD C(1)

SEND
SEND 'SEL CPL ',
SEND '1'
SEND

AST:ACCEPT TYPE PROMPT('1:continue   2:finish ')

IF TYPE=2 THEN
GO_TO BB
END_IF

REDAT(NOBU,NOBV)                    ----GET CONTROL POINTS
CCUV(NOBU,NOBV)                     ----CALCULATE c(n,i),c(m,j)

----NOW DECIDE THE NO. OF ELEMENTS ----------------------

BST:PRINT 'calculate the relative error'
ACCEPT NOPV PROMPT(' No. of elements each row (<31):   ')
NOPV=NOPV+1
NOPV=NOPV+NOPV-1
NOPU=4
REDAT1(NOPU,NOPV)                  -----CALCULATE PARAMETERS
PRINT 'dig one reference point'

NC1=0
GETDIG(1,1,NC1,C(1))

LOOP J=1 TO NOPV
I=1
UBV(I,J,NOPU,NOPV,NOBU,NOBV)        ----CALCULATE POINTS
END_LOOP

AJSUM=0.0
ASUM=0.0
LOOP J=1 TO NOPV-2
ABB=(R(1,J).X-R(1,J+2).X)^2.+(R(1,J).Y-R(1,J+2).Y)^2.
ABB=ABB+(R(1,J).Z-R(1,J+2).Z)^2.
ABB=ABB^0.5

AC=(R(1,J).X-R(1,J+1).X)^2.+(R(1,J).Y-R(1,J+1).Y)^2.
AC=AC+(R(1,J).Z-R(1,J+1).Z)^2.
AC=AC^0.5

BCC=(R(1,J+1).X-R(1,J+2).X)^2.+(R(1,J+1).Y-R(1,J+2).Y)^2.
BCC=BCC+(R(1,J+1).Z-R(1,J+2).Z)^2.
BCC=BCC^0.5

P1=0.5*(ABB+BCC+AC)
AIJ=(P1*(P1-ABB)*(P1-BCC)*(P1-AC))^0.5
ASUM=ASUM+AIJ

AD=(R(1,J).X-C(1).X)^2.+(R(1,J).Y-C(1).Y)^2.+(R(1,J).Z-C(1).Z)^2.
AD=AD^0.5

BD=(R(1,J+2).X-C(1).X)^2.+(R(1,J+2).Y-C(1).Y)^2.
BD=BD+(R(1,J+2).Z-C(1).Z)^2.
BD=BD^0.5
```

```
P2=0.5*(AD+BD+ABB)
AJ=(P2*(P2-AD)*(P2-ABB)*(P2-BD))^0.5

AJSUM=AJSUM+AJ

END_LOOP

RER=1.5*ASUM/AJSUM

PRINT 'the relative error is',RER

ACCEPT TYPE1 PROMPT('1:error acceptable   2:not acceptable')
IF TYPE1=2 THEN
GO_TO BST
END_IF

ACCEPT NOPU PROMPT('the new no. of elements for each row')
ACCEPT NOPV PROMPT('the new no. of elements for each colume')

NOPU=NOPU+1
NOPV=NOPV+1

REDAT1(NOPU,NOPV)              ----RE-SET THE PARAMETERS

---------------------------------------------------

PRINT 'case1 has a symmetric point'
PRINT 'case2 has a symmetric axis'
PRINT 'case3 has a symmetric plane'

BA: ACCEPT TYPE PROMPT(' 1:case1  2:case2  3:case3  4:go back ?  ')

IF TYPE=4 THEN
GO_TO AST
END_IF

LOOP I=1 TO NOPU
  LOOP J=1 TO NOPV

  UBV(I,J,NOPU,NOPV,NOBU,NOBV)

  END_LOOP
END_LOOP

IF TYPE=1 THEN

PRINT 'the vertex of the volume must be the origion'

VOL1=0.0

LOOP I=1 TO NOPU-1
  LOOP J=1 TO NOPV-1

  PRINT 'element ',I,'__',J

  ABB=(R(I,J).X-R(I,J+1).X)^2.+(R(I,J).Y-R(I,J+1).Y)^2.
  ABB=ABB+(R(I,J).Z-R(I,J+1).Z)^2.
  ABB=ABB^.5

  BCC=(R(I,J+1).X-R(I+1,J).X)^2.+(R(I,J+1).Y-R(I+1,J).Y)^2.
  BCC=BCC+(R(I,J+1).Z-R(I+1,J).Z)^2.
  BCC=BCC^.5

  AC=(R(I,J).X-R(I+1,J).X)^2.+(R(I,J).Y-R(I+1,J).Y)^2.
  AC=AC+(R(I,J).Z-R(I+1,J).Z)^2.
  AC=AC^.5
```

```
    P1=.5*(ABB+BCC+AC)

    BD=(R(I,J+1).X-R(I+1,J+1).X)^2.+(R(I,J+1).Y-R(I+1,J+1).Y)^2.
    BD=BD+(R(I,J+1).Z-R(I+1,J+1).Z)^2.
    BD=BD^.5

    DC=(R(I+1,J+1).X-R(I+1,J).X)^2.+(R(I+1,J+1).Y-R(I+1,J).Y)^2.
    DC=DC+(R(I+1,J+1).Z-R(I+1,J).Z)^2.
    DC=DC^.5

    P2=.5*(BD+DC+BCC)

    AIJ=(P1*(P1-ABB)*(P1-BCC)*(P1-AC))^.5+(P2*(P2-DC)*(P2-BCC)*(P2-BD))^.5

    OA=(R(I,J).X)^2.+(R(I,J).Y)^2.+(R(I,J).Z)^2.
    OA=OA^.5

    VOL1=VOL1+1./3.*OA*AIJ

    END_LOOP
  END_LOOP

  PRINT 'volume is:',VOL1

  ACCEPT VNO PROMPT('surface patch number=?  ')
  VOL(VNO)=VOL1

  ELSE IF TYPE=2 THEN

  PRINT 'symmetric axis must be Y '

  VOL1=0.

  LOOP I=1 TO NOPU-1
    LOOP J=1 TO NOPV-1

    PRINT 'element ',I,'__',J

    FA=(R(I,J).X)^2.+(R(I,J).Z)^2.
    FA=FA^.5

    AC=(R(I,J).X-R(I,J+1).X)^2.+(R(I,J).Y-R(I,J+1).Y)^2.
    AC=AC+(R(I,J).Z-R(I,J+1).Z)^2.
    AC=AC^.5

    ABB=(R(I,J).X-R(I+1,J).X)^2.+(R(I,J).Y-R(I+1,J).Y)^2.
    ABB=ABB+(R(I,J).Z-R(I+1,J).Z)^2.
    ABB=ABB^.5

    VOL1=VOL1+0.5*FA*ABB*AC

    END_LOOP
  END_LOOP

  PRINT 'volume is:',VOL1

  ACCEPT VNO PROPMT('surface patch number=?  ')
  VOL(VNO)=VOL1

  ELSE IF TYPE=3 THEN

  PRINT 'the symmetric plane must be XOY'
  VOL1=0.

  LOOP I=1 TO NOPU-1
```

```
   LOOP J=1 TO NOPV-1

   PRINT 'element ',I,'__',J

   ABB=((R(I,J).X-R(I,J+1).X)^2.+(R(I,J).Y-R(I,J+1).Y)^2.)^.5
   BCC=((R(I,J+1).X-R(I+1,J).X)^2.+(R(I,J+1).Y-R(I+1,J).Y)^2.)^.5
   AC=((R(I,J).X-R(I+1,J).X)^2.+(R(I,J).Y-R(I+1,J).Y)^2.)^.5
   P1=.5*(ABB+BCC+AC)
   BD=((R(I,J+1).X-R(I+1,J+1).X)^2.+(R(I,J+1).Y-R(I+1,J+1).Y)^2.)^.5
   DC=((R(I+1,J).X-R(I+1,J+1).X)^2.+(R(I+1,J).Y-R(I+1,J+1).Y)^2.)^.5
   P2=.5*(BD+DC+BCC)
   AIJ=(P1*(P1-ABB)*(P1-BCC)*(P1-AC))^.5+(P2*(P2-BD)*(P2-DC)*(P2-BCC))^.5

   VOL1=VOL1+AIJ*ABS(R(I,J).Z)

   END_LOOP
END_LOOP

PRINT 'volume is:',vol1

ACCEPT VNO PROMPT('surface patch number=?')
VOL(VNO)=VOL1

END_IF

ACCEPT TYPE1 PROMPT(' 1:do another surface  2:finish ')

IF TYPE1=1 THEN
  GO_TO AST
END_IF

ACCEPT VNO1 PROMPT('total number of patches=?  ')
VSUM=0.0

LOOP I=1 TO VNO1
VSUM=VSUM+VOL(I)
END_LOOP

BB:ECHO ON

PRINT 'total volume of ',VNO1,' patches =',VSUM

END PROC
```

RADIUS.UPL

This program calcultes the minimum radius of curvature of a surface model so that the maximum radius of the ball ended tool used to machine the model or the mould of the model can be determined. The vertices of the control polyhedra of the surface patches are input as follows.

```
-----------------------------------------------------------------
--                                                              --
-- Calculates maximum tool radius allowed to machine a surface --
--                                                              --
--       (column)                                               --
--                 3  * * * * ---> DIG IN THIS DIRECTION (ROWS)  --
--          u ^    2  * * * * --->                              --
--            |    1  * * * * --->                              --
--            |                                                 --
--            |--------> v (row)                                --
--                                                              --
-----------------------------------------------------------------

-----------------------------------------------------------------
--                  GROUP DECLARATIONS                          --
-----------------------------------------------------------------

GROUP

REAL U(8),V(8),CCU(8),CCV(8)        ----------parameters of the patch
COORD DRUU(8,8),DRUV(8,8),DRVV(8,8)  --------partial differials
COORD R(8,8),B(8,8),RU(8,8),RV(8,8),RUV(8,8) --and coordinate variables
COORD RUVAB(8,8),P1(8,8),P2(8,8)
COORD SUM(8)
INTEGER M,N,K

END GROUP

-----------------------------------------------------------------
-- THIS SUBROUTINE GETS INPUT INFORMATION OF A SURFACE AND      --
-- CALCULATES PARAMETRIC CONSTANTS                              --
-----------------------------------------------------------------

      PROC REDAT(IN INTEGER NOPU,NOPV;INOUT INTEGER NOBU,NOBV)

INTEGER NP,NOP,K1,NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
COORD B1(8),B2(8),B3(8),B4(8),B5(8),B6(8),B7(8),B8(8)

U(1)=0.0
U(NOPU)=1.0
LOOP NP=2 TO NOPU-1
  U(NP)=U(NP-1)+1.0/(REAL(NOPU)-1.0)
END_LOOP

V(1)=0.0
V(NOPV)=1.0
LOOP NOP=2 TO NOPV-1
  V(NOP)=V(NOP-1)+1.0/(REAL(NOPV)-1.0)
END_LOOP

PRINT 'dig the controlling points row by row'
PRINT 'give a <CR> for each row'
NEND1=0;  NEND2=0;  NEND3=0;  NEND4=0
NEND5=0;  NEND6=0;  NEND7=0;  NEND8=0
GETEND(8,1,NEND1,B1(1))
GETEND(8,1,NEND2,B2(1))
```

```
 GETEND(8,1,NEND3,B3(1))
 GETEND(8,1,NEND4,B4(1))
 PRINT 'row 5:'
 GETEND(8,1,NEND5,B5(1))
 PRINT 'row 6:'
 GETEND(8,1,NEND6,B6(1))
 PRINT 'row 7:'
 GETEND(8,1,NEND7,B7(1))
 PRINT 'row 8:'
 GETEND(8,1,NEND8,B8(1))

 NOBV=NEND1
 ACCEPT NOBU PROMPT('no. of rows:  ')

 LOOP K1=1 TO NOBV
   B(1,K1)=B1(K1);  B(2,K1)=B2(K1);  B(3,K1)=B3(K1);  B(4,K1)=B4(K1)
   B(5,K1)=B5(K1);  B(6,K1)=B6(K1);  B(7,K1)=B7(K1);  B(8,K1)=B8(K1)
 END_LOOP

 END PROC

 -------------------------------------------------------------
 -- THIS SUBROUTINE FURTHER CALCULATES THE CONSTANT MATRIX   --
 -- C(n,i) & C(m,j) IN THE BEZIER SURFACE FORM: r=UBV        --
 -------------------------------------------------------------

        PROC CCUV(IN INTEGER NOBU,NOBV)

 REAL CN,CNP,CNPB
 INTEGER NP1,NP2,J1

 CN=1.0
 LOOP NP1=2 TO NOBU
   CN=CN*REAL(NP1-1)
 END_LOOP

 CNP=1.0

 LOOP NP1=1 TO NOBU
   IF NP1=1 THEN
     CNP=1.0
   ELSE
     CNP=CNP*REAL(NP1-1)
   END_IF

   J1=NOBU-NP1
   IF NP1=NOBU THEN
     CNPB=1.0
   ELSE
     CNPB=1.0
     LOOP NP2=1 TO J1
       CNPB=CNPB*REAL(NP2)
     END_LOOP
   END_IF

   CCU(NP1)=CN/CNP/CNPB

 END_LOOP

 ------------------------------------

 CN=1.0

 LOOP NP1=2 TO NOBV
   CN=CN*REAL(NP1-1)
 END_LOOP
```

```
 CNP=1.0

 LOOP NP1=1 TO NOBV
   IF NP1=1 THEN
     CNP=1.0
   ELSE
     CNP=CNP*REAL(NP1-1)
   END_IF

   J1=NOBV-NP1
   IF NP1=NOBV THEN
     CNPB=1.0
   ELSE
     CNPB=1.0
     LOOP NP2=1 TO J1
       CNPB=CNPB*REAL(NP2)
     END_LOOP
   END_IF

   CCV(NP1)=CN/CNP/CNPB

END_LOOP

END PROC


-----------------------------------------------------------------
--   THIS SUBROUTINE CALCULATES THE PARTIAL DIFFERENTIALS      --
-----------------------------------------------------------------


       PROC UBV(IN INTEGER I,J,NOPU,NOPV,NOBU,NOBV)

REAL UI,VJ,DU(8),UR(8),VR(8),DV(8),CCC,UUU1,UUU2
INTEGER K1,K2,J1,N1,NP1,NP2
REAL URN1,VRN,DUN1,DVN,UUU3,UUU4,DDU(8),DDV(8)

UI=U(I)
VJ=V(J)
PRINT '  I=',I,';  J=',J

LOOP NP1=1 TO NOBU

J1=NOBU-NP1
CCC=CCU(NP1)
UR(NP1)=CCC*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1)
UUU1=CCC*REAL(NP1-1)*UI^REAL(NP1-2)*(1.0-UI)^REAL(J1)
UUU2=CCC*REAL(J1)*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1-1)
DU(NP1)=UUU1-UUU2

UUU1=REAL((NP1-1)*(NP1-2))*UI^REAL(NP1-3)*(1.0-UI)^REAL(J1)
UUU2=REAL((NP1-1)*J1)*UI^REAL(NP1-2)*(1.0-UI)^REAL(J1-1)
UUU3=REAL((NP1-1)*(NP1-2)*J1)*UI^REAL(NP1-3)*(1.0-UI)^REAL(J1-1)
UUU4=REAL((NP1-1)*J1*(J1-1))*UI^REAL(NP1-2)*(1.0-UI)^REAL(J1-2)
DDU(NP1)=CCC*(UUU1-UUU2-UUU3+UUU4)

END_LOOP

----------------------------------------------

LOOP NP1=1 TO NOBV

J1=NOBV-NP1
CCC=CCV(NP1)
```

```
VR(NP1)=CCC*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1)
UUU1=CCC*REAL(NP1-1)*VJ^REAL(NP1-2)*(1.0-VJ)^REAL(J1)
UUU2=CCC*REAL(J1)*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1-1)
DV(NP1)=UUU1-UUU2

UUU1=REAL((NP1-1)*(NP1-2))*VJ^REAL(NP1-3)*(1.0-VJ)^REAL(J1)
UUU2=REAL((NP1-1)*J1)*VJ^REAL(NP1-2)*(1.0-VJ)^REAL(J1-1)
UUU3=REAL((NP1-1)*(NP1-2)*J1)*VJ^REAL(NP1-3)*(1.0-VJ)^REAL(J1-1)
UUU4=REAL((NP1-1)*J1*(J1-1))*VJ^REAL(NP1-2)*(1.0-VJ)^REAL(J1-2)
DDV(NP1)=CCC*(UUU1-UUU2-UUU3+UUU4)

END_LOOP

-----------------------------------------------------------------

LOOP N=1 TO NOBV
  SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBU
    DUN1=DU(N1)
    SUM(N).X=DUN1*B(N1,N).X+SUM(N).X
    SUM(N).Y=DUN1*B(N1,N).Y+SUM(N).Y
    SUM(N).Z=DUN1*B(N1,N).Z+SUM(N).Z
  END_LOOP
END_LOOP

RU(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
  VRN=VR(N)
  RU(I,J).X=SUM(N).X*VRN+RU(I,J).X
  RU(I,J).Y=SUM(N).Y*VRN+RU(I,J).Y
  RU(I,J).Z=SUM(N).Z*VRN+RU(I,J).Z
END_LOOP

---------------------------------------

LOOP N=1 TO NOBV
  SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBU
    URN1=UR(N1)
    SUM(N).X=URN1*B(N1,N).X+SUM(N).X
    SUM(N).Y=URN1*B(N1,N).Y+SUM(N).Y
    SUM(N).Z=URN1*B(N1,N).Z+SUM(N).Z
  END_LOOP
END_LOOP

RV(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
  DVN=DV(N)
  RV(I,J).X=SUM(N).X*DVN+RV(I,J).X
  RV(I,J).Y=SUM(N).Y*DVN+RV(I,J).Y
  RV(I,J).Z=SUM(N).Z*DVN+RV(I,J).Z
END_LOOP

-----------------------------------------

LOOP N=1 TO NOBV
SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBU
  SUM(N).X=DDU(N1)*B(N1,N).X+SUM(N).X
  SUM(N).Y=DDU(N1)*B(N1,N).Y+SUM(N).Y
  SUM(N).Z=DDU(N1)*B(N1,N).Z+SUM(N).Z
  END_LOOP
END_LOOP

DRUU(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
```

```
DRUU(I,J).X=VR(N)*SUM(N).X+DRUU(I,J).X
DRUU(I,J).Y=VR(N)*SUM(N).Y+DRUU(I,J).Y
DRUU(I,J).Z=VR(N)*SUM(N).Z+DRUU(I,J).Z
END_LOOP

------------------------------------------

LOOP N=1 TO NOBV
SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBU
  SUM(N).X=UR(N1)*B(N1,N).X+SUM(N).X
  SUM(N).Y=UR(N1)*B(N1,N).Y+SUM(N).Y
  SUM(N).Z=UR(N1)*B(N1,N).Z+SUM(N).Z
  END_LOOP
END_LOOP

DRVV(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
DRVV(I,J).X=DDV(N)*SUM(N).X+DRVV(I,J).X
DRVV(I,J).Y=DDV(N)*SUM(N).Y+DRVV(I,J).Y
DRVV(I,J).Z=DDV(N)*SUM(N).Z+DRVV(I,J).Z
END_LOOP

------------------------------------------

LOOP N=1 TO NOBV
SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBU
  SUM(N).X=DU(N1)*B(N1,N).X+SUM(N).X
  SUM(N).Y=DU(N1)*B(N1,N).Y+SUM(N).Y
  SUM(N).Z=DU(N1)*B(N1,N).Z+SUM(N).Z
  END_LOOP
END_LOOP

DRUV(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
DRUV(I,J).X=DV(N)*SUM(N).X+DRUV(I,J).X
DRUV(I,J).Y=DV(N)*SUM(N).Y+DRUV(I,J).Y
DRUV(I,J).Z=DV(N)*SUM(N).Z+DRUV(I,J).Z
END_LOOP

END PROC

----------------------------------------------------------
-- THIS IS THE MAIN PROGRAM WHICH GIVES CHOICES AND OUTPUT  --
----------------------------------------------------------

                PROC MAIN

INTEGER TYPE,TYPE1,I,J,CNO
INTEGER NOPU,NOPV,NOBU,NOBV
REAL C,G11,G12,G21,G22,d11,d12,d21,d22
REAL KNM1,KNM2,KN1,KN2,EQA,EQB,EQC,MINR1,MINR2

ACCEPT NOPV PROMPT(' No of pois to be tested each row:   ')
ACCEPT NOPU PROMPT(' No of points each column:   ')

-----------------------------------------------

BST:REDAT(NOPU,NOPV,NOBU,NOBV)        -----CALL REDAT

CCUV(NOBU,NOBV)                       ------CALL CCUV

BA: ACCEPT TYPE PROMPT(' 1:CALCULATE  2:FINISH  3:START AGAIN  ?')

IF TYPE=1 THEN
```

```
 KNM1=0.0
 KNM2=0.0
 LOOP I=2 TO NOPU-1
   LOOP J=2 TO NOPV-1

     UBV(I,J,NOPU,NOPV,NOBU,NOBV)
     RUV(I,J).X=RU(I,J).Y*RV(I,J).Z-RU(I,J).Z*RV(I,J).Y
     RUV(I,J).Y=RU(I,J).Z*RV(I,J).X-RU(I,J).X*RV(I,J).Z
     RUV(I,J).Z=RU(I,J).X*RV(I,J).Y-RU(I,J).Y*RV(I,J).X
     C=(RUV(I,J).X^2.0+RUV(I,J).Y^2.0+RUV(I,J).Z^2.0)^0.5

     G11=(RU(I,J).X)^2.0+(RU(I,J).Y)^2.0+(RU(I,J).Z)^2.0
     G21=RU(I,J).X*RV(I,J).X+RU(I,J).Y*RV(I,J).Y+RU(I,J).Z*RV(I,J).Z
     G12=G21
     G22=(RV(I,J).X)^2.+(RV(I,J).Y)^2.+(RV(I,J).Z)^2.

     D11=RUV(I,J).X/C*DRUU(I,J).X+RUV(I,J).Y/C*DRUU(I,J).Y
     D11=D11+RUV(I,J).Z/C*DRUU(I,J).Z
     D21=RUV(I,J).X/C*DRUV(I,J).X+RUV(I,J).Y/C*DRUV(I,J).Y
     D21=D21+RUV(I,J).Z/C*DRUV(I,J).Z
     D12=D21
     D22=RUV(I,J).X/C*DRVV(I,J).X+RUV(I,J).Y/C*DRVV(I,J).Y
     D22=D22+RUV(I,J).Z/C*DRVV(I,J).Z
     EQA=G11*G22-G12*G21
     EQB=G11*D22+D11*G22-2.0*G12*D12
     EQC=D11*D22-D12*D21

     IF EQB^2.0-4.0*EQA*EQC < 0.0 THEN
     PRINT ' no real root '
     GO_TO LBB
     END_IF

     KN1=(-EQB+(EQB^2.0-4.0*EQA*EQC)^0.5)/2.0/EQA
     KN2=(-EQB-(EQB^2.0-4.0*EQA*EQC)^0.5)/2.0/EQA

     IF KN1 > 0.0 THEN
       IF KN1 > KNM1 THEN
       KNM1=KN1
       END_IF
     ELSE
       IF ABS(KN1) > KNM2 THEN
       KNM2=ABS(KN1)
       END_IF
     END_IF

     IF KN2 > 0.0 THEN
       IF KN2 > KNM1 THEN
       KNM1=KN2
       END_IF
     ELSE
       IF ABS(KN2) > KNM2 THEN
       KNM2=ABS(KN2)
       END_IF
     END_IF
LBB:ECHO ON
  END_LOOP
END_LOOP

SEND
SEND 'REPAINT'
SEND

MINR1=1.0/KNM1
MINR2=1.0/KNM2
PRINT 'min radii of tool for 2 sides are :',MINR1,'and',MINR2
```

```
ELSE IF TYPE=2 THEN
  GO_TO BB

ELSE IF TYPE=3 THEN
  GO_TO BST

END_IF

ACCEPT TYPE1 PROMPT(' 1:do another surface  2:finish ')

IF TYPE1=1 THEN
  GO_TO BST
END_IF

BB:ECHO ON

END PROC
```

BEZOFF.UPL

This program calculates the offset points to a part surface and generates curves from the offset points automatically. These curves are later used used for generating the offset surface to the part surface. The vertices of the control polyhedra of the surface patches of a model is input as follows.

```
------------------------------------------------------------------
--                                                              --
--      Offset Bezier surfaces                                  --
--                                                              --
--      (column)                                                --
--                3  * * * * ---> DIG IN THIS DIRECTION (ROWS)  --
--        u ^     2  * * * * --->                               --
--          |     1  * * * * --->    SIDE A ( . )               --
--          |                        SIDE B ( x )               --
--          |---------> v (row)                                 --
--                                                              --
------------------------------------------------------------------

------------------------------------------------------------------
--                  GROUP DECLARATIONS                          --
------------------------------------------------------------------

GROUP

REAL U(8),V(8),CCU(8),CCV(8)
COORD R(8,8),B(8,8),RU(8,8),RV(8,8),RUV(8,8)
COORD RUVAB(8,8),P1(8,8),P2(8,8)
COORD SUM(8)
INTEGER M,N,K

END GROUP

------------------------------------------------------------------
-- THIS SUBROUTINE GETS INPUT INFORMATION OF A SURFACE AND      --
-- CALCULATES PARAMETRIC CONSTANTS                              --
------------------------------------------------------------------

      PROC REDAT(IN INTEGER NOPU,NOPV;INOUT INTEGER NOBU,NOBV)

INTEGER NP,NOP,K1,NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
COORD B1(8),B2(8),B3(8),B4(8),B5(8),B6(8),B7(8),B8(8)

U(1)=0.0
U(NOPU)=1.0
LOOP NP=2 TO NOPU-1
  U(NP)=U(NP-1)+1.0/(REAL(NOPU)-1.0)
END_LOOP

V(1)=0.0
V(NOPV)=1.0
LOOP NOP=2 TO NOPV-1
  V(NOP)=V(NOP-1)+1.0/(REAL(NOPV)-1.0)
END_LOOP

PRINT 'dig the controlling points row by row'
PRINT 'give a <CR> for each row'
NEND1=0;  NEND2=0;  NEND3=0;  NEND4=0
NEND5=0;  NEND6=0;  NEND7=0;  NEND8=0
GETEND(8,1,NEND1,B1(1))
GETEND(8,1,NEND2,B2(1))
```

```
 GETEND(8,1,NEND3,B3(1))
 GETEND(8,1,NEND4,B4(1))
 PRINT 'row 5:'
 GETEND(8,1,NEND5,B5(1))
 PRINT 'row 6:'
 GETEND(8,1,NEND6,B6(1))
 PRINT 'row 7:'
 GETEND(8,1,NEND7,B7(1))
 PRINT 'row 8:'
 GETEND(8,1,NEND8,B8(1))

 NOBV=NEND1
 ACCEPT NOBU PROMPT('no. of rows:  ')

 LOOP K1=1 TO NOBV
   B(1,K1)=B1(K1);  B(2,K1)=B2(K1);  B(3,K1)=B3(K1);  B(4,K1)=B4(K1)
   B(5,K1)=B5(K1);  B(6,K1)=B6(K1);  B(7,K1)=B7(K1);  B(8,K1)=B8(K1)
 END_LOOP

 END PROC

 -------------------------------------------------------------
 -- THIS SUBROUTINE FURTHER CALCULATES THE CONSTANT MATRIX   --
 -- U, V IN THE BEZIER SURFACE FORM: r=UBV                   --
 -------------------------------------------------------------

        PROC CCUV(IN INTEGER NOBU,NOBV)

 REAL CN,CNP,CNPB
 INTEGER NP1,NP2,J1

 CN=1.0
 LOOP NP1=2 TO NOBU
   CN=CN*REAL(NP1-1)
 END_LOOP

 CNP=1.0

 LOOP NP1=1 TO NOBU
   IF NP1=1 THEN
     CNP=1.0
   ELSE
     CNP=CNP*REAL(NP1-1)
   END_IF

   J1=NOBU-NP1
   IF NP1=NOBU THEN
     CNPB=1.0
   ELSE
     CNPB=1.0
     LOOP NP2=1 TO J1
       CNPB=CNPB*REAL(NP2)
     END_LOOP
   END_IF

   CCU(NP1)=CN/CNP/CNPB

 END_LOOP

 ------------------------------------

 CN=1.0

 LOOP NP1=2 TO NOBV
   CN=CN*REAL(NP1-1)
 END_LOOP
```

```
CNP=1.0

LOOP NP1=1 TO NOBV
  IF NP1=1 THEN
    CNP=1.0
  ELSE
    CNP=CNP*REAL(NP1-1)
  END_IF

  J1=NOBV-NP1
  IF NP1=NOBV THEN
    CNPB=1.0
  ELSE
    CNPB=1.0
    LOOP NP2=1 TO J1
      CNPB=CNPB*REAL(NP2)
    END_LOOP
  END_IF

  CCV(NP1)=CN/CNP/CNPB

END_LOOP

END PROC



-----------------------------------------------------------------
-- THIS SUBROUTINE CALCULATES THE OFFSET POINTS OF A SURFACE  --
-----------------------------------------------------------------


       PROC UBV(IN INTEGER I,J,NOPU,NOPV,NOBU,NOBV)

REAL UI,VJ,DU(8),UR(8),VR(8),DV(8),CCC,UUU1,UUU2
INTEGER K1,K2,J1,N1,NP1,NP2
REAL URN1,VRN,DUN1,DVN

UI=U(I)
VJ=V(J)
PRINT '  I=',I,';  J=',J

IF I=1 THEN

  UR(1)=1.0
  LOOP K1=2 TO NOBU
    UR(K1)=0.0
  END_LOOP
  DU(1)=-REAL(NOBU-1)
  DU(2)=REAL(NOBU-1)
  LOOP K2=3 TO NOBU
    DU(K2)=0.0
  END_LOOP

ELSE IF I=NOPU THEN

  UR(NOBU)=1.0
  LOOP K1=1 TO NOBU-1
    UR(K1)=0.0
  END_LOOP
  DU(NOBU)=REAL(NOBU-1)
  DU(NOBU-1)=-REAL(NOBU-1)
  LOOP K2=1 TO NOBU-2
    DU(K2)=0.0
  END_LOOP
```

```
ELSE

   LOOP NP1=1 TO NOBU

      J1=NOBU-NP1
      CCC=CCU(NP1)
      UR(NP1)=CCC*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1)
      UUU1=CCC*REAL(NP1-1)*UI^REAL(NP1-2)*(1.0-UI)^REAL(J1)
      UUU2=CCC*REAL(J1)*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1-1)
      DU(NP1)=UUU1-UUU2

   END_LOOP

END_IF

---------------------------------------------

IF J=1 THEN
   VR(1)=1.0
   LOOP K1=2 TO NOBV
      VR(K1)=0.0
   END_LOOP
   DV(1)=-REAL(NOBV-1)
   DV(2)=REAL(NOBV-1)
   LOOP K2=3 TO NOBV
      DV(K2)=0.0
   END_LOOP

ELSE IF J=NOPV THEN
   VR(NOBV)=1.0
   LOOP K1=1 TO NOBV-1
      VR(K1)=0.0
   END_LOOP
   DV(NOBV)=REAL(NOBV-1)
   DV(NOBV-1)=-REAL(NOBV-1)
   LOOP K2=1 TO NOBV-2
      DV(K2)=0.0
   END_LOOP

ELSE

   LOOP NP1=1 TO NOBV
      J1=NOBV-NP1
      CCC=CCV(NP1)
      VR(NP1)=CCC*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1)
      UUU1=CCC*REAL(NP1-1)*VJ^REAL(NP1-2)*(1.0-VJ)^REAL(J1)
      UUU2=CCC*REAL(J1)*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1-1)
      DV(NP1)=UUU1-UUU2

   END_LOOP

END_IF

LOOP N=1 TO NOBV
   SUM(N)=[0.0,0.0,0.0]
   LOOP N1=1 TO NOBU
      URN1=UR(N1)
      SUM(N).X=URN1*B(N1,N).X+SUM(N).X
      SUM(N).Y=URN1*B(N1,N).Y+SUM(N).Y
      SUM(N).Z=URN1*B(N1,N).Z+SUM(N).Z
   END_LOOP
END_LOOP

R(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
```

```
   VRN=VR(N)
   R(I,J).X=SUM(N).X*VRN+R(I,J).X
   R(I,J).Y=SUM(N).Y*VRN+R(I,J).Y
   R(I,J).Z=SUM(N).Z*VRN+R(I,J).Z
 END_LOOP

 LOOP N=1 TO NOBV
   SUM(N)=[0.0,0.0,0.0]
   LOOP N1=1 TO NOBU
     DUN1=DU(N1)
     SUM(N).X=DUN1*B(N1,N).X+SUM(N).X
     SUM(N).Y=DUN1*B(N1,N).Y+SUM(N).Y
     SUM(N).Z=DUN1*B(N1,N).Z+SUM(N).Z
   END_LOOP
 END_LOOP

 RU(I,J)=[0.0,0.0,0.0]
 LOOP N=1 TO NOBV
   VRN=VR(N)
   RU(I,J).X=SUM(N).X*VRN+RU(I,J).X
   RU(I,J).Y=SUM(N).Y*VRN+RU(I,J).Y
   RU(I,J).Z=SUM(N).Z*VRN+RU(I,J).Z
 END_LOOP

 LOOP N=1 TO NOBV
   SUM(N)=[0.0,0.0,0.0]
   LOOP N1=1 TO NOBU
     URN1=UR(N1)
     SUM(N).X=URN1*B(N1,N).X+SUM(N).X
     SUM(N).Y=URN1*B(N1,N).Y+SUM(N).Y
     SUM(N).Z=URN1*B(N1,N).Z+SUM(N).Z
   END_LOOP
 END_LOOP

 RV(I,J)=[0.0,0.0,0.0]
 LOOP N=1 TO NOBV
   DVN=DV(N)
   RV(I,J).X=SUM(N).X*DVN+RV(I,J).X
   RV(I,J).Y=SUM(N).Y*DVN+RV(I,J).Y
   RV(I,J).Z=SUM(N).Z*DVN+RV(I,J).Z
 END_LOOP

 END PROC


 ---------------------------------------------------------------
 -- THIS IS THE MAIN PROGRAM WHICH GIVES CHOICES AND OUTPUT  --
 ---------------------------------------------------------------

                 PROC MAIN

 INTEGER TYPE,TYPE1,I,J,CNO,LNO
 INTEGER NOPU,NOPV,NOBU,NOBV,SOFF
 REAL H,C

 ACCEPT NOPV PROMPT(' NO OF OFFSET POINTS EACH ROW:   ')
 ACCEPT NOPU PROMPT(' NO OF OFFSET POINTS EACH COLUMN:   ')
 ACCEPT H PROMPT('offset value ( tool radius:  ')
 ACCEPT SOFF PROMPT('offset direction, 1: side A.  2: side B ')
 ACCEPT LNO PROMPT('layer no: ')

 ---------------------------------------------------

 BST:REDAT(NOPU,NOPV,NOBU,NOBV)       -----CALL REDAT

 CCUV(NOBU,NOBV)                      ------CALL CCUV
```

```
BA: ACCEPT TYPE PROMPT(' 1:CALCULATE  2:FINISH  3:START AGAIN  ?')

IF TYPE=1 THEN

LOOP I=1 TO NOPU
  LOOP J=1 TO NOPV

    UBV(I,J,NOPU,NOPV,NOBU,NOBV)
    RUV(I,J).X=RU(I,J).Y*RV(I,J).Z-RU(I,J).Z*RV(I,J).Y
    RUV(I,J).Y=RU(I,J).Z*RV(I,J).X-RU(I,J).X*RV(I,J).Z
    RUV(I,J).Z=RU(I,J).X*RV(I,J).Y-RU(I,J).Y*RV(I,J).X
    C=(RUV(I,J).X^2.0+RUV(I,J).Y^2.0+RUV(I,J).Z^2.0)^0.5

    IF SOFF=2 THEN

      P1(I,J).X=R(I,J).X+(H/C)*RUV(I,J).X
      P1(I,J).Y=R(I,J).Y+(H/C)*RUV(I,J).Y
      P1(I,J).Z=R(I,J).Z+(H/C)*RUV(I,J).Z

    ELSE IF SOFF=1 THEN

      P1(I,J).X=R(I,J).X-(H/C)*RUV(I,J).X
      P1(I,J).Y=R(I,J).Y-(H/C)*RUV(I,J).Y
      P1(I,J).Z=R(I,J).Z-(H/C)*RUV(I,J).Z

    END_IF

  END_LOOP
END_LOOP

SEND
SEND 'REPAINT'
SEND

LOOP I=1 TO NOPU
  SEND
  SEND 'SMO CPO DEG ',NOPV-1,':',
  LOOP J=1 TO NOPV
    SEND 'X',P1(I,J).X,'Y',P1(I,J).Y,'Z',P1(I,J).Z,',',
  END_LOOP
  SEND
END_LOOP

SEND
SEND 'REPAINT'
SEND

ELSE IF TYPE=2 THEN
  GO_TO BB

ELSE IF TYPE=3 THEN
  GO_TO BST

END_IF

ACCEPT TYPE1 PROMPT(' 1:do another surface  2:finish ')

IF TYPE1=1 THEN
  GO_TO BST
END_IF

BB:ECHO ON

END PROC
```

CUSPH.UPL

This program calculates the maximum scallop cusp height left on the finished surface of a part. It also selects the maximum toolpath interval for a given ball ended tool. The vertices of the control polyhedra of a surface model are input as follows.

```
------------------------------------------------------------------
--                                                              --
--                                                              --
--        (0,1)  3  * * * * ---> DIG IN THIS DIRECTION (ROWS)   --
--        u ^    2  * * * * --->                                --
--          |    1  * * * * --->                                --
--          |                                                   --
--          |--------> v (0,1)                                  --
--                                                              --
------------------------------------------------------------------


GROUP

REAL U(8),V(8)
COORD B(8,8),RU(8,8),RV(8,8),RUV(8,8)
COORD CL(8,8),SUM(8)
REAL CH(8,8)
INTEGER M,N,K

END GROUP
------------------------------------------------Group declarations


    PROC REDAT(INOUT INTEGER NOPU,NOPV,NOBU,NOBV;IN REAL US,UF,VS,VF)
    -----------------------------------------------------------------

INTEGER NP,NOP,K1,NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
COORD B1(8),B2(8),B3(8),B4(8),B5(8),B6(8),B7(8),B8(8)
--------------------
ACCEPT NOPU PROMPT('  NO. OF POINTS ALONG U:')
ACCEPT NOPV PROMPT('  NO. OF POINTS ALONG V:')

U(1)=US
U(NOPU)=UF
LOOP NP=2 TO NOPU-1
U(NP)=U(NP-1)+(UF-US)/(REAL(NOPU)-1.0)
END_LOOP
--------------------
V(1)=VS
V(NOPV)=VF
LOOP NOP=2 TO NOPV-1
V(NOP)=V(NOP-1)+(VF-VS)/(REAL(NOPV)-1.0)
END_LOOP
PRINT '  U:',U(1),'  ',U(2),'  ',U(3),'  ',U(4)
PRINT '  V:',V(1),'  ',U(2),'  ',V(3),'  ',U(4)
--------------------
PRINT 'dig the controlling points row by row'
PRINT 'give a <CR> for each row'
NOBV=0;  NEND2=0;  NEND3=0;  NEND4=0
NEND5=0;  NEND6=0;  NEND7=0;  NEND8=0
GETEND(8,1,NOBV,B1(1))
GETEND(8,1,NEND2,B2(1))
GETEND(8,1,NEND3,B3(1))
GETEND(8,1,NEND4,B4(1))
PRINT 'row 5:'
```

```
GETEND(8,1,NEND5,B5(1))
PRINT 'row 6:'
GETEND(8,1,NEND6,B6(1))
PRINT 'row 7:'
GETEND(8,1,NEND7,B7(1))
PRINT 'row 8:'
GETEND(8,1,NEND8,B8(1))

ACCEPT NOBU PROMPT('  no. of ROWS : ')

LOOP K1=1 TO NOBV
B(1,K1)=B1(K1);  B(2,K1)=B2(K1);  B(3,K1)=B3(K1);  B(4,K1)=B4(K1)
B(5,K1)=B5(K1);  B(6,K1)=B6(K1);  B(7,K1)=B7(K1);  B(8,K1)=B8(K1)
END_LOOP
---------------------

END PROC
---------------------------------------------------Input information

       PROC UBV(IN INTEGER I,J,NOPU,NOPV,NOBU,NOBV)
       -------------------------------------------

REAL UI,VJ,DU(8),UR(8),CN,CNP,CNPB,VR(8),DV(8),CCC,UUU1,UUU2
INTEGER K1,K2,J1,NP1,NP2,NP3,N1
REAL URN1,VRN,DUN1,DVN
------------------------
UI=U(I)
VJ=V(J)
PRINT '  I=',I,';  J=',J
-------------------------
IF UI<0.001 THEN
UR(1)=1.0
LOOP K1=2 TO NOBU
UR(K1)=0.0
END_LOOP
DU(1)=-REAL(NOBU-1)
DU(2)=REAL(NOBU-1)
LOOP K2=3 TO NOBU
DU(K2)=0.0
END_LOOP
-------------------------
ELSE IF (1.0-UI)<0.001 THEN
UR(NOBU)=1.0
LOOP K1=1 TO NOBU-1
UR(K1)=0.0
END_LOOP
DU(NOBU)=REAL(NOBU-1)
DU(NOBU-1)=-REAL(NOBU-1)
LOOP K2=1 TO NOBU-2
DU(K2)=0.0
END_LOOP
--------------------------
ELSE
CN=1.0
LOOP NP3=1 TO NOBU
--------
IF NP3=1 THEN
CN=1.0
ELSE
CN=CN*REAL(NP3-1)
END_IF
--------
END_LOOP

CNP=1.0
LOOP NP1=1 TO NOBU
```

```
IF NP1=1 THEN
CNP=1.0
ELSE
CNP=CNP*REAL(NP1-1)
END_IF
----------------------------

J1=NOBU-NP1
IF NP1=NOBU THEN
CNPB=1.0
ELSE
CNPB=1.0
LOOP NP2=1 TO J1
CNPB=CNPB*REAL(NP2)
END_LOOP
END_IF

CCC=CN/CNP/CNPB
UR(NP1)=CCC*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1)
UUU1=CCC*REAL(NP1-1)*UI^REAL(NP1-2)*(1.0-UI)^REAL(J1)
UUU2=CCC*REAL(J1)*UI^REAL(NP1-1)*(1.0-UI)^REAL(J1-1)
DU(NP1)=UUU1-UUU2
END_LOOP

END_IF

IF VJ<0.001 THEN
VR(1)=1.0
LOOP K1=2 TO NOBV
VR(K1)=0.0
END_LOOP
DV(1)=-REAL(NOBV-1)
DV(2)=REAL(NOBV-1)
LOOP K2=3 TO NOBV
DV(K2)=0.0
END_LOOP

ELSE IF (1.0-VJ)<0.001 THEN
VR(NOBV)=1.0
LOOP K1=1 TO NOBV-1
VR(K1)=0.0
END_LOOP
DV(NOBV)=REAL(NOBV-1)
DV(NOBV-1)=-REAL(NOBV-1)
LOOP K2=1 TO NOBV-2
DV(K2)=0.0
END_LOOP
ELSE

CN=1.0
LOOP NP3=1 TO NOBV
IF NP3=1 THEN
CN=1.0
ELSE
CN=CN*REAL(NP3-1)
END_IF
END_LOOP

CNP=1.0
LOOP NP1=1 TO NOBV
IF NP1=1 THEN
CNP=1.0
ELSE
CNP=CNP*REAL(NP1-1)
END_IF
```

```
J1=NOBV-NP1
IF NP1=NOBV THEN
CNPB=1.0
ELSE
CNPB=1.0
LOOP NP2=1 TO J1
CNPB=CNPB*REAL(NP2)
END_LOOP
END_IF

CCC=CN/CNP/CNPB
VR(NP1)=CCC*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1)
UUU1=CCC*REAL(NP1-1)*VJ^REAL(NP1-2)*(1.0-VJ)^REAL(J1)
UUU2=CCC*REAL(J1)*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(J1-1)
DV(NP1)=UUU1-UUU2

END_LOOP

END_IF

LOOP N=1 TO NOBV
SUM(N)=[0.0,0.0,0.0]
LOOP N1=1 TO NOBU
DUN1=DU(N1)
SUM(N).X=DUN1*B(N1,N).X+SUM(N).X
SUM(N).Y=DUN1*B(N1,N).Y+SUM(N).Y
SUM(N).Z=DUN1*B(N1,N).Z+SUM(N).Z
END_LOOP
END_LOOP

RU(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
VRN=VR(N)
RU(I,J).X=SUM(N).X*VRN+RU(I,J).X
RU(I,J).Y=SUM(N).Y*VRN+RU(I,J).Y
RU(I,J).Z=SUM(N).Z*VRN+RU(I,J).Z
END_LOOP

LOOP N=1 TO NOBV
SUM(N)=[0.0,0.0,0.0]
LOOP N1=1 TO NOBU
URN1=UR(N1)
SUM(N).X=URN1*B(N1,N).X+SUM(N).X
SUM(N).Y=URN1*B(N1,N).Y+SUM(N).Y
SUM(N).Z=URN1*B(N1,N).Z+SUM(N).Z
END_LOOP
END_LOOP

RV(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBV
DVN=DV(N)
RV(I,J).X=SUM(N).X*DVN+RV(I,J).X
RV(I,J).Y=SUM(N).Y*DVN+RV(I,J).Y
RV(I,J).Z=SUM(N).Z*DVN+RV(I,J).Z
END_LOOP

END PROC

-----------------------------------------------Calculations

                PROC MAIN
                ---------

INTEGER TYPE,I,J,NOPU,NOPV,NOBU,NOBV,DMORE
REAL H,C,D,L,C1,C2,GAP,US,UF,VS,VF
-----------------------------------------------------
```

```
 WINDOW 5,5,24,64,79
 DISP_WIN=5
 CLEAR 5

 DISPLAY

   This program
  calculate mini
  mum gap of two
  cuts for a giv
  en cusp hight.
  You can  only
  calculate a
  small area:
  (u1<= u =<u2,
  (v1<= v =<v2)

      ***u2***
  +u  ***u1***
  ^   ****v1v2
  |   >*******
  |--------> +v
 $

 -------------------------------------------------

 ACCEPT US PROMPT('  START u VALUE: ')
 ACCEPT UF PROMPT('  FINISH u VALUE: ')
 ACCEPT VS PROMPT('  START v VALUE: ')
 ACCEPT VF PROMPT('  FINISH v VALUE: ')
 ACCEPT H PROMPT(' CUSP HIGHT ALLOWED: ')
 ACCEPT D PROMPT('  DIAMETER OF THE CUTTER: ')
 GAP=1000.0
 -------------------------------------------------
 L=2.0*(D*H)^0.5
 BST:REDAT(NOPU,NOPV,NOBU,NOBV,US,UF,VS,VF)

 BA: ACCEPT TYPE PROMPT(' CONTINUE: 1  FINISH: 2 REINPUT: 3 ?  ')

 IF TYPE=1 THEN

 LOOP I=1 TO NOPV
 LOOP J=1 TO NOPU

 UBV(I,J,NOPU,NOPV,NOBU,NOBV)

 RUV(I,J).X=RU(I,J).Y*RV(I,J).Z-RU(I,J).Z*RV(I,J).Y
 RUV(I,J).Y=RU(I,J).Z*RV(I,J).X-RU(I,J).X*RV(I,J).Z
 RUV(I,J).Z=RU(I,J).X*RV(I,J).Y-RU(I,J).Y*RV(I,J).X

 PRINT '  RUV:',RUV(I,J)
 C1=RV(I,J).X*RU(I,J).Z-RU(I,J).X*RV(I,J).Z
 C2=RV(I,J).Y*RU(I,J).Z-RU(I,J).Y*RV(I,J).Z
 CL(I,J).X=-RUV(I,J).Z*C2
 CL(I,J).Y=RUV(I,J).Z*C1
 CL(I,J).Z=RUV(I,J).X*C2-RUV(I,J).Y*C1

 PRINT '  CL:',CL(I,J)
 C=(CL(I,J).X^2.0+CL(I,J).Y^2.0+CL(I,J).Z^2.0)^0.5
 CH(I,J)=ABS(L/C*CL(I,J).Z)
 PRINT '  CH:',CH(I,J)
 PRINT '  GAP:',GAP
 IF GAP>CH(I,J) THEN
 GAP=CH(I,J)
 END_IF
```

```
END_LOOP
END_LOOP

PRINT '  MIN GAP:',GAP

ACCEPT DMORE PROMPT('  1:DO ONE MORE  2:FINISH  ')

  IF DMORE=1 THEN
  GO_TO BST
  END_IF

ELSE IF TYPE=3 THEN
GO_TO BST

END_IF

BB:ECHO ON

END PROC
-----------------------------------------------------end
```

BEZTP.UPL

This program generates toolpath curves on the offset surfaces to the part surfaces. The vertices of the control polyhedra of a offset surface are input as follows.

```
-------------------------------------------------------------------
--                                                               --
--                                                               --
--          (0,1)  3   * * * * ---> DIG IN THIS DIRECTION (ROWS)  --
--          u ^    2   * * * * --->                               --
--            |    1   * * * * --->                               --
--            |                                                   --
--            |--------> v (0,1)                                  --
--                                                               --
-------------------------------------------------------------------


GROUP

REAL U(8),V(8),CCU(8),CCV(8),H(100)
COORD R(100,8),B(8,8)
INTEGER M,N,K,DIF(100)

END GROUP
------------------------------------------------Group declarations


PROC REDAT(IN INTEGER NOPV;INOUT INTEGER NOBU,NOBV)
----------------------------------------------------

INTEGER NP,NOP,K1,NEND1,NEND2,NEND3,NEND4,NEND5,NEND6,NEND7,NEND8
COORD B1(8),B2(8),B3(8),B4(8),B5(8),B6(8),B7(8),B8(8)
--------------------
V(1)=0.0
V(NOPV)=1.0
LOOP NOP=2 TO NOPV-1
V(NOP)=V(NOP-1)+1.0/(REAL(NOPV)-1.0)
END_LOOP
--------------------
PRINT 'dig the controlling points row by row'
PRINT 'give a <CR> for each row'
NOBV=0;  NEND2=0;  NEND3=0;  NEND4=0
NEND5=0;  NEND6=0;  NEND7=0;  NEND8=0
GETEND(8,1,NOBV,B1(1))
GETEND(8,1,NEND2,B2(1))
GETEND(8,1,NEND3,B3(1))
GETEND(8,1,NEND4,B4(1))
PRINT 'row 5:'
GETEND(8,1,NEND5,B5(1))
PRINT 'row 6:'
GETEND(8,1,NEND6,B6(1))
PRINT 'row 7:'
GETEND(8,1,NEND7,B7(1))
PRINT 'row 8:'
GETEND(8,1,NEND8,B8(1))

ACCEPT NOBU PROMPT('  no. of ROWS : ')

LOOP K1=1 TO NOBV
B(1,K1)=B1(K1);  B(2,K1)=B2(K1);  B(3,K1)=B3(K1);  B(4,K1)=B4(K1)
```

```
B(5,K1)=B5(K1);  B(6,K1)=B6(K1);  B(7,K1)=B7(K1);  B(8,K1)=B8(K1)
END_LOOP

END PROC
--------------------------------------------------Input information

       PROC CCUV(IN INTEGER NOBU,NOBV)
       -------------------------------
REAL CN,CNP,CNPB
INTEGER NP1,NP2,J1

CN=1.0
  LOOP NP1=2 TO NOBU
  CN=CN*REAL(NP1-1)
  END_LOOP

CNP=1.0
  LOOP NP1=1 TO NOBU
  IF NP1=1 THEN
    CNP=1.0
  ELSE
    CNP=CNP*REAL(NP1-1)
  END_IF

  J1=NOBU-NP1
  IF NP1=NOBU THEN
    CNPB=1.0
  ELSE
    CNPB=1.0
    LOOP NP2=1 TO J1
    CNPB=CNPB*REAL(NP2)
    END_LOOP
  END_IF

  CCU(NP1)=CN/CNP/CNPB
  END_LOOP

-------------------------------  CCU

CN=1.0
  LOOP NP1=2 TO NOBV
  CN=CN*REAL(NP1-1)
  END_LOOP

CNP=1.0
  LOOP NP1=1 TO NOBV
  IF NP1=1 THEN
    CNP=1.0
  ELSE
    CNP=CNP*REAL(NP1-1)
  END_IF

  J1=NOBV-NP1
  IF NP1=NOBV THEN
    CNPB=1.0
  ELSE
    CNPB=1.0
    LOOP NP2=1 TO J1
    CNPB=CNPB*REAL(NP2)
    END_LOOP
  END_IF

CCV(NP1)=CN/CNP/CNPB

  END_LOOP                --------CCV
```

```
END PROC

--------------------------------------------------

PROC RUBV(IN INTEGER I,J,NOPV,NOBU,NOBV,CNO,LNO,CXYZ;IN REAL TOL)
-----------------------------------------------------------------
COORD SUM(8)
REAL AH(8),UR(8),VR(8)
REAL VJ,CCC,UA,UB,UT,FA,FB,FT,UJ
INTEGER K1,NP1,I1,K2,NP2,N1

VJ=V(J)

IF J=1 THEN
  VR(1)=1.0
  LOOP K1=2 TO NOBV
  VR(K1)=0.0
  END_LOOP

ELSE IF J=NOPV THEN
  VR(NOBV)=1.0
  LOOP K1=1 TO NOBV-1
  VR(K1)=0.0
  END_LOOP

ELSE
  LOOP NP1=1 TO NOBV
  CCC=CCV(NP1)
  VR(NP1)=CCC*VJ^REAL(NP1-1)*(1.0-VJ)^REAL(NOBV-NP1)
  END_LOOP

END_IF


--------------------------CALCULATE VR

LOOP K1=1 TO NOBU
  AH(K1)=0.0
  LOOP NP1=1 TO NOBV
    IF CXYZ=1 THEN
       AH(K1)=VR(NP1)*B(K1,NP1).X+AH(K1)
    ELSE IF CXYZ=2 THEN
       AH(K1)=VR(NP1)*B(K1,NP1).Y+AH(K1)
    ELSE
       AH(K1)=VR(NP1)*B(K1,NP1).Z+AH(K1)
    END_IF
  END_LOOP
END_LOOP

---------------------------CALCULATE AY(NOBU)

----To calculate uj, use iteration   -------
-------------------------------------------

UA=0.0
UB=1.0
FA=AH(1)-H(I)
FB=AH(NOBU)-H(I)
IF ABS(FA)<0.0001 THEN
UJ=UA
ELSE IF ABS(FB)<0.0001 THEN
     UJ=UB
     ELSE IF FA*FB<0.0 THEN
          EE:UT=(UA+UB)/2.0
            LOOP NP1=1 TO NOBU
            CCC=CCU(NP1)
```

```
          UR(NP1)=CCC*UT^REAL(NP1-1)*(1.0-UT)^REAL(NOBU-NP1)
          END_LOOP
        FT=0.0
          LOOP NP1=1 TO NOBU
          FT=FT+AH(NP1)*UR(NP1)
          END_LOOP
          FT=FT-H(I)
          IF FT*FA<0.0 THEN
          FB=FT
          UB=UT
          ELSE
          FA=FT
          UA=UT
          END_IF
          IF ABS(FB-FA)>TOL THEN
          GO_TO EE
          END_IF

        UJ=(UA+UB)/2.0
        ELSE
        DIF(I)=2
        GO_TO DD
END_IF
------------------------------------uj
IF ABS(UJ)<0.0001 THEN
   UR(1)=1.0
   LOOP NP1=2 TO NOBU
   UR(NP1)=0.0
   END_LOOP
ELSE IF ABS(UJ-1.0)<0.0001 THEN
   UR(NOBU)=1.0
   LOOP NP1=1 TO NOBU-1
   UR(NP1)=0.0
   END_LOOP
ELSE
   LOOP NP1=1 TO NOBU
     CCC=CCU(NP1)
     UR(NP1)=CCC*UJ^REAL(NP1-1)*(1.0-UJ)^REAL(NOBU-NP1)
   END_LOOP
END_IF
---------------------------------------UR
LOOP N=1 TO NOBU
SUM(N)=[0.0,0.0,0.0]
  LOOP N1=1 TO NOBV
  SUM(N).X=VR(N1)*B(N,N1).X+SUM(N).X
  SUM(N).Y=VR(N1)*B(N,N1).Y+SUM(N).Y
  SUM(N).Z=VR(N1)*B(N,N1).Z+SUM(N).Z
  END_LOOP
END_LOOP

R(I,J)=[0.0,0.0,0.0]
LOOP N=1 TO NOBU
R(I,J).X=SUM(N).X*UR(N)+R(I,J).X
R(I,J).Y=SUM(N).Y*UR(N)+R(I,J).Y
R(I,J).Z=SUM(N).Z*UR(N)+R(I,J).Z
END_LOOP

IF CXYZ=1 THEN
   R(I,J).X=H(I)
ELSE IF CXYZ=2 THEN
   R(I,J).Y=H(I)
ELSE
   R(I,J).Z=H(I)
END_IF
PRINT 'R(I,J):',R(I,J)
INSERT POINT COLOR(CNO) LAYER(LNO) LOC(R(I,J))
```

```
GO_TO CC

DD:PRINT '   NO INTERSECTION'
GO_TO FF
CC:PRINT '   TOLERANCE :',TOL
FF:ECHO ON

END PROC

---------------------------------------------Calculations

                PROC MAIN
                ---------

INTEGER TYPE,I,J,CNO,LNO,NOPU,NOPV,MIL(100),NU
INTEGER PAT,NOBU,NOBV,DMORE,CXYZ,NENT,IEND
REAL H1,GAP,TOL
FILE F1
---------------------------------------------------
WINDOW 5,5,24,64,79
DISP_WIN=5
CLEAR 5

DISPLAY

   This program
 calculate tool
 paths on   the
 offset surface
 the   digitize
 direction   is
 shown below:

 'x' ----inside
 '.' ---outside

 +u   ****
 ^    ****
 |    >***
 |--------> +v
$

---------------------------------------------------
SEND
SEND 'SEL CPL  1'
SEND

ACCEPT PAT PROMPT(' PATCH NO:')
ACCEPT CXYZ PROMPT(' 1: CUT IN X AXIS; 2: Y; 3:Z ')
ACCEPT NOPV PROMPT(' NO OF POINTS USED FOR EACH CURVE ')
ACCEPT H1 PROMPT(' VALUE OF THE FIRST PLANE:')
ACCEPT GAP PROMPT(' THE GAP BETWEEN TWO ADJACENT PLANES :  ')
ACCEPT NOPU PROMPT(' NO. OF CUTS ALONG  THE AXIS:  ')
ACCEPT CNO PROMPT(' colour no')
ACCEPT LNO PROMPT('layer no')
ACCEPT TOL PROMPT(' tolerance:')
---------------------------------------------------

BST:OPEN F1 'PATCH.DAT'
WRITE F1,'PATCH NO: ',PAT
LOOP I=1 TO NOPU
H(I)=H1+REAL(I-1)*GAP
DIF(I)=1
WRITE F1,' CUT NO: ',I,'  CUT AXIS:',CXYZ,'  VALUE=',H(I)
END_LOOP
CLOSE F1
```

```
REDAT(NOPV,NOBU,NOBV)      ------CALL INPUT SUB

CCUV(NOBU,NOBV)                   ------CALL CCUV SUB

BA: ACCEPT TYPE PROMPT(' CONTINUE: 1  FINISH: 2 REINPUT: 3 ?  ')

IF TYPE=1 THEN

LOOP I=1 TO NOPU
LOOP J=1 TO NOPV

PRINT ' I=',I,' J=',J,' of (',NOPU,',',NOPV,')'

RUBV(I,J,NOPV,NOBU,NOBV,CNO,LNO,CXYZ,TOL)

END_LOOP
END_LOOP

SEND
SEND 'REPAINT'
SEND
LOOP I=1 TO NOPU
IF DIF(I)=1 THEN
   SEND 'SMO CPO DEG ',NOPV-1,':',
     LOOP J=1 TO NOPV
     SEND 'X',R(I,J).X,'Y',R(I,J).Y,'Z',R(I,J).Z,',',
     END_LOOP
   SEND
END_IF
END_LOOP
SEND 'REPAINT'
SEND

--ACCEPT NU PROMPT(' DIG THE SPO TO BE CHANGED POFF:') ENTANY NEWLINE
--SEND 'CHA SPO POFF: MIPTR',NU
--SEND 'REPAINT'
--SEND

--PRINT 'DIG THE CPOLES TO BE CHANGED POLYGON ON'
--NENT=0
--GETENT(100,NENT,MIL(1),IEND)
--LOOP I=1 TO NENT
--SEND
--SEND 'CHA CPO PON: MIPTR ',MIL(I)
--SEND
--END_LOOP

--SEND
--SEND 'REPAINT'
--SEND

ACCEPT DMORE PROMPT('  1:DO ONE MORE  2:FINISH  ')

  IF DMORE=1 THEN
  GO_TO BST
  END_IF

ELSE IF TYPE=3 THEN
GO_TO BST

END_IF

BB:ECHO ON

END PROC
```

--end

LINPATH.UPL

This program generates line segments from the curves created on the offset surfaces. The line are later used as toolpath geometry. The maximum chordal deviation is calculted and this determines the maximum length of the line segments.

```
--------------------------
---  Group delarition  ---
--------------------------

GROUP

REAL U(100),CCU(8),VAR1(100),VAR2(100)
COORD R(100),B(8)
INTEGER M,N,K

END GROUP
------------------------------------------------

       PROC REDAT(IN INTEGER NOPU;INOUT INTEGER NOBU)
       ----------------------------------------------

INTEGER NP
--------------------
U(1)=0.0
U(NOPU)=1.0
LOOP NP=2 TO NOPU-1
U(NP)=U(NP-1)+1.0/(REAL(NOPU)-1.0)
END_LOOP
--------------------
PRINT 'dig the controlling points'
NOBU=0
GETEND(8,1,NOBU,B(1))

END PROC
---------------------------------------------------Input information

       PROC CCUV(IN INTEGER NOBU)
       --------------------------
REAL CN,CNP,CNPB
INTEGER NP1,NP2,J1

CN=1.0
  LOOP NP1=2 TO NOBU
  CN=CN*REAL(NP1-1)
  END_LOOP

CNP=1.0
  LOOP NP1=1 TO NOBU
    IF NP1=1 THEN
    CNP=1.0
    ELSE
    CNP=CNP*REAL(NP1-1)
    END_IF

  J1=NOBU-NP1
    IF NP1=NOBU THEN
    CNPB=1.0
```

```
      ELSE
      CNPB=1.0
      LOOP NP2=1 TO J1
      CNPB=CNPB*REAL(NP2)
      END_LOOP
      END_IF

   CCU(NP1)=CN/CNP/CNPB
   END_LOOP

------------------------------------CALCULATE CCU

END PROC

       PROC UBV(IN INTEGER I,NOPU,NOBU,OXYZ)
       -------------------------------------

REAL UI,UR(8)
INTEGER K1
------------------------
UI=U(I)
PRINT '  I=',I
---------------------------CALCULATE UR
IF I=1 THEN
   UR(1)=1.0
   LOOP K1=2 TO NOBU
   UR(K1)=0.0
   END_LOOP
ELSE IF I=NOPU THEN
   UR(NOBU)=1.0
   LOOP K1=1 TO NOBU-1
   UR(K1)=0.0
   END_LOOP
ELSE
   LOOP K1=1 TO NOBU
   UR(K1)=CCU(K1)*UI^REAL(K1-1)*(1.0-UI)^REAL(NOBU-K1)
   END_LOOP
END_IF
------------------------------CALCULATE R

R(I)=[0.0,0.0,0.0]
IF OXYZ=1 THEN

  LOOP K1=1 TO NOBU
  R(I).X=B(1).X
  R(I).Y=R(I).Y+UR(K1)*B(K1).Y
  R(I).Z=R(I).Z+UR(K1)*B(K1).Z
  END_LOOP

  VAR1(I)=R(I).Y
  VAR2(I)=R(I).Z

ELSE IF OXYZ=2 THEN

  LOOP K1=1 TO NOBU
  R(I).X=R(I).X+UR(K1)*B(K1).X
  R(I).Y=B(1).Y
  R(I).Z=R(I).Z+UR(K1)*B(K1).Z
  END_LOOP

  VAR1(I)=R(I).Z
  VAR2(I)=R(I).X

ELSE

  LOOP K1=1 TO NOBU
```

```
   R(I).X=R(I).X+UR(K1)*B(K1).X
   R(I).Y=R(I).Y+UR(K1)*B(K1).Y
   R(I).Z=B(1).Z
   END_LOOP

   VAR1(I)=R(I).X
   VAR2(I)=R(I).Y

 END_IF

 END PROC

 ------------------------------------------------

      PROC ERR(IN INTEGER I,NOPU,NOBU,OXYZ;INOUT REAL DEV)
      -----------------------------------------------------

 REAL UI,DU(8),DDU(8),U1,U2,U3,U4,U5,DR1,DR2,DDR1,DDR2,LB,RERR
 INTEGER K1,K2

 UI=U(I)
 ---------------------------------------CAL DU,DDU
 IF I=1 THEN
    DU(1)=-REAL(NOBU-1)
    DU(2)=REAL(NOBU-1)
    LOOP K2=3 TO NOBU
    DU(K2)=0.0
    END_LOOP

    DDU(1)=REAL((NOBU-1)*(NOBU-2))
    DDU(2)=-2.0*REAL(NOBU-2)
    DDU(3)=2.0
    LOOP K1=4 TO NOBU
    DDU(K1)=0.0
    END_LOOP
 ELSE IF I=NOBU THEN
    DU(NOBU)=REAL(NOBU-1)
    DU(NOBU-1)=-REAL(NOBU-1)
    LOOP K2=1 TO NOBU-2
    DU(K2)=0.0
    END_LOOP

    DDU(NOBU)=REAL((NOBU-1)*(NOBU-2))
    DDU(NOBU-1)=-2.0*REAL(NOBU-2)
    DDU(NOBU-2)=2.0
    LOOP K1=1 TO NOBU-3
    DDU(K1)=0.0
    END_LOOP
 ELSE
    LOOP K1=1 TO NOBU
    U1=REAL(K1-1)*UI^REAL(K1-2)*(1.0-UI)^REAL(NOBU-K1)
    U2=REAL(NOBU-K1)*UI^REAL(K1-1)*(1.0-UI)^REAL(NOBU-K1-1)
    DU(K1)=CCU(K1)*(U1-U2)

    U3=REAL((K1-1)*(K1-2))*UI^REAL(K1-3)*(1.0-UI)^REAL(NOBU-K1)
    U4=-2.0*REAL((K1-1)*(NOBU-K1))*UI^REAL(K1-2)*(1.0-UI)^REAL(NOBU-K1-1)
    U5=REAL((NOBU-K1)*(NOBU-K1-1))*UI^REAL(K1-1)*(1.0-UI)^REAL(NOBU-K1-2)
    DDU(K1)=CCU(K1)*(U3+U4+U5)
    END_LOOP

 END_IF

 ---------------------------CAL ERROR

 DR1=0.0
 DDR1=0.0
```

```
 DR2=0.0
 DDR2=0.0

 IF OXYZ=1 THEN
    LOOP K1=1 TO NOBU
    DR1=DR1+DU(K1)*B(K1).Y
    DR2=DR2+DU(K1)*B(K1).Z
    DDR1=DDR1+DDU(K1)*B(K1).Y
    DDR2=DDR2+DDU(K1)*B(K1).Z
    END_LOOP
 ELSE IF OXYZ=2 THEN
    LOOP K1=1 TO NOBU
    DR1=DR1+DU(K1)*B(K1).Z
    DR2=DR2+DU(K1)*B(K1).X
    DDR1=DDR1+DDU(K1)*B(K1).Z
    DDR2=DDR2+DDU(K1)*B(K1).X
    END_LOOP
 ELSE
    LOOP K1=1 TO NOBU
    DR1=DR1+DU(K1)*B(K1).X
    DR2=DR2+DU(K1)*B(K1).Y
    DDR1=DDR1+DDU(K1)*B(K1).X
    DDR2=DDR2+DDU(K1)*B(K1).Y
    END_LOOP
 END_IF

 LB=((VAR1(I+1)-VAR1(I))^2.0+(VAR2(I+1)-VAR2(I))^2.0)^0.5

 RERR=ABS(DR1^3.0)*(1.0+(DR2/DR1)^2.0)^0.6667/ABS(DDR2*DR1-DR2*DDR1)

 DEV=RERR-(RERR^2.0-(LB/2.0)^2.0)^0.5

 END PROC

 ----------------------------------------------Calculations

                 PROC MAIN
                 ---------

 INTEGER TYPE,I,CNO,LNO,NOPU,NOBU,OXYZ,ER,BOS
 REAL DEV,MDEV
 ---------------------------------------------------
 SEND
 SEND 'SEL CPL  1'
 SEND

 BST:ACCEPT CNO PROMPT(' colour no')
 ACCEPT LNO PROMPT('layer no')
 ACCEPT OXYZ PROMPT('CURVE IN WHICH PLANE 1:YOZ  2:ZOX  3:XOY')
 ---------------------------------------------------

 DST:ACCEPT NOPU PROMPT(' NO. OF LINE SEGMENTS:  ')
 NOPU=NOPU+1

 REDAT(NOPU,NOBU)

 CCUV(NOBU)                      ------CALL CCUV

 BA:ACCEPT TYPE PROMPT(' 1:CALCULATE  2: FINISH  3: START AGAIN  ? ')

 IF TYPE=1 THEN

 LOOP I=1 TO NOPU

   UBV(I,NOPU,NOBU,OXYZ)
```

```
END_LOOP

ACCEPT ER PROMPT(' 1:CALCULATE ERROR  2:DO NOT CALCULATE')
IF ER=1 THEN
MDEV=0.0
LOOP I=1 TO NOPU-1
  ERR(I,NOPU,NOBU,OXYZ,DEV)
  IF DEV>MDEV THEN
     MDEV=DEV
  END_IF
END_LOOP

PRINT 'MAX DEV :',MDEV
ACCEPT BOS PROMPT(' 1: DEV ACCEPTED   2:DECREASED DEV')

IF BOS=2 THEN
GO_TO DST
END_IF

END_IF

LOOP I=1 TO NOPU-1
INSERT LINE COLOR(CNO) LAYER(LNO) ENDS(R(I+1),R(I))
END_LOOP

SEND
SEND 'REPAINT'
SEND

GO_TO BA

ELSE IF TYPE=2 THEN
GO_TO BB

ELSE IF TYPE=3 THEN
GO_TO BST

ENDIF

BB:ECHO ON

END PROC
--------------------------------end
```

Appendix Five

SOME TYPICAL NC PART PROGRAMS

(.NC1)

GENERATED BY THE NC PROCESSOR

BR1.NC1 - A typical Part Program used for the roughing of the bottle mould

```
prog BR1  v3       5/31/1989
001..BEGIN/SUB1
002..LXYXY/L1 X-12 Y225.4 X-41 Y216.49
003..LXYXY/L2 X-41 Y216.49 X-42.48 Y193.85
004..LXYXY/L3 X-42.48 Y193.85 X-42.13 Y188.28
005..LXYXY/L4 X-42.13 Y188.28 X-35.7 Y169.74
006..LXYXY/L5 X-35.7 Y169.74 X-34.74 Y163.61
007..LXYXY/L6 X-34.74 Y163.61 X-35.34 Y143.62
008..LXYXY/L7 X-35.34 Y143.62 X-37.09 Y71.92
009..LXYXY/L8 X-37.09 Y71.92 X-37.59 Y47.1
010..LXYXY/L9 X-37.59 Y47.1 X-39.12 Y41.98
011..LXYXY/L10 X-39.12 Y41.98 X-41.14 Y38.69
012..LXYXY/L11 X-41.14 Y38.69 X-42.57 Y33.37
013..LXYXY/L12 X-42.57 Y33.37 X-42.74 Y0
014..LXYXY/L13 X12 Y225.4 X41 Y216.49
015..LXYXY/L14 X41 Y216.49 X42.48 Y193.85
016..LXYXY/L15 X42.48 Y193.85 X42.13 Y188.28
017..LXYXY/L16 X42.13 Y188.28 X35.7 Y169.74
018..LXYXY/L17 X35.7 Y169.74 X34.74 Y163.61
019..LXYXY/L18 X34.74 Y163.61 X35.34 Y143.62
020..LXYXY/L19 X35.34 Y143.62 X37.09 Y71.92
021..LXYXY/L20 X37.09 Y71.92 X37.59 Y47.1
022..LXYXY/L21 X37.59 Y47.1 X39.12 Y41.98
023..LXYXY/L22 X39.12 Y41.98 X41.14 Y38.69
024..LXYXY/L23 X41.14 Y38.69 X42.57 Y33.37
025..LXYXY/L24 X42.57 Y33.37 X42.74 Y0
026..LXYXY/L25 X42.74 Y0 X-42.74 Y0
027..LXYXY/L26 X0 Y225.4 X-12 Y225.4
028..PXY/P1 X0.41 Y211.47
029..LXYXY/L27 X12 Y225.4 X0 Y225.4
030..PXY/P2 X-2.12 Y211.39
031..PXY/P3 X4.78 Y211.13
032..GROUP/GP2
033..GROUP/L26 L1 L2 L3 L4 L5 L6
034..GROUP/L7 L8 L9 L10 L11 L12 L25
035..GROUP/L24 L23 L22 L21 L20 L19 L18
036..GROUP/L17 L16 L15 L14 L13 L27
037..GROUP/GP3
038..END/
039..CLEARP/Z20
040..TLCHG/X0 Y280 T6 RPM3000 F500
041..TOOL/LR0 DIA16
042..MOVE/X0 Y211 Z20
043..FEDTO/X0 Y211 Z10
044..FEDTO/X0 Y211 Z-5
045..CALL/SUB1 FR0 MX0 MY0
046..FEDTO/X0 Y211 Z20
```

BD1.NC1 - A typical Part Program used for the finishing of the bottle mould

```
prog BD1  v5      5/31/1989
001..BEGIN/SUB1
002..TRANSL/ANG-90 ANG0 ANG0 D0 D5 D0
003..LXYXY/L1 X-37.72 Y0 X-37.72 Y-4.64
004..LXYXY/L2 X-36.29 Y-7.19 X-34.08 Y-8.5
005..LXYXY/L3 X-34.08 Y-8.5 X-31.79 Y-9.74
006..LXYXY/L4 X-31.79 Y-9.74 X-29.41 Y-10.9
007..LXYXY/L5 X-29.41 Y-10.9 X-26.95 Y-11.97
008..LXYXY/L6 X-26.95 Y-11.97 X-24.43 Y-12.97
009..LXYXY/L7 X-24.43 Y-12.97 X-21.85 Y-13.87
010..LXYXY/L8 X-21.85 Y-13.87 X-19.22 Y-14.68
011..LXYXY/L9 X-19.22 Y-14.68 X-16.55 Y-15.4
012..LXYXY/L10 X-16.55 Y-15.4 X-13.84 Y-16.01
013..LXYXY/L11 X-13.84 Y-16.01 X-11.1 Y-16.52
014..LXYXY/L12 X-11.1 Y-16.52 X-8.33 Y-16.92
015..LXYXY/L13 X-8.33 Y-16.92 X-5.56 Y-17.2
016..LXYXY/L14 X-5.56 Y-17.2 X-2.78 Y-17.37
017..LXYXY/L15 X-37.72 Y-4.64 X-37.66 Y-5.22
018..LXYXY/L16 X-37.66 Y-5.22 X-37.52 Y-5.74
019..LXYXY/L17 X-37.52 Y-5.74 X-37.27 Y-6.23
020..LXYXY/L18 X-37.27 Y-6.23 X-36.87 Y-6.71
021..LXYXY/L19 X-36.87 Y-6.71 X-36.29 Y-7.19
022..LXYXY/L20 X37.72 Y0 X37.72 Y-4.64
023..LXYXY/L21 X36.29 Y-7.19 X34.08 Y-8.5
024..LXYXY/L22 X34.08 Y-8.5 X31.79 Y-9.74
025..LXYXY/L23 X31.79 Y-9.74 X29.41 Y-10.9
026..LXYXY/L24 X29.41 Y-10.9 X26.95 Y-11.97
027..LXYXY/L25 X26.95 Y-11.97 X24.43 Y-12.97
028..LXYXY/L26 X24.43 Y-12.97 X21.85 Y-13.87
029..LXYXY/L27 X21.85 Y-13.87 X19.22 Y-14.68
030..LXYXY/L28 X19.22 Y-14.68 X16.55 Y-15.4
031..LXYXY/L29 X16.55 Y-15.4 X13.84 Y-16.01
032..LXYXY/L30 X13.84 Y-16.01 X11.1 Y-16.52
033..LXYXY/L31 X11.1 Y-16.52 X8.33 Y-16.92
034..LXYXY/L32 X8.33 Y-16.92 X5.56 Y-17.2
035..LXYXY/L33 X5.56 Y-17.2 X2.78 Y-17.37
036..LXYXY/L34 X2.78 Y-17.37 X0 Y-17.41
037..LXYXY/L35 X37.72 Y-4.64 X37.66 Y-5.22
038..LXYXY/L36 X37.66 Y-5.22 X37.52 Y-5.74
039..LXYXY/L37 X37.52 Y-5.74 X37.27 Y-6.23
040..LXYXY/L38 X37.27 Y-6.23 X36.87 Y-6.71
041..LXYXY/L39 X36.87 Y-6.71 X36.29 Y-7.19
042..LXYXY/L40 X0 Y-17.41 X-2.78 Y-17.37
043..PXY/P1 X-27.41 Y2.92
044..PXY/P2 X27.89 Y1.95
045..GROUP/GP1
046..GROUP/L1 L15 L16 L17 L18 L19 L2
047..GROUP/L3 L4 L5 L6 L7 L8 L9
048..GROUP/L10 L11 L12 L13 L14 L40 L34
049..GROUP/L33 L32 L31 L30 L29 L28 L27
```

```
050..GROUP/L26 L25 L24 L23 L22 L21 L39
051..GROUP/L38 L37 L36 L35 L20
052..GROUP/GP2
053..END/
054..RPLANE/R20
055..TLCHG/X0 Y280 T2 RPM3000 F500
056..TOOL/LR0 DIA0
057..MOVE/X0 Y280 Z20
058..MOVE/X0 Y5 Z20
059..FEDTO/X0 Y5 Z0
060..CALL/SUB1 FR0 MX0 MY0
061..BEGIN/SUB2
062..TRANSL/ANG-90 ANG0 ANG0 D0 D7 D0
063..LXYXY/L41 X-37.72 Y0 X-37.72 Y-4.55
064..LXYXY/L42 X37.72 Y0 X37.72 Y-4.55
065..LXYXY/L43 X-37.72 Y-4.55 X-37.66 Y-5.13
066..LXYXY/L44 X-37.66 Y-5.13 X-37.52 Y-5.65
067..LXYXY/L45 X-37.52 Y-5.65 X-37.27 Y-6.15
068..LXYXY/L46 X-37.27 Y-6.15 X-36.88 Y-6.62
069..LXYXY/L47 X-36.88 Y-6.62 X-36.29 Y-7.11
070..LXYXY/L48 X-36.29 Y-7.11 X-34.08 Y-8.42
071..LXYXY/L49 X-34.08 Y-8.42 X-31.79 Y-9.65
072..LXYXY/L50 X-31.79 Y-9.65 X-29.41 Y-10.82
073..LXYXY/L51 X-29.41 Y-10.82 X-26.95 Y-11.9
074..LXYXY/L52 X-26.95 Y-11.9 X-24.43 Y-12.89
075..LXYXY/L53 X-24.43 Y-12.89 X-21.85 Y-13.8
076..LXYXY/L54 X-21.85 Y-13.8 X-19.22 Y-14.61
077..LXYXY/L55 X-19.22 Y-14.61 X-16.55 Y-15.32
078..LXYXY/L56 X-16.55 Y-15.32 X-13.84 Y-15.94
079..LXYXY/L57 X-13.84 Y-15.94 X-11.1 Y-16.45
080..LXYXY/L58 X-11.1 Y-16.45 X-8.33 Y-16.84
081..LXYXY/L59 X-8.33 Y-16.84 X-5.56 Y-17.13
082..LXYXY/L60 X-5.56 Y-17.13 X-2.78 Y-17.3
083..LXYXY/L61 X-2.78 Y-17.3 X0 Y-17.34
084..LXYXY/L62 X0 Y-17.34 X2.78 Y-17.3
085..LXYXY/L63 X2.78 Y-17.3 X5.56 Y-17.13
086..LXYXY/L64 X5.56 Y-17.13 X8.33 Y-16.84
087..LXYXY/L65 X8.33 Y-16.84 X11.1 Y-16.45
088..LXYXY/L66 X11.1 Y-16.45 X13.84 Y-15.94
089..LXYXY/L67 X13.84 Y-15.94 X16.55 Y-15.32
090..LXYXY/L68 X16.55 Y-15.32 X19.22 Y-14.61
091..LXYXY/L69 X19.22 Y-14.61 X21.85 Y-13.8
092..LXYXY/L70 X21.85 Y-13.8 X24.43 Y-12.89
093..LXYXY/L71 X24.43 Y-12.89 X26.95 Y-11.9
094..LXYXY/L72 X26.95 Y-11.9 X29.41 Y-10.82
095..LXYXY/L73 X29.41 Y-10.82 X31.79 Y-9.65
096..LXYXY/L74 X31.79 Y-9.65 X34.08 Y-8.42
097..LXYXY/L75 X34.08 Y-8.42 X36.29 Y-7.11
098..LXYXY/L76 X37.72 Y-4.55 X37.66 Y-5.13
099..LXYXY/L77 X37.66 Y-5.13 X37.52 Y-5.65
100..LXYXY/L78 X37.52 Y-5.65 X37.27 Y-6.15
101..LXYXY/L79 X37.27 Y-6.15 X36.88 Y-6.62
102..LXYXY/L80 X36.88 Y-6.62 X36.29 Y-7.11
103..PXY/P3 X28.97 Y1.93
```

```
104..PXY/P4 X-28.25 Y3.2
105..GROUP/GP3
106..GROUP/L42 L76 L77 L78 L79 L80 L75
107..GROUP/L74 L73 L72 L71 L70 L69 L68
108..GROUP/L67 L66 L65 L64 L63 L62 L61
109..GROUP/L60 L59 L58 L57 L56 L55 L54
110..GROUP/L53 L52 L51 L50 L49 L48 L47
111..GROUP/L46 L45 L44 L43 L41
112..GROUP/GP4
113..END/
114..CALL/SUB2 FR0 MX0 MY0
115..BEGIN/SUB3
116..TRANSL/ANG-90 ANG0 ANG0 D0 D9 D0
117..LXYXY/L81 X-37.72 Y0 X-37.72 Y-4.47
118..LXYXY/L82 X-37.72 Y-4.47 X-37.66 Y-5.04
119..LXYXY/L83 X-37.66 Y-5.04 X-37.52 Y-5.57
120..LXYXY/L84 X-37.52 Y-5.57 X-37.27 Y-6.06
121..LXYXY/L85 X-37.27 Y-6.06 X-36.88 Y-6.54
122..LXYXY/L86 X-36.88 Y-6.54 X-36.29 Y-7.02
123..LXYXY/L87 X-36.29 Y-7.02 X-34.09 Y-8.33
124..LXYXY/L88 X-34.09 Y-8.33 X-31.79 Y-9.57
125..LXYXY/L89 X-31.79 Y-9.57 X-29.41 Y-10.74
126..LXYXY/L90 X-29.41 Y-10.74 X-26.95 Y-11.82
127..LXYXY/L91 X-26.95 Y-11.82 X-24.43 Y-12.81
128..LXYXY/L92 X-24.43 Y-12.81 X-21.85 Y-13.72
129..LXYXY/L93 X-21.85 Y-13.72 X-19.22 Y-14.53
130..LXYXY/L94 X-19.22 Y-14.53 X-16.55 Y-15.25
131..LXYXY/L95 X-16.55 Y-15.25 X-13.84 Y-15.86
132..LXYXY/L96 X-13.84 Y-15.86 X-11.1 Y-16.37
133..LXYXY/L97 X-11.1 Y-16.37 X-8.33 Y-16.77
134..LXYXY/L98 X-8.33 Y-16.77 X-5.56 Y-17.06
135..LXYXY/L99 X-5.56 Y-17.06 X-2.78 Y-17.22
136..LXYXY/L100 X37.72 Y0 X37.72 Y-4.47
137..LXYXY/L101 X37.72 Y-4.47 X37.66 Y-5.04
138..LXYXY/L102 X37.66 Y-5.04 X37.52 Y-5.57
139..LXYXY/L103 X37.52 Y-5.57 X37.27 Y-6.06
140..LXYXY/L104 X37.27 Y-6.06 X36.88 Y-6.54
141..LXYXY/L105 X36.88 Y-6.54 X36.29 Y-7.02
142..LXYXY/L106 X36.29 Y-7.02 X34.09 Y-8.33
143..LXYXY/L107 X34.09 Y-8.33 X31.79 Y-9.57
144..LXYXY/L108 X31.79 Y-9.57 X29.41 Y-10.74
145..LXYXY/L109 X29.41 Y-10.74 X26.95 Y-11.82
146..LXYXY/L110 X26.95 Y-11.82 X24.43 Y-12.81
147..LXYXY/L111 X24.43 Y-12.81 X21.85 Y-13.72
148..LXYXY/L112 X21.85 Y-13.72 X19.22 Y-14.53
149..LXYXY/L113 X19.22 Y-14.53 X16.55 Y-15.25
150..LXYXY/L114 X16.55 Y-15.25 X13.84 Y-15.86
151..LXYXY/L115 X13.84 Y-15.86 X11.1 Y-16.37
152..LXYXY/L116 X11.1 Y-16.37 X8.33 Y-16.77
153..LXYXY/L117 X8.33 Y-16.77 X5.56 Y-17.06
154..LXYXY/L118 X5.56 Y-17.06 X2.78 Y-17.22
155..LXYXY/L119 X2.78 Y-17.22 X0 Y-17.27
156..LXYXY/L120 X0 Y-17.27 X-2.78 Y-17.22
157..PXY/P5 X-28.49 Y2.45
```

```
158..PXY/P6 X28.79 Y2.15
159..GROUP/GP5
160..GROUP/L81 L82 L83 L84 L85 L86 L87
161..GROUP/L88 L89 L90 L91 L92 L93 L94
162..GROUP/L95 L96 L97 L98 L99 L120 L119
163..GROUP/L118 L117 L116 L115 L114 L113 L112
164..GROUP/L111 L110 L109 L108 L107 L106 L105
165..GROUP/L104 L103 L102 L101 L100
166..GROUP/GP6
167..END/
168..CALL/SUB3 FR0 MX0 MY0
169..BEGIN/SUB4
170..TRANSL/ANG-90 ANG0 ANG0 D0 D11 D0
171..LXYXY/L121 X-37.72 Y0 X-37.72 Y-4.38
172..LXYXY/L122 X-37.72 Y-4.38 X-37.66 Y-4.96
173..LXYXY/L123 X-37.66 Y-4.96 X-37.52 Y-5.48
174..LXYXY/L124 X-37.52 Y-5.48 X-37.27 Y-5.97
175..LXYXY/L125 X-37.27 Y-5.97 X-36.88 Y-6.45
176..LXYXY/L126 X-36.88 Y-6.45 X-36.3 Y-6.93
177..LXYXY/L127 X-36.3 Y-6.93 X-34.09 Y-8.25
178..LXYXY/L128 X-34.09 Y-8.25 X-31.79 Y-9.49
179..LXYXY/L129 X-31.79 Y-9.49 X-29.41 Y-10.66
180..LXYXY/L130 X-29.41 Y-10.66 X-26.95 Y-11.74
181..LXYXY/L131 X-26.95 Y-11.74 X-24.43 Y-12.74
182..LXYXY/L132 X-24.43 Y-12.74 X-21.85 Y-13.64
183..LXYXY/L133 X-21.85 Y-13.64 X-19.22 Y-14.46
184..LXYXY/L134 X-19.22 Y-14.46 X-16.55 Y-15.17
185..LXYXY/L135 X-16.55 Y-15.17 X-13.84 Y-15.79
186..LXYXY/L136 X-13.84 Y-15.79 X-11.1 Y-16.3
187..LXYXY/L137 X-11.1 Y-16.3 X-8.33 Y-16.7
188..LXYXY/L138 X-8.33 Y-16.7 X-5.56 Y-16.98
189..LXYXY/L139 X-5.56 Y-16.98 X-2.78 Y-17.15
190..LXYXY/L140 X37.72 Y0 X37.72 Y-4.38
191..LXYXY/L141 X37.72 Y-4.38 X37.66 Y-4.96
192..LXYXY/L142 X37.66 Y-4.96 X37.52 Y-5.48
193..LXYXY/L143 X37.52 Y-5.48 X37.27 Y-5.97
194..LXYXY/L144 X37.27 Y-5.97 X36.88 Y-6.45
195..LXYXY/L145 X36.88 Y-6.45 X36.3 Y-6.93
196..LXYXY/L146 X36.3 Y-6.93 X34.09 Y-8.25
197..LXYXY/L147 X34.09 Y-8.25 X31.79 Y-9.49
198..LXYXY/L148 X31.79 Y-9.49 X29.41 Y-10.66
199..LXYXY/L149 X29.41 Y-10.66 X26.95 Y-11.74
200..LXYXY/L150 X26.95 Y-11.74 X24.43 Y-12.74
201..LXYXY/L151 X24.43 Y-12.74 X21.85 Y-13.64
202..LXYXY/L152 X21.85 Y-13.64 X19.22 Y-14.46
203..LXYXY/L153 X19.22 Y-14.46 X16.55 Y-15.17
204..LXYXY/L154 X16.55 Y-15.17 X13.84 Y-15.79
205..LXYXY/L155 X13.84 Y-15.79 X11.1 Y-16.3
206..LXYXY/L156 X11.1 Y-16.3 X8.33 Y-16.7
207..LXYXY/L157 X8.33 Y-16.7 X5.56 Y-16.98
208..LXYXY/L158 X5.56 Y-16.98 X2.78 Y-17.15
209..LXYXY/L159 X2.78 Y-17.15 X0 Y-17.2
210..LXYXY/L160 X0 Y-17.2 X-2.78 Y-17.15
211..PXY/P7 X28.43 Y2.31
```

```
212..PXY/P8 X-28.43 Y3.57
213..GROUP/GP7
214..GROUP/L140 L141 L142 L143 L144 L145 L146
215..GROUP/L147 L148 L149 L150 L151 L152 L153
216..GROUP/L154 L155 L156 L157 L158 L159 L160
217..GROUP/L139 L138 L137 L136 L135 L134 L133
218..GROUP/L132 L131 L130 L129 L128 L127 L126
219..GROUP/L125 L124 L123 L122 L121
220..GROUP/GP8
221..END/
222..CALL/SUB4 FR0 MX0 MY0
223..BEGIN/SUB5
224..TRANSL/ANG-90 ANG0 ANG0 D0 D13 D0
225..LXYXY/L161 X-37.72 Y0 X-37.72 Y-4.3
226..LXYXY/L162 X-37.72 Y-4.3 X-37.66 Y-4.87
227..LXYXY/L163 X-37.66 Y-4.87 X-37.52 Y-5.39
228..LXYXY/L164 X-37.52 Y-5.39 X-37.28 Y-5.89
229..LXYXY/L165 X-37.28 Y-5.89 X-36.88 Y-6.37
230..LXYXY/L166 X-36.88 Y-6.37 X-36.3 Y-6.85
231..LXYXY/L167 X-36.3 Y-6.85 X-34.09 Y-8.16
232..LXYXY/L168 X-34.09 Y-8.16 X-31.79 Y-9.41
233..LXYXY/L169 X-31.79 Y-9.41 X-29.41 Y-10.58
234..LXYXY/L170 X-29.41 Y-10.58 X-26.95 Y-11.66
235..LXYXY/L171 X-26.95 Y-11.66 X-24.43 Y-12.66
236..LXYXY/L172 X-24.43 Y-12.66 X-21.85 Y-13.57
237..LXYXY/L173 X-21.85 Y-13.57 X-19.22 Y-14.38
238..LXYXY/L174 X-19.22 Y-14.38 X-16.55 Y-15.1
239..LXYXY/L175 X-16.55 Y-15.1 X-13.83 Y-15.72
240..LXYXY/L176 X-13.83 Y-15.72 X-11.1 Y-16.22
241..LXYXY/L177 X-11.1 Y-16.22 X-8.33 Y-16.62
242..LXYXY/L178 X-8.33 Y-16.62 X-5.56 Y-16.91
243..LXYXY/L179 X-5.56 Y-16.91 X-2.78 Y-17.08
244..LXYXY/L180 X37.72 Y0 X37.72 Y-4.3
245..LXYXY/L181 X37.72 Y-4.3 X37.66 Y-4.87
246..LXYXY/L182 X37.66 Y-4.87 X37.52 Y-5.39
247..LXYXY/L183 X37.52 Y-5.39 X37.28 Y-5.89
248..LXYXY/L184 X37.28 Y-5.89 X36.88 Y-6.37
249..LXYXY/L185 X36.88 Y-6.37 X36.3 Y-6.85
250..LXYXY/L186 X36.3 Y-6.85 X34.09 Y-8.16
251..LXYXY/L187 X34.09 Y-8.16 X31.79 Y-9.41
252..LXYXY/L188 X31.79 Y-9.41 X29.41 Y-10.58
253..LXYXY/L189 X29.41 Y-10.58 X26.95 Y-11.66
254..LXYXY/L190 X26.95 Y-11.66 X24.43 Y-12.66
255..LXYXY/L191 X24.43 Y-12.66 X21.85 Y-13.57
256..LXYXY/L192 X21.85 Y-13.57 X19.22 Y-14.38
257..LXYXY/L193 X19.22 Y-14.38 X16.55 Y-15.1
258..LXYXY/L194 X16.55 Y-15.1 X13.83 Y-15.72
259..LXYXY/L195 X13.83 Y-15.72 X11.1 Y-16.22
260..LXYXY/L196 X11.1 Y-16.22 X8.33 Y-16.62
261..LXYXY/L197 X8.33 Y-16.62 X5.56 Y-16.91
262..LXYXY/L198 X5.56 Y-16.91 X2.78 Y-17.08
263..LXYXY/L199 X2.78 Y-17.08 X0 Y-17.12
264..LXYXY/L200 X-2.78 Y-17.08 X0 Y-17.12
265..PXY/P9 X-27.95 Y1.8
```

```
266..PXY/P10 X29.16 Y1.68
267..GROUP/GP9
268..GROUP/L161 L162 L163 L164 L165 L166 L167
269..GROUP/L168 L169 L170 L171 L172 L173 L174
270..GROUP/L175 L176 L177 L178 L179 L200 L199
271..GROUP/L198 L197 L196 L195 L194 L193 L192
272..GROUP/L191 L190 L189 L188 L187 L186 L185
273..GROUP/L184 L183 L182 L181 L180
274..GROUP/GP10
275..END/
276..CALL/SUB5 FR0 MX0 MY0
277..BEGIN/SUB6
278..TRANSL/ANG-90 ANG0 ANG0 D0 D15 D0
279..LXYXY/L201 X-37.72 Y0 X-37.72 Y-4.21
280..LXYXY/L202 X-37.72 Y-4.21 X-37.66 Y-4.78
281..LXYXY/L203 X-37.66 Y-4.78 X-37.52 Y-5.31
282..LXYXY/L204 X-37.52 Y-5.31 X-37.28 Y-5.8
283..LXYXY/L205 X-37.28 Y-5.8 X-36.88 Y-6.28
284..LXYXY/L206 X-36.88 Y-6.28 X-36.3 Y-6.76
285..LXYXY/L207 X-36.3 Y-6.76 X-34.06 Y-8.1
286..LXYXY/L208 X-34.06 Y-8.1 X-31.74 Y-9.35
287..LXYXY/L209 X-31.74 Y-9.35 X-29.34 Y-10.53
288..LXYXY/L210 X-29.34 Y-10.53 X-26.87 Y-11.62
289..LXYXY/L211 X-26.87 Y-11.62 X-24.33 Y-12.62
290..LXYXY/L212 X-24.33 Y-12.62 X-21.75 Y-13.52
291..LXYXY/L213 X-21.75 Y-13.52 X-19.11 Y-14.34
292..LXYXY/L214 X-19.11 Y-14.34 X-16.44 Y-15.05
293..LXYXY/L215 X-16.44 Y-15.05 X-13.74 Y-15.66
294..LXYXY/L216 X-13.74 Y-15.66 X-11.01 Y-16.17
295..LXYXY/L217 X-11.01 Y-16.17 X-8.26 Y-16.56
296..LXYXY/L218 X-8.26 Y-16.56 X-5.51 Y-16.84
297..LXYXY/L219 X-5.51 Y-16.84 X-2.75 Y-17.01
298..LXYXY/L220 X37.72 Y0 X37.72 Y-4.21
299..LXYXY/L221 X37.72 Y-4.21 X37.66 Y-4.78
300..LXYXY/L222 X37.66 Y-4.78 X37.52 Y-5.31
301..LXYXY/L223 X37.52 Y-5.31 X37.28 Y-5.8
302..LXYXY/L224 X37.28 Y-5.8 X36.88 Y-6.28
303..LXYXY/L225 X36.88 Y-6.28 X36.3 Y-6.76
304..LXYXY/L226 X36.3 Y-6.76 X34.06 Y-8.1
305..LXYXY/L227 X34.06 Y-8.1 X31.74 Y-9.35
306..LXYXY/L228 X31.74 Y-9.35 X29.34 Y-10.53
307..LXYXY/L229 X29.34 Y-10.53 X26.87 Y-11.62
308..LXYXY/L230 X26.87 Y-11.62 X24.33 Y-12.62
309..LXYXY/L231 X24.33 Y-12.62 X21.75 Y-13.52
310..LXYXY/L232 X21.75 Y-13.52 X19.11 Y-14.34
311..LXYXY/L233 X19.11 Y-14.34 X16.44 Y-15.05
312..LXYXY/L234 X16.44 Y-15.05 X13.74 Y-15.66
313..LXYXY/L235 X13.74 Y-15.66 X11.01 Y-16.17
314..LXYXY/L236 X11.01 Y-16.17 X8.26 Y-16.56
315..LXYXY/L237 X8.26 Y-16.56 X5.51 Y-16.84
316..LXYXY/L238 X5.51 Y-16.84 X2.75 Y-17.01
317..LXYXY/L239 X2.75 Y-17.01 X0 Y-17.05
318..LXYXY/L240 X-2.75 Y-17.01 X0 Y-17.05
319..PXY/P11 X28.55 Y2.32
```

```
320..PXY/P12 X-28.25 Y3.4
321..GROUP/GP11
322..GROUP/L220 L221 L222 L223 L224 L225 L226
323..GROUP/L227 L228 L229 L230 L231 L232 L233
324..GROUP/L234 L235 L236 L237 L238 L239 L240
325..GROUP/L219 L218 L217 L216 L215 L214 L213
326..GROUP/L212 L211 L210 L209 L208 L207 L206
327..GROUP/L205 L204 L203 L202 L201
328..GROUP/GP12
329..END/
330..CALL/SUB6 FR0 MX0 MY0
331..BEGIN/SUB7
332..TRANSL/ANG-90 ANG0 ANG0 D0 D17 D0
333..LXYXY/L241 X-37.72 Y0 X-37.72 Y-4.13
334..LXYXY/L242 X-37.72 Y-4.13 X-37.66 Y-4.7
335..LXYXY/L243 X-37.66 Y-4.7 X-37.52 Y-5.22
336..LXYXY/L244 X-37.52 Y-5.22 X-37.28 Y-5.71
337..LXYXY/L245 X-37.28 Y-5.71 X-36.88 Y-6.19
338..LXYXY/L246 X-36.88 Y-6.19 X-36.3 Y-6.67
339..LXYXY/L247 X-36.3 Y-6.67 X-34.06 Y-8.01
340..LXYXY/L248 X-34.06 Y-8.01 X-31.74 Y-9.27
341..LXYXY/L249 X-31.74 Y-9.27 X-29.34 Y-10.45
342..LXYXY/L250 X-29.34 Y-10.45 X-26.87 Y-11.54
343..LXYXY/L251 X-26.87 Y-11.54 X-24.33 Y-12.54
344..LXYXY/L252 X-24.33 Y-12.54 X-21.75 Y-13.45
345..LXYXY/L253 X-21.75 Y-13.45 X-19.11 Y-14.26
346..LXYXY/L254 X-19.11 Y-14.26 X-16.44 Y-14.98
347..LXYXY/L255 X-16.44 Y-14.98 X-13.74 Y-15.59
348..LXYXY/L256 X-13.74 Y-15.59 X-11.01 Y-16.09
349..LXYXY/L257 X-11.01 Y-16.09 X-8.26 Y-16.49
350..LXYXY/L258 X-8.26 Y-16.49 X-5.51 Y-16.77
351..LXYXY/L259 X-5.51 Y-16.77 X-2.75 Y-16.93
352..LXYXY/L260 X37.72 Y0 X37.72 Y-4.13
353..LXYXY/L261 X37.72 Y-4.13 X37.66 Y-4.7
354..LXYXY/L262 X37.66 Y-4.7 X37.52 Y-5.22
355..LXYXY/L263 X37.52 Y-5.22 X37.28 Y-5.71
356..LXYXY/L264 X37.28 Y-5.71 X36.88 Y-6.19
357..LXYXY/L265 X36.88 Y-6.19 X36.3 Y-6.67
358..LXYXY/L266 X36.3 Y-6.67 X34.06 Y-8.01
359..LXYXY/L267 X34.06 Y-8.01 X31.74 Y-9.27
360..LXYXY/L268 X31.74 Y-9.27 X29.34 Y-10.45
361..LXYXY/L269 X29.34 Y-10.45 X26.87 Y-11.54
362..LXYXY/L270 X26.87 Y-11.54 X24.33 Y-12.54
363..LXYXY/L271 X24.33 Y-12.54 X21.75 Y-13.45
364..LXYXY/L272 X21.75 Y-13.45 X19.11 Y-14.26
365..LXYXY/L273 X19.11 Y-14.26 X16.44 Y-14.98
366..LXYXY/L274 X16.44 Y-14.98 X13.74 Y-15.59
367..LXYXY/L275 X13.74 Y-15.59 X11.01 Y-16.09
368..LXYXY/L276 X11.01 Y-16.09 X8.26 Y-16.49
369..LXYXY/L277 X8.26 Y-16.49 X5.51 Y-16.77
370..LXYXY/L278 X5.51 Y-16.77 X2.75 Y-16.93
371..LXYXY/L279 X2.75 Y-16.93 X0 Y-16.98
372..LXYXY/L280 X-2.75 Y-16.93 X0 Y-16.98
373..PXY/P13 X-29.64 Y1.93
```

```
374..PXY/P14 X30.12 Y1.99
375..GROUP/GP13
376..GROUP/L241 L242 L243 L244 L245 L246 L247
377..GROUP/L248 L249 L250 L251 L252 L253 L254
378..GROUP/L255 L256 L257 L258 L259 L280 L279
379..GROUP/L278 L277 L276 L275 L274 L273 L272
380..GROUP/L271 L270 L269 L268 L267 L266 L265
381..GROUP/L264 L263 L262 L261 L260
382..GROUP/GP14
383..END/
384..CALL/SUB7 FR0 MX0 MY0
385..BEGIN/SUB9
386..TRANSL/ANG-90 ANG0 ANG0 D0 D19 D0
387..LXYXY/L281 X-37.72 Y0 X-37.72 Y-4.04
388..LXYXY/L282 X-37.72 Y-4.04 X-37.66 Y-4.61
389..LXYXY/L283 X-37.66 Y-4.61 X-37.52 Y-5.13
390..LXYXY/L284 X-37.52 Y-5.13 X-37.28 Y-5.62
391..LXYXY/L285 X-37.28 Y-5.62 X-36.88 Y-6.11
392..LXYXY/L286 X-36.88 Y-6.11 X-36.3 Y-6.59
393..LXYXY/L287 X-36.3 Y-6.59 X-34.06 Y-7.93
394..LXYXY/L288 X-34.06 Y-7.93 X-31.74 Y-9.19
395..LXYXY/L289 X-31.74 Y-9.19 X-29.34 Y-10.37
396..LXYXY/L290 X-29.34 Y-10.37 X-26.87 Y-11.46
397..LXYXY/L291 X-26.87 Y-11.46 X-24.33 Y-12.46
398..LXYXY/L292 X-24.33 Y-12.46 X-21.75 Y-13.37
399..LXYXY/L293 X-21.75 Y-13.37 X-19.11 Y-14.19
400..LXYXY/L294 X-19.11 Y-14.19 X-16.44 Y-14.9
401..LXYXY/L295 X-16.44 Y-14.9 X-13.74 Y-15.51
402..LXYXY/L296 X-13.74 Y-15.51 X-11.01 Y-16.02
403..LXYXY/L297 X-11.01 Y-16.02 X-8.26 Y-16.41
404..LXYXY/L298 X-8.26 Y-16.41 X-5.51 Y-16.7
405..LXYXY/L299 X-5.51 Y-16.7 X-2.75 Y-16.86
406..LXYXY/L300 X37.72 Y0 X37.72 Y-4.04
407..LXYXY/L301 X37.72 Y-4.04 X37.66 Y-4.61
408..LXYXY/L302 X37.66 Y-4.61 X37.52 Y-5.13
409..LXYXY/L303 X37.52 Y-5.13 X37.28 Y-5.62
410..LXYXY/L304 X37.28 Y-5.62 X36.88 Y-6.11
411..LXYXY/L305 X36.88 Y-6.11 X36.3 Y-6.59
412..LXYXY/L306 X36.3 Y-6.59 X34.06 Y-7.93
413..LXYXY/L307 X34.06 Y-7.93 X31.74 Y-9.19
414..LXYXY/L308 X31.74 Y-9.19 X29.34 Y-10.37
415..LXYXY/L309 X29.34 Y-10.37 X26.87 Y-11.46
416..LXYXY/L310 X26.87 Y-11.46 X24.33 Y-12.46
417..LXYXY/L311 X24.33 Y-12.46 X21.75 Y-13.37
418..LXYXY/L312 X21.75 Y-13.37 X19.11 Y-14.19
419..LXYXY/L313 X19.11 Y-14.19 X16.44 Y-14.9
420..LXYXY/L314 X16.44 Y-14.9 X13.74 Y-15.51
421..LXYXY/L315 X13.74 Y-15.51 X11.01 Y-16.02
422..LXYXY/L316 X11.01 Y-16.02 X8.26 Y-16.41
423..LXYXY/L317 X8.26 Y-16.41 X5.51 Y-16.7
424..LXYXY/L318 X5.51 Y-16.7 X2.75 Y-16.86
425..LXYXY/L319 X2.75 Y-16.86 X0 Y-16.91
426..LXYXY/L320 X-2.75 Y-16.86 X0 Y-16.91
427..PXY/P15 X28.37 Y2.33
```

```
428..PXY/P16 X-28.25 Y3.35
429..GROUP/GP15
430..GROUP/L300 L301 L302 L303 L304 L305 L306
431..GROUP/L307 L308 L309 L310 L311 L312 L313
432..GROUP/L314 L315 L316 L317 L318 L319 L320
433..GROUP/L299 L298 L297 L296 L295 L294 L293
434..GROUP/L292 L291 L290 L289 L288 L287 L286
435..GROUP/L285 L284 L283 L282 L281
436..GROUP/GP16
437..END/
438..CALL/SUB9 FR0 MX0 MY0
```

RFNC20.NC1 - A typical Part Program used for the machining of the cab roof model

Machining direction: parallel to the XOZ plane (The axes of the machine is shown in Figure 4.6)

```
prog RFNC20  v5      5/31/1989
001..£££//Job = rfnc20.M01
002..£££/MILL
003..£££//Tool number = 1
004..BEGIN/SUB1
005..TRANSL/ANG-90 ANG0 ANG0 D0 D-0.8 D0
006..PXY/P1 X145.04 Y-4.51
007..PXY/P2 X177.15 Y-38.14
008..LXYXY/L1 X145.04 Y-4.51 X147,48 Y-4.61
009..LXYXY/L2 X147.48 Y-4.61 X149.89 Y-4.95
010..LXYXY/L3 X149.89 Y-4.95 X152.25 Y-5.5
011..LXYXY/L4 X152.25 Y-5.5 X154.54 Y-6.25
012..LXYXY/L5 X154.54 Y-6.25 X156.75 Y-7.18
013..LXYXY/L6 X156.75 Y-7.18 X158.87 Y-8.25
014..LXYXY/L7 X158.87 Y-8.25 X160.87 Y-9.46
015..LXYXY/L8 X160.87 Y-9.46 X162.74 Y-10.78
016..LXYXY/L9 X162.74 Y-10.78 X164.49 Y-12.2
017..LXYXY/L10 X164.49 Y-12.2 X166.1 Y-13.7
018..LXYXY/L11 X166.1 Y-13.7 X167.57 Y-15.27
019..LXYXY/L12 X167.57 Y-15.27 X168.9 Y-16.89
020..LXYXY/L13 X168.9 Y-16.89 X170.1 Y-18.54
021..LXYXY/L14 X170.1 Y-18.54 X171.15 Y-20.23
022..LXYXY/L15 X171.15 Y-20.23 X172.08 Y-21.92
023..LXYXY/L16 X172.08 Y-21.92 X172.88 Y-23.63
024..LXYXY/L17 X172.88 Y-23.63 X173.57 Y-25.33
025..LXYXY/L18 X173.57 Y-25.33 X174.17 Y-27.02
026..LXYXY/L19 X174.17 Y-27.02 X174.68 Y-28.7
027..LXYXY/L20 X174.68 Y-28.7 X175.13 Y-30.35
028..LXYXY/L21 X175.13 Y-30.35 X175.53 Y-31.97
029..LXYXY/L22 X175.53 Y-31.97 X175.9 Y-33.57
030..LXYXY/L23 X175.9 Y-33.57 X176.28 Y-35.13
031..LXYXY/L24 X176.28 Y-35.13 X176.69 Y-36.65
032..LXYXY/L25 X176.69 Y-36.65 X177.15 Y-38.14
033..GROUP/GP1 L1 L2 L3 L4 L5 L6
034..GROUP/L7 L8 L9 L10 L11 L12 L13
035..GROUP/L14 L15 L16 L17 L18 L19 L20
036..GROUP/L21 L22 L23 L24 L25 GP2
037..END/
038..RPLANE/R20
039..TLCHG/X0 Y0 T1 RPM1500 F500
040..TOOL/LR1 DIA0
041..MOVE/X0 Y0 Z20
042..MOVE/X145 Y-0.8 Z20
043..FEDTO/X145 Y-0.8 Z0
044..CALL/SUB1 FR0 MX0 MY0
045..£££//End M. APROFILE- rfnc20.M01
046..BEGIN/SUB2
047..TRANSL/ANG-90 ANG0 ANG0 D0 D-4.8 D0
048..PXY/P3 X177.15 Y-38.14
049..PXY/P4 X145.04 Y-4.51
```

```
050..LXYXY/L26 X177.15 Y-38.14 X176.69 Y-36.65
051..LXYXY/L27 X176.69 Y-36.65 X176.28 Y-35.13
052..LXYXY/L28 X176.28 Y-35.13 X175.9 Y-33.57
053..LXYXY/L29 X175.9 Y-33.57 X175.53 Y-31.97
054..LXYXY/L30 X175.53 Y-31.97 X175.13 Y-30.35
055..LXYXY/L31 X175.13 Y-30.35 X174.68 Y-28.7
056..LXYXY/L32 X174.68 Y-28.7 X174.17 Y-27.02
057..LXYXY/L33 X174.17 Y-27.02 X173.57 Y-25.33
058..LXYXY/L34 X173.57 Y-25.33 X172.88 Y-23.63
059..LXYXY/L35 X172.88 Y-23.63 X172.08 Y-21.92
060..LXYXY/L36 X172.08 Y-21.92 X171.15 Y-20.23
061..LXYXY/L37 X171.15 Y-20.23 X170.1 Y-18.54
062..LXYXY/L38 X170.1 Y-18.54 X168.9 Y-16.89
063..LXYXY/L39 X168.9 Y-16.89 X167.57 Y-15.27
064..LXYXY/L40 X167.57 Y-15.27 X166.1 Y-13.7
065..LXYXY/L41 X166.1 Y-13.7 X164.49 Y-12.2
066..LXYXY/L42 X164.49 Y-12.2 X162.74 Y-10.78
067..LXYXY/L43 X162.74 Y-10.78 X160.87 Y-9.46
068..LXYXY/L44 X160.87 Y-9.46 X158.87 Y-8.25
069..LXYXY/L45 X158.87 Y-8.25 X156.75 Y-7.18
070..LXYXY/L46 X156.75 Y-7.18 X154.54 Y-6.25
071..LXYXY/L47 X154.54 Y-6.25 X152.25 Y-5.5
072..LXYXY/L48 X152.25 Y-5.5 X149.89 Y-4.95
073..LXYXY/L49 X149.89 Y-4.95 X147.48 Y-4.61
074..LXYXY/L50 X147.48 Y-4.61 X145.04 Y-4.51
075..GROUP/GP3 L26 L27 L28 L29 L30 L31
076..GROUP/L32 L33 L34 L35 L36 L37 L38
077..GROUP/L39 L40 L41 L42 L43 L44 L45
078..GROUP/L46 L47 L48 L49 L50 GP4
079..END/
080..CALL/SUB2 FR0 MX0 MY0
081..£££//End M. APROFILE- rfnc20.M02
082..BEGIN/SUB3
083..TRANSL/ANG-90 ANG0 ANG0 D0 D-8.8 D0
084..PXY/P5 X145.04 Y-4.54
085..PXY/P6 X177.19 Y-38.15
086..LXYXY/L51 X145.04 Y-4.54 X147.46 Y-4.64
087..LXYXY/L52 X147.46 Y-4.64 X149.86 Y-4.98
088..LXYXY/L53 X149.86 Y-4.98 X152.21 Y-5.53
089..LXYXY/L54 X152.21 Y-5.53 X154.5 Y-6.27
090..LXYXY/L55 X154.5 Y-6.27 X156.71 Y-7.18
091..LXYXY/L56 X156.71 Y-7.18 X158.82 Y-8.25
092..LXYXY/L57 X158.82 Y-8.25 X160.82 Y-9.45
093..LXYXY/L58 X160.82 Y-9.45 X162.69 Y-10.76
094..LXYXY/L59 X162.69 Y-10.76 X164.44 Y-12.17
095..LXYXY/L60 X164.44 Y-12.17 X166.06 Y-13.66
096..LXYXY/L61 X166.06 Y-13.66 X167.53 Y-15.22
097..LXYXY/L62 X167.53 Y-15.22 X168.87 Y-16.83
098..LXYXY/L63 X168.87 Y-16.83 X170.07 Y-18.48
099..LXYXY/L64 X170.07 Y-18.48 X171.13 Y-20.16
100..LXYXY/L65 X171.13 Y-20.16 X172.07 Y-21.86
101..LXYXY/L66 X172.07 Y-21.86 X172.88 Y-23.56
102..LXYXY/L67 X172.88 Y-23.56 X173.58 Y-25.26
103..LXYXY/L68 X173.58 Y-25.26 X174.18 Y-26.96
```

```
1Ø4..LXYXY/L69 X174.18 Y-26.96 X174.7 Y-28.64
1Ø5..LXYXY/L7Ø X174.7 Y-28.64 X175.15 Y-3Ø.3
1Ø6..LXYXY/L71 X175.15 Y-3Ø.3 X175.56 Y-31.93
1Ø7..LXYXY/L72 X175.56 Y-31.93 X175.94 Y-33.54
1Ø8..LXYXY/L73 X175.94 Y-33.54 X176.32 Y-35.11
1Ø9..LXYXY/L74 X176.32 Y-35.11 X176.73 Y-36.65
11Ø..LXYXY/L75 X176.73 Y-36.65 X177.19 Y-38.15
111..GROUP/GP5 L51 L52 L53 L54 L55 L56
112..GROUP/L57 L58 L59 L6Ø L61 L62 L63
113..GROUP/L64 L65 L66 L67 L68 L69 L7Ø
114..GROUP/L71 L72 L73 L74 L75 GP6
115..END/
116..CALL/SUB3 FRØ MXØ MYØ
117..£££//End M. APROFILE- rfnc2Ø.MØ3
118..BEGIN/SUB4
119..TRANSL/ANG-9Ø ANGØ ANGØ DØ D-12.8 DØ
12Ø..PXY/P7 X177.23 Y-38.16
121..PXY/P8 X145.Ø3 Y-4.58
122..LXYXY/L76 X177.23 Y-38.16 X176.77 Y-36.64
123..LXYXY/L77 X176.77 Y-36.64 X176.36 Y-35.Ø9
124..LXYXY/L78 X176.36 Y-35.Ø9 X175.97 Y-33.51
125..LXYXY/L79 X175.97 Y-33.51 X175.59 Y-31.89
126..LXYXY/L8Ø X175.59 Y-31.89 X175.17 Y-3Ø.25
127..LXYXY/L81 X175.17 Y-3Ø.25 X174.71 Y-28.58
128..LXYXY/L82 X174.71 Y-28.58 X174.19 Y-26.9
129..LXYXY/L83 X174.19 Y-26.9 X173.58 Y-25.2
13Ø..LXYXY/L84 X173.58 Y-25.2 X172.87 Y-23.49
131..LXYXY/L85 X172.87 Y-23.49 X172.Ø5 Y-21.79
132..LXYXY/L86 X172.Ø5 Y-21.79 X171.11 Y-2Ø.Ø9
133..LXYXY/L87 X171.11 Y-2Ø.Ø9 X17Ø.Ø4 Y-18.42
134..LXYXY/L88 X17Ø.Ø4 Y-18.42 X168.84 Y-16.77
135..LXYXY/L89 X168.84 Y-16.77 X167.49 Y-15.17
136..LXYXY/L9Ø X167.49 Y-15.17 X166.Ø1 Y-13.62
137..LXYXY/L91 X166.Ø1 Y-13.62 X164.4 Y-12.14
138..LXYXY/L92 X164.4 Y-12.14 X162.64 Y-1Ø.74
139..LXYXY/L93 X162.64 Y-1Ø.74 X16Ø.76 Y-9.43
14Ø..LXYXY/L94 X16Ø.76 Y-9.43 X158.77 Y-8.25
141..LXYXY/L95 X158.77 Y-8.25 X156.66 Y-7.19
142..LXYXY/L96 X156.66 Y-7.19 X154.46 Y-6.29
143..LXYXY/L97 X154.46 Y-6.29 X152.17 Y-5.55
144..LXYXY/L98 X152.17 Y-5.55 X149.83 Y-5.Ø1
145..LXYXY/L99 X149.83 Y-5.Ø1 X147.44 Y-4.68
146..LXYXY/L1ØØ X147.44 Y-4.68 X145.Ø3 Y-4.58
147..GROUP/GP7 L76 L77 L78 L79 L8Ø L81
148..GROUP/L82 L83 L84 L85 L86 L87 L88
149..GROUP/L89 L9Ø L91 L92 L93 L94 L95
15Ø..GROUP/L96 L97 L98 L99 L1ØØ GP8
151..END/
152..CALL/SUB4 FRØ MXØ MYØ
153..£££//End M. APROFILE- rfnc2Ø.MØ4
154..BEGIN/SUB5
155..TRANSL/ANG-9Ø ANGØ ANGØ DØ D-16.8 DØ
156..PXY/P9 X145.Ø3 Y-4.62
157..PXY/P1Ø X177.27.Y-38.17
```

```
158..LXYXY/L101 X145.03 Y-4.62 X147.42 Y-4.71
159..LXYXY/L102 X147.42 Y-4.71 X149.8 Y-5.04
160..LXYXY/L103 X149.8 Y-5.04 X152.14 Y-5.58
161..LXYXY/L104 X152.14 Y-5.58 X154.41 Y-6.3
162..LXYXY/L105 X154.41 Y-6.3 X156.61 Y-7.2
163..LXYXY/L106 X156.61 Y-7.2 X158.71 Y-8.25
164..LXYXY/L107 X158.71 Y-8.25 X160.71 Y-9.42
165..LXYXY/L108 X160.71 Y-9.42 X162.59 Y-10.72
166..LXYXY/L109 X162.59 Y-10.72 X164.35 Y-12.11
167..LXYXY/L110 X164.35 Y-12.11 X165.97 Y-13.58
168..LXYXY/L111 X165.97 Y-13.58 X167.45 Y-15.12
169..LXYXY/L112 X167.45 Y-15.12 X168.8 Y-16.72
170..LXYXY/L113 X168.8 Y-16.72 X170.01 Y-18.36
171..LXYXY/L114 X170.01 Y-18.36 X171.09 Y-20.03
172..LXYXY/L115 X171.09 Y-20.03 X172.03 Y-21.73
173..LXYXY/L116 X172.03 Y-21.73 X172.86 Y-23.43
174..LXYXY/L117 X172.86 Y-23.43 X173.58 Y-25.14
175..LXYXY/L118 X173.58 Y-25.14 X174.19 Y-26.84
176..LXYXY/L119 X174.19 Y-26.84 X174.73 Y-28.53
177..LXYXY/L120 X174.73 Y-28.53 X175.19 Y-30.2
178..LXYXY/L121 X175.19 Y-30.2 X175.61 Y-31.85
179..LXYXY/L122 X175.61 Y-31.85 X176 Y-33.48
180..LXYXY/L123 X176 Y-33.48 X176.39 Y-35.07
181..LXYXY/L124 X176.39 Y-35.07 X176.81 Y-36.64
182..LXYXY/L125 X176.81 Y-36.64 X177.27 Y-38.17
183..GROUP/GP9 L101 L102 L103 L104 L105 L106
184..GROUP/L107 L108 L109 L110 L111 L112 L113
185..GROUP/L114 L115 L116 L117 L118 L119 L120
186..GROUP/L121 L122 L123 L124 L125 GP10
187..END/
188..CALL/SUB5 FR0 MX0 MY0
189..£££//End M. APROFILE- rfnc20.M05
190..BEGIN/SUB6
191..TRANSL/ANG-90 ANG0 ANG0 D0 D-20.8 D0
192..PXY/P11 X177.3 Y-38.18
193..PXY/P12 X145.02 Y-4.66
194..LXYXY/L126 X177.3 Y-38.18 X176.84 Y-36.64
195..LXYXY/L127 X176.84 Y-36.64 X176.42 Y-35.06
196..LXYXY/L128 X176.42 Y-35.06 X176.03 Y-33.45
197..LXYXY/L129 X176.03 Y-33.45 X175.63 Y-31.82
198..LXYXY/L130 X175.63 Y-31.82 X175.21 Y-30.16
199..LXYXY/L131 X175.21 Y-30.16 X174.74 Y-28.48
200..LXYXY/L132 X174.74 Y-28.48 X174.2 Y-26.78
201..LXYXY/L133 X174.2 Y-26.78 X173.57 Y-25.08
202..LXYXY/L134 X173.57 Y-25.08 X172.85 Y-23.37
203..LXYXY/L135 X172.85 Y-23.37 X172.01 Y-21.67
204..LXYXY/L136 X172.01 Y-21.67 X171.06 Y-19.97
205..LXYXY/L137 X171.06 Y-19.97 X169.98 Y-18.31
206..LXYXY/L138 X169.98 Y-18.31 X168.76 Y-16.67
207..LXYXY/L139 X168.76 Y-16.67 X167.41 Y-15.08
208..LXYXY/L140 X167.41 Y-15.08 X165.92 Y-13.54
209..LXYXY/L141 X165.92 Y-13.54 X164.29 Y-12.08
210..LXYXY/L142 X164.29 Y-12.08 X162.54 Y-10.7
211..LXYXY/L143 X162.54 Y-10.7 X160.66 Y-9.42
```

```
212..LXYXY/L144 X160.66 Y-9.42 X158.66 Y-8.25
213..LXYXY/L145 X158.66 Y-8.25 X156.56 Y-7.21
214..LXYXY/L146 X156.56 Y-7.21 X154.37 Y-6.33
215..LXYXY/L147 X154.37 Y-6.33 X152.1 Y-5.61
216..LXYXY/L148 X152.1 Y-5.61 X149.77 Y-5.08
217..LXYXY/L149 X149.77 Y-5.08 X147.4 Y-4.75
218..LXYXY/L150 X147.4 Y-4.75 X145.02 Y-4.66
219..GROUP/GP11 L126 L127 L128 L129 L130 L131
220..GROUP/L132 L133 L134 L135 L136 L137 L138
221..GROUP/L139 L140 L141 L142 L143 L144 L145
222..GROUP/L146 L147 L148 L149 L150 GP12
223..END/
224..CALL/SUB6 FR0 MX0 MY0
225..£££//End M. APROFILE- rfnc20.M06
226..BEGIN/SUB7
227..TRANSL/ANG-90 ANG0 ANG0 D0 D-24.8 D0
228..PXY/P13 X145.01 Y-4.7
229..PXY/P14 X177.33 Y-38.19
230..LXYXY/L151 X145.01 Y-4.7 X147.38 Y-4.79
231..LXYXY/L152 X147.38 Y-4.79 X149.74 Y-5.11
232..LXYXY/L153 X149.74 Y-5.11 X152.06 Y-5.64
233..LXYXY/L154 X152.06 Y-5.64 X154.32 Y-6.35
234..LXYXY/L155 X154.32 Y-6.35 X156.51 Y-7.23
235..LXYXY/L156 X156.51 Y-7.23 X158.61 Y-8.26
236..LXYXY/L157 X158.61 Y-8.26 X160.6 Y-9.41
237..LXYXY/L158 X160.6 Y-9.41 X162.48 Y-10.68
238..LXYXY/L159 X162.48 Y-10.68 X164.24 Y-12.06
239..LXYXY/L160 X164.24 Y-12.06 X165.87 Y-13.51
240..LXYXY/L161 X165.87 Y-13.51 X167.36 Y-15.04
241..LXYXY/L162 X167.36 Y-15.04 X168.72 Y-16.62
242..LXYXY/L163 X168.72 Y-16.62 X169.94 Y-18.25
243..LXYXY/L164 X169.94 Y-18.25 X171.03 Y-19.92
244..LXYXY/L165 X171.03 Y-19.92 X171.99 Y-21.61
245..LXYXY/L166 X171.99 Y-21.61 X172.83 Y-23.31
246..LXYXY/L167 X172.83 Y-23.31 X173.56 Y-25.02
247..LXYXY/L168 X173.56 Y-25.02 X174.19 Y-26.73
248..LXYXY/L169 X174.19 Y-26.73 X174.74 Y-28.43
249..LXYXY/L170 X174.74 Y-28.43 X175.22 Y-30.12
250..LXYXY/L171 X175.22 Y-30.12 X175.65 Y-31.79
251..LXYXY/L172 X175.65 Y-31.79 X176.05 Y-33.43
252..LXYXY/L173 X176.05 Y-33.43 X176.45 Y-35.05
253..LXYXY/L174 X176.45 Y-35.05 X176.86 Y-36.64
254..LXYXY/L175 X176.86 Y-36.64 X177.33 Y-38.19
255..GROUP/GP13 L151 L152 L153 L154 L155 L156
256..GROUP/L157 L158 L159 L160 L161 L162 L163
257..GROUP/L164 L165 L166 L167 L168 L169 L170
258..GROUP/L171 L172 L173 L174 L175 GP14
259..END/
260..CALL/SUB7 FR0 MX0 MY0
261..£££//End M. APROFILE- rfnc20.M07
262..BEGIN/SUB8
263..TRANSL/ANG-90 ANG0 ANG0 D0 D-28.8 D0
264..PXY/P15 X177.35 Y-38.21
265..PXY/P16 X145.01 Y-4.74
```

```
266..LXYXY/L176 X177.35 Y-38.21 X176.89 Y-36.64
267..LXYXY/L177 X176.89 Y-36.64 X176.47 Y-35.04
268..LXYXY/L178 X176.47 Y-35.04 X176.07 Y-33.41
269..LXYXY/L179 X176.07 Y-33.41 X175.66 Y-31.75
270..LXYXY/L180 X175.66 Y-31.75 X175.23 Y-30.08
271..LXYXY/L181 X175.23 Y-30.08 X174.74 Y-28.38
272..LXYXY/L182 X174.74 Y-28.38 X174.19 Y-26.68
273..LXYXY/L183 X174.19 Y-26.68 X173.55 Y-24.97
274..LXYXY/L184 X173.55 Y-24.97 X172.81 Y-23.25
275..LXYXY/L185 X172.81 Y-23.25 X171.97 Y-21.55
276..LXYXY/L186 X171.97 Y-21.55 X171 Y-19.86
277..LXYXY/L187 X171 Y-19.86 X169.9 Y-18.2
278..LXYXY/L188 X169.9 Y-18.2 X168.68 Y-16.58
279..LXYXY/L189 X168.68 Y-16.58 X167.31 Y-15
280..LXYXY/L190 X167.31 Y-15 X165.82 Y-13.48
281..LXYXY/L191 X165.82 Y-13.48 X164.19 Y-12.03
282..LXYXY/L192 X164.19 Y-12.03 X162.43 Y-10.67
283..LXYXY/L193 X162.43 Y-10.67 X160.55 Y-9.41
284..LXYXY/L194 X160.55 Y-9.41 X158.55 Y-8.26
285..LXYXY/L195 X158.55 Y-8.26 X156.46 Y-7.25
286..LXYXY/L196 X156.46 Y-7.25 X154.27 Y-6.38
287..LXYXY/L197 X154.27 Y-6.38 X152.02 Y-5.67
288..LXYXY/L198 X152.02 Y-5.67 X149.71 Y-5.15
289..LXYXY/L199 X149.71 Y-5.15 X147.36 Y-4.84
290..LXYXY/L200 X147.36 Y-4.84 X145.01 Y-4.74
291..GROUP/GP15 L176 L177 L178 L179 L180 L181
292..GROUP/L182 L183 L184 L185 L186 L187 L188
293..GROUP/L189 L190 L191 L192 L193 L194 L195
294..GROUP/L196 L197 L198 L199 L200 GP16
295..END/
296..CALL/SUB8 FR0 MX0 MY0
297..FEDTO/X145.01 Y20 Z0
298..£££//End M. APROFILE- rfnc20.M08
299..£££//End of Entire JOB rfnc20
```

RFNC50.NC1 - A typical Part Program used for the machining of the cab roof model

Machining direction: parallel to the XOY plane
(The axes of the machine is shown in Figure 4.6)

```
prog RFNC50  v4      5/31/1989
001..£££/COMMAND file - RFNC50.TXT
002..£££//Job = RFNC50.M01
003..£££/CUT IN XOY PLANE
004..£££//Tool number = 2
005..£££//Drawing = rooftp2
006..£££//Total ENTITIES =  12
007..BEGIN/SUB1
008..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-26,5
009..PXY/P1 X149.04 Y-185.07
010..PXY/P2 X161.67 Y-171.67
011..LXYXY/L1 X149.04 Y-185.07 X151.02 Y-184.02
012..LXYXY/L2 X151.02 Y-184.02 X152.75 Y-183
013..LXYXY/L3 X152.75 Y-183 X154.27 Y-181.97
014..LXYXY/L4 X154.27 Y-181.97 X155.61 Y-180.92
015..LXYXY/L5 X155.61 Y-180.92 X156.79 Y-179.82
016..LXYXY/L6 X156.79 Y-179.82 X157.84 Y-178.67
017..LXYXY/L7 X157.84 Y-178.67 X158.76 Y-177.47
018..LXYXY/L8 X158.76 Y-177.47 X159.58 Y-176.24
019..LXYXY/L9 X159.58 Y-176.24 X160.28 Y-175
020..LXYXY/L10 X160.28 Y-175 X160.87 Y-173.79
021..LXYXY/L11 X160.87 Y-173.79 X161.34 Y-172.66
022..LXYXY/L12 X161.34 Y-172.66 X161.67 Y-171.67
023..GROUP/GP1 L1 L2 L3 L4 L5 L6
024..GROUP/L7 L8 L9 L10 L11 L12 GP2
025..END/
026..RPLANE/R20
027..TLCHG/X0 Y0 T2 RPM1500 F500
028..TOOL/LR1 DIA0
029..MOVE/X0 Y0 Z20
030..MOVE/X149.04 Y-185.07 Z20
031..FEDTO/X149.04 Y-185 Z0
032..CALL/SUB1 FR0 MX0 MY0
033..£££//End M. APROFILE- RFNC50.M01
034..£££/COMMAND file - KEY2.TXT
035..£££//Job = RFNC50.M02
036..BEGIN/SUB2
037..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-29.5
038..PXY/P3 X163.63 Y-171.37
039..PXY/P4 X153.54 Y-184.49
040..LXYXY/L13 X163.63 Y-171.37 X163.38 Y-172.26
041..LXYXY/L14 X163.38 Y-172.26 X163.06 Y-173.32
042..LXYXY/L15 X163.06 Y-173.32 X162.64 Y-174.48
043..LXYXY/L16 X162.64 Y-174.48 X162.13 Y-175.71
044..LXYXY/L17 X162.13 Y-175.71 X161.5 Y-176.95
045..LXYXY/L18 X161.5 Y-176.95 X160.75 Y-178.18
046..LXYXY/L19 X160.75 Y-178.18 X159.88 Y-179.38
047..LXYXY/L20 X159.88 Y-179.38 X158.88 Y-180.52
048..LXYXY/L21 X158.88 Y-180.52 X157.74 Y-181.6
049..LXYXY/L22 X157.74 Y-181.6 X156.47 Y-182.62
```

```
050..LXYXY/L23 X156.47 Y-182.62 X155.07 Y-183.58
051..LXYXY/L24 X155.07 Y-183.58 X153.54 Y-184.49
052..GROUP/GP3 L13 L14 L15 L16 L17 L18
053..GROUP/L19 L20 L21 L22 L23 L24 GP4
054..END/
055..CALL/SUB2 FR0 MX0 MY0
056..£££//Job = RFNC50.M03
057..BEGIN/SUB3
058..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-32.5
059..PXY/P5 X157.23 Y-183.83
060..PXY/P6 X165.21 Y-171.06
061..LXYXY/L25 X157.23 Y-183.83 X158.34 Y-183.06
062..LXYXY/L26 X158.34 Y-183.06 X159.43 Y-182.17
063..LXYXY/L27 X159.43 Y-182.17 X160.46 Y-181.17
064..LXYXY/L28 X160.46 Y-181.17 X161.41 Y-180.07
065..LXYXY/L29 X161.41 Y-180.07 X162.25 Y-178.91
066..LXYXY/L30 X162.25 Y-178.91 X162.97 Y-177.7
067..LXYXY/L31 X162.97 Y-177.7 X163.58 Y-176.46
068..LXYXY/L32 X163.58 Y-176.46 X164.07 Y-175.23
069..LXYXY/L33 X164.07 Y-175.23 X164.46 Y-174.05
070..LXYXY/L34 X164.46 Y-174.05 X164.76 Y-172.93
071..LXYXY/L35 X164.76 Y-172.93 X165 Y-171.92
072..LXYXY/L36 X165 Y-171.92 X165.21 Y-171.06
073..GROUP/GP5 L25 L26 L27 L28 L29 L30
074..GROUP/L31 L32 L33 L34 L35 L36 GP6
075..END/
076..CALL/SUB3 FR0 MX0 MY0
077..£££//Job = RFNC50.M04
078..BEGIN/SUB4
079..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-35.5
080..PXY/P7 X166.52 Y-170.76
081..PXY/P8 X156.17 Y-186.54
082..LXYXY/L37 X166.52 Y-170.76 X166.24 Y-172.32
083..LXYXY/L38 X166.24 Y-172.32 X165.9 Y-173.81
084..LXYXY/L39 X165.9 Y-173.81 X165.49 Y-175.22
085..LXYXY/L40 X165.49 Y-175.22 X164.99 Y-176.57
086..LXYXY/L41 X164.99 Y-176.57 X164.4 Y-177.87
087..LXYXY/L42 X164.4 Y-177.87 X163.71 Y-179.11
088..LXYXY/L43 X163.71 Y-179.11 X162.94 Y-180.29
089..LXYXY/L44 X162.94 Y-180.29 X162.09 Y-181.41
090..LXYXY/L45 X162.09 Y-181.41 X161.18 Y-182.46
091..LXYXY/L46 X161.18 Y-182.46 X160.24 Y-183.43
092..LXYXY/L47 X160.24 Y-183.43 X159.29 Y-184.3
093..LXYXY/L48 X159.29 Y-184.3 X158.36 Y-185.07
094..LXYXY/L49 X158.36 Y-185.07 X157.5 Y-185.71
095..LXYXY/L50 X157.5 Y-185.71 X156.75 Y-186.21
096..LXYXY/L51 X156.75 Y-186.21 X156.17 Y-186.54
097..GROUP/GP7 L37 L38 L39 L40 L41 L42
098..GROUP/L43 L44 L45 L46 L47 L48 L49
099..GROUP/L50 L51 GP8
100..END/
101..CALL/SUB4 FR0 MX0 MY0
102..£££//End M. APROFILE- RFNC50.M04
103..£££/COMMAND file - KEY5.TXT
```

```
104..BEGIN/SUB5
105..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-38.5
106..PXY/P9 X158.3 Y-186.6
107..PXY/P10 X167.67 Y-170.51
108..LXYXY/L52 X158.3 Y-186.6 X159.21 Y-185.72
109..LXYXY/L53 X159.21 Y-185.72 X160.14 Y-184.88
110..LXYXY/L54 X160.14 Y-184.88 X161.06 Y-184.03
111..LXYXY/L55 X161.06 Y-184.03 X161.95 Y-183.17
112..LXYXY/L56 X161.95 Y-183.17 X162.8 Y-182.26
113..LXYXY/L57 X162.8 Y-182.26 X163.6 Y-181.31
114..LXYXY/L58 X163.6 Y-181.31 X164.34 Y-180.29
115..LXYXY/L59 X164.34 Y-180.29 X165.01 Y-179.21
116..LXYXY/L60 X165.01 Y-179.21 X165.61 Y-178.07
117..LXYXY/L61 X165.61 Y-178.07 X166.13 Y-176.87
118..LXYXY/L62 X166.13 Y-176.87 X166.57 Y-175.62
119..LXYXY/L63 X166.57 Y-175.62 X166.94 Y-174.34
120..LXYXY/L64 X166.94 Y-174.34 X167.25 Y-173.04
121..LXYXY/L65 X167.25 Y-173.04 X167.48 Y-171.76
122..LXYXY/L66 X167.48 Y-171.76 X167.67 Y-170.51
123..GROUP/GP9 L52 L53 L54 L55 L56 L57
124..GROUP/L58 L59 L60 L61 L62 L63 L64
125..GROUP/L65 L66 GP10
126..END/
127..CALL/SUB5 FR0 MX0 MY0
128..£££//End M. APROFILE- RFNC50.M05
129..£££/COMMAND file - KEY6.TXT
130..BEGIN/SUB6
131..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-41.5
132..PXY/P11 X168.74 Y-170.3
133..PXY/P12 X159.74 Y-186.9
134..LXYXY/L67 X168.74 Y-170.3 X168.49 Y-172
135..LXYXY/L68 X168.49 Y-172 X168.15 Y-173.64
136..LXYXY/L69 X168.15 Y-173.64 X167.73 Y-175.19
137..LXYXY/L70 X167.73 Y-175.19 X167.23 Y-176.67
138..LXYXY/L71 X167.23 Y-176.67 X166.67 Y-178.06
139..LXYXY/L72 X166.67 Y-178.06 X166.06 Y-179.37
140..LXYXY/L73 X166.06 Y-179.37 X165.4 Y-180.59
141..LXYXY/L74 X165.4 Y-180.59 X164.7 Y-181.72
142..LXYXY/L75 X164.7 Y-181.72 X163.99 Y-182.75
143..LXYXY/L76 X163.99 Y-182.75 X163.25 Y-183.7
144..LXYXY/L77 X163.25 Y-183.7 X162.52 Y-184.54
145..LXYXY/L78 X162.52 Y-184.54 X161.79 Y-185.29
146..LXYXY/L79 X161.79 Y-185.29 X161.08 Y-185.93
147..LXYXY/L80 X161.08 Y-185.93 X160.39 Y-186.47
148..LXYXY/L81 X160.39 Y-186.47 X159.74 Y-186.9
149..GROUP/GP11 L67 L68 L69 L70 L71 L72
150..GROUP/L73 L74 L75 L76 L77 L78 L79
151..GROUP/L80 L81 GP12
152..END/
153..CALL/SUB6 FR0 MX0 MY0
154..£££//End M. APROFILE- RFNC50.M06
155..£££/COMMAND file - KEY7.TXT
156..BEGIN/SUB7
157..TRANSL/ANG0 ANG0 ANG0 D0 D0 D-44.5
```

```
158..PXY/P13 X160.54 Y-187.19
159..PXY/P14 X169.96 Y-170
160..LXYXY/L82 X160.54 Y-187.19 X161.22 Y-186.79
161..LXYXY/L83 X161.22 Y-186.79 X161.92 Y-186.28
162..LXYXY/L84 X161.92 Y-186.28 X162.64 Y-185.64
163..LXYXY/L85 X162.64 Y-185.64 X163.37 Y-184.89
164..LXYXY/L86 X163.37 Y-184.89 X164.1 Y-184.02
165..LXYXY/L87 X164.1 Y-184.02 X164.83 Y-183.05
166..LXYXY/L88 X164.83 Y-183.05 X165.54 Y-181.97
167..LXYXY/L89 X165.54 Y-181.97 X166.24 Y-180.79
168..LXYXY/L90 X166.24 Y-180.79 X166.91 Y-179.51
169..LXYXY/L91 X166.91 Y-179.51 X167.54 Y-178.14
170..LXYXY/L92 X167.54 Y-178.14 X168.14 Y-176.68
171..LXYXY/L93 X168.14 Y-176.68 X168.68 Y-175.13
172..LXYXY/L94 X168.68 Y-175.13 X169.17 Y-173.5
173..LXYXY/L95 X169.17 Y-173.5 X169.6 Y-171.79
174..LXYXY/L96 X169.6 Y-171.79 X169.96 Y-170
175..GROUP/GP13 L82 L83 L84 L85 L86 L87
176..GROUP/L88 L89 L90 L91 L92 L93 L94
177..GROUP/L95 L96 GP14
178..END/
179..CALL/SUB7 FR0 MX0 MY0
180..FEDTO/X169.96 Y-170 Z20
181..£££//End of Entire JOB RFNC50
```

RFNC60.NC1 - A typical Part Program used for the machining of the cab roof model

Machining direction: parallel to the YOZ plane (The axes of the machine is shown in Figure 4.6)

```
prog RFNC60  v4      5/31/1989
001..£££/TOOL NO = 2
002..BEGIN/SUB1
003..TRANSL/ANG0 ANG90 ANG0 D147 D0 D0
004..PXY/P1 X-17.52 Y-171.87
005..PXY/P2 X-31.03 Y-188.75
006..LXYXY/L1 X-17.52 Y-171.87 X-18.18 Y-173.57
007..LXYXY/L2 X-18.18 Y-173.57 X-18.82 Y-175.17
008..LXYXY/L3 X-18.82 Y-175.17 X-19.47 Y-176.67
009..LXYXY/L4 X-19.47 Y-176.67 X-20.14 Y-178.09
010..LXYXY/L5 X-20.14 Y-178.09 X-20.86 Y-179.44
011..LXYXY/L6 X-20.86 Y-179.44 X-21.64 Y-180.72
012..LXYXY/L7 X-21.64 Y-180.72 X-22.49 Y-181.93
013..LXYXY/L8 X-22.49 Y-181.93 X-23.4 Y-183.08
014..LXYXY/L9 X-23.4 Y-183.08 X-24.38 Y-184.16
015..LXYXY/L10 X-24.38 Y-184.16 X-25.42 Y-185.16
016..LXYXY/L11 X-25.42 Y-185.16 X-26.51 Y-186.09
017..LXYXY/L12 X-26.51 Y-186.09 X-27.64 Y-186.93
018..LXYXY/L13 X-27.64 Y-186.93 X-28.79 Y-187.66
019..LXYXY/L14 X-28.79 Y-187.66 X-29.93 Y-188.27
020..LXYXY/L15 X-29.93 Y-188.27 X-31.03 Y-188.75
021..GROUP/GP1 L1 L2 L3 L4 L5 L6
022..GROUP/L7 L8 L9 L10 L11 L12 L13
023..GROUP/L14 L15 GP2
024..END/
025..RPLANE/R20
026..TLCHG/X0 Y0 T2 RPM1500 F500
027..TOOL/LR1 DIA0
028..MOVE/X0 Y0 Z20
029..MOVE/X147 Y-171.87 Z20
030..FEDTO/X147 Y-171.87 Z0
031..CALL/SUB1 FR0 MX0 MY0
032..£££//End M. APROFILE- RFNC60.M01
033..BEGIN/SUB2
034..TRANSL/ANG0 ANG90 ANG0 D150 D0 D0
035..PXY/P3 X-31.62 Y-187.82
036..PXY/P4 X-18.25 Y-171.94
037..LXYXY/L16 X-31.62 Y-187.82 X-30.42 Y-187.17
038..LXYXY/L17 X-30.42 Y-187.17 X-29.3 Y-186.54
039..LXYXY/L18 X-29.3 Y-186.54 X-28.25 Y-185.89
040..LXYXY/L19 X-28.25 Y-185.89 X-27.26 Y-185.21
041..LXYXY/L20 X-27.26 Y-185.21 X-26.3 Y-184.47
042..LXYXY/L21 X-26.3 Y-184.47 X-25.37 Y-183.66
043..LXYXY/L22 X-25.37 Y-183.66 X-24.46 Y-182.75
044..LXYXY/L23 X-24.46 Y-182.75 X-23.57 Y-181.74
045..LXYXY/L24 X-23.57 Y-181.74 X-22.7 Y-180.62
046..LXYXY/L25 X-22.7 Y-180.62 X-21.86 Y-179.4
047..LXYXY/L26 X-21.86 Y-179.4 X-21.04 Y-178.07
048..LXYXY/L27 X-21.04 Y-178.07 X-20.26 Y-176.65
049..LXYXY/L28 X-20.26 Y-176.65 X-19.52 Y-175.14
```

```
050..LXYXY/L29 X-19.52 Y-175.14 X-18.85 Y-173.56
051..LXYXY/L30 X-18.85 Y-173.56 X-18.25 Y-171.94
052..GROUP/GP3 L16 L17 L18 L19 L20 L21
053..GROUP/L22 L23 L24 L25 L26 L27 L28
054..GROUP/L29 L30 GP4
055..END/
056..CALL/SUB2 FR0 MX0 MY0
057..£££//End M. APROFILE- RFNC60.M02
058..BEGIN/SUB3
059..TRANSL/ANG0 ANG90 ANG0 D153 D0 D0
060..PXY/P5 X-19.4 Y-172
061..PXY/P6 X-33.11 Y-187.22
062..LXYXY/L31 X-19.4 Y-172 X-19.93 Y-173.19
063..LXYXY/L32 X-19.93 Y-173.19 X-20.43 Y-174.34
064..LXYXY/L33 X-20.43 Y-174.34 X-20.91 Y-175.44
065..LXYXY/L34 X-20.91 Y-175.44 X-21.4 Y-176.49
066..LXYXY/L35 X-21.4 Y-176.49 X-21.92 Y-177.52
067..LXYXY/L36 X-21.92 Y-177.52 X-22.48 Y-178.52
068..LXYXY/L37 X-22.48 Y-178.52 X-23.12 Y-179.49
069..LXYXY/L38 X-23.12 Y-179.49 X-23.84 Y-180.45
070..LXYXY/L39 X-23.84 Y-180.45 X-24.67 Y-181.4
071..LXYXY/L40 X-24.67 Y-181.4 X-25.63 Y-182.35
072..LXYXY/L41 X-25.63 Y-182.35 X-26.74 Y-183.3
073..LXYXY/L42 X-26.74 Y-183.3 X-28.03 Y-184.25
074..LXYXY/L43 X-28.03 Y-184.25 X-29.5 Y-185.22
075..LXYXY/L44 X-29.5 Y-185.22 X-31.19 Y-186.21
076..LXYXY/L45 X-31.19 Y-186.21 X-33.11 Y-187.22
077..GROUP/GP5 L31 L32 L33 L34 L35 L36
078..GROUP/L37 L38 L39 L40 L41 L42 L43
079..GROUP/L44 L45 GP6
080..END/
081..CALL/SUB3 FR0 MX0 MY0
082..£££//End M. APROFILE- RFNC60.M03
083..BEGIN/SUB4
084..TRANSL/ANG0 ANG90 ANG0 D156 D0 D0
085..PXY/P7 X-35.3 Y-186.55
086..PXY/P8 X-21.08 Y-172
087..LXYXY/L46 X-35.3 Y-186.55 X-34.61 Y-186.15
088..LXYXY/L47 X-34.61 Y-186.15 X-33.78 Y-185.7
089..LXYXY/L48 X-33.78 Y-185.7 X-32.85 Y-185.19
090..LXYXY/L49 X-32.85 Y-185.19 X-31.82 Y-184.6
091..LXYXY/L50 X-31.82 Y-184.6 X-30.74 Y-183.92
092..LXYXY/L51 X-30.74 Y-183.92 X-29.61 Y-183.13
093..LXYXY/L52 X-29.61 Y-183.13 X-28.47 Y-182.25
094..LXYXY/L53 X-28.47 Y-182.25 X-27.33 Y-181.26
095..LXYXY/L54 X-27.33 Y-181.26 X-26.22 Y-180.16
096..LXYXY/L55 X-26.22 Y-180.16 X-25.15 Y-178.97
097..LXYXY/L56 X-25.15 Y-178.97 X-24.14 Y-177.7
098..LXYXY/L57 X-24.14 Y-177.7 X-23.21 Y-176.34
099..LXYXY/L58 X-23.21 Y-176.34 X-22.38 Y-174.93
100..LXYXY/L59 X-22.38 Y-174.93 X-21.67 Y-173.47
101..LXYXY/L60 X-21.67 Y-173.47 X-21.08 Y-172
102..GROUP/GP7 L46 L47 L48 L49 L50 L51
103..GROUP/L52 L53 L54 L55 L56 L57 L58
```

```
104..GROUP/L59 L60 GF8
105..END/
106..CALL/SUB4 FRØ MXØ MYØ
107..£££//End M. APROFILE- RFNC60.MØ4
108..BEGIN/SUB5
109..TRANSL/ANGØ ANG9Ø ANGØ D159 DØ DØ
110..PXY/P9 X-23.47 Y-171.9
111..PXY/P1Ø X-34.13 Y-183.5
112..LXYXY/L61 X-23.47 Y-171.9 X-23.92 Y-172.78
113..LXYXY/L62 X-23.92 Y-172.78 X-24.36 Y-173.67
114..LXYXY/L63 X-24.36 Y-173.67 X-24.83 Y-174.56
115..LXYXY/L64 X-24.83 Y-174.56 X-25.33 Y-175.45
116..LXYXY/L65 X-25.33 Y-175.45 X-25.89 Y-176.34
117..LXYXY/L66 X-25.89 Y-176.34 X-26.51 Y-177.21
118..LXYXY/L67 X-26.51 Y-177.21 X-27.2 Y-178.Ø8
119..LXYXY/L68 X-27.2 Y-178.Ø8 X-27.96 Y-178.92
120..LXYXY/L69 X-27.96 Y-178.92 X-28.78 Y-179.73
121..LXYXY/L7Ø X-28.78 Y-179.73 X-29.65 Y-18Ø.51
122..LXYXY/L71 X-29.65 Y-18Ø.51 X-3Ø.57 Y-181.24
123..LXYXY/L72 X-3Ø.57 Y-181.24 X-31.5 Y-181.92
124..LXYXY/L73 X-31.5 Y-181.92 X-32.42 Y-182.53
125..LXYXY/L74 X-32.42 Y-182.53 X-33.31 Y-183.Ø6
126..LXYXY/L75 X-33.31 Y-183.Ø6 X-34.13 Y-183.5`
127..GROUP/GF9 L61 L62 L63 L64 L65 L66
128..GROUP/L67 L68 L69 L7Ø L71 L72 L73
129..GROUP/L74 L75 GF1Ø
130..END/
131..CALL/SUB5 FRØ MXØ MYØ
132..FEDTO/X2Ø Y-183.5 ZØ
```

Appendix Six

THE POST-PROCESSED NC PART PROGRAMS

(.NC2)

BR1.NC2 - The post processed Part Program (BR1.NC1)

```
O1001~~~~~~~~~~~~~~
N0001G0G90G40G80G21G54
N0002T06M06
N0003G00X0.0Y211.0S3000M03
N0004G43Z10.0H06
N0005Z-5.0
N0006X-2.122Y211.392
N0007G01Y217.4F500.0
N0008X-10.802
N0009X-33.375Y210.465
N0010X-34.463Y193.839
N0011X-34.217Y189.863
N0012X-27.907Y171.685
N0013X-26.718Y164.117
N0014X-27.345Y143.402
N0015X-29.092Y71.741
N0016X-29.611Y45.855
N0017X-31.757Y38.684
N0018X-33.717Y35.494
N0019X-34.575Y32.294
N0020X-34.698Y8.0
N0021X34.698
N0022X34.575Y32.294
N0023X33.717Y35.494
N0024X31.757Y38.684
N0025X29.611Y45.855
N0026X29.092Y71.741
N0027X27.345Y143.402
N0028X26.718Y164.117
N0029X27.907Y171.685
N0030X34.217Y189.863
N0031X34.463Y193.839
N0032X33.375Y210.465
N0033X10.802Y217.4
N0034X4.775
N0035G00Y211.126
N0036G01Z20.0
N0037X0.0Y211.0
N0038M09
N0039M05
N0040M30
%
~
```

BD1.NC2 - The post processed Part Program (BD1.NC1)

```
O0002~~~~~~~~~~~~~
N0001G0G90G40G80G21G54
N0002T02M06
N0003G00G43Z20.0H02S3000M03
N0004X0.0Y280.0
N0005Y5.0
N0006G01Z0.0F500.0
N0007X-27.409Z2.919
N0008X-37.725
N0009Z-4.638
N0010X-37.657Z-5.216
N0011X-37.519Z-5.742
N0012X-37.271Z-6.234
N0013X-36.874Z-6.711
N0014X-36.291Z-7.192
N0015X-34.084Z-8.5
N0016X-31.787Z-9.736
N0017X-29.408Z-10.895
N0018X-26.954Z-11.974
N0019X-24.434Z-12.967
N0020X-21.854Z-13.872
N0021X-19.223Z-14.684
N0022X-16.548Z-15.398
N0023X-13.836Z-16.012
N0024X-11.096Z-16.520
N0025X-8.334Z-16.918
N0026X-5.559Z-17.203
N0027X1.455Z-17.624
N0028X8.334Z-16.918
N0029X11.096Z-16.520
N0030X13.836Z-16.012
N0031X16.548Z-15.398
N0032X19.223Z-14.684
N0033X21.854Z-13.872
N0034X24.434Z-12.967
N0035X26.954Z-11.974
N0036X29.408Z-10.895
N0037X31.787Z-9.736
N0038X34.084Z-8.5
N0039X36.291Z-7.192
N0040X36.874Z-6.711
N0041X37.271Z-6.234
N0042X37.519Z-5.742
N0043X37.657Z-5.216
N0044X37.725Z-4.638
N0045Z1.955
N0046X27.891
N0047X28.975Y7.0Z1.931
N0048X37.725
N0049Z-4.552
```

```
N0050X37.658Z-5.130
N0051X37.520Z-5.655
N0052X37.272Z-6.147
N0053X36.876Z-6.625
N0054X36.293Z-7.106
N0055X34.085Z-8.416
N0056X31.787Z-9.654
N0057X29.407Z-10.815
N0058X26.953Z-11.895
N0059X24.433Z-12.890
N0060X21.853Z-13.796
N0061X19.222Z-14.609
N0062X16.547Z-15.324
N0063X13.836Z-15.938
N0064X11.096Z-16.446
N0065X8.335Z-16.845
N0066X5.560Z-17.130
N0067X-1.456Z-17.551
N0068X-8.335Z-16.845
N0069X-11.096Z-16.446
N0070X-13.836Z-15.938
N0071X-16.547Z-15.324
N0072X-19.222Z-14.609
N0073X-21.853Z-13.796
N0074X-24.433Z-12.890
N0075X-26.953Z-11.895
N0076X-29.407Z-10.815
N0077X-31.787Z-9.654
N0078X-34.084Z-8.416
N0079X-36.293Z-7.106
N0080X-36.876Z-6.625
N0081X-37.272Z-6.147
N0082X-37.520Z-5.655
N0083X-37.658Z-5.129
N0084X-37.725Z-4.552
N0085Z3.196
N0086X-28.252
N0087X-28.493Y9.0Z2.449
N0088X-37.725
N0089Z-4.467
N0090X-37.658Z-5.043
N0091X-37.520Z-5.568
N0092X-37.273Z-6.060
N0093X-36.877Z-6.538
N0094X-36.295Z-7.020
N0095X-34.085Z-8.333
N0096X-31.787Z-9.572
N0097X-29.406Z-10.735
N0098X-26.952Z-11.817
N0099X-24.432Z-12.813
N0100X-21.852Z-13.720
N0101X-19.221Z-14.533
N0102X-16.547Z-15.249
N0103X-13.835Z-15.864
```

```
N0104X-11.096Z-16.372
N0105X-8.335Z-16.771
N0106X-5.560Z-17.056
N0107X1.454Z-17.478
N0108X8.335Z-16.771
N0109X11.096Z-16.372
N0110X13.835Z-15.864
N0111X16.547Z-15.249
N0112X19.221Z-14.533
N0113X21.852Z-13.720
N0114X24.432Z-12.813
N0115X26.952Z-11.817
N0116X29.406Z-10.735
N0117X31.787Z-9.572
N0118X34.085Z-8.333
N0119X36.295Z-7.020
N0120X36.877Z-6.538
N0121X37.273Z-6.060
N0122X37.520Z-5.568
N0123X37.658Z-5.043
N0124X37.725Z-4.467
N0125Z2.148
N0126X28.794
N0127X28.433Y11.0Z2.305
N0128X37.725
N0129Z-4.382
N0130X37.659Z-4.956
N0131X37.521Z-5.480
N0132X37.274Z-5.973
N0133X36.879Z-6.452
N0134X36.296Z-6.934
N0135X34.086Z-8.249
N0136X31.787Z-9.491
N0137X29.406Z-10.655
N0138X26.951Z-11.738
N0139X24.431Z-12.736
N0140X21.851Z-13.644
N0141X19.221Z-14.458
N0142X16.546Z-15.174
N0143X13.835Z-15.789
N0144X11.096Z-16.299
N0145X8.335Z-16.698
N0146X5.560Z-16.983
N0147X-1.454Z-17.406
N0148X-8.335Z-16.698
N0149X-11.096Z-16.299
N0150X-13.835Z-15.789
N0151X-16.546Z-15.174
N0152X-19.221Z-14.458
N0153X-21.851Z-13.644
N0154X-24.431Z-12.736
N0155X-26.951Z-11.738
N0156X-29.406Z-10.655
```

```
N0157X-31.787Z-9.491
N0158X-34.086Z-8.249
N0159X-36.296Z-6.934
N0160X-36.879Z-6.452
N0161X-37.274Z-5.973
N0162X-37.521Z-5.480
N0163X-37.659Z-4.956
N0164X-37.725Z-4.382
N0165Z3.570
N0166X-28.433
N0167X-27.951Y13.0Z1.799
N0168X-37.725
N0169Z-4.297
N0170X-37.659Z-4.869
N0171X-37.522Z-5.393
N0172X-37.275Z-5.886
N0173X-36.880Z-6.365
N0174X-36.298Z-6.847
N0175X-34.087Z-8.165
N0176X-31.786Z-9.409
N0177X-29.405Z-10.575
N0178X-26.951Z-11.659
N0179X-24.430Z-12.658
N0180X-21.850Z-13.567
N0181X-19.220Z-14.382
N0182X-16.545Z-15.1
N0183X-13.835Z-15.715
N0184X-11.096Z-16.225
N0185X-8.335Z-16.624
N0186X-5.560Z-16.910
N0187X1.453Z-17.333
N0188X8.335Z-16.624
N0189X11.096Z-16.225
N0190X13.835Z-15.715
N0191X16.545Z-15.1
N0192X19.220Z-14.382
N0193X21.850Z-13.567
N0194X24.430Z-12.658
N0195X26.951Z-11.659
N0196X29.405Z-10.575
N0197X31.786Z-9.409
N0198X34.087Z-8.165
N0199X36.298Z-6.847
N0200X36.880Z-6.365
N0201X37.275Z-5.886
N0202X37.522Z-5.393
N0203X37.659Z-4.869
N0204X37.725Z-4.297
N0205Z1.679
N0206X29.156
N0207X28.553Y15.0Z2.318
N0208X37.725
N0209Z-4.212
N0210X37.659Z-4.782
```

```
N0211X37.523Z-5.306
N0212X37.276Z-5.799
N0213X36.882Z-6.278
N0214X36.3Z-6.761
N0215X34.061Z-8.096
N0216X31.738Z-9.352
N0217X29.337Z-10.527
N0218X26.867Z-11.616
N0219X24.335Z-12.617
N0220X21.748Z-13.525
N0221X19.114Z-14.337
N0222X16.441Z-15.051
N0223X13.736Z-15.661
N0224X11.007Z-16.165
N0225X8.262Z-16.560
N0226X5.507Z-16.841
N0227X-1.442Z-17.257
N0228X-8.262Z-16.560
N0229X-11.007Z-16.165
N0230X-13.736Z-15.661
N0231X-16.441Z-15.051
N0232X-19.114Z-14.337
N0233X-21.748Z-13.525
N0234X-24.335Z-12.617
N0235X-26.867Z-11.616
N0236X-29.337Z-10.527
N0237X-31.738Z-9.352
N0238X-34.061Z-8.096
N0239X-36.3Z-6.761
N0240X-36.882Z-6.278
N0241X-37.276Z-5.799
N0242X-37.523Z-5.306
N0243X-37.659Z-4.782
N0244X-37.725Z-4.212
N0245Z3.402
N0246X-28.252
N0247X-29.637Y17.0Z1.932
N0248X-37.725
N0249Z-4.127
N0250X-37.660Z-4.695
N0251X-37.523Z-5.218
N0252X-37.277Z-5.712
N0253X-36.883Z-6.192
N0254X-36.302Z-6.675
N0255X-34.062Z-8.012
N0256X-31.738Z-9.270
N0257X-29.336Z-10.447
N0258X-26.866Z-11.538
N0259X-24.334Z-12.539
N0260X-21.747Z-13.449
N0261X-19.113Z-14.262
N0262X-16.441Z-14.976
N0263X-13.736Z-15.587
N0264X-11.007Z-16.092
```

```
N0265X-8.262Z-16.487
N0266X-5.507Z-16.768
N0267X1.441Z-17.184
N0268X8.262Z-16.487
N0269X11.007Z-16.092
N0270X13.736Z-15.587
N0271X16.441Z-14.976
N0272X19.113Z-14.262
N0273X21.747Z-13.449
N0274X24.334Z-12.539
N0275X26.866Z-11.538
N0276X29.336Z-10.447
N0277X31.738Z-9.270
N0278X34.062Z-8.012
N0279X36.302Z-6.675
N0280X36.883Z-6.192
N0281X37.277Z-5.712
N0282X37.523Z-5.218
N0283X37.660Z-4.695
N0284X37.725Z-4.127
N0285Z1.993
N0286X30.119
N0287X28.372Y19.0Z2.330
N0288X37.725
N0289Z-4.042
N0290X37.660Z-4.608
N0291X37.524Z-5.131
N0292X37.279Z-5.625
N0293X36.885Z-6.105
N0294X36.303Z-6.588
N0295X34.063Z-7.928
N0296X31.738Z-9.189
N0297X29.336Z-10.367
N0298X26.865Z-11.459
N0299X24.333Z-12.462
N0300X21.746Z-13.373
N0301X19.113Z-14.187
N0302X16.440Z-14.901
N0303X13.736Z-15.513
N0304X11.007Z-16.018
N0305X8.262Z-16.413
N0306X5.507Z-16.695
N0307X-1.440Z-17.111
N0308X-8.262Z-16.413
N0309X-11.007Z-16.018
N0310X-13.736Z-15.513
N0311X-16.440Z-14.901
N0312X-19.113Z-14.187
N0313X-21.746Z-13.373
N0314X-24.333Z-12.462
N0315X-26.865Z-11.459
N0316X-29.336Z-10.367
N0317X-31.738Z-9.189
N0318X-34.063Z-7.928
```

```
N0319X-36.303Z-6.588
N0320X-36.885Z-6.105
N0321X-37.279Z-5.625
N0322X-37.524Z-5.131
N0323X-37.660Z-4.608
N0324X-37.725Z-4.042
N0325Z3.354
N0326X-28.252
N0327M09
N0328M05
N0329M30
%
~
```

RFNC20.NC2 - The post processed Part Program (RFNC20.NC1)

```
O0020~~~~~~~~~~~~~~
N0001G0G90G40G80G21G54
N0002T01M06
N0003G00G43Z20.0H01S1500M03
N0004X0.0Y0.0
N0005X145.0Y-0.8
N0006G01Z0.0F500.0
N0007X145.042Z-4.510
N0008X147.477Z-4.608
N0009X149.887Z-4.947
N0010X152.249Z-5.503
N0011X154.544Z-6.254
N0012X156.755Z-7.176
N0013X158.866Z-8.251
N0014X160.866Z-9.459
N0015X162.743Z-10.782
N0016X164.490Z-12.201
N0017X166.101Z-13.702
N0018X167.573Z-15.268
N0019X168.904Z-16.886
N0020X170.095Z-18.542
N0021X171.151Z-20.226
N0022X172.077Z-21.925
N0023X172.881Z-23.630
N0024X173.574Z-25.332
N0025X174.169Z-27.023
N0026X174.680Z-28.698
N0027X175.127Z-30.350
N0028X175.526Z-31.972
N0029X175.903Z-33.566
N0030X176.280Z-35.126
N0031X176.685Z-36.652
N0032X177.147Z-38.142
N0033Y-4.8
N0034X176.685Z-36.652
N0035X176.280Z-35.126
N0036X175.903Z-33.565
N0037X175.526Z-31.972
N0038X175.127Z-30.350
N0039X174.681Z-28.698
N0040X174.169Z-27.023
N0041X173.574Z-25.332
N0042X172.881Z-23.630
N0043X172.077Z-21.925
N0044X171.151Z-20.225
N0045X170.095Z-18.542
N0046X168.904Z-16.886
N0047X167.573Z-15.268
N0048X166.101Z-13.702
N0049X164.490Z-12.201
```

```
N0050X162.743Z-10.782
N0051X160.866Z-9.459
N0052X158.866Z-8.251
N0053X156.755Z-7.176
N0054X154.544Z-6.254
N0055X152.249Z-5.503
N0056X149.886Z-4.947
N0057X147.477Z-4.608
N0058X145.042Z-4.509
N0059X145.037Y-8.8Z-4.543
N0060X147.459Z-4.640
N0061X149.857Z-4.976
N0062X152.211Z-5.526
N0063X154.5Z-6.268
N0064X156.707Z-7.182
N0065X158.817Z-8.247
N0066X160.816Z-9.445
N0067X162.694Z-10.757
N0068X164.443Z-12.167
N0069X166.058Z-13.659
N0070X167.534Z-15.217
N0071X168.871Z-16.828
N0072X170.069Z-18.479
N0073X171.132Z-20.159
N0074X172.065Z-21.856
N0075X172.877Z-23.561
N0076X173.578Z-25.265
N0077X174.180Z-26.960
N0078X174.699Z-28.640
N0079X175.152Z-30.299
N0080X175.559Z-31.932
N0081X175.940Z-33.534
N0082X176.322Z-35.108
N0083X176.730Z-36.646
N0084X177.193Z-38.150
N0085X177.230Y-12.8Z-38.160
N0086X176.770Z-36.641
N0087X176.360Z-35.090
N0088X175.975Z-33.509
N0089X175.587Z-31.892
N0090X175.175Z-30.250
N0091X174.715Z-28.584
N0092X174.189Z-26.899
N0093X173.579Z-25.2
N0094X172.870Z-23.495
N0095X172.051Z-21.790
N0096X171.110Z-20.095
N0097X170.040Z-18.419
N0098X168.836Z-16.773
N0099X167.494Z-15.168
N0100X166.013Z-13.618
N0101X164.395Z-12.135
N0102X162.643Z-10.735
N0103X160.764Z-9.433
```

```
N0104X158.766Z-8.246
N0105X156.659Z-7.190
N0106X154.457Z-6.285
N0107X152.174Z-5.551
N0108X149.828Z-5.007
N0109X147.440Z-4.675
N0110X145.031Z-4.578
N0111X145.025Y-16.8Z-4.616
N0112X147.421Z-4.712
N0113X149.798Z-5.040
N0114X152.135Z-5.578
N0115X154.412Z-6.305
N0116X156.610Z-7.201
N0117X158.714Z-8.246
N0118X160.712Z-9.424
N0119X162.592Z-10.716
N0120X164.345Z-12.106
N0121X165.967Z-13.580
N0122X167.452Z-15.122
N0123X168.799Z-16.720
N0124X170.010Z-18.361
N0125X171.086Z-20.033
N0126X172.034Z-21.726
N0127X172.860Z-23.431
N0128X173.576Z-25.138
N0129X174.194Z-26.841
N0130X174.727Z-28.531
N0131X175.193Z-30.204
N0132X175.612Z-31.856
N0133X176.004Z-33.483
N0134X176.393Z-35.074
N0135X176.806Z-36.638
N0136X177.271Z-38.170
N0137X177.303Y-20.8Z-38.182
N0138X176.837Z-36.637
N0139X176.422Z-35.060
N0140X176.029Z-33.457
N0141X175.632Z-31.820
N0142X175.208Z-30.160
N0143X174.736Z-28.480
N0144X174.196Z-26.784
N0145X173.571Z-25.078
N0146X172.848Z-23.369
N0147X172.014Z-21.665
N0148X171.059Z-19.974
N0149X169.977Z-18.305
N0150X168.760Z-16.670
N0151X167.408Z-15.079
N0152X165.918Z-13.545
N0153X164.294Z-12.080
N0154X162.538Z-10.699
N0155X160.658Z-9.417
N0156X158.661Z-8.250
N0157X156.560Z-7.214
```

```
N0158X154.366Z-6.327
N0159X152.096Z-5.608
N0160X149.768Z-5.076
N0161X147.401Z-4.751
N0162X145.019Z-4.656
N0163X145.013Y-24.8Z-4.698
N0164X147.382Z-4.793
N0165X149.738Z-5.114
N0166X152.057Z-5.640
N0167X154.320Z-6.351
N0168X156.509Z-7.229
N0169X158.608Z-8.255
N0170X160.603Z-9.413
N0171X162.484Z-10.685
N0172X164.241Z-12.056
N0173X165.868Z-13.512
N0174X167.362Z-15.038
N0175X168.719Z-16.623
N0176X169.941Z-18.253
N0177X171.030Z-19.917
N0178X171.992Z-21.606
N0179X172.833Z-23.310
N0180X173.563Z-25.021
N0181X174.194Z-26.730
N0182X174.741Z-28.430
N0183X175.219Z-30.118
N0184X175.649Z-31.785
N0185X176.420Z-34.948
N0186X176.864Z-36.636
N0187X177.331Z-38.194
N0188X177.353Y-28.8Z-38.208
N0189X176.885Z-36.637
N0190X176.466Z-35.037
N0191X175.651Z-31.716
N0192X175.226Z-30.078
N0193X174.742Z-28.384
N0194X174.189Z-26.678
N0195X173.551Z-24.966
N0196X172.814Z-23.254
N0197X171.966Z-21.550
N0198X170.999Z-19.863
N0199X169.903Z-18.202
N0200X168.676Z-16.578
N0201X167.314Z-15.0
N0202X165.816Z-13.482
N0203X164.186Z-12.035
N0204X162.428Z-10.673
N0205X160.547Z-9.411
N0206X158.553Z-8.264
N0207X156.457Z-7.247
N0208X154.274Z-6.378
N0209X152.017Z-5.674
N0210X149.707Z-5.154
N0211X147.363Z-4.837
```

```
N0212X145.007Z-4.744
N0213X145.010Z20.0
N0214M09
N0215M05
N0216M30
%
~
```

RFNC50.NC2 - The post processed Part Program (RFNC50.NC1)

```
OØØ5Ø~~~~~~~~~~~~~~
NØØØ1GØG9ØG4ØG8ØG21G54
NØØØ2TØ2MØ6
NØØØ3GØØG43Z2Ø.ØHØ2S15ØØMØ3
NØØØ4XØ.ØYØ.Ø
NØØØ5X149.Ø39Y-185.Ø74
NØØØ6GØ1X149.Ø4ØY-185.ØF5ØØ.Ø
NØØØ7ZØ.Ø
NØØØ8X149.Ø39Y-185.Ø74Z-26.5
NØØØ9X151.Ø18Y-184.Ø17
NØØ1ØX152.749Y-182.995
NØØ11X154.268Y-181.972
NØØ12X156.25ØY-1[illegible]
NØØ13X157.837Y-178.[illegible]
NØØ14X158.764Y-177.471
NØØ15X159.577Y-176.24Ø
NØØ16X16Ø.28ØY-175.ØØ3
NØØ17X16Ø.869Y-173.795
NØØ18X161.336Y-172.664
NØØ19X161.667Y-171.666
NØØ2ØX163.631Y-171.37ØZ-29.5
NØØ21X163.382Y-172.261
NØØ22X163.Ø57Y-173.316
NØØ23X162.643Y-174.481
NØØ24X162.128Y-175.7Ø6
NØØ25X161.5Y-176.951
NØØ26X16Ø.754Y-178.184
NØØ27X159.881Y-179.379
NØØ28X158.879Y-18Ø.521
NØØ29X157.189Y-182.127
NØØ3ØX155.Ø74Y-183.575
NØØ31X153.544Y-184.493
NØØ32X157.231Y-183.835Z-32.5
NØØ33X158.345Y-183.Ø62
NØØ34X159.432Y-182.167
NØØ35X16Ø.461Y-181.165
NØØ36X161.4Ø6Y-18Ø.Ø72
NØØ37X162.246Y-178.9Ø8
NØØ38X162.973Y-177.697
NØØ39X163.58ØY-176.463
NØØ4ØX164.Ø72Y-175.235
NØØ41X164.46ØY-174.Ø45
NØØ42X164.761Y-172.927
NØØ43X165.ØØ1Y-171.919
NØØ44X165.211Y-171.Ø63
NØØ45X166.524Y-17Ø.763Z-35.5
NØØ46X166.241Y-172.322
NØØ47X165.9Ø2Y-173.8Ø6
NØØ48X165.489Y-175.221
NØØ49X164.99ØY-176.575
```

```
N0050X164.398Y-177.872
N0051X163.713Y-179.112
N0052X162.941Y-180.293
N0053X162.092Y-181.412
N0054X161.184Y-182.440
N0055X160.239Y-183.429
N0056X158.845Y-184.710
N0057X157.5Y-185.715
N0058X156.754Y-186.210
N0059X156.173Y-186.536
N0060X158.296Y-186.595Z-38.5
N0061X159.213Y-185.724
N0062X160.920Y-184.169
N0063X161.951Y-183.168
N0064X162.804Y-182.262
N0065X163.604Y-181.305
N0066X164.343Y-180.289
N0067X165.013Y-179.209
N0068X165.610Y-178.067
N0069X166.130Y-176.867
N0070X166.574Y-175.620
N0071X166.944Y-174.340
N0072X167.245Y-173.044
N0073X167.484Y-171.758
N0074X167.670Y-170.508
N0075X168.737Y-170.298Z-41.5
N0076X168.491Y-172.005
N0077X168.152Y-173.636
N0078X167.729Y-175.191
N0079X167.232Y-176.665
N0080X166.671Y-178.058
N0081X166.056Y-179.365
N0082X165.396Y-180.586
N0083X164.703Y-181.716
N0084X163.986Y-182.754
N0085X163.254Y-183.697
N0086X162.519Y-184.543
N0087X161.790Y-185.288
N0088X161.076Y-185.931
N0089X160.388Y-186.469
N0090X159.737Y-186.9
N0091X160.542Y-187.188Z-44.5
N0092X161.220Y-186.794
N0093X161.922Y-186.277
N0094X162.641Y-185.641
N0095X163.370Y-184.888
N0096X164.102Y-184.024
N0097X164.828Y-183.050
N0098X165.543Y-181.972
N0099X166.239Y-180.792
N0100X166.908Y-179.514
N0101X167.543Y-178.142
N0102X168.137Y-176.680
N0103X168.683Y-175.131
```

```
N0104X169.173Y-173.499
N0105X169.6Y-171.787
N0106X169.957Y-170.0
N0107Z-24.5
N0108X169.960
N0109M09
N0110M05
N0111M30
%
~
```

RFNC60.NC2 - The post processed Part Program (RFNC60.NC1)

```
O0060~~~~~~~~~~~~~~
N0001G0G90G40G80G21G54
N0002T02M06
N0003G00G43Z20.0H02S1500M03
N0004X0.0Y0.0
N0005X147.0Y-171.870
N0006G01Z0.0F500.0
N0007Y-171.873Z-17.522
N0008Y-173.574Z-18.180
N0009Y-175.170Z-18.819
N0010Y-176.673Z-19.466
N0011Y-178.093Z-20.141
N0012Y-179.439Z-20.863
N0013Y-180.717Z-21.641
N0014Y-181.929Z-22.486
N0015Y-183.076Z-23.399
N0016Y-184.156Z-24.380
N0017Y-185.162Z-25.421
N0018Y-186.089Z-26.514
N0019Y-186.925Z-27.643
N0020Y-187.658Z-28.788
N0021Y-188.272Z-29.926
N0022Y-188.750Z-31.027
N0023X150.0Y-187.824Z-31.617
N0024Y-187.168Z-30.418
N0025Y-186.536Z-29.304
N0026Y-185.892Z-28.254
N0027Y-185.213Z-27.256
N0028Y-184.474Z-26.296
N0029Y-183.657Z-25.365
N0030Y-182.749Z-24.458
N0031Y-181.739Z-23.570
N0032Y-180.623Z-22.702
N0033Y-179.399Z-21.856
N0034Y-178.071Z-21.037
N0035Y-176.648Z-20.255
N0036Y-175.138Z-19.521
N0037Y-173.562Z-18.848
N0038Y-171.937Z-18.255
N0039X153.0Y-171.995Z-19.4
N0040Y-173.192Z-19.933
N0041Y-174.338Z-20.431
N0042Y-175.435Z-20.913
N0043Y-176.495Z-21.402
N0044Y-177.521Z-21.919
N0045Y-178.518Z-22.483
N0046Y-179.494Z-23.115
N0047Y-180.453Z-23.836
N0048Y-181.403Z-24.667
N0049Y-182.348Z-25.629
```

```
N0050Y-183.296Z-26.741
N0051Y-184.250Z-28.025
N0052Y-185.219Z-29.502
N0053Y-186.209Z-31.191
N0054Y-187.221Z-33.115
N0055X156.0Y-186.548Z-35.299
N0056Y-186.149Z-34.609
N0057Y-185.702Z-33.782
N0058Y-185.191Z-32.845
N0059Y-184.599Z-31.821
N0060Y-183.916Z-30.736
N0061Y-183.134Z-29.610
N0062Y-182.249Z-28.468
N0063Y-181.258Z-27.329
N0064Y-180.165Z-26.215
N0065Y-178.975Z-25.145
N0066Y-177.696Z-24.138
N0067Y-176.343Z-23.212
N0068Y-174.929Z-22.383
N0069Y-173.475Z-21.668
N0070Y-172.002Z-21.082
N0071X159.0Y-171.904Z-23.467
N0072Y-172.783Z-23.917
N0073Y-173.668Z-24.362
N0074Y-174.559Z-24.828
N0075Y-175.449Z-25.332
N0076Y-176.335Z-25.890
N0077Y-177.213Z-26.511
N0078Y-178.076Z-27.201
N0079Y-178.919Z-27.959
N0080Y-179.734Z-28.780
N0081Y-180.511Z-29.654
N0082Y-181.243Z-30.566
N0083Y-181.920Z-31.497
N0084Y-182.530Z-32.422
N0085Y-183.062Z-33.310
N0086Y-183.504Z-34.128
N0087Y-183.5Z-34.130
N0088Z20.0
N0089M09
N0090M05
N0091M30
%
~
```