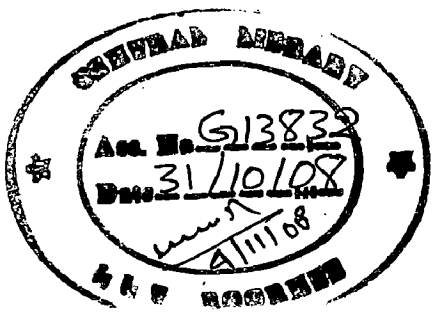
TOOL PATH GENERATION FOR FREEFORM SURFACES

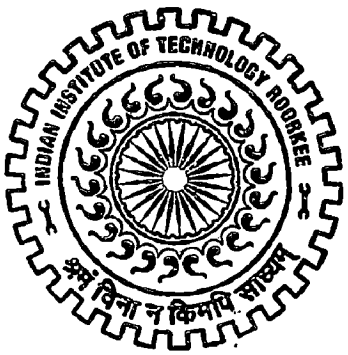
A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
MECHANICAL ENGINEERING
(With Specialization in CAD/CAM & ROBOTICS)

By

M. SATISH CHANDRA





DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2008

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation titled **"TOOL PATH GENERATION FOR FREEFORM SURFACES"**, in partial fulfillment of the requirements for the award of degree of **Master of Technology** in **Mechanical Engineering** (with specialization in **CAD/CAM, Robotics**) submitted to the **Department of Mechanical and Industrial Engineering, Indian Institute of Technology Roorkee**, is an authentic record of my own work carried out under the supervision of **Dr N. K. Mehta**, Professor, Department of Mechanical and Industrial Engineering, IIT Roorkee and **Dr P. M. Pathak**, Assistant Professor, Department of Mechanical and Industrial Engineering, IIT Roorkee.

I have not submitted the matter incorporated in this dissertation report to any other institute or university for the award of any degree.

Date: 27/06/2008

Place: Roorkee

(M. SATISH CHANDRA)

This is to certify that above statement made by candidate is correct to best of my knowledge and belief.

(Dr N. K. Mehta)
Professor
MIED
IIT, Roorkee

(Dr P. M. Pathak)
Assistant Professor
MIED
IIT, Roorkee

ACKNOWLEDGEMENT

It is my privilege to express my deep sense of gratitude and sincere thanks to **Dr.N.K.MEHTA, Professor** and **Dr.P.M.PATHAK, Assistant Professor** in the Department of **Mechanical & Industrial Engineering, Indian Institute of Technology Roorkee, India** for their valuable guidance. Working under their guidance is a privilege and an excellent learning experience that I will cherish for a long time.

I express my sincere thanks to **Dr.PRADEEP KUMAR, Prof & Head, Department of Mechanical & Industrial Engineering** for his encouragement in bringing out this dissertation report.

I am very thankful to my parents & all of friends for their never ending encouragement in bringing out this project report to the form which it is in now

Date: 13-06-2008

Place: Roorkee.

(M.SATISH CHANDRA)

ABSTRACT

Freeform surfaces now-a-days have many applications in industries like aeronautics, automobiles and their manufacturing is a challenging task for manufacturing industries. In manufacturing of these sculptured parts from prismatic stock rough machining usually constitutes most of the machining time owing to the significant difference between the stock and the part shape. In the metal cutting industry, a large amount of time is often spent on data exchange, numerical control (NC) tool path generation and NC program preparation for sculptured parts. Generally manufacturing for these sculptured parts is done on 5-axis machine due to their complexity. In the present work an approach called slicing technique is used to convert the 3D geometry (machined on 5-axis) of sculptured part to 2D geometric (machined on 2 ½ or 3-axis) level and then a simple zig-zag path strategy is applied for each slice according to its level of details (LOD). Basically the system includes four components: - a data extraction module, a volume slicing module, a tool path module and Cutter Location (CL) file generation module.

A CAD design model is usually documented in a particular data format. The geometric and topological messages may not be easily "visualized" or understood by the tool path generator (a system or a human programmer). Therefore the information needed for tool path generation has to be extracted from CAD files and converted in to usable format for generating tool paths. This is done by the data extraction module.

The volume slicing module slices the freeform surface horizontally layer-by-layer to improve the machining efficiency. The reason is that planar path is more advantageous and moreover it reduces a 3D problem to series of 2D problems.

The tool path module will generate the zig-zag tool path for each layer based on its level of details. This module is applied for both roughing and finishing operations.

The cutter location (CL) file generation module generates a CL file which contains the co-ordinate movements of tool that is the TOOL PATH to get the final desired shape.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

T1	-	Total machining time
N	-	Number of layers
t_i	-	Total production time of i th layer
t_m	-	Tool rapid traverse time of ith layer
t_{ci}	-	Cutting time for ith layer
B_i	-	Control points (curve)
$B_{i\,j}$	-	Control points (surface)
u	-	Parametric value
$J_{n\,,\,i}$ (u)	-	Bernstein basis function
n	-	No. of control points
w	-	Parametric value
$K_{m,\,j}$ (w)	-	Bernstein basis function
$N_{i,\,k}$ (u)	-	B-Spline basis function
$M_{j\,l}$ (w)	-	B-spline basis function
x_i , y_i	-	Knots
P_{CL}	-	Cutter Location point
P_{CC}	-	Cutter Contact point
d	-	distance to travel
T	-	Tolerance range
r	-	Cutter radius

ABBREVATIONS

NC	-	Numerical Control
CNC	-	Computer Numerical Control
2D, 2 ½ D, 3D	-	Dimensional
MCU	-	Machine Control Unit
PTP	-	Point To Point
CL	-	Cutter Location
TAD	-	Tool Approach Direction
CAD	-	Computer Aided Design
CAM	-	Computer Aided Manufacturing
MFR	-	Machining Feature Recognition
LOD	-	Level Of Details
WP	-	Work Piece
CC	-	Cutter Contact
STEP	-	Standard for the Exchange of Product Model Data
IGES	-	Initial Graphics Exchange Specification
OS	-	Optimum Stock

CHAPTER 1
INTRODUCTION

Design and manufacturing are two major components in the engineering aspect of production. Product designs that cannot be realized through manufacturing processes are a waste of effort. On the other hand, manufacturing processes cannot be effective unless backed by good design and process plan. The problem of design and machining of complex profiles is especially relevant for many industries such as aerospace, automobile etc.

In engineering design we need to represent an object (mathematical model) with precise drawings, perform requisite analysis to possibly optimize and finally manufacture it. The free form surfaces play an important role in representing these types of complicated surfaces. Three dimensional, or space curves play an important role in engineering design and manufacturing of a diverse range of products, e.g.: - automobiles, ship hulls, air craft fuselages, propeller blades, etc...They also play an important role in the description and interpretation of physical phenomena e.g.: - in geogly, physics and medical science. Also these polynomial curves have wide application for trajectory planning of robots. One sort of technique for curve or surface generation constrains the curves to pass through existing data points. They are commonly known as "*curve fitting*" techniques. They are particularly suited to shape description, where the basic shape is arrived at by experimental evaluation or mathematical calculation. However there is another class of shape design problem that depends on both aesthetic and functional requirements. For such class of design, control points are used to define the curves. These methods are frequently referred to as "*curve fairing*" techniques. This technique has the flexibility to change the shape of the curve by simply shifting the control points. Examples are Bezier and B-spline curves and surfaces.

Now a day's most of the industries use NC machines for manufacturing freeform surfaces. These machines require part program as an input to make a particular component. This part program consists of location of tool movements in detail that is the sequence of co-ordinate point's to get the final desired shape. So it is necessary to calculate all the coordinate points required, in the design stage itself. Initial material can take a number of forms, the most common of which are bar stock, plate, casting, forging, or may be just a slab of metal. With this raw material as a base, the process planner must prepare a list of processes to convert this normally predetermined

material into a predetermined final shape. Once the process planning is ready, the manufacturing can begin by preparing CNC part programs for CNC machines.

1.1 Computer-Aided Part Programming

Cutter path generation module is the focus of this dissertation. A CNC system consists of three basic components:

1. A program of instructions
2. A machine control unit
3. Processing equipment

The program of instructions is the detailed set of step-by-step commands that direct the actions of the processing equipment. In machine tool application, the program of instructions is called a part program. In this application, the individual commands refer to positions of a cutting tool relative to the work table on which the work-part is held. The program is coded on a suitable medium for submission to the machine control unit.

The machine control unit (MCU) consists of a micro computer and related control hardware that stores the program of instructions and executes it by converting each command into mechanical actions of the processing equipment, one command at a time. The MCU includes control system software, calculation algorithm, and translation software to convert the NC part program into a usable format for the MCU. Today, all MCUs are based on computer technology, hence computer numerical control (CNC) is referred to NC system.

The last component of NC system is the processing equipment. It accomplishes the processing steps to transform the starting work-part into a finished part. Its operation is controlled directly by the MCU, which in turn is driven by instructions contained in the part program. In NC machines, the processing equipment consists of the worktable and spindle as well as the motors and controls to drive them.

There are different types of movements accomplished by MCU whose features are explained below. MCU systems for NC can be divided into three types, namely

1. Point-to-Point
2. Straight cut.
3. Continuous path

Point-to-point systems, also called positioning systems, move the work table to a programmed location without considering the path taken to get to that location. Once the move has been completed, some processing action is accomplished by the work head at the location such as drilling or punching a hole.

Straight cut control systems are capable of moving the cutting tool parallel to one of the major axes at controlled rate suitable to the machining. With this type of NC system it is not possible to combine movements in more than a single axis direction.

Continuous path systems generally refer to systems, which are capable of continuous simultaneous control of two or more axes. This provides control of the tool trajectory relative to work-part. This enables the system to generate three dimensional contours in the work-part.

The term contouring is used when continuous path control is used for simultaneous control of two or more axes in machining operations. One of the important aspects of contouring is interpolation. The path that a contouring type NC system is required to generate often consists of circular arcs and other smooth nonlinear shapes. To cut along a curved path, the curve must be divided into a series of straight line segments that approximate the curve. The tool is commanded to machine each line segment in succession so that the machine surface closely matches the desired shape. The maximum error between the desired surface and the finished surface can be controlled by the length of the individual line segments which is one of the main tasks of this dissertation. These issues will be discussed more in detail in the following sections.

The computer's role in computer-aided part programming consists of the following tasks

1. input translation
2. arithmetic and cutter offset computations
3. editing
4. post processing

The input translation module converts the coded instructions contained in the program into computer-usable form, preparatory to further processing.

The arithmetic module consists of a set of subroutines to perform the mathematical computations required to define the part surface and generate the tool path, including compensation for cutter

offset. The individual subroutines are called by the various statements used in the part programming language. The arithmetic computations are performed on the PROFIL file. The arithmetic module frees the programmer from the time consuming and error-prone geometry and trigonometry calculations to concentrate on issues related to work-part processing. The output of this module is a file called CLFILE, which stands for "Cutter Location (CL) file". This file consists mainly of tool path data.

In the editing phase, the CLFILE is edited, and a new file is generated called CLDATA. CLDATA provides readable data on cutter locations and machine tool operating commands. The machine tool commands can be converted to specific instructions during post processing. Some of the editing of CLFILE involves processing of special functions associated with the part programming language. The output of the editing phase is a part program in a format that can be post processed for the given machine tool on which the job will be accomplished.

The final task is post processing, in which the cutter location data and machining commands in the CLDATA file are converted into low-level code that can be interpreted by the NC controller for a specific machine tool. The output of post processing is a part program consisting of G-codes, *x*-, *y*-, and *z*-coordinates, S, F, M, and other functions in word address format [1].

1.2 Types of Tool Paths

The commonly used tool path distribution strategies are [1]

1. Zig-zag or raster curves
2. Contour curves
3. Spiral curves
4. Space filling curves
5. Sequential generated curves

1. Zig-zag Curves:

The most commonly used tool path distribution strategies is the zig-zag strategy, due to the simple algorithm involved in calculating the spanning elements. This strategy involves filling the domain with parallel rays which are trimmed at the boundaries as shown in the fig 1.1.

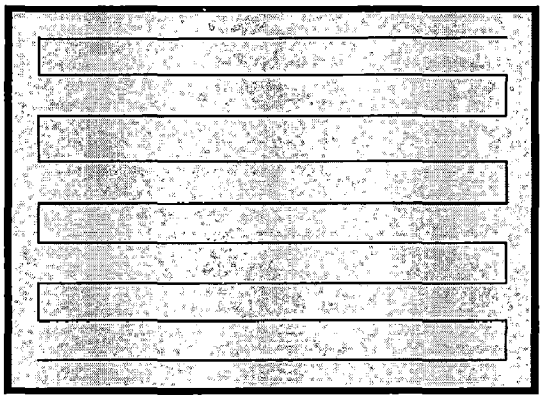


fig: 1.1 Zig-Zag path

2. Contour Curves:

The contour strategy is advantageous when the boundary contours are included as spanning elements, since a uniform boundary results. This strategy involves shrinking/expanding contours till the entire domain is spanned as shown in the fig: 1.2.

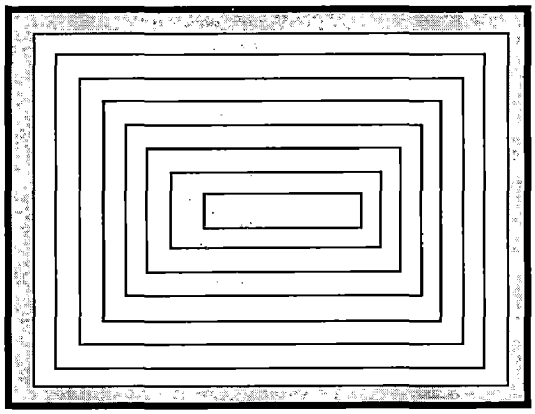


fig: 1. 2 Contour path

3. Space Filling Curves:

The previous strategies give directionality or lay to the surface finish on the manufactured part. The space filling strategy avoids the directionality by frequent changes in orientation of the spanning elements by means of recursive algorithms. The disadvantage of using the space filling strategy is that the overall length of the spanning elements in general is large, increasing manufacturing time. In addition, the number of short spanning elements is disadvantageous to the NC machine, since the tool is unable to accelerate to the specified feed rate. The space filling curves is as shown in the fig: 1.3.

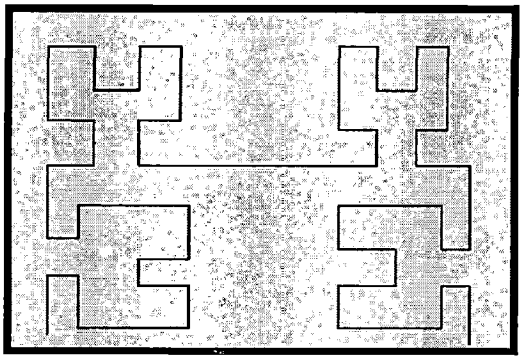


fig: 1.3 Space filling path

4. Sequential Curves:

This strategy involves sequentially generating spanning elements starting with a given initial spanning element. The sequential distribution strategy is advantageous due to the flexibility of generating various geometries of spanning elements. The disadvantage of this strategy is the complexity involved in calculating spanning elements.

fig: 1.4 schematically shows the above strategy of tool path distribution on a unit square by means of spanning elements. These spanning elements on a unit square are then calculated, such that they lie on the designed part, by means of tool path calculation methods discussed in the next section. It is in the tool path calculation methods that the spacing between the tool paths and geometric accuracy is determined.

fig: 1.4 Sequential path

The types of tool paths if we consider the shape of the stock are explained below:

a) *Stock-offset pattern (SO):* This is generated by moving the cutting tool along a series of curves offset inwards from the stock contour. Since there are only a few non-cutting movements in machining, the approach is reasonably efficient for most cutting shapes.

b) *Component-offset pattern (CO):* This is generated along a series of curves offset outwards from the component contour and the cutting-tool movement is limited within the stock boundary. This pattern is frequently used in practice owing to its simplicity in tool-path calculation. The disadvantage of this method is that it contains numerous non-cutting rapid traverse motions.

c) *Stock-component-offset pattern (SCO):* This is generated by a combination of the two previous patterns. The pattern is efficient where the stock and component profiles are similar. The disadvantage is that it also contains numerous non-cutting rapid traverse motions as well as numerous plunge and retract motions.

d) *Parallel-offset pattern (PO):* This is generated by moving the cutting tool along a series of offsets parallel to the longest edge of the stock. This pattern is most efficient when the component profile has no, or very small, islands. It becomes inefficient when a part has multiple islands or islands with seriously non-convex shapes.

e) *Max-min-offset pattern (MMO):* In this pattern, the cutting tool moves along a series of parallel offsets as it does in the parallel pattern. The difference is that there are two kinds of parallel offsets combined which have two orthogonal directions. When generating such a tool-path pattern, the cutting tool first moves along the maximum length edge of the stock, and then when an island arises in the tool path, it changes direction and moves along the minimum length edge. This pattern is suitable for parallel-piped stock with relatively large and long islands. For other cases, it is usually inefficient.

These tool paths are generally used when we consider the shape of the stock. But as discussed earlier, the shape of the stock is generally bar stock, plate, casting, forging, or maybe just a slab of metal. In this dissertation work the freeform surface is produced from cubic stock which encompasses the whole surface.

CHAPTER 2

LITERATURE REVIEW & PROBLEM FORMULATION

Tool selection and tool path is one of the most significant process considerations for deciding how a machined product will be manufactured. The tool selection activity has direct impact on the machinability and machine tool performance. Consequently the integration of tool path and tool selection in design is key step towards the goals of efficient manufacturing.

P.G.Maropoulos and R.P.Baker [2] says that among the varied and complex task of the process planning [3] the tool selection functionality can be considered important, since the influence of tooling related issues also affects the tool path as shown in fig: 2.1.

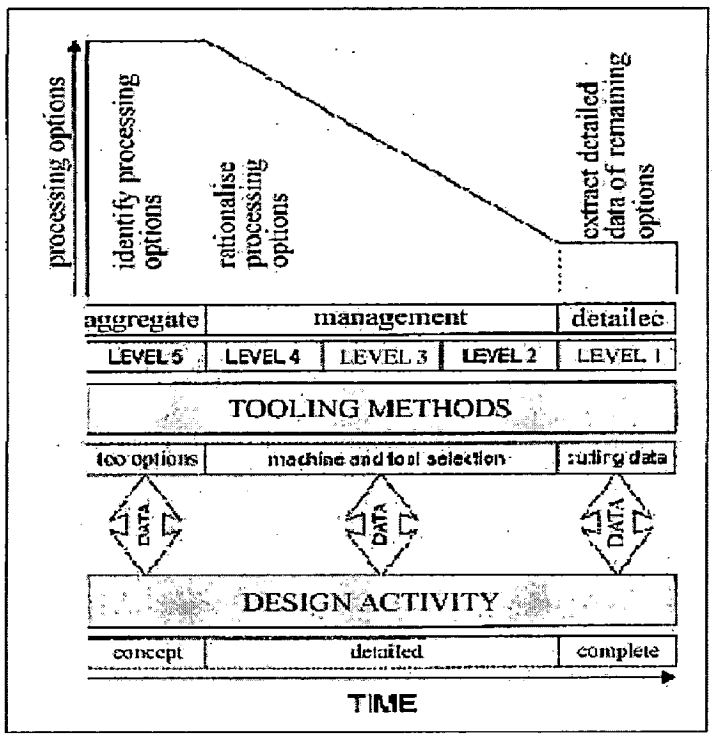


fig: 2.1 Vertical integration of tooling(VIT) methods within design and process planning [2].

Feature based design tool path:-

The common properties of features provide the data necessary to define the operations required to manufacture them. The most important factor is the approach direction of the tool, which describes the orientation of the feature and the direction of depth of cut as shown in fig: 2.2.

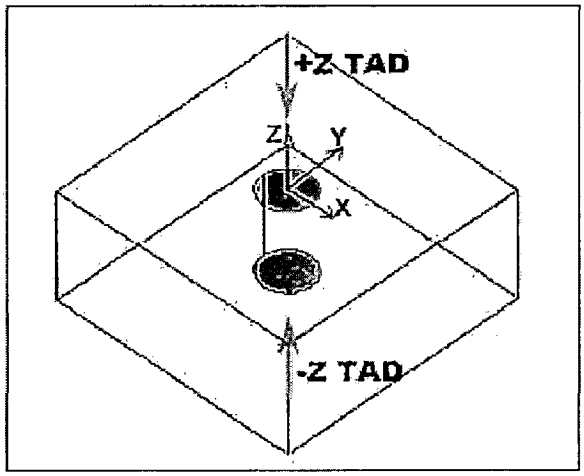


fig: 2.2 Possible tool approach directions(TAD) for a drilled through-hole [2]

The estimation of machining costs can involve two types of data requirement. The most accurate method requires specification of material, tools, work holding and details of the operations used to manufacture the component, where as the simplest method requires the shape and size of the initial stock material, the quantity of material removed and the cost of material removal per unit volume. A method that attempts compromise between the two is that, the machining costs are influenced by cutting conditions. So final conclusion is that machining cost is directly influenced by machining time which is influenced mainly by the choice of tools used [2].

Machining cost=machining time X machining cost per unit time

The machining time here can be calculated by the path lengths of the feature that is **tool travel lengths**. The research about the features is presented by **C.Ou-Yang and T.S.Lin** [4] in which machining times are estimated based on the type of feature.

Several researchers have worked on optimizing the machining parameters such as spindle speed, cutter travels rate, depth of cut, number of teeth [4][2][22]. Also choosing the optimal tool path can reduce the tool travel time considerably and also provided significant reduction of machining costs. **Wang et al** [5] have reported that different mill travel methods may result in dramatic variation of total length of cut.

The majority of industrial milling task can be performed using 2.5D milling. This is due to the fact that a large number of mechanical parts can be machined in 2.5D and even the more

complicated objects are usually produced by 2.5D roughing and 3D-5D finishing. Thus th computation of tool path for pocket machining is one of the most important issues in Compute Aided Manufacturing.

Bor-Tyng Sheen and Chun-Fong [6] have proposed automatic methods of recognition o machining features and tool path generation for them in 2, 2 ½, 3-axis CNC Milling machines. I which he has proposed algorithm for recognition of features by dividing the features simply a VPI (virtual pocket island) and RPI (real pocket island) covering a stock for the work piece. I this a virtual boundary is automatically added to the work piece and during slicing, the concret portion intersects with the slicing plane to form four faces. The faces compromise non machining region. The face profile and the virtual pocket profile forms the top portion of VPI a shown in fig: 2.3.

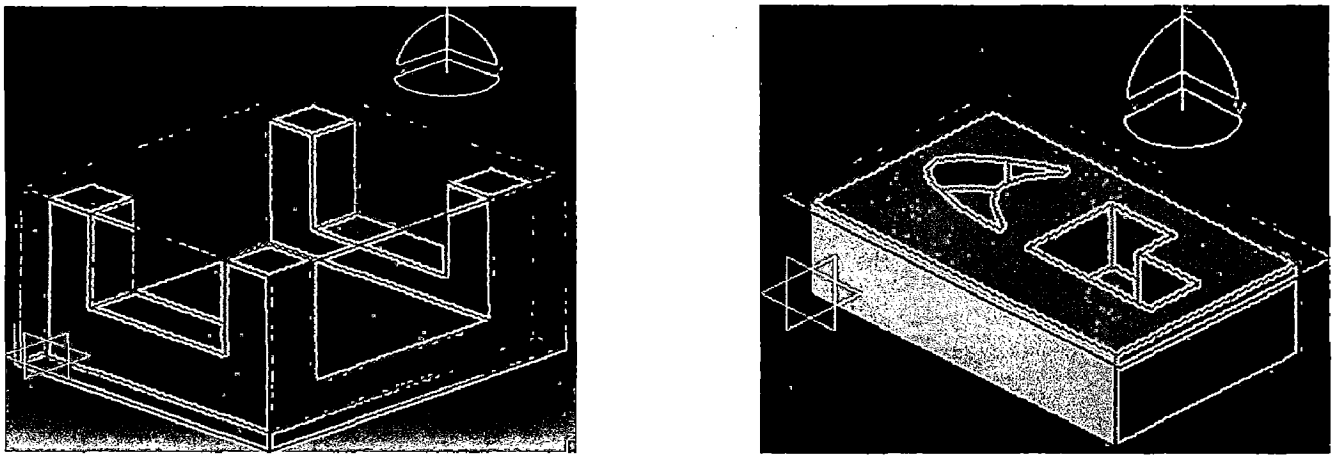


fig: 2.3 VPI & RPI

Unlike VPI, needed adding a virtual pocket profile, the RPI owns its pocket profile as illustrate in fig: 2.3.Different kinds of tool paths can be generated automatically for various machinin features to improve the cutter efficiency.

To produce complex shapes efficiently the cutter needs to change its application direction o orientation to reach the local areas. Advanced 5-axies machine provides a solution for this, but 3 axis machines are commonly used in manufacturing due to their low cost and high accuracy. I this 3-axis machine the machine tool simultaneously controls the relative movements betwee the part and the cutter along its three primary axes. A 5-axes CNC machine is more versatil because it controls five motions continuously and simultaneously. One of its unique advantage

is continuous adjustment of the cutter orientation while the tool is cutting. ***Zezhong C.Chen et al*** [7] has proposed a cost effective 3 ½ ½ axis machine instead of 5-axies for sculptural parts. This reduces the machining time [7].

According to **Y.N.Hu et al** [8] when milling a sculptural part, different tool-paths patterns will take different machining times. The total machining time T, is dominated by two terms, namely cutting time T_c and the tool rapid traverse time T_m. The total production time is then

$$T1=\sum^{N}t_i=\sum^{N}t_{ci}+\sum^{N}t_{mi} \qquad (2.1)$$

Where

N= number of layers.

t_i = total production time of i th layer.

t_m = tool rapid traverse time of ith layer.

t_{ci} = cutting time for ith layer.

So far, there are two major approaches in CNC tool path generation for rough machining of sculptured cavity parts: the offsetting approach and the contour-map approach. The offsetting approach is to calculate the point trajectories which represent the offset surfaces of a sculptured part, and a ball-end cutter will go through such point trajectories during machining. When the shape of the cavity part and the shape of the stock are quite different, a considerable number of blank cuts may be created.

The second method uses planes to slice the stock so as to obtain a number of parallel cutting layers. Until recently, the surplus volume cut at each layer was still determined by a fixed cutting depth and the intersecting curves of the slicing plane with the part and the stock. Cuts in each layer follow the two-dimensional offset approach. When using this approach, the tool paths are easily generated only if the interference and accuracy calculation of cut are ignored. Therefore, both of these approaches are not possible for obtaining high productivity in rough machining and should be improved to solve the problems stated above,

Y. N. Hu et al. [8] described a knowledge-based reasoning system to optimize the machining parameters for rough machining. The advantages and drawbacks of various feasible tool-path patterns, including stock-offset, component-offset, stock/component offset, parallel offset,

proportional blending offset, and max-min-offset, as well as their performances on different cutting-layer shapes are evaluated. The knowledge acquired from the evaluation of tool path patterns is modeled and implemented in a knowledge-based system. Parameters that are used to analyze the layer shapes and methods that are used for planning tool path patterns are also described. As a result, for a part to be machined, the optimal multiple-tool-path pattern, called the comprehensive tool-path pattern can be generated for every given 2D cutting-layer shape using this knowledge based system. The slicing process is as shown in fig: 2.4.

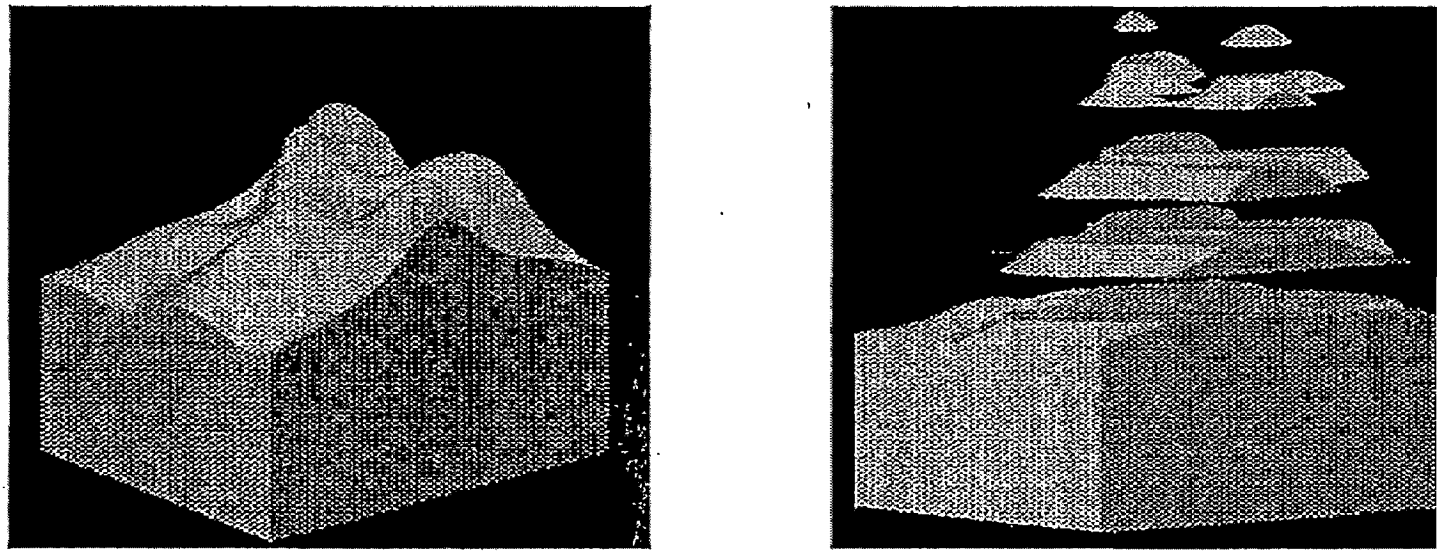


fig: 2.4 Slicing the work piece [8]

Peng WU et al [9] have proposed a methodology of path generation for the subdivision surfaces. The path planning for rough cut is proposed by Z-map model. This model divides the height of work piece at the grid points and stores it in an array. Then the grid generation along X and Y directions are determined, and the tool path pattern for rough and finished cuts are generated using the slicing technique by calculating every grid point.

Chun-Fong You and Chih-Hsing Chu [10] have described grid height method for the generation of tool path generation of NC rough cut machining for solid models and the freeform surfaces by grid height in which the grid is generated for the component. Each grid represents the square range in the X-, Z-, plane corresponding to the element in the spatial array, then proper height is computed to approximate contact surface at the range of X-, Z- plane. Subsequently

different methods like interference detection, pocket machining by offset path generation, pair-wise offset and uneven offset are used for free pocketing [11].

Dave Carswell and Nick Lavery [12] have proposed a computer aided design (CAD) tool specifically developed for rapid and easy design of solid models for surfboard and sailboard fins. This tool simplifies the lofting of advanced fin cross-sectional foils, in this instance based upon the family of standard airfoil series set by the National Advisory Committee for Aeronautics (NACA), whilst retaining a basic parametric description at each cross-section. This describes the way in which B-spline surfaces are created from 2D profile splines, and are then used to generate 3D geometrical surfaces of the fins.

W. BiJhm and Braunschweig [13] has proposed Cubic B-Spline Curves and Surfaces in Computer Aided Geometric Design. Representations of cubic and bicubic splines are given, combining the advantages of B-splines with the handiness of Bezier technique. The Bezier points of spline curves and surfaces are found by forming convex combinations of nodes. The given algorithms are suited especially for computer aided geometric design

The methods explained up to know for free form surfaces are computationally hard to calculate. **Peng WU et al** [9] have explained about the Z-mapping model, but in this every grid point is to be calculated for rough and finish cuts therefore the tool path becomes computationally hard. **Chun-Fong You and Chih-Hsing Chu** [10] also proposed the grid height method in which too grid points are necessary to be calculated [9] and tool path generation for rough cuts is complex. The methods presented in [8]-[10] are complex for the calculation of rough cuts and may involve numerical methods to calculate. Different tool paths also proposed for minimizing by **Y.N.Hu et al** [8].

All this computational hardness must be removed so that the rough machining time should be reduced and to get desired surface. We consider a free form surface and proper mathematical equations are used to compute tool path trajectory without any over cutting, to remove maximum unwanted material from the blank. As reported in the published work [6]-[13] the approximation of freeform surfaces in to grids does not produce the actual surface if the grid size is considerably large. And generally this grid shape (square) may not approximate the corners of

the freeform surface. So an approach is needed, which represents the freeform surface and generates the tool path.

Milling operation is the primary machining process used in the manufacturing of freeform surfaces and is divided into two stages, the rough stage and the finish stage. In rough stage, the part is machined in incremental layers and then cutter removes most of the material on the surface so as to avoid damage of tool or/and machine. In finish stage, the surface is machined smoothly by approximating surface using line segments to get desired part with predetermined accuracy and shape. The tool paths in finish stage are important, since the tool path directly affects the accuracy and manufacturing time of the manufactured part.
In milling operation, a rotating spindle touches the surface at cutter contact point (CC) and moves to next CC point linearly, that is a curved path is approximated by a straight line segment. The accuracy of this linear approximation depends on the length of the linear segments and is governed by the tolerance specified for the concerned surface.

When we design a part, one of the main tasks is defining geometry of the part. A surface is the image of a sufficiently regular mapping of a set of points in a domain into a 3D space and expressed as

$$Q(u, v) = (x(u, v); y(u, v); z(u, v))$$

When the domain of a surface is the xy-plane of the given Cartesian coordinate system, the parametric surface equation given by

$$Z = f(x, y)$$

In this research, we develop an efficient approach to generate tool paths for NC machining of free-form surfaces. As such the primary goals of this research are

1. Extracting the data from the CAD model.
2. Slicing the freeform surface horizontally layer-by-layer to improve the machining efficiency. The reason is that planar path is more advantageous and moreover it reduces a 3D problem to series of 2D problems.

3. Developing a tool path generation module which will generate the zig-zag tool path for each layer based on its level of details. This module is applied for both roughing and finishing operations.
4. Developing CL file generation module in which a CL file is generated which contains the co-ordinate movements of tool that is the TOOL PATH to get the final desired shape.

CHAPTER 3

MATHEMATICAL REPRESENTATION OF FREEFORM SURFACES

The fundamental issue here is to determine an appropriate mathematical/computational framework for representing freeform surfaces that would be capable of performing necessary geometrical operations.

3.1 Modeling of Part

Three-dimensional, or space curves and surfaces play an important role in the design and manufacture of a diverse range of products, e.g., automobile bodies, ship hulls, aircraft fuselages and wings, propeller blades, etc. One sort of technique for curve or surface generation constrains the curves to pass through existing data points. They are commonly known as "*curve fitting*" techniques. They are particularly suited to shape description, where the basic shape is arrived at by experimental evaluation or mathematical calculation. One common example is cubic spline curve & surface. However there is another class of shape design problem that depends on both aesthetic and functional requirements. Examples are the skin of car bodies, sculptures, shoes and other fashionable wear, ornaments, decorative items etc. For such class of design, control points are used to define the curves. These methods are frequently referred to as "*curve fairing*" techniques [14]. In this work curve fairing techniques are used to model the complex surfaces. The specific advantage of using this class of surfaces is that the user can interactively change the shape of the surface by simply shifting the respective locations of the control points or some other simple aspect of the surfaces.

3.2 Introduction to Curves

Three dimensional or space curves such as roof top of car, the fuselage of an aircraft or wash basin can be created by motion of curves in space in a specified manner. This may be sweep, revolution and smooth merging of discontinuities.

The principles of curve design are as follows

- The *shape* of the curve should be controlled by placing only a few number of data points. The curve created should behave like an elastic string that a designer can manipulate to give a desired shape.

- The curve should be *synthetically* composed of polynomials of low degree to avoid undue oscillations and minimize computation time and complexity.
- The curve model should have *affine property* ensuring shape independence from the coordinate frame of reference. This makes it possible to treat the curve as the real model in space.
- Since in the real design and manufacturing complex shapes have to be modeled and manufactured, it is more suitable to join together several segments of curves, fulfilling position slope and curvature continuity.
- *Parametric* description is preferred over implicit or explicit forms as it provides an articulated representation of curve segment in three dimensions.

3.2.1 Representation of space curves

Three dimensional space curves are represented non-parametrically or parametrically. An explicit non- parametric representation is

$$x = x$$

$$y = f(x)$$

$$z = g(x)$$

In general a parametric space curve is represented as shown in fig: 3.1

$$x = x(u)$$

$$y = y(u)$$

$$z = z(u)$$

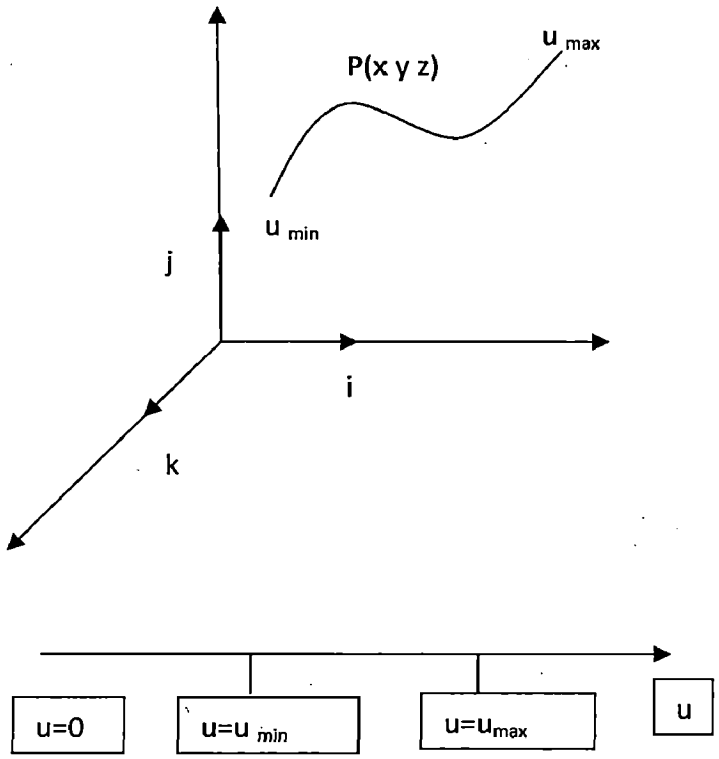


fig: 3.1 curve representation

Where parameter t varies over a given range $u_1 \leq u \leq u_2$. Considering the explicit non-parametric representation above we note that x itself can be considered a parameter that is

$$x = u$$

$$y = f(u)$$

$$z = g(u)$$

Synthetic curves are suitable for designing complex shapes that may not be represented by analytical curves. Mathematically these are polynomial representation which provides more control and may be derived from a given set of data points via interpolation of curve fitting. The various types of synthetic curves are Bezier curve, B-Spline curve which will be explained in subsequent sections.

3.3 Bezier Curves and Surfaces

Bezier curves are used in computer graphics to produce curves which appear reasonably smooth. Mathematically, they are a special case of cubic Hermite interpolation, whereas polygonal lines use linear interpolation. In this case curves are constructed as a sequence of cubic segments, rather than linear ones. Hermite interpolating polynomials are constructed in terms of derivatives at endpoints, whereas Bezier curves use a construction, in which the interpolating polynomials depend on certain control points. These curves are used very frequently in computer graphics because they are very easy to construct and the programming is easy as well. The main properties of Bezier curves and surfaces are given [15] below:

- The curve/surface does not generally pass through the control points except for the corners of the control point grid.
- The curve/surface is contained within the convex hull of the control points.
- The degree of polynomial defining the curve is always one less than the number of control points.

fig: 3.2 shows the different types of Bezier curves.Mathematically a parametric Bezier curve is define by equation 3.1:

$$p(u) = \sum_{i=0}^{n} B_i J_{n,i}(u) \qquad 0 \le u \le 1 \qquad (3.1)$$

Where

$$J_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad \text{and} \quad \binom{n}{i} = \frac{n!}{i!(n-i)}$$

$j_{n,i}(u)$ is the i^{th} , n^{th}-order Bernstein basis function. Also 0^0 =1 and 0! = 1 is adopted.

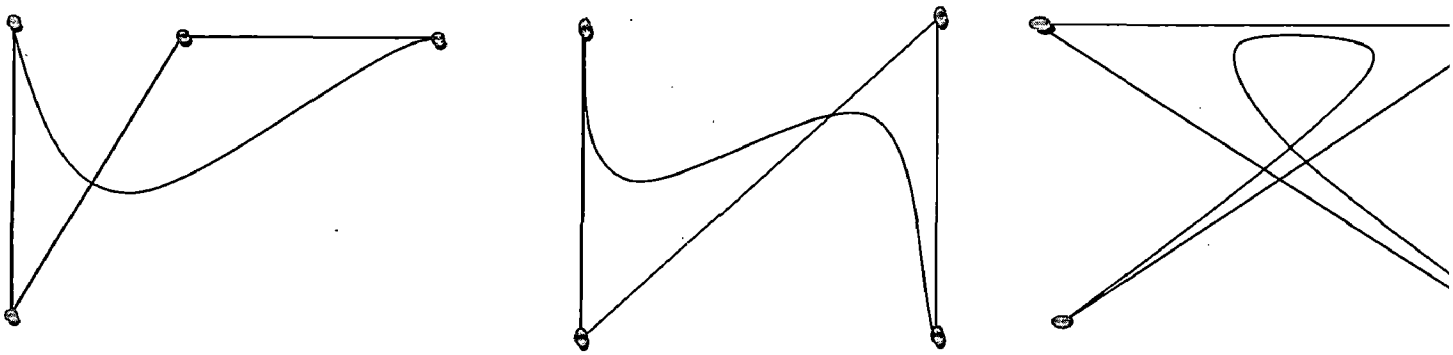


fig: 3. 2 Bezier curves for cubic's

The Bezier surface is formed as the Cartesian product of the blending functions of two orthogonal Bezier curves as shown in fig: 3.3 and is represented by equation 3.2:

$$Q(u, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} J_{n,i}(u) K_{m,j}(w) \qquad (3.2)$$

Where

$$0 \le u \le 1$$

$$0 \le w \le 1$$

Where $j_{n,i}(u)$ and $K_{m,j}(w)$ are the Bernstein basis functions in the u and w parametric directions. The B_{ij}'s are the vertices of the defining polygon net. The indices n, m are one less than the number of polygon vertices in u and w directions respectively.

$$J_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-i} \qquad \binom{n}{i} = \frac{n!}{i!(n-i)}$$

$$J_{m,j}(w) = \binom{m}{j} w^j (1-w)^{m-j} \qquad \binom{m}{j} = \frac{m!}{j!(n-j)!}$$

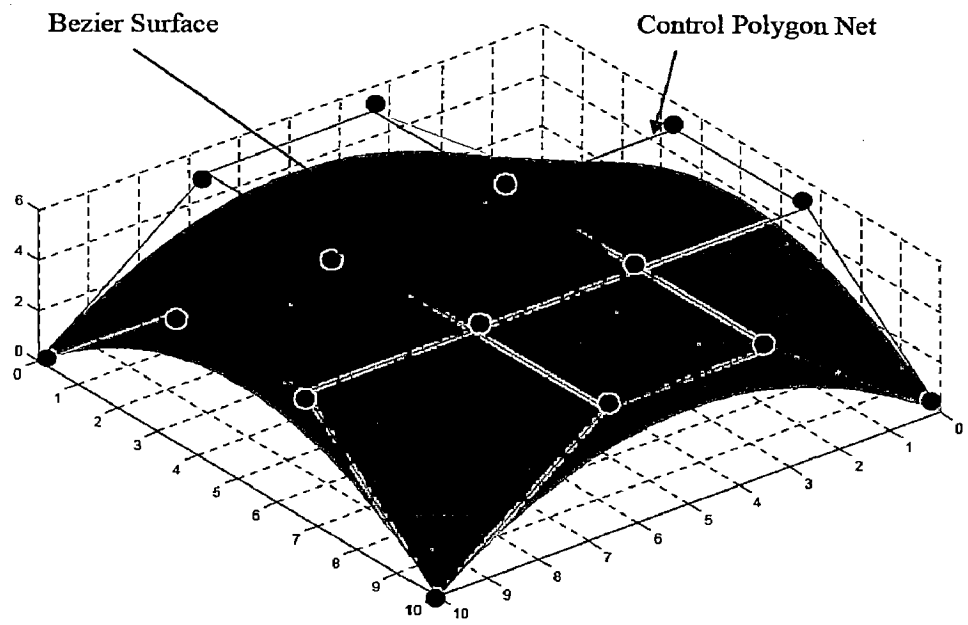


fig: 3.3 Bezier Surface with 4 x 4 polygons net

3.4 B-spline Curves and Surfaces

B-spline curves share many important properties with Bezier curves, because the former is a generalization of the latter. Moreover, B-spline curves have some properties that are considered better than those of Bezier curves. The properties of the Bernstein basis function, as listed below, limit the flexibility of resulting Bezier curves and surfaces:

1. Number of specified polygon vertices fixes the order of resulting polynomial which defines the curve.
2. The ability to produce a local change within a curve is eliminated due to global nature of the Bernstein basis function.

There is another basis called B-spline basis, which contains the Bernstein basis as a special case. B-spline curves require more information (*i.e.*, the degree of the curve and a *knot vector*) and a more complex theory than Bezier curves. But, they have many advantages to offset this shortcoming:

1. A B-spline curve can be a Bezier curve.
2. B-spline curves satisfy all important properties that Bezier curves have.
3. B-spline curves provide more control flexibility than Bezier curves can do.
   For example, the degree of a B-spline curve is separated from the number of control points. More precisely, we can use lower degree curves and still maintain a large number of control points. We can change the position of a control point without globally changing the shape of the whole curve (local modification property).
4. Since B-spline curves satisfy the strong convex hull property, they have a finer shape control as shown in fig: 3.4. Moreover, there are other techniques for designing and editing the shape of a curve such as changing knots, knot insertion etc.

In parametric form a B-spline curve is given by equation 3.3:

$$Q(u) = \sum_{i=0}^{n} B_i N_{ik}(u) \qquad u_{\min} \le u \le u_{\max} \qquad 2 \le k \le n \qquad (3.3)$$

Where

$$N_{i,1}(u) = 1 \; if x_i \le u \le x_{i+1}$$

= 0 otherwise

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

The values of x_i are the elements of a *knot vector* satisfying the relation $x_i \leq x_{i+1}$. The fig: 3.5 shows B-spline surface and is given by equation 3.5:

$$Q(u, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} N_{ik}(u) M_{jl}(w) \qquad (3.4)$$

Where

$$N_{ik}(u) = \frac{(u - x_i)N_{ik}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1k-1}(u)}{x_{i+k} - x_{i+1}}$$

$N_{ik}(u) = 1$ if $x_i \leq u \leq x_{i+1}$

= 0 *Otherwise*

$$M_{jl}(w) = \frac{(w - y_j)M_{jl}(w)}{y_{j+l-1} - y_j} + \frac{(y_{i+k} - w)M_{j+1l-1}(w)}{y_{j+l} - y_{j+1}}$$

$M_{jl}(w) = 1$ if $y_j \leq w \leq y_{j+1}$

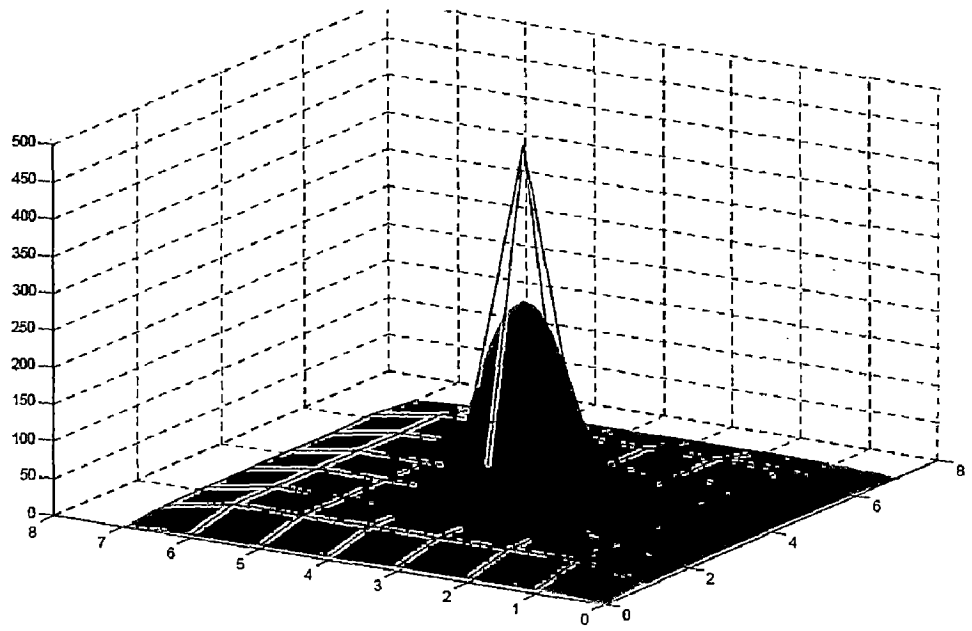


fig: 3.4 Local control of B-spline surface

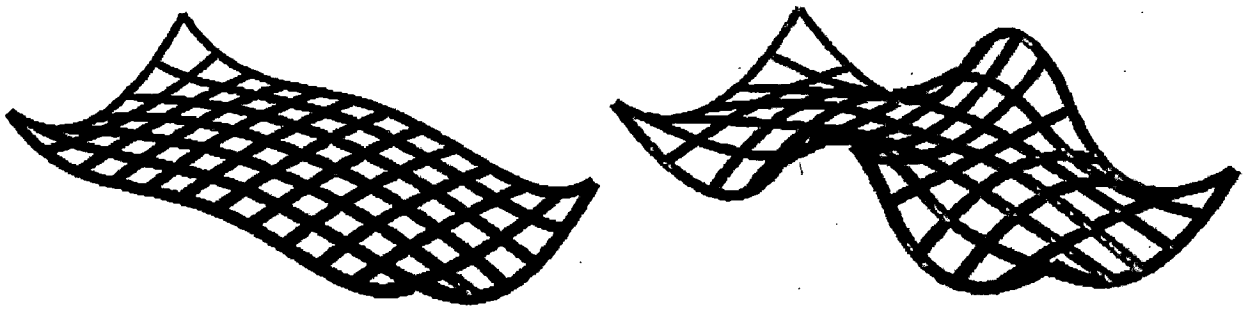


fig: 3.5B-spline surface

CHAPTER 4
DATA EXTRACTION

A new international data exchange standard STEP (Standard for the Exchange of Product Model Data) standard contains the product data covering the entire product life cycle and has a neutral format that is independent of any software package and is not restricted to any particular hardware platform.

The base line functionality for exchange STEP product data is achieved via a physical file containing data instances according to the model schema. The STEP file structure is language based and is described by an unambiguous context free grammar to facilitate parsing by software. The grammar is expressed in Wirth syntax notation. The information contained with the file is in free format and thus not column dependent. The file is organized in modular manner and consists of several sections. The STEP file is begun by key word ISO-10303-21and is terminated by keyword END-ISO-10303-21, and in similar fashion sections are delimited by keywords.

The content of a section is limited to the entity instances, i.e., the description of object of interest. Briefly the data format is as follows. Each entity instance has an identifier of the form #N. where N is the unique integer. Each individual entity has a name. The data for an entity instance follows the type name and is enclosed in parentheses. A datum can be either "primitive" like integer, real or string, etc., or it may be reference to an other entity instance within the file. Such a reference has the form #N where N is the entity number of reference instance. Entities may be referred before they are defined within the file.

A STEP file consists of three types of data namely: Descriptive, Geometrical and Topological and is divided in to two major sections: Header section and Data section as shown in the fig: 4.1. The information about the STEP translator version and the type of CAD software used to build the model is included in the header section. The Data section consists of geometrical entity definition and topological elements like faces, loops and bounds. Reference between elements is provided by instance ids or pointers (which may be nested). These instance ids or pointers by

themselves have no semantic meaning except to identify an instance in a STEP file. The sequence of instance in a STEP file is not specified by the standard.

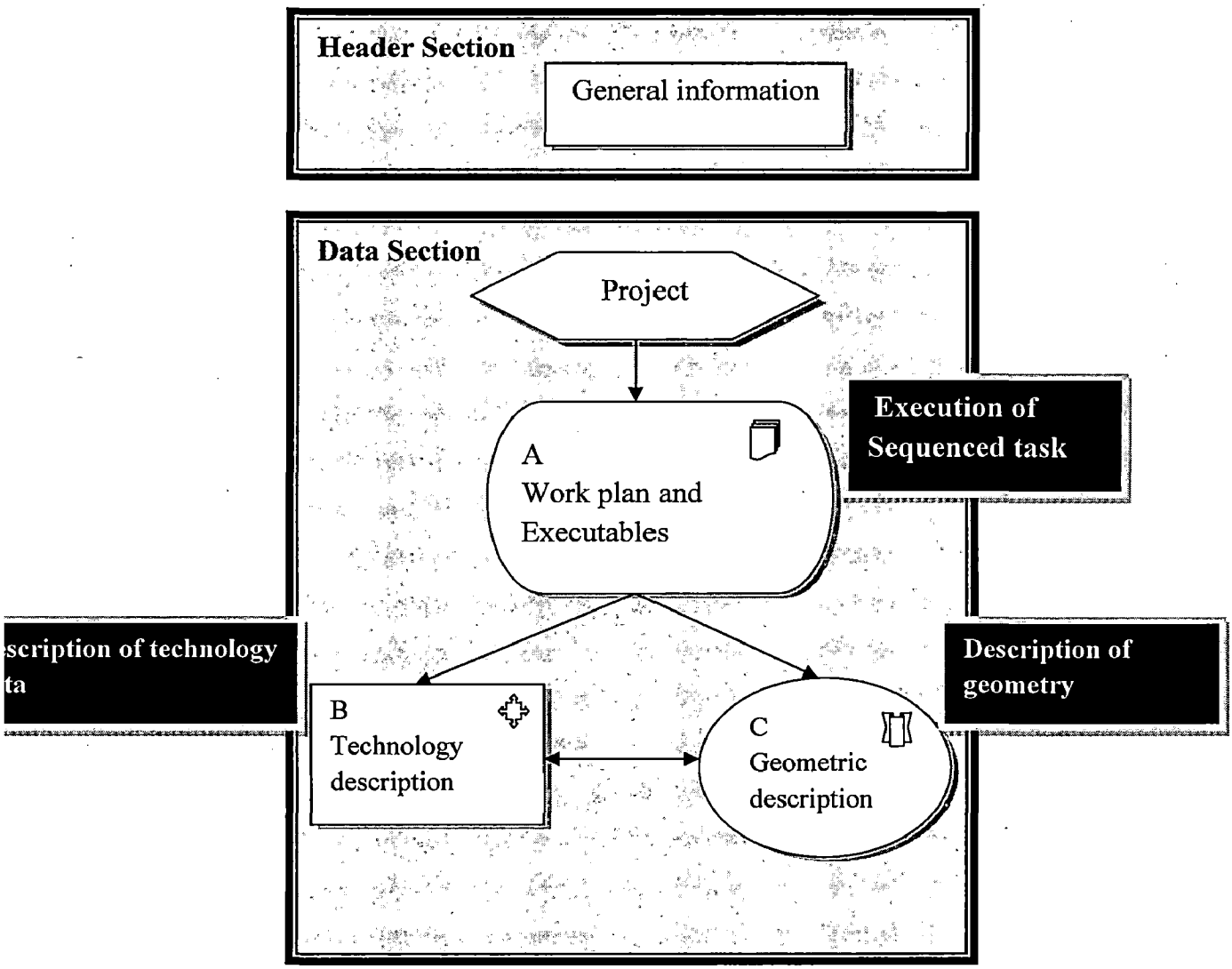


fig: 4.1 structure of STEP file

The STEP standard consists of many parts. Part 42 which is a resource for boundary representation of a product model, is of particular interest in tool path planning. A brief description of some STEP elements is provided below while more detailed definitions along with their attributes are available in the ISO/TC/184/SC4N141 Committee Draft Standards[17][18] The entire model is represented by variety geometrical entities and topological elements arranged in data section.

The entire data in the step file is stored in a hierarchical tree-like structure in bottom-up fashion. When considering machining planning of single part the root element of the tree can be considered to get the details. The data required for specific application can be extracted starting from the top element to the bottom element by descending down the tree, therefore the entire STEP file of the product can be described as an inverted tree like structure with a functional element at the top and geometrical element at the bottom. Other functional elements in between will be arranged in between. This structure is illustrated in fig: 4.2.

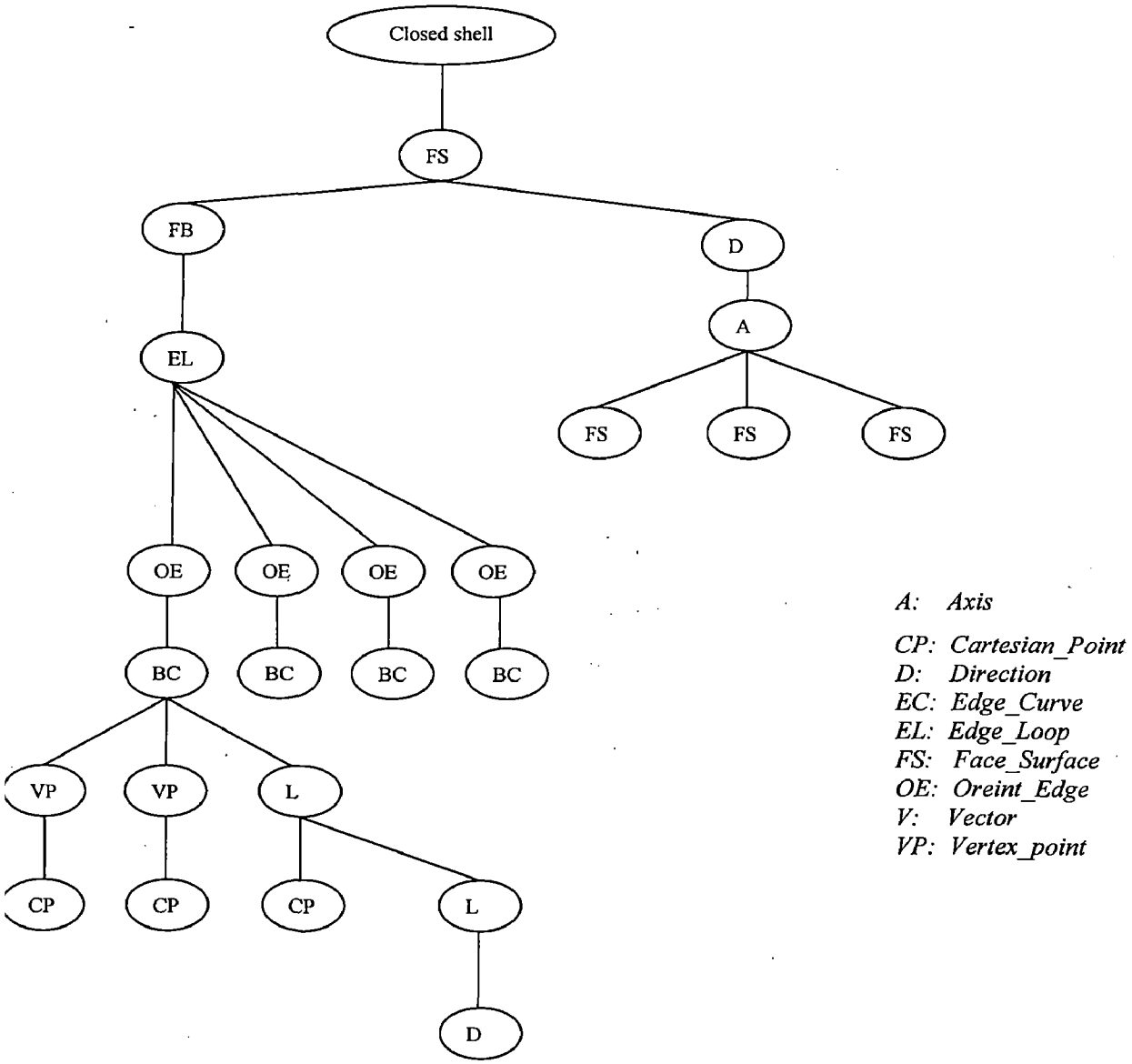


fig: 4.2 hierarchical structures of entities

This STEP standard provides a large amount of product information throughout the life cycle of a product which includes the data needed for the design, manufacturing and business applications. It is essential to develop an efficient data extraction for specific application such as process planning and tool path planning. For this purpose a hierarchical approach is proposed for data extraction in this research. To ensure the proper connectivity among the different functional elements all the associated pointers are traced in this process. Details of this are explained in the next section

4.1 Requirements for Free-Form Shape Entities

The B-spline curve and B-spline surface entities, as defined for the STEP standard, will be explored in some detail. These entities provide the data-exchange capability for freeform curves and surfaces. This capability was required to be on at least the same level as that already offered by IGES, SET. In particular, the following technical requirements were set for free-form shape design with STEP:

- Commonly used representations such as Bezier curves, power-basis coefficients and B-splines must be supported.
- The number of different entities must be kept to a minimum.
- No exchange of redundant data.
- High numerical stability.
- Automatic implementation-software generation must be supported(EXPRESS).

The equation provided in the STEP document for the B-spline curve is given by equation 4.1:

$$Q(u) = \sum_{i=0}^{n} B_i N_{i,k}(u) \qquad 0 \le u \le u_{max} \tag{4.1}$$

Where n is the upper index on the control points B_i (i.e. there are n + 1 control points) and $N_{i,k}$ (u) are the normalized B-spline basis functions

Similarly, for the B-spline surface given by equation 4.2:

$$Q(u,w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} N_{ik}(u) M_{jl}(w) \tag{4.2}$$

$$N_{ik}(u) = \frac{(u - x_i) N_{ik}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1k-1}(u)}{x_{i+k} - x_{i+1}}$$

$N_{ik}(u) = 1 \quad \text{if } x_i \leq u \leq x_{i+1}$

$= 0$ *Otherwise*

The actual data that need to be exchanged for B-Spline surface is

- The 3D control points (polygon or polyhedron)
- The value(s) of the algebraic degree(s),
- The knot vector(s).

4.2 STEP definition of the B-spline curve and surface

For the STEP standard free-form curves and surfaces will be represented exclusively by B-spline curves and the B-spline surfaces. That is the definitions of Equations 4.1 and 4.2 are adopted. No separate Bezier or monomial representations are provided, and there are no separate entities for specific B-spline types. It is shown below that the two STEP B-spline entities are multipurpose efficient. Generally these functions fall into two categories: default-value generators (FUNCTION default ...) and attribute-value checkers (FUNCTION constraints...). The constraints functions check the limits of the parameters (degree, number of control points, number of knots etc.) and the consistency between them. The entity B spline_curve has three non optional attributes: degree, upper_index on control_points and control_points, corresponding to n, k and B_{ij} respectively. This means that, during actual data exchange, these three data items must physically exist, or else the entity (and therefore the data communication) is incorrect. However, there are many other situations that lead to an invalid B-spline entity, owing to the WHERE clause. It is the purpose of the EXPRESS B-spline declaration to define precisely which mixtures of present and absent attribute values are valid. The starting point for this is that, mathematically, a B-spline curve is only well defined when all the symbols on the right-hand side of Equation 1 have a value (i.e. including the knots u, and the weights w,). The necessary condition for the B-spline curve is that each variable in Equation 4.2 has either an explicit value from the data or is properly derived from other variables. The consistency of the entity for the four B-spline curve cases is verified below in Table: 1.

B-spline curve attributes	STEP physical data
Degree	B_SPLINE_CURVE.,degree
(Number of control points -- 1)	*(#N),#(N),.....*
Number of knot multiplicities	*F.,U.,(value, value)*
Knot multiplicities	*(value)*
Knots	*unspecified F.,U., (value, value)*

Table 1: Example of data needed for B-spline representation with STEP

The above procedure is summarized by the following algorithm I used to extract data for Tool path generation.

Algorithm 1(Extract geometric data from STEP *part file)*

Read STEP file

If the number of elements in the file > 0

then record the pointer addresses of all elements (i.e., the freeform surfaces) in the B_SPLINE_SURFACE and return the number of elements found: NC, *continue:*

else return message " The drawing file is empty ".

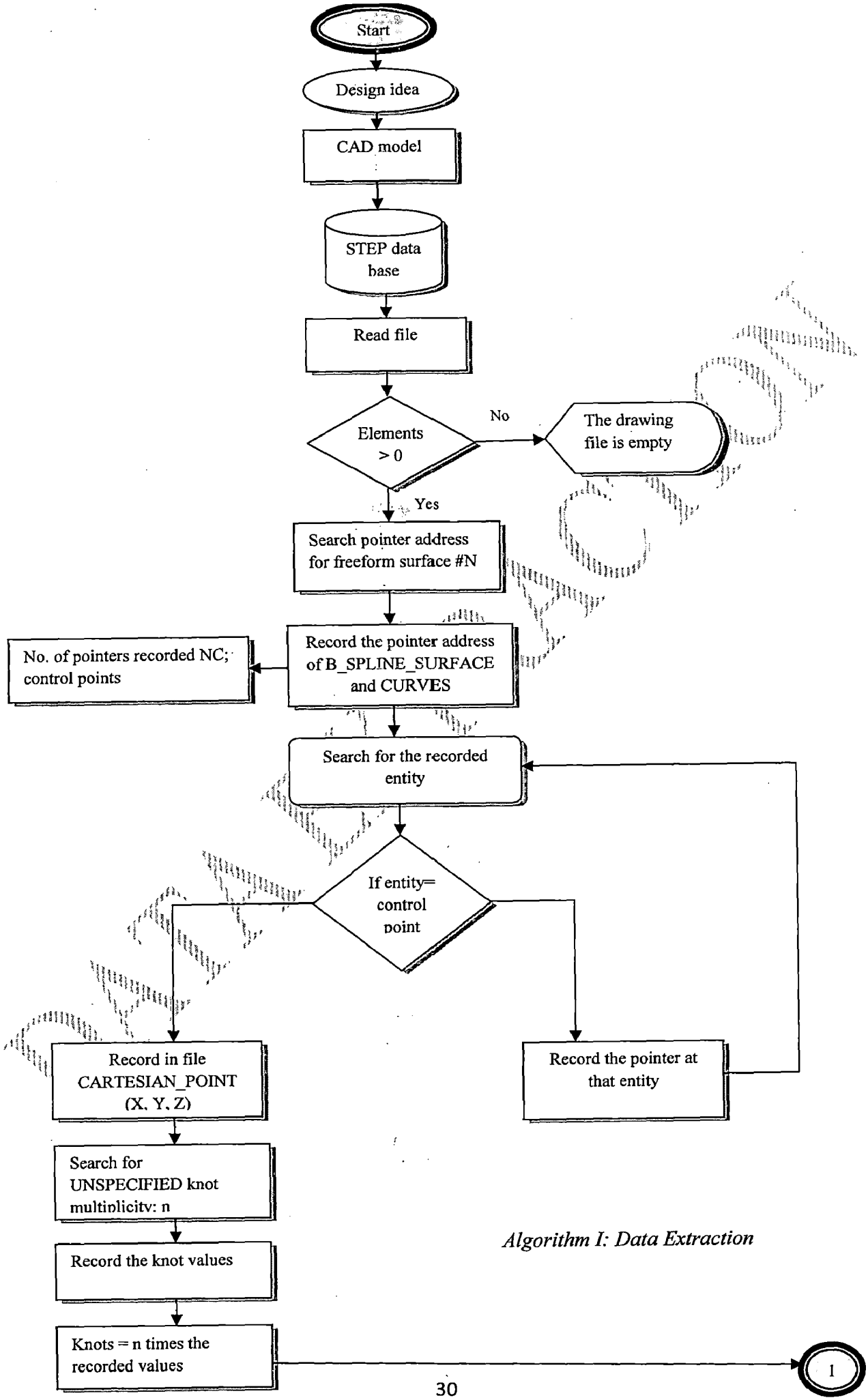
DO record the pointer addresses of all groups of entities and return number of elements found in one group: NP : DO j=1 to NC

if the element referred are the pointer addresses of control points then refer to the CATESIAN_POINTS entity in that referred element. Else continue;

Do record the pointer address of associated CARTESIAN_POINT and return its elements (X, Y, Z).

Do refer to the UNSPECIFIED parametric direction knot multiplicities: n and the knot values.

if if knot multiplicity>0 store n times of the knot value in an array



Algorithm I: Data Extraction

Example:

Let us take an example of freeform surface as shown in fig: 4.3.

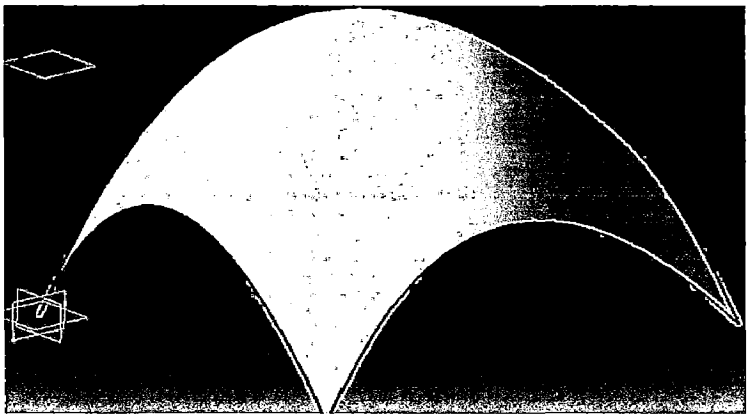


fig: 4.3 Example freeform surface

The STEP file for this model is as shown in fig: 4.4

freeformsurface - WordPad

File Edit View Insert Format Help

```
#167=ORIENTED_EDGE('',*,*,#162,.F.) ;
#170=OPEN_SHELL('Fill.1',(#169)) ;
#171=SHELL_BASED_SURFACE_MODEL('NONE',(#170)) ;
#49=SHAPE_REPRESENTATION(' ',(#48),#46) ;
#169=ADVANCED_FACE('',(#168),#114,.F.) ;
#4=APPLICATION_PROTOCOL_DEFINITION('international standard','config_control_design',1994,#1) ;
#32=APPROVAL_DATE_TIME(#13,#21) ;
#51=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',5,(#52,#53,#54,#55,#56,#57,#58,#59,#60),.UNSPECIFIED.,.F.,.U.
#69=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',5,(#70,#71,#72,#73,#74,#75,#76,#77,#78),.UNSPECIFIED.,.F.,.U.
#84=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',5,(#85,#86,#87,#88,#89,#90,#91,#92,#93),.UNSPECIFIED.,.F.,.U.
#99=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',5,(#100,#101,#102,#103,#104,#105,#106,#107,#108),.UNSPECIFIED
#131=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',3,(#132,#133,#134,#135),.UNSPECIFIED.,.F.,.U.,(4,4),(0.,112.
#141=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',3,(#142,#143,#144,#145),.UNSPECIFIED.,.F.,.U.,(4,4),(0.,104.
#149=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',3,(#150,#151,#152,#153),.UNSPECIFIED.,.F.,.U.,(4,4),(0.,112.
#157=B_SPLINE_CURVE_WITH_KNOTS('BSpline Curve',3,(#158,#159,#160,#161),.UNSPECIFIED.,.F.,.U.,(4,4),(0.,104.
#114=B_SPLINE_SURFACE_WITH_KNOTS('BSpline Surface',3,3,((#115,#116,#117,#118),(#119,#120,#121,#122),(#123,#
#11=CALENDAR_DATE(2008,9,4) ;
#30=CC_DESIGN_APPROVAL(#21,(#16,#6,#14)) ;
#18=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#13,#17,(#16)) ;
#29=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#13,#28,(#14)) ;
#17=DATE_TIME_ROLE('classification_date') ;
#28=DATE_TIME_ROLE('creation_date') ;
#27=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#26,(#16)) ;
#33=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#34,(#6)) ;
#35=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#36,(#6,#14)) ;
```

For Help, press F1

fig: 4.4 STEP file

The freeform surface shown in fig: 4.3 is represented as B-spline surface in the STEP file. The data required to be extracted from the STEP file are: its control points, curves, parametric values, knots, number of knots and its degree in specified parametric direction. All the numbers (#N) act like pointer. The surface in the step file is represented as B_SPLINE_SURFACE_WITH_KNOTS which is addressed with a pointer (#N). Each pointer again addresses to a pointer or its root elements. As shown in the fig: 4.4 the pointer (#114 which is required surface) refers to a bunch of pointers (#N) again. In Algorithm I, we search all the pointers and fetch the required parameters of the B-spline. We initially find all the pointers (#N) which represent the B-spline knots of the surface, after which we search the pointers that represent the B-spline surface. Once we achieve these pointers, we get the information regarding the B-spline surface such as control points, knots, knot multiplicity and degree of the surface. These attributes are recorded for further use in B-spline surface calculations (described in next chapter). The above work of searching on STEP file was carried out using C.

CHAPTER 5

SLICING B-SPLINE SURFACE

Within STEP files surface models may be represented as B-spline surfaces with parameters u and w ranging from zero to one. The Pi,j terms are 3D net points of the defining polygon and Ni,k and Mj,l are B-spline basis functions of order k and l respectively. The B-spline basis functions are defined by the Cox-deBoor recursion formulas.

The values of x_i are elements of a knot vector satisfying the relation $x_i \leq x_{i+1}$ and the convention $0 / 0 = 0$ is adopted. The STEP file contains the net points, knot vectors and upper index of both sums. With this information a 3D point on the surface is specified by its parameters u and w. To slice the B-spline surfaces, the model is first re orientated so that the normal of the cutting plane specified by the user becomes the z axis of the CAD model. Each contour start point is found by setting parameter u to zero and finding the value of w that gives a z co-ordinate equal to the current layer height. An iterative function solving method with adaptive step size is used to find w. Once u and w are known, the x and y co-ordinates can be found. The contour is then traced by increasing parameter u by the inverse of the number of points used, and finding the new w value to satisfy the contour height. fig 6.1 shows the slicing and algorithm II slices the freeform surface and generates the points required.

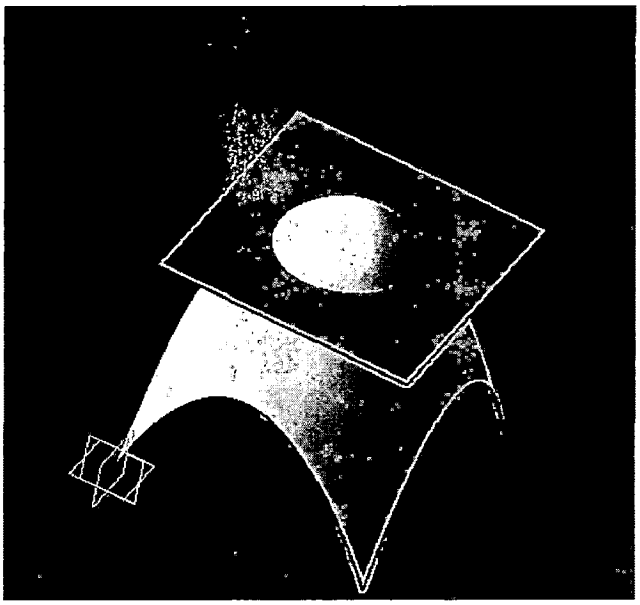
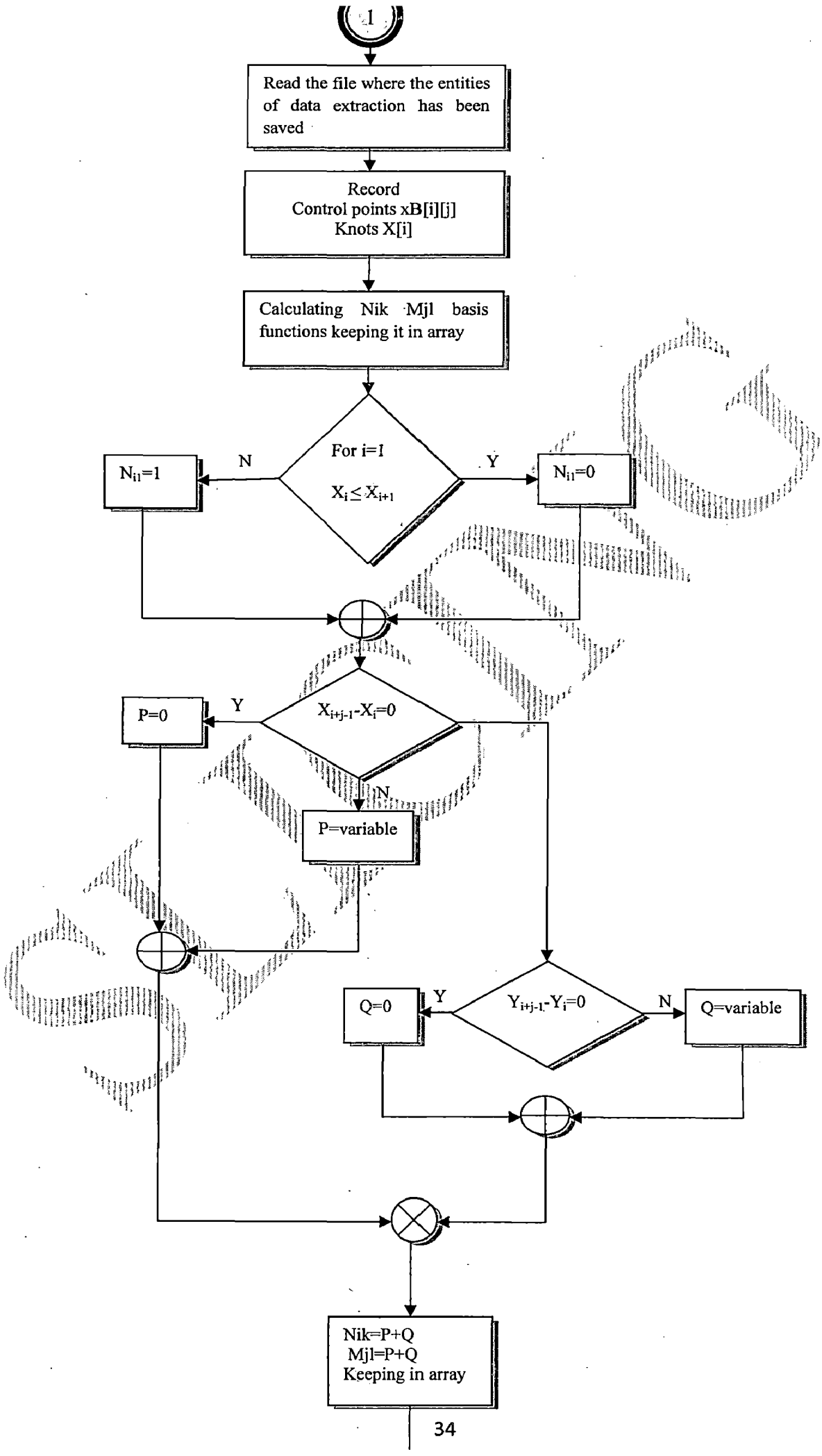
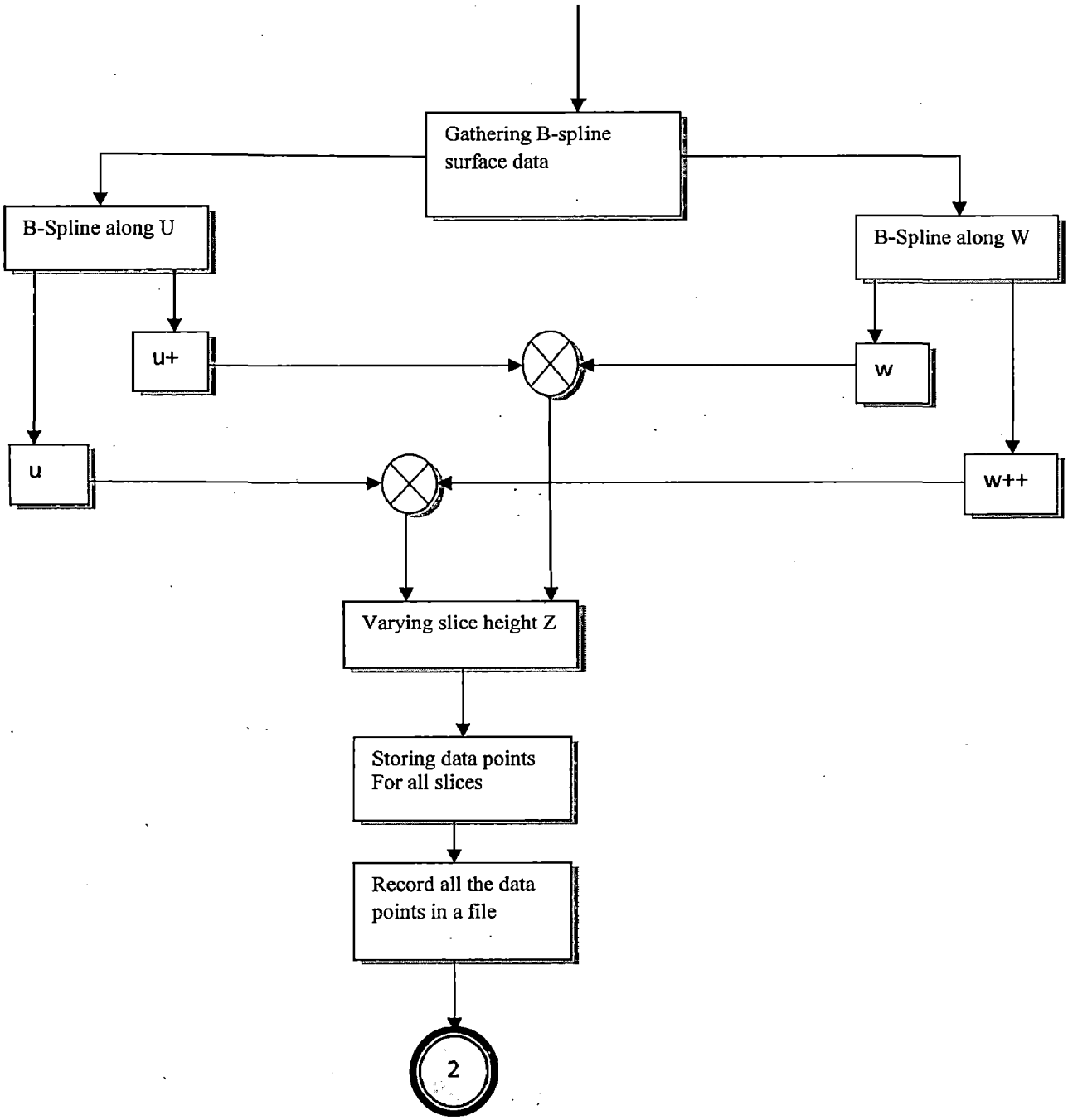


fig: 6.1 Slicing


1
Read the file where the entities of data extraction has been saved
Record
Control points xB[i][j]
Knots X[i]
Calculating Nik Mjl basis functions keeping it in array
For i=I
$X_i \leq X_{i+1}$
N
Y
$N_{i1}=1$
$N_{i1}=0$
P=0
Y
$X_{i+j-1}-X_i=0$
N
P=variable
Q=0
Y
$Y_{i+j-1}-Y_i=0$
N
Q=variable
Nik=P+Q
Mjl=P+Q
Keeping in array



Algorithm II: slicing

This program is developed for B-Spline surface slicing. It directly generates the points on the surface at particular slice at any parametric value within the range R. As B-spline basis function is a cox and de-boor recursive formula, all the first order basis functions are calculated first then the second order and so on... as is evident from fig: 5.2 explain well. That is to find $N_{1\,3}$ we need $N_{1\,2}$ & $N_{2\,2}$ thus it forms a network for k =3.

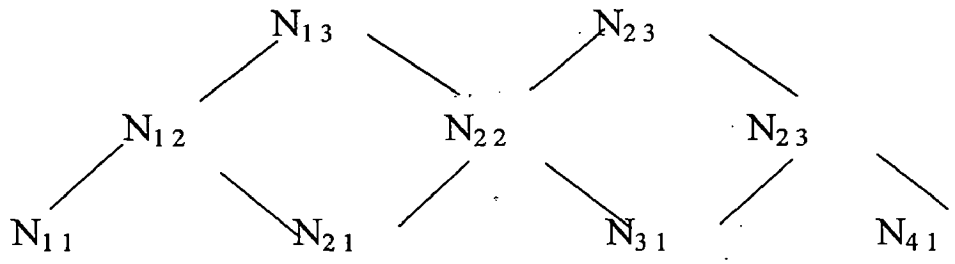


fig: 5.2 B-Spline basis function network

5.1 B-SPLINE SURFACE FITTING THROUGH GRID DATA

Generation of B-spline surface from known control polygon is discussed here. The inverse formulation is also done i.e given the known set of data points in the form of grid on a surface and there is need to determine the defining control polygon net that best interpolates that data. The points obtained as discussed previously are used to fit the surface.

To start the formulation let the parametric equation for B-spline surface in equation 5.1 is,

$$Q(u, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} N_{ik}(u) M_{jl}(w) \qquad (5.1)$$

In this Q (u, w) are the known surface data points and B_{ij} are the unknown defining control polygon net vertices. The $N_{i\,k}(u)$ an $M_{j\,l}(w)$ basis function can be determined provided that the parametric values u and w are known at the surface data points. For a single surface data point the equation can be written as

$D_{1,1}(u_1, w_1) =$

$N_{1,k}(u_1)[\, M_{1,l}(w_1)\, B_{1,1} + \ldots\ldots\ldots\ldots\ldots + M_{m+1,l}(w_1) B_{1,m+1}] +$

.

.

.

$N_{n+1,k}(u_1)[\, M_{1,l}(w_1)\, B_{n+1,1} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + M_{m+1,l}(w_1) B_{n+1,m+1}]$

The system simultaneous equation resulting from each data point is expressed in matrix form as given below

[D]=[C][B]

The required defining control polygon net can be obtained by matrix inversion method

$$[B]=[C]^{-1}[D] \quad (5.2)$$

Now the control polygon net vertices obtained using equation can be used to generate a B-spline curve. And the original B-spline equation is used to obtain the points on that particular slice plane.

Example:

The data extracted from the STEP file is provided for the B-spline equations 3.4. Algorithm II generates a file which contains the contour points of particular slice plane. The fig: 5.3 shows X, Y, Z values in order. As we observe that the Z-value remains constant for particular slice plane. The Z value here is represented as the slice height. Varying the Z value indicates the change in slice height.

The file is as shown in fig: 5.3.

File Edit Format View Help

```
-40.0,31.73998617,50.0
-40.0,13.10974336,50.0
-30.0,37.47531868,50.0
-30.0,8.546151915,50.0
-28.0,37.91356659,50.0
-28.0,8.215712042,50.0
-18.0,36.82548543,50.0
-18.0,9.040720809,50.0
-42.0,29.37289479,50.0
-42.0,15.13169724,50.0
-38.0,33.50085343,50.0
-38.0,11.65974234,50.0
-34.0,35.97726703,50.0
-34.0,9.694648964,50.0
-32.0,36.83360604,50.0
-32.0,9.034506329,50.0
-26.0,38.15023312,50.0
-26.0,8.038292505,50.0
-24.0,38.17930776,50.0
-24.0,8.016545839,50.0
2.0,37.98665449,50.0
-22.0,8.160844314,50.0
-20.0,37.54816835,50.0
-20.0,8.491051103,50.0
-16.0,35.75671167,50.0
-16.0,9.866258219,50.0
-12.0,32.04043121,50.0
-12.0,12.8591426,50.0
-10.0,28.53750316,50.0
-10.0,15.86602801,50.0
-44.0,25.41467687,50.0
-36.0,34.88196292,50.0
-36.0,10.55337168,50.0
-36.0,34.88196292,50.0
-36.0,10.55337168,50.0
-14.0,34.23439755,50.0
-14.0,34.23439755,50.0
-14.0,11.06880094,50.
```

X Y Z

fig: 5.3 profile points for one slice plane

In calculating coordinate values of every slice plane we make slicing in Z direction where all X,Y coordinates which correspond to a given Z slice are found using B-spline equations. For finding these coordinates from B-spline equations we change u, w parameters at particular value of Z. All the values of XY are recorded in an array and this is used for generating the tool path for particular slice plane. There will be huge data generated in this section. All this huge data is stored separately in an array. fig: 5.3 shows the data generated only for the slice plane at Z=50. All this data is stored for further use for generating tool path, which is explained in the next subsequent chapter.

CHAPTER 6
TOOL PATH STRATEGY

From a programmer's point of view a NC machine moves its tool by specifying coordinates of the destination points and the machining modes, e.g., rapid traversal and linear or circular interpolations, under which the tool moves to the new locations. Tool location is usually specified by the center point of the tool tip. Tool path design depends mainly on the geometry of the part to be machined and the size and shape of the selected tool. Basic requirements for tool path generation include:

- The union of all tool movements, i.e., its envelope must cover the entire area of the material to be removed.
- There is no gouging, i.e., unwanted cut, or collision between the tool and the part surface or any part of the machining setup, e.g., fixtures.
- The machined part surface must be within the required tolerances specified in the part design.

One fundamental task in tool path design is to compute the tool center offset from the part surface [20]. fig 6.1, shows two examples of tool path design for 2 ½D milling operations. On the left is a profile operation by driving the tool along a single tool radius offset from the part boundary. On the right is a pocketing operation where material inside the pocket profile is to be removed. The strategy used in this example uses consecutive offsets from the boundary with a constant step over size. The tool path starts from the center, where the tool needs to plunge or ramp down in Z direction, then spirals along the offsets one at a time towards the outer boundary. The sharp corners of the pocket cannot be machined completely without gouging and the machined part has rounded corners with radius equal to the tool radius.

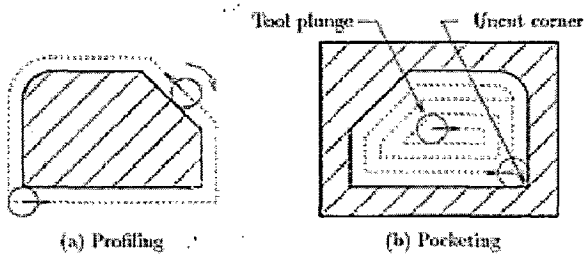


fig: 6.1 tool path uncut areas

6.1 Roughing

In our case the 3D problem is reduced to series of 2D problem as we discussed slicing in previous section (chapter 5). Now in this section we discuss the application of simple zig-zag path strategy for each slice. The profile obtained after slicing is offset through certain distance to get the rough cut profile. This rough cut profile is approximated by number of small straight lines which makes the actual profile as shown in the figure: 6.3. The cutter moves in a zig-zag path to remove most of the unwanted material from the stock. Before applying zig-zag path the selection of stock boundary is necessary. The stock of the surface is selected such that it covers the whole surface without any excess of material. The edge control points of the freeform surface and the total height of the surface makes the optimum stock (OS). The procedure is explained here.

1) The control points of four edges of the surface ABCD are recorded
2) The maximum height (H_{max}) of the surface is recorded.
3) Joining the four control points to form the closed shape (rectangle).
4) Projection of four edge control points ABCD to H_{max} covers the whole surface.

This forms the closed cubic structure covering the whole surface as shown in the fig: 6.2.

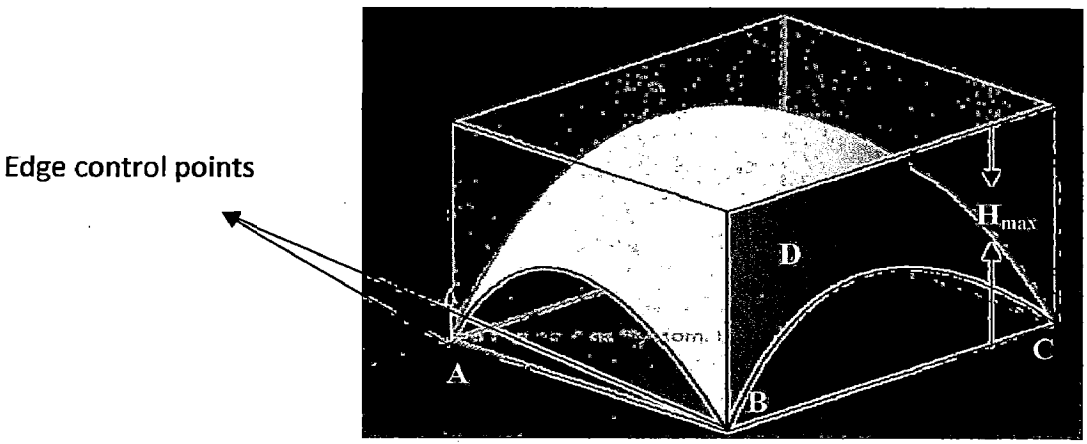


fig: 6.2 Optimum stock

For one slice the tool enters at one edge of the stock boundary and exits at another edge following zig-zag path in between as shown in the fig: 6.3 , this process removes most of the unwanted material. The CL data is calculated by the equation [21]

$$P_{CL} = P_{CC} + tool\ offset \quad (6.1)$$

Where

P_{CL} = Cutter Location point; P_{CC} = Cutter Contact point;

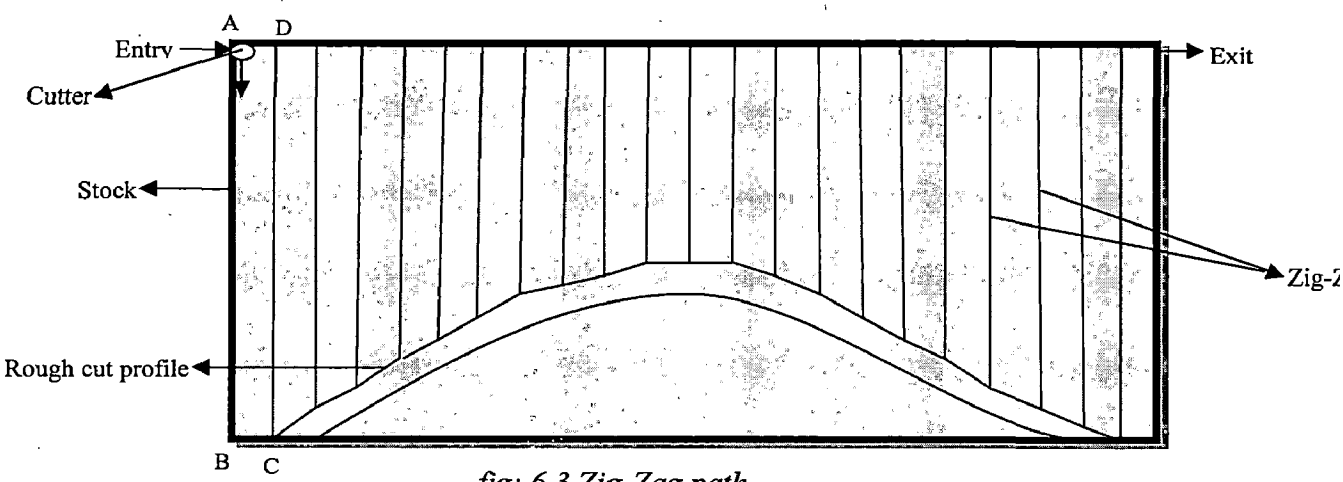


fig: 6.3 Zig-Zag path

As explained in the equation 6.1 the tool offset value depends on the tool diameter. The tool travels along ABCD as shown in the figure 6.3; like that it covers the whole area of machining. Let us calculate the tool path for one zig-zag path by enlarging it as shown in the fig 6.4:

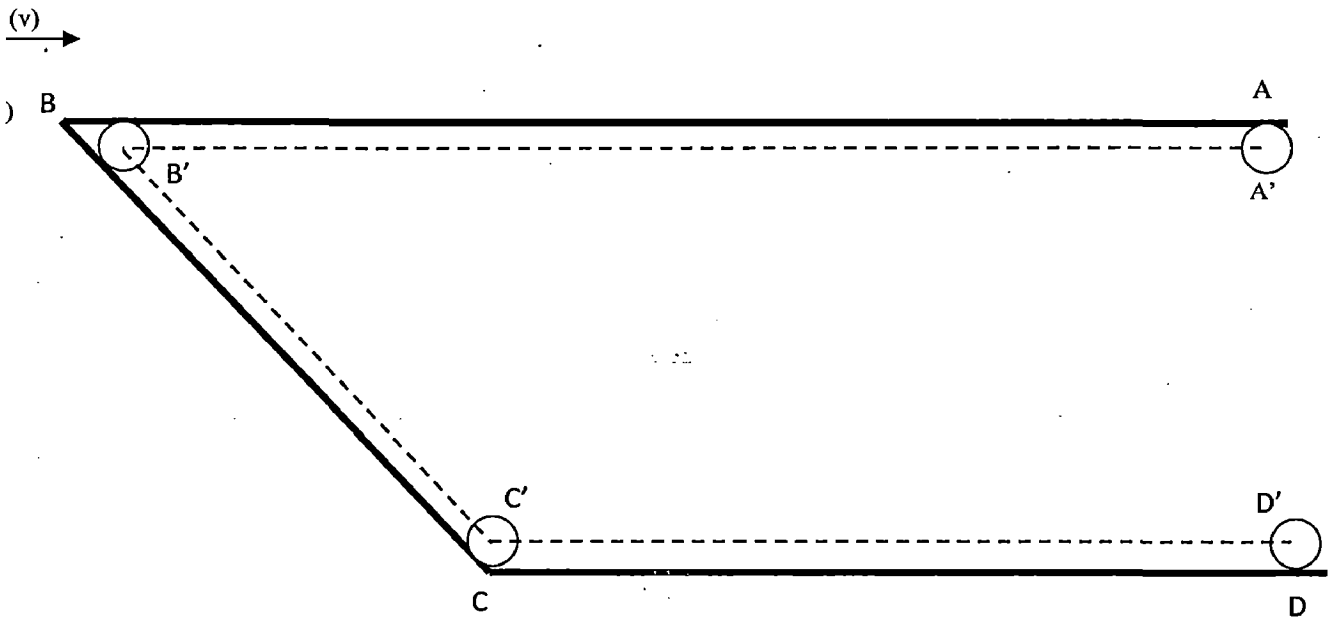


fig: 6.4 Actual tool path

Tool path		X	Y
AB	A'B'	-	- (AB- ΔY)
BC	B'C'	BF	FC
CD	C'D'	-	(CD- δ)

Table: 2 Actual tool travel

The profile path is ABCD but the tool travels along A'B'C'D', this actual tool path must be necessarily calculated. The table shows the actual tool path that the tool should travel along, this is calculated by simple trigonometry.

Where the $\Delta Y = r \tan(\theta/2)$

$$\delta = r \tan(90-\theta)$$

As the tool follows the path A'B'C'D', but there exists a small gap between B and B', this is called the uncut area. For each zig-zag path we remain with these uncut areas. If this occurs the whole slice plane will remain with uncut area formation, this may cause tool collision with those non cut areas when the tool enters in to the next slice plane. One way of reducing this uncut area is decreasing the diameter of the tool. But as we decrease the diameter of the tool the tool travel increases and also there is no guarantee that it will reduce the problem completely [29]. This problem can be completely removed by the method called first pass, in which the tool travels along the boundary and the profile that we get for roughing with diameter offset. By this method we can eliminate the uncut area formation. The fig: 6.5 shows the first pass on one of the slice plane

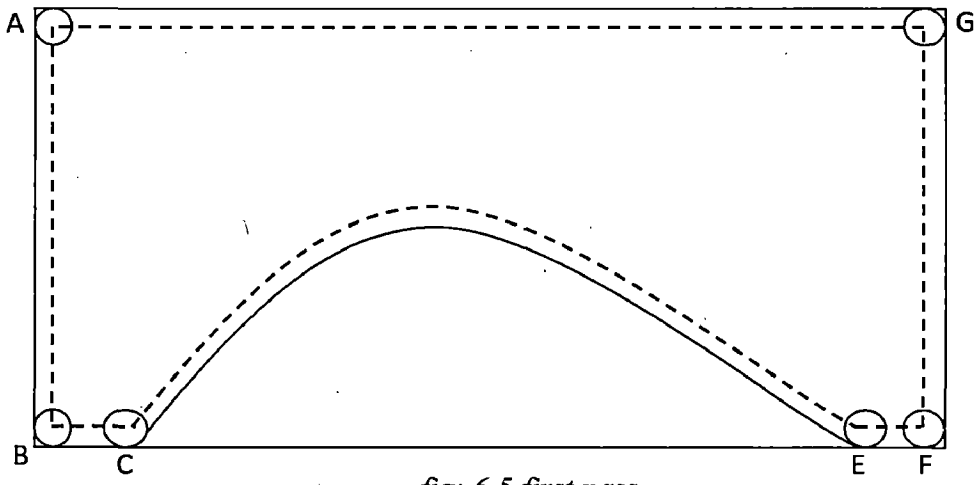
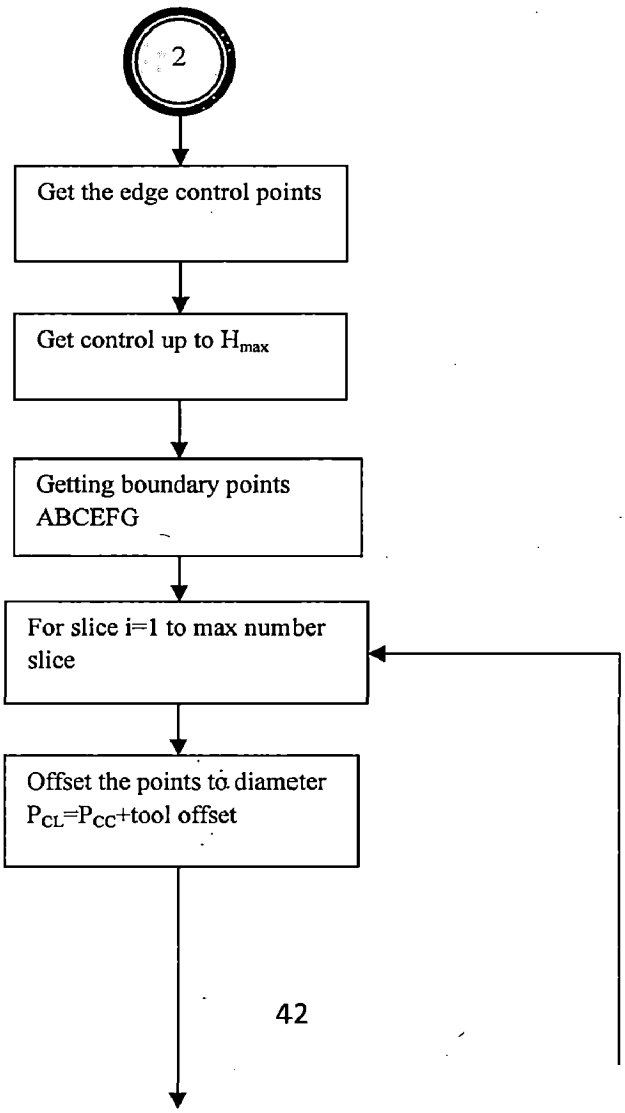
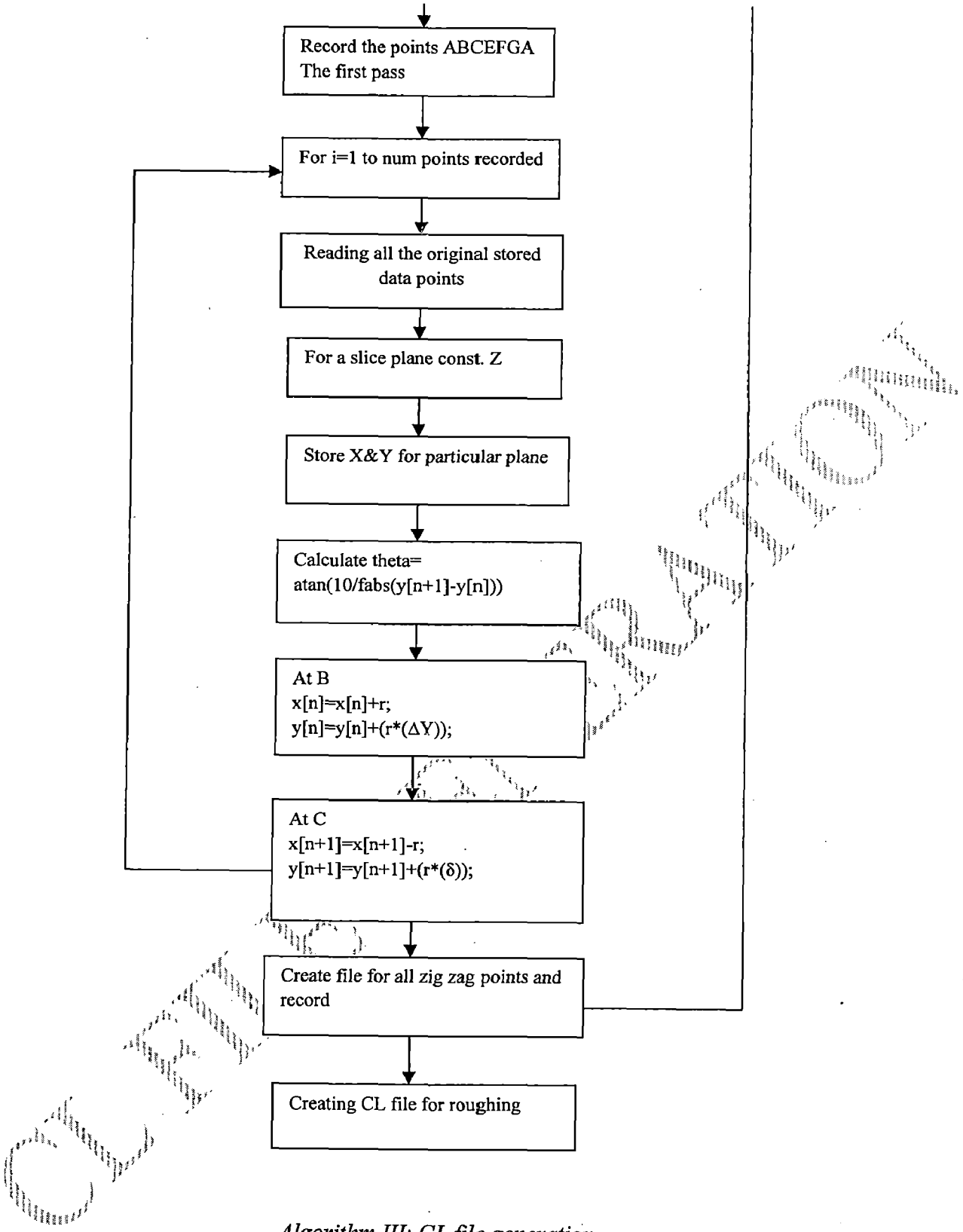


fig: 6.5 first pass

The tool travels along ABCEFGA as shown in the fig: 6.5 in each slice plane before it starts the zig-zag path operation. Then the zig-zag path is applied based on the original rough profile points. This process helps to eliminate the uncut area formed and is applied for every slice plane. The Algorithm III is used to get CL points.





Algorithm III: CL file generation

6.2 Finishing

Finishing involves a slow pass across the material in very fine steps to produce the finished part. In finishing, the step between one pass and another is minimal. Feed rates are low and spindle speeds are raised to produce an accurate surface.

The finishing operation involves

- ✓ The slow and minimal step between one pass and the other
- ✓ The tolerance that should achieve

The tolerance of the product depends on the application of that product. Whatever so, our goal is to reach the given tolerance range by the user

6.2.1 Tolerance

In the design phase of the manufacturing process, products are specified with nominal dimensions and tolerances. The tolerance means the allowable variability for certain geometric dimensions or forms. If the product is manufactured and measured within the tolerance range, then it is deemed a good product. Otherwise, it is a bad product. Therefore, it is very important to understand the relationship between the design and manufacturing in terms of tolerance specification. As a result, the tolerance specifications have to be revised from time to time in the manufacturing phase following the design phase. Therefore, it is very meaningful to research and develop scientific methodologies for determining the tolerance specifications during the design phase itself.

The tolerance in the tool path can be defined by the following criteria

1) For getting the shape within the tolerance range the slice height should be kept in mind. The larger slice height will not achieve the tolerance limit given.
2) And the tool diameter is also selected properly
   Higher diameter may not reach the required tolerance limit

3) Lastly the distance to travel should be such that it should be within the tolerance range. So that the surface finish is good.

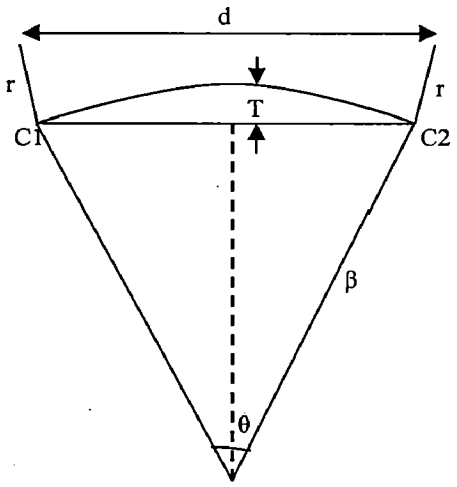


fig: 6.6 Tolerance

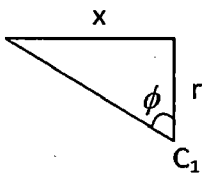
The minimum distance travel (d) should be calculated so that we can be within the range of given tolerance. Let us take a small portion of the curve represented by CC. which makes an angle θ at β distance. The tool of radius r travels from C_1 to C_2. The minimum distance d that would better approximate this curve is calculated.

$$C_1C_2=2\beta \sin\theta/2 \tag{6.2}$$

$$T=\beta-\beta \cos\theta/2 \tag{6.3}$$

$$\cos\frac{\theta}{2}=\left(\frac{T-\beta}{\beta}\right)$$

$$\left(\cos\frac{\theta}{2}\right)^2=\left(\frac{T-\beta}{\beta}\right) \tag{6.4}$$



$x = r \sin\phi$

$C_1C_2 = d + r \sin\phi$ (6.5)

Equating equation: 6.2 & 6.5

$2\beta \sin\theta/2 = d + r \sin\phi$

$2\beta (1-\cos\theta/2)^2 = d+r$ (6.6)

as ϕ is very small.

Substituting equation: 6.4 in equation 6.6 gives

$$d = 2\sqrt{2Tr - T^2} \quad (6.2)$$

Where d= distance to travel

T=tolerance range

r= cutter radius

The d in this equation gives us the minimum distance to travel maintaining the given tolerance ranges.

As we know the B-spline equation

$$Q(u, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} N_{ik}(u) M_{jl}(w)$$

For getting the finish cut data we keep one of the parametric value (u) as constant and vary other parametric value (w) to its maximum. This gets all the points at one particular position of u. Increasing u value step by step and at each particular value of u incrementing the value of w to its maximum we get all the points at that particular place. Covering the whole surface by this process we get the co-ordinate points required for the tool path and the CL file is generated to keep track of all the points for finishing the given surface.

Example:

The fig: 6.7 shows the CL file which is generated by the Algorithm III.

File Edit Format View Help

```
GOTO  /   -13.77646,   38.49815,   50.18397
GOTO  /   -13.79478,   38.50992,   50.18397
GOTO  /   -15.31918,   39.40554,   50.18397
GOTO  /   -16.65851,   40.05264,   50.18397
GOTO  /   -16.86191,   40.15040,   50.18397
GOTO  /   -18.40463,   40.70813,   50.18397
GOTO  /   -19.94736,   41.14667,   50.18397
GOTO  /   -21.49008,   41.42021,   50.18397
GOTO  /   -23.03281,   41.59537,   50.18397
GOTO  /   -24.57553,   41.68375,   50.18397
GOTO  /   -26.11826,   41.61402,   50.18397
GOTO  /   -26.45087,   41.59537,   50.18397
GOTO  /   -27.66098,   41.48400,   50.18397
GOTO  /   -29.20371,   41.22968,   50.18397
GOTO  /   -30.74643,   40.87903,   50.18397
GOTO  /   -32.28915,   40.44989,   50.18397
GOTO  /   -33.29445,   40.05264,   50.18397
GOTO  /   -33.83188,   39.83258,   50.18397
GOTO  /   -35.37460,   39.16333,   50.18397
GOTO  /   -36.63718,   38.50992,   50.18397
GOTO  /   -36.91733,   38.36440,   50.18397
GOTO  /   -38.46006,   37.38450,   50.18397
GOTO  /   -39.05396,   36.96719,   50.18397
GOTO  /   -40.00278,   36.29767,   50.18397
GOTO  /   -41.03642,   35.42447,   50.18397
GOTO  /   -41.54551,   34.94302,   50.18397
GOTO  /   -42.66171,   33.88174,   50.18397
GOTO  /   -43.08823,   33.43347,   50.18397
GOTO  /   -43.95161,   32.33902,   50.18397
GOTO  /   -44.63095,   31.46085,   50.18397
GOTO  /   -45.05101,   30.79630,   50.18397
GOTO  /   -45.99012,   29.25357,   50.18397
GOTO  /   -46.17368,   28.88943,   50.18397
GOTO  /   -46.64619,   27.71084,   50.18397
GOTO  /   -47.17665,   26.16812,   50.18397
GOTO  /   -47.55169,   24.62539,   50.18397
GOTO  /   -47.71640,   23.12305,   50.18397
GOTO  /   -47.71972,   23.08267,   50.18397
GOTO  /   -47.76842,   21.53995,   50.18397
GOTO  /   -47.63916,   19.99722,   50.18397
GOTO  /   -47.30395,   18.45449,   50.18397
GOTO  /   -46.83639,   16.91177,   50.18397
```

fig: 6.7 CL file

The CL file which is generated is used for co-ordinate movement of tool to move to the next location. Combining all tool movements will get the tool path for the whole surface.

CHAPTER 7

IMPLEMENTATION OF ALGORITHMS

Our algorithm here generates a CL file which contains the co-ordinate movements of the tool. The proposed solution approach was developed and implemented. There are some examples for which the CL points were generated using the algorithm I, II, III and were ready to subsequently machine on 3-axis milling machine. The software used to implement proposed algorithms is:

- BLOOD SHED DEV C++
- CATIA V5 R12
- AUTOCAD 2005

The algorithms I, II, III were coded in BLOOD SHED DEV C++ on a personal computer (Pentium III, 1.0 GHhz CPU, 256 Mb of physical memory) operating under Microsoft XP.
The desired surface was generated by CATIA V5 R12 using X, Y, and Z coordinates generated by BLOOD SHED DEV C++.
The STEP file of the particular model is as shown in the fig: 7.1.

fig: 7.1 STEP file format

CASE: I

Let us take a freeform surface as shown in fig: 7.2

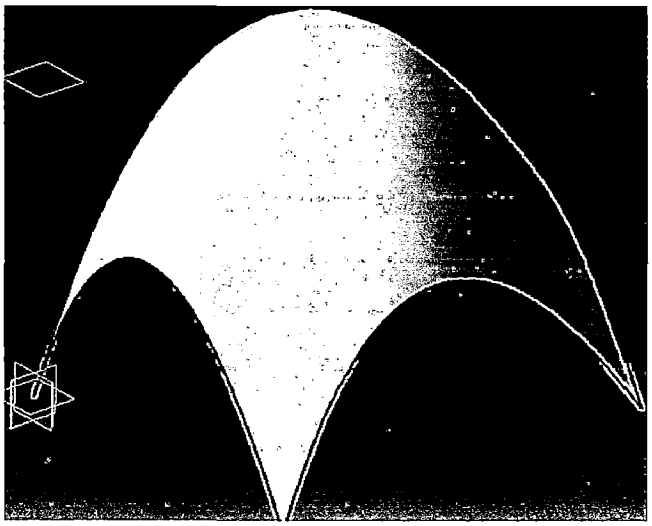


fig: 7.2 freeform surface

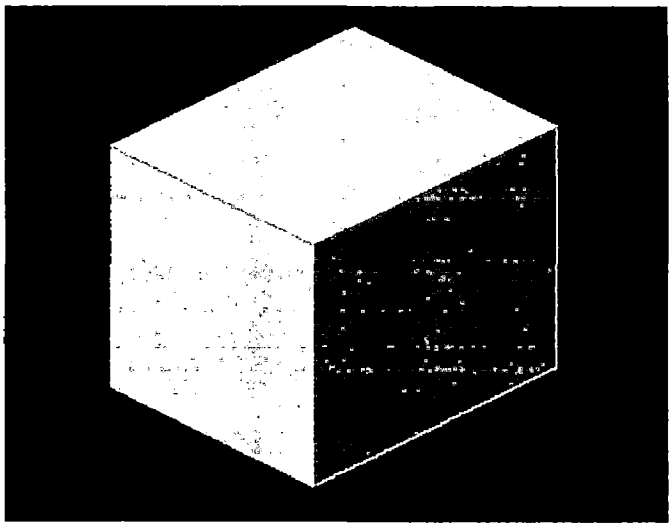


fig: 7.3 Stock

As we explained earlier the desired surface is generated from the cubic optimum stock as shown in the fig: 7.3. From this stock the rough machining starts in a zig-zag way slice-by-slice. The zig-zag path and the slicing techniques have been explained in the previous chapter 5. The fig: 7.4 here will show us the procedure clearly.

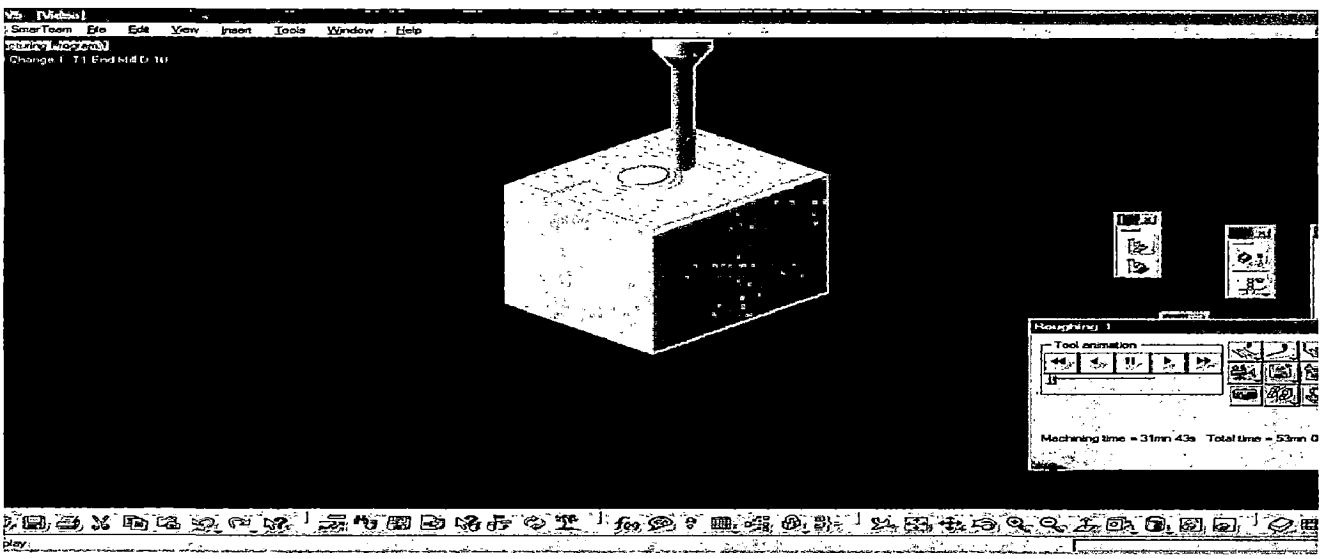


fig: 7.4 zig-zag pattern

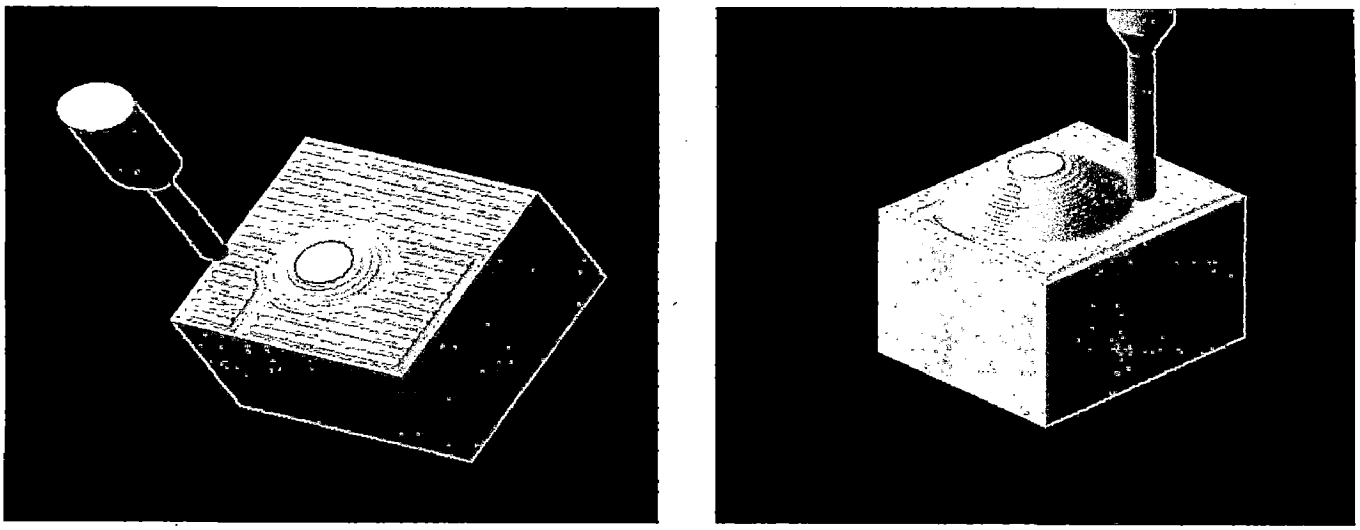


fig: 7.5 first pass

fig: 7.5 shows the first pass which passes through the profile once before going to actual process. In the first pass the tool moves along the boundary of the stock and also the profile that we get on particular plane.

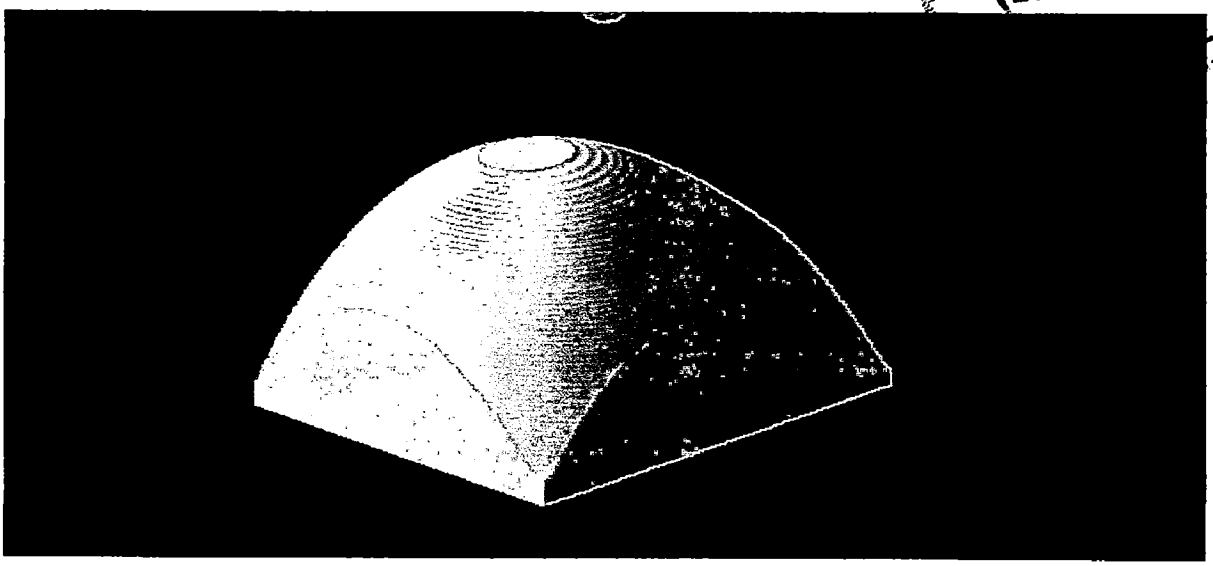


fig: 7.6 surface after roughing

fig: 7.6 shows the final Surface after Rough cut

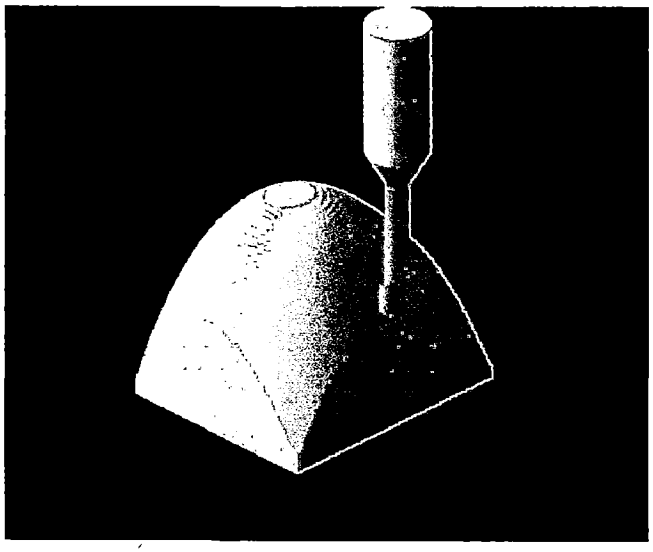


fig: 7.7 finishing cut

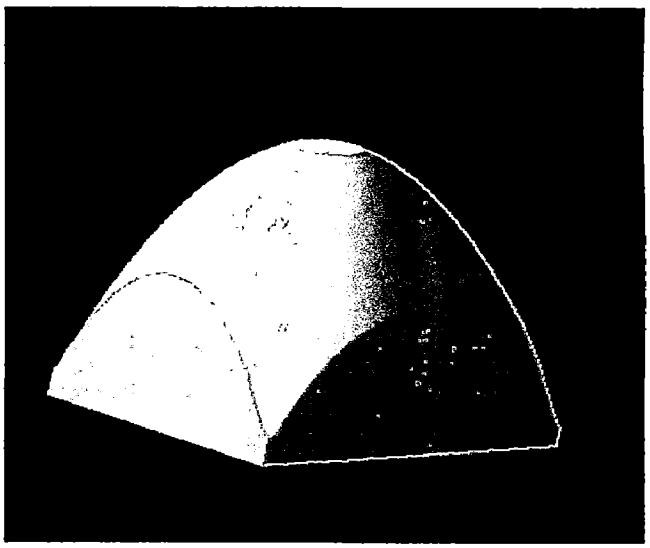


fig: 7.8 final surface

This is the final generated free form surface after finishing cut.

CASE: II

Let us take the example of car bonnet and solve for generating tool path (CL file).

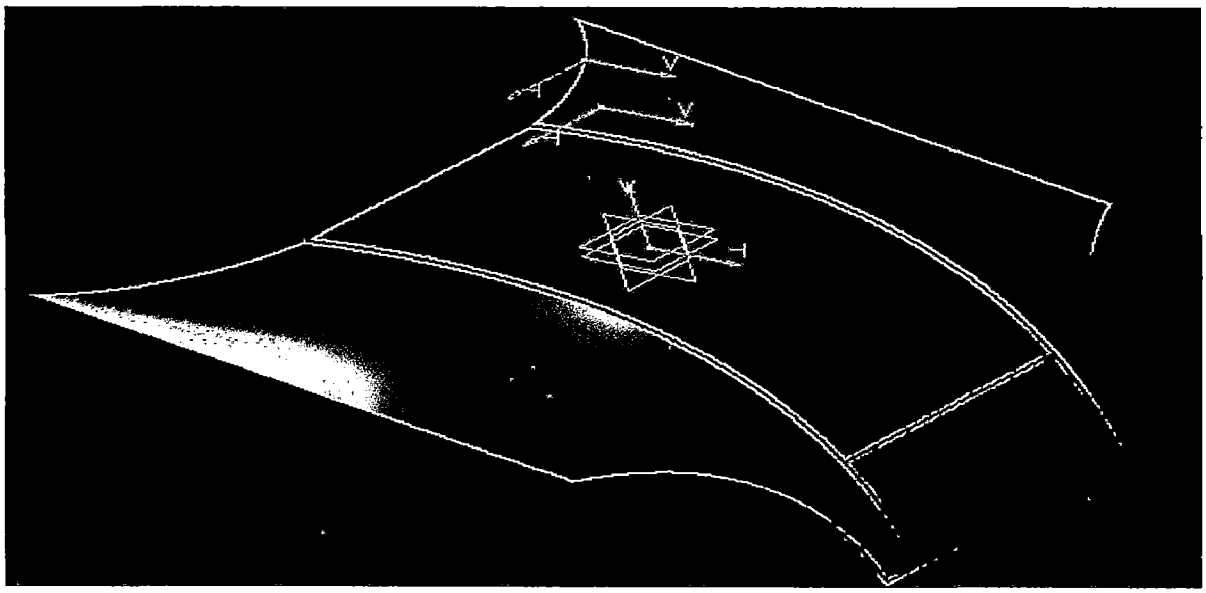


fig: 7.9 Maruthi Swift Bonnet

This is the shape of the MARUTHI SWIFT car bonnet. This involves two algorithms I and II. The first one is for the back of the bonnet to remove the material and the second one is for removal of unwanted stock material to get the desired surface. The figure here under shows the tool path.

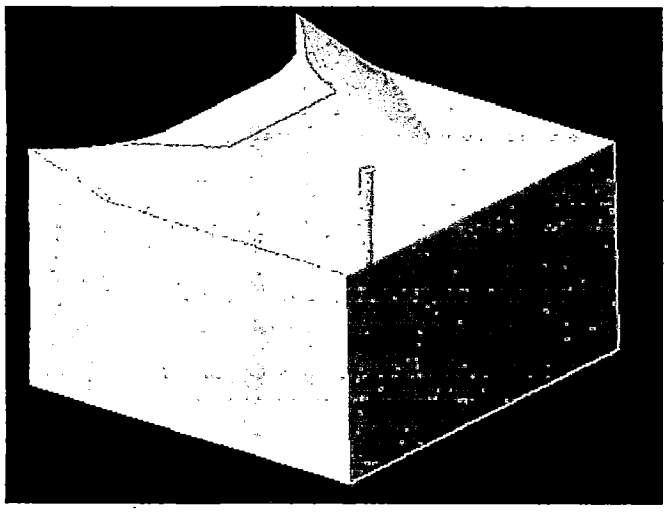


fig:7.10 Roughing of the car bonnet following zig- zag path

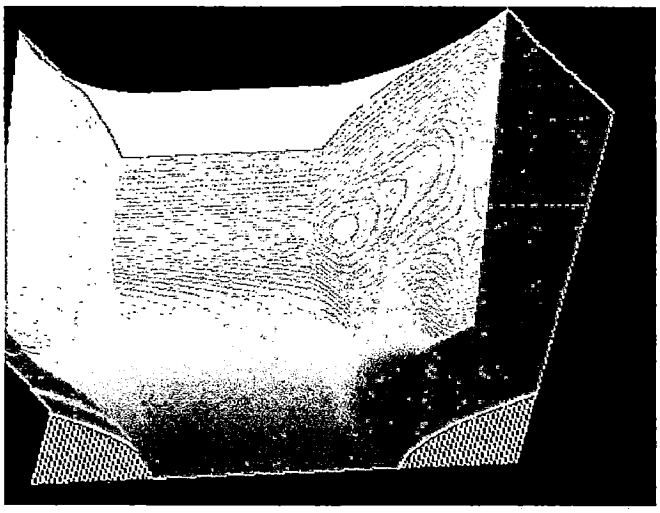


fig: 7.11Ffinishing operation

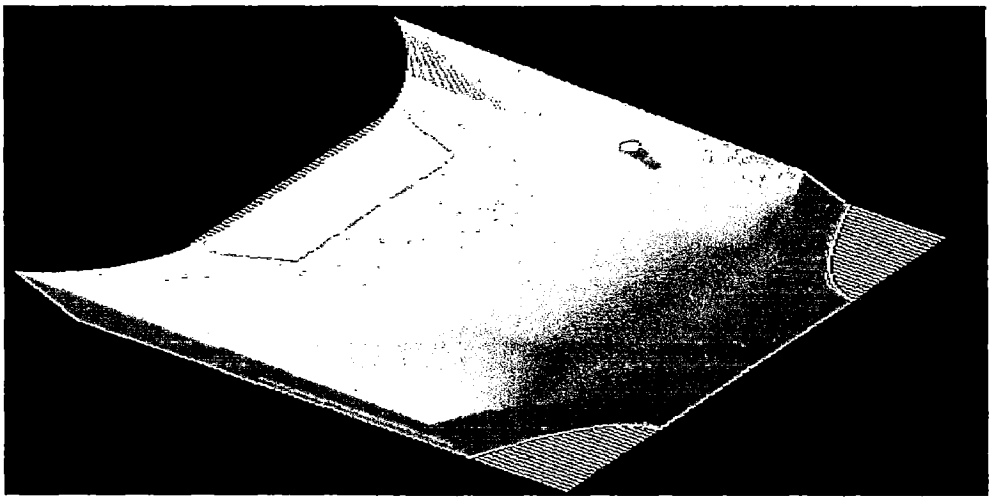


fig: 7.12Ffinal surface generated after finishing

CHAPTER 8

RESEARCH SUMMARY AND CONCLUSIONS

A. RESEARCH CONTRIBUTION AND CONCLUSION

The proposed algorithm for tool path generation was developed and implemented successfully through the integration of mathematical modeling used for calculating CL file.

In the present work an approach called slicing technique is used to get the 3D geometry (machined on 5-axis) of sculptured part to 2D geometric (machined on 2 ½ or 3-axis) level and then a simple zig-zag path strategy is applied for each slice according to their level of details

The 3D problem is reduced to series of 2D problem by slicing and in each slice plane the material between the stock boundary and the freeform surface is removed by calculating CL data for each slice plane. Basically the system includes four components: - a data extraction module, a volume slicing module, a tool path module, a Cuter Location (CL) file generation module. The flow chart for CL file generation is shown in fig: 8.1.

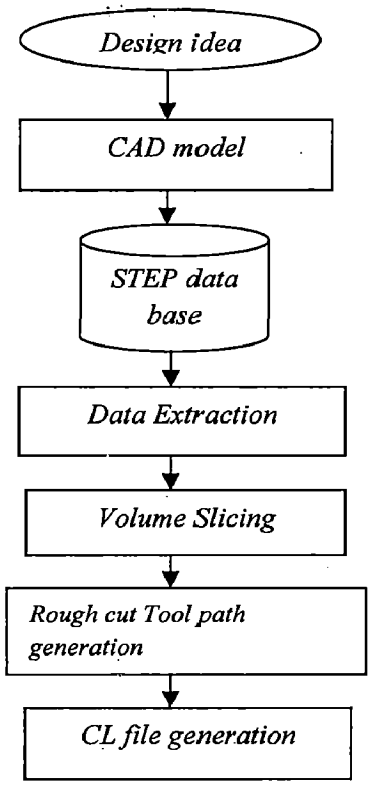


fig: 8.1 flow chart for CL file

1. *Data extraction*

   A CAD design model is usually documented in a particular data format. The geometric and topological messages may not be easily "visualized" or understood by the tool path generator (a system or a human programmer). Therefore the information needed for tool path generation has to be extracted from CAD files and converted in to usable format for generating tool paths.

2. *Volume slicing*

   This module slices the freeform surface horizontally layer by layer to improve the machining efficiency. The reason is that planar path is more advantageous and moreover it reduces a 3D problem to series of 2D problems.

3. *Tool path*

   The tool path module will generate the zig-zag tool path for each layer based on its level of details. This module is applied for both roughing and finishing operations

4. *CL file generation*

   A CL file is generated in this module which contains the co-ordinate movements of tool that is the TOOL PATH to get the final desired shape.

Finally the CL data for all slice planes are combined together to get the CL data for the given surface, which is afterwards used for NC program preparation to produce the sculptured part. As the Tool path is automatically generated based on STEP file and no intermediate data exchange is needed shorter manufacturing cycle can be expected. This system can be easily implemented in PCs as all the codes are written in C++. This reduces the rough machining time and hence increases the productivity.

B. FUTURE DIRECTION

While this dissertation gives us an increased understanding of NC tool path generation, it also gives us several ideas for future research. Some of these directions described below are extensions of this dissertation.

Integration: Tool path generation is part of the process planning in manufacturing systems as described in chapter I. Therefore, the module of tool path generation has to be integrated into

other processes of manufacturing. The integration of data generated from the different module in process planning would be a direct extension of this dissertation.

Bonded surfaces: The application of the system is also extended to surfaces which are bonded together.

9.REFERENCES

[1] Young Keun Choi "T*ool Path Generation and 3D Tolerance Analysis for Free-Form Surfaces"* Texas A&M University may (2004)

[2] P.G.Maropoulos, R.P.Baker "*Integration of tool selection with design, part-1*" Journal of Material Processing Technology, 107(2000), 127-134.

[3] L.Alting, H.C.Zhang, "*computer aided process planning"* International Journal of Production and Reasearch, (1989), 2139-2160.

[4] C.Ou-Yang, T.S.Lin, *"Developing an integrated frame work for feature-based early manufacturing cost estimation"* International Journal of Advanced Manufacturing Technology.13 (1997), 618-629.

[5] H.P.Wang, H.Chang R.A.Wyskand A.Chandawarkar, "O*n the efficiency of NC tool path planning for face milling operation*" Journal of Engineering for Industry ASME109 (1987), 370-376.

[6] Bor-Tyng Sheen, Chun-Fong, "*Machining Feature Recognition and tool path Generation for 3-axies CNC milling*" Journal of Computer-Aided Design .38(2005), 1-10.

[7] Zezhong C.Chen,Zuomin Dong ,Geoffry W.Vickers "*Automated surface subdivision and tool path generation for 3 ½ ½ axis CNC machine for sculptural parts*" Journal of Computer in Industry 50(2003), 319-331.

[8] Y.N.Hu,W.C.Y.H.Chen and Z.D.Zhou "*Tool path planning for rough machining of cavity by layer shape analysis*" Journal of Advanced Manufacturing Technology.14(1998), 321-329

[9] Peng WU, Heromasa SUZUKI and Kiwamu Kase "*Three axes NC cutter path generation for the subdivision surfaces with the Z-map*" JSME International Journal, 48(2005), 757-762.

[10] Chun-Fong You,Chih-Hsing Chu "*An automatic path generation method of NC rough cut machining from solid models*" Journal of Computers in Industry, 26 (1995), 161-173.

[11] D.C.H.Yang, Z.Han "*Interference detection and optimal tool selection in 3-axis machining of free-form surface of free-form surfaces*" Journal of Computer-Aided Design 31(1999), 303-315.

[12] Dave Carswell, Nick Lavery "*3D solid fin model construction from 2D shapes using non-uniform rational B-spline surfaces*" Journal of Advances in Engineering Software vol.37 pp 491–501. (2006)

[13] W. BiJhm, Braunschweig "*Cubic B-Spline Curves and Surfaces in Computer Aided Geometric Design*" computing 19, pp29-34(1977) Springer Verlag

[14] David F.Rogers J.Alan Adams "Mathamatical Elements for Computer Graphics"Tata McGraw-Hill publishing ltd. Second edition 2002.

[15] Anupam Saxena Birendra Sahay "Computer Aided Engineering Design" Anamaya publishers 2005.

[16] Ibrahim Zeid "Mastering in CAD/CAM" Tata McGraw-Hill publishing ltd. Edition- 2007.

[17] ISO, *product data representation and exchange, part 42: integrated resource: geometrical and topological representation,* SO CD10303-42, International Organization for Standardization, ISO TC 184/SC4 N141, 1992.

[18] ISO, *Application protocol: Mechanical Design Using Boundary representation, product data representation and Exchange*, ISO TC 184/SC4 WG3 N277, 1994.

[19] M.Liang, S.Ahamed, B.Van den Berg "*A STEP based tool path generation system for rough machining of planar surfaces"* Journal of Computer in Industry 32(1996) 219-231.

[20] Chih-Cheng Ho "*Feature-based process planning and automatic numerical control part programming"* university of utha. Department of Computer Science.

[21] N.K.Mehta "Machine Tool Design and Numerical Control" 1996,1984 Tata McGraw-Hill publishing. second edition.

[22] P.G.Maropoulos, R.P.Baker, K.Y.G.Paramor "*Integration of tool selection with design part-2*" Journal of Material Processing Technology 107(2000), 135-142.

[23] K.L.Chui, W.K. Chui, K.M.Yu "*Direct 5-axis tool-path generation from point cloud input using 3Dbiarc fitting"* Journal of Robotics and Computer Integrated Manufacturing 24(2008) 270-286.

[24] W. BiJhm, Braunschweig "*Cubic B-Spline Curves and Surfaces in Computer Aided Geometric Design*" Computing 19, pp29-34(1977) springer verlag .

[25] MITSUBHISHI,"Mitsubishi Materials: Turning tools, Rotating tools, Tooling solutions" general catalogue 2007-2008.

[26] Mitsubhishi Materials Kobe Tools, www.mitsubishicarbide.com

[27] Sheng H, Chuang and C.C.Pan *"Rough Cut Tool Path Planning for B-spline Surface Using Convex Hull Boxes"* International Journal of Advanced Manufacturing Technology(1998)14:85-92.

[28] W.Boehm *"Generating the Bezier points of B-spline curves and surfaces"* Computer Aided Design (1981) 13(6), pp.365-366.

[28] J.Jeong and K.Kim *"Tool Path Generation for Machining FreeForm Pockets Using Voronoi Diagrams"* International Journal of Advanced Manufacturing Technology(1998)14: 876-881.

[29] S.C.Park, B.K.Choi *"Uncut free pocketing tool-paths generation using pairwise offset algorithm"* Journal of Computer Aided Design 33(2001)739-746.

[30] Chao-Ton SU, MU-Chen Chen *"Computer-aided optimization of multi-pass turning operation for continuous forms on CNC lathes"* IIE Transaction (1999) 31,583-596.

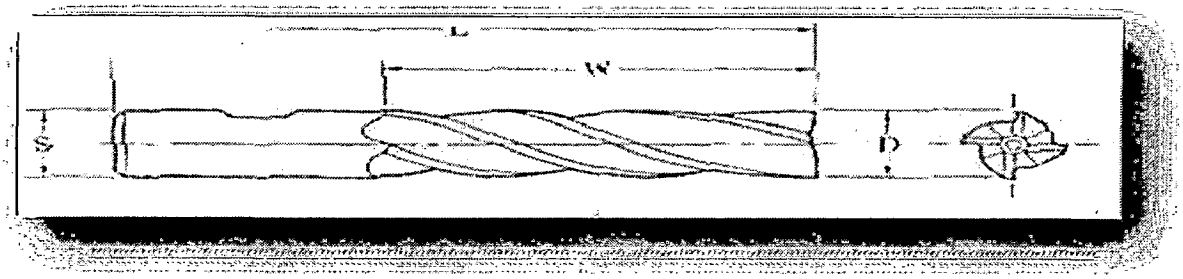
[31] Abhijit Deshmukh and Hsu-Pin (Ben) Wang *"Tool path planning for NC Milling of Convex Polygonal Faces: Minimisation of Non-Cutting Areas"* International Journal of Advanced Manufacturing Technology(1993) 8:17-24.

[32] Peter Scallan "*Process planning, the design/manufacturing interface*" Butter worth Heine Mann (2003).

[33] M. P. Groover, *Automation, Production Systems, and computer Integrated Manufacturing*, Upper Saddle River, NJ: Prentice-Hall, 1987.

TOOL SELECTION TABLE

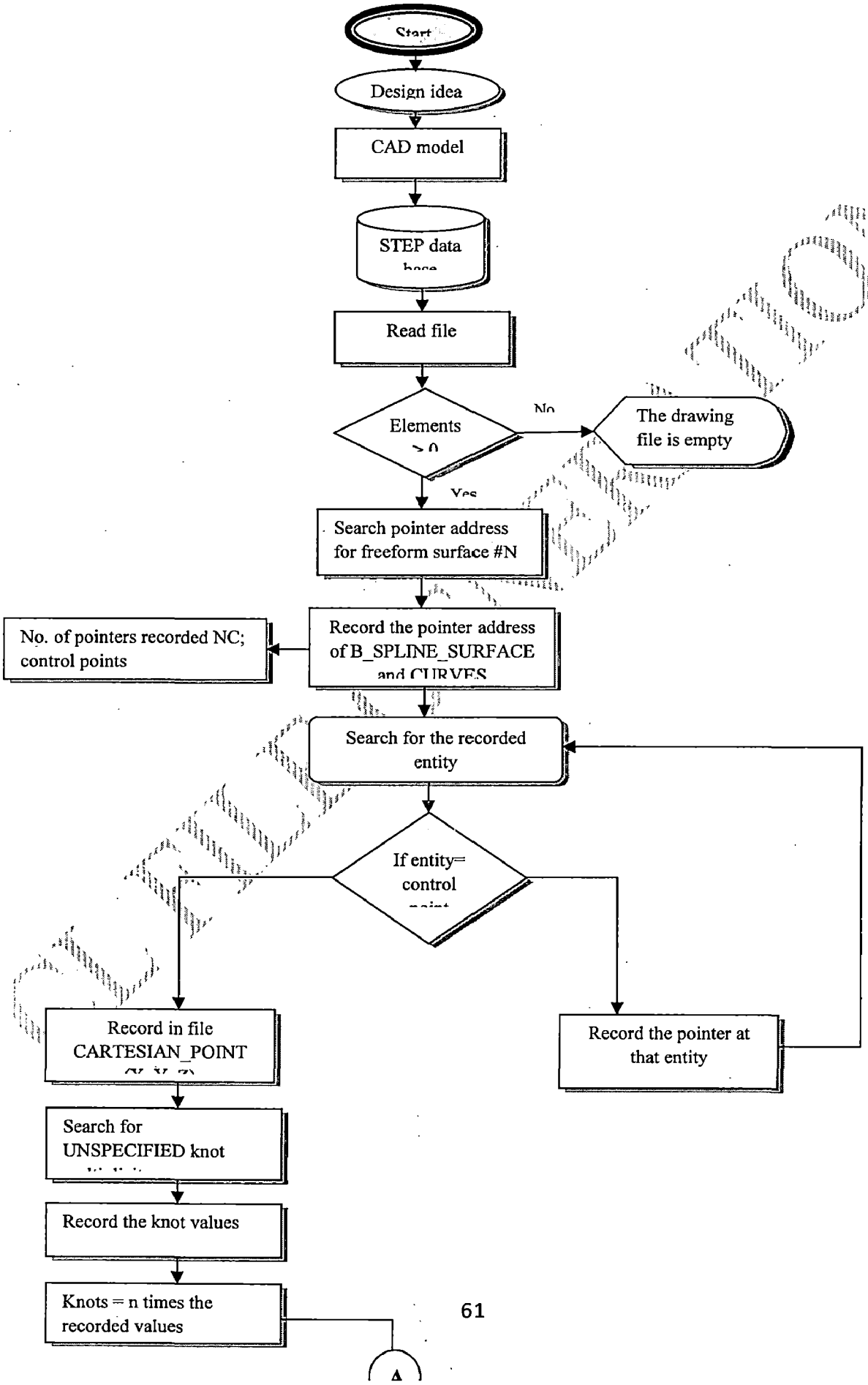
Multi flute medium helix single-end, end mill with Weldon shanks cutter

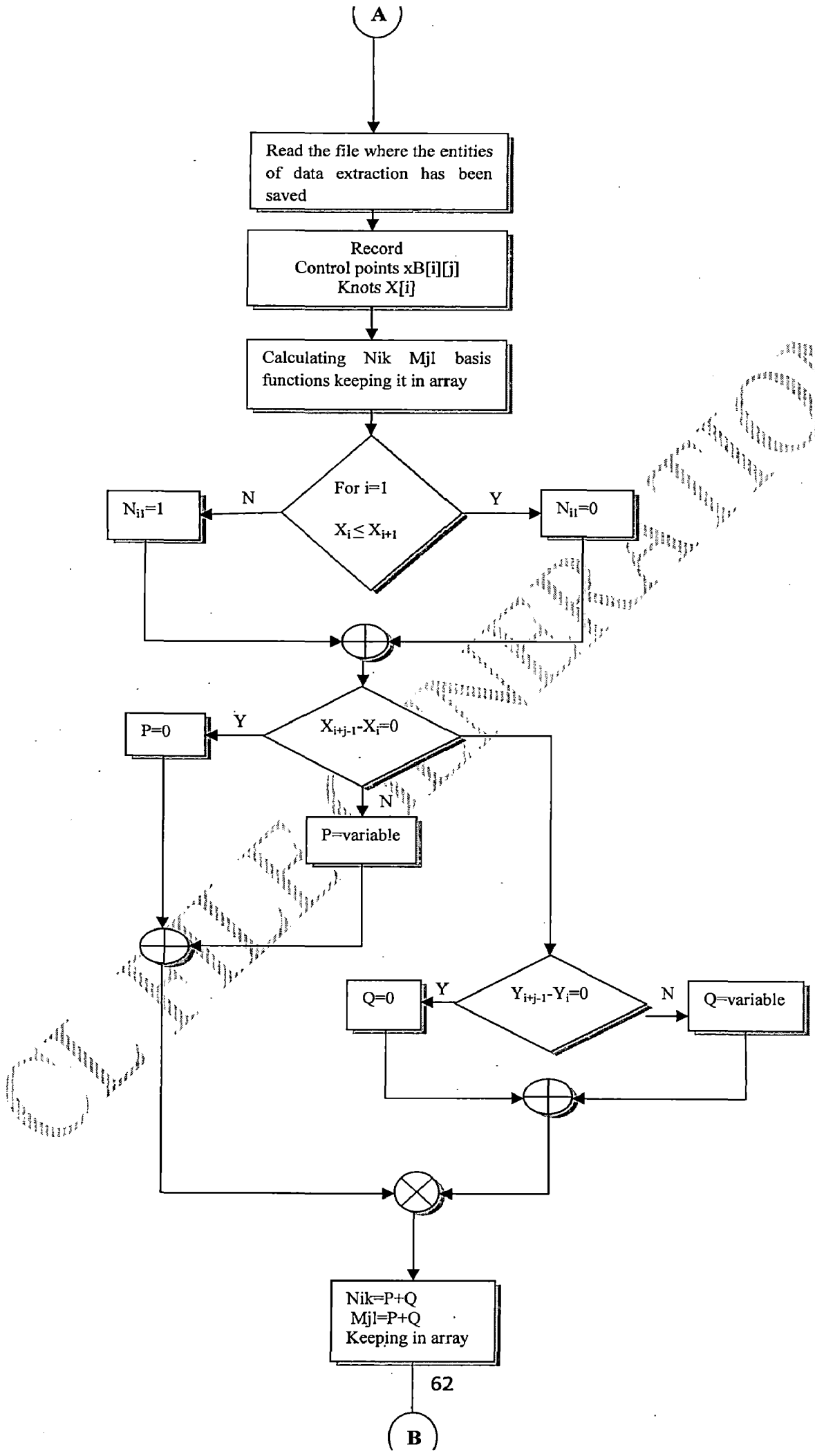


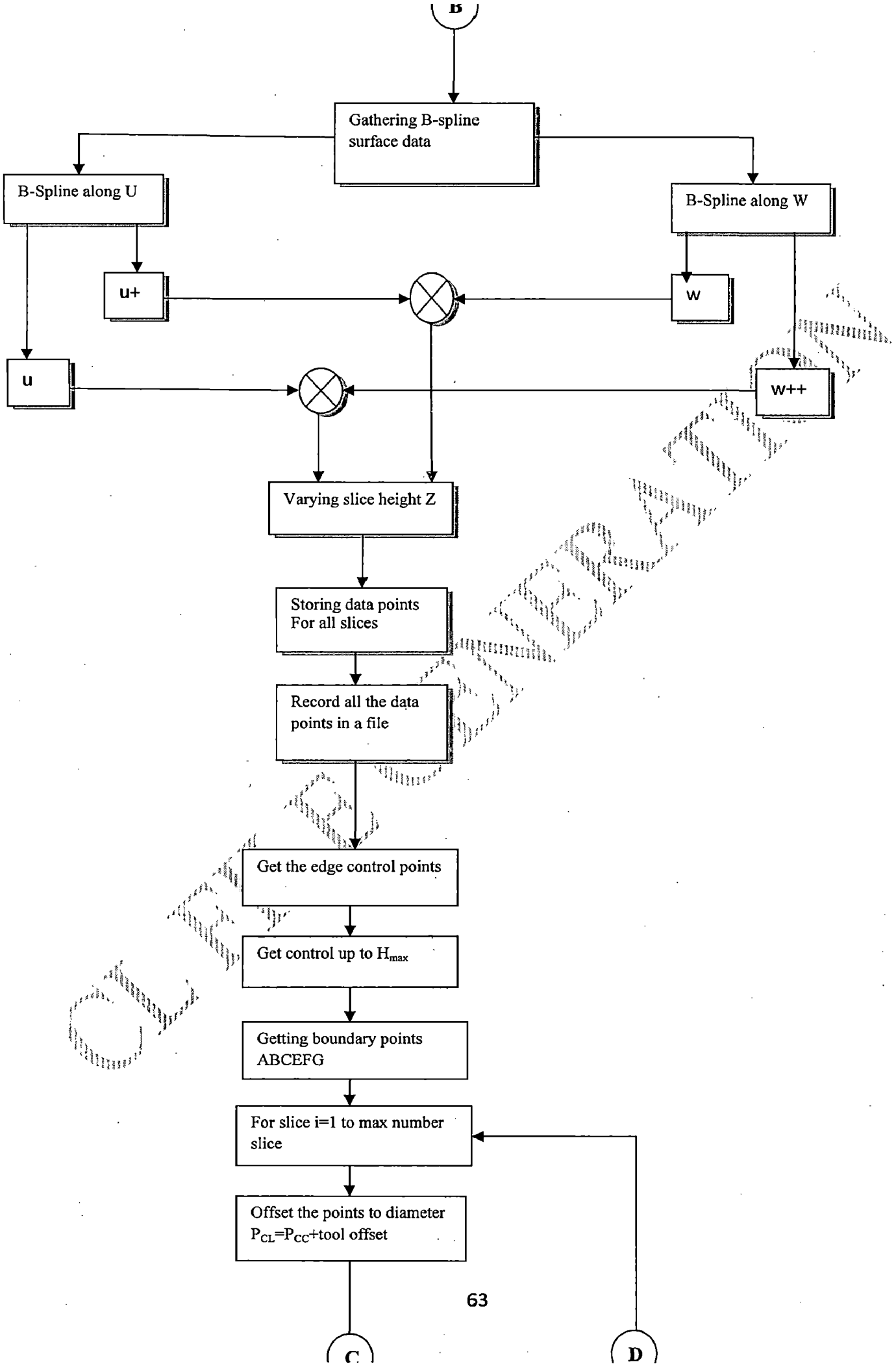
All dimensions are in inches. Cutter is high-speed steel. Helix angle is greater than 19degrees but not more than 39degrees. Tolerance ranges of -0.0001 to -0.0005

Code	Range of size	Order number	Radius of ball nose	dia	Length of cut	Overall length	Shank dia	No. of flute
VC-255B	R0.5-6	VC255BR0050	0.5	1	1	40	4	2
		VC255BR0075S06	0.75	1.5	1.5	40	6	2
		VC255BR0600	6	12	12	75	12	2
VC-2MB	R0.2-12.5	VC2MBR0020	0.2	0.4	0.8	38	3	2
		VC2MBR0080	0.8	1.6	4	40	4	2
		VC2MBR0100	1	2	6	60	6	2
		VC2MBR0300	3	6	12	80	6	2
		VC2MBR1250	12.5	25	55	180	25	2

Table: A Tool parameters


Start
Design idea
CAD model
STEP data
Read file
Elements
No
The drawing file is empty
Yes
Search pointer address for freeform surface #N
Record the pointer address of B_SPLINE_SURFACE and CURVES
No. of pointers recorded NC; control points
Search for the recorded entity
If entity= control
Record in file CARTESIAN_POINT
Search for UNSPECIFIED knot
Record the knot values
Knots = n times the recorded values
Record the pointer at that entity


A
Read the file where the entities of data extraction has been saved
Record
Control points xB[i][j]
Knots X[i]
Calculating Nik Mjl basis functions keeping it in array
For i=1
$X_i \leq X_{i+1}$
N
$N_{il}=1$
Y
$N_{il}=0$
$X_{i+j-1}-X_i=0$
Y
P=0
N
P=variable
$Y_{i+j-1}-Y_i=0$
Y
Q=0
N
Q=variable
Nik=P+Q
Mjl=P+Q
Keeping in array
B


B
Gathering B-spline surface data
B-Spline along U
B-Spline along W
u+
w
u
w++
Varying slice height Z
Storing data points For all slices
Record all the data points in a file
Get the edge control points
Get control up to H_{max}
Getting boundary points ABCEFG
For slice i=1 to max number slice
Offset the points to diameter $P_{CL}=P_{CC}$+tool offset
C
D

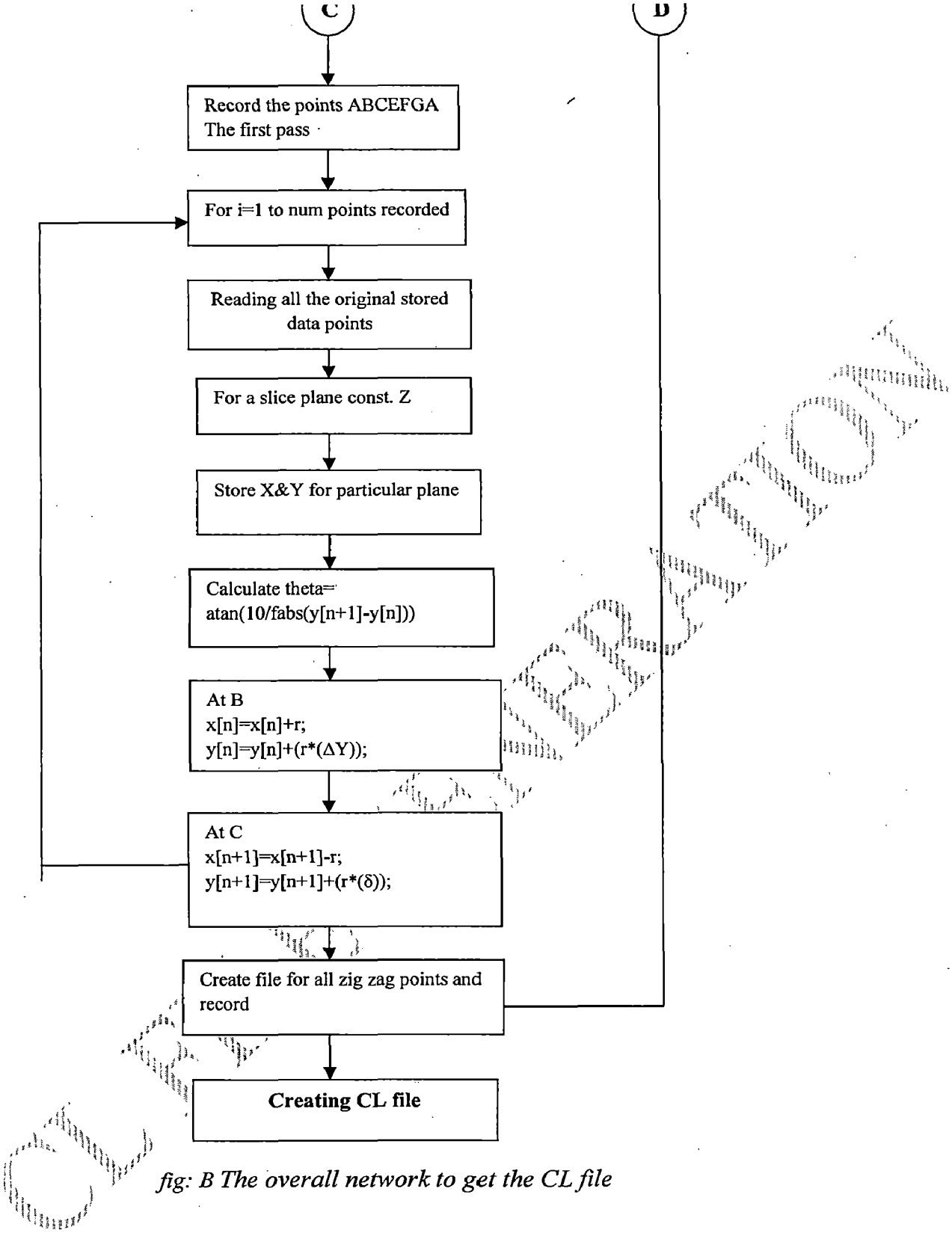


fig: B The overall network to get the CL file