TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta Strojní
Katedra Výrobních Systémů

REALIZACE OBECNÝCH PLOCH
(FREE-FORM SURFACE MANUFACTURING)

Ing. Nguyen Van Tuong

TEZE DISERTAČNÍ PRÁCE

Obor:	Konstrukce strojů a zařízení
Zaměření:	Obráběcí a montážní stroje
Školitel:	Prof. Ing. Přemysl Pokorný, CSc.

Předložená doktorská práce byla vypracována v rámci doktorského studia na Katedře výrobních systémů, Fakultě strojní Technické univerzity v Liberci.

Disertant: Ing. Nguyen Van Tuong
Technická Univerzita v Liberci
Fakulta Strojní
Katedra Výrobních Systémů

Školitel: Prof. Ing. Přemysl Pokorný, CSc.
Technická Univerzita v Liberci
Fakulta Strojní
Katedra Výrobních Systémů

Oponenti:

Teze byla rozeslána dne......................................
Obhajoba disertační práce se koná dne........................,
před komisí v oboru „Konstrukce strojů a zařízení“ Fakulty strojní v..................hodin.

Před obhajobou je možné se doktorskou seznámit prací na děkanátě Fakulty strojní Technické univerzity v Liberci, Hálkova 6, 461 17 Liberec 1.

ABSTRAKT

Povrchy obecných tvarů mohou být obrobeny na 3 - nebo 5-osých NC strojích. Při výběru řezných nástrojů pro obrobení určité obecné plochy, je průměr nástroje omezen určitou hodnotou, která nesmí způsobit místní podřezání při obrábění konkávního a sedlového tvaru. Je zřejmé, že obrábění trvá dlouho, pokud se používá jen jeden nástroj k obrobení celého povrchu. Proto, pokud konkávní či sedlový tvar jsou obrobeny odděleně jedním nebo dvěma odpovídajícími nástroji a ostatní oblasti mohou být rychle obrobeny nástroji které jsou větší, může být čas obrábění značně snížen.

Tato práce se zaměřuje především na vývoj nových postupů obrábění obecných ploch. Odvozeně od dílčích zakřivení povrchu, obecný tvar může být rozdělen na oblasti konvexní, konkávní a sedlové a v těchto oblastech se uplatní technika kódování řetězců k určení ohraničení jednotlivých oblastí. Každou dílčí oblast lze pak obrobit nejvhodnějším nástrojem, který může umožnit velké snížení celkového času k obrobení povrchu. Pro tento účel byl vyvinut Matlab® program který provádí rozdělení povrchu a určí souřadnice všech bodů na hranicích oblastí. Rekonstrukce obecného povrchu a simulace drah nástroje v oddělených částech je uskutečněno pomocí Pro/Engineer® Wildfire®.

V disertační práci jsou některé aplikace prezentovány za účelem ukázat, že navržená metoda může snížit časy obrábění v porovnání s tradičními metodami, v nichž se celá plocha obrobí pouze jedním nástrojem. Některé testy jsou rovněž provedeny na multi-funkčním (víceosém) stroji k ověření a ohodnocení navržené metody. Z výsledků, lze dospět k závěru, že navržená metoda je úspěšná.

Klíčová slova:
Obecná plocha, členění povrchu, 5-osé obrábění, čas obrábění.

ABSTRACT

The free-form surfaces can be machined on 3- or 5-axis CNC machines. When selecting a cutting tool to machine a particular free-form surface, the tool diameter is restricted to an established value that must not cause local gouging when machining the concave and saddle regions. Obviously, it would take long if only one tool is used to machine the entire surface. Therefore, if the concave and saddle regions are machined separately by one or two relevant tools; and the other regions are rapidly milled by the bigger ones, the machining time can be then significantly reduced.

This dissertation mainly focuses on developing a practical method to machine free-form surfaces. Based on the surface curvatures, a free-form surface is partitioned into convex, concave and saddle regions in which the chain code technique is applied to determine the boundaries of each region. Each partitioned region then can be milled by the most suitable tool that can lead a big reduction in the total machining time of the surface. For implementation purpose, a Matlab® program was developed to perform the surface partitioning task and obtain the coordinates of all points on the boundaries. The freeform surface reconstruction and simulation of the tool path of partitioned regions were illustrated via Pro/Engineer® Wildfire®.

In the dissertation, some application examples are presented to show that the proposed method can reduce the machining time, compared to the traditional method in which the whole surface was machined by only one cutter. Some experiments are also done on a multi-tasking machine to validate the proposed method. From the machining results, it can be concluded that the proposed method has been successfully implemented.

Keywords:

free-form surface, surface partitioning, 5-axis machining, machining time.

CONTENT

1. Introduction 6

2. Research objective 6

3. Research methods 7

3.1 Representing the mathematical model of free-form surfaces 7

3.2 Surface partitioning 8

3.2.1 Surface geometry 8

3.2.2 Surface partitioning 8

3.3 Defining the boundaries of regions 10

3.4 Checking interference and correcting tool postures 11

3.5 Generating tool paths 12

3.5.1 Local gouging avoidance and definition of tool diameter 12

3.5.2 Definition of tool orientation and global gouging avoidance 12

3.5.3 Tool path generation 13

3. 6 Machining time comparison 14

3. 7 Machining experiments... 14

4. Applications and results... 14

4.1 Application 1.. 14

4.2 Application 2.. 16

4.3 Application 3.. 19

4.4 Experiments... 22

5. Conclusion... 23

5.1 Summary and conclusions 23

5.2 Contributions.. 24

5.3 Future work.. 24

References.. 25

Publication.. 28

1. INTRODUCTION

Free-form surfaces, also called sculptured surfaces, are widely used in industries. These surfaces are often produced in three stages: roughing, finishing, and benchwork (grinding and polishing). The time spent on the finishing and benchwork stages is dependent on the size of the scallops. It is stated that over 78% of the total production time is spent on finishing, grinding, and polishing.

Traditionally, free-form surfaces are machined on three-axis NC machines using ball-end cutters. In five-axis machining of free-form surfaces, the tool can be ball-end cutter, toroidal cutter or flat-end cutter and the tool orientation can be changed during machining. Regardless of machining in three- or five-axis mode, the productivity will be low if there is only one cutting tool used to machine the entire surface. This is because of the limitation of the tool diameter that depends on the surface curvature. When choosing a tool for a particular surface, the diameter tool is restricted to a determined value that must not cause local gouging when the tool is machining in concave and saddle regions. For examples, if a ball-end cutter is used, the cutter radius must be smaller than the minimum radius of curvature of the machined surface to avoid local gouging.

In general, a free-form surface has regions such as convex, concave, plane and saddle. Therefore, if the concave and saddle regions are machined separately by one or two suitable tools and the other regions are milled faster by a bigger tool, then the machining time may be significantly reduced. To do this, the design surface should be partitioned into regions. In CAD/CAM packages such as Catia®, Pro/Engineer®, Unigraphics NX®,... users can not separately choose every region and its boundary in the computer-aided manufacturing (CAM) programming stage, so that the surface should be partitioned into patches in the modeling stage of computer-aided design (CAD) model of the free-form surface.

Another critical problem in the free-form surface machining is gouging. Local gouging can be avoided if the user chooses the right cutting tool as mentioned above. By using CAD/CAM systems, users can find out the collision (global gouging) when simulating the tool path. Some NC verification systems such as VeriCut®, NCSIMUL®,... also offer the ability that visually identify regions that collision occurs. In these systems, users must manually correct the tool path by adjusting the G codes in the NC verification systems or they have to regenerate the tool path in the CAM systems by restructuring some machining parameters. At the moment, this method is quite popularly used in practice. Although this method is easy to implement, it is time-consuming in most cases when deep concave and saddle areas are involved. However, if the design surface is divided into regions then each region can be machined by a different tool path strategy. In this case, some special techniques can be effectively used to orient the tool in which gouge-free tool paths can be successfully generated for deep concave and saddle regions. In these techniques, the tool axis always goes through a curve or a point.

2. RESEARCH OBJECTIVE

The main objective of this research is to develop a practical machining method that can decrease the machining time and improve the surface quality by subdividing free-form surfaces into patches. In this study, a surface partitioning method that bases on the surface properties is proposed. The boundaries of each region are also defined for further use. The tool position and orientation for collisions free are defined as well.

3. RESEARCH METHODS

In order to achieve this objective, many tasks related to mathematical calculations and CAD/CAM should be implemented. The followings are some main tasks in this research.

1) Describe mathematically free-form surfaces and calculate the surface properties such as surface normal and curvatures.
2) Develop a surface partitioning method to the free-form surface into patches according to the surface properties.
3) Develop a method for determining the boundaries of regions on the design surface.
4) Create the CAD model of the design surface.
5) Generate free-gouge tool paths for the partitioned surface.
6) Compare the proposed methodology to the traditional methodologies in term of machining time by simulations.
7) Perform machining experiments.

In the research, a Matlab® program was developed for calculating tasks. CAD/CAM tasks were done by Pro/Engineer®. The machining experiments were performed on the multi-tasking machine Mazak® Intergrex 100-IV.

3.1 Representing the mathematical model of free-form surfaces

Generally, free-form surfaces may be designed directly on a computer using control points, in forms such as Bezier surface, B-spline surface and NURBS. These surfaces are the most popular forms that define free form surfaces in CAD/CAM systems. In this study, B-spline surfaces are used as examples.

A B-spline surface with control points is defined by

$$S(u,v)=\sum_{i=1}^{n+1}\sum_{j=1}^{m+1}B_{i,j}N_{i,k}(u)M_{j,l}(v) \qquad (1)$$

$$u_{min} \le u \le u_{max}\ ,\ v_{min} \le v \le v_{max}$$

where $N_{i,k}(u)$ and $M_{j,l}(v)$ are the B-spline basis functions in the bi-parametric u and v directions, respectively; $B_{i,j}$ are the vertices of a polygonal control net.

$$N_{i,1}(u)=\begin{cases}1 & \text{if } \mathrm{x_i} \le u \le x_{i+1}\\ 0 & \text{otherwise}\end{cases} \qquad (2)$$

$$N_{i,k}(u)=\frac{(u-x_i)N_{i,k-1}(u)}{x_{i+k-1}-x_i}+\frac{(x_{i+k}-u)N_{i+1,k-1}(u)}{x_{i+k}-x_{i+1}} \qquad (3)$$

$$M_{j,1}(v)=\begin{cases}1 & \text{if } \mathrm{y_j} \le v \le y_{j+1}\\ 0 & \text{otherwise}\end{cases} \qquad (4)$$

$$M_{j,l}(v) = \frac{(v - y_j)N_{j,l-1}(v)}{y_{j+l-1} - y_i} + \frac{(y_{j+l} - v)N_{j+1,l-1}(v)}{y_{j+l} - y_{j+1}} \tag{5}$$

where the x_i and y_j are elements of knot vectors.

3.2 Surface partitioning

3.2.1 Surface geometry

The followings are some geometric parameters of a surface *S(u,v)*

- The unit normal vector, *n*, at a point (*u*,*v*) on a surface *S(u, v)* can be computed from the relation:

$$n(u,v) = \frac{S_u \times S_v}{\left|S_u \times S_v\right|} \tag{6}$$

where S*u* and S*v* are the tangent vectors along *u* and *v* parametric directions.

- The Gaussian curvature, K, and the mean curvature, H, of the surface S(*u, v*), at a point P(x, y, z), is formulated as

$$K = \frac{LN - M^2}{EG - F^2} = K_{\min} K_{\max} \tag{7}$$

$$H = \frac{1}{2}\left(\frac{EN - 2FM + GL}{EG - F^2}\right) = \frac{1}{2}(K_{\min} + K_{\max}) \tag{8}$$

where

$$\mathrm{E} = \frac{\partial S}{\partial u}\frac{\partial S}{\partial u}\ ;\ \mathrm{F}= \frac{\partial S}{\partial u}\frac{\partial S}{\partial v}\ ;\ \mathrm{G}= \frac{\partial S}{\partial v}\frac{\partial S}{\partial v} \tag{9}$$

$$\mathrm{L} = n\frac{\partial^2 S}{\partial u^2}\ ;\ \mathrm{M}= n\frac{\partial S^2}{\partial u \partial v}\ ;\ \mathrm{N}= n\frac{\partial^2 S}{\partial v^2} \tag{10}$$

$K_{\min}$ and $K_{\max}$ are principal curvatures, which are the maximum and minimum of the normal curvature, given by:

$$\begin{aligned} K_{\max} &= H + \sqrt{H^2 - K} \\ K_{\min} &= H - \sqrt{H^2 - K} \end{aligned} \tag{11}$$

K and H are used for surface partitioning according to the surface shape relationship with geometric parameters as shown in Table 1. The surface, in this research, will be partitioned into patches of concave, saddle and convex (included plane).

Gaussian curvature (K)	Mean curvature (H)	Local surface shape
$K \geq 0$	$H < 0$	Convex
$K \geq 0$	$H > 0$	Concave
$K < 0$	$H < 0$ or $H > 0$	Saddle
$K = 0$	$H = 0$	Plane

Table I: Surface curvatures and local surface shape at a point

3.2.2 Surface partitioning

The objective of surface partitioning is to divide a free-form surface into a number of regions. In each region the points have the same characteristics. These regions, also called patches, can be machined separately with different types of cutter and setups. In this study, a free-form surface is partitioned into convex, concave, and saddle regions, based on the Gaussian curvature and the mean curvature. For implementation, a numerical method should be applied.

The following are main steps for surface partitioning.

a) Presenting the mathematical model of the free-form surface,

b) Sampling the surface in u and v directions to get a set of grid points,

c) Calculating Gaussian curvature (K) and mean curvature (H) at each grid point,

d) Grouping the neighboring concave and saddle points to form concave regions and saddle regions, respectively. The last portion of points forms convex regions (including plane regions).

The specific algorithm for surface partitioning is as follows:

Algorithm1: partitioning free-form surfaces

```
Input  A free-form surface S(u,v) {mathematical model}
Output Sets of points belonging to three type of regions {convex+plane, concave and saddle}
Begin
1.  Sample the input surface to get a set of grid points {p} and store all points in matrix M
2.  Calculate parameters K and H at every point $\mathbf{p}_{i,j}$
3.  FOR each grid point $\mathbf{p}_{i,j}$ of the set point {p}
     IF $K \geq 0$ and $H \leq 0$ THEN save that point in matrix $M_1$ {data matrix contains points on convex and plane regions}
     END IF
     IF $K \geq 0$ and $H > 0$ THEN save that point in matrix $M_2$ {data matrix contains points on concave regions}
     IF $K < 0$ THEN save that point in matrix $M_3$ {data matrix contains points on saddle regions}
     END IF
 END LOOP
End
```

When saving points of each region into a matrix, the following method is used. After grouping points on the surface, for each group of points, the points are encoded into an integer and saved with their indices in a matrix. The points which do not belong to that group are then encoded as zeros. The encoding procedure of the points for each region R_k can be expressed by the characteristic function as follows

$$\chi(x,y,z,k) = \begin{cases} 1\ (2 \text{ or } 3) & \text{if } (x,y,z) \in R_k \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where k denotes the region of convex, concave or saddle.

In algorithm1, matrices M_1, M_2 and M_3 have the same size as that of the matrix M. They contain the identification numbers of points which belong to convex, concave and saddle regions, respectively. In these matrices, the points on the surface of the same region are presented by a number, say 1 or 2 or 3, and the other points of other regions are presented by zero. Each character in the matrix owns its indices that represent the location of the corresponding point on the surface in *u* and *v* directions. Hence, the coordinates of points on the surface are maintained. These matrices are very useful for further calculations. Figure 1 below shows the structures of matrices M_1 M_2 and M_3.

0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	0	0	1	1	0	0
1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

M_1

2	2	2	2	2	2	2	0	0
0	0	2	2	2	2	0	0	0
0	0	0	2	2	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	2	2	0	0	0	0
0	0	2	2	2	2	0	0	0
2	2	2	2	2	2	2	0	0

M_2

0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	3	3	3
0	0	0	0	0	0	3	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3

M_3

Figure 1: An example of the structure of matrices M_1 M_2 and M_3.

3.3 Defining the boundaries of regions

The boundaries of a patch should be defined so that they can be used as the factors for trimming off the patch from the entire surface when modeling that surface in CAD/CAM packages. One solution that can be used to define the patch boundaries is to connect the grid points on the border of each region after partitioning. The boundaries of these patches can be represented by applying the chains codes method from the image processing field.

In image processing field, chain codes are used to represent a boundary of an object in a binary image. In the Freeman chain code technique, the boundary representation is based on 4-connectivity or 8-connectivity of segments. The direction of each segment is coded by using a numbering scheme illustrated in Figure 2. If the partitioned surfaces are coded as binary images and the concave, convex or saddle regions are considered as objects on the images, hence the boundaries of these objects can be obtained by tracing the exterior pixels.

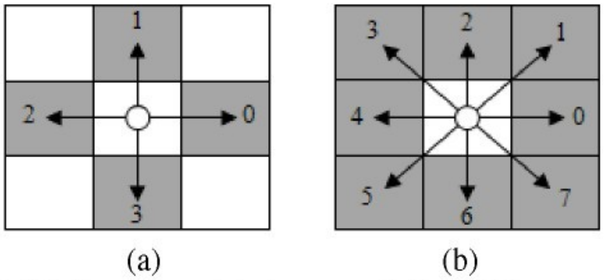


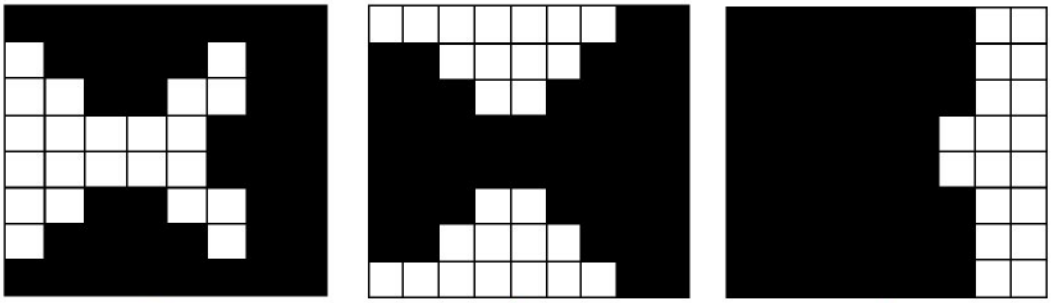
Figure 2: Chain coding: (a) 4-connectivity, (b) 8- connectivity.

As mentioned above, each data matrix after partitioning ($M_{1,}$ M_2 and M_3) contains only two values of number, 0 and 1 or 2 or 3 depending on regions of convex, concave or saddle, respectively. The structure of these matrices is similar to the structure of the matrix of pixels of a binary image which is a digital image that has only two possible values for each pixel. In the correlative images of these matrices, the background is 0s and the foreground is 1s or 2s or 3s, as shown in Figure 3.

An M-function named "boundaries" in Matlab® language for finding the boundaries of objects in a binary image based on the Freeman chain code is presented in Ref. [11]. This function traces the exterior boundaries of the objects in a binary image with pixels 0 as the background. The syntax of this function is as follows

$$B = boundaries(f, conn, dir) \qquad (13)$$

where f is the binary image, *conn* specifies the desired connectivity of the output boundaries and its values are 4 or 8 (the default), *dir* can be *cw* (the default) or *ccw* that specifies the direction in which the boundaries are traced clockwise or counter clockwise direction, and output B is a cell array that contains the coordinates of the boundaries found.



I_1 I_2 I_3

Figure 3: The correlative images of matrices M_1, M_2 and M_3.

Because matrices M_1, M_2 and M_3 can be considered as the matrices of binary images, the binary image f in Function (13) can be replaced by matrices M_1, M_2 and M_3. The result is that the coordinates of all points on the boundaries of the regions can be defined. These coordinates are not the coordinates of the points on the design surface but they own their indices that are equivalent to those of the points on the surface in u and v direction. Hence, some further calculations are required to get the coordinates of points on the boundaries on the three-directional (3D) surface.

3.4 Create the CAD model of free-form surfaces

Because Pro/Engineer® does not support to directly create B-spline surfaces from their control points, so that a replacement technique is required. In our research, ibl files (files with the extension .ibl) are used to create B-spline surfaces from imported datum curves. In the ibl file format, there is a list of points for each curve. These point are represented with x, y and z coordinates and they are gathered from the Matlab® program.

From the results of executed Matlab® program, the coordinates of the boundary points are also obtained. These points are used to create the boundary curves. In Pro/Engineer®, the Offset Coordinate System Datum Point Tool can be used to create the spline going through points. The Surface Trim tool in the Style tool can be used to divide the design surface into regions. In this case, the boundary curves are the trimming factors.

3.5 Generate free-gouge tool paths for the partitioned surfaces

3.5.1 Local gouging avoidance and definition of tool diameter

A ball-end cutter will not cause any local gouges on the design surface if the cutter radius is smaller than the smallest radius of the surface curvature in the selected region. In this reseach, the possible maximum radius of a ball-end cutter can be executed by calculating the maximum value ($K^S_{\max}$) of the maximum curvatures of the selected region. The possible maximum radius of the cutter is calculated by

$$R_{\max} = \frac{1}{K^S_{\max}}$$

For toroidal and flat-end cutters, to avoid local gouging, the effective radius of the cutter must be smaller than the minimum radius of curvature of the surface. The effective radius of the flat-end cutter, R_{eff}, can be compute as

$$R_{eff} = \frac{R}{\sin\theta} \tag{1.2}$$

where R is the radius of the cutter and θ is the inclination angle between the unit normal and the tool axis, also called the Sturz angle, as shown in Figure 4.

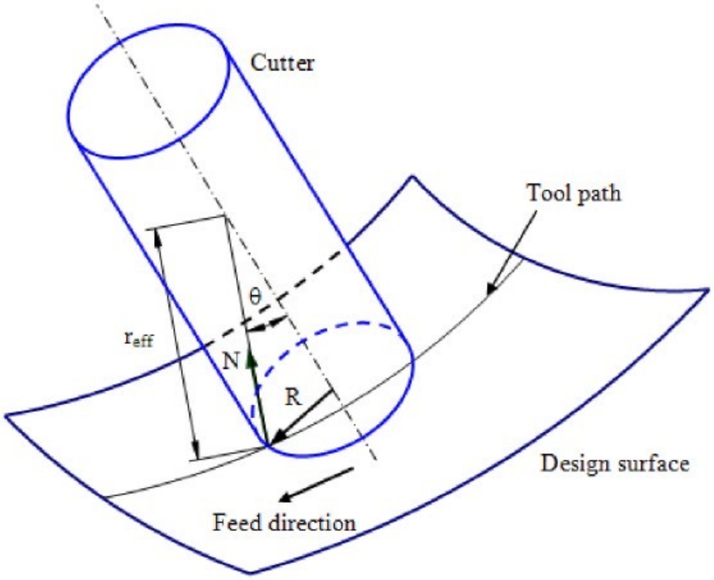


Figure 4: Effective radius of a flat-end cutter used for 5-axis surface machining.

From the possible maximum radius of the cutter, the tool diameter for each region can be easily defined. Howerver, sometimes a smaller diameter should be applied to avoid gouging. A combination of reducing the tool diameter and choosing the suitable Sturz angle may also be uztilized for global gouging avoidance.

3.5.2 Definition of tool orientation and global gouging avoidance

In 5-axis milling lead and tilt angles are used to define the tool orientation. The lead angle is the rotation of the tool axis about the cross-feed axis, whereas the tilt angle is the rotation about the feed axis with respect to the surface normal. In practice, the lead and tilt angles are often selected by trial and based on experiences of users. In

Pro/Engineer®, these angles are specified by parameters LEAD_ANGLE and TILT_ANGLE. In each NC sequence, when establishing the machining parameters, the values of the lead and tilt angles should be entered by trial as follows.

1. Choose primary values for the lead and tilt angles:
2. Choose the surfaces to be detected. They are the neighbouring regions of the region to be machined.
3. Simulate the tool paths and perform gouge checking.
    - If there are no gouges, accept the values of the lead and tilt angles
    - If gouges occur, define the maximum absolute value of the gouge distance.
4. Recalculate the values of the lead and tilt angle based on the maximum absolute value of the gouge distance.
5. Repeat steps 3 and 4 until no gouges occur.

3.5.3 Tool path generation

For comparison the machining time of the finish pass between the proposed method and the conventional method, some constrains should be applied for both methods when generating the gouge-free tool paths. More concretely, the machining strategy and machining conditions for the two methods should be similar. Regardless of machining by any tool, the values of the following parameters for finishing should be the same: (1) cutting speed, (2) feed rate per tooth, (3) stock left after rough machining, and (4) scallop height.

The tool path generation methods that can maintain a constant scallop height are called the *iso-scallop tool path generation* methods. In these methods, the tool path interval is variable so that the scallop height is constant along the CC path. Scallops are ridges, cusps and other surface protrusions left between adjacent overlapping tool passes that extend above the design surface profile. The tool pass interval, the tool type and surface characteristics determine the size of scallops left on the surface. In Pro/Engineer®, the parameter SCALLOP_HGT is used to control tool step by specifying the maximum allowable scallop for surface milling. Figure 5 represents scallops left after machining a plane by a ball-end cutter.

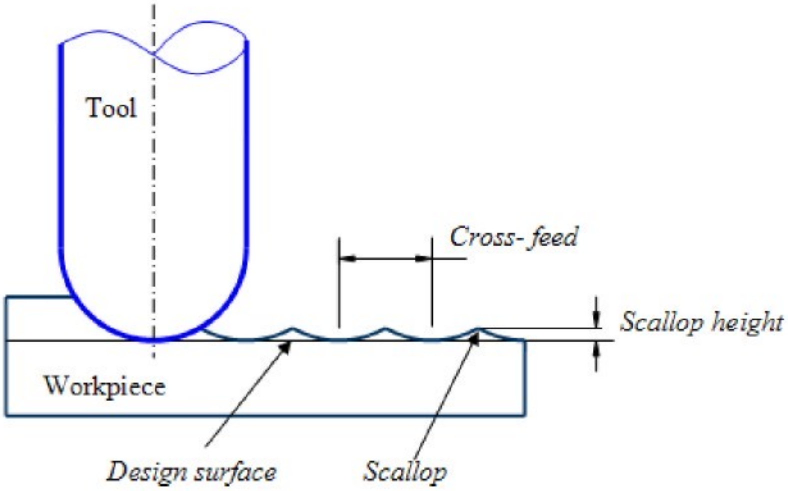


Figure 5: Scallops left after machining

3.6 MACHINING TIME COMPARISON

In this research, machining time is considered as an important criterion to evaluate the machining efficiency. The machining time for finish cut should be compared between the proposed method and the traditional one for machining a given surface. In order to get the machining time of the operation for each application, the NC program was generated by a post processor of the 5-axis CNC Shoda Router. This post processor is available inside Pro/Manufacturing®.

3.7 MACHINING EXPERIMENTS

In order to confirm that the proposed method is valid in practice, some machining experiments were conducted on the multi-tasking machine Mazak® Integrex IV-100.

4. APPLICATIONS AND RESULTS

A number of applications have been implemented in our research. At first, some simple cases were conducted for easy implementation and checking the developed algorithms. After having confirmed that all algorithms worked well based on those simple cases, some complicated cases were carried out. For comparison purposes, the machining parameters applied for the two methods are the same to all application, as shown in Table 2. Three typical examples of B-spline surfaces are presented in the dissertation.

Parameter	Rough	Finish
Cutting speed (m/min)	35	50
Feed (mm/rev/tooth)	0.2	0.1
Scallop height (mm)	-	0.02
Left stock (mm)	0.5	0
Depth of cut (mm)	3	0.5
Width of cut (mm)	3	tool radius

Table 2: Machining parameters.

4.1 Application 1

Figure 6 shows a B-spline surface defined by 5 × 5 control net as given in Table 3 with the uniform knot vector [0 1 2 3 4 5 6 7] in both parametric directions.

(-40,-40,0)	(-20,-40,0)	(0,-40,0)	(20,-40,0)	(40,-40,0)
(-40,-20,0)	(-20,-20,0)	(0,-20,0)	(20,-20,0)	(40,-20,0)
(-40,0,0)	(-20,0,0)	(0,0,-30)	(20,0,0)	(40,0,0)
(-40,20,0)	(-20,20,0)	(0,20,-40)	(20,20,0)	(40,20,0)
(-40,40,0)	(-20,40,0)	(0,40,-20)	20,40,0)	40,40,0)

Table 3: Control net of a B-spline surface

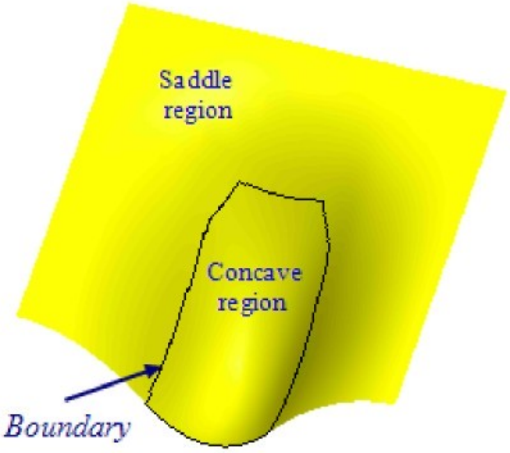


Fig. 5 Partitioned surface (application1).

The machining strategy for this surface can be as follow

1) Roughing cut in 3-axis mode by a two-flute flat-end mill of 10 mm in diameter,
2) Finishing cut in 5-axis mode for the saddle region by a four-flute ball-end cutter of 20 mm in diameter,
3) Finishing cut in 5-axis mode for the concave region by a two-flute ball-end cutter of 10 mm in diameter.

For comparison purpose, the finishing cut is also performed for the entire surface by a two-flute ball-end cutter of 10 mm in diameter. Figures 6 below shows an example the tool paths generated for each region under the condition that the cutter axis always goes through a fixed straight line over the stock. The machining simulation are given in Figure 7.

Table 4 presents the results about the machining time and tool path length for the finishing cut estimated by Pro/Engineer®. The tool path length for finishing of the proposed method approximately equals 85% to that of the traditional method. This reduction leads to a saving up to nearly 20% of the machining time for finishing cut of the proposed method, compared to that of the traditional method. It is a very impressive number. Hence, in this case, it can be said that the proposed method is really effective in term of machining time.

Method	**Tool path length (mm)**	**Estimated machining time (min)**
Traditional method (un-partitioned surface)	516.96 (100%)	10.71 (100%)
Proposed method (partitioned surface)	442.14 (85.5%)	8.61 (80.4%)

Table 4: Estimated machining time and tool path length comparison (application 1)

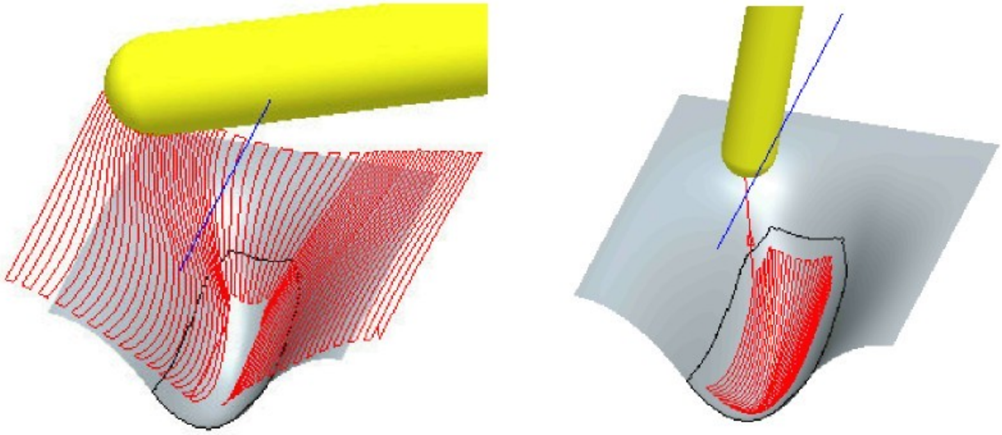


Figure 6: Example of tool paths for regions (application 1).

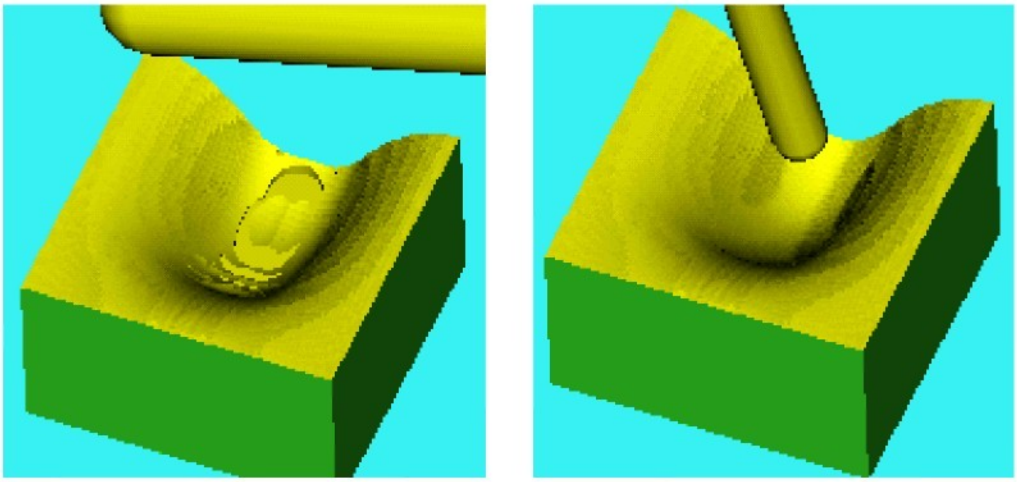


Figure 7: Machining simulation (application 1): after the finishing cut for the saddle region (left) and after the finishing cut for the concave region (right).

4.2 Application 2

Figure 8 shows a B-spline surface defined by 5 × 4 control net as given in Table 5 with the uniform knot vector [0 1 2 3 4 5 6 7] in *u* direction and [0 1 2 3 4 5 6] in *v* direction. In this example, the surface is divided into four patches, a convex patch, a saddle patch and two concave patches.

(0,0, 10)	(0,15,30)	(0,30,40)	(0,50,50)	(0,70,40)	(0,85,30)	(0,100,10)
(15,0,30)	(15,15,0)	(15,30,50)	(15,50,60)	(15,70,50)	(15,85,0)	(15,100,30)
(30,0,40)	(30,15,60)	(30,30,70)	(30,50,70)	(30,70,70)	(30,85,60)	(30,100,40)
(50,0,45)	(50,15,60)	(50,30,70)	(50,50,75)	(50,70,70)	(50,85,60)	(50,100,45)
(70,0,40)	(70,15,60)	(70,30,70)	(70,50,70)	(70,70,70)	(70,85,60)	(70,100,40)
(85,0,30)	(85,15,50)	(85,30,40)	(85,50,10)	(85,70,40)	(85,85,50)	(85,100,30)
(100,0,5)	(100,15,15)	(100,30,30)	(100,50,20)	(100,70,30	(100,85,15)	(100,100,5)

Table 5: Control net of a B-spline surface for application 3

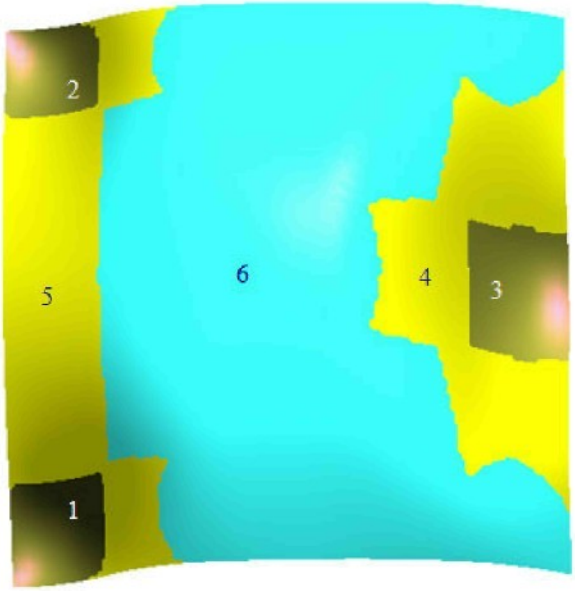


Figure 8: Partitioned surface (application2)
1, 2 & 3- concave regions; 4 & 5- saddle regions; 6- convex region.

The machining strategy for this case can be as follow:

1) Roughing cut in 3-axis mode by a 2-flute flat-end mill of 12 mm in diameter,
2) Finishing cut in 5-axis mode for the convex region by a 4-flute ball-end cutter of 25 mm in diameter,
3) Finishing cut in 5-axis mode for the two saddle regions by a 2-flute ball-end cutter of 18 mm in diameter,
4) Finishing cut in 5-axis mode for the three concave regions by a 2-flute ball-end cutter of 6 mm in diameter.

In order to compare to the traditional method, the un-partitioned surface is milled by a 6 mm two-flute ball-end cutter with the same machining parameters and strategies.

The lead angle (α) and tilt angle (β) for regions as follows:

- Region 1: $\alpha = \beta = -5$ (deg)
- Region 2: $\alpha = \beta = 5$ (deg)
- 3, 4, 5, 6: $\alpha = 5$ (deg); $\beta = 0$ (deg)

The tool paths for each region and the machining simulation of this application are illustrated in Figure 9.

The machining time estimation and the tool path length of this application are illustrated in Table 6.

Method	**Tool path length (mm)**	**Estimated machining time (min)**
Traditional method	2000.99 (100%)	42.21 (100%)
Proposed method	1282.50 (64.1%)	50.03 (118.5%)

Table 6: Estimated machining time and tool path length comparison (application 2)

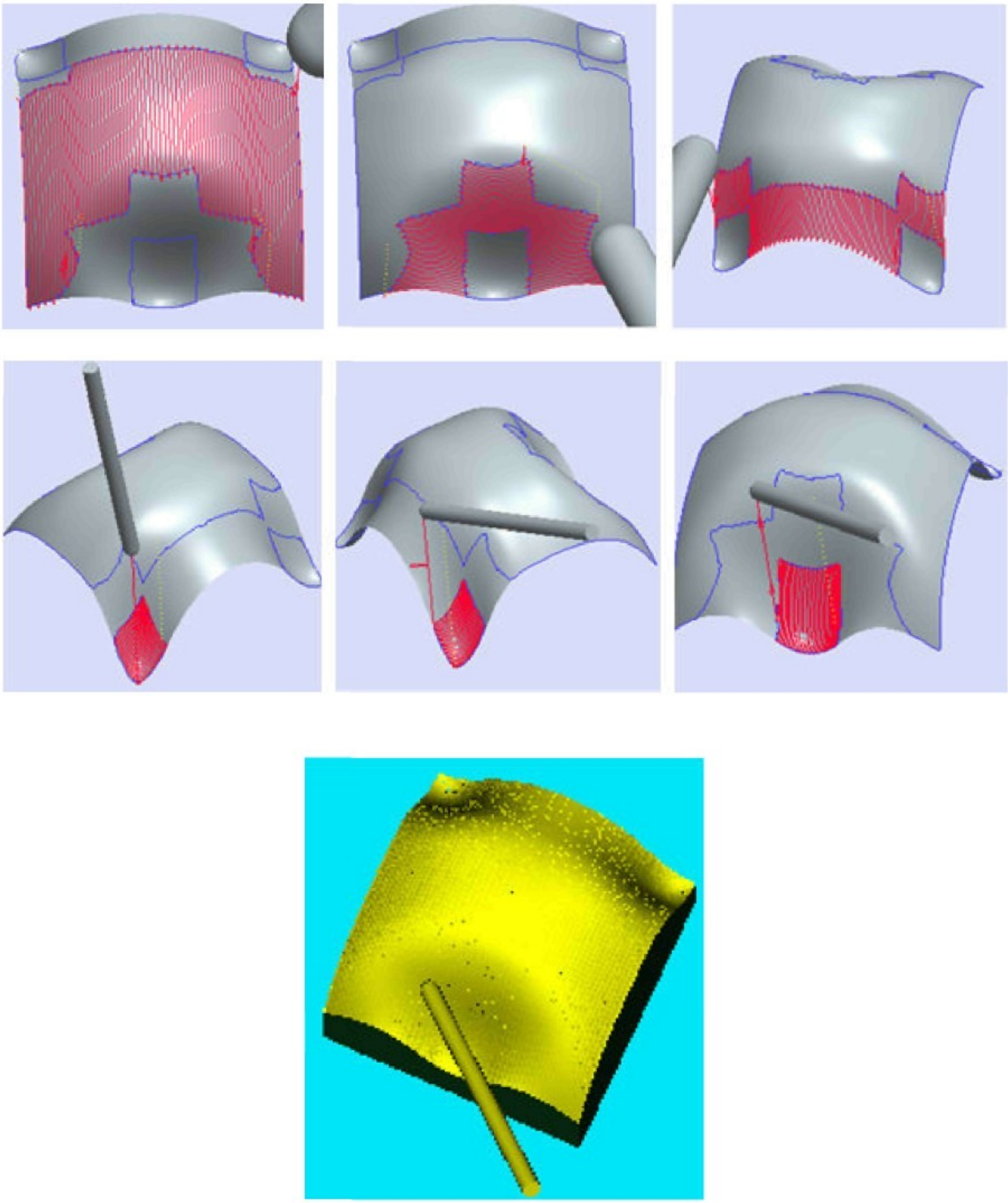


Figure 9: Tool paths for each region and simulation result.

By applying the strategy above, it takes 50.03 minutes for finishing by the proposed method. Meanwhile, it takes 41.21 minutes for the finishing cut by the traditional method. It means that the strategy above gives a negative effect on machining time. There are some other strategies that can be applied to get a shorter time for machining the surface above. The following are two possibilities.

1) The strategy above is also used but the convex region is now mill by a 2-flute flat end cutter of 10 mm in diameter. In this case the tool path length and the estimated machining time of the finishing cut are 968.77 mm and 39.42 minutes, respectively. The finishing time of this strategy is about 7% shorter than that of the traditional method.

2) The surface is partitioned into 4 regions. It consists of 3 concave regions and the last. Here, there is a combination of two saddle regions 4, 5 and convex 6 (Figure 8) into one region. The combined region is milled by an 8 mm 2-flute flat-end cutter. The tool path created for this region is illustrated as shown in Figure 10. In this case the tool path length and the estimated machining time of the finishing cut are 759.21 mm and 27.68 minutes, respectively. The finishing time of this strategy is about 34% shorter than that of the traditional method. Hence, it can be said that with this machining strategy the proposed method is much better than the traditional method in terms of machining time.

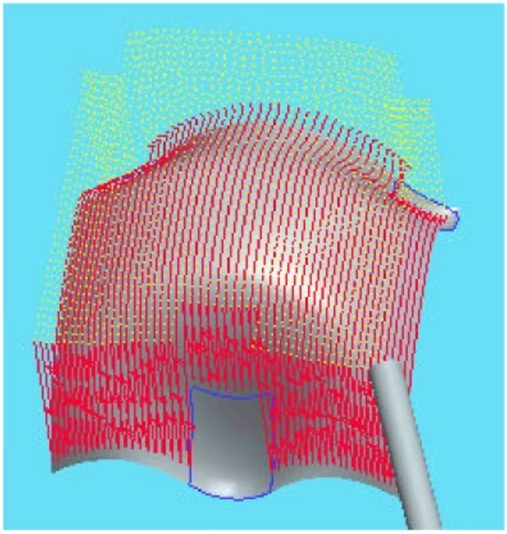


Figure 9: Tool paths for the combined region.

4.3 Application 3

In this application, the proposed method is applied on a B-spline surface whose control net is given in Table 7 and the uniform knot vectors are [0 1 2 3 4 5 6 7] in u direction and [0 1 2 3 4 5 6] in v direction. This surface is divided into six regions: one convex region, three saddle region and two concave regions. The CAD model of the design surface with its regions created in Pro/Engineer® is represented in Figure 10.

(0,0,0)	(0,25,25)	(0,50,25)	(0,75,0)
(20,0,0)	(20,25,50)	(20, 50,25)	(20,75,50)
(40,0,0)	(40,25,75)	(40,50,50)	(40,75,75)
(60,0,0)	(60,25,50)	(60,50,25)	(60,75,50)
(80,0,0)	(80,25,25)	(80,50,25)	(80,75,0)

Table 7: Control net of a B-spline surface for application 3

The machining strategy for this surface can be as follow

1) Roughing cut in 3-axis mode by a flat-end mill of 10 mm in diameter,
2) Finishing cut in 5-axis mode for saddle region 3 by a 4-flute ball-end cutter of 24 mm in diameter,
3) Finishing cut in 5-axis mode for the two concave regions by a 4-flute ball-end cutter of 35 mm in diameter,
4) Finishing cut in 5-axis mode for the convex region and saddle regions 2 by a 2-flute end mill of 10 mm in diameter.

The postures of all tools are defined by the lead and tilt angle which are 5 degrees and 0 degree, respectively.

Like other applications, the original surface was also planned to machine with the same strategy and parameters by a 4-flute ball-end cutter of 24 mm in diameter. The machining time for this surface is then compared with the partitioned one, as shown in Table 8. Figure 11 shows the tool paths for each regions and the machining simulation result.

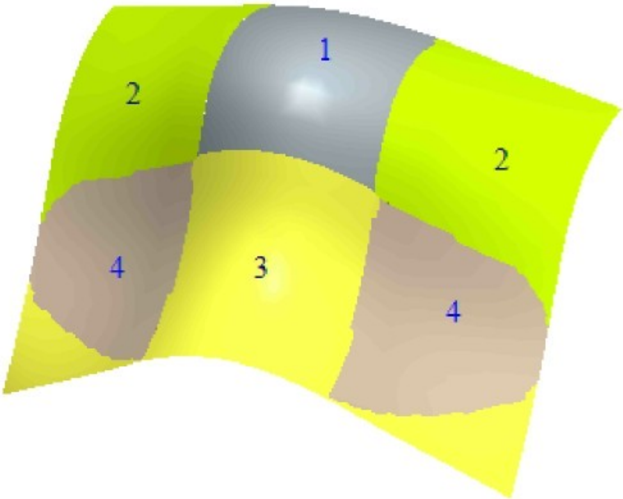


Figure 10: Partitioned surface (application 3).
1- convex region, 2 & 3- saddle region; 4- concave region;

Method	Tool path length (mm)	Estimated machining time (min)
Traditional method	430.78 (100%)	17.07 (100%)
Proposed method	303.66 (70.5%)	15.63 (91.5%)

Table 8: Estimated machining time and tool path length comparison (application 3)

From Table 8, it can be seen that, in this application, the proposed method also has an advantage in terms of machining time. This is because in the finishing stage the tool path length of the proposed method is 29.5% shorter than that of the traditional method. And the result is that it takes 15.63 minutes for finishing region by region of the surface. While, if the whole surface is machined by a ball-end cutter of 24 mm in diameter, the machining time is 17.07 minutes, about 9% longer than the former.

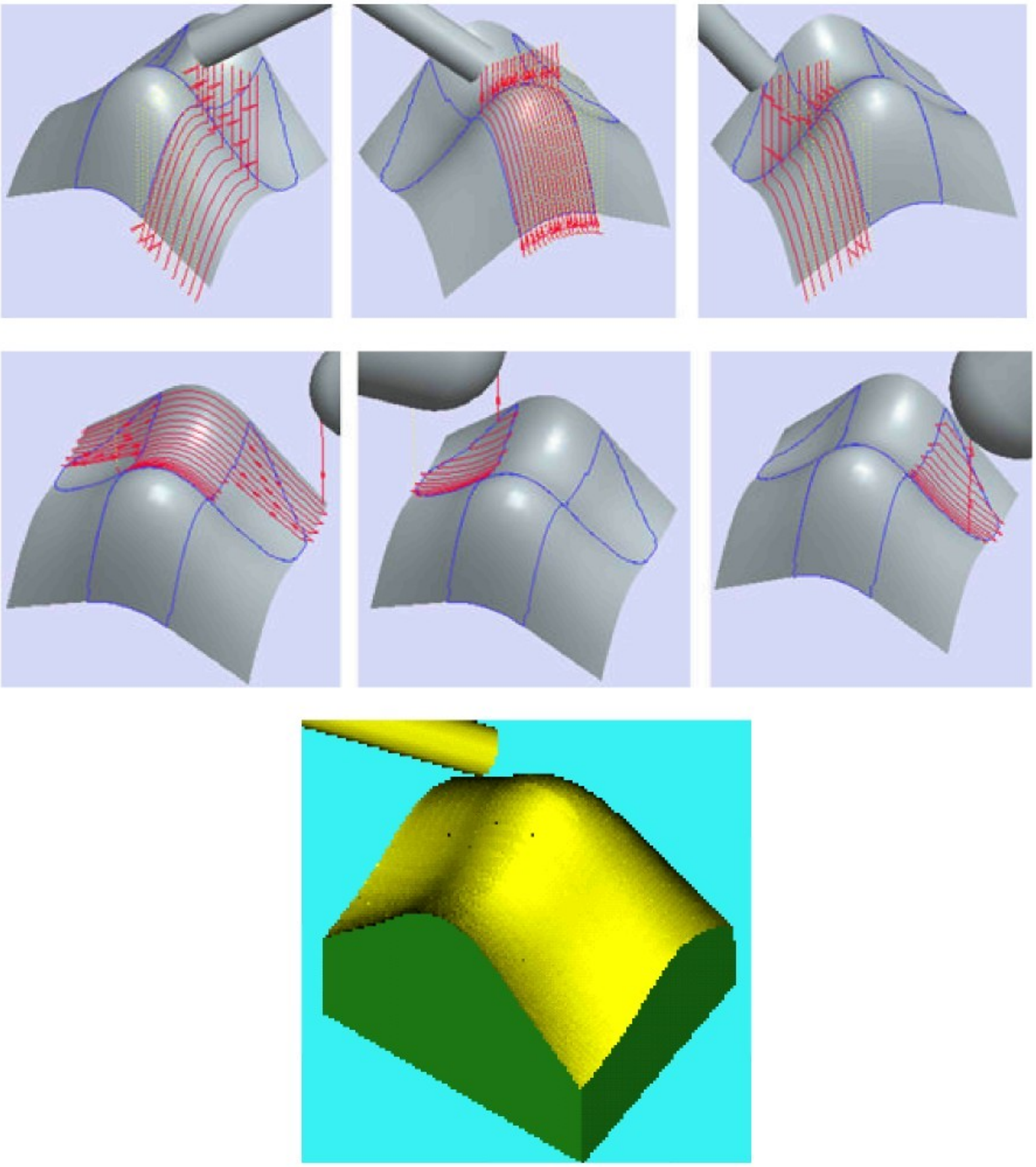


Figure 11: Tool paths for each region and machining simulation result (application 3).

According to the distribution of regions 1 and 2 (Figure 10), it is not necessary to machine them separately. These regions can be jointed and machined by the same tool as presented above. Because the maximum curvatures of region 3 and 4 (Figure 10) are quite high and the difference between them is not much, these regions can also be combined into one. This combined region can be milled by a 24mm 4-flute ball-end cutter. The tool paths for these two large regions are shown in Figure 11.

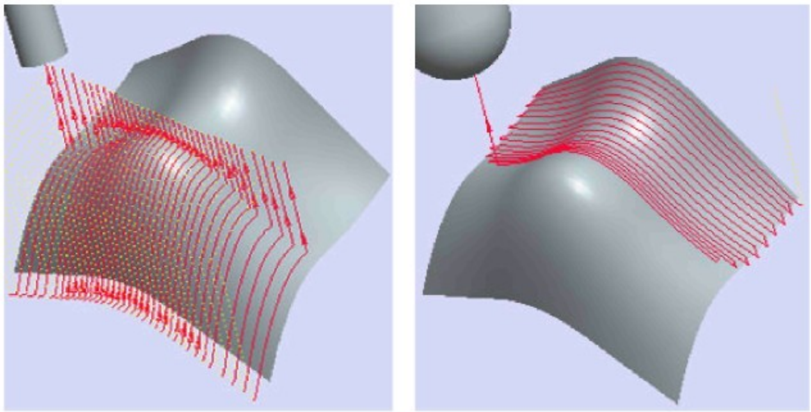


Figure 11: Tool paths for the two combined regions.

The simulation result shows that the tool path length for the two combined regions is 305.19 mm and the estimated machining time is 13.20 minutes. It means by dividing the design surface into 2 regions, the machining time reduces up to 23%, compared to the conventional method.

The three applications above show that in particular situations the machining time can considerably reduce if the freeform surfaces are partitioned in to convex, concave and saddle regions. This is a great potential for machining that kind of surfaces. However, it is realised that sometimes the design surface should not be divided into a smaller number of regions which it has in nature to get a higher machining efficiency.

4.4 Experiments

In order to validate the implementation in practice, some experiments are required. The followings are two samples machined on the Integrex 100-IV machine. The materials of the workpieces are made from artificial wood.

Figure 12 shows the result of the machining sequences of the surface mentioned in application 1. In this test, the rough cut was done by a 10 mm end mill. The concave and saddle regions were milled by 6 mm and 12 mm ball end cutters, respectively. The scallop height on the surface was set to 0.02 mm.

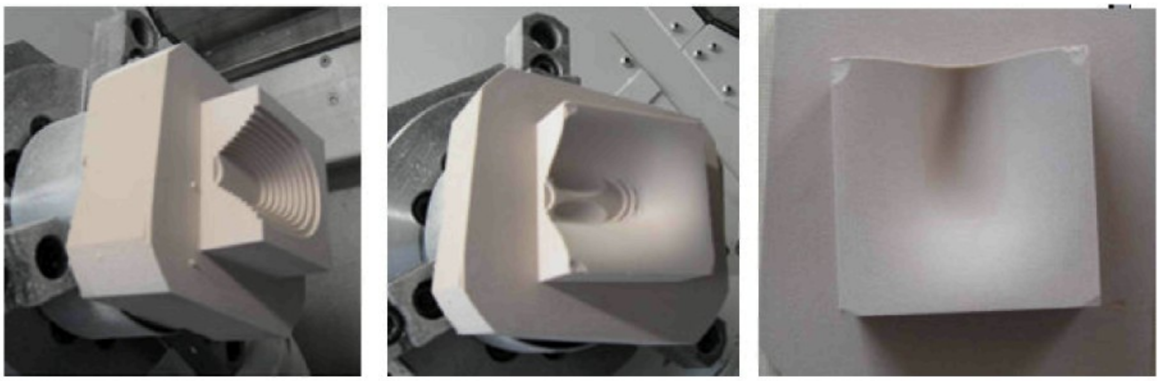


Figure 12: Machining sequences of the surface (application 1).

In Figure 13 are the results of the machining sequences of the surface in application 2. In this test, the rough cut was done by a 10 mm end mill. The finishing cut for convex, concave and saddle regions was done by a 12 mm ball end cutters. To see regions easily, the tool paths for each region were planed in different directions from the neighbour ones. The scallop height on the surface was set to 0.05 mm.

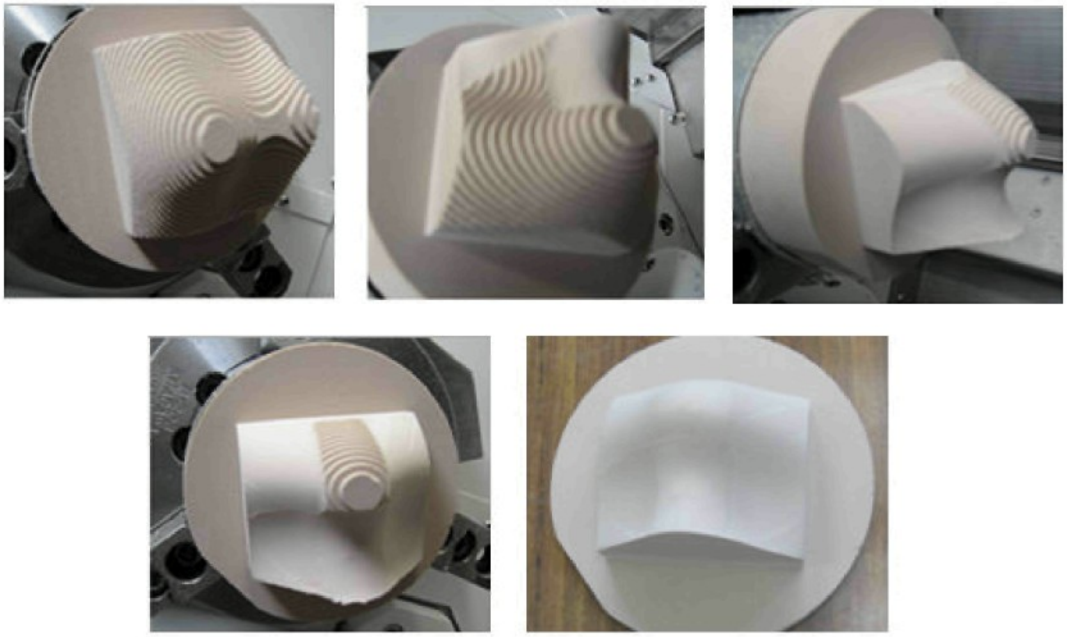


Figure 13: Machining sequences of the surface (application 2).

The fact that the tests did not use the same tools and machining parameters, which illustrated in Section 6.3, does not affect the validation of the implementation in practice. The test results still show that the proposed method has been successfully implemented.

5. CONCLUSIONS

5.1 Summary and conclusions

In the research, the algorithms for surface partitioning and defining the boundaries of regions were developed and successfully implemented. Compared to the traditional method, the proposed method can considerably decrease the machining time for the finishing cut. Although there are only few experiments done, but from the test results, it can be stated that the proposed method has been successfully verified. The proposed method can be applied in practice. Like other methodologies, machining free-form surfaces based on partitioning has advantages and disadvantages.

The main advantages of the proposed method are listed below:

1) The approach is practical and it is easy to perform.
2) By applying the proposed method, the machining efficiency can be improved.
3) The surface quality can be improved.
4) It is easy to create gouge-free tool paths for regions where the Sturz angle method cannot be applied.

Some main drawbacks of the dissertation can be enumerated as follows:

1) Other kinds of free-form surfaces such as Bézier surface and NURBS surface

are not accepted for the Matlab® program developed for this study.

2) To get a high accuracy of the boundaries, a long calculating time in Matlab® when large surfaces are involved.
3) The method for determining the tool orientation when generating the tool paths is tedious and time-consuming in most cases.

5.2 Contributions

Some important contributions of the dissertation are:

1) This dissertation proposed and developed a new method that allows to machine free-form surfaces based on surface partitioning. This method is practical and very workable, and can be applied in industry.
2) The dissertation brought out an effective method for surface partitioning. The applications in the dissertation show that this method is robust and efficient.
3) A Matlab® program was developed to fulfill the calculating tasks in the research such as surface partitioning, finding the boundaries of regions on the free-form surfaces, and so forth.

5.3 Future work

The following are some recommendations for future work.

1) *Adding some other kinds of free-form surfaces such as Bézier and NURBS surfaces*:

   At present, the Matlab® program is only available for B-spline surfaces. In practice, Bézier and NURBS surfaces are also very popular in definition of free-form surfaces. Therefore, it is necessary to expand the ability of the program to diversify applications of Bézier and NURBS surfaces.
2) *Developing an active method for determining the Sturz angle for each region*:

   An algorithm for global gouging detection and correction can be proposed and should be implemented in the early stage. The result of the algorithm will be the value of the Sturz angle that defines the tool posture without gouging. From the value of the Sturz angle of each region, the lead and tilt angles can be defined and they are used as some important input parameters in the CAM stage.
3) *Developing a plug-in integrated in some CAD/CAM packages*

   For the CAD/CAM packages that allow implementation of algorithms and macro-programming works, the application of the proposed surface partitioning method would be useful, and can be customized by the end-user for the specific computer aided process planning tasks. The surface partitioning plug-in could therefore be developed and integrated into commercially available packages.

REFERENCES

[1] Bey M., Bendifallah M., Kader S.and Boukhalfa K. Cutting tool combination and machining strategy affectation based on the determination of local shapes for free form surfaces, *International Conference on Smart Manufacturing Application*, 2008, pp 120-125.

[2] Budak E., Ozturk E., and Tunc L.T. Modeling and simulation of 5-axis milling processes, *CIRP Annals - Manufacturing Technology*, 2009, Vol 58, pp 347–350.

[3] Chang T.C., Wysk R.A., and Wang H.P. Computer-aided Manufacturing, Prentice Hall, New Jersey, 1998.

[4] Chen Z.C., Dong Z. and Vickers G.W. Automated surface subdivision and tool path generation for 3½½-axis CNC machining of sculptured parts. *Computers in Industry*, 2003, Vol 50, No 3, pp 319-331.

[5] Chiou C.J. and Lee Y.S. A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Computer-Aided Design*. Vol 34, No 5. pp 357-371.

[6] Choi B.K. and Jerard R.B. Sculptured surface machining: Theory and applications. Springer-Verlag, New York, 1999.

[7] Ding S., Mannan M.A., Poo A.N., Yang D.C.H. and Han Z. Adaptive iso-planar tool path generation for machining of free-form surfaces. Computer-Aided Design, 2003, Vol 35, No 3, pp 141-153.

[8] Dragomatz D. and Mann S. A classified bibliography of literature on nc tool path generation. *Computer-Aided Design*, 1997, Vol 29, No 3.

[9] Elber G. and Cohen E. Second order surface analysis using hybrid symbolic and numeric operators. *Transactions on Graphics*, 1993, Vol 12, No 2, pp 160-178.

[10] Feng H.Y. and Li H. Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer Aided Design*, 2002, Vol 34, No 9, pp 647–654.

[11] Gonzalez C.R., Woods R.E. and Eddins S.L. Digital image processing using matlab. Prentice Hall, 2004.

[12] Gray P., Bedi S. and Ismail F. Rolling ball method for 5-axis surface machining, *Computer-Aided Design* , 2003, Vol 35, No 4, pp 347–357.

[13] Gray P., Bedi S. and Ismail F. Arc-intersection method for 5-axis tool positioning, *Computer-Aided Design*, 2005, Vol 37, No 7, pp 663–674.

[14] Ho M.C., Hwang Y.R., and Hu C.H. Five-axis tool orientation smoothing using quaternion interpolation algorithm, *International Journal of Machine Tools and Manufacture*, 2003, Vol. 43, No. 12, pp 1256-1267.

[15] Ho S., Sarma S. and Adachi Y. Real-time interference analysis between a tool and an environment, *Computer-Aided Design*,2001, Vol 33, No 13, pp 935–947.

[16] Hosseinkhani Y., Akbari J. and Vafaeesefat A. Penetration–elimination method for five-axis CNC machining of sculptured surfaces, *International Journal of Machine Tools and Manufacture*, 2007, Vol 47, No 10, pp 1625–1635.

[17] Ilushin O., Elber G., Halperin D., Wein R. and Kim M.S. Precise global interference detection in multi-axis NC-machining, *Computer-Aided Design*, 2005, Vol 37, No 9, pp 909–920.

[18] Jensen C.G. Analysis & synthesis of multi-axis sculptured surface machining. D.Ph. Thesis, Purdue University, USA,1993.

[19] Jensen C.G., Red W.E. and Pi J. Tool selection for five-axis curvature matched machining. *Computer-Aided Design*, 2002, Vol 34, No 3, pp 251-266.

[20] Lauwers B., Dejonghe P. and Kruth J.P. Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation, *Computer-Aided Design*, 2003, Vol 35, No 5, pp 421–432.

[21] Lee Y.S. Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *Computer-Aided Design*, 1997, Vol 29, No 7, pp 507–521.

[22] Lee Y.S. and Chang T.C. 2-phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design*, 1995, Vol 27, No 10, pp 715–729

[23] Li L. L. and Zhang Y. F. Cutter selection for 5-axis milling based on surface decomposition, *2004 8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 6-9th December 2004, Vol. 3, pp 1863-1868

[24] Li L.L. and Zhang Y.F. Flat-end cutter accessibility determination in 5-axis milling of sculptured surfaces. *Computer-Aided Design & Applications*, 2005, Vol 2, No 1-4, pp 203-212.

[25] Li L.L. and Zhang Y.F. An integrated approach towards process planning for 5-axis milling of sculptured surfaces based on cutter accessibility map. *Computer-Aided Design & Applications*, 2006, Vol 3, No 1-4, pp 249-258

[26] Lin R.S. and Koren Y. Efficient tool-path planning for machining free-form surfaces. *ASME Journal of Engineering for Industry*, 1996, Vol 118, No 1, pp 20–28.

[27] Lo C.C. Efficient cutter-path planning for 5-axis surface machining with a flat-end cutter, *Computer-Aided Design*, 1999, Vol 31, No 9, pp 557–566

[28] Lo C.C. CNC machine tool surface interpolator for ball-end milling of free-form surfaces. *International Journal of Machine Tools And Manufacture*, 2000, Vol 40, No 3, pp 307–326.

[29] Lu G. In: Leung C.H.C. (Ed.) Visual information systems. Springer, Berlin, 1997, pp 135-150.

[30] Madhavulu G., Sender S.V.N.A., and Ahmed B. CAD/CAM solutions for efficient machining of turbomachinery components by Sturz milling method, *Proceedings of the 1996 IEEE IECON 22nd International Conference*, Vol. 3, pp.1490-1495.

[31] Makhanov S.S. and Anotaipaiboon W. Advanced numerical methods to optimize cutting operations of five-axis milling machines. Berlin: Springer-Verlag, 2007.

[32] McMahon C. and Browne J. CAD/CAM principles, practice and manufacturing management, Prentice Hall, Harlow, England, 1998.

[33] Misra D., Sundararajan V. and Wright P.K. Zig-Zag Tool Path Generation for Sculptured Surface Finishing. *DIMACS Workshop on Computer Aided Design and Manufacturing*, October 7 - 9, 2003.

[34] My C.A., Boheza E.L.J., Makhanov S.S., Munlin M., Phien H.N. and Mario T. On 5-axis freeform surface machining optimization: vector field clustering approach. *International Journal of CAD/CAM*, 2005, Vol. 5, No.1.

[35] Radzevich S..P. CAD/CAM of Sculptured surfaces on multi-axis NC machine: The DG/K-based approach. Morgan & Claypool, USA, 2008.

[36] Rao A. and Sarma R. On local gouging in five-axis sculptured surface machining using flat-end tools. *Computer-Aided Design*, 2000, Vol 32, No 7, pp 409–420

[37] Rao N., Ismail F. and Bedi S. Tool path planning for five-axis machining using the principle axis method, *International Journal of Machine Tools and Manufacture*, 1997, Vol 37, No 7, pp 1025–1040.

[38] Ritter G. X. and Wilson J. N. Handbook of computer vision algorithms in image algebra, CRC Press, 1996.

[39] Rogers D.F. An introduction to NURBS with historical perspective. Morgan Kaufmann Publishers, San Francisco, 2001.

[40] Roman A, Bedi S. and Ismail F. Three-half and half-axis patch-by-patch NC machining of sculptured surfaces. *International Journal of Advanced Manufacturing Technology*, 2006, Vol 29, No 5-6, pp 524-531.

[41] Roman A. Surface partitioning for 3+2-axis Machining. D.Phil. Thesis, University of Waterloo, Canada, 2007.

[42] Sonka M., Hlavac V. and Boyle R. Image processing, analysis, and machine vision, Thompson Learning, Toronto, Canada, 2008.

[43] Suresh K. and Yang D.C.H. Constant scallop-height machining of free-form surfaces. *ASME Journal of Engineering for Industry*, 1994, Vol 116, No 2, pp253–259.

[44] Tournier C. and Lartigue C. 5-axis Iso-scallop Tool Paths along Parallel Planes. *Computer-Aided Design and Applications*, 2008, Vol 5, No 1-4, pp 278-286.

[45] Tunc L. T. and Budak E. Extraction of 5-axis milling conditions from CAM data for process simulation, *International Journal of Advanced Manufacturing Technology*, 2008, DOI 10.1007/s00170-008-1735-7

[46] Tournier C. and Duc E. Iso-scallop tool path generation in 5-axis milling. *International Journal of Advanced Manufacturing Technology*, 2005, Vol 25, No 9-10, pp 867-875.

[47] Vickers G.W. and Quan K.W. Ball-mills versus end-mills for curved surface machining. *ASME Journal of Eng. for Industry*, 1999, Vol 111, pp 22–26.

[48] Wang J. Global finish curvature matched machining. M. S. Thesis, Brigham Young University, USA, 2005

[49] Wang Y.J., Zuomin D.Z. and Vickers G.W. A 3D curvature gouge detection and elimination method for 5-axis CNC milling of curved surfaces. *The International Journal of Advanced Manufacturing Technology*, 2007, Vol 33, pp 524-531.

[50] Warkentin A., Hoskins P., Ismail F., and Bedi S. Computer-aided 5-axis machining. In: C.T. Leondes, (Ed.), Computer-aided design, engineering, and manufacturing: Systems techniques and applications. London: CRC Press LLC, 119-152, 2001.

[51] Wang N. and Tang K. Five-axis tool path generation for a flat-end tool based on iso-conic partitioning. *Computer-Aided Design*, 2008, Vol. 40, pp 1067-1079.

[52] Parametric Technology Corporation, Pro/Engineer Wildfire help,.

[53] Planit Holdings Limited, EdgeCAM help.

[54] The Mathworks, Matlab product help.

PUBLICATIONS

[1] Tuong N.V. and Pokorny P. Modeling concave globoidal cam with swinging roller follower: a case study. *Proceeding of World Academy of Science, Engineering and Technology*, 2008, Vol. 32, pp 180-186.

[2] Tuong N.V. and Pokorny P. CAD techniques on modeling of globoidal cam. *Proceeding of 2nd International Conference, Production system - Today and Tomorrow*, TUL, Liberec, 2008.

[3] Tuong N.V. and Pokorny P. Modeling concave globoidal cam with Indexing turret follower: A case study. *International Journal of Computer Integrated Manufacturing*, 2009, Vol. 22, No. 10, pp 940-946.

[4] Tuong N.V. and Pokorny P. A study on making animation and checking interference of globoidal cam. *Modern Machinery (MM) Science Journal*, 2009.

[5] Tuong N.V. and Pokorny P. A case study of modeling concave globoidal cam (book chapter), in: "Advanced Technologies", In-Tech, Austria, 2009.

[6] Tuong N.V., Pokorny P. and Hieu L.C. Free-form surface partitioning for 5-axis CNC milling based on surface curvature and Chain Codes. *The international conference on Innovative Production Machines and Systems (IPROMS),* 2009.

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA STROJNÍ

Ing. Nguyen Van TUONG

REALIZACE OBECNÝCH PLOCH

(FREE-FORM SURFACE MANUFACTURING)

DOKTORSKÁ PRÁCE

2009

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA STROJNÍ KATEDRA VÝROBNÍCH SYSTÉMŮ

Obor: Konstrukce strojů a zařízení
Zaměření: Obráběcí a montážní stroje

REALIZACE OBECNÝCH PLOCH

(FREE-FORM SURFACE MANUFACTURING)

Ing. Nguyen Van Tuong

Školitel: Prof. Ing. Přemysl Pokorný, CSc.

Počet stran	: 102
Počet obrázků	: 54
Počet tabulek	: 11
Počet modelů	: 02
Počet příloh	: 01 CD-ROM

V Liberci 16. Září 2009

ABSTRAKT

Povrchy obecných tvarů mohou být obrobeny na 3 - nebo 5-osých NC strojích. Při výběru řezných nástrojů pro obrobení určité obecné plochy, je průměr nástroje omezen určitou hodnotou, která nesmí způsobit místní podřezání při obrábění konkávního a sedlového tvaru. Je zřejmé, že obrábění trvá dlouho, pokud se používá jen jeden nástroj k obrobení celého povrchu. Proto, pokud konkávní či sedlový tvar jsou obrobeny odděleně jedním nebo dvěma odpovídajícími nástroji a ostatní oblasti mohou být rychle obrobeny nástroji které jsou větší, může být čas obrábění značně snížen.

Tato práce se zaměřuje především na vývoj nových postupů obrábění obecných ploch. Odvozeně od dílčích zakřivení povrchu, obecný tvar může být rozdělen na oblasti konvexní, konkávní a sedlové a v těchto oblastech se uplatní technika kódování řetězců k určení ohraničení jednotlivých oblastí. Každou dílčí oblast lze pak obrobit nejvhodnějším nástrojem, který může umožnit velké snížení celkového času k obrobení povrchu. Pro tento účel byl vyvinut Matlab® program který provádí rozdělení povrchu a určí souřadnice všech bodů na hranicích oblastí. Rekonstrukce obecného povrchu a simulace drah nástroje v oddělených částech je uskutečněno pomocí Pro/Engineer® Wildfire®.

V disertační práci jsou některé aplikace prezentovány za účelem ukázat, že navržená metoda může snížit časy obrábění v porovnání s tradičními metodami, v nichž se celá plocha obrobí pouze jedním nástrojem. Některé testy jsou rovněž provedeny na multi-funkčním (víceosém) stroji k ověření a ohodnocení navržené metody. Z výsledků, lze dospět k závěru, že navržená metoda je úspěšná.

Klíčová slova:
Obecná plocha, členění povrchu, 5-osé obrábění, čas obrábění.

ABSTRACT

The free-form surfaces can be machined on 3- or 5-axis CNC machines. When selecting a cutting tool to machine a particular free-form surface, the tool diameter is restricted to an established value that must not cause local gouging when machining the concave and saddle regions.. Obviously, it would take long if only one tool is used to machine the entire surface. Therefore, if the concave and saddle regions are machined separately by one or two relevant tools; and the other regions are rapidly milled by the bigger ones, the machining time can be then significantly reduced.

This dissertation mainly focuses on developing a practical method to machine free-form surfaces. Based on the surface curvatures, a freeform surface is partitioned into convex, concave and saddle regions in which the chain code technique is applied to determine the boundaries of each region. Each partitioned region then can be milled by the most suitable tool that can lead a big reduction in the total machining time of the surface. For implementation purpose, a Matlab® program was developed to perform the surface partitioning task and obtain the coordinates of all points on the boundaries. The freeform surface reconstruction and simulation of the tool path of partitioned regions were illustrated via Pro/Engineer® Wildfire®.

In the dissertation, some application examples are presented to show that the proposed method can reduce the machining time, compared to the traditional method in which the whole surface was machined by only one cutter. Some experiments are also done on a multi-tasking machine to validate the proposed method. From the machining results, it can be concluded that the proposed method has been successfully implemented.

Keywords:
free-form surface, surface partitioning, 5-axis machining, machining time.

Prohlášení

Byl jsem seznámen s tím, že na mou doktorskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména §60 (školní dílo) a §35 (o nevýdělečném užití díla k vnitřní potřebě školy).

Beru na vědomí, že Technická univerzita v Liberci má právo na uzavření licenční smlouvy o užití mé práce a prohlašuji, že souhlasí m s případným užitím mé práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své doktorské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem Technické univerzity v Liberci, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

V Liberci 16, Září 2009 ……………………………..
vlastnoruční podpis

Místopřísežné prohlášení

Místopřísežně prohlašuji, že jsem doktorskou práci vypracoval samostatně s použitím uvedené literatury, pod vedením vedoucího práce.

V Liberci 16, Září 2009 ……………………………..
vlastnoruční podpis

ACKNOWLEDGEMENTS

First and foremost I would like to deeply thank my advisor Prof. Přemysl Pokorný for all his supports and guidance through the past three years. He had been spending much time on me throughout the research, from choosing the theme of the research till correcting carefully the dissertation.

I would like to thank all members of my graduate committee for their helpful advice and encouragement.

I would like to thank the Technical University of Liberec, especially the Department of Manufacturing System. I would like to thank M.Sc. Jiří Šafka and M.Sc. Hoang Sy Tuan for their time in helping me to do the reseach.

I would like to express appreciation to all of my friends who have helped me to overcome the difficulties in research and life for the last three years.

Finally, I would like to extend my thanks to my family, my wife and my little daughter, who continuously supported and encouraged me.

TABLE OF CONTENTS

List of figures viii

List of tables x

List of abbreviations xi

Nomenclature xii

Chapter 1 INTRODUCTION 1

1.1 Motivation 1

1.2 Research objective and research methodology 5

1.3 Organization of the Dissertation 6

1.4 Summary 7

Chapter 2 MATHEMATICAL REPRESENTATION OF FREE-FORM SURFACES AND SURFACE GEOMETRIES 8

2.1 Introduction of free-form surface representation 8

2.2 Mathematical representation of free form surfaces 8

2.3 Surface geometry 12

2.4 Summary 16

Chapter 3 SURFACE PARTITIONING AND SURFACE PATCH BOUNDARIES DEFINITION 17

3.1 Introduction of free-form surface partitioning 17

3.2 Related works 18

3.2 Surface partitioning 21

3.3 Surface patch boundaries definition 24

3.3.1 Requirement and solution 24

3.3.2 Chain codes 24

3.3.3 Defining patch boundaries with Matlab 26

3.4 Summary 28

Chapter 4 GOUGE DETECTION AND CORRECTION 29

4.1 Introduction of gouge detection and correction 29

4.2 Local gouging avoidance and definition of tool diameter 32

4. 3 Global gouging avoidance and definition of tool orientation 33

4.3.1 Overview 33

4. 3.2 Lead and tilt angles 34

4.3.3 Implementation of gouge avoidance 36

4.4 Summary 38
Chapter 5 TOOL PATH GENERATION 39
5.1 Introduction of tool path generation.......... 39
5.2 Tool path generation methods 39
5.3 Implementation of tool path generation 42
5.4 Summary 43
Chapter 6 APPLICATIONS 44
6.1 Matlab® functions and files.......... 44
6.1.1 The M-file 44
6.1.2 The M-function for calculating the geometric parameters of the surface 45
6.1.3 Other M-functions 46
6.2 Building CAD models of freeform surfaces 46
6.2.1 ibl files 46
6.2.2 Building CAD models of free-form surfaces in Pro/Engineer® Wildfire® 48
6.3 Applications and discussion 50
6.3.1 Overview 50
6.3.2 Application 1 52
6.3.3 Application 2 57
6.3.4 Application 3 64
6.4 Experiments 69
6.4.1 Experiment 1 69
6.4.2 Experiment 2 70
6.5 Summary 72
Chapter 7 CONCLUSIONS 73
7.1 Summary and conclusion 73
7.2 Contributions.......... 74
7.3 Future work 75
References 77
Publications 81
Appendix 82

LIST OF FIGURES

Figure 1.1 Some dies in industry 2

Figure 1.2 Typical cutters used for 5-axis surface machining 3

Figure 1.3 Effective radius of a flat-end cutter used for 5-axis surface machining 4

Figure 1.4 Flow chart of the research 6

Figure 1.5 Mazak Integrex 100-IV 7

Figure 2.1 A Bézier surface and its control polyhedron 9

Figure 2.2 A B-spline surface and its control points 11

Figure 2.3 Unit normal vector and tangent plane at a point 13

Figure 2.4 Curvature of a free-form surface 14

Figure 2.5 Gaussian curvatures and mean curvatures of a B-spline surface 15

Figure 3.1 A free-form surface and its regions 17

Figure 3.2 An example of the structure of matrices M_1 M_2 and M_3 23

Figure 3.3 Chain coding 25

Figure 3.4 An example chain code 25

Figure 3.5 The correlative images of matrices M_1, M_2 and M_3 27

Figure 4.1 Local gouging 29

Figure 4.2 Global gouging 29

Figure 4.3 An example of the information about detected gouges in Pro/Engineer 31

Figure 4.4 An example of global gouging 34

Figure 4.5 Coordinate systems and lead and tilt angles 35

Figure 4.6 Interference when machining deep concave region.................. 37

Figure 4.7 Tool axis always goes through (a) a curve and (b) a point 37

Figure 5.1 CC-Based tool path generation methods 40

Figure 5.2 Scallops left after machining 42

Figure 6.1 The structure of a curve in the ibl file format 47

Figure 6.2 An example of creating datum points for 3D curve 48

Figure 6.3 An example of trimming a region on a surface 48

Figure 6.4 Flow chart of the implementation 50

Figure 6.5 Illustration of a freeform surface in Matlab® (application 1) 52

Figure 6.6 Original surface with the boundary between two regions 54

Figure 6.7 Partitioned regions (application 1) 54

Figure 6.8 Example of tool paths for partitioned regions (application 1) 55

Figure 6.9 Machining simulation (application 1) 56

Figure 6.10 Illustration of a freeform surface in Matlab® (application 2) 58

Figure 6.11 Partitioned surface (application2) 59

Figure 6.12 Example of tool paths for each region (application 2) 61

Figure 6.13 Tool paths for the combined region 63

Figure 6.14 Machining simulation (application 2) 63

Figure 6.15 Illustration of a freeform surface in Matlab (application 3) 64

Figure 6.16 Partitioned surface (application 3) 65

Figure 6.17 Tool paths for each region (applicaticn 3) 67

Figure 6.18 Machining simulation (application 3) 68

Figure 6.19: Tool paths for the two combined regions 68

Figure 6.20 Machining sequences of the surface (test 1) 70

Figure 6.21 Machining sequences of the surface (test 2) 71

LIST OF TABLES

Table 1: Control net of a B-spline surface 52

Table 2: Machining parameters of the finishing cut for 1st application 54

Table 3: Machining time estimation (application 1) 57

Table 4: Control net of a B-spline surface for application 3 57

Table 5: Maximum curvatures K_{max} of regions and probable maximum radii R_{max} of the ball-end cutter used to mill regions 59

Table 6: Lead and tilt angles for each region 60

Table 7: Estimated machining time and tool path length comparison (application 2) 62

Table 8: Estimated machining time and tool path length comparison (recommended strategy, application 2) 62

Table 9: Control net of a B-spline surface for application 3 64

Table 10: Maximum curvatures K_{max} of regions and probable maximum radii R_{max} of the ball-end cutter used to mill regions 65

Table 11: Estimated machining time and tool path length comparison (application 3) 66

LIST OF ABBREVIATIONS

3D	Three-directional
CAD	Computer-aided design
CAM	Computer-aided manufacturing
CC	Cutter contact
ccw	Counter clockwise
CL	Cutter location
CNC	Computer numerical control
cw	Clockwise
dir	Direction
NC	Numerical control
NURBS	Non-Uniform Rotational Bsplines

NOMENCLATURE

B	Boundary function
Bi,n(u)	Bernstein basis function of degree n in the u
Bj,m(v)	Bernstein basis function of degrees m in the v parametric direction
c	Length of the chain code
C	Cross-feed
c{(0),c(1),…,c(n-1)}	Elements of chain codes
E, F, G	Fundamental magnitudes of the first order
f	Binary image
F	Feed
H	Mean curvature, H
$h_{i,j}$	Homogeneous coordinates of the control points
I_1, I_2, and I_3	Correlative images of matrices M_1, M_2 and M_3
K	Gaussian curvature
K_{min} and K_{max},	Principal curvatures
K^S_{max}	Maximum surface curvature of a region
L. M. N	Fundamental magnitudes of the second order
$N_{i,k}(u)$	B-spline basis functions in the bi-parametric u direction
n_S	Unit normal vector
n_x, n_y and n_x	Components of unit normal vector
M, M_1, M_2 and M_3	Matrices
$M_{j,l}(v)$	B-spline basis functions in the bi-parametric v direction
O	Tool tip
$p_{i,j}$	Control points
p	CC point
R	Radius of cutter
R_{eff}	Effective radius
R_k	Region on the surface
R_{max}	Maximum radius of cutters
Q	Plane containing the normal to the surface at a CC point

Su	Tangent vectors along u parametric direction
Sv	Tangent vectors along v parametric direction
S(u,v)	Free-form surface
t	Tangent vector to S(u,v)
T	Tool axis vector
Tp(S)	Tangent plane at p on surface S
u, v	Parametric directions
$u_{min,}$ umax	Limits of u
vmin, vmax	Limits of v
x_i , x_j	Elements of knot vectors
x, y and z	Coordinates of a point on a surface
α	Lead angle
β	Tilt angle
θ	Sturz angle
χ	Characteristic function
φ	Screw angle
Φ1	The first fundamental form,
Φ2	The second fundamental form

Chapter 1

INTRODUCTION

1.1 Motivation

Free-form surfaces, also sometimes called sculptured surfaces, are smooth regular surfaces, the major parameters of local topology of which in differential vicinity of any two infinitely closed to each other points differ from each other [35]. These surfaces are usually used for the following applications [3]:

(i) Die and mold design and manufacturing,

(ii) Automobile, ship, and aircraft body design, and

(iii) Commercial artwork

Some examples of dies in industry are shown in Figure 1.1.

Free-form surfaces are often produced in three stages: roughing, finishing, and bench-work [50]. During the rough machining, the material removal rate is high, results in high cutting forces and the surface quality is not so important. On the other hand, in the finish machining stage, the material removal rate is low and the tool is not subjected to high cutting forces. This stage also requires high surface quality. On the surface after finishing still remains scallops hence the bench-work consisting of grinding and polishing is required to remove these scallops. This manual stage is time-consuming and the manual polishing causes errors and undesirable irregularities. The time spent on the finishing and bench-work stages is dependent on the size of the scallops and the surface area. For small molds and dies, it is stated that over 78% of the total production time is spent on finishing, grinding, and polishing [50].

Traditionally, free-form surfaces are machined on three-axis numerical control (NC) machines using ball-end cutters. In this case, the tool axis is fixed and easy to position with respect to the part surface. In general, the overall productivity and material removal rate of finish surface machining with ball-end cutter is very low [48] and the process is inefficient [18, 47]. One of the reasons is that with a given diameter of ball-end mill and a scallop height in finish cut, the surface requires very high density of tool paths in concave regions that concave regions. In five-axis machining of free-form surfaces, the tools can be ball-end cutter, toroidal cutter or flat-end cutter and the tool orientation can be changed during machining. These tools are shown in Figure 1.2. In comparison to three-axis machining, five-axis machining offers some advantages in producing free-form surfaces. Five-axis machining reduces the machining time and improves the surface finish.

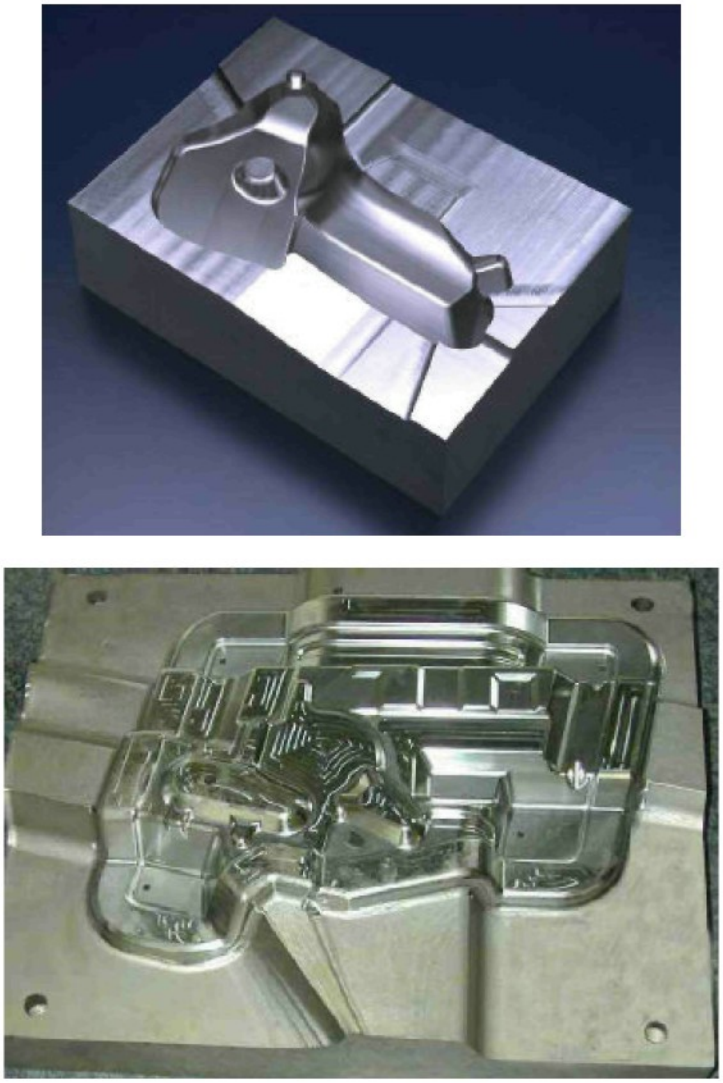


Figure 1.1: Some dies in industry [46].

When choosing a tool for a particular surface, the diameter tool is restricted to a determined value that must not cause local gouging when the tool is machining in concave and saddle regions. If a ball-end cutter is used, the cutter radius must be smaller than the minimum radius of curvature of the machined surface to avoid local gouging [6, 49]. Therefore, the maximum radius of the selected ball end mill can be calculated as follows

$$R_{\max} = \frac{1}{K_{\max}^{S}} \tag{1.1}$$

where, $K_{\max}^{S}$ is the maximum surface curvature of the region to be machined.

For toroidal and flat-end cutters, to avoid local gouging, the effective radius of the cutter must be smaller than the minimum radius of curvature of the surface [24, 25, 36]. The effective radius of the flat-end cutter, R_{eff}, can be compute as [51]

$$R_{eff} = \frac{R}{\sin\theta} \tag{1.2}$$

where R is the radius of the cutter and θ is the inclination angle between the unit normal and the tool axis, also called the Sturz angle [14, 30, 50], as shown in Figure 1.3. It can be seen that the value of the effective radius depends on the Sturz angle. On the other hand, a ball-end cutter only has a constant effective radius and it equals the radius of the cutter.

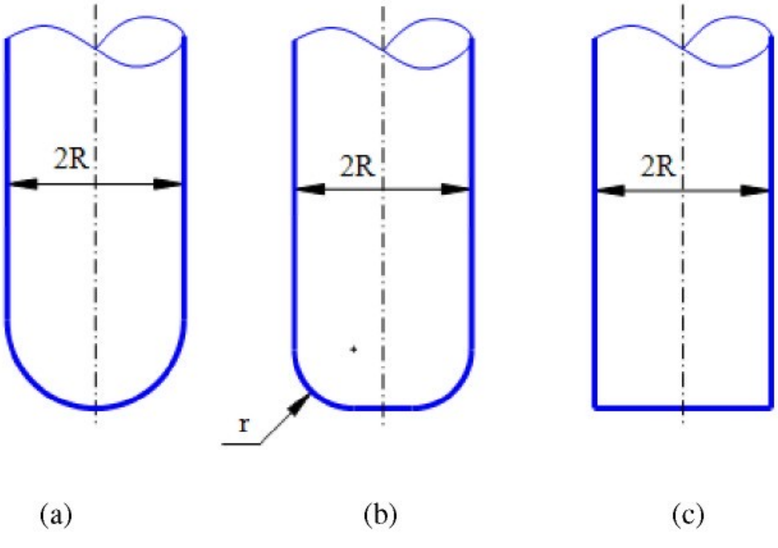


Figure 1.2: Typical cutters used for 5-axis surface machining:
(a) ball-end cutter; (b) toroidal cutter; (c) flat-end cutter

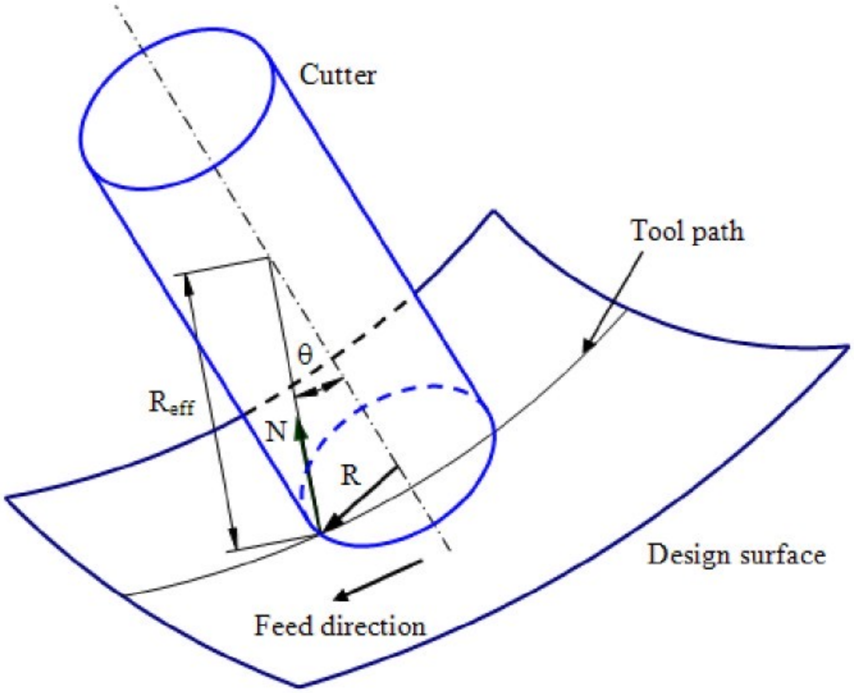


Figure 1.3: Effective radius of a flat-end cutter used for 5-axis surface machining.

In general, a free-form surface has regions such as convex, concave, plane and saddle [9, 35, 39]. Therefore, if the concave and saddle regions are machined separately by one or two suitable tools and the other regions are milled faster by a bigger tool, then the machining time may be reduced significantly. To do this, the design surface should be partitioned into regions. In CAD/CAM (computer-aided design/computer-aided manufacturing) packages such as Catia®, Pro/Engineer®, Unigraphics NX®,… users can not separately choose every region and its boundary in the CAM programming stage, so that the surface should be partitioned into patches in the modeling stage of CAD model of the free-form surface.

Another critical problem in the free-form surface machining is gouging. Local gouging can be avoided if the user chooses the right cutting tool as mentioned above. By using CAD/CAM systems, users can find out the collision (global gouging) when simulating the tool path. Some NC verification systems such as VeriCut®, NCSIMUL®, also offer the ability that visually identifies regions that collision occurs. In these systems, users must manually correct the tool path by adjusting the G codes in the NC verification systems or they have to regenerate the tool path in the CAM systems by restructuring some machining parameters. At the moment, this method is quite popularly used in practice.

Although this method is easy to implement, it is time-consuming in most cases when deep concave and saddle areas are involved. However, if the design surface is divided into regions then each region can be machined by a different tool path strategy. In this case, some special techniques can effectively be used to orient the tool in which gouge-free tool paths can successfully be generated for deep concave and saddle regions. In these techniques, the tool axis always goes through a curve or a point. Refer Section 4.3.3 to get some more information about these techniques.

1.2 Research objective and research methodology

The main objective of this research is to develop a practical machining method that can decrease the machining time and improve the surface quality by subdividing free-form surfaces into patches. In this study, a surface partitioning method that bases on the surface properties is proposed. The boundaries of each region are also defined for further uses. The tool position and orientation for collisions free are defined as well.

In order to achieve this objective, many tasks related to mathematical calculations and CAD/CAM should be implemented. The followings are some main tasks in this research.

a. Describe mathematically free-form surfaces and calculate the surface properties such as surface normal and curvatures.

b. Develop a surface partitioning method to the free-form surface into patches according to the surface properties.

c. Develop a method for determining the boundaries of regions on the design surface.

d. Create the CAD model of the design surface.

e. Generate gouge-free tool paths for the partitioned surface.

f. Compare the proposed methodology to the traditional methodologies in term of machining time by simulations.

g. Perform machining experiments.

The following chart presents the overview and the procedure of the research.

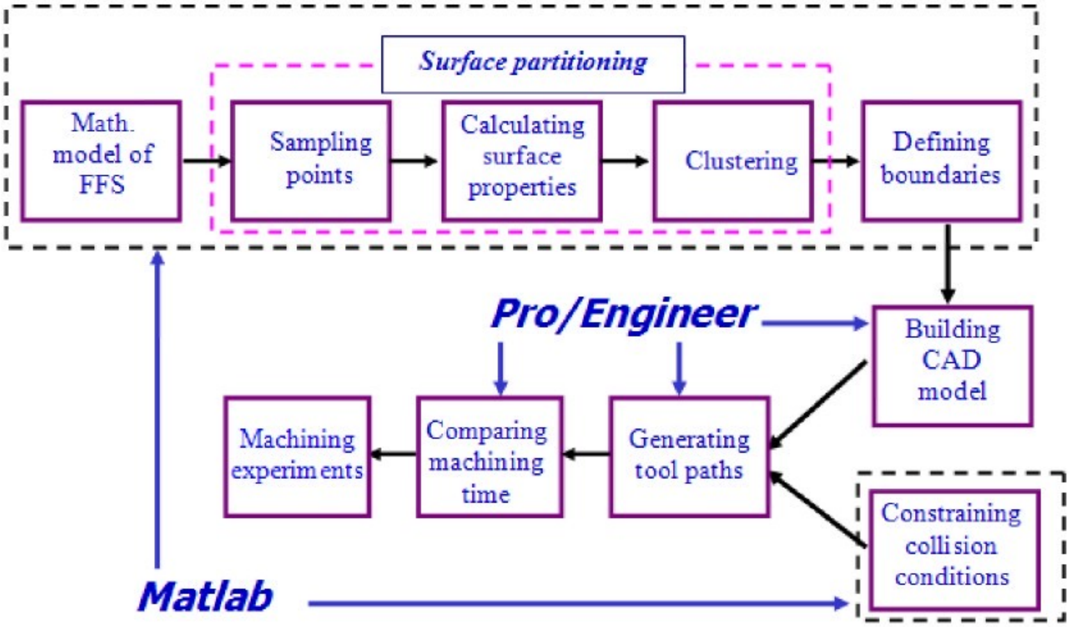


Fig 1.4: Flow chart of the research.

There are two powerful tools were used in the research. They are Matlab® and Pro/Engineer®. Matlab® is used for mathematical calculations and programming tasks, whereas Pro/Engineer® is used for CAD/CAM tasks. The machining experiments were performed on a multi-tasking machine Mazak® Intergrex 100-IV series as shown in Figure 1.5. This machine is available in the computer numerical control (CNC) laboratory of the Department of Manufacturing System, Technical University of Liberec.

1.3 Organization of the Dissertation

The dissertation is organized as follows: An introduction is presented in this chapter. In Chapter 2, the mathematical expression of free-form surfaces and the surface geometry are represented. The geometric parameters of a surface such as unit normal vector, curvatures will be used for subdividing a surface into convex, concave and saddle regions. The surface partitioning method is proposed in Chapter 3. A special technique based on chain codes for surface patch boundaries definition is also proposed in the last portion of this chapter. The method for gouge detection between tools and the design surface and the correction of tool postures are presented in Chapter 4. In Chapter 5, the method of tool path generation for each separate region of the surface is introduced. The implementation of surface partitioning and other related tasks are conducted in Chapter 6. Some machining experiment results are also presented in this chapter. Lastly, some conclusions are given in Chapter 7.

Figure 1.5: Mazak Integrex 100-IV.

1.4 Summary

In this chapter, an introduction of the research on free-form surface machining based on surface partitioning is presented. The motivation, the research objective and the research methodology are briefly introduced.

Some important tasks that should be carried out in the research are subdividing a free-form surface into convex, concave and saddle regions based on the surface properties, defining the boundaries of separate regions and developing the interference-free method for tool positioning when machining each region. Matlab® and Pro/Engineer® will be used as tools for the calculation and program and CAD/CAM purposes.

Chapter 2

MATHEMATICAL REPRESENTATION OF FREE-FORM SURFACES AND SURFACE GEOMETRIES

2.1 Introduction of free-form surface representation

Free-form surfaces can be designed directly by using CAD or CAD/CAM packages or obtained through surface fitting from a digitized data set. In the former case, designers have to move the control points on the screen to get the desired shape of the surface to be created. In the latter case, often used in reverse engineering, a scanner digitizes surfaces of an existing object to get a set of points and feed them into a mathematical model for surface fitting [3]. In this research a “forward engineering” as the former case is implemented. Here, the mathematical representations of free-form surfaces will be given as input data for further calculation and programming.

2.2 Mathematical representation of free form surfaces

Generally, as mentioned before, free-form surfaces may be designed directly on a computer using control points, in forms such as Bezier surface, B-spline surface, non-uniform rational B-spline surface (NURBS). These surfaces are the most popular forms that define free-form surfaces in CAD/CAM systems.

Bézier surface

Let *Bi,n*(*u*) and *Bj,m*(*v*) be the Bernstein basis functions of degrees *n* and *m* in the *u* and *v* parametric directions, respectively. A Bézier surface with control points $p_{i,j}$, $(0 \le i \le n, 0 \le j \le m)$ is the parametric surface defined by [39]

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{i,j} B_{i,n}(u) B_{j,m}(v) \tag{2.1}$$

where

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \tag{2.2}$$

$$B_{j,m}(v) = \binom{m}{j} v^j (1-v)^{m-j} \tag{2.3}$$

with

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{2.4}$$

$$\binom{m}{j} = \frac{m!}{j!(m-j)!} \tag{2.5}$$

The parameter curves of a Bézier surface are spatial Bézier curves. These curves form the four edges of the Bézier surface as illustrated in Figure 2.1. Bézier surfaces only pass through the corner points of the characteristic polygon and at these points their edge curves are tangential to the edges of the characteristic polygon. These surfaces have a number of limitations. They only allow global modification, and they are somewhat constraining if smooth transition between adjacent patches is to be achieved [32].

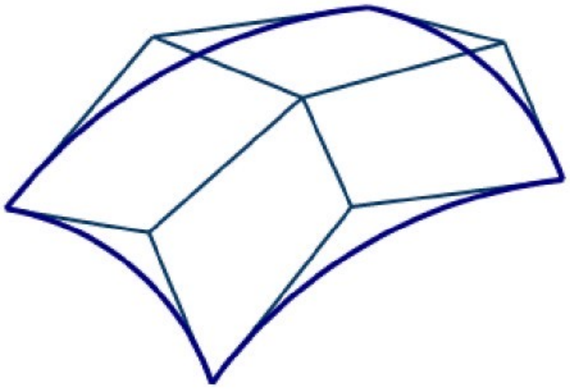


Figure 2.1: A Bézier surface and its control polyhedron.

B-spline Surfaces

A B-spline surface with control points is defined by [39]

$$S(u,v)=\sum_{i=1}^{n+1}\sum_{j=1}^{m+1}B_{i,j}N_{i,k}(u)M_{j,l}(v) \tag{2.6}$$

$u_{min} \leq u \leq u_{max}$, $v_{min} \leq v \leq v_{max}$

where $\mathrm{N}_{\mathrm{i,k}}(u)$ and $\mathrm{M}_{\mathrm{j,l}}(v)$ are the B-spline basis functions in the bi-parametric u and v directions, respectively; $B_{i,j}$ are the vertices of a polygonal control net.

$$N_{i,1}(u)=\begin{cases}1 & \text{if } \mathrm{x}_{\mathrm{i}} \leq u \leq x_{i+1} \\ 0 & \text{otherwise}\end{cases} \tag{2.7}$$

$$N_{i,k}(u)=\frac{(u-x_i)N_{i,k-1}(u)}{x_{i+k-1}-x_i}+\frac{(x_{i+k}-u)N_{i+1,k-1}(u)}{x_{i+k}-x_{i+1}} \tag{2.8}$$

$$M_{j,1}(v)=\begin{cases}1 & \text{if } \mathrm{y}_{\mathrm{j}} \leq v \leq y_{j+1} \\ 0 & \text{otherwise}\end{cases} \tag{2.9}$$

$$M_{j,l}(v)=\frac{(v-y_j)N_{j,l-1}(v)}{y_{j+l-1}-y_i}+\frac{(y_{j+l}-v)N_{j+1,l-1}(v)}{y_{j+l}-y_{j+1}} \tag{2.10}$$

where the x_{i} and y_j are elements of knot vectors.

B-spline surfaces overcome the limitations of Bézier surfaces. These surfaces approximate a characteristic polygon, as shown in Figure 2.2, and pass through the corner points of the polygon. At these points, the edges of the surface are tangential to the edge of the polygon. A control point of the surface influences the surface only over a limited portion of the parametric space of variables u and v [32].

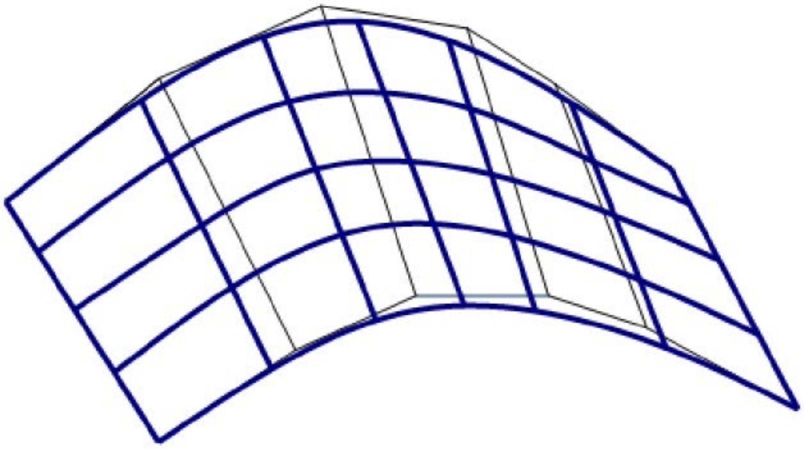


Figure 2.2: A B-spline surface and its control points.

NURBS

A *NURBS surface* with control points is defined by [39]

$$S(u,v) = \frac{\sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} p_{i,j} N_{i,k}(u) M_{j,l}(v)}{\sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,l}(v)} \tag{2.11}$$

where the $p_{i,j}$ are the three-dimensional coordinates and $h_{i,j}$ are the homogeneous coordinates of the control points; $N_{i,k}(u)$ and $M_{j,l}(v)$ are the non-rational B-spline basis functions. Note that the same knot values and parameter ranges as those in Equation 2.6 are used.

The NURBS surface equation is a general form that includes the B-spline surface equation and it has the advantage of exactly representing the quadric cylindrical, conical, spherical, paraboloidal, and hyperboloidal surfaces [32].

In practice, the free-form surface of a sculptured part may consist of many composite surfaces, and each surface can be represented in any form [4]. Here, to illustrate the approach and to make calculations simpler, we assume that the design surface is a single surface and some B-spline surfaces are used as examples in this study.

2.3 Surface geometry

Surfaces can be represented in implicit, explicit, or parametric form. The implicit form for a surface is the set of points (x, y, z) that satisfy the equation as follows

$$f(x, y, z) = 0 \qquad (2.12)$$

An explicit equation of a surface is presented as

$$z = f(x, y) \qquad (2.13)$$

Equations (2.12) and (2.13) illustrate that the points on a surface have two degrees of freedom which are directly controlled by the *x* and *y* coordinates and they are also called the non-parametric forms of a surface. While, the parametric form [35] for a surface in space is

$$S(u,v) = \begin{pmatrix} S_x(u,v) \\ S_y(u,v) \\ S_z(u,v) \end{pmatrix} \qquad (2.14)$$

The followings are some geometric parameters of a surface.

- Unit normal vector at a point

Given a free-form surface *S(u, v)* and p is a regular point on this surface. At this point, S*u* and S*v* are the tangent vectors along *u* and *v* parametric directions. Hence S*u* and S*v* are non-parallel vectors, and a vector perpendicular to them both is the *unit normal vector* to the surface, given by [35]

$$n_S(u,v) = \frac{S_u \times S_v}{|S_u \times S_v|} \qquad (2.15)$$

as shown in Figure 2.3. Any vector **t** perpendicular to n_S is called a *tangent vector* to *S(u,v)* at point *p*. The plane that contains all the tangent vectors to surface *S* at point *p* is called the *tangent plane* at point *p*, and denoted Tp(*S*), as shown in Figure 2.3.

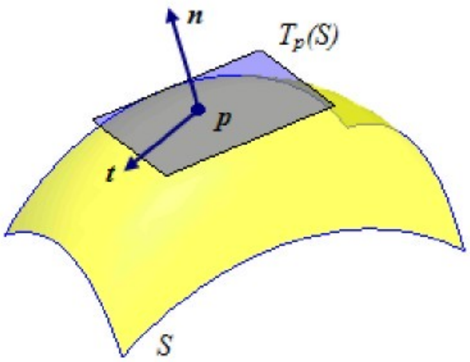


Figure 2.3: Unit normal vector and tangent plane at a point.

- The first fundamental form,Φ1:

The first fundamental form of a smooth regular surface describes the metric properties of a free-form surface *S,* is defined by [35]

$$\Phi 1= \mathrm{d}S.\mathrm{d}S = E\mathrm{d}u^2 + 2F\mathrm{d}u\mathrm{d}v + G\mathrm{d}v^2 \quad (2.16)$$

where

$$E = \frac{\partial S}{\partial u}\frac{\partial S}{\partial u} \; ; F=\frac{\partial S}{\partial u}\frac{\partial S}{\partial v} \; ; G=\frac{\partial S}{\partial v}\frac{\partial S}{\partial v} \quad (2.17)$$

are the fundamental magnitudes of the first order.

- The second fundamental form,Φ2

The second fundamental form describes the curvature of a free-form surface, is defined by [35]

$$\Phi 2= -\mathrm{d}n_S .\mathrm{d}S = L\mathrm{d}u^2 + 2M\mathrm{d}u\mathrm{d}v + N\mathrm{d}v^2 \quad (2.18)$$

where

$$L = n_S \frac{\partial^2 S}{\partial u^2} \; ; M=n_S \frac{\partial S^2}{\partial u \partial v} \; ; \; N=n_S \frac{\partial^2 S}{\partial v^2} \quad (2.19)$$

are the fundamental magnitudes of the second order.

- Gaussian curvature, K, and mean curvature, H

Given a free-form surface S(*u,v*), p is an arbitrary point on it. The curve of intersection of a plane (Q) containing the normal to the surface at point *p* and the surface has a curvature, as shown in Figure 2.4. As the plane (Q) rotates about the normal, the curvature changes. Euler showed that unique directions for which the curvature is a minimum and a maximum exist. The curvatures in these directions are called the principal curvatures, K_{min} and K_{max}, and the principal curvature directions are orthogonal.

The Gaussian curvature, K, the mean curvature, H, and the principal curvatures, K_{min} and K_{max}, of the surface S(*u, v*), at a point p, is formulated as [35]

$$K = \frac{LN - M^2}{EG - F^2} = K_{\min} K_{\max} \tag{2.20}$$

$$H = \frac{1}{2}\left(\frac{EN - 2FM + GL}{EG - F^2}\right) = \frac{1}{2}(K_{\min} + K_{\max}) \tag{2.21}$$

$$K_{\max} = H + \sqrt{H^2 - K} \tag{2.22}$$

$$K_{\min} = H - \sqrt{H^2 - K} \tag{2.23}$$

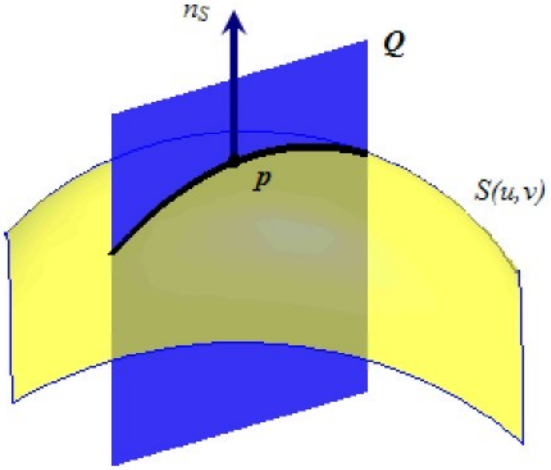
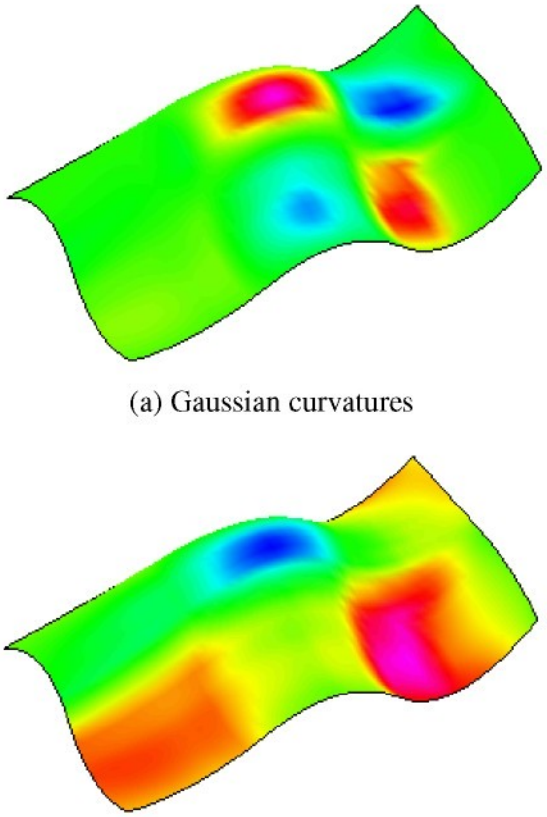


Figure 2.4: Curvature of a bi-parametric surface.

There are a number of techniques that can be used to present the Gaussian and mean curvatures of a surface. Depending on the display capabilities, the curvatures can be displayed in forms of contour plots, colors or gray scales [39]. In Figure 2.5 are the Gaussian curvatures and mean curvatures of a B-spline surface created in Pro/Engineer® Wildfire® and they are in form of shaded colour.



(a) Gaussian curvatures

(b) Mean curvatures

Figure 2.5: Gaussian curvatures and mean curvatures of a B-spline surface.

Based on the values of Gaussian, mean, and principal curvatures, the surface points can be divided into six different types as follows [31]:

- *Concave elliptic point*: If $K > 0$ and $H > 0$, the surface lies entirely on the surface normal side of the tangent plane in its neighbourhood; both K_{max} and K_{min} are positive.

- *Convex elliptic point*: If $K > 0$ and $H < 0$, the surface lies entirely on the opposite side of the tangent plane in its neighbourhood; both K_{max} and K_{min} are negative.

- *Hyperbolic point*: If K <0, the local surface lies entirely on both sides of the tangent plane in its neighbourhood; K_{max} and K_{min} have different signs.

- *Concave parabolic point*: If $K = 0$ and $H > 0$, the surface lies entirely on the surface normal side of the tangent plane in its neighbourhood. One of the principal curvatures is zero and the other is positive.

- *Convex parabolic point*: If $K = 0$ and $H < 0$, the surface lies entirely on the opposite side of the tangent plane in its neighbourhood. One of the principal curvatures is zero and the other is negative.

- *Planar umbilic point:* If $K = 0$ and $H = 0$, the surface lies entirely in the tangent plane in the neighbourhood of the umbilic point.

In order to trichotomize free-form surfaces into convex (included plane), concave and saddle regions, the local surface shape around a point can also be divided simply into three different types as follows:

a) $K \geq 0$ and $H \leq 0$: local surface shape is convex

b) $K \geq 0$ and $H > 0$: local surface shape is concave

c) $K < 0$ and $H \neq 0$: local surface shape is saddle.

While Gaussian curvature and mean curvature are used for surface partitioning, unit normal vectors and principal curvatures are utilized in the task of checking interference between the tool and the design surface and correcting the tool postures (see Chapter 4).

2.4 Summary

In this chapter, three mathematical types of Bézier, B-spline and NURBS surfaces are presented as free-form surfaces. Some B-spline surfaces are used as examples in the research for surface partitioning. Some important surface properties such as unit normal vector and curvatures are also introduced in this chapter. Once having partitioned a surface into separate regions of convex, concave and saddle, each region can be machined solely by different tool. The method for surface partitioning and its implementation is proposed in the next chapter.

Chapter 3

SURFACE PARTITIONING AND SURFACE PATCH BOUNDARIES DEFINITION

3.1 Introduction of free-form surface partitioning

In general, a free-form surface often has three types of regions such as convex, concave, and saddle. These regions can be divided separately from the design surface based on surface curvatures as introduced in chapter 2. Figure 3.1 shows a free-form surface with its regions. In this chapter, after presenting some related works, we propose the methods for surface partitioning based on Gaussian and mean curvatures and defining the boundaries of each sole region thanks to the chain codes technique in the image processing filed.

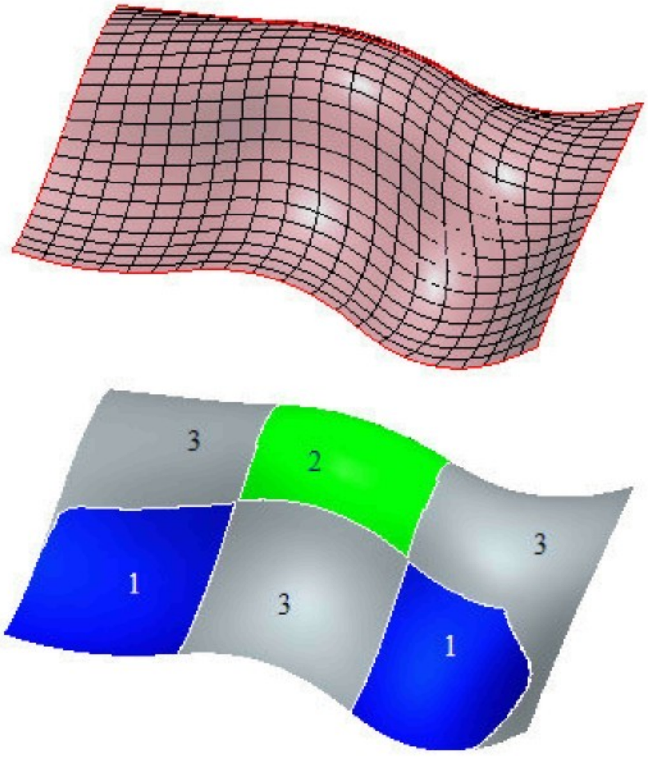


Figure 3.1: A free-form surface and its regions:
1- concave region, 2- convex region, 3-saddle region.

3.2 Related works

For the purpose of improving the productivity and the surface finish when machining free-form surfaces, many surface partitioning methods have been implemented. The following are some typical researches in this field.

In the study of Chen et al. [4], the geometric properties of free-form surfaces such as Gaussian and mean curvatures, maximum and minimum curvatures and the surface normal are used to form a multi-dimensional vector for surface partitioning. First, in the rough surface subdivision, based on the surface curvatures, the grid points are roughly grouped into three regions namely convex, concave, and saddle. Then, the subtractive fuzzy clustering method and the fuzzy C-means method are used to perform the fine surface subdivision. The former is used for estimating the number and centers of the surface patches and the latter is applied to optimize the locations of cluster centers for the grid points of the patch. The patch boundaries are then defined by using the Voronoi diagram. This method can divide a free-form surface into simple patches so that each patch can be properly accessed in a part setup and efficiently machined on a 3-axis CNC machine equipped a tilt-rotary table (3+2 axis CNC machining).

Roman and his colleagues [40] also used fuzzy C-means method to partition free-form surface for 3+2 axis CNC machining. In their study, the geometric properties of the surface are used as surface partitioning parameters and the patch boundaries are delimited by using the nearest neighbour method in the u–v plane. Roman [41] also utilized the k-means clustering method to subdivide a surface into patches and the Minimum Intra-Class Distance method was implemented for the identification of boundaries of these patches.

The concept of "isophotes" and "light region" were applied for surface partitioning by Ding et al [7]. In this study, isophotes are defined as points on a surface that have the same light intensity from a given source. The angle between the surface normal at a point and a given reference direction is called isophote angle. A light region is a region in which the isophote angles are smaller than a prescribed value. The boundary of this region is an isoline that consists of isophotes with the same isophote angle. By applying isophotes, a free-form surface is partitioned into different regions according to the slope made with the normal direction of parallel intersecting planes.

Bey et al. [1] proposed a methodology for finishing free form surfaces on 3-axes

CNC milling machines by considering the local shapes of the surfaces, the choice of machining strategies and the optimal cutter. There are four stages in the proposed methodology. Firstly, the design surface (in NURBS form) is approximated into triangles. Secondly, the geometric properties, such as normal vector and curvatures, are calculated to identify the local shape of every vertex. The vertices are then grouped into distinct regions of convex, concave or saddle. Thirdly, the interference detection, correction and determination of the optimal cutter for each region are done. Lastly, the machining strategy for each region is chosen and the tool paths are generated as well. Among the four stages, the surface portioning stage plays a vital role. The research used C++ and Open GL for implementation purpose.

In order to maximize the machining efficiency by using the largest feasible cutter (ball-nose, filleted, or end-mill), Li and Zhang [23] also decomposed free-form surfaces into convex, concave and saddle regions based on the surface curvatures. They called the concave and saddle regions as the critical regions where local gouging was likely to happen. These regions are subjects for local gouging analysis in cutter selection. A NURBS patch was represented as a free-form surface for the implementation purpose. The algorithm for surface subdivision and for local gouging checking was implemented in C++ and OpenGL environment. The output of their work is the optimal cutter under the local gouging-free condition, and this tool is used to machine the entire surface. The following is the algorithm for surface subdivision in for surface subdivision in their work [23].

a) Sample the design surface in *(u, v)* frame into a set of grid points

b) Calculate the Gaussian curvature and mean curvature at each point

c) Find the points with the concave and saddle local surface geometry

d) Group the neighbouring concave and saddle points into concave clusters and saddle clusters, respectively. For each cluster, link the boundary points to form a closed region.

Li and Zhang also used the algorithm above for checking whether a given flat-end cutter was able to mill the entire surface without gouging in 5-axis machining [24]. However, in their works, the free-form surfaces were just decomposed into regions in the calculating stage to determine the optimal tool size and the feasible orientations of the tool. The CAD models of the surfaces were not partitioned into convex, concave and saddle regions and there was only one cutter used to machine the whole surface.

By analyzing the curvatures of a free-form surface, Elber and Cohen [9] developed another method for surface partitioning. In their work, the second order surface analysis is used to understand the curvature characteristics and the shape of the surface. They developed a hybrid method using symbolic and numeric operators to compute curvature properties and then to create trimmed regions which are solely convex, concave, and saddle. The same method was used to determine the bounds of the regions. Bézier and NURBS surface representations were used and all surfaces were created using the Alpha_1 solid modeller which was developed at the University of Utah. According to the authors, the curvature estimation techniques are local since they make use of local surface information only. Local information is inferior to global information in complex settings. Hence, they used symbolic techniques to help make decisions for surface portioning based on the entire aspect of a surface. They had utilized scalar and vector fields, whose definitions are derived from different attributes of the design surface, to analyse the design surface. The zero set of the second fundamental form computed symbolically as a scalar field is used to partition the design surface into a trichotomy of convex, concave and saddle regions.

From the results of Elber and Cohen's work, there is no doubt that they have implemented successfully their method for surface partitioning. However, their hybrid approach using both symbolic and numeric operations for computing curvature properties is rather complicated. They used the curve on which Gaussian curvatures $K(u,v)$ equal zero to divide free-form surfaces. It means that the curve on which Gaussian curvatures are zero should be defined. Yet, Gaussian curvatures $K(u,v)$ is a high order expression of u and v even for low order NURBS surface patches [23]. Hence, it is very difficult to solve analytically the equation above. From their paper, it can be seen that high-level math skills are needed to solve the problems of their approach.

In conclusion, from the literature it can be stated that the approaches using surface curvatures have been adopted by many researches for surface partitioning. Many methods have been proposed and successfully implemented. However, many of them are rather complicated and not effective. In our study, a simple but effective method is proposed. Here, based on the surface curvatures, free-form surfaces are also partitioned into convex, concave, and saddle regions. The boundaries of each patch are defined by using Chain

Codes method in the image processing field. The output can be used directly for the CAD/CAM tasks which can be carry out by commercial CAD/CAM packages in industries.

3.2 Surface partitioning

The objective of surface partitioning is to divide a free-form surface into a number of regions. In each region the points have the same characteristics. These regions, also called patches, can be machined separately with different types of cutter and strategies. In this study, a free-form surface is partitioned into convex, concave, and saddle regions, based on the Gaussian curvature. For implementation, a numerical method should be applied.

The following are main steps for surface partitioning.

(1) Presenting the mathematical model of the free-form surface,

(2) Sampling the surface in *u* and *v* directions to get a set of grid points,

(3) Calculating Gaussian curvature and mean curvature at each grid point,

(4) Grouping the neighboring concave and saddle points to form concave regions and saddle regions, respectively. The last portion of points forms convex regions (including plane regions).

The specific algorithm for surface partitioning is as follows:

Algorithm1: partitioning free-form surfaces

Input

A free-form surface S(*u,v*) {*mathematical model*}

Output

Sets of points belonging to three type of regions {*convex+plane, concave and saddle*}

Begin

(1) Sample the input surface to get a set of grid points {**p**} and store all points in matrix M

(2) Calculate parameters K and H at every point $\mathbf{p}_{i,j}$

```
(3) FOR each grid point $\mathbf{p}_{i,j}$ of the set point {$\mathbf{p}$}
      IF $K \geq 0$ and $H \leq 0$ THEN save that point in matrix $M_1$ {data matrix contains points on convex and plane regions}
      END IF
       IF $K \geq 0$ and $H > 0$ THEN save that point in matrix $M_2$ {data matrix contains points on concave regions}
      IF $K < 0$ THEN save that point in matrix $M_3$ {data matrix contains points on saddle regions}
      END IF
  END LOOP
End
```

When saving points of each region into a matrix, the following method is used. For each group of points, after grouping points on the surface, the points are encoded into an integer and saved with their indices in a matrix. The points which do not belong to that group are then encoded as zeros. The encoding procedure of the points for each region R_k can be expressed by the characteristic function as follows

$$\chi(x,y,z,k) = \begin{cases} 1\ (2 \text{ or } 3) & \text{if } (x,y,z) \in R_k \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where k denotes the region of convex, concave or saddle.

In algorithm1, matrices M_1, M_2 and M_3 have the same size as that of the matrix M. They contain the identification numbers of points which belong to convex, concave and saddle regions, respectively. In these matrices, the points on the surface of the same region are presented by a number, say 1 or 2 or 3, and the other points of other regions are presented by zero. Each character in the matrix owns its indices that represent the location of the corresponding point on the surface in *u* and *v* directions. Hence, the coordinates of points on the surface are maintained. These matrices are very useful for further calculations. Figure 3.2 below shows the structures of matrices M_1, M_2 and M_3. In this study, the above algorithm was programmed with MATLAB 7.6.0.

0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	0	0	1	1	0	0
1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

(a) Matrix M_1

2	2	2	2	2	2	2	0	0
0	0	2	2	2	2	0	0	0
0	0	0	2	2	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	2	2	0	0	0	0
0	0	2	2	2	2	0	0	0
2	2	2	2	2	2	2	0	0

(b) Matrix M_2

0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	3	3	3
0	0	0	0	0	0	3	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	3	3

(c) Matrix M_3

Figure 3.2: An example of the structure of matrices M_1 M_2 and M_3.

3.3 Surface patch boundaries definition

3.3.1 Requirement and solution

In the surface partitioning process, a surface is divided into patches. However, the boundaries of these patches have not yet been defined. The boundaries of a patch should be defined so that they can be used as the factors for trimming off that patch from the entire surface when modeling that surface in CAD/CAM packages. One solution that can be used to define the patch boundaries is to connect the grid points on the border of each region after partitioning. The boundaries of these patches can be represented by applying the chain codes method from the image processing field.

3.3.2 Chain codes

Chain codes are a notation for recording the list of edge points along a contour. They are often used to represent a boundary of an object, or other one-pixel-wide lines in images [42]. The object border is defined by a connected sequence of straight-line segments of specified length and direction. There are a number of advantages of chain code representation. First of all, it is efficient and compact. Secondly, the chain codes are translation invariant. Lastly, the chain codes can be used to compute many shape features, such as the length of the boundary, the area and the perimeter [29, 38].

In the Freeman chain code technique, the boundary representation is based on 4-connectivity or 8-connectivity of segments. The direction of each segment is coded by using a numbering scheme illustrated in Figure 3.3. The principle for finding the boundary of an object using the chain codes is that each boundary pixel of an object has an adjacent boundary pixel neighbour whose direction from the given boundary pixel can be specified by a unique number between 0 and 3 (4-connectivity) or 0 and 7 (8-connectivity). When tracing the object, for any pixel, its eight neighboring pixels should be considered in the case of 8-connectivity. This is done with the same orientation throughout the entire image [42]. The same procedure is applied for the case of 4-connectivity. The 8-directional chain code is more accurate to describe a boundary than the 4-directional chain code, and the code length may be smaller as well [29].

The algorithms for border detection of a region in an image using chain codes are given in References [38, 42]. The chain code representation for the boundary of a binary image is a sequence of integers as follows [38]

$$\mathbf{c} = \{\mathbf{c}(0), \mathbf{c}(1), \ldots, \mathbf{c}(n-1)\} \tag{3.2}$$

from the set {0, 1, …, 7} for 8-connectivity or {0,1,2,3} for 4-connectivity.

In expression (3.2), **c** is called the length of the chain code and this is the number of elements in the sequence. The first element in the sequence, **c**(0) is called the *initial point,* the last element, **c**(n - 1), is called the *terminal point* of the code. Figure 3.4 shows an example of a chain code where 8-neighborhoods are used. In this figure, the reference pixel starting the chain is marked by an arrow.

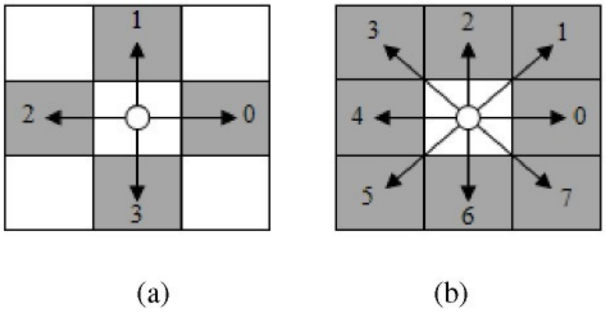


Figure 3.3: Chain coding: (a) 4-connectivity, (b) 8- connectivity.

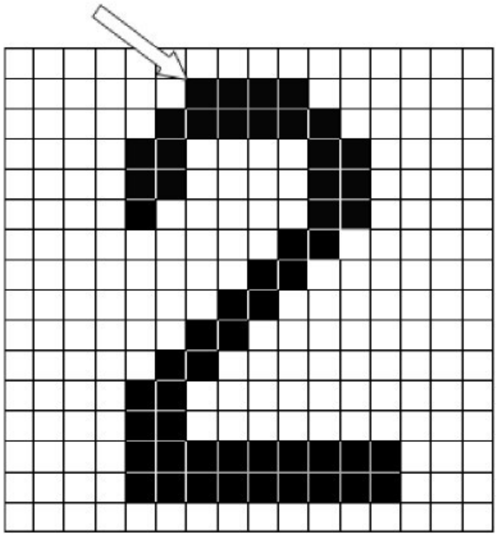


Figure 3.4: An example chain code [42]:

00077665555556600000006444444442221111112234445652211.

As can be seen from Figure 3.4, the boundary of the object is represented as a sequence of coded numbers. Each number can store information about the position of the correlative pixel. Therefore, if a free-form surface is considered as a binary image and regions, such as convex, concave or saddle, are considered as objects of the image, then the Freeman chain code technique can be utilized for determining the boundaries of regions of the surface. The implementation for defining the boundaries of regions of the surface is presented in the next subsection.

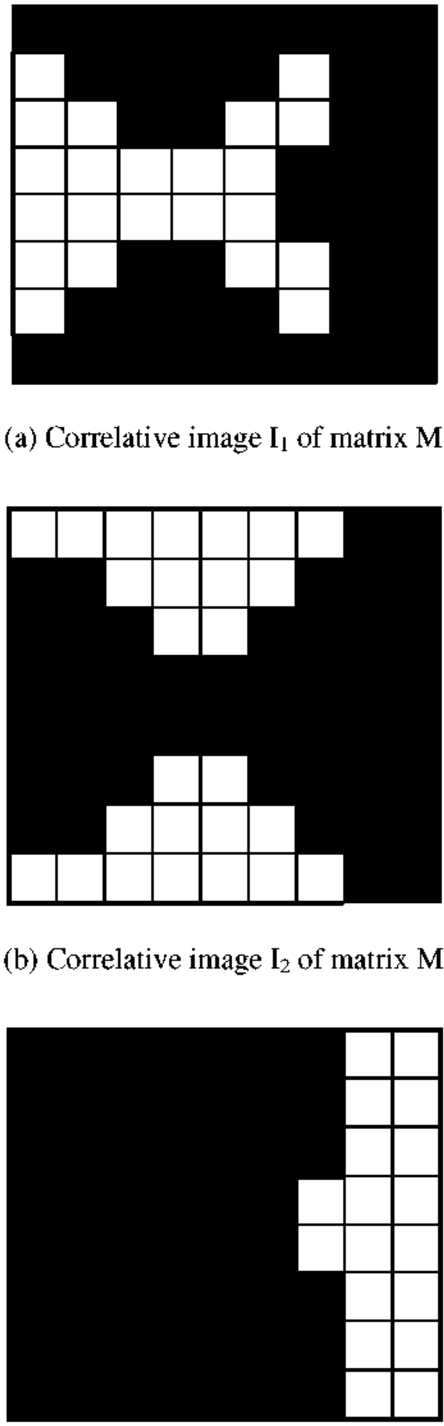
3.3.3 Defining patch boundaries with Matlab®

In Matlab, the Image Processing Toolbox is a collection of M-files that extend the capability of the Matlab environment for the solution of digital image processing. An M-function file named "boundaries" in Matlab language for finding the boundaries of objects in a binary image based on the Freeman chain code is presented in Reference [11]. This function traces the exterior boundaries of the objects in a binary image with pixels 0 as the background. The syntax of this function file is as follows

$$B = boundaries(f,\ conn,\ dir) \tag{3.3}$$

where *f* is the binary image, *conn* specifies the desired connectivity of the output boundaries and its values are 4 or 8 (the default), *dir* can be *cw* (the default) or *ccw* that specifies the direction in which the boundaries are traced clockwise or counter clockwise direction, and output B is a cell array that contains the coordinates of the boundaries found.

As mentioned in Section 3.2, each data matrix after partitioning (M_1, M_2 and M_3) contains only two values of number, 0 and 1 or 2 or 3 depending on regions of convex, concave or saddle, respectively. The structure of these matrices is similar to the structure of the matrix of pixels of a binary image which is a digital image that has only two possible values for each pixel. In the correlative images of these matrices, the background is 0s and the foreground is 1s or 2s or 3s. The correlative "images", say I_1, I_2, and I_3, of matrices M_1, M_2 and M_3 are shown in Figure 3.5.



(a) Correlative image I_1 of matrix M_1

(b) Correlative image I_2 of matrix M_2

(c) Correlative image I_3 of matrix M_3

Figure 3.5: The correlative images of matrices M_1, M_2 and M_3.

Because matrices M_1, M_2 and M_3 can be considered as the matrices of binary images, the binary image *f* in Function (3.3) can be replaced by matrices M_1, M_2 and M_3.

The result is that the coordinates of all "pixels" on the boundaries of the objects in "images" $I_{1,}$ I_{2}, and I_3 can be defined. These coordinates are not the coordinates of the points on the design surface but they own their indices that are equivalent to those of the points on the surface in u and v direction. Hence, some further calculations are required to get the coordinates of points on the boundaries on the three-directional (3D) surface.

In this study, in order to get a high accuracy of the boundary definition, the default values of *conn* and *dir* in Function (3.3) are used. Thus, the syntax of Function (3.3) is now simpler as

$$B = boundaries(f) \tag{3.4}$$

Once having the coordinates, the boundary curves can be easily created along with the CAD model of the free-form surface on many CAD/CAM systems. In this study, Pro/Engineer® Wildfire® is used to create the CAD model of the free-form surface and the boundaries of the concave and saddle regions as well.

3.4 Summary

In this chapter, some typical works related to surface partitioning is presented. Based on surface curvatures, the algorithm for partitioning free-form surfaces is proposed. To implement this algorithm, a special technique in which the points on the surface are stored in matrices is applied. The structure of these matrices is similar to that of the binary image.

Some basic information of chain codes is also introduces. Here, chain code technique is used for determining the boundaries of the convex, concave and saddle regions on the surface. A Matlab® function file named *boundaries* will be used for the purpose of implementation.

Chapter 4

GOUGE DETECTION AND CORRECTION

4.1 Introduction of gouge detection and correction

Local and global gouging is one of the most critical problems in free-form surface machining. Local gouging refers to the removal of an excess material in the vicinity of the cutter contact (CC) point as the tool move along the tool path. At every CC point, local gouging occurs when the radius of the local surface curvature is smaller than that of the cutter as depicted in Figure 4.1. Global gouging is defined as the interference between the tool body and the surface that need to be machined as shown in Figure 4.2.

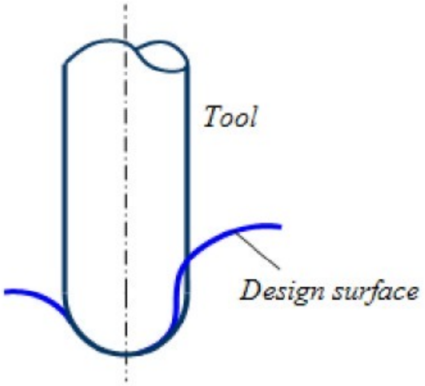


Figure 4.1: An example of local gouging

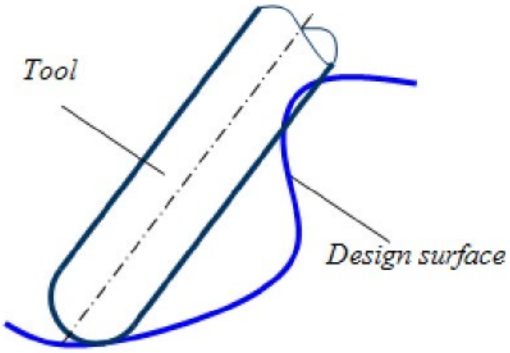


Figure 4.2: An example of global gouging

There are a number of works that related to gouge detection and correction done by many researchers. The following are some typical researches.

By comparing the effective curvature of the swept surface of the tool with the normal curvature of the surface at the contact point, several techniques on local gouging detection and correction have been developed [19, 21, 27, 36, 37]. Besides, some other approaches are presented by other researches such as rolling ball and arc-intersection methods by Gray et al [12, 13], penetration–elimination method by Hosseinkhani [16].

For 5-axis gouge detection and correction, a number of researches with different strategies have also been done. Lee and Chang [22] proposed a 2-phase approach to global tool interference avoidance. First, a quick feasibility checking is used to find the conservative feasible tool orientation by using the control mesh of the surface. Second, a detailed feasibility checking uses the exact surface to detect the tool collision. With haptic rendering application, Ho et al. [15] detected and corrected in real-time the collision between the tool and the workpiece to be machined. Ilushin et al. [17] used space subdivision techniques to solve the problem of global collision. To perform effective collision avoidance, Lauwers et al. [20] developed a system that integrates the tool path generation module, the post-processing and machine simulation.

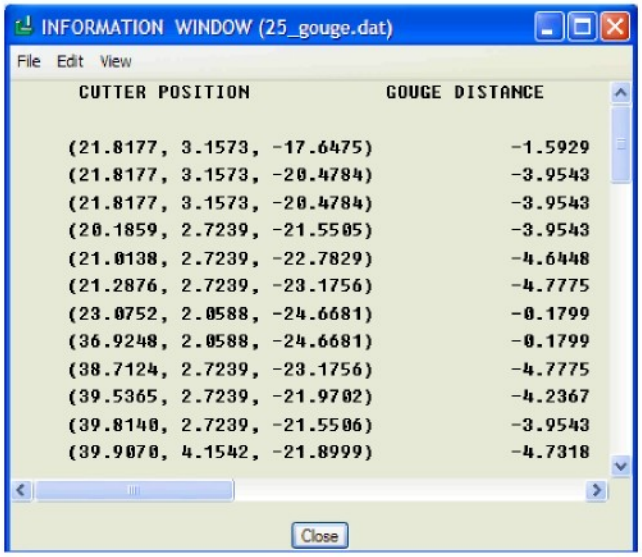
Global gouging can be avoided by finding the suitable range of lead and tilt angles of the tool. It means to find the right tool posture at every cutter contact point when the tool moves on the tool path. The feasible tool approach direction can be determined in two ways [19]:

a) Define a given direction of tool that is feasible or not,

b) Calculate the feasible range of all tool directions at every cutter contact points on the surface.

In the first method, if the given direction is not feasible, then a new direction must be applied and the process should be repeated until the suitable one is found. The method is easy to carry out but sometime it is time-consuming and we can not get the optimal tool direction. This drawback can be solved by using the second method. However, this method is very difficult to execute. For a given part surface, the global gouging is effected by tool radius, tool effective length and tool holder diameter.

Nowadays, gouging can be detected in many commercial CAD/CAM packages. For example, in Pro/Engineer, gouge checking capabilities are provided to allow users a quick check of cutter location (CL) data for milling NC sequences. Users can perform gouge checking in every NC sequence or in the whole operation for individual or all surfaces of the part. To check gouge for the whole operation, the CL data file is required. If no gouges are detected upon completing the check, the system will inform users a message that there are no gouges on the detected surface(s). Otherwise, if gouges are detected, the system will inform users the value of the first detected gouge and can highlight the places of the CL points. The system can also supply the information such as cutter position where gouges occur and the correlative gouge distances as well, as shown in Figure 4.3.



```
     CUTTER POSITION            GOUGE DISTANCE

    (21.8177, 3.1573, -17.6475)          -1.5929
    (21.8177, 3.1573, -20.4784)          -3.9543
    (21.8177, 3.1573, -20.4784)          -3.9543
    (20.1859, 2.7239, -21.5505)          -3.9543
    (21.0138, 2.7239, -22.7829)          -4.6448
    (21.2876, 2.7239, -23.1756)          -4.7775
    (23.0752, 2.0588, -24.6681)          -0.1799
    (36.9248, 2.0588, -24.6681)          -0.1799
    (38.7124, 2.7239, -23.1756)          -4.7775
    (39.5365, 2.7239, -21.9702)          -4.2367
    (39.8140, 2.7239, -21.5506)          -3.9543
    (39.9070, 4.1542, -21.8999)          -4.7318
```

Figure 4.3: An example of the information about detected gouges in Pro/Engineer.

In the example above, the surfaces to be protected from gouging can be (a) drive surfaces (surfaces to be machined), (b) check surfaces (other surfaces on the part), or (c) both. These surfaces are manually added or removed by users.

In this study, by surface partitioning, the design surface is partitioned into regions. Thanks to this advantage of surface partitioning, the gouge detection and correction can be effectively performed in CAD/CAM packages in two forms as follows.

a) Defining the most suitable tools for machining the regions under the condition of local gouge-free.

b) Choosing the reasonable orientation of the tool when machining each region and detecting global gouging.

In the second form, a minimum number of detecting surfaces can be chosen by evaluating the distribution characteristics of the region to be machined and the whole partitioned surface. Hence, the processing time of the CAD/CAM system can be reduced. These two forms will be described in detail in the next subsections.

4.2 Local gouging avoidance and definition of tool diameter

It is obviously that a ball-end cutter will not cause any local gouges on the design surface if the cutter radius is smaller than the radii of the surface curvatures in the vicinity of the CC point. And of course, in the case of flat-end and toroidal cutters, if the effective radius of the cutter is less than the minimum radius of curvature of the surface patch then local gouging will not be occurred when machining that patch. To calculate the minimum radius of curvature of a region, the maximum surface curvature should be known. Hence, form the value of the maximum surface curvature of that region, the possible maximum radius of the cutter can be chosen, as presented by Equation (1.1) and (1.2). This possible maximum radius is the base for choosing the tool diameter to machine the region.

In this reseach, the possible maximum radius of a ball-end cutter can be executed by two ways as follows.

1. In Matlab®:
   - Calculate the principal curvatures (K_{min} and $K_{max)}$ at every point of the selected region, and store all the values of K_{max} in matrix.
   - Extract the maximum value from that matrix, that is K^{S}_{max} .
   - Calculate the possible maximum radius of the cutter by Equation (1.1).
2. In Pro/Engineer®:
   - Create the partitioned surface.
   - Use the Geometry Analysis tool with the radius option to get the minimum radius for the selected region, or

- Use the curvature option in the Geometry Analysis tool to evaluate the curvature of surface and get the maximum value for the selected region, and
- Calculate the possible maximum radius of the cutter by Equation (1.1).

For flat-end mill and toroidal end mill, the procudures above can also be applied, but here the effective radius of the cutter is compared to the minimum radius of curvature of the surface. Hence, R_{eff} is used instead of R_{max} (in Equation (1.1)). Remember that when changing the Sturz angle, this results in the change of the effective radius of a flat-end cutter or a torroidal cutter, then local gouging may occur. Therefore care must be taken when choosing the Sturz angle for machining surfaces with flat-end and toroidal cutters. Normally, users can choose a suitable Sturz angle for machining each patch thanks to some simple calculations.

From the possible maximum radius of the cutter, the diameter of the tool used to machine each region can be easily defined. Howerver, sometimes a smaller diameter should be applied to avoid gouging. A combination of reducing the tool diameter and choosing the suitable Sturz angle may also be uztilized for global gouging avoidance.

4.3 Global gouging avoidance and definition of tool orientation

4.3.1 Overview

In free-form surface machining, a ball-end, flat-end or toroidal cutters can be used to mill saddle and concave regions. If the radius or the effective of the cuter is smaller than the minimum radius of curvature of the region, then local gouging will not occur, as mentioned above. However, the tool can cause global gouging in the following cases:

1. The tool is machining in a concave region, global gouging may happen between the cylindrical body of the tool and other saddle or convex regions nearby. An example of this case is given in Figure 4.4.
2. The tool is machining in a saddle region, global gouging may happen in this region itself, other neighbouring saddle or convex regions.
3. The tool is machining in a convex region, global gouging may happen in other

neighbouring saddle or convex regions.

Figure 4.5 below shows a ball-end cutter machining in a concave region near a convex region. In this case, if the distance from the tool axis to the nearest point on the part surface is smaller than the cutter radius, the interference will occur. To avoid interference, the tool should be inclined to another direction that performs a smaller Sturz angle (θ) so that the distance above is longer than the cutter radius. In industry, the Sturz angle is typically between 5 and 10^0 [19]. In practice, the Sturz angle is defined by lead and tilt angles. These angles are introduced in the next subsection.

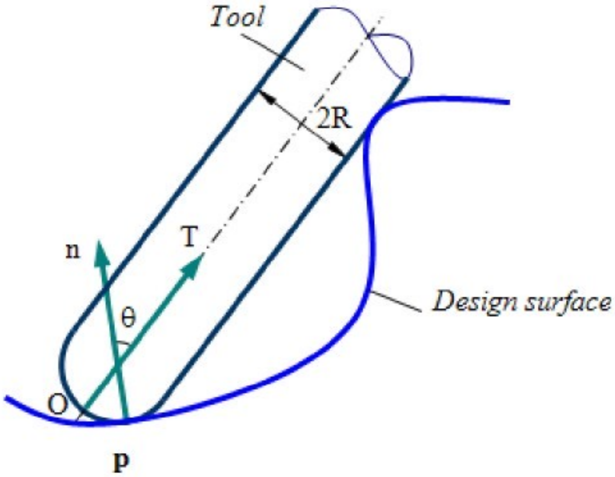


Figure 4.4: An example of global gouging.

4.3.2 Lead and tilt angles

Compared with 3-axis milling, 5-axis milling is more complicated due to the two additional degrees of freedom. The major difference between these two operations is the existence of the lead and tilt angles. The lead angle is the rotation of the tool axis about the cross-feed axis, whereas the tilt angle is the rotation about the feed axis with respect to the surface normal [45], as shown in Figure 4.5. These angles can be zero, positive or negative [2] and they define the tool posture during machining. If the lead and tilt angles are zero then the tool axis will coincide with the normal vector at the CC point.

In Figure 4.5, OXYZ is the workpiece coordinate system and $pFCn_S$ is the local coordinate system where the lead and tilt angles are defined. The local coordinate system is established by feed (F), cross-feed (C) and surface normal (n_S), used to analyze the cutting operation at the CC point (p). Since the cross-feed direction is perpendicular to the feed direction, hence F, C, n_S vectors establish an orthogonal basis.

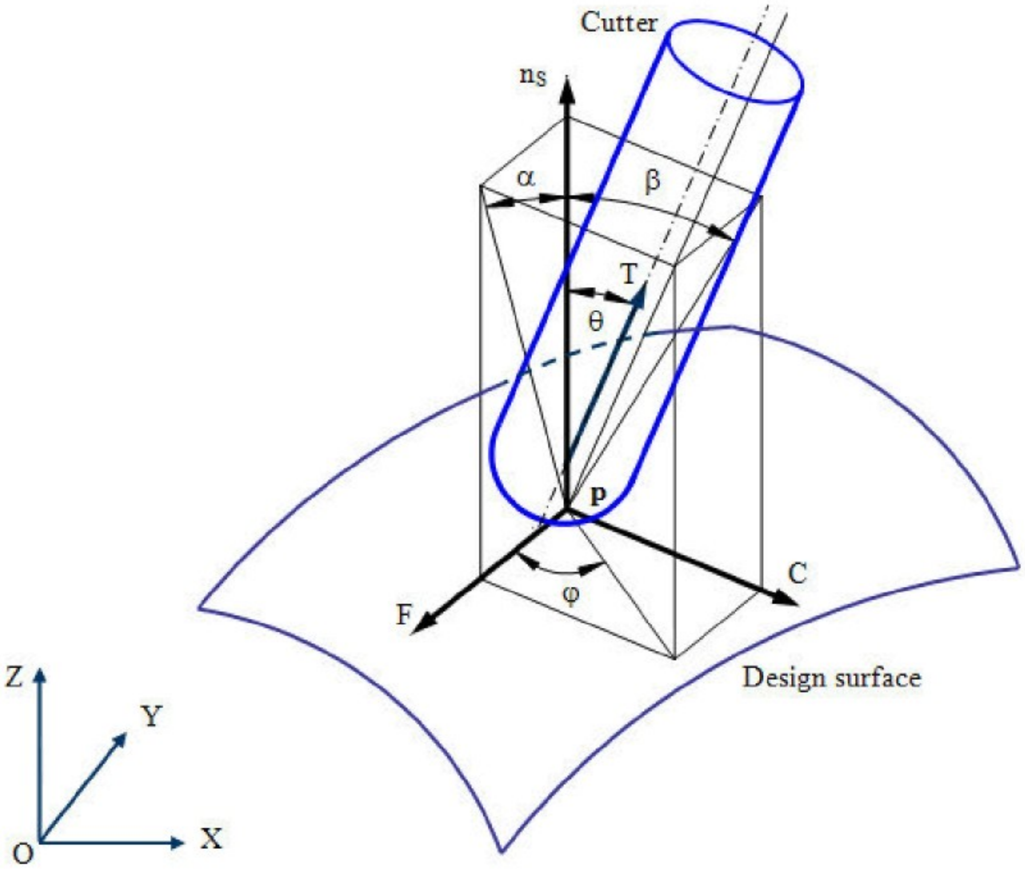


Figure 4.5: Coordinate systems and lead and tilt angles.

α- lead angle, β-tilt angle, θ-Sturz angle, φ-screw angle.

Investigating the optimal values of these angles for a particular cutting process is not easy and really challenging. In practice, the lead and tilt angles are often selected by trial and based on experiences of users. These angles are used as some of the input parameters for the CAM stage in 5-axis machining free-form surfaces. For instance, in Pro/Engineer®, these angles are specified by parameters LEAD_ANGLE and TILT_ANGLE [52]. They define the tool orientation with respect to the surface normal for

5-Axis surface milling NC sequences. As the same manner, EdgeCAM uses "Tilt Angle" and "Lead Lag Angle" for tool axis control in the case of the tool is rotated from its surface normal orientation in two planes [53].

4.3.3 Implementation of gouge avoidance

In this research, to machine region by region of the partitioned surface, 5-axis machining mode is applied for the finishing stage and Pro/Engineer® Wildfire® is used to generate the tool paths for the design surface. In the CAM stage, besides choosing the right tool path strategy for each region, defining the tool orientation that is gouge-free is also very important. Therefore, care must be taken when determining the lead and tilt angles. The following is the procedure for global gouging detection and correction implemented in the research.

In each NC sequence, when establishing the machining parameters, the values of the lead and tilt angles should be entered by trial as follows.

1. Choose primary values for the lead and tilt angles:
2. Choose the surfaces to be detected. They are the neighbouring regions of the region to be machined.
3. Simulate the tool paths and perform gouge checking.
    - If there are no gouges, accept the values of the lead and tilt angles
    - If gouges occur, define the maximum absolute value of the gouge distance.
4. Recalculate the values of the lead and tilt angle based on the maximum absolute value of the gouge distance.
5. Repeat steps 3 and 4 until no gouges occur.

The method above is to apply for the strategy in which the tool axis always makes a constant angle with the normal vector at every CC point (called the Sturz angle method). Sometimes, this method cannot be used because of global gouging, e.g. machining deep concave regions, as shown in Figure 4.6. In this case, a strategy with variable angles between the tool axis and normal vector should be applied. The following are some methods which can satisfy that requirement:

(i) The tool axis always goes through a curve, as shown in Figure 4.7(a).

(ii) The tool axis always goes through a specified point, as shown in Figure 4.7(b).

These suggested methods may be very effective with the partitioned surfaces. This is because the regions on that kind of surfaces can be separately machined by different tools and tool path strategies.

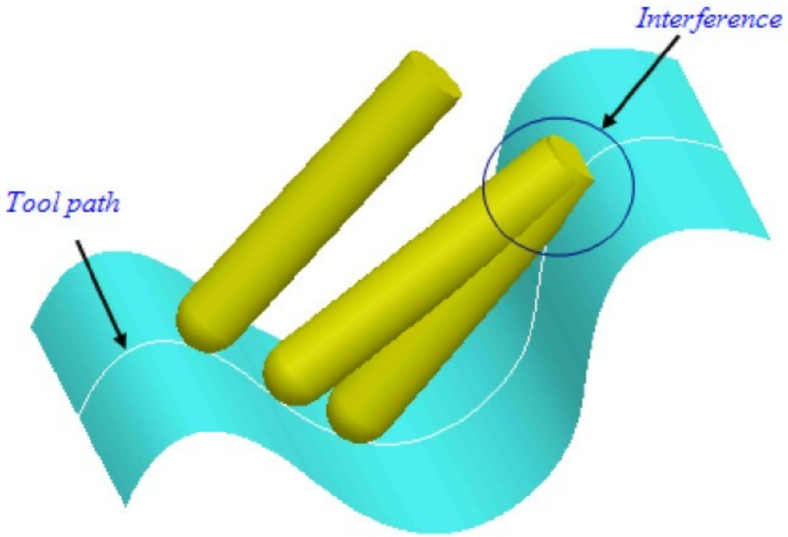
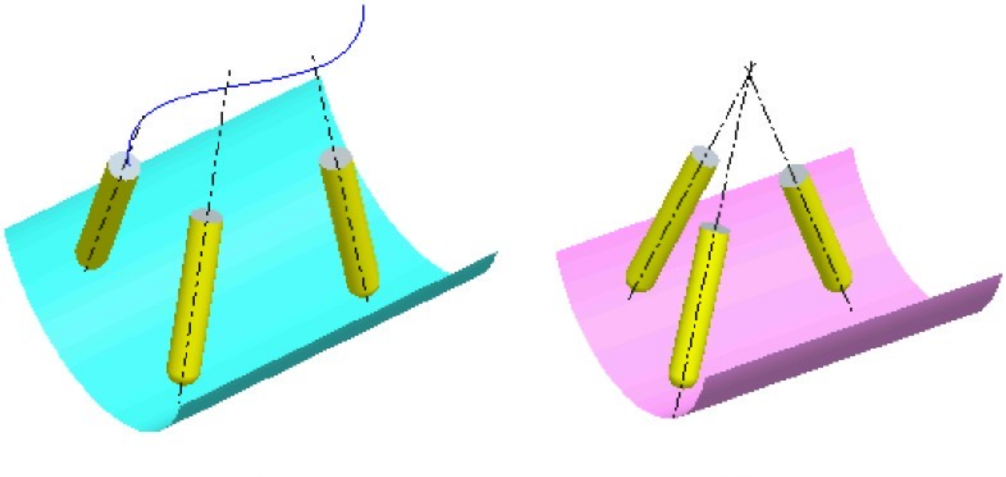


Figure 4.6: Interference when machining deep concave region.



(a) (b)

Figure 4.7: Tool axis always goes through (a) a curve and (b) a point.

4.4 Summary

In this chapter, the background of gouge detection and correction is presented. For local gouging avoidance, it is necessary to choose cutters whose radii (or effective radii) are smaller than the minimum radius of curvatures of the surface patch. Besides, a practical method for global gouging detection and correction in Pro/Engineer is also proposed. This method uses the lead and tilt angles to define the right postures of the cutter that there is no global gouge between the cutter and the design surface.

Chapter 5

TOOL PATH GENERATION

5.1 Introduction of tool path generation

In free-form surface machining, tool path is a series of straight lines and arcs while the design surface is a surface whose curvatures vary. Thus, the design surface is approximated by a series of straight lines and arcs for machining. This approximation results in an inaccurately machined surface. This kind of errors always exists and can be accepted if the errors are within tolerance. Getting a machined surface that is as close as possible to the design surface is one of the machining goals.

An NC tool path used to machine a free-form surface consists of a set of tool positions. The NC controller interpolates sequentially between these points as the tool moves from point to point [50]. Computing a sequence of CL-points from a given surface is the main objective of tool path generation. To generate CL data for machining free-form surfaces, commercial CAD/CAM systems such as Catia®, Pro/Engineer®, Unigraphic® NX®, SurfaceCAM® can be used. In general, users have to determine many parameters related to the machining process for the system. After generating the CL data, post processing is done to get NC codes for machining.

5.2 Tool path generation methods

Depending on the type of tool path generation surface, CC-surface or CL-surface, tool path generation methods are classified into two groups: CC-based method and CL-based method. CC-surface is also the design surface while CL-surface is a surface defined by the trajectory of the reference point of the cutting tool when the cutting tool move over the design surface [6].

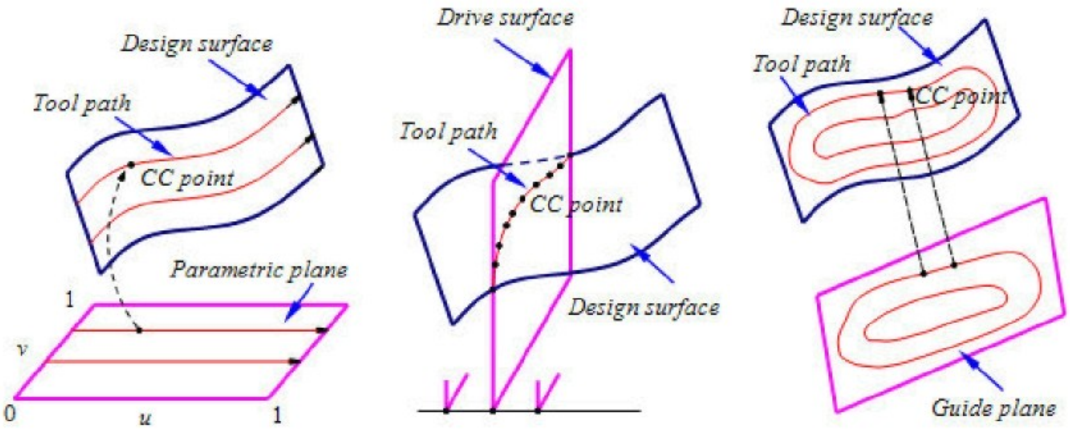
- CC-based method

In the CC-based method, by sampling a sequence of CC points from the design surface, tool paths are generated and then CC-points are converted to CL-points. The CC-point is a point on the design surface at which the cutting tool makes a tangential contact and the CL-point is the corresponding reference point on the cutting tool to specify its location in the NC program [6]. This method can be classified in to 3 main categories as (1) parametric method; (2) drive surface method; (3) guide plane method depending on the path planning techniques [33].

1) *Parametric method*: tool paths are planed on parametric domain and they are mapped back to the design surface, as shown in Figure 5.1(a). Iso-parametric tool path is one of the earliest techniques of this method.

2) *Drive surface method*: tool paths are generated by intersecting the design surface with a series of planes called drive surfaces. The intersecting curves are used for generating tool paths, as shown in Figure 5.1(b). Iso-planar tool path is one of the techniques of this method.

3) *Guide plane method*: tool paths are planed on separate guide plane and then they are mapped back to the design surface as depicted in Figure 5.1(c).



(a) (b) (c)

Figure 5.1: CC-Based tool path generation methods [33].

- CL-based method

In the CL-based method, first, the CL-surface has to be generated from the design surface. This surface is then used as a path generation surface. This method can also be classified in to three main categories as depicted above.

In general, it is very difficult to obtain the optimum tool paths for machining free-form surfaces. Up to now, a number of methods have been developed to generate tool paths for machining free-form surfaces as mentioned in References [8, 51]. The following are some commonly used methods [5].

1) *Iso-parametric methods*

In these methods, uniformly spaced parametric lines in the parametric domain are generated then mapped on the design surface. The result is that the tool paths are non-uniformly spaced curves because of the non-uniform transformation between parametric and Euclidean spaces. Therefore, the scallop heights on the machined surface are varied and thus a non-uniform surface finish is obtained. This method may be the most common used method due to its straightforwardness.

2) *Iso-planar methods*

In these methods, tool paths are planned as the intersection curves between the design surface and a family of parallel planes. These methods are often preferred for three-axis machining [51]. Similar to the iso-parametric methods, the iso-planar methods do not correspond to a constant scallop height

3) *Iso-scallop methods*

The tool path generation methods that can maintain a constant scallop height are called the *iso-scallop tool path generation* methods. In these methods, the tool path interval is variable so that the scallop height is constant along the CC path. These methods have been developed to achieve high efficient surface machining [5] in three-axis machining as well as in five-axis machining, and were proposed in References [10, 26, 28, 43, 44, 46].

Scallops are ridges, cusps and other surface protrusions left between adjacent overlapping tool passes that extend above the design surface profile [31]. The tool pass interval, the tool type and surface characteristics determine the size of scallops left on the surface [50]. Figure 5.2 represents scallops left after machining a plane by a ball-end cutter. In five-axis machining, controlling scallop height is an important factor because a small scallop height significantly reduces the time for manual grinding and polishing in the benchwork stage.

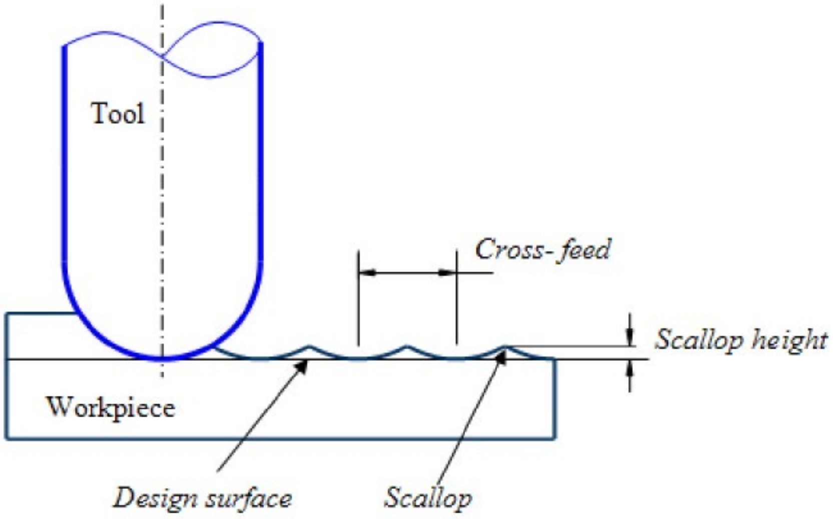


Figure 5.2: Scallops left after machining by a ball nose cutter.

5.3 Implementation of tool path generation

There is no new algorithm for generating tool paths developed in this research. To evaluate the efficiency of surface partitioning in term of machining time, in comparison with the traditional method in which the whole surface is machined by one cutter, the iso-scallop method is utilized. Currently, scallop height control, that can provide a constant surface finish, is available in commercial softwares such as Pro/Engineer®, EdgeCAM®, Turbocam®, SurfCAM®, VISI CAM®, FeatureCAM®,… In some softwares, the term "cusp height" is used instead of the term "scallop height", for example EdgeCAM® ans GibbCAM®. In our research, Pro/Engineer® is used to generate the tool paths for each separate region. In Pro/Engineer®, parameter SCALLOP_HGT is used to control tool step by specifying the maximum allowable scallop for surface milling [52].

In the research, for comparison purpose, the machining time of several surfaces are tested. The design surfaces are machined by two ways with the same scallop height value. The first one is that the entire surface is machined by only one cutter and the last one is that the partitioned surface is machined patch by patch by different cutters. And of course, some other machining parameters are also identical for the two ways (refer chapter 6 for more information).

5.4 Summary

In this chapter, a short introduction of tool path generation methods is presented. The iso-scallop method is used for generating the tool paths of the design surface. In order to compare the machining time between the proposed method and the traditional one, the design surfaces are milled with the same scallop height value. The parameter SCALLOP_HGT in Pro/Engineer® is used for this purpose.

Chapter 6

APPLICATIONS

The main objective of this chapter is to illustrate the applications of the proposed methodologies that are presented in the previous chapters. In this chapter, three different examples are given to demonstrate the implementation of the proposed method on free-form surface machining. In each example, four important tasks including surface partitioning, defining the region boundaries, generating the tool paths for each region and machining simulation the operation for the whole surface are presented. The results of comparisons in terms of machining time between the proposed method (with surface partitioning) and the traditional method (without surface partitioning) are also represented. Besides, for understanding purpose, some explanations about the Matlab® function and script files, building CAD models of free-form surfaces are expressed before illustrating the application. In order to confirm that the proposed method is valid in practice, some machining experiments were conducted on the multi-tasking machine Mazak® Integrex IV-100. The experiment results are also shown in the last portion of the chapter.

6.1 Matlab® function and script files

In this study, a Matlab® program that consists of and some function and script files were written and developed. The M-script file executes the two main tasks of surface partitioning and finding the boundaries of all regions. The M-function files are utilized to support for the M-script file. The following are some characteristics and applications of these Matlab® function and script files.

6.1.1 The M-script file

The inputs of the M-script file are the coordinates of the points of the control net and knot vectors that define the B-spline surface used for surface partitioning. The following are some important outputs of this file:

1) 3D coordinates of points on the design surface,

2) 3D coordinates of points of different regions on the design surface,

3) 3D coordinates of points on the boundaries of the regions,

4) Maximum values of the principal curvatures of each region,

5) Figure of the design surface with highlighted points on the boundaries of regions.

The 3D grid points on the design surface are used to create the datum curves of the surface and to form a set of points for surface partitioning. To create the datum curves (see Section 6.2.1), a very limited points are used, while a numerous of points are required to accurately partition surface into regions. The higher the density of the grid points, the more accurately the partitioned regions. Therefore, it is better to choose two different values of the grid distance on the surface. For this purpose, first the M-script file should be run with a small value of the grid distance to get the 3D points for creating the datum curves of the B-spline surface. Last, a large one is applied to get a dense 3D grid points for surface partitioning.

6.1.2 The M-function file for calculating the geometric parameters of the surface

This function file is used to compute some geometric parameters of the design surface. These parameters are used for the surface partitioning procedure and some other purposes. The following are computations that have to be done in this function file:

1) Computation of unit normal vectors at every point on the design surface, based on Equation (2.15): unit normal vectors are used to calculate the fundamental magnitudes (L, M and N) of the second order, which then in turn are used to compute Gaussian and mean curvatures.

2) Computation of Gaussian curvatures at every point, based on Equations (2.17) – (2.20).

3) Computation of mean curvatures at every point, based on Equations (2.17) – (2.19), (2.21): Gaussian and mean curvatures are used to group points on the surface into regions.

4) Computation of principal curvatures at every point, based on Equations (2.20) – (2.23): principal curvatures are used to define the maximum diameter of the cutters to machine regions.

In this function file, the 3D coordinates x, y and z of the points on the surface are the input parameters. The output parameters of the function file are unit normal vector $\mathbf{n}_S$, K, H, K_{max}, and K_{min}.

6.1.3 Other M-function files

In order to create B-spline surfaces in the main M-file (the M-script file), some supporting function files were also written. They are function files for determining the B-spline basis functions, knot vector, etc... To get the points on the boundaries of the regions, an M-function file named *boundaries* [11] is also required.

6.2 Building CAD models of freeform surfaces

6.2.1 ibl file format

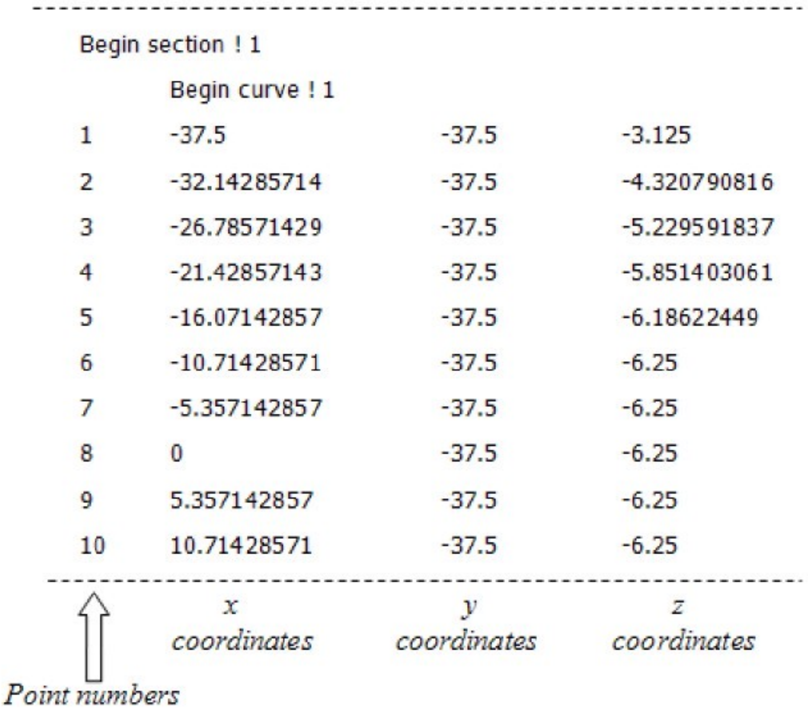
Because Pro/Engineer® Wildfire® does not support to directly create B-spline surfaces from their control points, so that a replacement technique is required. In our research, ibl files (files with the extension .ibl) are used in the procedure of creating B-spline surfaces from imported datum curves. In fact, the datum curves are created in form of an ibl file and this file is then imported into Pro/Engineer® to create a B-spline surface. It should be noted that the datum curve is a spline created from more than two points [52] and the B-spline surface to be created goes through these curves.

Basically, ibl files can be created in the VB6.0 or C++ Builder program. However, they can also be created by Notepad or WordPad applications. An ibl file is a text file that contains geometry defined by 3D points. The structure of the ibl file format that has 9 curves is as follows

```
Open   Arclength
   Begin section ! 1
```

```
    Begin curve !  1
    Points list
Begin section !  2
    Begin curve !  2
    Points list
.
.
.
Begin section !  9
    Begin curve !  9
    Points list
```

In the ibl file format, there is a list of points for each curve. These points are represented by x, y and z coordinates. Actually, they are datum points which Pro/Engineer® will fit splines. The points for creating the curves in the ibl file can be received by sampling points on the design surface in Matlab®. Figure 6.1 shows the structure of a curve through 10 points in the ibl file format.



```
Begin section ! 1
        Begin curve ! 1
1       -37.5           -37.5     -3.125
2       -32.14285714    -37.5     -4.320790816
3       -26.78571429    -37.5     -5.229591837
4       -21.42857143    -37.5     -5.851403061
5       -16.07142857    -37.5     -6.18622449
6       -10.71428571    -37.5     -6.25
7       -5.357142857    -37.5     -6.25
8       0               -37.5     -6.25
9       5.357142857     -37.5     -6.25
10      10.71428571     -37.5     -6.25
```

x
coordinates
y
coordinates
z
coordinates
Point numbers

Figure 6.1: The structure of a curve in the ibl file format.

6.2.2 Building CAD models of free-form surfaces in Pro/Engineer® Wildfire®

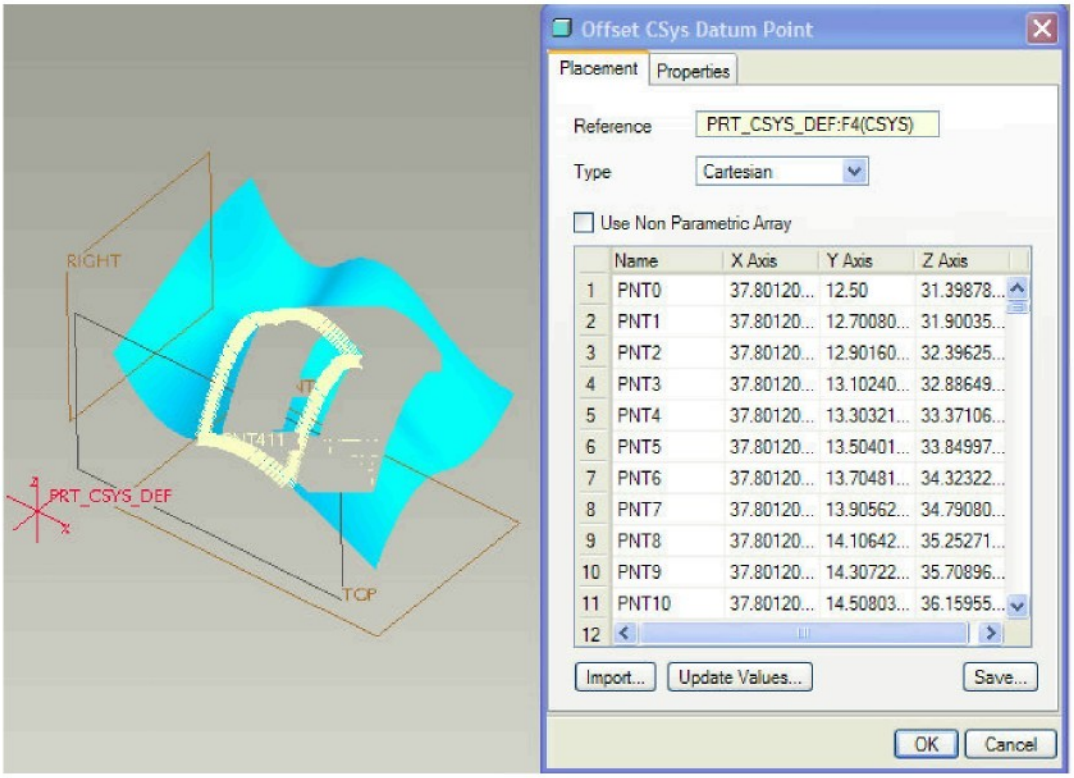
In order to create the CAD model of a free-form surface in Pro/Engineer® Wildfire®, first an ibl file should be performed. Once having the ibl file, the following procedure can be used to import a surface from an ibl file:

1. Choose Insert > Advanced > Blend from file > Surface,
2. Select the coordinate system that curves will refer to. Normally, the default coordinate system PRT_CSYS_DEF is chosen,
3. Choose the file name (with the extension .ibl),
4. Choose the direction of adding material.

As mentioned earlier, it is better to split the design surface into regions in the CAD stage. The result is that in the CAM stage users can choose region by region to generate the tool paths. For the splitting purpose, the boundary curves of partitioned regions are required. These 3D curves can be created from the boundary points which their coordinates are obtained from the calculating results in Matlab®.

To create a 3D curve through datum points, the Offset Coordinate System Datum Point Tool can be used. In this tool, the coordinates of the points should be stored in files with the extension .pts and they are imported form the Import tab in the Offset CSys Datum Point window as in Figure 6.2. The coordinate system and its type are also necessary when using this tool. An example of creating 3D curve through datum points, using the Offset Coordinate System Datum Point Tool is illustrated in Figure 6.2. The Insert a Datum Curve tool can be used to get a spline curve through points.

The Surface Trim tool in the Style tool can be used to divide the design surface into regions. In this case, the 3D curve mentioned in the paragraph above is an important factor for trimming. Figure 6.3 shows an example of trimming the convex region of the surface illustrated in Figure 6.2.



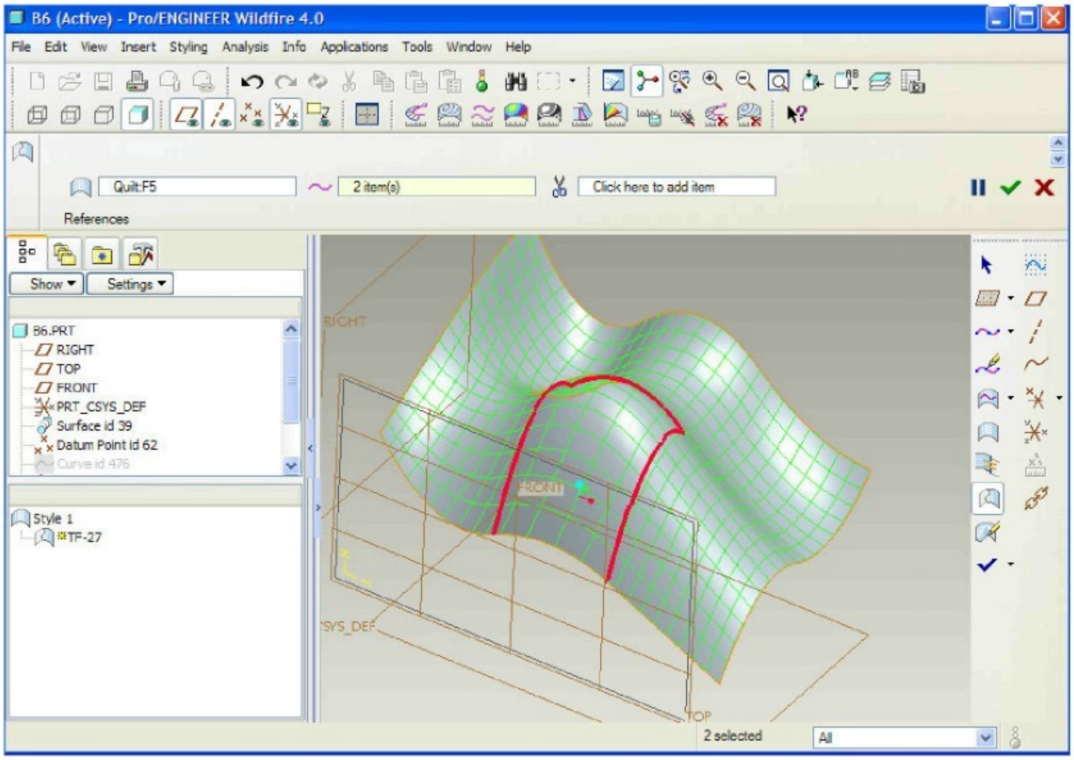
Reference PRT_CSYS_DEF:F4(CSYS)

Type Cartesian

□ Use Non Parametric Array

	Name	X Axis	Y Axis	Z Axis
1	PNT0	37.80120...	12.50	31.39878...
2	PNT1	37.80120...	12.70080...	31.90035...
3	PNT2	37.80120...	12.90160...	32.39625...
4	PNT3	37.80120...	13.10240...	32.88649...
5	PNT4	37.80120...	13.30321...	33.37106...
6	PNT5	37.80120...	13.50401...	33.84997...
7	PNT6	37.80120...	13.70481...	34.32322...
8	PNT7	37.80120...	13.90562...	34.79080...
9	PNT8	37.80120...	14.10642...	35.25271...
10	PNT9	37.80120...	14.30722...	35.70896...
11	PNT10	37.80120...	14.50803...	36.15955...
12				

Figure 6.2: An example of creating datum points for 3D curve.



Quilt:F5 | 2 item(s) | Click here to add item

2 selected | All

Figure 6.3: An example of trimming a region on a surface.

6.3 Applications and discussion

6.3.1 Overview

A number of examples have been implemented in our research. At first, some simple cases were conducted for easy implementation and checking the developed algorithms. In these cases, we have chosen the B-spline surfaces which have only two or three regions. After having confirmed that all algorithms worked well based on those simple cases, some complicated cases were carried out. The latter is to aim to improve that the proposed method can be available for many different applications. The implementation procedure is performed as shown in Figure 6.4 below.

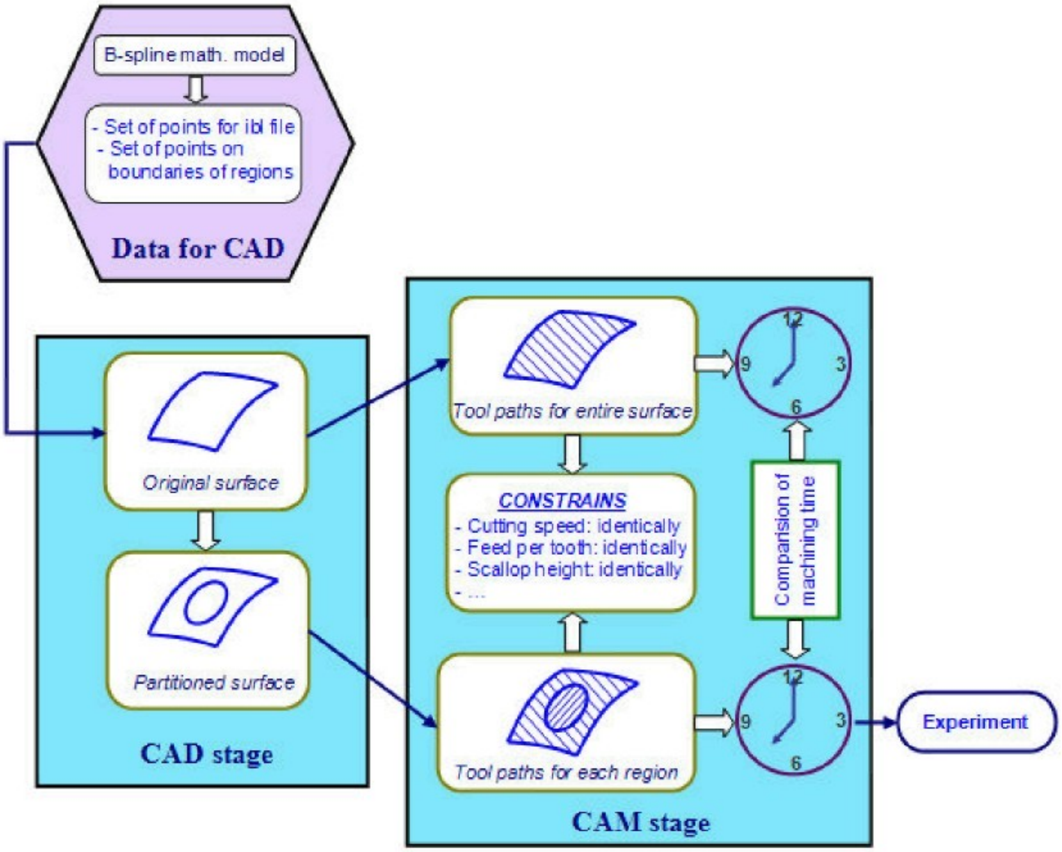

B-spline math. model
- Set of points for ibl file
- Set of points on boundaries of regions
Data for CAD
Original surface
Partitioned surface
CAD stage
Tool paths for entire surface
CONSTRAINS
- Cutting speed: identically
- Feed per tooth: identically
- Scallop height: identically
- ...
Tool paths for each region
CAM stage
Comparision of machining time
12
3
6
9
12
3
6
9
Experiment

Figure 6.4: Flow chart of the implementation.

The first stage in the implementation procedure is to create the data for the CAD stage. This stage is mainly done in Matlab®. According to the calculating results from

Matlab®, the following data is useful for further work:

1) Number of regions,

2) Coordinates of the points on the boundaries,

3) Values of maximum curvatures on regions,

From these data, the B-spline surface and its regions can be easily created in Pro/Engineer as described in Section 6.2.

For determining the tool axis orientation, lead and tilt angles can be applied. Here, the cutter axis always establishes a fixed inclination angle with respect to the surface normal. In other words, the surface normal vector is used as the reference for tool axis definition. However, in the cases that the suitable lead and tilt angles for tool positioning can not be found, a variable inclination angle of the tool with respect to the surface normal should be used. In Pro/Engineer®, the parameter IGNORE_SURF_NORMS should be chosen to compute the tool axis definition without respect to the surface normal, instead of the parameter USE_SURF_NORMS (the default) [52].

Machining time is an important criterion. Therefore, it should be compared between the proposed method and the traditional one for machining a given surface. In this study, the operation for machining a given surface is performed in two passes, i.e. rough and finish. To compare the machining time of the finish pass between the proposed method and the conventional method, some constrains should be applied for both methods when generating the gouge-free tool paths. More concretely, the machining strategy and machining conditions for the two methods should be similar. Regardless of machining by any tool, the values of the following parameters for finishing should be the same:

1) Cutting speed,

2) Feed rate per tooth,

3) Stock left after rough machining, and

4) Scallop height.

In order to get the machining time of the operation for each application, the NC program was generated by a post processor of the 5-axis CNC Shoda Router. This post processor is available inside Pro/Manufacturing®.

In the next subsections, three typical cases are presented. Two of them were experimented to validate the implementation.

6.3.2 Application 1

Assume a B-spline surface defined by 5 × 5 control net as in Table 1. The uniform knot vector in both parametric directions is [0 1 2 3 4 5 6 7]. This surface can be constructed in Matlab® as shown in Figure 6.5. In this Figure, the points on the boundary are highlighted.

Table 1: Control net of a B-spline surface

(-40,-40,0)	(-20,-40,0)	(0,-40,0)	(20,-40,0)	(40,-40,0)
(-40,-20,0)	(-20,-20,0)	(0,-20,0)	(20,-20,0)	(40,-20,0)
(-40,0,0)	(-20,0,0)	(0,0,-30)	(20,0,0)	(40,0,0)
(-40,20,0)	(-20,20,0)	(0,20,-40)	(20,20,0)	(40,20,0)
(-40,40,0)	(-20,40,0)	(0,40,-20)	20,40,0)	40,40,0)

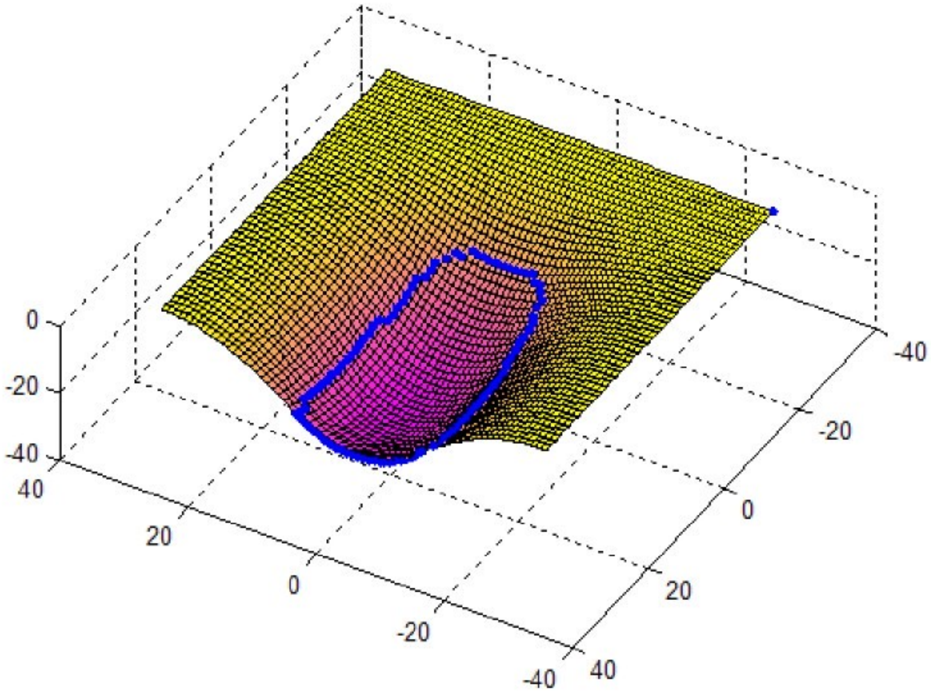


Figure 6.5: Illustration of a freeform surface in Matlab® (application 1).

The calculating results from Matlab® shows that there are only two regions on this surface: one concave region and one saddle region. The maximum curvatures on the concave and saddle regions are 0.1830 mm^{-1} and 0.0840 mm^{-1}, respectively. As a consequence, there is only one boundary required to split the design surface into regions. And if ball-end cutters are used, to avoid local gouging, their radius must be smaller than 5.465 mm and 11.905 mm to machine the concave and saddle regions, respectively.

The CAD model of the design surface and the boundary between the two regions created in Pro/Engineer® are illustrated in Figure 6.6. The partitioned surface with its two regions is shown in Figure 6.7.

In this application, the machining strategy for the partitioned surface can be as follow

1) Roughing cut in 3-axis mode by a two-flute flat-end mill of 10 mm in diameter,
2) Finishing cut in 5-axis mode for the saddle region by a four-flute ball-end cutter of 20 mm in diameter,
3) Finishing cut in 5-axis mode for the concave region by a two-flute ball-end cutter of 10 mm in diameter.

Assuming that the workpiece made from wrought medium-carbon alloy steels (275 – 325 HB), and carbide tools are used. The machining parameters for the first application are given in Table 2. These parameters are also used for the traditional method to machine the design surface with a two-flute ball-end cutter of 10 mm in diameter.

It must be noted that Pro/Engineer® can use parameters SCALLOP_HGT and STEP_OVER [52] to control the scallop height and the step-over distance of the tool respectively in the same sequence. In this case, the system will automatically calculate the scallop height which will be generated by the STEP_OVER and then compares that value to that of SCALLOP_HGT. The system will use the lesser of the two [52]. Hence, in Table 2, the width of cut is chosen by a rather big value so that the system will choose the value of the scallop height parameter.

Table 2: Machining parameters for 1st application

Parameter	Rough	Finish
Cutting speed (m/min)	35	50
Feed (mm/rev/tooth)	0.2	0.1
Scallop height (mm)	-	0.02
Left stock (mm)	0.5	0
Depth of cut (mm)	3	0.5
Width of cut (mm)	3	tool radius

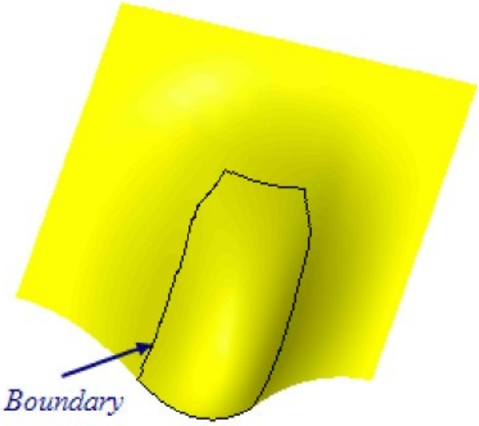


Figure 6.6: Original surface with the boundary between two regions.

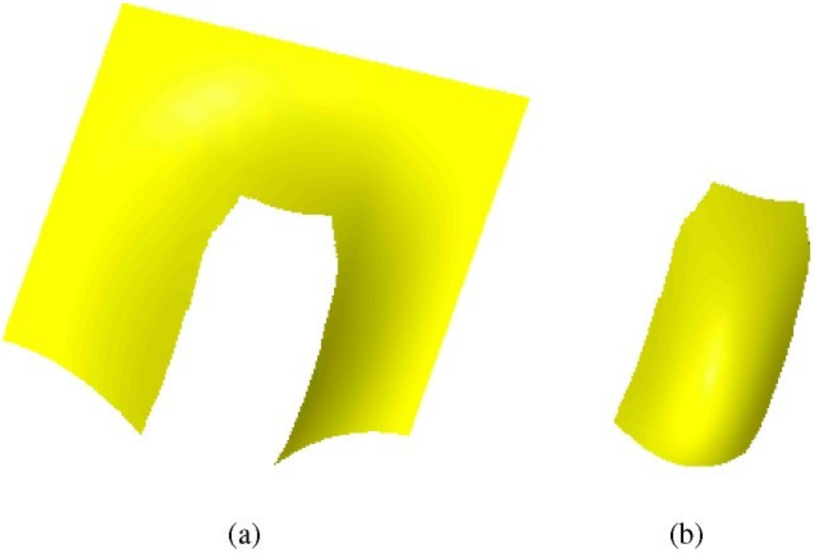
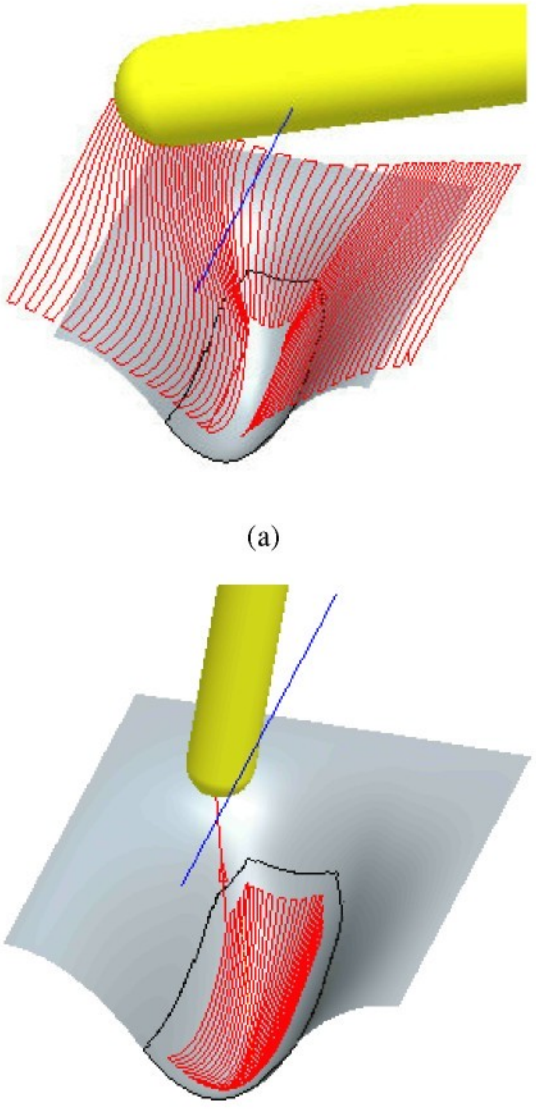


Figure 6.7: Partitioned regions (application 1): (a) saddle region and (b) concave region

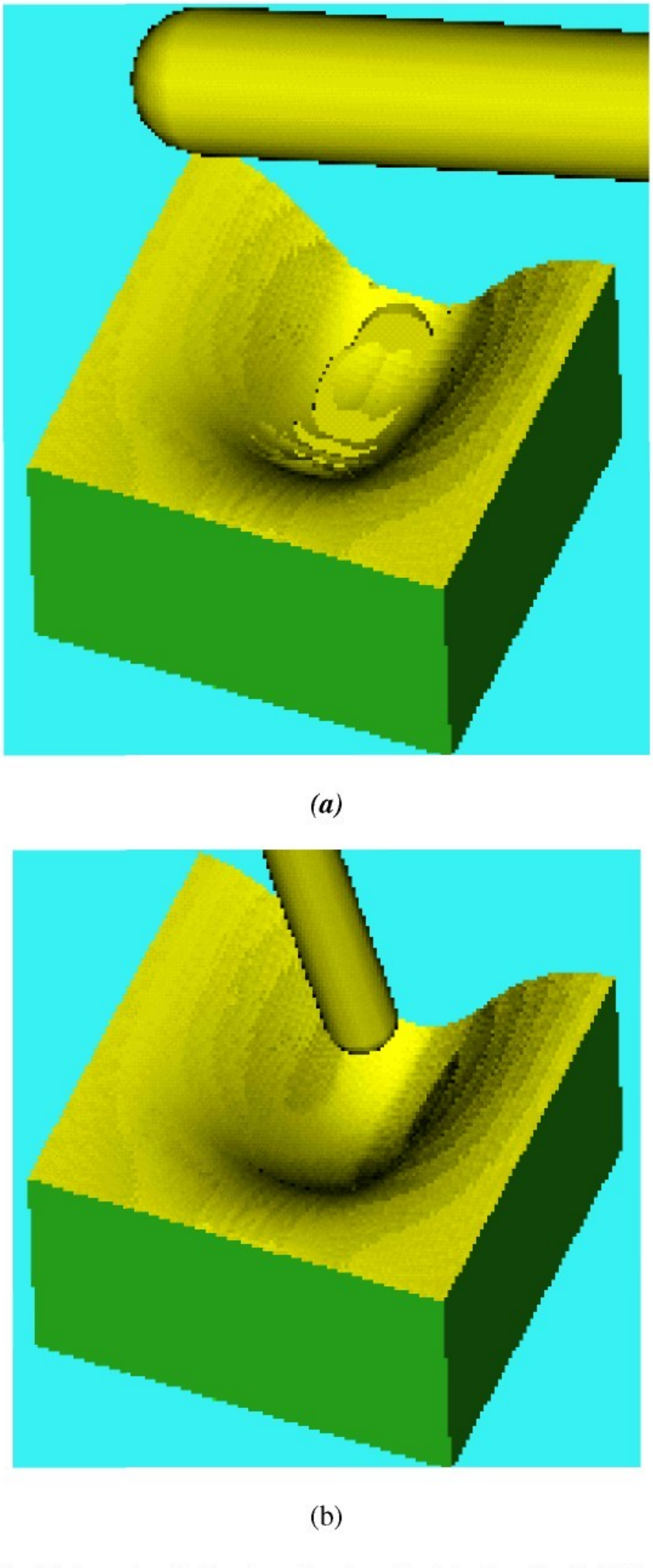
The tool paths for each region on the design surface can be created with different strategies. In this example, the Sturz angle method is not available because the global gouging will occur on the saddle region. Figures 6.8 below shows an example the tool paths generated for each region under the condition that the cutter axis always goes through a fixed straight line over the stock. Figure 6.9 illustrates the results of the machining simulation of the surface after the finishing cut for the two regions.



(a)

(b)

Figure 6.8 Example of tool paths for partitioned regions (application 1): (a) tool paths for saddle region, and (b) tool paths for concave region.



(a)

(b)

Figure 6.9: Machining simulation (application 1): (a) after the finishing cut for the saddle region and (b) after the finishing cut for the concave region.

Table 3 presents the results about the machining time and tool path length for the finishing cut estimated by Pro/Engineer®. It is easy to make a simple comparison between the two methods. It can be seen that the proposed method has much shorter time in machining. The tool path length for finishing of the proposed method approximately equals 85% to that of the traditional method. This reduction leads to a saving up to nearly 20% of the machining time for finishing cut of the proposed method, compared to that of the traditional method. It is a very impressive number. Hence, in this case, it can be said that the proposed method is really effective in term of machining time.

Table 3: Estimated machining time and tool path length comparison (application 1)

Method	**Tool path length (mm)**	**Estimated machining time (min)**
Traditional method (un-partitioned surface)	516.25 (100%)	21.29 (100%)
Proposed method (partitioned surface)	441.84 (85.5%)	17.06 (80.1%)

The application above is a very simple case. From this application, it can be realized that the proposed method can be successfully implemented to partition a simple surface into regions and each region can be separately machined by different cutting tools. However, some other complicated cases should be carried out for a more comprehensive view. In the next subsections are some typical ones that we have done in the research.

6.3.3 Application 2

Table 4 presents the 7×7 control net of a B-spline surface with the uniform knot vectors [0 1 2 3 4 5 6 7 8 9] in u and v directions. The shape of the surface and the boundaries of regions (in form of highlighted points) on the surface created in Matlab® are shown in Figure 6.10. In this example, the surface is partitioned into six regions by four boundaries as illustrated in Figure 6.11.

In comparison with the previous one, this application is much more complicated. Here, the design surface can be partitioned into six regions by four boundaries as shown in Figure 6.10. The CAD model of the design surface with its regions created in Pro/Engineer® is illustrated in Figure 6.11. The values of the maximum curvature of the concave and saddle regions are given in Table 5. From these values, the probable maximum radius of the ball-end cutter used to mill each region is also calculated, as shown in Table 5.

Table 4: Control net of a B-spline surface for application 3

(0,0, 10)	(0,15,30)	(0,30,40)	(0,50,50)	(0,70,40)	(0,85,30)	(0,100,10)
(15,0,30)	(15,15,0)	(15,30,50)	(15,50,60)	(15,70,50)	(15,85,0)	(15,100,30)
(30,0,40)	(30,15,60)	(30,30,70)	(30,50,70)	(30,70,70)	(30,85,60)	(30,100,40)
(50,0,45)	(50,15,60)	(50,30,70)	(50,50,75)	(50,70,70)	(50,85,60)	(50,100,45)
(70,0,40)	(70,15,60)	(70,30,70)	(70,50,70)	(70,70,70)	(70,85,60)	(70,100,40)
(85,0,30)	(85,15,50)	(85,30,40)	(85,50,10)	(85,70,40)	(85,85,50)	(85,100,30)
(100,0,5)	(100,15,15)	(100,30,30)	(100,50,20)	(100,70,30	(100,85,15)	(100,100,5)

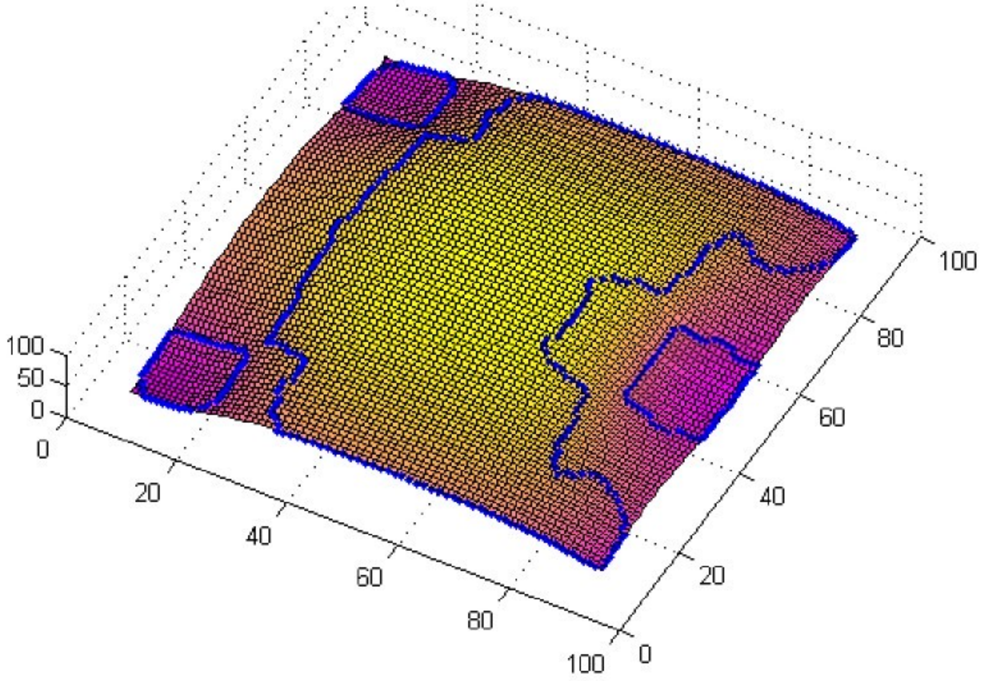


Figure 6.10: Illustration of a freeform surface in Matlab® (application 2).

Table 5: Maximum curvatures K_{max} of regions and probable maximum radii R_{max} of the ball-end cutter used to mill these regions

Region	**K_{max}, mm^{-1}**	**R_{max}, mm**
1, 2	0. 2229	4.486
3	0.2113	4.733
4	0.1010	9.901
5	0.1054	9.488

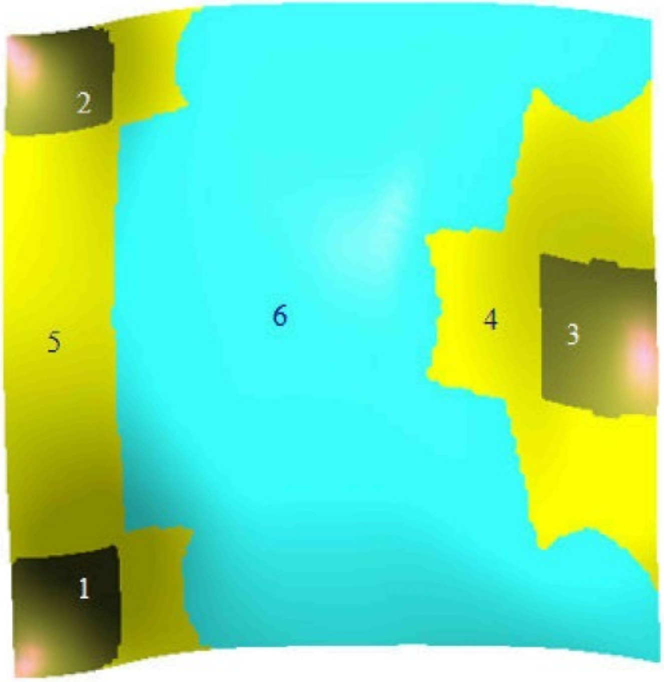


Figure 6.11: Partitioned surface (application2)
1, 2 & 3- concave regions; 4 & 5- saddle regions; 6- convex region.

There are three concave regions on the surface. According to the maximum curvatures measured on the concave regions, they can be only milled by ball-end cutter of 8 mm in diameter or smaller for local gouging avoidance. Meanwhile, the other regions can be milled by much bigger ball-end or flat-end cutters. The machining strategy for this surface can be as follow:

1) Roughing cut in 3-axis mode by a 2-flute flat-end mill of 12 mm in diameter,

2) Finishing cut in 5-axis mode for the convex region by a 4-flute ball-end cutter of 25 mm in diameter,

3) Finishing cut in 5-axis mode for the two saddle regions by a 2-flute ball-end cutter of 18 mm in diameter,

4) Finishing cut in 5-axis mode for the three concave regions by a 2-flute ball-end cutter of 6 mm in diameter.

Assume that the workpiece and tool materials are the same as those in application 1. The machining parameters for the second application are similar to those given in Table 2. These parameters are also used for the case of the un-partitioned surface which is milled by a 6 mm 2-flute ball-end cutter. For determining the tool orientation when machining each region, the values of the lead and tilt angle are shown in Table 6.

Table 6: Lead and tilt angles for each region

Region	**Lead angle (deg)**	**Tilt angle (deg)**
1	-5	-5
2	5	5
3, 4, 5 & 6	7	0

. Figures 6.12 shows an example of the tool paths generated for each region. The machining time estimation and the tool path length of this application are illustrated in Table 7.

With the machining parameters and the tool path strategy above, the tool path length of the proposed method is much shorter (only 64%) compared to the tool path length of the traditional method. However, there is an increase in the finishing time of the proposed method. It takes 50.03 minutes for finishing by the proposed method. Meanwhile, it takes 41.21 minutes for the finishing cut by the traditional method. It means the machining time of the proposed method is about 19% longer than that of the traditional method. This is because when using the big cutters the revolution of the tools and their feed rates become too small to maintain constant cutting speeds as applied for the small ones.

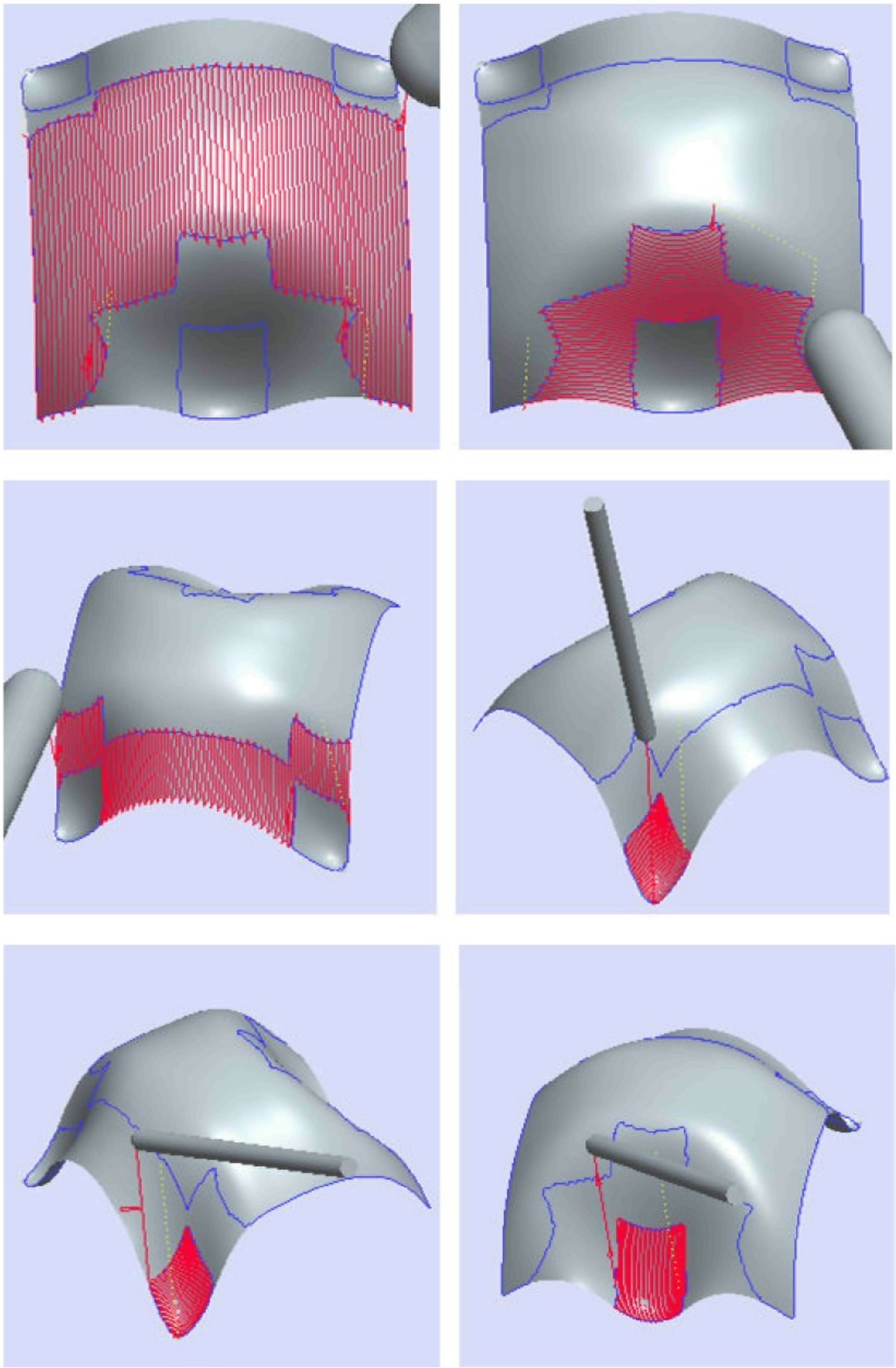


Figure 6.12: Example of tool paths for each region (application 2).

Table 7: Estimated machining time and tool path length comparison (application 2)

Method	**Tool path length (mm)**	**Estimated machining time (min)**
Traditional method (un-partitioned surface)	2000.99 (100%)	42.21 (100%)
Proposed method (partitioned surface)	1282.50 (64.1%)	50.03 (118.5%)

Obviously that the strategy above gives a negative effect on machining time. Therefore it is not good to apply that strategy for the proposed method. There are some strategies that can be applied to get a shorter time for machining the surface above. The following are two possibilities.

a) The strategy above is also used but the convex region is now milled by a 2-flute flat-end cutter of 10 mm in diameter. In this case the tool path length and the estimated machining time of the finishing cut are 968.77 mm and 39.42 minutes, respectively. The finishing time of this strategy is about 7% shorter than that of the traditional method.

b) The surface is partitioned into 4 regions. It consists of 3 concave regions and the last. Here, there is a combination of two saddle regions 4, 5 and convex 6 (Figure 6. 11) into one region. The combined region is milled by an 8 mm 2-flute flat-end cutter. The tool path created for this region is illustrated in Figure 6.13. The result of the machining simulation for the whole surface is illustrated in Figure 6.14. The machining time estimation and the tool path length of this option are illustrated in Table 8.

From Table 8, it can be seen that this option can lead to a big reduction of machining time. The simulation result shows that the tool path length of the proposed method for finishing cut equals about 40% to that of the traditional method. This yields to a big saving up to 34% in finishing time compared to the traditional method. The proposed method takes only 27.68 minutes for finishing cut while the traditional method needs 42.21 minutes to finish the whole surface. Hence, it can be said that with this machining strategy the proposed method is much better than the traditional method in terms of machining time.

Table 8: Estimated machining time and tool path length comparison (recommended strategy, application 2)

Method	Tool path length (mm)	Estimated machining time (min)
Traditional method (un-partitioned surface)	2000.99 (100%)	42.21 (100%)
Proposed method (recommended strategy)	759.21 (40%)	27.68 (65.6%)

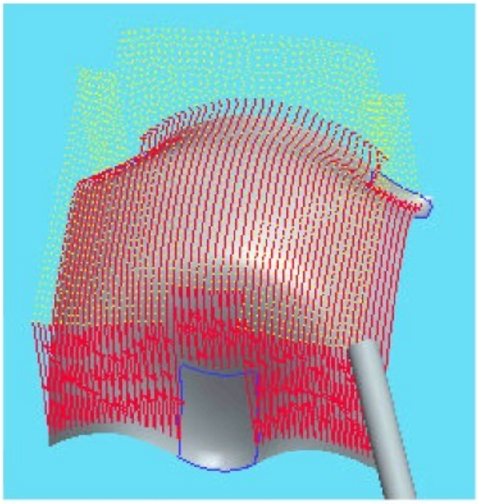


Figure 6.13: Tool paths for the combined region.

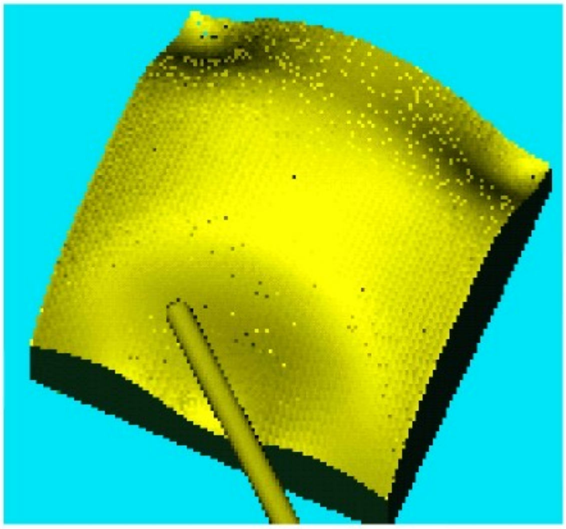


Figure 6.14: Machining simulation (application 2).

6.3.4 Application 3

In this application, the proposed method is applied on a B-spline surface whose control net is given in Table 9 and the uniform knot vectors are [0 1 2 3 4 5 6 7] in *u* direction and [0 1 2 3 4 5 6] in *v* direction. Figure 6.15 shows the Matlab® constructed surface of the design surface. This surface is divided into six regions: one convex region, three saddle region and two concave regions. The CAD model of the design surface with its regions created in Pro/Engineer® is represented in Figure 6.16.

Table 9: Control net of a B-spline surface for application 3

(0,0,0)	(0,25,25)	(0,50,25)	(0,75,0)
(20,0,0)	(20,25,50)	(20, 50,25)	(20,75,50)
(40,0,0)	(40,25,75)	(40,50,50)	(40,75,75)
(60,0,0)	(60,25,50)	(60,50,25)	(60,75,50)
(80,0,0)	(80,25,25)	(80,50,25)	(80,75,0)

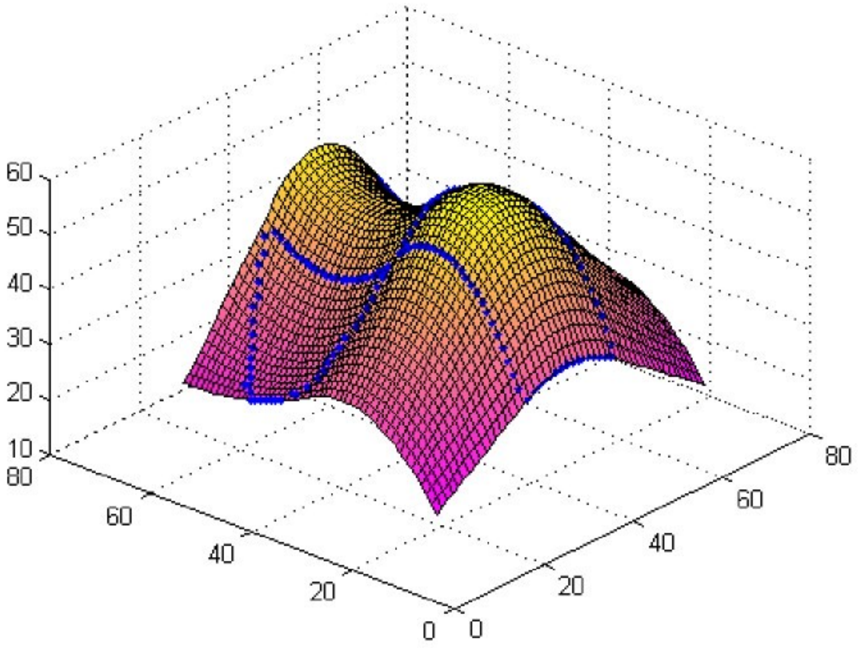


Figure 6.15: Illustration of a freeform surface in Matlab (application 3).

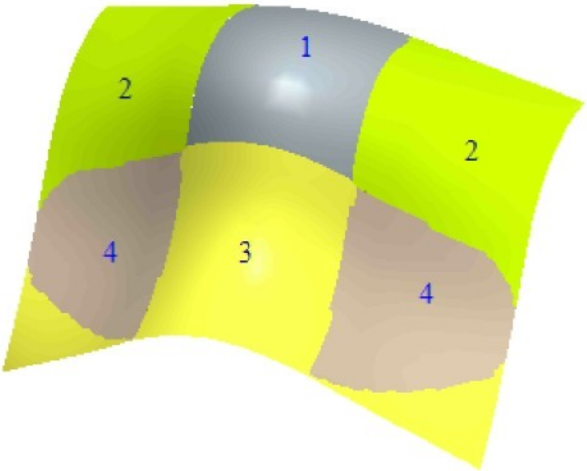


Figure 6.16: Partitioned surface (application 3).
1- convex region, 2 & 3- saddle region; 4- concave region;

Table 10 below shows the values of the maximum curvatures of the concave and saddle regions on the surface. If ball-end cutters are used to machine these regions, the cutter radii can be chosen by referring their probable maximum radii which are calculated from the maximum curvatures, as shown in Table 10.

Table 10: Maximum curvatures K_{max} of regions and probable maximum radii R_{max} of the ball-end cutter used to mill regions

Region	**K_{max}, mm^{-1}**	**R_{max} , mm**
2	0.0181	55.248
3	0.0808	12.376
4	0.0537	18.622

It is impractical that a 110 mm ball-end cutter is used to mill saddle regions 2. Hence, a flat end mill may be a better choice to machine these regions. This tool is also used to machine the convex region. The machining strategy for this surface can be as follow

1) Roughing cut in 3-axis mode by a flat-end mill of 10 mm in diameter,
2) Finishing cut in 5-axis mode for saddle region 3 by a 4-flute ball-end cutter of 24 mm in diameter,

3) Finishing cut in 5-axis mode for the two concave regions by a 4-flute ball-end cutter of 35 mm in diameter,

4) Finishing cut in 5-axis mode for the convex region and saddle regions 2 by a 2-flute end mill of 10 mm in diameter.

Assume that the workpiece and tool materials are the same as those in application 1. The machining parameters of the finishing cut for this application are given in Table 2. The tool postures are defined by the lead and tilt angle which are 5 degrees and 0 degree, respectively. Figure 6.17 shows an example of the tool paths generated for each region. The machining simulation of the operation for this application is presented in Figure 6.18.

Like other applications, the original surface is also planned to machine with the same strategy and parameters by a 4-flute ball-end cutter of 24 mm in diameter. The machining time for this surface is then compared to that of the partitioned one, as shown in Table 11.

Table 11: Estimated machining time and tool path length comparison (application 3)

Method	**Tool path length (mm)**	**Estimated machining time (min)**
Traditional method (un-partitioned surface)	430.78 (100%)	17.07 (100%)
Proposed method (partitioned surface)	303.66 (70.5%)	15.63 (91.5%)

From Table 11, it can be seen that, in this application, the proposed method also has an advantage in terms of machining time. This is because in the finishing stage the tool path length of the proposed method is 29.5% shorter than that of the traditional method. And the result is that it takes 15.63 minutes for finishing region by region of the surface. While, if the whole surface is machined by a ball-end cutter of 24 mm in diameter, the machining time is 17.07 minutes, about 9% longer than the former.

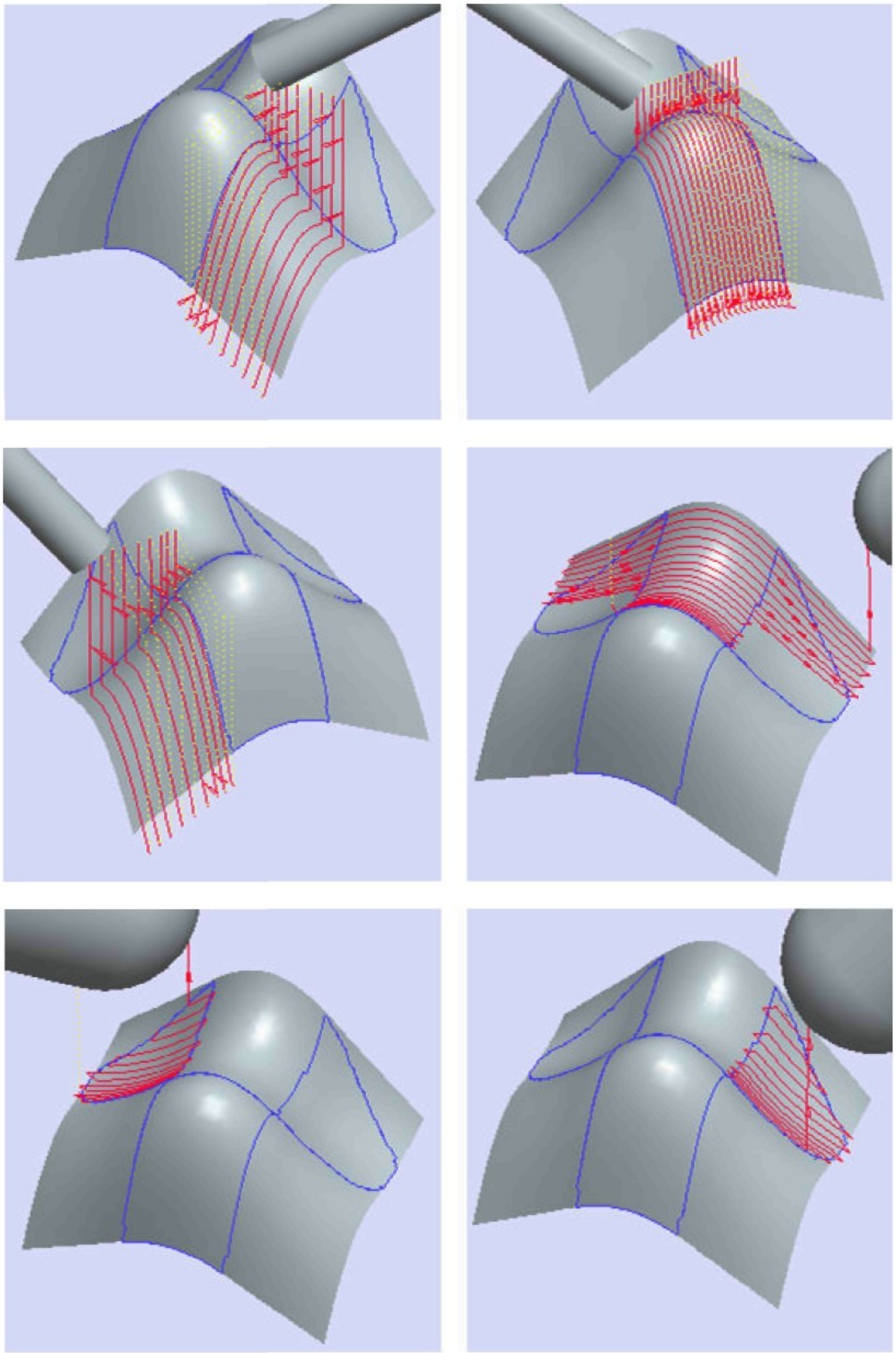


Figure 6.17: Tool paths for each region (application 3).

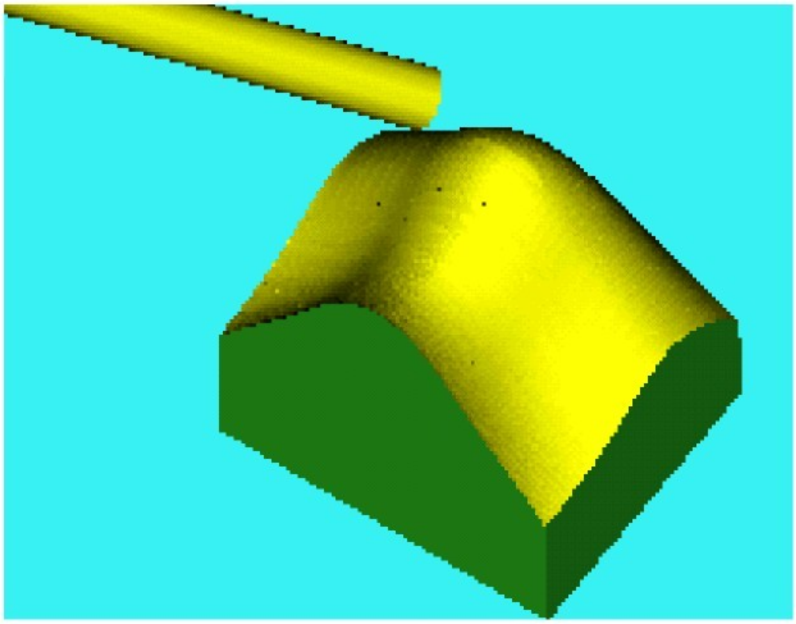


Figure 6.18: Machining simulation (application 3).

According to the distribution of regions 1 and 2 (Figure 6.16), it is not necessary to machine them separately. These regions can be jointed and machined by the same tool as presented above. Because the maximum curvatures of region 3 and 4 (Figure 6.16) are quite high and the difference between them is not much, these regions can also be combined into one. This combined region can be milled by a 24mm 4-flute ball-end cutter. The tool paths for these two large regions are shown in Figure 6.19.

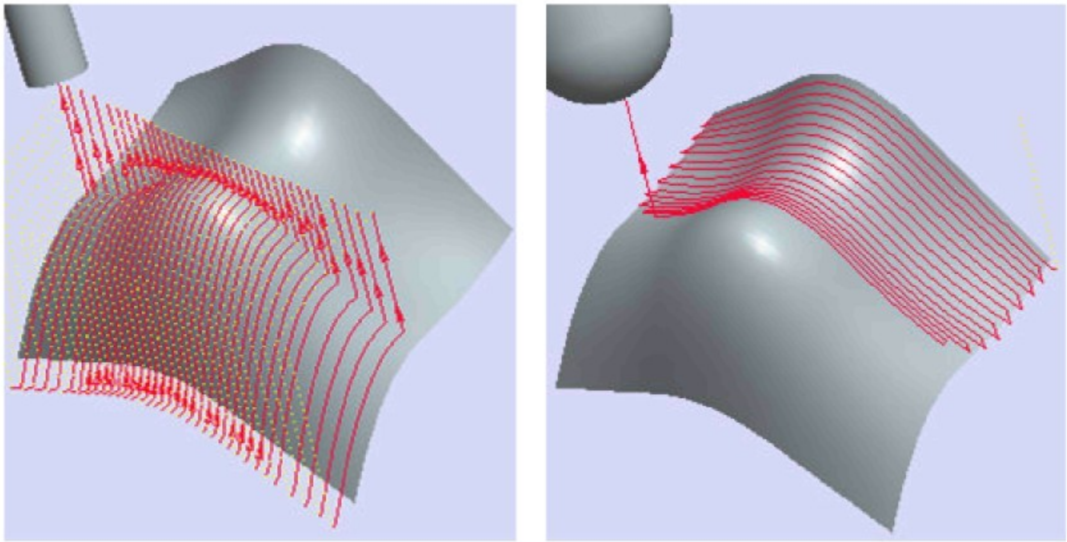


Figure 6.19: Tool paths for the two combined regions.

The calculation result shows that the tool path length for the two combined regions is 305.19 mm and the estimated machining time is about 13.20 minutes. It means by dividing the design surface into 2 regions, the machining time reduces up to 23%, compared to the conventional method.

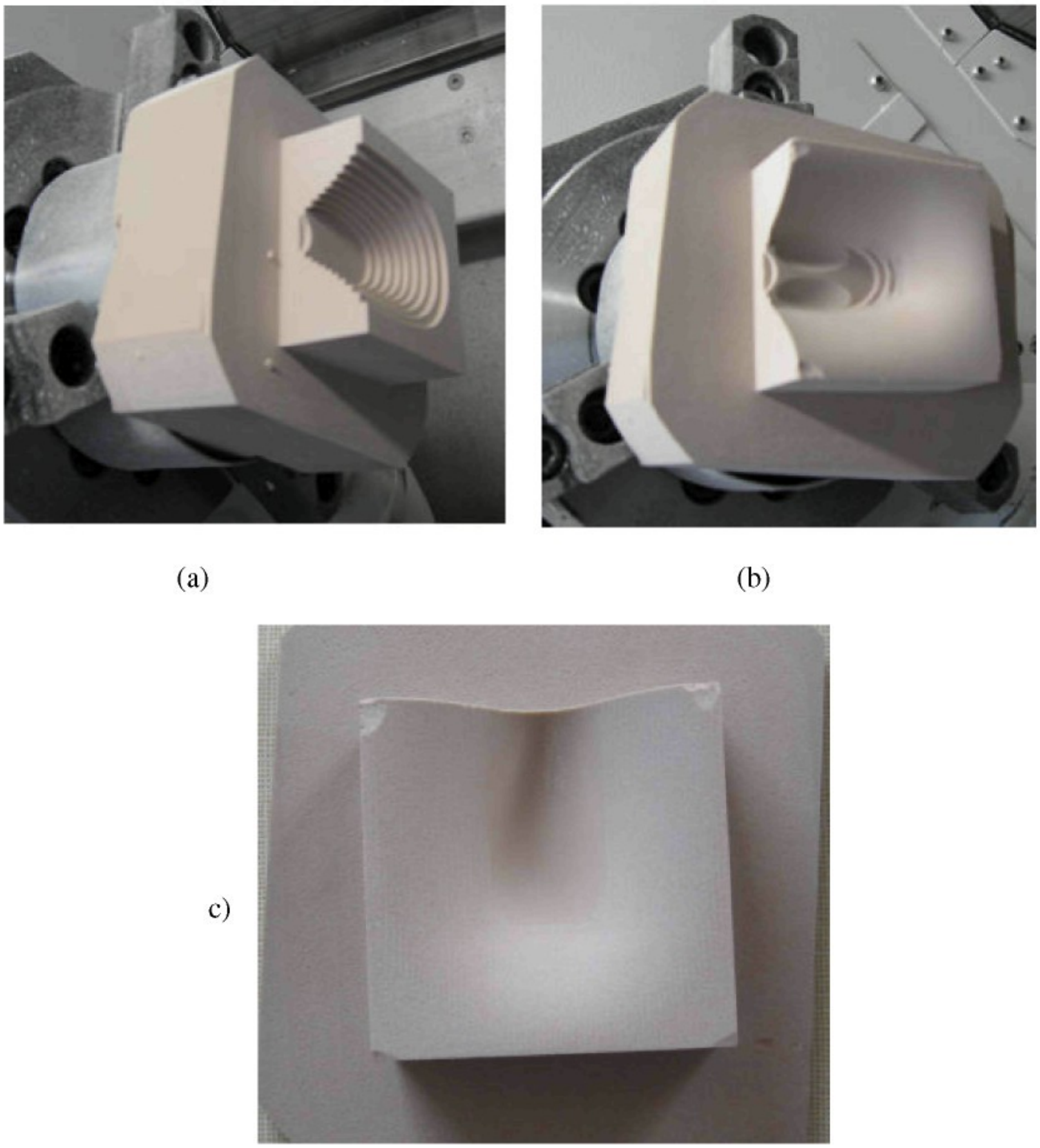
The three applications above and many other cases as well in our research show that in particular situations the machining time can considerably reduce if the freeform surfaces are partitioned in to convex, concave and saddle regions. This is a great potential for machining that kind of surfaces. However, it is realised that sometimes the design surface should not be divided into regions which it has in nature. A smaller number of regions can be a better choice of to get a higher machining efficiency.

6.4 Experiments

The three examples above are typical cases that we have done in the research. From these examples, it can be stated that all algorithms presented in the previous chapters have been verified and the proposed method has been successfully implemented in theory. The implementations above, including surface partitioning, tool path generation, simulation and post processing, are conducted within the Pro/Engineer environment. In order to validate the implementation in practice, some experiments are required. The followings are two samples machined on the Integrex 100-IV, one of a series of multi-tasking machines from Mazak® Corporation. The materials of the workpieces are made from artificial wood.

6.4.1 Experiment 1

Figure 6.20 shows the result of the machining sequences of the surface mentioned in application 1. In this test, the rough cut was done by a 10 mm end mill. The concave and saddle regions were milled by 6 mm and 12 mm ball end cutters, respectively. These tools are available in the CNC laboratory of the department. The scallop height on the surface was set to 0.02 mm.



(a)

(b)

c)

Figure 6.20: Machining sequences of the surface (application 1): (a) part after the roughing cut; (b) part after the finishing cut for the saddle region; (c) final part

6.4.2 Experiment 2

In Figure 6.21 are the results of the machining sequences of the surface in application 2. In this test, the rough cut was done by a 10 mm end mill. The finishing cut for convex, concave and saddle regions was done by a 12 mm ball-end cutter. To see regions easily, the tool paths for each region were planed in different directions from the neighbour ones. The scallop height on the surface was set to 0.05 mm.

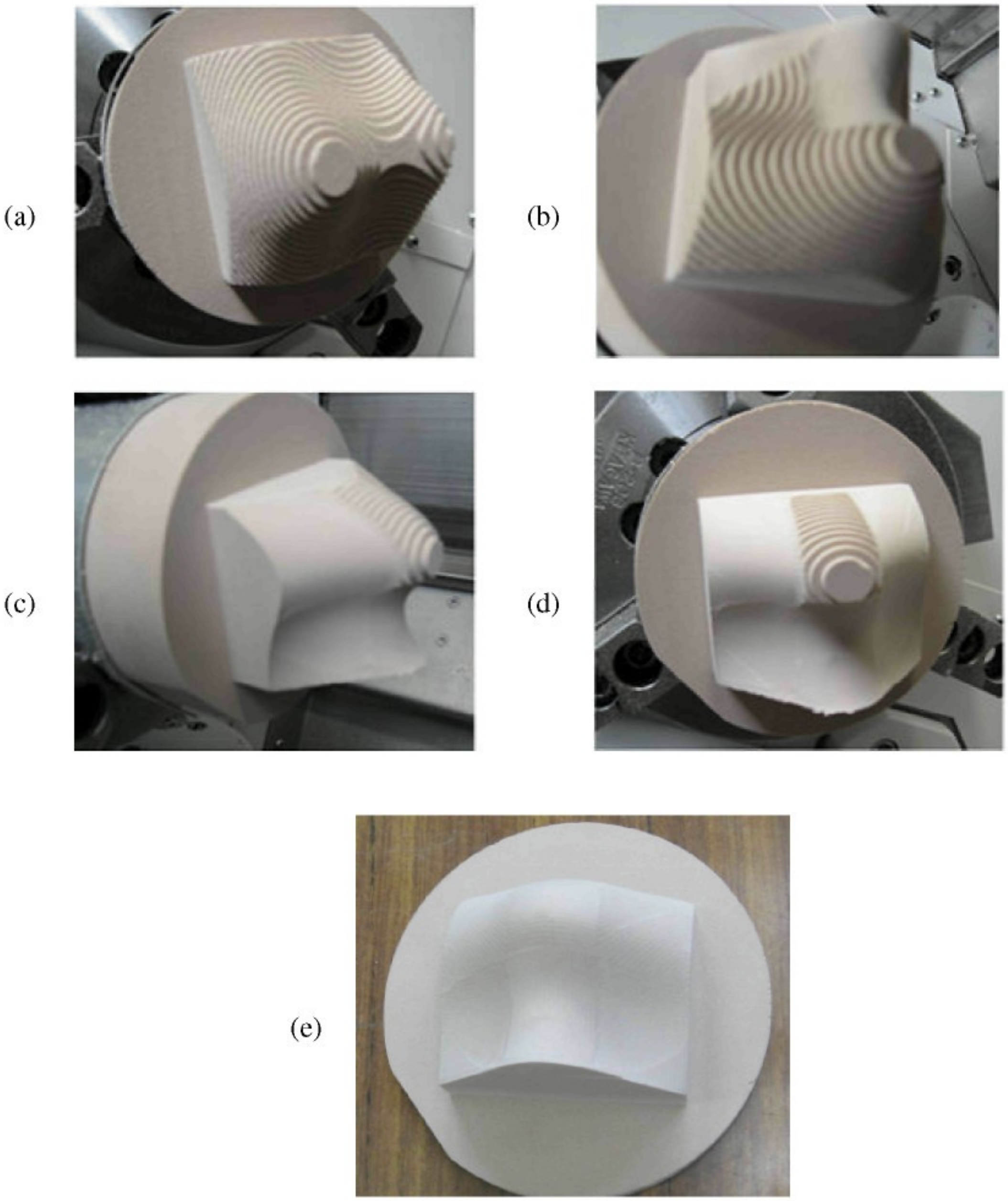


Figure 6.21: Machining sequences of the surface (application 2):
(a) part after the rough cut; (b), (c) & (d) part after some sequences; (e) final part.

In the two tests above the appropriate tools were not used. This was because at that moment other suitable tools were not available at the laboratory. The fact that the tests did not use the same tools and machining parameters, which illustrated in Section 6.3, does not affect the validation of the implementation in practice. The test results still show that the proposed method has been successfully implemented.

6.5 Summary

In this chapter, some useful information about Matlab® function and files, building CAD models of partitioned free-form surfaces is given. The implementations, including surface partitioning, tool path generation, simulation and post processing, are successfully carried out for three different cases of B-spline surfaces. To evaluate the proposed method in terms of machining time, compared to the traditional method, some comparisons have been executed inside the Pro/Engineer environment. Besides, two experiments are also presented to show that the proposed method can be performed in practice. These experiments were done with success on a 5-axis CNC machine.

Chapter 7

CONCLUSIONS

This chapter summarizes the research conducted in this dissertation. Some conclusions of the dissertation are also given. In addition, some synoptic contributions achieved from the research and some recommendations for future work are provided, too.

7.1 Summary and conclusions

Free-form surfaces are widely used in a variety of products in industry. The free-form surface machining process is time-consuming and expensive. Therefore, a reliable and practical machining method that can significantly reduce the machining time may be very useful for applications in industry. In this research, the proposed method in which the free-form surface can be machined region by region with different tools has accomplished this goal.

In the research, the algorithms for surface partitioning and defining the boundaries of regions were developed and implemented successfully. Based on the surface curvatures, a free-form surface can be partitioned into regions whose shapes are convex, concave or saddle. The boundaries of these regions are defined as well. Therefore, the entire surface can be finished separately by several sequences with different tools. The concave and saddle regions can be milled by a small tool and a bigger tool can be applied for the last regions. Compared to the traditional method, the proposed method can considerably decrease the machining time for the finishing cut.

Although there are only few experiments done, but from the test results, it can be stated that the proposed method has been verified successfully. The proposed method can be applied in practice. Like other methodologies, machining free-form surfaces based on partitioning has advantages and disadvantages.

The main advantages of the proposed method are listed below:

1) The approach is practical and it is easy to perform with the support of some popular commercial softwares such as Matlab® and Pro/Engineer®.

2) The machining time can be reduced significantly because the larger cutting tools can be used instead of the small ones to machine the regions on the surface, especially the convex and saddle regions. It means that the machining efficiency can be improved considerably.

3) The surface quality can be improved. This is because the scallop heigh left on the surface by the big cutting tools is smaller than that of produced by the small ones.

4) It is easy to create gouge-free tool paths for regions where the Sturz angle method cannot be applied.

Some main drawbacks of the dissertation can be enumerated as follows:

1) B-spline surfaces were used as examples to illustrate the proposed method, while other kinds of free-form surfaces such as Bézier surface and NURBS surface are not accepted for the Matlab® program developed for this study.

2) To get a high accuracy of the boundaries, the density of the grid points on the design surface should be high. This can cause a long calculation time in Matlab® when large surfaces are involved. It is due to the fact that the algorithms work on the discrete points to calculate the surface properties and so forth.

3) The method for determining the tool orientation when generating the tool paths is tedious and time-consuming in most cases.

7.2 Contributions

This dissertation proposed and developed a new method that allows to machine free-form surfaces based on surface partitioning. In this method, the free-form surface for the CAM programming is a partitioned surface with separate regions. This provides CAD/CAM users the flexibility in choosing the most suitable tool to machine each region. Moreover, the gouge-free tool paths for regions are also easy to achieve. As presented in the dissertation, this method is practical and very workable, and can be applied in industry. This is the primary contribution of the dissertation.

Another important contribution is that the dissertation brought out an effective method for surface partitioning. This method is based on the surface curvatures to divide the free-form surfaces into regions, and uses chain code technique to determine the boundaries of the regions on the surface. Thanks to this combination, it is easy to conduct the task of surface partitioning. The applications in the dissertation show that this method is robust and efficient.

Last but not least, a Matlab® program was developed to fulfill the calculation tasks in the research such as surface partitioning, finding the boundaries of regions on the free-form surfaces, and so forth. Some output parameters of the program are the coordinates of 3D points on the surface to be partitioned and on the boundaries. These output parameters can be easily exported into Excel® and then they can be used as important inputs for CAD purposes in most CAD/CAM softwares.

7.3 Future work

Although this dissertation makes some useful contributions to the field of free-form surface machining, it is necessary to carry out some more work to improve the dissertation. As listed above, the dissertation has some drawbacks that need to be overcome. Besides, the proposed method may be developed to a higher level for an ease of use (to end users). The following are some recommendations for future work.

1) *Adding some other kinds of free-form surfaces such as Bézier and NURBS surfaces*:

   At present, all applications in the research are implemented for B-spline surfaces. In practice, Bézier and NURBS surfaces are also very popular in definition of free-form surfaces. Therefore, it is necessary to expand the ability of the program to diversify applications. This work can be easily carried out by developing the M-function files for creating those kinds of surfaces. These function files will support the written Matlab® program.

2) *Developing an active method for determining the Sturz angle for each region*:

   An algorithm for global gouging detection and correction can be proposed and should be implemented in the early stage. This algorithm will be used to detect the interference between the cutter and the design surface when the tool is machining the concave and saddle regions. Some information from surface partitioning stage can be used for gouging detection. The result of the algorithm will be the value of the Sturz angle that defines the tool posture without gouging. From the value of the Sturz angle of each region, the lead and tilt angles can be defined and they are used as some important input parameters in the CAM stage.

3) *Developing a plug-in integrated in some CAD/CAM packages*

For the CAD/CAM packages that allow implementation of algorithms and macro-programming works, the application of the proposed surface partitioning method would be useful, and can be customized by the end-user for the specific computer aided process planning tasks. The surface partitioning plug-in could therefore be developed and integrated into commercially available packages. In this case, some languages such as C and C++ may be used for programming.

REFERENCES

[1] Bey M., Bendifallah M., Kader S.and Boukhalfa K. Cutting tool combination and machining strategy affectation based on the determination of local shapes for free form surfaces, *International Conference on Smart Manufacturing Application*, 2008, pp 120-125.

[2] Budak E., Ozturk E., and Tunc L.T. Modeling and simulation of 5-axis milling processes, *CIRP Annals - Manufacturing Technology*, 2009, Vol 58, pp 347–350.

[3] Chang T.C., Wysk R.A., and Wang H.P. Computer-aided Manufacturing, Prentice Hall, New Jersey, 1998.

[4] Chen Z.C., Dong Z. and Vickers G.W. Automated surface subdivision and tool path generation for 3½½-axis CNC machining of sculptured parts. *Computers in Industry*, 2003, Vol 50, No 3, pp 319-331.

[5] Chiou C.J. and Lee Y.S. A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Computer-Aided Design*. Vol 34, No 5. pp 357-371.

[6] Choi B.K. and Jerard R.B. Sculptured surface machining: Theory and applications. Springer-Verlag, New York, 1999.

[7] Ding S., Mannan M.A., Poo A.N., Yang D.C.H. and Han Z. Adaptive iso-planar tool path generation for machining of free-form surfaces. Computer-Aided Design, 2003, Vol 35, No 3, pp 141-153.

[8] Dragomatz D. and Mann S. A classified bibliography of literature on nc tool path generation. *Computer-Aided Design*, 1997, Vol 29, No 3.

[9] Elber G. and Cohen E. Second order surface analysis using hybrid symbolic and numeric operators. *Transactions on Graphics*, 1993, Vol 12, No 2, pp 160-178.

[10] Feng H.Y. and Li H. Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer Aided Design*, 2002, Vol 34, No 9, pp 647–654.

[11] Gonzalez C.R., Woods R.E. and Eddins S.L. Digital image processing using matlab. Prentice Hall, 2004.

[12] Gray P., Bedi S. and Ismail F. Rolling ball method for 5-axis surface machining, *Computer-Aided Design* , 2003, Vol 35, No 4, pp 347–357.

[13] Gray P., Bedi S. and Ismail F. Arc-intersection method for 5-axis tool positioning, *Computer-Aided Design*, 2005, Vol 37, No 7, pp 663–674.

[14] Ho M.C., Hwang Y.R., and Hu C.H. Five-axis tool orientation smoothing using quaternion interpolation algorithm, *International Journal of Machine Tools and Manufacture*, 2003, Vol. 43, No. 12, pp 1256-1267.

[15] Ho S., Sarma S. and Adachi Y. Real-time interference analysis between a tool and an environment, *Computer-Aided Design*,2001, Vol 33, No 13, pp 935–947.

[16] Hosseinkhani Y., Akbari J. and Vafaeesefat A. Penetration–elimination method for five-axis CNC machining of sculptured surfaces, *International Journal of Machine Tools and Manufacture*, 2007, Vol 47, No 10, pp 1625–1635.

[17] Ilushin O., Elber G., Halperin D., Wein R. and Kim M.S. Precise global interference detection in multi-axis NC-machining, *Computer-Aided Design*, 2005, Vol 37, No 9, pp 909–920.

[18] Jensen C.G. Analysis & synthesis of multi-axis sculptured surface machining. D.Ph. Thesis, Purdue University, USA,1993.

[19] Jensen C.G., Red W.E. and Pi J. Tool selection for five-axis curvature matched machining. *Computer-Aided Design*, 2002, Vol 34, No 3, pp 251-266.

[20] Lauwers B., Dejonghe P. and Kruth J.P. Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation, *Computer-Aided Design*, 2003, Vol 35, No 5, pp 421–432.

[21] Lee Y.S. Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *Computer-Aided Design*, 1997, Vol 29, No 7, pp 507–521.

[22] Lee Y.S. and Chang T.C. 2-phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design*, 1995, Vol 27, No 10, pp 715–729

[23] Li L. L. and Zhang Y. F. Cutter selection for 5-axis milling based on surface decomposition, *2004 8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 6-9th December 2004, Vol. 3, pp 1863- 1868

[24] Li L.L. and Zhang Y.F. Flat-end cutter accessibility determination in 5-axis milling of sculptured surfaces. *Computer-Aided Design & Applications*, 2005, Vol 2, No 1-4, pp 203-212.

[25] Li L.L. and Zhang Y.F. An integrated approach towards process planning for 5-axis milling of sculptured surfaces based on cutter accessibility map. *Computer-Aided Design & Applications*, 2006, Vol 3, No 1-4, pp 249-258

[26] Lin R.S. and Koren Y. Efficient tool-path planning for machining free-form surfaces. *ASME Journal of Engineering for Industry*, 1996, Vol 118, No 1, pp 20–28.

[27] Lo C.C. Efficient cutter-path planning for 5-axis surface machining with a flat-end cutter, *Computer-Aided Design*, 1999, Vol 31, No 9, pp 557–566

[28] Lo C.C. CNC machine tool surface interpolator for ball-end milling of free-form surfaces. *International Journal of Machine Tools And Manufacture*, 2000, Vol 40, No 3, pp 307–326.

[29] Lu G. In: Leung C.H.C. (Ed.) Visual information systems. Springer, Berlin, 1997, pp 135-150.

[30] Madhavulu G., Sender S.V.N.A., and Ahmed B. CAD/CAM solutions for efficient machining of turbomachinery components by Sturz milling method, *Proceedings of the 1996 IEEE IECON 22nd International Conference*, Vol. 3, pp.1490-1495.

[31] Makhanov S.S. and Anotaipaiboon W. Advanced numerical methods to optimize cutting operations of five-axis milling machines. Berlin: Springer-Verlag, 2007.

[32] McMahon C. and Browne J. CAD/CAM principles, practice and manufacturing management, Prentice Hall, Harlow, England, 1998.

[33] Misra D., Sundararajan V. and Wright P.K. Zig-Zag Tool Path Generation for Sculptured Surface Finishing. *DIMACS Workshop on Computer Aided Design and Manufacturing*, October 7 - 9, 2003

[34] My C.A., Boheza E.L.J., Makhanov S.S., Munlin M., Phien H.N. and Mario T. On 5-axis freeform surface machining optimization: vector field clustering approach. *International Journal of CAD/CAM*, 2005, Vol. 5, No.1.

[35] Radzevich S..P. CAD/CAM of Sculptured surfaces on multi-axis NC machine: The DG/K-based approach. Morgan & Claypool, USA, 2008.

[36] Rao A. and Sarma R. On local gouging in five-axis sculptured surface machining using flat-end tools. *Computer-Aided Design*, 2000, Vol 32, No 7, pp 409–420

[37] Rao N., Ismail F. and Bedi S. Tool path planning for five-axis machining using the principle axis method, *International Journal of Machine Tools and Manufacture*, 1997, Vol 37, No 7, pp 1025–1040.

[38] Ritter G. X. and Wilson J. N. Handbook of computer vision algorithms in image algebra, CRC Press, 1996.

[39] Rogers D.F. An introduction to NURBS with historical perspective. Morgan Kaufmann Publishers, San Francisco, 2001.

[40] Roman A, Bedi S. and Ismail F. Three-half and half-axis patch-by-patch NC machining of sculptured surfaces. *International Journal of Advanced Manufacturing Technology*, 2006, Vol 29, No 5-6, pp 524-531.

[41] Roman A. Surface partitioning for 3+2-axis Machining. D.Phil. Thesis, University of Waterloo, Canada, 2007.

[42] Sonka M., Hlavac V. and Boyle R. Image processing, analysis, and machine vision, Thompson Learning, Toronto, Canada, 2008.

[43] Suresh K. and Yang D.C.H. Constant scallop-height machining of free-form surfaces. *ASME Journal of Engineering for Industry*, 1994, Vol 116, No 2, pp253–259.

[44] Tournier C. and Lartigue C. 5-axis Iso-scallop Tool Paths along Parallel Planes. *Computer-Aided Design and Applications,* 2008, Vol 5, No 1-4, pp 278-286.

[45] Tunc L. T. and Budak E. Extraction of 5-axis milling conditions from CAM data for process simulation, *International Journal of Advanced Manufacturing Technology*, 2008, DOI 10.1007/s00170-008-1735-7

[46] Tournier C. and Duc E. Iso-scallop tool path generation in 5-axis milling. *International Journal of Advanced Manufacturing Technology*, 2005, Vol 25, No 9-10, pp 867-875.

[47] Vickers G.W. and Quan K.W. Ball-mills versus end-mills for curved surface machining. *ASME Journal of Eng. for Industry*, 1999, Vol 111, pp 22–26.

[48] Wang J. Global finish curvature matched machining. M. S. Thesis, Brigham Young University, USA, 2005

[49] Wang Y.J., Zuomin D.Z. and Vickers G.W. A 3D curvature gouge detection and elimination method for 5-axis CNC milling of curved surfaces. *The International Journal of Advanced Manufacturing Technology*, 2007, Vol 33, pp 524-531.

[50] Warkentin A., Hoskins P., Ismail F., and Bedi S. Computer-aided 5-axis machining. In: C.T. Leondes, (Ed.), Computer-aided design, engineering, and manufacturing: Systems techniques and applications. London: CRC Press LLC, 119-152, 2001.

[51] Wang N. and Tang K. Five-axis tool path generation for a flat-end tool based on iso-conic partitioning. *Computer-Aided Design*, 2008, Vol. 40, pp 1067-1079.

[52] Parametric Technology Corporation, Pro/Engineer Wildfire help,.

[53] Planit Holdings Limited, EdgeCAM help.

[54] The Mathworks, Matlab product help.

PUBLICATIONS

[1] Tuong N.V. and Pokorny P. Modeling concave globoidal cam with swinging roller follower: a case study. *Proceeding of World Academy of Science, Engineering and Technology*, 2008, Vol. 32, pp 180-186.

[2] Tuong N.V. and Pokorny P. CAD techniques on modeling of globoidal cam. *Proceeding of 2nd International Conference, Production system - Today and Tomorrow*, TUL, Liberec, 2008.

[3] Tuong N.V. and Pokorny P. Modeling concave globoidal cam with Indexing turret follower: A case study. *International Journal of Computer Integrated Manufacturing*, 2009, Vol. 22, No. 10, pp 940-946.

[4] Tuong N.V. and Pokorny P. A study on making animation and checking interference of globoidal cam. *Modern Machinery (MM) Science Journal*, 2009.

[5] Tuong N.V. and Pokorny P. A case study of modeling concave globoidal cam (book chapter), in: "Advanced Technologies", In-Tech, Austria, 2009.

[6] Tuong N.V., Pokorny P. and Hieu L.C. Free-form surface partitioning for 5-axis CNC milling based on surface curvature and Chain Codes. *The international conference on Innovative Production Machines and Systems (IPROMS),* 2009.

APPENDIX

Appendix 1: Some M-function and script files

```
function Ni = BasisFunction(i,k,t,xknot)
 % This function calculates the basic functions of B-spline surfaces
NiL = N_Left(i,k,t,xknot);
NiR = N_Right(i,k,t,xknot);
Ni = max(NiL,NiR);

%Go from the left
function Nl = N_Left(i,k,t,xknot)
if (k==1)
   if (xknot(i)<=t) & (t<xknot(i+1))
      Nl = 1;
   else
      Nl = 0;
   end
   return;
end

if (xknot(i+k-1)~=xknot(i))
   N1 = (t-xknot(i)).*N_Left(i,k-1,t,xknot)/(xknot(i+k-1)-xknot(i));
else
   N1 = 0;
end
if (xknot(i+k)~=xknot(i+1))
   N2 = (xknot(i+k)-t).*N_Left(i+1,k-1,t,xknot)/(xknot(i+k)-xknot(i+1));
else
   N2 = 0;
end
 Nl = N1+N2;

%Go from the right
function Nr = N_Right(i,k,t,xknot)
if (k==1)
   if (xknot(i)<t) & (t<=xknot(i+1))
      Nr = 1;
   else
      Nr = 0;
   end
   return;
end
if (xknot(i+k-1)~=xknot(i))
   N1 = (t-xknot(i)).*N_Right(i,k-1,t,xknot)/(xknot(i+k-1)-xknot(i));
else
   N1 = 0;
end
if (xknot(i+k)~=xknot(i+1))
   N2 = (xknot(i+k)-t).*N_Right(i+1,k-1,t,xknot)/(xknot(i+k)-xknot(i+1));
else
   N2 = 0;
end
```

```
 Nr = N1+N2;
...........................
function xi = KnotOpenUniform(k,N)
 % This function calculates the knot vector
xi(1:k) = 0;
for i=k+1:N
   xi(i) = (i-k)/(N-k+1);
end
xi(N+1:N+k) = 1;
...........................
function MN = MatrixBasisFunction(knot,N,xknot,t)
% This function create the matrix of the basic function
for i = 1:length(t)
   for j=1:N
      MN(i,j) = BasisFunction(j,knot,t(i),xknot);
   end
end
..............................

function Q = B_Spline(Bij,Nxi,Myj)
 % This is the main function to create the B-spline surface
nx = size(Nxi,1);
ny = size(Myj,1);
Q = zeros(nx,ny,3);
 for ix=1:nx
   for iy=1:ny
      for i=1:size(Nxi,2)
         for j=1:size(Myj,2)
            Q(ix,iy,:) = Q(ix,iy,:)+Nxi(ix,i)*Bij(i,j,:)*Myj(iy,j);
         end
      end
   end
end
.......................

function [K,H,Kmax,Kmin,nx,ny,nz] = surfaceproperties(X,Y,Z)
% This function calculates Gaussian curvature (K), mean curvature (H), principle
% curvatures (Kmax, Kmin), unit normal vectors (n) and their components  (nx, ny, nz)
% at every grid point on a free-form surface.
% X, Y and Z are 2D array of points on the surface.

% Calculate the first derivatives
[Xu,Xv] = gradient(X);
[Yu,Yv] = gradient(Y);
[Zu,Zv]  = gradient(Z);

% Calculate the second derivatives
[Xuu,Xuv] = gradient(Xu);
[Yuu,Yuv] = gradient(Yu);
[Zuu,Zuv]  = gradient(Zu);
 [Xuv,Xvv] = gradient(Xv);
[Yuv,Yvv] = gradient(Yv);
[Zuv,Zvv]  = gradient(Zv);

% Reshape 2D arrays into vectors
```

```
Xu  = Xu(:);   Yu  = Yu(:);   Zu  = Zu(:);
Xv  = Xv(:);   Yv  = Yv(:);   Zv  = Zv(:);
Xuu = Xuu(:);  Yuu = Yuu(:);  Zuu = Zuu(:);
Xuv = Xuv(:);  Yuv = Yuv(:);  Zuv = Zuv(:);
Xvv = Xvv(:);  Yvv = Yvv(:);  Zvv = Zvv(:);

Xu    =  [Xu Yu Zu];
Xv    =  [Xv Yv Zv];
Xuu   =  [Xuu Yuu Zuu];
Xuv   =  [Xuv Yuv Zuv];
Xvv   =  [Xvv Yvv Zvv];

% Calculate the fundamental magnitudes of the first order
E    =  dot(Xu,Xu,2);
F    =  dot(Xu,Xv,2);
G    =  dot(Xv,Xv,2);

% Calculate the unit normal vectors and thier components
m    =  cross(Xu,Xv,2);
p    =  sqrt(dot(m,m,2));
n    =  m./[p p p];
nx   =  n(:,1);
nx   =  reshape(nx,size(X));
ny   =  n(:,2);
ny   =  reshape(ny,size(X));
nz   =  n(:,3);
nz   =  reshape(nz,size(X));

% Calculate the fundamental magnitudes of the second order
L    =  dot(Xuu,n,2);
M    =  dot(Xuv,n,2);
N    =  dot(Xvv,n,2);
[s,t] = size(Z);

% Calculate Gaussian curvatures
K = (L.*N - M.^2)./(E.*G - F.^2);
K = reshape(K,s,t);

% Calculate mean curvatures
H = (E.*N + G.*L - 2.*F.*M)./(2*(E.*G - F.^2));
H = reshape(H,s,t);

% Calculate principal curvatures
Kmax = H + sqrt(H.^2 - K);
Kmin = H - sqrt(H.^2 - K);
.......................................................
function [lcPmax,lcPmin] = MaxMinLocalCurvature(M,gPmax,gPmin)
% This function calculates the critical values of the principal curvatures
Vmax = [];
Vmin = [];
for i = 1:size(M,1)
   for j = 1:size(M,2)
      if (M(i,j)~=0)
         Vmax = [Vmax,gPmax(i,j)];
         Vmin = [Vmin,gPmin(i,j)];
```

```
        end
    end
end

lcPmax = max(Vmax);
lcPmin = min(Vmin);
……………………..
```

M-script file

```
% This is the main M-file that executes the surface partitioning tasks

clear all; clc;

% The vertices of the polygonal control net of the B-spline surface to be created (application 1)
Bij(1,1,:) = [-40,-40,0]; Bij(1,2,:) =[-20,-40,0];  Bij(1,3,:) = [0,-40,0]; Bij(1,4,:) = [20,-40,0];  Bij(1,5,:) = [40,-40,0];
Bij(2,1,:) = [-40,-20,0]; Bij(2,2,:) = [-20,-20,0]; Bij(2,3,:) = [0,-20,0]; Bij(2,4,:) = [20,-20,0];  Bij(2,5,:) = [40,-20,0];
Bij(3,1,:) = [-40,0,0];   Bij(3,2,:) = [-20,0,0];   Bij(3,3,:) = [0,0,-30]; Bij(3,4,:) = [20,0,0];     Bij(3,5,:) = [40,0,0];
Bij(4,1,:) = [-40,20,0];  Bij(4,2,:) = [-20,20,0];  Bij(4,3,:) =[0,20,-40]; Bij(4,4,:) = [20,20,0];   Bij(4,5,:) = [40,20,0];
Bij(5,1,:) = [-40,40,0];  Bij(5,2,:) = [-20,40,0];  Bij(5,3,:) =[0,40,-20]; Bij(5,4,:) = [20,40,0];   Bij(5,5,:) = [40,40,0];

% Calculate the knot vectors of the design surface
kx = 3; ky = 3;
Nx = size(Bij,1);
Ny = size(Bij,2);

yknot =[0:Nx+kx-1];
ty = linspace(ky-1,Ny,70);
xknot =[0:Nx+kx-1];
tx = linspace(kx-1,Nx,50);

Nxi = MatrixBasisFunction(kx,Nx,xknot,tx);
Myj = MatrixBasisFunction(ky,Ny,yknot,ty);

Qt = B_Spline(Bij,Nxi,Myj);

figure; hold off;
surf(Qt(:,:,1),Qt(:,:,2),Qt(:,:,3));
xlabel('x'); ylabel('y'); zlabel('z');
x = Qt(:,:,1); y = Qt(:,:,2); z = Qt(:,:,3);

% Create 3 coordinates of points of the grid
Point3D = [];
for i=1:size(x,1)
    for j=1:size(x,2)
        Point3D = [Point3D; x(i,j),y(i,j),z(i,j)];
   end
end

% Plot points of the grid
%plot3(Point3D(:,1),Point3D(:,2),Point3D(:,3),'*');

% Calculate surface properties from M-function "properties"
M = zeros(size(x));
[K,H,Kmax,Kmin,nx,ny,nz] = properties(x,y,z);

surf(x,y,z); hold on; %3-D shaded surface plot
```

```
Index1_3 = []; %Concave
Index2_4 = []; %Convex
Index5 = [];   %Saddle

for i = 1:size(K,1)
  for j = 1:size(K,2)
    %if (K(i,j)==0) & (H(i,j)==0)
      %M(i,j) = 0;
      %Index0 = [Index0;i,j];
    %end
    if (K(i,j)>=0) & (H(i,j)>0)
      M(i,j) = 1;
      Index1_3 = [Index1_3;i,j];
    end
    if (K(i,j)>=0) & (H(i,j)<=0)
      M(i,j) = 2;
      Index2_4 = [Index2_4;i,j];
    end
    if (K(i,j)<0) & (H(i,j)~=0)
      M(i,j) = 5;
      Index5 = [Index5;i,j];
    end
  end
end

%Points of concave regions
if length(Index1_3) ~= 0
  P1_3 = [];
  norm1_3 = [];
  for i = 1:size(Index1_3,1)
    P1_3 = [P1_3;x(Index1_3(i,1),Index1_3(i,2)),y(Index1_3(i,1),Index1_3(i,2)),...
          z(Index1_3(i,1),Index1_3(i,2))];
    norm1_3 = [norm1_3;nx(Index1_3(i,1),Index1_3(i,2)),ny(Index1_3(i,1),Index1_3(i,2)),...
          nz(Index1_3(i,1),Index1_3(i,2))];
  end
end
%Points of convex regions
if length(Index2_4) ~= 0
  P2_4 = [];
  norm2_4 = [];
  for i = 1:size(Index2_4,1)
    P2_4 = [P2_4;x(Index2_4(i,1),Index2_4(i,2)),y(Index2_4(i,1),Index2_4(i,2)),...
          z(Index2_4(i,1),Index2_4(i,2))];
    norm2_4 = [norm2_4;nx(Index2_4(i,1),Index2_4(i,2)),ny(Index2_4(i,1),Index2_4(i,2)),...
          nz(Index2_4(i,1),Index2_4(i,2))];
  end
end
%Points of saddle regions
if length(Index5) ~= 0
  P5 = [];
  norm5 = [];
  for i = 1:size(Index5,1)
    P5 = [P5;x(Index5(i,1),Index5(i,2)),y(Index5(i,1),Index5(i,2)),z(Index5(i,1),Index5(i,2))];
    norm5 = [norm5;nx(Index5(i,1),Index5(i,2)),ny(Index5(i,1),Index5(i,2)),...
          nz(Index5(i,1),Index5(i,2))];
  end
end

% Find the boundaries of concave regions
M13 = M;
for i = 1:size(M,1)
  for j = 1:size(M,2)
    if M(i,j)~=1
      M13(i,j) = 0;
    end
  end
end

B13 = boundaries(M13);
n13 = length(B13);
% Return points on the boundaries of concave regions
PB13cell = cell(0);
for i=1:n13
  Bn13 = B13{i};
  PBn13 = [];
  for j = 1:size(Bn13,1)
    PBn13 = [PBn13;x(Bn13(j,1),Bn13(j,2)),y(Bn13(j,1),Bn13(j,2)),z(Bn13(j,1),Bn13(j,2))];
  end
  PB13cell = cat(1,PB13cell, PBn13);
end

%Find the boundaries of convex regions
M24 = M;
for i = 1:size(M,1)
  for j = 1:size(M,2)
    if M(i,j)~=2
      M24(i,j) = 0;
    end
  end
end

B24 = boundaries(M24);
n24 = length(B24);
% Return points on the boundaries of convex regions
PB24cell = cell(0);
for i=1:n24
  Bn24 = B24{i};
  PBn24 = [];
  for j = 1:size(Bn24,1)
    PBn24 = [PBn24;x(Bn24(j,1),Bn24(j,2)),y(Bn24(j,1),Bn24(j,2)),z(Bn24(j,1),Bn24(j,2))];
  end
  PB24cell = cat(1,PB24cell, PBn24);
end

%Find the boundaries of saddle regions
M5 = M;
for i = 1:size(M,1)
  for j = 1:size(M,2)
    if M(i,j)~=5
      M5(i,j) = 0;
```

```
        end
    end
end

B5 = boundaries(M5);
n5 = length(B5);
%Return points on the boundaries of saddle regions
PB5cell = cell(0);
for i=1:n5
    Bn5 = B5{i};
    PBn5 = [];
    for j = 1:size(Bn5,1)
        PBn5 = [PBn5;x(Bn5(j,1),Bn5(j,2)),y(Bn5(j,1),Bn5(j,2)),z(Bn5(j,1),Bn5(j,2))];
    end
    PB5cell = cat(1,PB5cell, PBn5);
end

%Display points on boundaries
%figure;hold on;
%for i=1:length(PB5cell)
%PB5 = PB5cell{i};
%plot3(PB5(:,1),PB5(:,2),PB5(:,3),'k.');
%end

for i=1:length(PB13cell)
PB13 = PB13cell{i};
plot3(PB13(:,1),PB13(:,2),PB13(:,3),'b.');
end

for i=1:length(PB24cell)
PB24 = PB24cell{i};
plot3(PB24(:,1),PB24(:,2),PB24(:,3),'b.');
end
% Calculate the maximum and minimum values of principal curvatures
[Kmax13,Kmin13] = MaxMinLocalCurvature(M13,Kmax,Kmin);%Concave regions
[Kmax24,Kmin24] = MaxMinLocalCurvature(M24,Kmax,Kmin);%Convex regions
[Kmax5,Kmin5] = MaxMinLocalCurvature(M5,Kmax,Kmin);%Saddle regions
```