

Matlab-based Calculation Method for Partitioning a Free-form Surface into Regions

Tuong Nguyen Van¹, Natasa Naprstkova²

¹Faculty of Mechanical Engineering, Nha Trang University, 02 Nguyen Dinh Chieu St., Nha Trang City, Vietnam. E-mail: tuongnv@ntu.edu.vn

²Faculty of Mechanical Engineering, University of Jan Evangelista in Ústí nad Labem, Pasteurova 3334/7, 400 01 Usti nad Labem. Czech Republic. E-mail: natasa.naprstkova@ujep.cz

A Matlab-based calculation method which can be used to partition a free-form surface into regions is presented in this paper. Based on surface curvatures and Freeman chain code technique, a free-form surface can be divided into convex, concave and saddle regions. A Matlab program containing some M-function and script files was developed to create the mathematical model of the free-form surface, calculate the surface properties, and find the points on the boundaries of the regions. Two surfaces were given as examples to show that the program runs well and output of the program was used to create the CAD (Computer Aided Design) model of the partitioned surface in Creo Parametric.

Keywords: Surface Partitioning, Free-form Surface, Matlab Programming, Surface Curvature.

1 Introduction

Parts with free-form surfaces, also called sculptured surfaces, have become the reality of series production in industry when CAD systems and modern scanning instruments entered the re-engineering process [1]. This kind of surface is designed to meet aesthetic or functional requirements [2]. Generally, a free-form surface can contain plane, convex, concave, and saddle regions [3, 4]. Free-form surfaces can be machined on 3-axis numerical control (NC) machines using ball-end cutters or on 5-axis NC machines using either ball-end cutters, toroidal cutters or flat-end cutters. Regardless of which machining axis mode is used (3- or 5-axis), the machining productivity will be low if only one cutter is used over the entire surface. This is due to tool size limitations. For a distinct surface, the tool diameter is restricted by a determined value that must not cause local gouging when milling concave or saddle regions. Therefore, if a free-form surface is partitioned by its different regions, separate and suitable tools can be used to machine each region to improve the overall machining efficiency. [4, 5]

Surface partitioning is a method which can be utilized to partition a surface into sub-patches. Some studies performed surface partitioning without using surface curvatures, while others used surface curvatures for surface partitioning. The followings are some studies which utilized surface curvatures as the main parameters to divide a surface into separate regions.

To compute curvature properties, Elber and Cohen [6] developed a hybrid method using symbolic and numeric operators. Giri et al. [7] divided a free-form surface into regions where the boundaries were the loci of extreme curvatures. In the work of Chen et al. [8] the fuzzy C-means method and Voronoi diagram were used to perform the surface subdivision and Fuzzy Logic Toolbox in Matlab was used for programming. The fuzzy C-means method was also selected and implemented for surface partitioning in [9, 10, 11, 12]. In the study of Bey et al. [13], the design surface was first approximated into triangles, then normal vectors and curvatures were calculated

to identify the local shape of every vertex. The vertices were then grouped into distinct regions of convex, concave or saddle. C++ and Open GL were used for implementation. C++ was also used for programming by Liu et al. [12] and Li and Zhang [14, 15] to divide free-form surfaces into regions based on surface curvatures. Li et al. [16] introduced a three-step method for surface partitioning with relevance criteria. They used Matlab to analyze and calculate. A tensor-based method was presented by Liu et al. [17] to divide a free-form surface into sub-regions, considering both the surface geometry and the cutter shape. Visual Studio was used to develop the proposed method.

Although the methods mentioned above were all successfully implemented, some of them were rather complicated and required a high level of calculation and programming for implementation. In the authors' previous work [18], a simpler but effective method for surface partitioning was proposed. This method was also adopted by Shiv et al. [19]. Here, a free-form surface could be partitioned into plane, concave, convex, and saddle regions, based on surface curvatures. The boundaries of each region were defined by using the chain codes method and a Matlab program was developed to do the calculations. This paper presents the Matlab-based calculation method used for free-form surface partitioning. [20, 21]

2 Surface geometry

Given a free-form surface $S(u,v) = [S_x(u,v), S_y(u,v), S_z(u,v)]$. The followings are some geometric parameters of a surface.

The unit normal vector at a point (u,v) can be calculated [13], see (1):

$$n(u,v) = (S_u \times S_v) / |S_u \times S_v| \quad (1)$$

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) at a point on the surface are formulated as [13], see (2) and (3):

$$K = \frac{LN - M^2}{EG - F^2} \quad (2)$$

$$H = \frac{EN - 2FM + GL}{2(EG - F^2)} \quad (3)$$

where E, F, G, L, M, and N are the parameters of the first and second fundamental forms, see (4).

$$E = \frac{\partial S}{\partial u} \frac{\partial S}{\partial u}; F = \frac{\partial S}{\partial u} \frac{\partial S}{\partial v}; G = \frac{\partial S}{\partial v} \frac{\partial S}{\partial v}; L = n \frac{\partial^2 S}{\partial u^2}; M = n \frac{\partial^2 S}{\partial u \partial v}; N = n \frac{\partial^2 S}{\partial v^2} \quad (4)$$

The property of the local surface shape around the point can be divided into four different types as follows [11]:

- (a) $K = 0$ and $H = 0$: local surface shape is planar,
- (b) $K \geq 0$ and $H > 0$: local surface shape is concave,
- (c) $K \geq 0$ and $H < 0$: local surface shape is convex,
- (d) $K < 0$ and $H < 0$ or $H > 0$: local surface shape is saddle.

3 Matlab-based calculation for surface partitioning

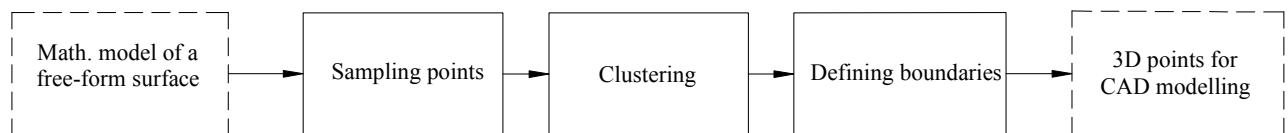


Fig. 1 Flowchart for free-form surface partitioning

Sampling points mean the mathematical model of a free-form surface is first developed. This model can be an explicit equation or a B-spline surface which uses control points as input data. In the first stage, the points with their 3D coordinates on the surface are created to get a set of grid point. The grid is equally spaced in both u and v direction with a distinct density. In the Matlab program, the explicit equation was directly expressed by some commands in a script file. In the case B-spline surface, some M-function files were developed.

In the second stage (Clustering), the points on the surface are clustered into different groups based on their properties. Firstly, Gaussian curvature, and mean curvature are calculated at every grid point on the surface. Secondly, based on the values of K and H , the grid points are stored in a matrix named M . Matrix M is a rectangle array of numbers arranged in rows and columns. Each number in matrix M expresses a point on the surface and is coded by the number 1, 2 or 3 based on whether the point belongs to concave, convex, and saddle regions, respectively. The number of rows and columns in matrix M equals the number of points in both parametric directions. Lastly, to prepare data for the next stage, a special technique is applied to convert matrix M into three matrices named M_1 , M_2 , and M_3 , which contain points on the concave, convex, and saddle regions, respectively.

In this stage (Clustering), two M-function files are used for calculating the surface properties and clustering points into regions. The first M-function file performs the following tasks:

- Calculating the unit normal vectors, based on Equation (1).
- Calculating the fundamental magnitudes of the first order and the second order, based on Equation (4).
- Calculating Gaussian curvatures, based on Equation (2).
- Calculating mean curvatures, based on Equation (3).

The second M-function file codes and stores points with the same properties in matrix M , then converts matrix M into three matrices M_1 , M_2 , and M_3 . These tasks can be performed by applying 2 algorithms as follows:

Algorithm 1: code and store points which have the same properties

```

FOR each grid point  $P_{i,j}$  of the set point  $\{P\}$ 
  IF  $K(i, j) \geq 0$  and  $H(i, j) > 0$ 
    THEN  $M(i, j) = 1$  {code and store points of concave regions}
    Index the location of the corresponding point
  END IF
  IF  $K(i, j) \geq 0$  and  $H(i, j) \leq 0$ 
    THEN  $M(i, j) = 2$  {code and store points of convex regions}
    Index the location of the corresponding point
  END IF

```

```

IF  $K(i, j) < 0$  and  $H(i, j) \neq 0$ 
  THEN  $M(i, j) = 3$  {code and store points of saddle regions}
  Index the location of the corresponding point
END IF
END LOOP

```

Algorithm 2: convert M into $M1$, $M2$, and $M3$

```

Mk =  $M \{k = 1, 2, 3\}$ 
FOR each element of matrix  $M$ 
  IF  $M(i, j) \neq k$  THEN  $Mk(i, j) = 0$ 
  END IF
END LOOP

```

It can be seen that matrices $M1$, $M2$, and $M3$ have the same size as that of matrix M . In matrices $M1$, $M2$, and $M3$, the points on the surface of the same region are expressed by 1 or 2 or 3, and the other points of other regions are presented by 0. With this characteristic, matrices $M1$, $M2$, and $M3$ can be considered as matrices of binary images which have only two possible values for each pixel. In this case, non-zero elements are the pixels of the objects in a binary image and zero elements are the pixels of the background.

In the third stage, the Freeman chain code technique is used to determine the boundary points of the regions on the surface. In image processing, the Freeman chain code can be used to represent object boundaries on a binary image by connecting pixels on the boundaries in straight-line segments [18]. The chain code consists of a sequence of integers which represent the location of the correlative pixels.

Gonzalez et al. [19] presented an M-function for finding the boundaries of objects in a binary image based on the Freeman chain code as follows, see (5):

$$B = \text{boundaries}(f, \text{conn}, \text{dir}) \quad (5)$$

In Function (5), f is a binary image, conn has a value of 4 or 8 (the default) that defines the desired connectivity of the output boundaries, dir defines the direction in which the boundaries are traced in a clockwise (the default) or counter clockwise direction, and output B is a cell array that contains the coordinates of the points on the boundaries. In this study, f represents matrices $M1$, $M2$ and $M3$, the values of conn and dir can be set to the defaults.

By executing Function (5), the number of objects in the image and number of boundary pixels (with indexed row and column coordinates) for the corresponding objects can be obtained. This means that the locations of the points on the region boundaries of the surface can be defined in both parametric directions. However, the 3D coordinates of the boundary points cannot yet be defined. Therefore, in the script file, after the command that executes Function (5), additional commands are needed for further processing to retrieve the 3D coordinates of the boundary points. The boundary points are then saved in cell arrays named PB1, PB2, and PB3 which contain points on the boundaries of concave, convex, and saddle regions, respectively.


4 Application

A number of examples were carried out in our research. In this paper, two typical cases are presented for the purpose of demonstration.

Tab. 1 shows a 5×4 control net of the B-spline surface to be partitioned. The uniform knot vectors in u and v directions are $[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ and $[0, \ 1, \ 2, \ 3, \ 4, \ 5, \ 6]$, respectively.

Tab. 2 Control net of a B-spline surface

(0,0,75)	(0,25,50)	(0,50,100)	(0,75,75)
(25,0,0)	(25,25,50)	(25, 50,25)	(25,75,50)
(50,0,0)	(50,25,75)	(50,50,50)	(50,75,75)
(75,0,0)	(75,25,50)	(75,50,25)	(75,75,50)
(100,0,0)	(100,25,25)	(100,50,25)	(100,75,50)




File Edit View Graphics Debug Desktop

M <10x10 double>

	1	2	3	4	5	6	7	8	9	10
1	3	3	3	3	3	3	3	3	3	3
2	3	3	3	3	3	3	2	2	2	2
3	3	3	3	3	3	2	2	2	2	2
4	3	3	3	1	3	2	2	2	2	2
5	1	1	1	1	1	3	3	3	3	3
6	1	1	1	1	1	3	3	3	3	3
7	1	1	1	1	1	3	3	3	3	3
8	3	3	1	1	3	2	2	2	2	2
9	3	3	3	3	3	3	2	2	2	2
10	3	3	3	3	3	3	2	2	2	2

Fig. 2 Structure of matrix M



File Edit View Graphics Debug Desktop

M2 <10x10 double>

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	2	2	2	2
3	0	0	0	0	0	2	2	2	2	2
4	0	0	0	0	0	2	2	2	2	2
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	2	2	2	2	2
9	0	0	0	0	0	0	2	2	2	2
10	0	0	0	0	0	0	2	2	2	2

Fig. 3 Structure of matrix $M2$

By executing the Matlab program developed in the study, the points on the design surface and the points on the boundaries could be obtained. The points on the design surface were used to modelling the original of the surface, while the boundary points were used to create the boundary curves in CAD environment. To achieve a high accuracy of the boundary curves, a high density of grid points should be performed. However, for a better illustration of the structure of matrices M1, M2, and M3, a grid of points of 10×10 is introduced in this section. The following figures show the main output of the program.

Fig. 2 shows the structure of matrix M (size 10×10). In this matrix, numbers 1, 2, and 3 denote the points which belong to concave, convex, and saddle regions, respectively. It can be seen that there is 1 concave region (18 points), 2 convex regions (27 points) and 1 saddle region (52 points) on the surface. Matrix M was then converted to matrices M1, M2, and M3. Fig. 3 presents the converted structure of matrix M2 as an example.

Variable Editor - PB1			
File Edit View Graphics			
PB1 <13x3 double>			
	1	2	3
6	54.1667	34.7222	57.7589
7	62.5000	34.7222	52.2377
8	70.8333	29.1667	45.9105
9	70.8333	23.6111	43.4757
10	62.5000	18.0556	42.9784
11	62.5000	12.5000	31.2500
12	54.1667	12.5000	34.0278
13	45.8333	12.5000	34.0278

Variable Editor - PB2cell{1,1}			
File Edit View Graphics			
PB2cell{1,1} <12x3 double>			
	1	2	3
5	20.8333	62.5000	50
6	29.1667	62.5000	45.8333
7	37.5000	62.5000	50
8	37.5000	56.9444	45.6790
9	37.5000	51.3889	43.8272
10	37.5000	45.8333	44.4444
11	37.5000	40.2778	47.5309
12	29.1667	40.2778	43.0898

Variable Editor - PB2cell{2,1}			
File Edit View Graphics			
PB2cell{2,1} <12x3 double>			
	1	2	3
5	70.8333	62.5000	43.0556
6	79.1667	62.5000	38.8889
7	87.5000	62.5000	37.5000
8	87.5000	56.9444	32.8704
9	87.5000	51.3889	30.0926
10	87.5000	45.8333	29.1667
11	79.1667	45.8333	32.0988
12	70.8333	40.2778	40.0377

Variable Editor - PB3			
File Edit View Graphics			
PB3 <52x3 double>			
	1	2	3
45	37.5000	34.7222	52.2377
46	29.1667	29.1667	48.0710
47	37.5000	23.6111	50.3858
48	37.5000	18.0556	42.9784
49	37.5000	12.5000	31.2500
50	29.1667	12.5000	29.8611
51	20.8333	12.5000	34.0278
52	12.5000	12.5000	43.7500

Fig. 4 Examples of points on the boundaries of concave, convex, and saddle region

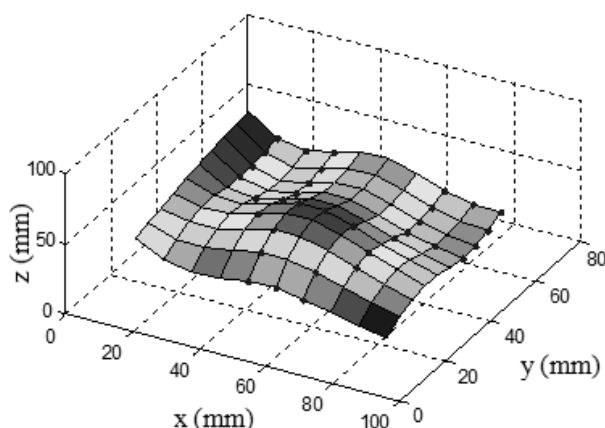


Fig. 5 Matlab figure of the B-spline surface and points on the boundaries of regions

The 3D coordinates of the points of the surface and the boundary points obtained by executing the Matlab program can be saved as Excel files. Once the coordinates are obtained, the boundary curves can be easily created along with the CAD model of the design surface in CAD environment. In this study, Creo Parametric 3.0 was used for modelling the design surface. Fig. 6 shows the CAD model of the original surface with the boundary curves. The input data for CAD modelling were obtained by

One of the most important output sets of the Matlab program is the 3D coordinates of the boundary points. Fig. 4 represents the structure of the cell arrays which contain the 3D coordinates of the boundary points for each region. In these cell arrays, column 1, 2, and 3 indicate the coordinates of x, y, and z of the Cartesian coordinate system, respectively. Cell array PB1 contains 13 points on the boundary of the concave region whereas cell array PB3 has 52 points on the boundary of the saddle region. The two convex regions have 12 points for each boundary which are stored in cell arrays PB2cell{1,1} and PB2cell{2,1}. However, the convex and concave regions share the same boundary with the saddle region. For modelling of a free-form surface in CAD systems, the boundaries of the concave and convex regions can be employed for surface partitioning, while the boundaries of the saddle regions can be ignored. The Matlab figure of the surface with the plotted points on the boundaries is presented in Fig. 5.

sampling points with a grid point of 50×50 in two parametric directions. The boundary curves were used as trim factors to split the design surface into separate regions. The partitioned surface is illustrated in Fig. 7. Once achieving the design surface with separate regions, the tool paths for each region can be generated by different strategies. An example of the tool paths for 5-axis machining with different tools for each partitioned surface is shown in Fig. 8, 9 and 10.

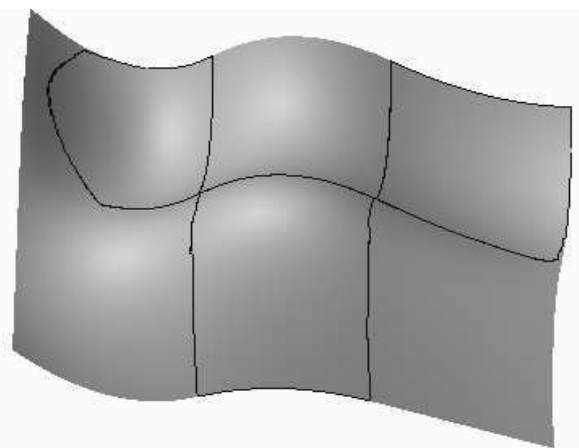


Fig. 6 B-spline surface created in Creo Parametric – original surface and region boundaries

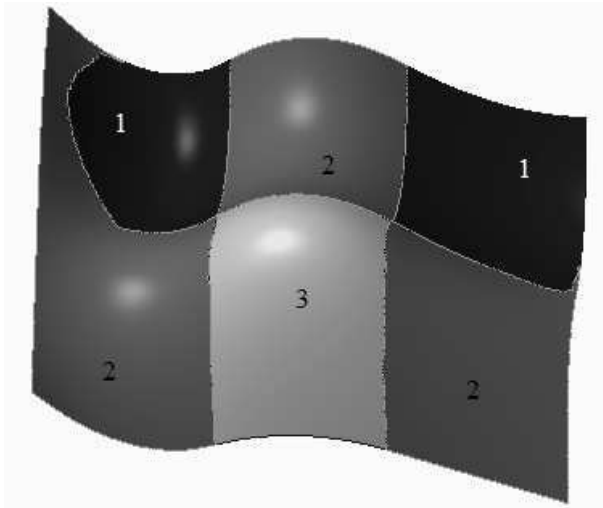


Fig. 7 B-spline surface created in Creo Parametric – partitioned regions: 1- concave region, 2- saddle region, 3- convex region

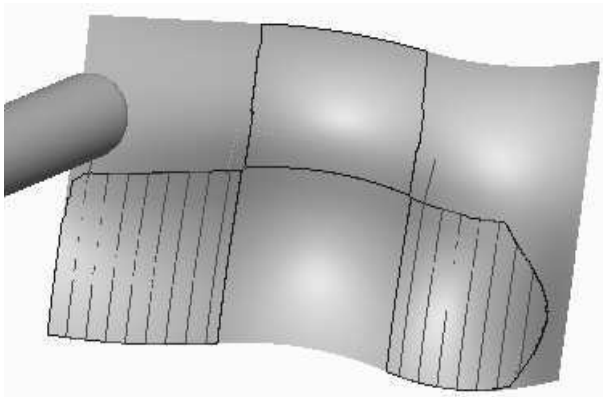


Fig. 8 5-axis tool paths for partitioned surface – milling 2 concave regions by a 12 mm ball-end mill

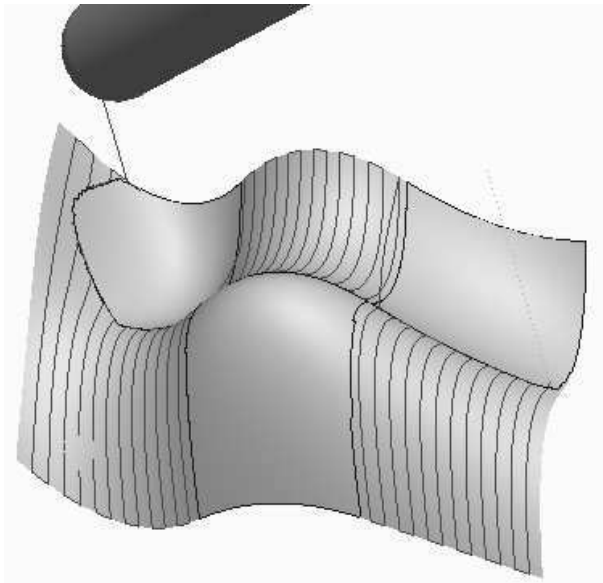


Fig. 9 5-axis tool paths for partitioned surface – milling the saddle region by a 16 mm ball-end mill

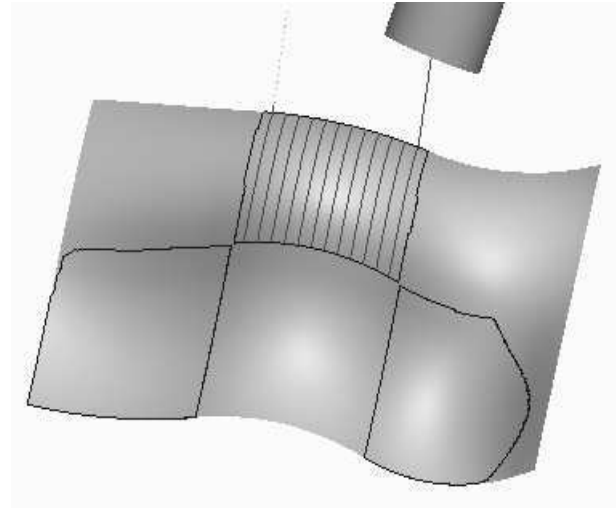


Fig. 10 5-axis tool paths for partitioned surface – milling the convex region by a 12 mm flat-end mill

In this study, the calculation for surface partitioning was performed with Matlab 2011b. The Matlab program developed in this research was run on a notebook (Intel Core i5-3337U, 1.8 GHz CPU, 4GB of RAM), operating under Microsoft Windows 10. When the grid of the design surface was set to 10x10, the running time of the program was 0.51 seconds. When the grid was set to 50x50 and 200x200, the executive time of the program was 0.81 seconds and 11.75 seconds, respectively. The higher the density of the grid, the longer the running time. Although running time did not present a problem in the current research. It may be of concern if a larger, more complicated surface is introduced.

Fig. 11 illustrates the Matlab figure of a surface expressed by function (6). This surface has 1 convex region, 1 concave region and 1 saddle region. Its CAD model is shown in Fig. 12 and 13.

$$z = -e^{-[(x-y^2)^2+y^2]} \quad (6)$$

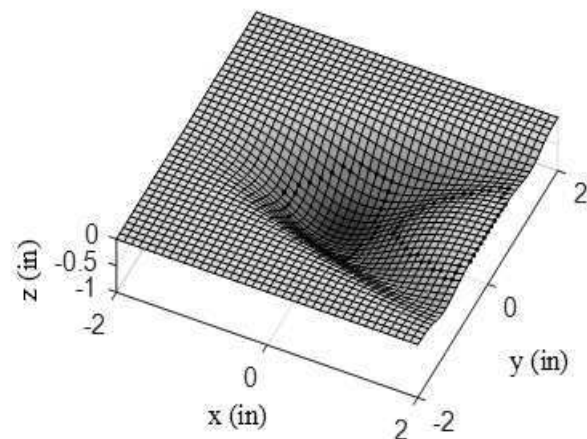


Fig. 11 Matlab figure of surface $z = -e^{-[(x-y^2)^2+y^2]}$ and points on the boundaries

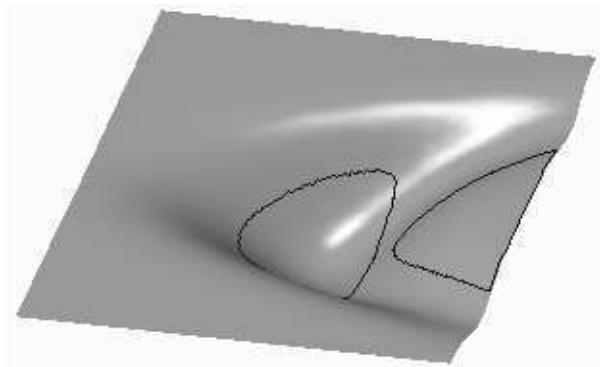


Fig. 12 Surface $z = -e^{-(x-y^2)^2+y^2}$ created in Creo Parametric – original surface and region boundaries

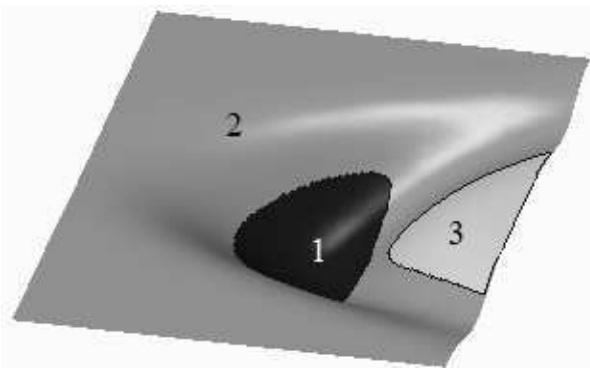


Fig. 13 Surface $z = -e^{-(x-y^2)^2+y^2}$ created in Creo Parametric – partitioned regions, 1- concave region, 2- saddle region, 3- convex region

From these applications, it can be realized that the proposed Matlab program can be successfully implemented to obtain points on the boundaries of the regions of the design surface. These points can be used for surface partitioning purposes in Creo Parametric. This program can be expanded to apply to other kinds of free-form surfaces such as Bezier and NURBS if M-function files for creating a mathematical models of those kinds of surfaces are developed. However, in practice, the free-form surface of a sculptured part may consist of many patches, and each patch can be represented in any form. If the Matlab program can handle a composite surface, this could lead to more realistic applications and thereby applications that are more meaningful in practice.

5 Conclusion

In this paper, a Matlab program was developed to do the calculating task for free-form surface partitioning. In the program, two special algorithms were used to cluster points on the surface into different groups based on surface curvatures, and make the matrices of the clustered points available to use the Freeman chain code technique in the image processing field. Hence, the points on the region boundaries can be determined. The developed program can be used effectively to partition a free-form surface in the form of an explicit equation or a B-spline surface into convex, concave, and saddle regions. The output data of the program can be used in Creo Parametric or

other CAD packages to create the CAD model of the surface with separate regions. Developing the Matlab program which can divide a composite free-form surface into regions is a consideration for future work.

Acknowledgement

The work of this paper is supported by Nha Trang University, Vietnam and project No. CZ.1.05/4.1.00/11.0260 EDIMARE.

References

- [1] DOVICA, M., BUSA J., SPISAK, E., FABIAN, M., IZOL, P., FABIANOVA, J. (2016). Assessment of complex free form surfaces with surface profile deviation. In: *Manufacturing Technology*, Vol. 16, No. 4, pp. 673-680. FME JEPU. Czech Republic.
- [2] SHIV, J.K., GAIN, S.K., SAHU, L.K. (2016). Partitioning of a freeform surface for machining with optimized tool path. In: *International Journal of Innovative Science, Engineering & Technology*, Vol. 3, No. 7, pp. 10-23. IJSET. India.
- [3] ROGERS, D.F. (2001). *An introduction to NURBS with historical perspective*. Morgan Kaufmann Publishers, San Francisco.
- [4] DOVICA, M., BUŠA, J., SPISAK, E., FABIAN, M., IZOL, I., FABIANOVA, J. (2016) Assessment of Complex Free Form Surfaces with Surface Profile Deviation. In: *Manufacturing Technology*, Vol. 16, No. 4, pp. 673-680. FME JEPU. Czech Republic.
- [5] XU, J., LUO, S., CHEN, S.X. (2016) Numerical Control Simulation and Experimental Study on End Milling Ordinary Cycloidal Gear in Five-axis CNC Machine Tool. In: *Manufacturing Technology*, Vol. 16, No. 6, pp. 1388-1393. FME JEPU. Czech Republic.
- [6] ELBER, G., COHEN, E. (1993). Second order surface analysis using hybrid symbolic and numeric operators. In: *Transactions on Graphics*, Vol. 12, No. 2, pp. 160-178. ACM. China.
- [7] GIRI, V., BEZBARUAH, D., BUBNA, P., CHOUDHURY, A.R. (2005). Selection of master cutter paths in sculptured surface machining by employing curvature principle. In: *International Journal of Machine Tools and Manufacture*, Vol. 45, No. 10, pp. 1202–1209. Elsevier. Nederland.
- [8] CHEN, Z.C., DONG, Z., VICKERS, G.W. (2003). Automated surface subdivision and tool path generation for 3½-axis CNC machining of sculptured parts. In: *Computers in Industry*, Vol. 50, No. 3, pp. 319-331. Elsevier. Nederland.
- [9] ROMAN, A, BEDI, S., ISMAIL, F. (2006). Three-half and half-axis patch-by-patch NC machining of sculptured surfaces. In: *International Journal of*

- Advanced Manufacturing Technology*, Vol. 29, No. 5-6, pp. 524-531. Springer. Germany.
- [10] ROMAN, A., BAROCIO, E., HUEGEL, J.C., BEDI, S. (2015). Rolling ball method applied to 3½-axis machining for tool orientation and positioning and path planning. In: *Advances in Mechanical Engineering*, Vol. 7, No.12, pp. 1-12. SAGE Publishing. UK.
- [11] YOU, S.K., LUI, X.C., DENG, H.G., LIU, J. (2012). A novel 5-axis machining method for sculptured surface. In: *Advanced Materials Research*, Vols. 562-564, pp. 855-860. TTP. Switzerland.
- [12] LIU, Y., ZHAO, G., ZAVALNYI, O., XIAO, W. (2019). Tool path generation for partition machining of T-spline surface based on local refinement. In: *International Journal of Advanced Manufacturing Technology*, February 2019, pp. 1-14. SJR. Germany.
- [13] BEY, M., BENDIFALLAH, M., KADER, S., BOUKHALFA, K. (2008). Cutting tool combination and machining strategy affectation based on the determination of local shapes for free form surfaces. In: *Proceeding of International Conference on Smart Manufacturing Application*, pp. 120-125. Goyang-Si, South Korea.
- [14] LI, L.L., ZHANG, Y.F. (2004). Cutter selection for 5-axis milling based on surface decomposition. In: *8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, Vol. 3, pp. 1863- 1868. China.
- [15] LI, L.L., ZHANG, Y.F. (2005). Flat-end cutter accessibility determination in 5-axis milling of sculptured surfaces. In: *Computer-Aided Design & Applications*, Vol. 2, No. 1-4, pp. 203-212. Taylor & Francis, England.
- [16] LI, L., CHEN, B., LIU, F., LI, C. (2014). Complexity analysis and calculation for sculptured surface in multi-axis CNC machining based on surface subdivision. In: *International Journal of Advanced Manufacturing Technology*, Vol. 71, pp.1433-1444. Springer. Germany.
- [17] LIU, X., LI, Y., MA, S., LEE, C.H. (2015). A tool path generation method for freeform surface machining by introducing the tensor property of machining strip width. In: *Computer-Aided Design*, Vol. 66, Issue C, pp. 1-13. Elsevier. Nederland.
- [18] TUONG, N.V., POKORNY P. (2010). A practical approach for partitioning free-form surfaces. In: *International Journal of Computer Integrated Manufacturing*, Vol. 23, No. 11, pp 992-1001. Taylor & Francis, England.
- [19] SHIV, J.K., GAIN, S.K., SAHU, L.K. (2016). Partitioning of a Freeform Surface for Machining with Optimized Tool Path. In: *International Journal of Innovative Science, Engineering & Technology*, Vol. 3, No. 7, pp. 10-23. IJISSET. India.
- [20] SONKA, M., HLAVAC, V., BOYLE, R. (2008). *Image processing, analysis, and machine vision*. Thompson Learning, Toronto, Canada.
- [21] GONZALEZ, C.R., WOODS, R.E., EDDINS, S.L. (2004) *Digital image processing using Matlab*. Pearson Prentice Hall. New York. USA.