

**GENERATING EFFICIENT MILLING TOOL PATHS
ACCORDING TO A PREFERRED FEED DIRECTION FIELD**

by

Guillermo Hiroki Kumazawa

B. Sc., Instituto Tecnológico y de Estudios Superiores de Monterrey, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

November 2012

© Guillermo Hiroki Kumazawa, 2012

Abstract

A new method is presented to generate ball-end milling tool paths for the efficient three-axis machining of sculptured surfaces. The fundamental principle of the presented method is to generate the tool paths according to a preferred feed direction (PFD) field derived from the surface to be machined. In this work, the PFD at any point on the surface is the feed direction that maximizes the machining strip width. Theoretically, tool paths that always follow the direction of maximum machining strip width at each cutter contact point on the surface would maximize material removal, which leads to the shortest overall tool path length.

Scallops are generated when a surface is machined using three-axis ball-end mills. There is no redundant machining if the scallop height is always maximized and the neighboring machining strips do not overlap. Unfortunately, these overlaps commonly exist for tool paths always following the preferred directions. Such redundant machining can be reduced via iso-scallop tool paths. Nonetheless, iso-scallop tool paths do not in general follow the preferred feed directions.

To attain maximum machining efficiency via generating the shortest overall tool path length, the presented method analyzes the PFD field of the surface and segments the surface into distinct regions with similar PFD's by identifying the degenerate points and generating their separatrices. The tool paths of each region are generated by the iso-scallop method to mitigate redundant machining. Since a sequential approach is employed to generate the iso-scallop tool paths, an initial tool path is selected in such a way that the growing deviations of the subsequent tool paths from the PFD's are not significant. The proposed method has been validated with numerous case studies, showing that the generated tool paths have a shorter overall length compared with those generated by the existing methods.

Table of Contents

Abstract.....	ii
Table of Contents	iii
List of Tables	iv
List of Figures	v
Nomenclature	viii
Acknowledgements.....	ix
Dedication	x
1. Introduction.....	1
1.1. Conventional Tool Path Generation Methods	3
1.2. Surface Topology-Based Methods	6
2. Preferred Feed Direction.....	9
2.1. Machining Strip Width.....	9
2.2. Preferred Feed Direction Field.....	16
3. Iso-scallop Tool Path Generation.....	21
4. Methodology.....	25
4.1. Degenerate Points.....	27
4.1.1. Tensor Field Representation	27
4.1.2. PFD Pathlines	30
4.1.3. Detection of Degenerate Points	38
4.2. Segmentation	45
4.2.1. Trisector Points.....	45
4.2.1. Wedge Points	48
4.2.3 Merged Wedge Points	49
4.3. Principal Tool Path Evaluation	54
4.3.1. Iso-scallop Tool Path Drift	54
4.3.2. Principal Tool Path Evaluation Procedure.....	55
5. Case Studies and Results Comparison.....	59
6. Conclusions.....	87
References.....	89
Appendix.....	92
Appendix A Bicycle CAD Model Renderings	92

List of Tables

Table 1	Results comparison with proposed method, $r = 10$ (mm), $h = 0.2$ (mm).....	86
----------------	---	----

List of Figures

Figure 1	Illustration of the side-steps when a) there is no redundant machining and b) there is redundant machining.	3
Figure 2	Geometry of the Swept Profile and Machining Strip Width.	15
Figure 3	Example of a Preferred Feed Direction Field.	18
Figure 4	Graph comparing W vs ϕ on a CC point.	19
Figure 5	Tool paths with maximum scallop height following the direction of a) maximum strip width W_1 and b) minimum strip width W_2	20
Figure 6	Geometry of the iso-scallop tool path generation method.	24
Figure 7	Flowchart of the proposed method.	26
Figure 8	Rectangular grid with tensor components.	29
Figure 9	Illustration of the bilinear interpolation method.	33
Figure 10	Comparison between pathlines (step = 1 mm) created by a) the tensor-bilinear method (no extension beyond grid) and b) using the step-marching method.	34
Figure 11	Sample surface covered with pathlines.	35
Figure 12	Illustration of overlaps of sample pathlines' machining strip width.	36
Figure 13	Comparison between the iso-scallop tool paths generated from one of the sample design surface pathlines and its PFD field pathlines.	37
Figure 14	Comparison between a design surface's PFD pathlines and the difference between W_{max} and W_{min} at sample points.	41
Figure 15	Illustration of a Trisector Degenerate Point.	42
Figure 16	Illustration of a Wedge Degenerate Point.	43
Figure 17	Illustration of a Merged Wedge Degenerate Point.	44
Figure 18	Evaluation of Separatrices in a Trisector Point.	47
Figure 19	Evaluation of Separatrices in a Wedge Point.	51
Figure 20	Illustration of the surrounding pathlines of a Merged Wedge Point.	52
Figure 21	Illustration of a segmented surface and its degenerate points.	53
Figure 22	Illustration of the Hausdorff distance d_H in a) pair of open curves, b) pair of closed curves.	57
Figure 23	Illustration of the Principal Tool Path.	57

Figure 24 Iso-scallop tool paths compared with pathlines using a) the region border and b) the Principal Tool Path	58
Figure 25 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.	62
Figure 26 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.	63
Figure 27 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.	64
Figure 28 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).....	65
Figure 29 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.	66
Figure 30 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.	67
Figure 31 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.	68
Figure 32 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).....	69
Figure 33 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.	70
Figure 34 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods	71
Figure 35 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.	72
Figure 36 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).....	73
Figure 37 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.	74
Figure 38 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.	75
Figure 39 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.	76
Figure 40 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).....	77
Figure 41 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.	78

Figure 42 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.	79
Figure 43 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.	80
Figure 44 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).....	81
Figure 45 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), proposed method.....	82
Figure 46 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.	83
Figure 47 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.....	84
Figure 48 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).	85

Nomenclature

$\mathbf{C}(t)$	Parametric expression of a curve.
$\tilde{\mathbf{C}}(t)$	Projection of a parametric curve $\mathbf{C}(t)$ on the x - y plane.
CAD	Computer aided design.
CAM	Computer aided manufacturing.
CC	Cutter contact.
CL	Cutter location.
CNC	Computer numerical control.
d_H	Hausdorff distance.
\mathbf{e}	Eigenvector.
\mathbf{F}	Pathline.
\mathbf{f}	Feed vector.
h	Scallop height.
IGES	Initial Graphics Exchange Specification.
\mathbf{n}	Normal vector.
P_{CC}	Cutter contact point.
P_{CL}	Cutter location point.
PFD	Preferred feed direction.
r	Ball end radius.
$\mathbf{S}(u, v)$	Parametric expression of a surface.
SC	Scallop curve.
\mathbf{T}	Tensor.
W	Machining strip width.
u, v	Variables in the parametric domain.
\mathbf{v}	Vector.
x, y, z	Three-dimensional variables in the Cartesian domain.
ϕ	Feed angle.
λ	Eigenvalue.
θ	Angle.

Acknowledgements

I offer my gratitude to my supervisor, Dr. Hsi-Yung (Steve) Feng, for his diligent support and supervision throughout the course of my studies. This work would not have been completed without his deep insight, knowledge, and expertise.

I would also like to thank my colleagues of the UBC CAD/CAM/CAI Research Laboratory for their support and help during the course of my studies.

My deepest thanks go to my family, who provided me with morale and spirit to continue with my studies despite the distance.

This work was funded in part by the Foreign Affairs and International Trade Canada (DFAIT), the Consejo Nacional de Ciencia y Tecnología, México (CONACYT), and the Natural Sciences and Engineering Research Council of Canada (NSERC). This work wishes to acknowledge the website GRABCAD (<http://grabcad.com/>) and the CAD model author Dusan Tijanac for permitting the use of the bicycle seat CAD model for academic purposes.

Dedication

To my parents

1. Introduction

Sculptured surfaces have seen applications in many fields, especially in the aerospace, automotive, electronic, medical, and even in the artistic industry. These surfaces are characterized by their smooth shape and may include features such as valleys, mounts, or blends to fulfill requirements such as an optimized airflow or a desirable ergonomic shape. Many computer aided design (CAD) programs contain built-in functions that allow for the modeling and simulation of sculptured surfaces in the form of non-uniform rational b-splines (NURBS), allowing the sculptured surfaces to gain more acceptance in the industry. In order to manufacture components with sculptured surfaces, 3-axis or 5-axis computer numerical control (CNC) mills with a variety of tool ends are used to machine either the desired part itself or their corresponding mold. Although 5-axis CNC machines provide a higher versatility and machining speed compared to their 3-axis counterparts, the latter are still widely used due to their lower costs, simplicity, and widespread availability.

When machining a design surface on a workpiece with a 3-axis mill, scallops, also known as cusps, will inevitably be left between tool paths. The scallop height at each point of the scallop curve depends on the adjacent distance between tool paths. A smaller scallop height requires a shorter side-step between adjacent tool paths and consequently, a larger amount of tool paths are required to cover the entire surface that add up to a longer overall machining tool path length. Tool path overlapping occurs when the side-step chosen creates a point on a scallop curve with a height that is shorter than the specified tolerance. The effects of redundant tool paths are illustrated in Fig. 1, where h is the scallop height requirement. It can be observed that overlapping tool paths have a smaller side-step than a part where the maximum scallop height is always kept. A smaller side-step leads to a larger amount of machining passes.

The finishing stages of a machining operation take the longest time because the specified profile tolerance for a finished surface is frequently very tight so that a smoother surface can be obtained. This results in a large number of machining passes. A constant feed rate is commonly used during the finishing stages and therefore a longer tool path length will inevitably lead to a longer machining time. Ways to decrease this overall tool path length has been a topic of continuous research in the field of computer aided manufacturing (CAM). Methods would be proposed to mathematically determine the machining tool paths needed to obtain the desired sculptured shapes under the specified scallop height tolerance. Earlier methods would require the user to input information for an initial tool path from which the rest of the overall tool paths would be generated on the surface. These methods will be considered in this work as conventional tool path generation methods. The conditions of the initial tool path would define the rest of the machining passes and therefore, even one method for creating the machining strategy of a surface may have an infinite number of possible overall toolpaths. Subsequent research would seek to find the global machining strategy that can optimize the total tool path length. These strategies analyze the input surface with the goal of finding the unique and ideal tool path that minimizes the machining length. The main objective of these methods is to generate tool paths with as little overlapping tool paths as possible, without the need of the input of initial conditions of machining. Ideally, one optimal overall tool path will be the best suited for a particular input surface. In this work, these methods will be known as surface topology-based methods.

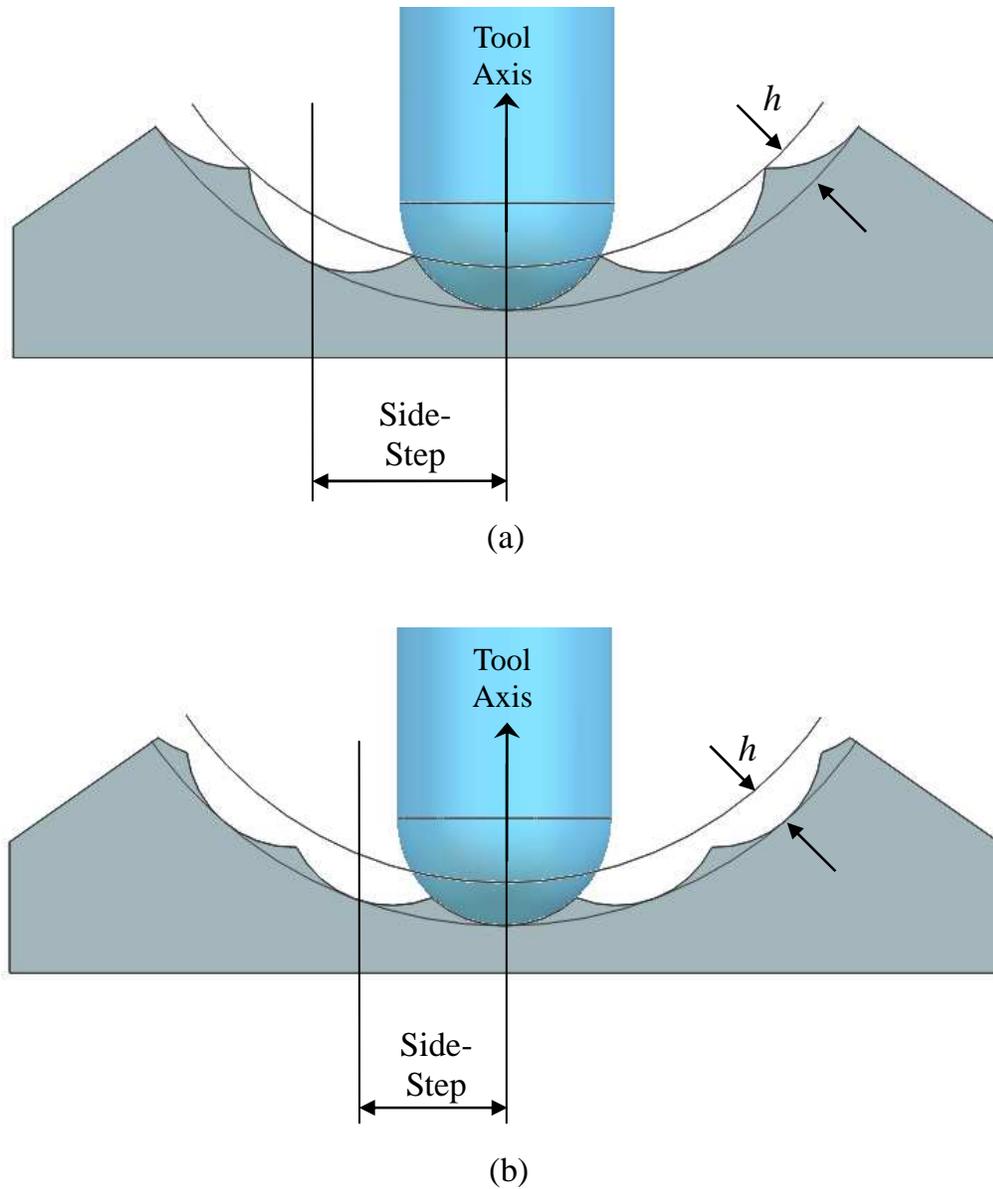


Figure 1 Illustration of the side-steps when a) there is no redundant machining and b) there is redundant machining.

1.1. Conventional Tool Path Generation Methods

The conventional tool path generation methods can be roughly categorized into iso-parametric, iso-planar and iso-scallop. The iso-parametric method is the earliest of these three methods and was proposed by Loney and Ozsoy [1]. A design surface is a three-dimensional parametric

surface expressed by two variables, u and v . This method generates a cutter contact (CC) path on the design surface by choosing an initial parametric variable u or v and keeping it constant while increasing the value of the other parametric variable that describes the surface. The generated curves are known as iso-curves. The advantage of using iso-curves is that these curves do not intersect each other and recreate the contour of the parametric surface. The next parametric variable would be chosen so that the next iso-parametric curve would not have points with a scallop height higher than the one specified. This method is mathematically convenient and it ensures that the entire surface is covered by machining passes. However, depending on the surface, the tool paths generated by this method may be very dense in some zones, as pointed out by Elber and Cohen [2]. As an improvement of the iso-parametric method, their method proposes the use of iso-curves as CC tool paths with discontinuities in regions where the iso-curves would be too dense. Later methods would also use the iso-parametric method along with triangular meshes [11] or adaptive grids [12] to obtain larger step-sizes between adjacent tool paths. Their main aim was to obtain iso-parametric tool paths that more closely resembled tool paths with the benefits of the iso-scallop tool path method. These methods still do not generate an optimized overall tool path with a shorter length than the iso-scallop method because the iso-parametric method does not ensure overlapping tool paths would not occur at any point of the surface.

The iso-planar method was first proposed by Huang and Oliver [3] as an improvement over the iso-parametric method. This method consists on using a series of parallel planes whose intersections with the design surface would generate the CC tool paths. The separation between the intersection planes provides a better control on the resulting scallop height than the iso-parametric method because the separation is controlled by a Cartesian distance

between planes instead of using parametric curves. The advantage of the iso-planar method is that it addresses the problem of regions with dense tool paths observed in the iso-parametric method. However, the tool path generated by the iso-planar method is still longer if compared with the iso-scallop method because there will still be pairs of adjacent tool paths where the scallop height is not maximized unless the surface is completely planar. Improvements based on the iso-planar method have been proposed. For example Ding et al. [9] implemented an adaptive iso-planar method, also called the iso-phote method. In their strategy, the design surface was segmented in different regions according to boundaries determined by the slope of the surface. The concept of segmenting the surface stems from the fact that regions with a larger slope require shorter side steps to fulfill a given scallop height compared to regions with a smaller slope. However, this method still requires input from the user regarding the limits of the slope used for surface segmentation. The iso-phote regions' size would greatly vary depending on the slope that was used as a threshold to segment a surface.

A mathematical method that generated an iso-scallop tool path was proposed by Suresh and Yang [4]. The iso-scallop method relies on generating the adjacent tool paths based on the scallop curve created by the previous tool path. This method ensures that the scallop height of the points of the scallop curves remain as high as the specified tolerance. This way, the generated tool paths cause no redundant machining, resulting in a tool path with a smaller overall length than the tool paths generated by the iso-parametric or iso-planar methods. Methods have been proposed that improve on the calculation efficiency (Lin and Koren [5]) and their accuracy by using swept profiles (Sarma and Dutta [6]). The important drawback of these strategies is that these methods assumed that the tangent vectors of the cutter location path, the scallop curve, and the adjacent cutter location path are all parallel. This assumption

causes inaccuracies in the location of the points of the scallop curve given an initial tool path. The calculation methods proposed by Feng and Li [7] and Tournier and Duc [8] provide a scallop curve that is accurately calculated because the tangent vectors of the scallop, CC, and CL paths are obtained according to their three-dimensional components. This strategy is adopted in the present work to obtain the adjacent iso-scallop tool paths and explained in section 3. Methods that would attempt to improve the speed of the three-dimensional calculations would be proposed (Yoon [10]). The most important characteristic of the conventional iso-scallop strategies is that they still require the input of an initial tool path from which the adjacent tool paths would be generated. For a given surface, since an infinite number of initial tool paths exist, there is also an infinite number of possible overall tool paths that can produce a desired design surface from the workpiece. This leads to a direction of research that attempts to analyze the topology of the surface with the ultimate goal of finding the overall tool path that minimizes the total machining length.

1.2. Surface Topology-Based Methods

Research by Quinsat et al. [17] and Vijayaraghavan et al. [18] sought to find an optimized orientation of the workpiece to find the iso-planar (or raster) tool path that could minimize the tool path length (in iso-planar methods, changing the plane orientation, or machining direction, is the same as changing the workpiece orientation). The optimized orientation was determined based on local parameters of the surface that influenced the material removal rate or by a metric such as the mean scallop height resulting from a particular machining direction. However, these methods rely on using a single iso-planar direction along the entire surface, which in principle does not generate a minimized overall tool path length because there will be regions on the surface that will not benefit from the chosen initial plane orientation.

Attempts have been made to find the optimal initial tool path by analyzing regions of the surface and machining them using the iso-scallop method. The strategy proposed by Giri et al. [15] used the curvature of the input surface to find the loci of maximum convex curvature to determine the Master Cutter Path (MCP) from which the adjacent iso-scallop tool paths would be generated. The maximum convex curvature loci on the surface could provide a visualization of possible MCP's that could be used as the initial tool path. Their method then compares pairs of loci and chooses as MCP's a pair that is considered differing according to their gradients of their line segments. This method may lead to inconsistencies based on the input surface because segmentation heavily relies on the pairs of loci chosen to be compared. Moreover, the method still requires the user to make a final decision on the boundaries generated by the curvature loci in order to choose an initial tool path.

It has been proposed by Kim and Sarma [14] to treat the optimal machining directions on sampled points on the input surface as components of a vector field that can be fitted to a continuous vector field function. The criteria to determine the preferred direction also incorporates the kinematic aspects of the milling machine. The side-step from tool paths is also incorporated in the fitted function and the tool paths from each region can then be determined to maximize the side-step by solving an optimization problem. Surface segmentation was also carried out based on the outlets and inlets of the streamlines of the fitted vector field. The most important drawback of this method is that the optimal directions on a surface that lead to a greater material removal are actually bi-directional (the preferred direction is valid in both opposite directions) if the kinematic aspect is not considered. If the field becomes bi-directional, then the idea of outlets and inlets is no longer valid. A vector

field only considers one direction at each point and it cannot represent certain properties of a bi-directional field.

In this work, a method is proposed to determine a 3-axis machining strategy that attempts to decrease the overall tool path length with a given ball-end radius and scallop height. The method will not require input from the user regarding conditions for an initial tool path. The fundamental principle of the presented method is to generate the tool paths according to a *Preferred Feed Direction* (PFD) field that follows the directions of maximum machining strip width, a metric that indicates the amount of material removal. This ideal situation does not happen in general because if all the tool paths always follow the preferred direction, the machining strips of these paths generally overlap. Redundant machining can be attenuated by using the iso-scallop method, but adjacent tool paths generated from the iso-scallop method tend to stop following the preferred directions. The proposed method seeks to generate a machining tool path that can benefit from both following the preferred direction and avoiding redundant machining. The strategy consists on creating a discrete bi-directional field, or tensor field, that indicates the preferred machining directions on the surface. The criterion to decide the preferred machining direction is the machining strip width, which provides a measurement of material removal during a very small tool pass. The field is then analyzed and segmented into regions with similar PFD's. Inside each region, an initial tool path is chosen so the iso-scallop tool paths are generated in a way that these do not deviate significantly from the preferred directions. By using this proposed initial tool path, the deviation between the preferred direction field and the iso-scallop paths is decreased. This results in a generated tool path with a shorter overall length than using the conventional iso-scallop method.

2. Preferred Feed Direction

In order to analyze a design surface, a metric is required to determine what direction is the preferred at a certain point on the design surface. The criteria that indicate where would be a preferred machining direction is a topic of current research [19][25]. It is hypothesized that the tool path that follows the direction that removes the most material will have the shortest overall length. In this work, the *Machining Strip Width* is chosen as a metric of material removal. Its maximum defines the preferred direction and the method to calculate it on a given point on the design surface will be explained in section 2.1. The preferred direction is determined for a number of sampled points on the surface and the set of preferred directions build a field on the surface. This is explained in section 2.2.

2.1. Machining Strip Width

The concept of the Machining Strip Width (W) was first proposed by Lee and Ji [20] as a way to mathematically analyze 5-axis machining and to find optimal machining directions [13, 21]. W can still be calculated for a 3-axis machining to determine a preferred direction [22-23] and there are different ways to approximate it [24].

This section explains how W is evaluated. Let the ball-end mill radius be defined as r . The *scallop surface* is an offset of the design surface with a normal distance h , also known as the scallop height. In this section, it will be assumed that the motion of the ball-end mill's center is given as the *Cutter Location (CL)* path that is represented as the trajectory denoted by series of point positions (P_{CL}) of the center of the ball-end mill. The tool motion can be represented as a three-dimensional volume, called the *swept volume* that subtracts the volume enclosed between the design and scallop surface to create the machined finished volume. The swept

volume's contact with the design surface is a curve on the design surface called the Cutter Contact (CC) path, which is constituted by CC points (\mathbf{P}_{CC}). Every CC point has a matching CL point. In the case of three-axis machining, the CL point \mathbf{P}_{CL} at a given CC point \mathbf{P}_{CC} is calculated by multiplying the normal vector \mathbf{n}_{CC} by the tool radius. That is, the CC point and the CL point share the relation:

$$\mathbf{P}_{CL}(u, v) = \mathbf{P}_{CC}(u, v) + \mathbf{n}_{CC}r \quad (1)$$

At each CL point, three unit vectors are determined: \mathbf{n}_{CL} , an extension of \mathbf{n}_{CC} ; \mathbf{f}_{CL} , the tangent vector of the CL path at the CL point; and \mathbf{t}_{CL} , calculated as the unit vector of the cross product of \mathbf{n}_{CL} and \mathbf{f}_{CL} .

Mathematically, the expressions of \mathbf{n}_{CL} , \mathbf{f}_{CL} , and \mathbf{t}_{CL} are the following. Let $\mathbf{P}_S(u, v)$ be a point on the design surface and $\mathbf{CL}(t)$ the parametric expression of the curve that represents the CL path:

$$\mathbf{n}_{CL} = \mathbf{n}_{CC} = \frac{\mathbf{P}_{Su}(u, v) \times \mathbf{P}_{Sv}(u, v)}{|\mathbf{P}_{Su}(u, v) \times \mathbf{P}_{Sv}(u, v)|} \quad (2)$$

$$\mathbf{f}_{CL} = \mathbf{CL}'(t) = \frac{\frac{d\mathbf{CL}(t)}{dt}}{\left| \frac{d\mathbf{CL}(t)}{dt} \right|} \approx \frac{\mathbf{CL}(t+1) - \mathbf{CL}(t)}{|\mathbf{CL}(t+1) - \mathbf{CL}(t)|} \quad (3)$$

$$\mathbf{t}_{CL} = \frac{\mathbf{n}_{CL} \times \mathbf{f}_{CL}}{|\mathbf{n}_{CL} \times \mathbf{f}_{CL}|} \quad (4)$$

$\mathbf{P}_{Su}(u, v)$ and $\mathbf{P}_{Sv}(u, v)$ are the partial derivatives of $\mathbf{P}_S(u, v)$ with respect to the parameter u and v , respectively.

The swept volume is the volume enclosed by the surface constituted by the union of the swept profiles, which are intersection curves derived at each CL point. The tool volume of a ball-end mill is made by the union of two volumes: a half-sphere for the ball-end and a cylinder for the tool flank. The half-sphere is intersected by the plane created by the \mathbf{t}_{CL} and \mathbf{n}_{CL} vectors at the CL point. The generated intersection curve constitutes the *bottom swept profile*, that contacts the circle of the flank cylinder's bottom at two points S_a and S_b . The vector between S_a and S_b and the tool axis \mathbf{z}_T create a plane that intersects the tool flank cylinder and the resulting intersection curve is the *flank swept profile*. The union of the *bottom swept profile* and the *flank swept profile* curves result in the *swept profile*.

In three-axis machining with a ball-end mill, only the bottom swept profile at the ball-end is of interest. The flank swept profile is not required for the calculation of the scallop curve points because the flank is not supposed to perform the machining operation. Hence, the bottom swept profile will now be referred as the *swept profile*.

The swept profile can be calculated as the intersection of a half-sphere with radius r by the plane that goes through the sphere's center and is created by the two vectors \mathbf{n}_{CL} and \mathbf{t}_{CL} , as detailed in section 2.1. The intersection curve of a sphere with radius r with a plane that passes through the sphere center is a circle that also has a radius r . The parametric equation of a circle with radius r on the x - z plane is the following:

$$\mathbf{P}_{circle}(\theta) = \begin{bmatrix} r \cos \theta \\ 0 \\ r \sin \theta \\ 1 \end{bmatrix} \quad \text{such that } \theta = [-\pi, 0] \quad (5)$$

The curve is translated to the CL point and rotated so that it lies on the plane created by the vectors \mathbf{n}_{CL} and \mathbf{t}_{CL} using the translation and rotation matrices:

$$\mathbf{Rot} = \begin{bmatrix} \mathbf{t}_{x_{CL}} & \mathbf{f}_{x_{CL}} & \mathbf{n}_{x_{CL}} & 0 \\ \mathbf{t}_{y_{CL}} & \mathbf{f}_{y_{CL}} & \mathbf{n}_{y_{CL}} & 0 \\ \mathbf{t}_{z_{CL}} & \mathbf{f}_{z_{CL}} & \mathbf{n}_{z_{CL}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Trans} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{P}_{x_{CL}} \\ 0 & 1 & 0 & \mathbf{P}_{y_{CL}} \\ 0 & 0 & 1 & \mathbf{P}_{z_{CL}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The components of the rotation matrix \mathbf{Rot} are the components of the unit vectors \mathbf{n}_{CL} , \mathbf{f}_{CL} , and \mathbf{t}_{CL} as follows:

$$\mathbf{n}_{CL} = \begin{bmatrix} \mathbf{n}_{x_{CL}} \\ \mathbf{n}_{y_{CL}} \\ \mathbf{n}_{z_{CL}} \end{bmatrix}$$

$$\mathbf{f}_{CL} = \begin{bmatrix} \mathbf{f}_{x_{CL}} \\ \mathbf{f}_{y_{CL}} \\ \mathbf{f}_{z_{CL}} \end{bmatrix}$$

$$\mathbf{t}_{CL} = \begin{bmatrix} \mathbf{t}_{x_{CL}} \\ \mathbf{t}_{y_{CL}} \\ \mathbf{t}_{z_{CL}} \end{bmatrix}$$

The components of the rotation matrix \mathbf{Trans} are the components of the CL point $\mathbf{P}_{CL}(t)$ on the CL path as follows:

$$\mathbf{P}_{CL}(t) = \begin{bmatrix} \mathbf{P}_{x_{CL}}(t) \\ \mathbf{P}_{y_{CL}}(t) \\ \mathbf{P}_{z_{CL}}(t) \end{bmatrix}$$

Then, the swept profile is obtained by:

$$\mathbf{P}_{swept}(\theta) = \mathbf{P}_{i=1,\dots,3} \quad ; \quad \mathbf{P} = \mathbf{Trans} * \mathbf{Rot} * \mathbf{P}_{circle}(\theta) \quad (6)$$

The swept profile intersects the scallop surface in two points, P_a and P_b (an exception occurs when the ball-end mill is near the boundaries of the design surface, where the swept profile may intersect the scallop surface in only one point). At any given point on the design surface $\mathbf{P}_s(u, v)$, the corresponding scallop point $\mathbf{P}_{scall}(u, v)$ can be obtained as follows:

$$\mathbf{P}_{scall}(u, v) = \mathbf{P}_s(u, v) + \mathbf{n}_s h$$

Where \mathbf{n}_s is the normal vector of the design surface calculated at the point $\mathbf{P}_s(u, v)$.

The points P_a and P_b are obtained by solving the equation:

$$\mathbf{P}_{swept}(\theta) - \mathbf{P}_{scall}(u, v) = 0 \quad (7)$$

This equation yields two sets of solutions for θ , u , and v .

This equation is not solved analytically for most surfaces. The reason is that the offset surface of a complex parametric surface is not determined analytically. Instead, the normal is calculated at a point on the design surface and then multiplied by the offset. There are different methods to numerically solve this equation. In this work, the Newton-Rhapson method was applied and each of the two sets of solutions was obtained by choosing different initial values of θ and using the values of u and v of the CC point.

Given the two sets of u and v obtained from the previous equation, u_i, v_i , such that $i=1,2$, the points $\mathbf{P}_a(u_1, v_1)$ and $\mathbf{P}_b(u_2, v_2)$ are obtained as follows:

$$\mathbf{P}_a(u_1, v_1) = \mathbf{P}_S(u_1, v_1) + \mathbf{n}_a h$$

$$\mathbf{P}_b(u_2, v_2) = \mathbf{P}_S(u_2, v_2) + \mathbf{n}_b h$$

\mathbf{n}_a and \mathbf{n}_b are the normal unit vectors of the design surface using the pairs of u and v parameters from the solution:

$$\mathbf{n}_a = \frac{\mathbf{P}_{Su}(u_1, v_1) \times \mathbf{P}_{Sv}(u_1, v_1)}{|\mathbf{P}_{Su}(u_1, v_1) \times \mathbf{P}_{Sv}(u_1, v_1)|} \quad (8)$$

$$\mathbf{n}_b = \frac{\mathbf{P}_{Su}(u_2, v_2) \times \mathbf{P}_{Sv}(u_2, v_2)}{|\mathbf{P}_{Su}(u_2, v_2) \times \mathbf{P}_{Sv}(u_2, v_2)|}$$

The projected distance between \mathbf{P}_a and \mathbf{P}_b on the \mathbf{t}_{CL} vector results in W :

$$W = \left| \overline{\mathbf{P}_a \mathbf{P}_b} \cdot \mathbf{t}_{CL} \right| \quad (9)$$

Fig. 2 illustrates the geometry of the swept profile and the machining strip width W .

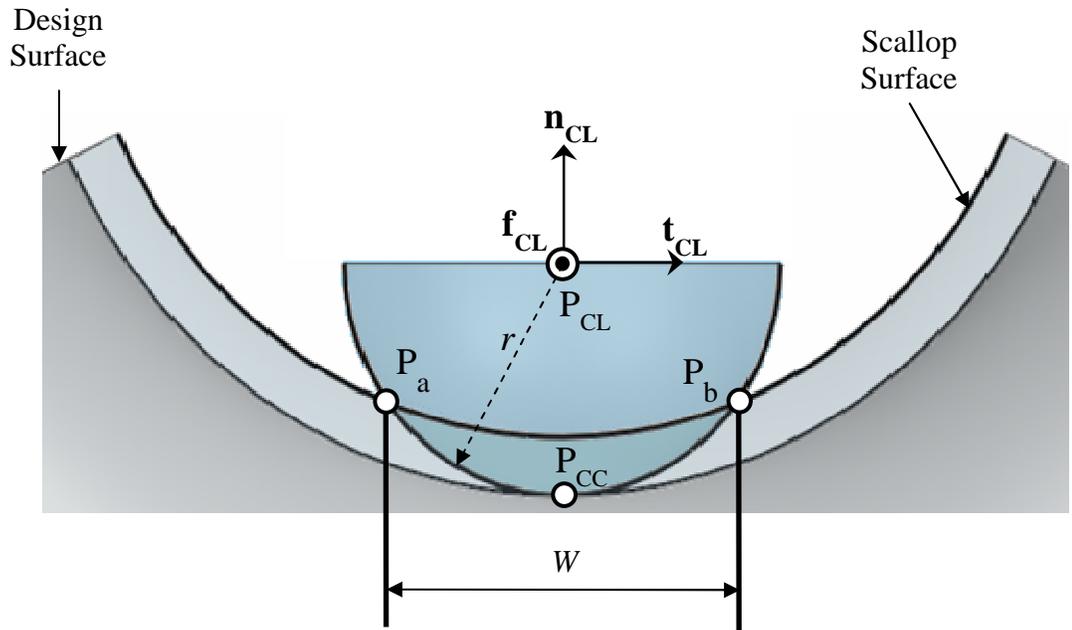
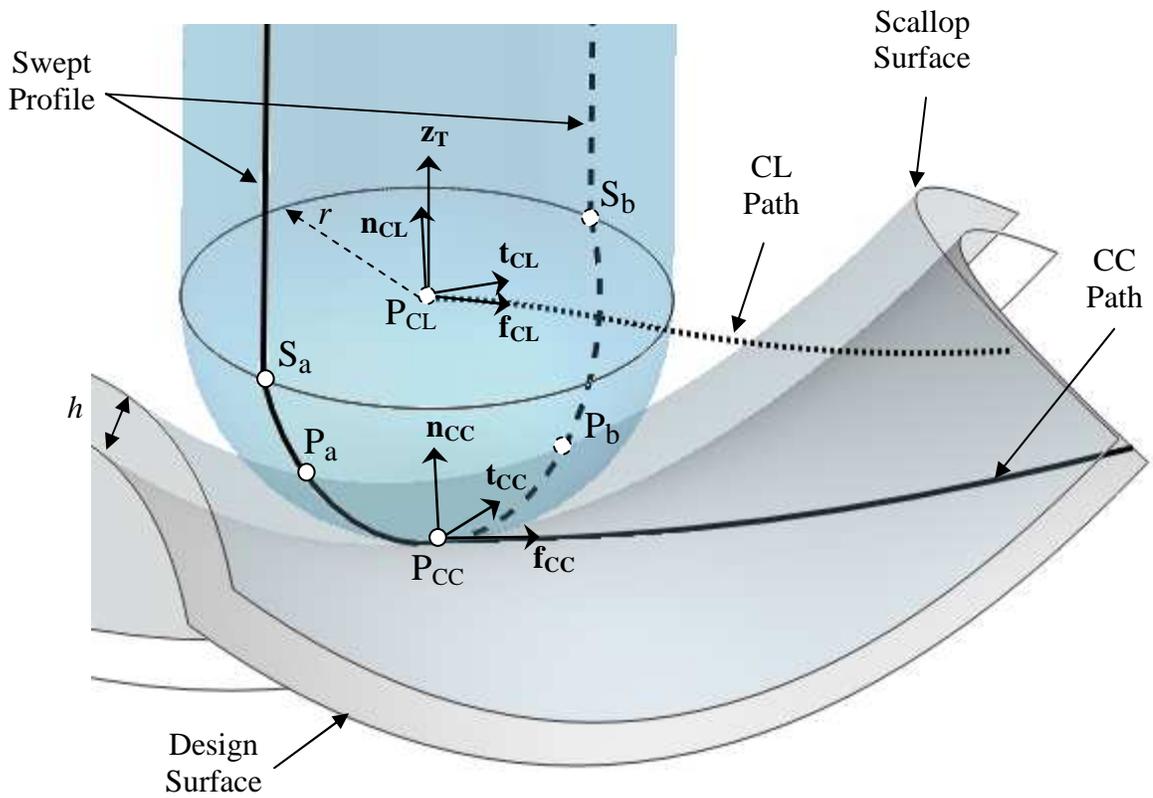


Figure 2 Geometry of the Swept Profile and Machining Strip Width.

2.2. Preferred Feed Direction Field

For each point, the feed directions that provide the maximum machining strip width W become the Preferred Feed Direction (PFD). It is important to note that, a given CC point, a value of W will be the same whether the machining feed direction is positive or negative. For every CC point sampled to find its preferred direction, there will be two opposing vectors, $+\mathbf{f}_{CL}$ and $-\mathbf{f}_{CL}$, that have the same magnitude of W . By sampling different points on the design surface and calculating the directions with the largest value of W , a field of preferred directions can be obtained. In three-axis machining, if the tool axis is the global z coordinate, every point on the surface, and thus each sampled CC point, will have a unique pair of global x - y coordinates. Each pair of vectors at each CC point can be projected into the x - y plane. This is illustrated in Fig. 3.

It is useful to represent the preferred direction as a scalar unit. In this case will be the *feed angle* ϕ and it is calculated as follows. A plane can be defined by its normal vector and a point. The normal vector is chosen to be \mathbf{n}_{CC} at the CC point P_{CC} . The plane perpendicular to \mathbf{n}_{CC} at P_{CC} is called the *feed plane*. The feed vector \mathbf{f}_{CC} lies on the feed plane. Next, a reference feed vector \mathbf{f}_{ref} , that lies on the feed plane, is evaluated as the cross product of the tool axis \mathbf{z}_T and the normal vector. The angle between the reference feed vector \mathbf{f}_{ref} and the feed vector \mathbf{f}_{CC} is the feed angle ϕ . Since the preferred direction consists of two opposing vectors, the feed angle can have a value of ϕ or $\phi + \pi$. If $\phi \geq \pi$, then π is subtracted from ϕ . Fig. 4 shows an example of a graph comparing W vs ϕ .

The resulting PFD field contains two vectors on every sampled point. It is because of this reason a PFD field cannot be analyzed as a vector field, where there is one single vector for

every point. In this work, the PFD field will be analyzed as a tensor field. A tensor is a mathematical component that describes the vectors' information and also their relations between them. The importance of following the preferred direction can be exemplified with a half-cylinder. This is illustrated in Fig. 5. In this case, the machining tool paths that follows the direction of minimum W result in a noticeably longer overall length than the tool paths that follow the direction of maximum W even if the iso-scallop height is kept at a maximum.

Referring to the analogy of a vector field, when integrating a vector field given an initial point, the result is a curve known as a streamline. In the research of Delmarcelle and Hesselink [27], the equivalent of a streamline for a tensor field is a “hyperstreamline trajectory”. In a 2D symmetrical tensor field, this curve grows in two directions at all times. In the present work, these curves will be known as *PFD pathlines*, or simply pathlines. The methods to generate the pathlines from a given PFD field will be detailed in Section 4.1.

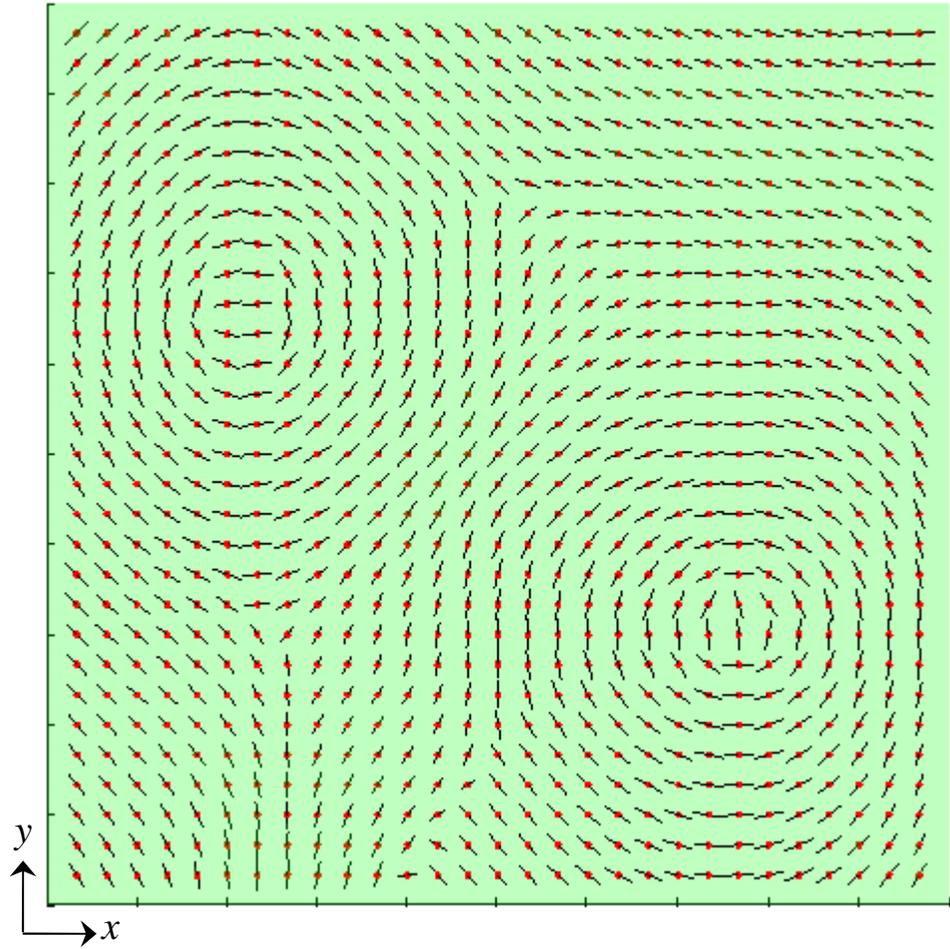
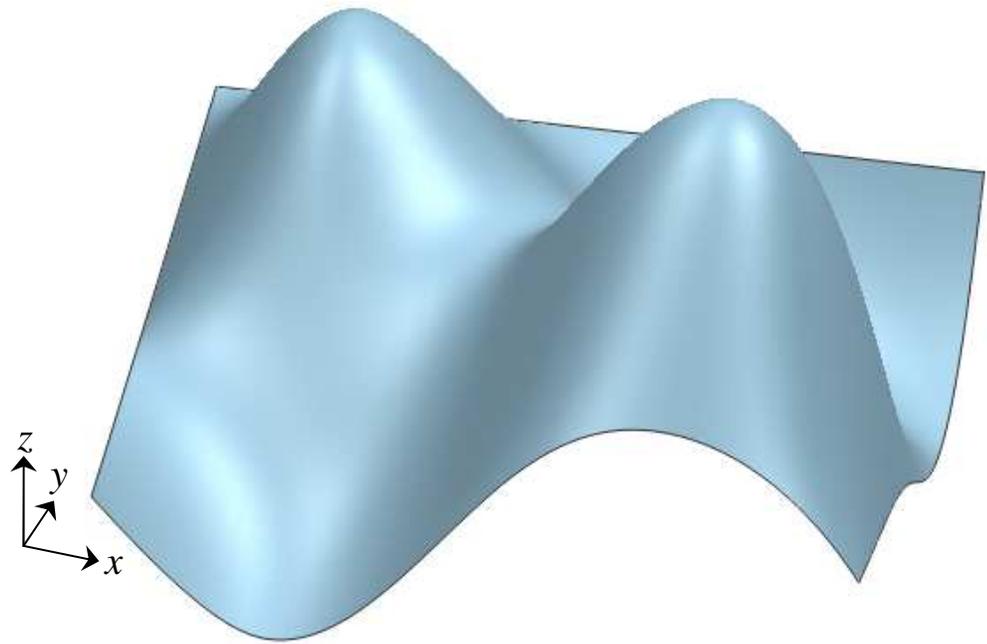


Figure 3 Example of a Preferred Feed Direction Field.

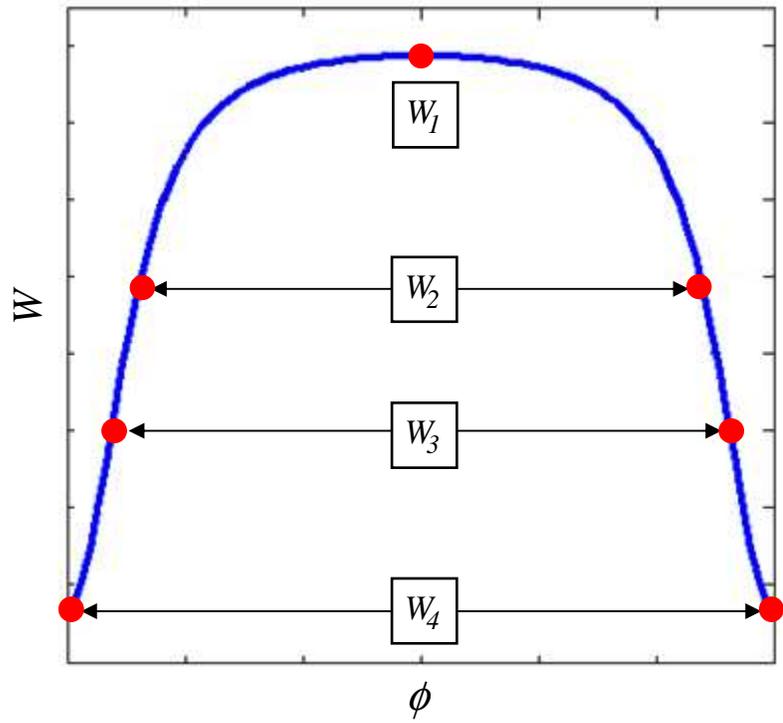
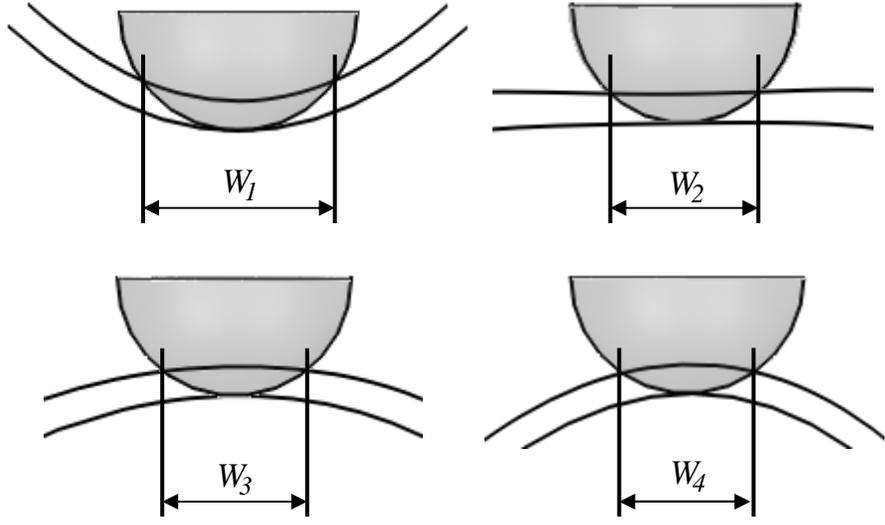


Figure 4 Graph comparing W vs ϕ on a CC point.

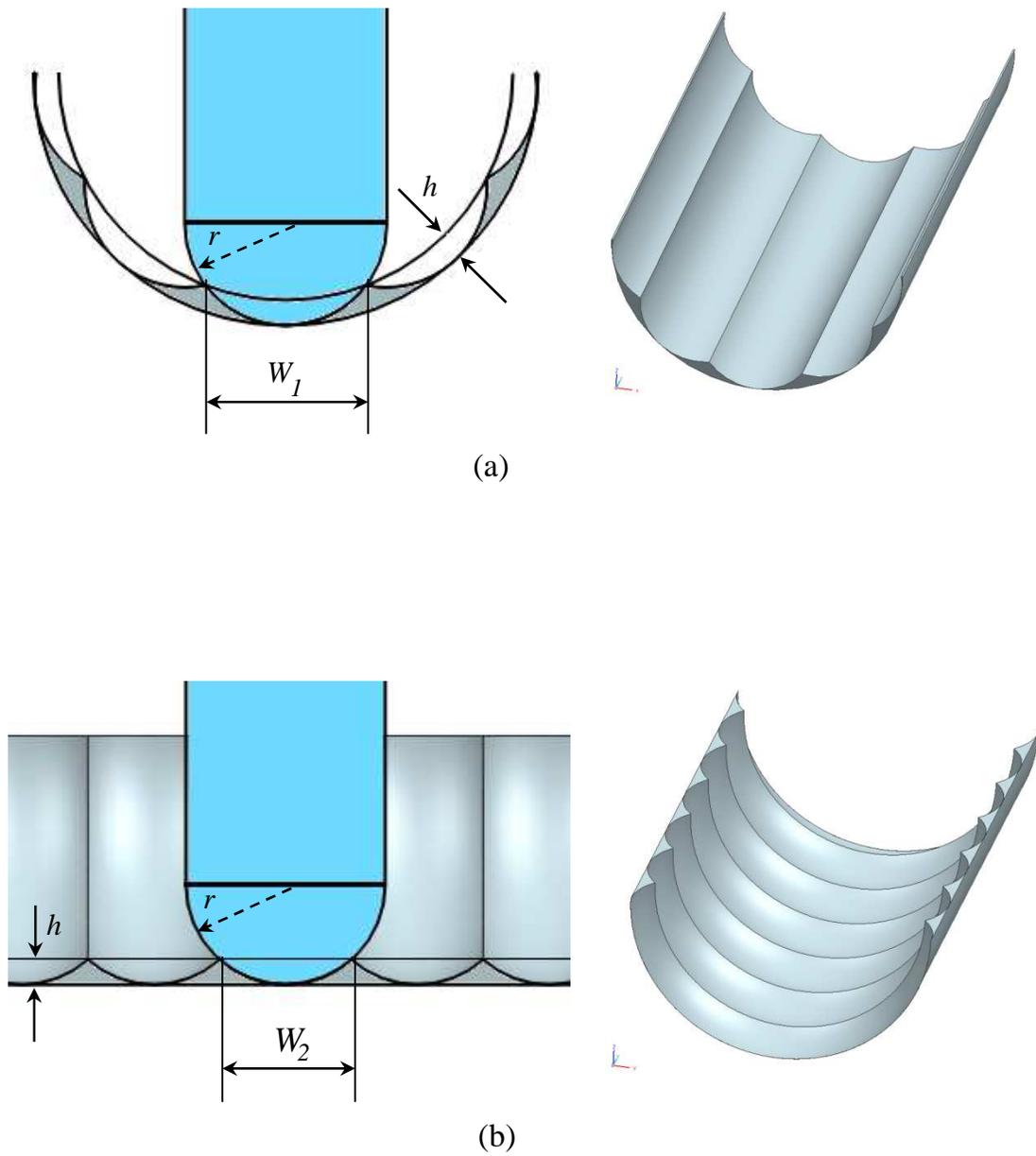


Figure 5 Tool paths with maximum scallop height following the direction of a) maximum strip width W_1 and b) minimum strip width W_2 .

3. Iso-scallop Tool Path Generation

In this section, the method to obtain the adjacent iso-scallop tool path of a given CC path will be explained. The method is an implementation based on the research by Feng and Li [7]. As mentioned in section 1.1, the earliest methods to obtain an adjacent iso-scallop tool used two-dimensional assumptions. The side step of a given CC point would be calculated using the local curvature. This assumption cannot be considered because the input CC point, its two scallop points P_a and P_b , and the adjacent scallop point do not necessarily lie on the same plane. The reason for this is that the CC point's feed direction, \mathbf{f}_{CC} , the tangent vector of the scallop curve at P_a and P_b , and the feed direction of the adjacent CC point that shares the scallop points P_a or P_b with the previous CC point are not parallel. Errors can arise on the generation of the scallop curve by using two-dimensional assumptions and the resulting three-dimensional CC paths may actually present overlapping tool paths.

Let an initial curve on the design surface $\mathbf{CC}(t)$ be a curve that is constituted by CC points $\mathbf{P}_{CC}(u(t), v(t))$. The CL path $\mathbf{CL}(t)$ is a curve that is constituted by CL points $\mathbf{P}_{CL}(u(t), v(t))$. For each CL point, the scallop points P_a and P_b are obtained as explained in Section 2.1. Once the P_a and P_b points of all the CL points are calculated, the scallop curves $\mathbf{SC}_{P_a}(t)$ and $\mathbf{SC}_{P_b}(t)$ are obtained. Since two scallop curves are formed from one CL path (except near or on the borders of the design surface or region of interest), two adjacent tool paths are created, one for each scallop curve.

To obtain the adjacent tool path, the tangent of the scallop curve at a point $\mathbf{SC}(t)$ is calculated. If the CC and CL paths are discrete, then the scallop curves will also be discrete and therefore their tangents are a linear vector from $\mathbf{SC}(t)$ to $\mathbf{SC}(t+1)$. Let that vector be

$\mathbf{f}_{\text{scall}}$. The adjacent tool path's CL point lies on the same plane that contains the scallop point $\mathbf{SC}(t)$ and is perpendicular to the vector $\mathbf{f}_{\text{scall}}$. Let that plane be called the SC-CL_{adj} plane. The adjacent CL point $\mathbf{CL}_{adj}(t)$ must satisfy two conditions: a) its distance to the scallop point $\mathbf{SC}(t)$ must be equal to the ball-end mill radius r , and b) its distance to the design surface must also be equal to r . These two conditions can be met if a point is obtained by the intersection of a circle lying on the SC-CL_{adj} plane with its center on $\mathbf{SC}(t)$ and the scallop surface. This intersection point will become the adjacent CL point $\mathbf{CL}_{adj}(t)$. Analytically, it is obtained as follows (for the case of the adjacent point using $\mathbf{SC}_{Pa}(t)$):

Given that $\mathbf{P}_a(u_1, v_1)$ was obtained previously, then the vector \mathbf{n}_a was also evaluated. The parametric curve of the circle is translated into the plane created by the vectors $\mathbf{f}_{\text{scall}}$ and \mathbf{n}_a in a similar way explained in section 2.1. The curve of the circle on the SC-CL_{adj} plane is expressed by:

$$\mathbf{P}_{SC-CLadj}(\theta) = \mathbf{P}_{i=1, \dots, 3} ; \quad \mathbf{P} = \mathbf{Trans}_{SC-CLadj} * \mathbf{Rot}_{SC-CLadj} * \mathbf{P}_{circle}(\theta) \quad (10)$$

The CL point lies on the offset surface of the design surface with an offset distance equal to the tool radius r . Then,

$$\mathbf{P}_{CL}(u, v) = \mathbf{P}_S(u, v) + \mathbf{n}_S r \quad (11)$$

where \mathbf{n}_S is the normal vector of the design surface calculated at the point $\mathbf{P}_S(u, v)$.

The u and v parameters are then solved from the equation:

$$\mathbf{P}_{SC-CLadj}(\theta) - \mathbf{P}_{CL}(u, v) = 0 \quad (12)$$

Similar to the calculation of P_a and P_b , this equation yields two sets of solutions for θ , u , and v . The adjacent CL point is the solution that gives the CL point location with the largest distance from the original CL point.

The procedure is then repeated until a border of the design surface or the region of interest is reached. Then, the adjacent paths are obtained from the scallop curve $\mathbf{SC}_{P_b}(t)$, which is composed by the P_b scallop points of the original CC path. Fig. 6 shows the geometric components of a CC and CL path, one of their scallop curve, and their adjacent CC and CL paths.

In the case an adjacent CC path does not reach a border, the CC path is extended by creating a plane with the normal and tangent vectors from either the start or end of the CC path and then intersecting the plane with the design surface. This will help to ensure that there will be no regions that are not machined when applying the iso-scallop method.

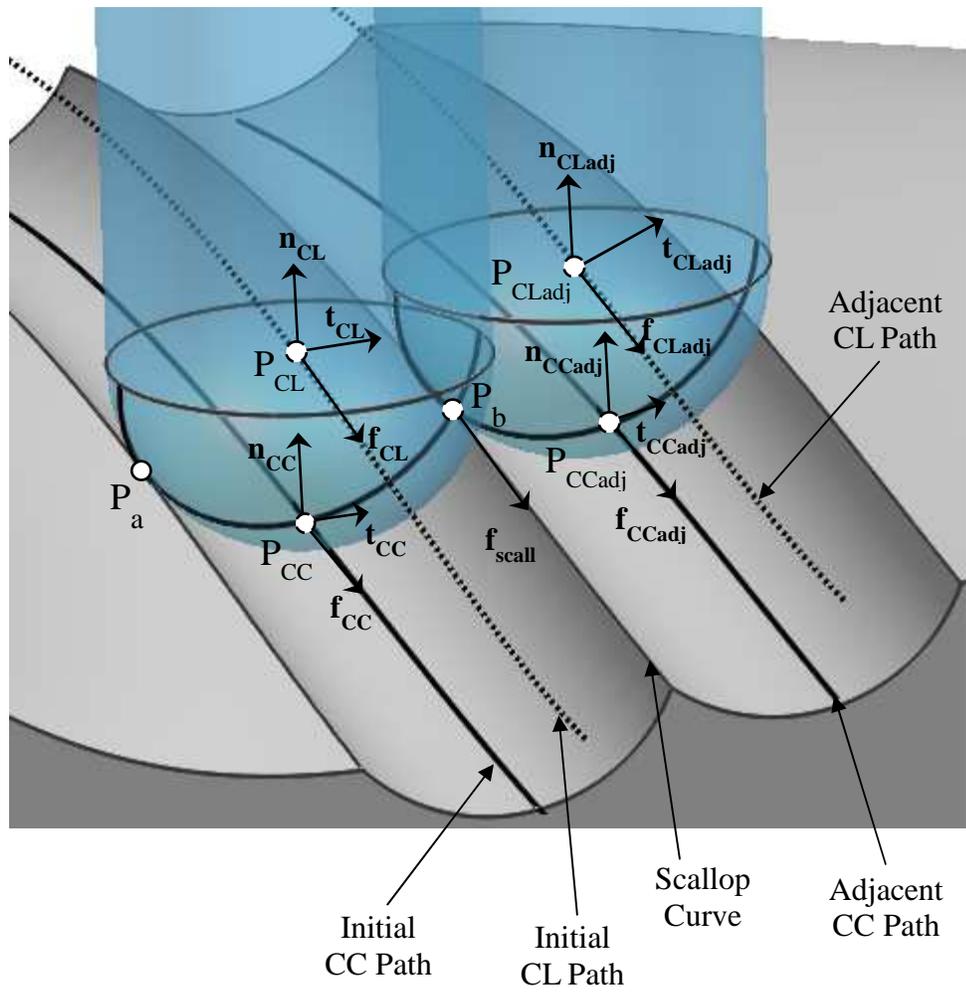


Figure 6 Geometry of the iso-scallop tool path generation method.

4. Methodology

In this section, the method to generate efficient tool paths on a design surface according to its PFD field will be explained in detail. Section 4.1 explains the concept of *degenerate points* according to a tensor field representation. Also detailed in this section is the concept of a *PFD pathline*, and its relationship to the degenerate points. In section 4.2, the *segmentation* of the design surface is explained. The concept of the *Principal Tool Path* as the initial iso-scallop tool path chosen to decrease *iso-scallop drift*, or deviation from the PFD field, is explained in section 4.3.

An overview of the proposed method is as follows. The strategy consists on segmenting an input sculptured surface into regions with similar PFD's according to the separatrices of the degenerate points of the PFD field. By segmenting the surface into regions with similar directions, the iso-scallop tool paths, characterized by having a similar direction, will tend to follow the directions of maximum material removal based on the machining strip width. The principal tool path for each region is then determined, and then used as the initial tool path for the adjacent iso-scallop tool paths that cover the region. The reason to use the principal tool path as the initial tool path is to decrease the effect iso-scallop drift, which is the deviation of the tool paths from the preferred directions. The algorithm is repeated for all regions until the entire input surface is covered. Fig. 7 shows a flowchart depicting the steps of the proposed method.

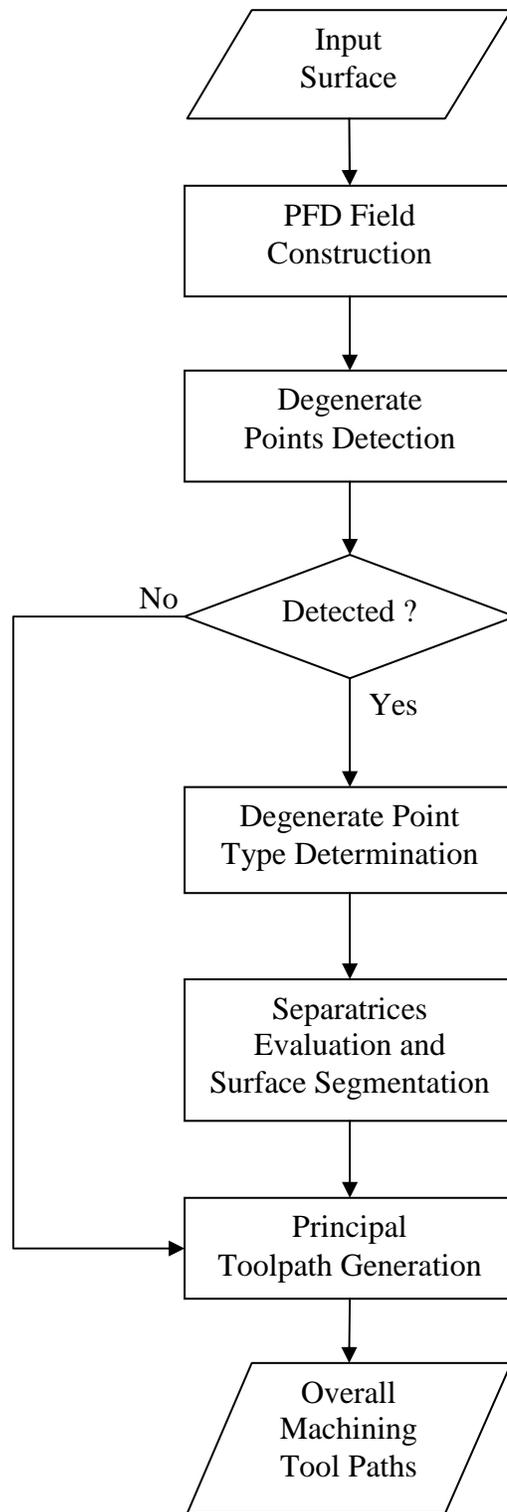


Figure 7 Flowchart of the proposed method.

4.1. Degenerate Points

Tensor fields have been a subject of research in the area of mathematics and applied to various research topics such as fluids and stress-strain mechanics. The study of components of tensor topology has been introduced by Delmarcelle and Hesselink [26-27] as a need to visualize and understand the information contained in tensor fields. Their work introduced the concept of *degenerate points* and their *separatrices* as the basic constituents of a tensor field. In this section, the concept and evaluation procedure of the *degenerate points* on a 3-axis machined sculptured surface will be explained.

4.1.1. Tensor Field Representation

In the case of a two-dimensional symmetrical tensor field, each point has a corresponding second-order tensor (a 2x2 matrix) from which the direction of the two vectors can be obtained. The direction information from the PFD field can be analyzed as two-dimensional because, as pointed out earlier, each point has a unique x - y coordinate along with the directions projected onto the x - y plane. The z component of the direction vector is not required in the analysis because there is a unique z component vector for a given x - y direction. It is also important to note that the preferred direction does not contain a scalar magnitude as in the case of vectors because only the direction is of interest.

Each sampled point contains a tensor in the form of:

$$\mathbf{T}(x, y) = \begin{pmatrix} T_{11}(x, y) & T_{12}(x, y) \\ T_{12}(x, y) & T_{22}(x, y) \end{pmatrix} \quad (13)$$

For a given point with the vectors of preferred directions $\mathbf{v}_1(x, y)$ and $\mathbf{v}_2(x, y) = -\mathbf{v}_1(x, y)$, the tensor $\mathbf{T}(x, y)$ is obtained as follows:

$$\mathbf{T}(x, y) = \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \quad (14)$$

The feature of a tensor $\mathbf{T}(x, y)$ is that it is equivalent to two orthogonal eigenvectors of the form:

$$\mathbf{v}_i(x, y) = \lambda_i(x, y)\mathbf{e}_i(x, y), \text{ where } i = 1, 2. \quad (15)$$

From the definition of eigenvalues and eigenvectors:

$$\mathbf{T}(x, y)\mathbf{e}_i(x, y) = \lambda_i(x, y)\mathbf{e}_i(x, y) \quad (16)$$

λ_i are the eigenvalues of $\mathbf{T}(x, y)$ and \mathbf{e}_i their corresponding unit eigenvectors. As noted before, the PFD field's components do not have a magnitude because only the direction is of interest.

Given a PFD field, each sampled point is then assigned a 2x2 matrix $\mathbf{T}(x, y)$. To obtain the tensor matrix $\mathbf{T}(x, y)$ for any given point on the surface, bilinear interpolation of the tensor components of the surrounding sampled points can be used.

The tensor components $\mathbf{T}(x, y)$ of can also be obtained by using bilinear interpolation. Consider a point (x, y) inside a rectangular grid with four points, each having their own tensor, as shown in Fig. 8, and its tensor matrix $\mathbf{T}(x, y)$ is to be determined.

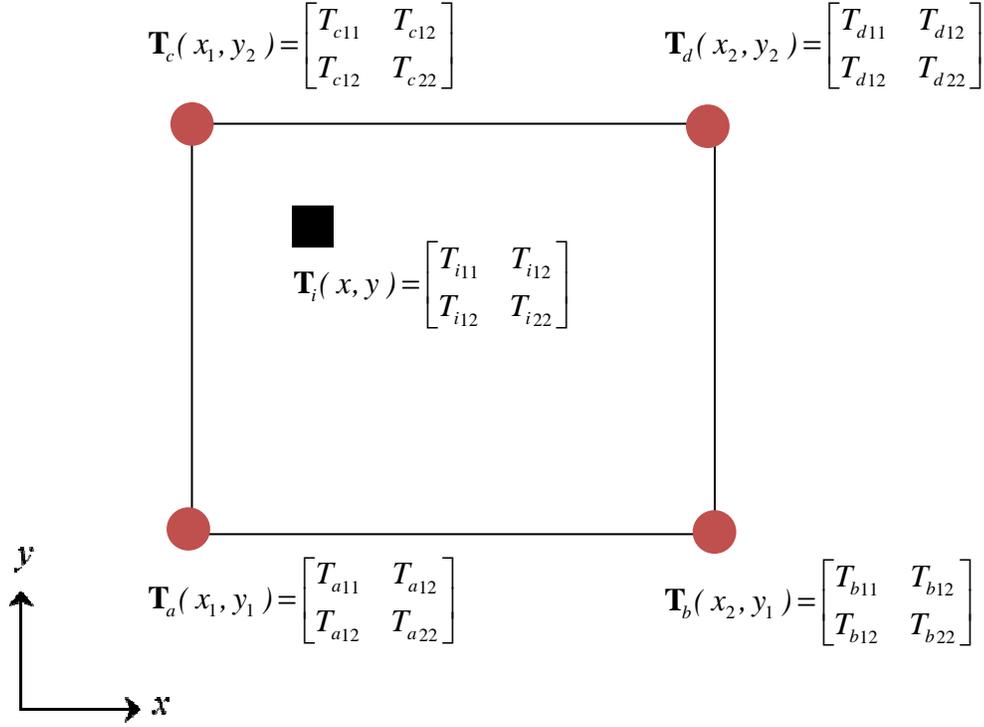


Figure 8 Rectangular grid with tensor components.

The point (x, y) is enclosed by the grid formed by the points (x_1, y_1) , (x_2, y_1) , (x_1, y_2) , and (x_2, y_2) , each with their tensor components \mathbf{T}_a , \mathbf{T}_b , \mathbf{T}_c , and \mathbf{T}_d respectively. The components of the tensor \mathbf{T}_i are obtained using bilinear interpolations as follows:

$$\mathbf{T}_{ijk} = \frac{A_{jk} + B_{jk} + C_{jk} + D_{jk}}{(x_2 - x_1)(y_2 - y_1)} \quad (17)$$

$$A_{jk} = \mathbf{T}_{a_{jk}}(x_2 - x)(y_2 - y)$$

$$B_{jk} = \mathbf{T}_{b_{jk}}(x - x_1)(y_2 - y)$$

$$C_{jk} = \mathbf{T}_{c_{jk}}(x_2 - x)(y - y_1)$$

$$D_{jk} = \mathbf{T}_{djk} (x - x_1)(y - y_1),$$

where $j = 1, 2$ and $k = 1, 2$, corresponding to the components of the tensor matrices.

Once the interpolated tensor matrix \mathbf{T}_i has been obtained, then the vectors $\mathbf{v}_1(x, y)$ and $\mathbf{v}_2(x, y)$ of the preferred direction at the interpolated point (x, y) are obtained as:

$$\mathbf{v}_1(x, y) = \begin{bmatrix} \sqrt{\mathbf{T}_{i11}} \\ \text{sign}(\mathbf{T}_{i12})\sqrt{\mathbf{T}_{i22}} \end{bmatrix} \quad (18)$$

$$\mathbf{v}_2(x, y) = -\mathbf{v}_1(x, y)$$

In this way, it is then possible to interpolate the preferred direction on any point inside the grid formed by the sampled points on the design surface.

4.1.2. PFD Pathlines

Pathlines in tensor fields are analogous to streamlines in vector fields. To obtain a streamline, the vector field function is integrated using a point as a starting point or initial condition. However, pathlines do not have only one direction at all times, but rather two, as the result of a second order symmetric tensor field having two opposing directions. The pathlines are theoretically created by taking an infinitesimal small step in any of the directions, and the directions in the tensor field are the tangents of the pathline and extended in both directions. Because a PFD field is not fitted into a function, the creation of pathlines needs to be done with a discrete (numerical) method. In a PFD field, there are two ways to create the pathlines on a given surface: a) step-marching method, and b) tensor-interpolation method. Both will be explained as follows.

The step-marching method consists on evaluating the direction of the maximum machining strip width on every step of the pathline. First, a *seed point* $\mathbf{P}_0(x, y, z)$ on the design surface is chosen as the initial point from where the pathline will grow. Then, W is evaluated at different feed directions \mathbf{f}_{CC} and the feed direction with the largest magnitude of W is chosen as the preferred direction. Two vectors are obtained, $+\mathbf{f}_{CC0}$ and $-\mathbf{f}_{CC0}$, and the pathline needs to grow on both directions. First, the direction $+\mathbf{f}_{CC0}$ is considered and a step Δs is taken into that direction. A new point is $\mathbf{P}_1(x, y, z)$ obtained, where the preferred direction is again chosen and two vectors are obtained, $+\mathbf{f}_{CC1}$ and $-\mathbf{f}_{CC1}$. These two vectors are compared with the previous feed vector $+\mathbf{f}_{CC0}$ as follows. A pair of vectors \mathbf{A} and \mathbf{B} is compared by calculating the cosine of the angle β between these two vectors. That is,

$$\cos \beta = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (19)$$

This value, $\cos \beta$, is also known as the *cosine similarity* of two vectors \mathbf{A} and \mathbf{B} and ranges from -1 to 1. If the value of the cosine similarity is positive, then that means that the vectors \mathbf{A} and \mathbf{B} are not opposing each other. If the cosine similarity is negative, it indicates that the vectors \mathbf{A} and \mathbf{B} are opposing. Since $+\mathbf{f}_{CC1}$ and $-\mathbf{f}_{CC1}$ are opposite, comparing each one with $+\mathbf{f}_{CC0}$ will yield a positive and a negative value of $\cos \beta$. The exception occurs when both are perpendicular to $+\mathbf{f}_{CC0}$. In that case, $\cos \beta$ will yield zero (since pathlines normally are continuous, very rarely $\cos \beta$ will yield a value of zero if the step size is sufficiently small). Either $+\mathbf{f}_{CC1}$ or $-\mathbf{f}_{CC1}$ is chosen whichever yields a positive cosine similarity and the next point with a step Δs is obtained. This is repeated until a border is reached. Once the pathline in the

direction of $+\mathbf{f}_{CC0}$ is finished, then the pathline in the direction of $-\mathbf{f}_{CC0}$ is grown from the original seed point $\mathbf{P}_0(x, y, z)$. This method is illustrated in Fig. 9.

The other method to create the pathlines is the tensor-interpolation method. As in the case of the step marching method, the pathline grows from the seed point $\mathbf{P}_0(x, y, z)$. The only difference from the step-marching method is that instead of evaluating different values of W at feed directions and finding the preferred feed direction at each point, the two vectors at each new point are obtained using the bilinear interpolation using the four surrounding points of the grid where each point of interest is located. This is repeated until the boundaries of the grid are reached. In order to extend the pathline beyond the grid, the tangent of the pathline is used.

The main difference between both methods to generate pathlines in a design surface's tensor field is the number of iterations and complexity of the functions needed. The step-marching method requires finding the value of W at various feed directions, all requiring solving a system of non-linear equations. On the other hand, the tensor-interpolation method requires only linear interpolation and therefore requires far fewer operations. The tensor-interpolation method is applied when finding the principal tool path explained in section 4.3.2 because a large number of pathlines need to be obtained to cover the region of interest. It should be noted, however, that a grid of sampled points with a low resolution will lead to greater errors if the tensor-interpolation method is used. As shown in Fig. 10, both methods provide a similar pathline when using an adequate resolution and step size. A collection of pathlines covering a sample surface is shown in Fig. 11.

As mentioned earlier, if the pathlines of maximum W are used as the machining paths, overlapping of machining strip widths generally occur, leading to redundant machining. Fig. 12 shows overlapping of W when sample pathlines are used as machining paths.

A pathline may also be used as the initial tool path from which iso-scallop tool paths may be generated. However, as it can be observed in Fig. 13, the adjacent tool paths eventually begin to differ from the pathlines of preferred direction. This effect is known as *iso-scallop drift* and will be explained in section 4.3.1. The need for segmentation according to similar PFD pathlines came from the concept that a single initial tool path for an entire surface may not be the best solution.

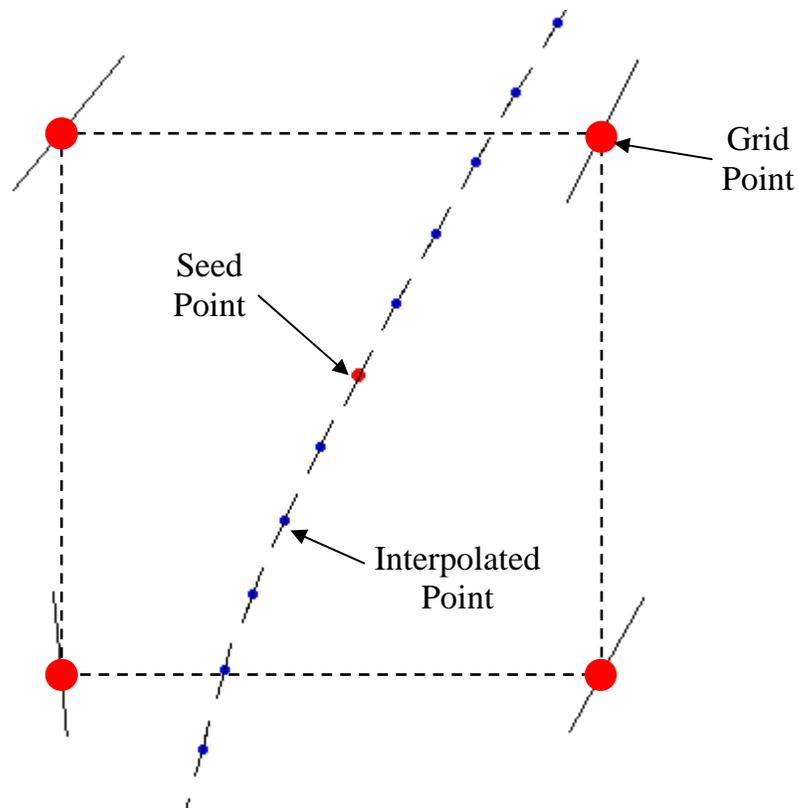
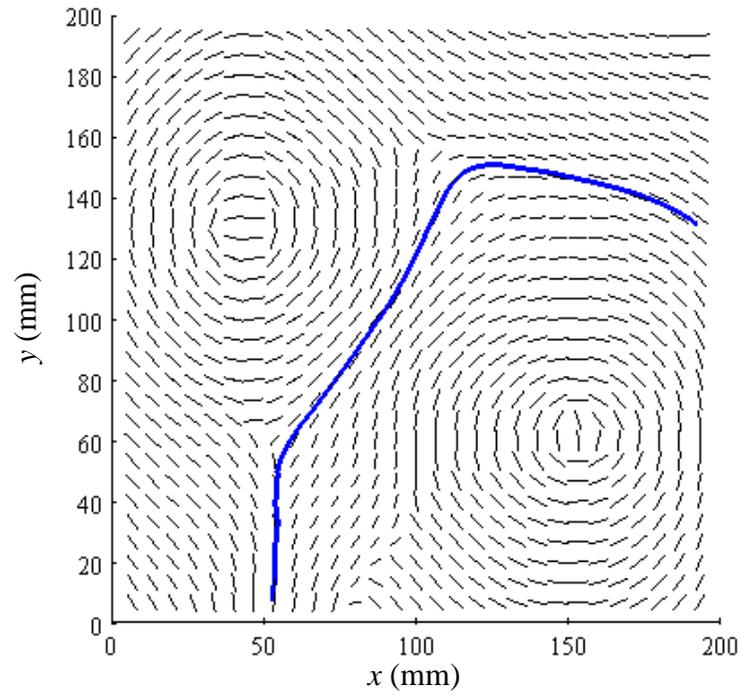
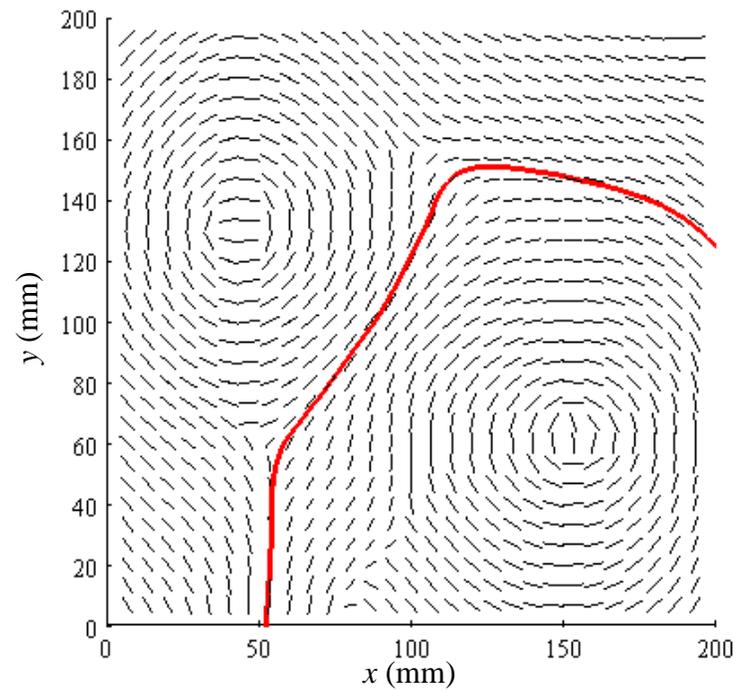


Figure 9 Illustration of the bilinear interpolation method.



(a)



(b)

Figure 10 Comparison between pathlines (step = 1 mm) created by a) the tensor-bilinear method (no extension beyond grid) and b) using the step-marching method.

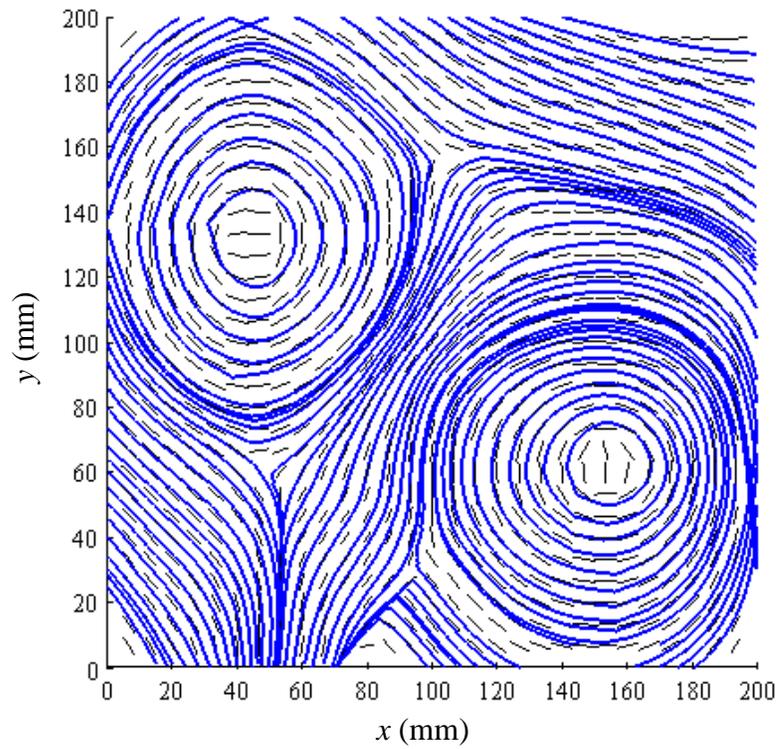
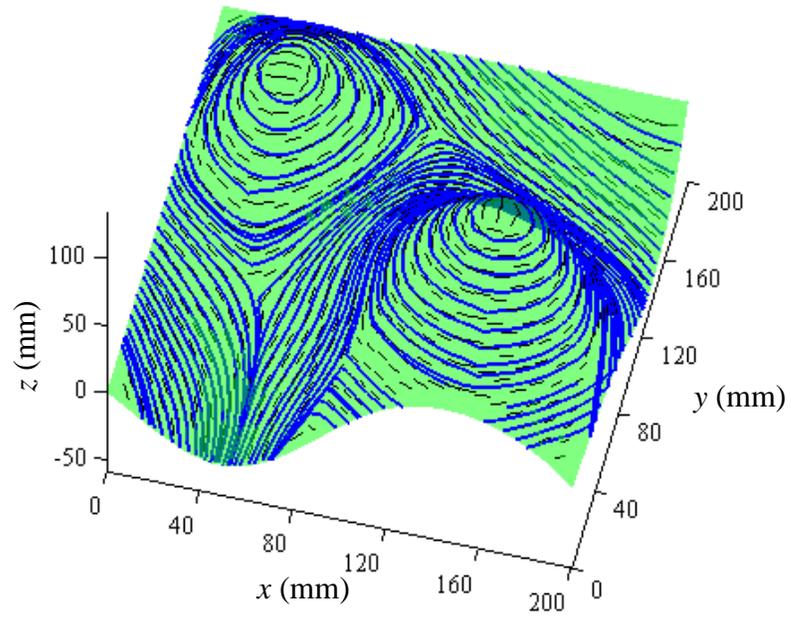


Figure 11 Sample surface covered with pathlines.

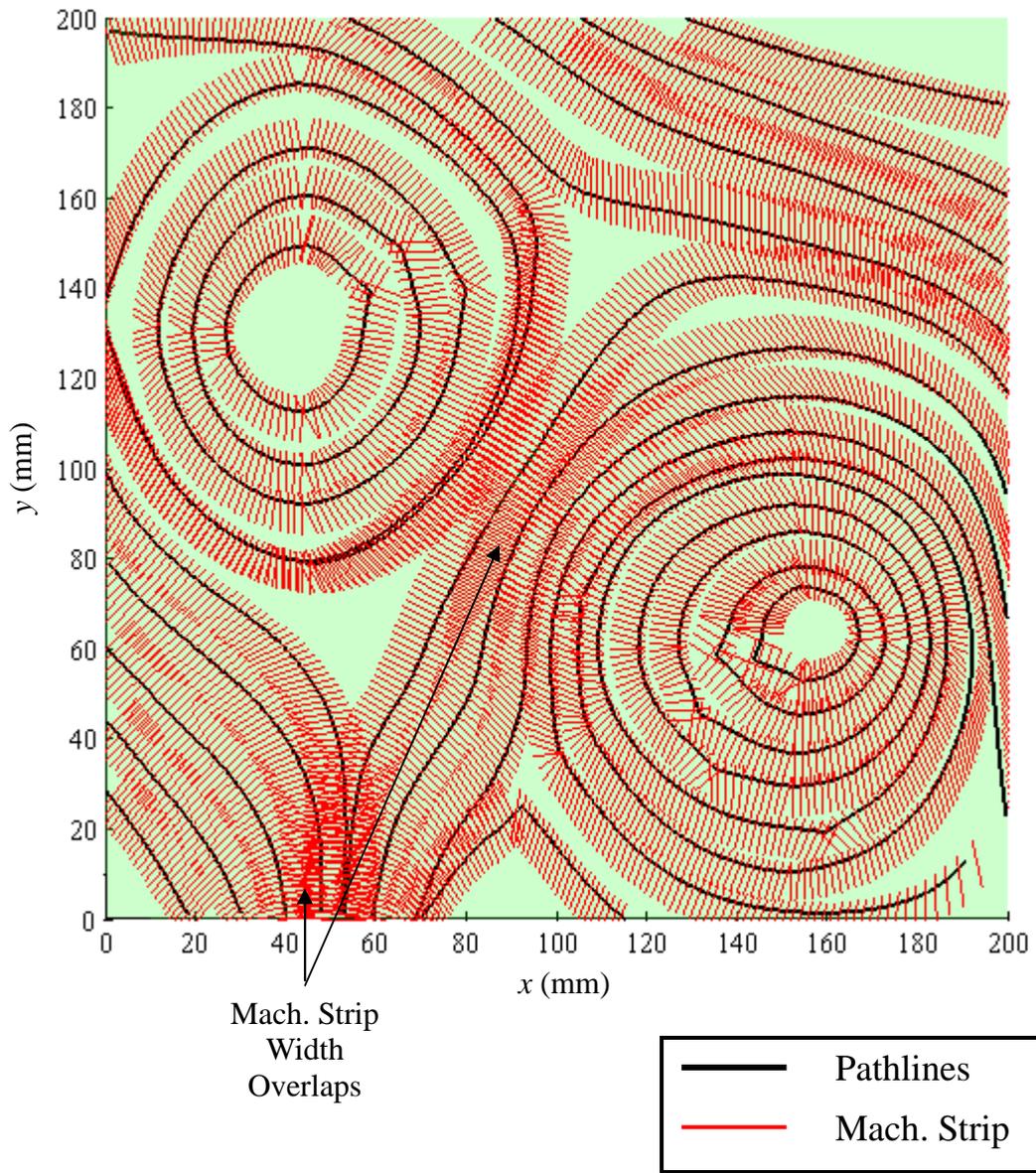


Figure 12 Illustration of overlaps of sample pathlines' machining strip width.

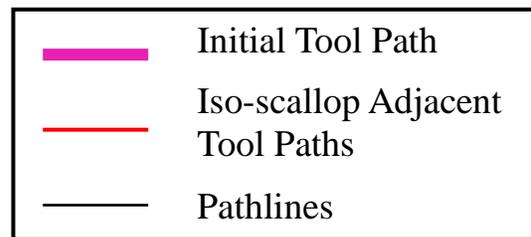
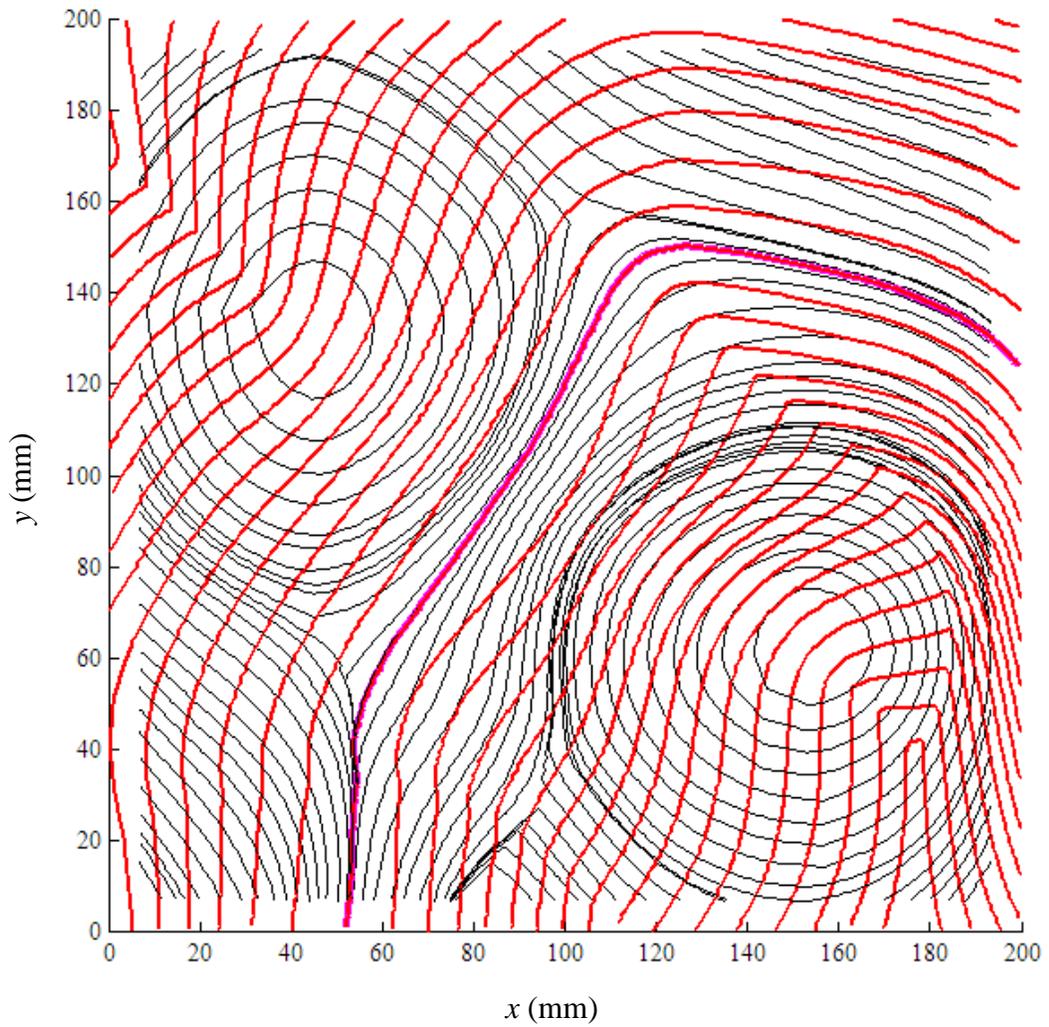


Figure 13 Comparison between the iso-scallop tool paths generated from one of the sample design surface pathlines and its PFD field pathlines.

4.1.3. Detection of Degenerate Points

Pathlines, similar to the case of streamlines in the vector field, do not intersect each other except at degenerate points. These points are the analogous of critical points in vector fields. As explained in [27], at a degenerate point $\mathbf{P}_{degen}(x, y)$ the eigenvalues of the tensor $\mathbf{T}(\mathbf{P}_{degen}(x, y))$ are equal. That is, $\lambda_1(\mathbf{P}_{degen}(x, y)) = \lambda_2(\mathbf{P}_{degen}(x, y))$. The tensor at a degenerate point has the form:

$$\mathbf{T}(\mathbf{P}_{degen}(x, y)) = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \quad (20)$$

From eq. (2), the tensor's components satisfy these conditions:

$$\mathbf{T}_{11}(\mathbf{P}_{degen}(x, y)) - \mathbf{T}_{22}(\mathbf{P}_{degen}(x, y)) = 0 \quad (21)$$

$$\mathbf{T}_{12}(\mathbf{P}_{degen}(x, y)) = 0 \quad (22)$$

Mathematically, this means that any unit eigenvector \mathbf{e}_i can be assigned to the value of λ such that $\mathbf{T}(\mathbf{P}_{degen}(x, y))\mathbf{e}(\mathbf{P}_{degen}(x, y)) = \lambda(\mathbf{P}_{degen}(x, y))\mathbf{e}(\mathbf{P}_{degen}(x, y))$ is fulfilled. That means that at the degenerate point, there is an infinite number of unit eigenvectors that can describe the tensor.

In the case of 3-axis machining, a degenerate point on the design surface corresponds to a CC point where there is not one single preferred direction, because all directions will be the preferred. Ideally, the value of W will be the same for all directions at a degenerate point. This happens, for example, when the surface at that point is completely planar. A completely plane will not have any preferred direction, since W will be the same for any direction at any point.

In that case, it could be said that all points in a surface are degenerate points. A simple method to detect the vicinity of degenerate points is to find the regions of the design surface where the difference between the maximum value of W , W_{max} , and the minimum one, W_{min} , is close to zero. Fig. 14 illustrates a comparison between the pathlines and the difference between W_{max} and W_{min} on a sample design surface.

Degenerate points are detected using bilinear interpolation of the grid of sampled points with their respective tensor matrix. The region enclosed by a grid of four sampled points is evaluated in very small discrete steps. If at a point inside the grid the relative values of equations (21) and (22) are below a threshold (very close to zero), then a degenerate point was detected. For an improved accuracy, the grids where possible degenerate points exist are further divided in sub-grids from where the tensor components are obtained by finding the preferred direction. These sub-grids are analyzed in the same way as the original grid to locate the degenerate points.

Degenerate points can be further divided in two main categories according to the characteristics of their surrounding pathlines: *trisectors* and *wedges*. A trisector point has three *separatrices* and a wedge point has only one *separatrix*. A *separatrix* is a pathline that intersects a degenerate point or is infinitely close to it, analogous to the asymptotes of a hyperbola, with the center of the hyperbola being the degenerate point. However, in a tensor field, a trisector point has three sectors of pathlines can be differentiated from their separatrices and a wedge point contains one single “fold” from the pathlines towards the separatrices. In a vector field, the occurrence of these three-sector or fold shapes from streamlines is impossible, since that would mean that vector field streamlines would have

opposing streamline directions. Fig. 15 illustrates a trisector point with its separatrices and Fig. 16 illustrates a wedge point.

The type of degenerate point is identified by analyzing the partial derivatives of the tensor components of surrounding points. Consider the following coefficients:

$$a = \frac{1}{2} \frac{\partial(T_{11} - T_{22})}{\partial x} \quad b = \frac{1}{2} \frac{\partial(T_{11} - T_{22})}{\partial y} \quad (23,24)$$

$$c = \frac{\partial T_{12}}{\partial x} \quad d = \frac{\partial T_{12}}{\partial y} \quad (25,26)$$

For a discrete case, δx and δy can be substituted by a very small discrete step Δx and Δy . The tensor components are evaluated at points with these steps and the derivatives can be approximated.

A characteristic derived from these coefficients is the coefficient δ , evaluated as:

$$\delta = ad - bc \quad (27)$$

When the value of δ is negative ($\delta < 0$), then the surrounding pathlines have a pattern consisting of three hyperbolic sectors divided by three separatrices. This indicates that the degenerate point is a trisector point. When the value of δ is positive ($\delta > 0$), the surrounding pathlines have a pattern of one hyperbolic sector, divided by one separatrix, indicating that the degenerate point acts as a wedge point. Mathematical proofs of these properties are explained in detail in the referenced material [27-28].

A special case of degenerate point occurs when $\delta = 0$. This situation indicates that there are two degenerate points that are very close and were detected as a single degenerate point. When

two trisector points are very close, the separatrix that joins these two points is very small and it results in a pattern that gives the impression of four distinct regions. In that case, the two trisector points are treated separately. When a pair of wedge points merge or are very close, the surrounding pathlines create circular concentric loops, and are treated as a separate case of degenerate point, known as the *merged wedge point*, as shown in Fig. 17. Once the degenerate points are located and their type identified, the next step is to evaluate the separatrices of each of the degenerate points.

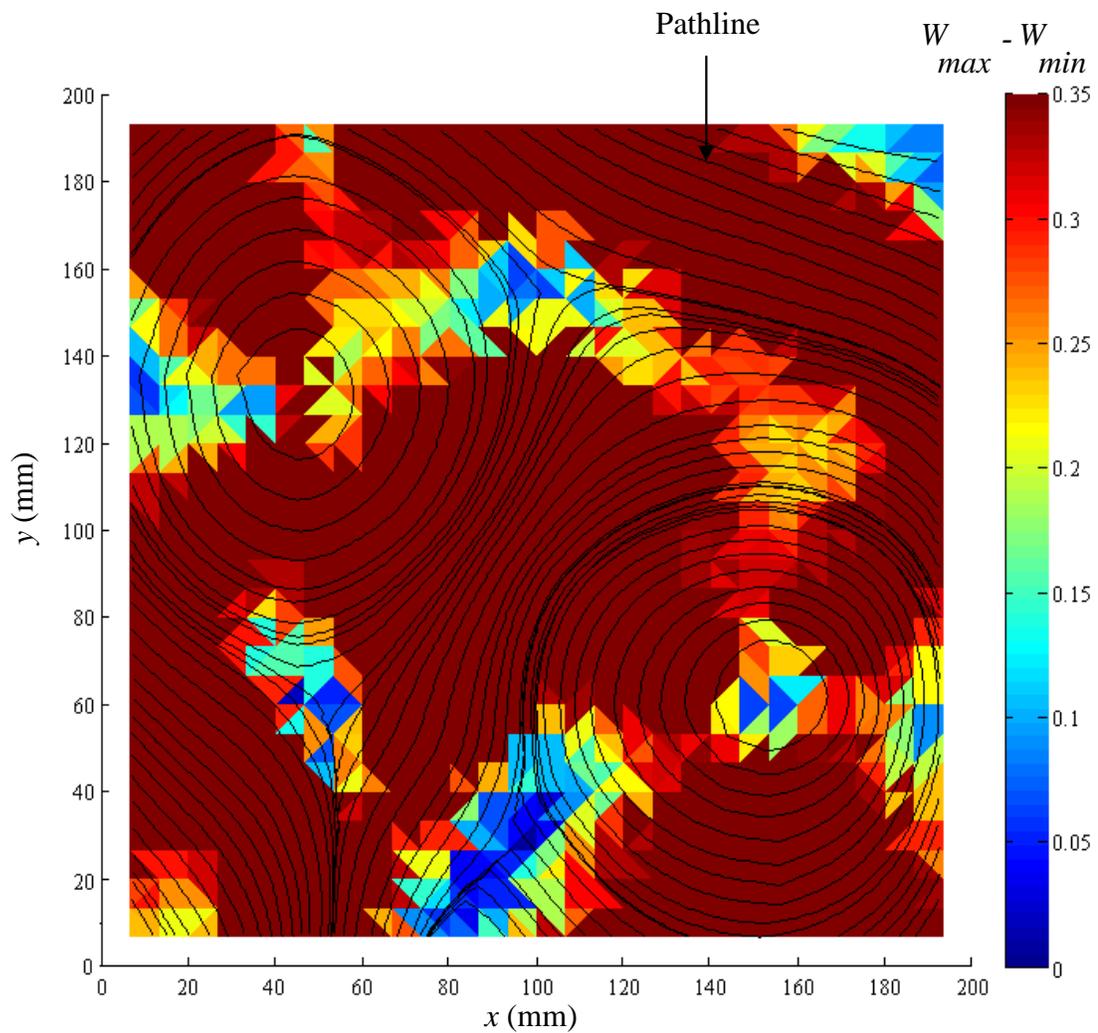


Figure 14 Comparison between a design surface’s PFD pathlines and the difference between W_{max} and W_{min} at sample points.

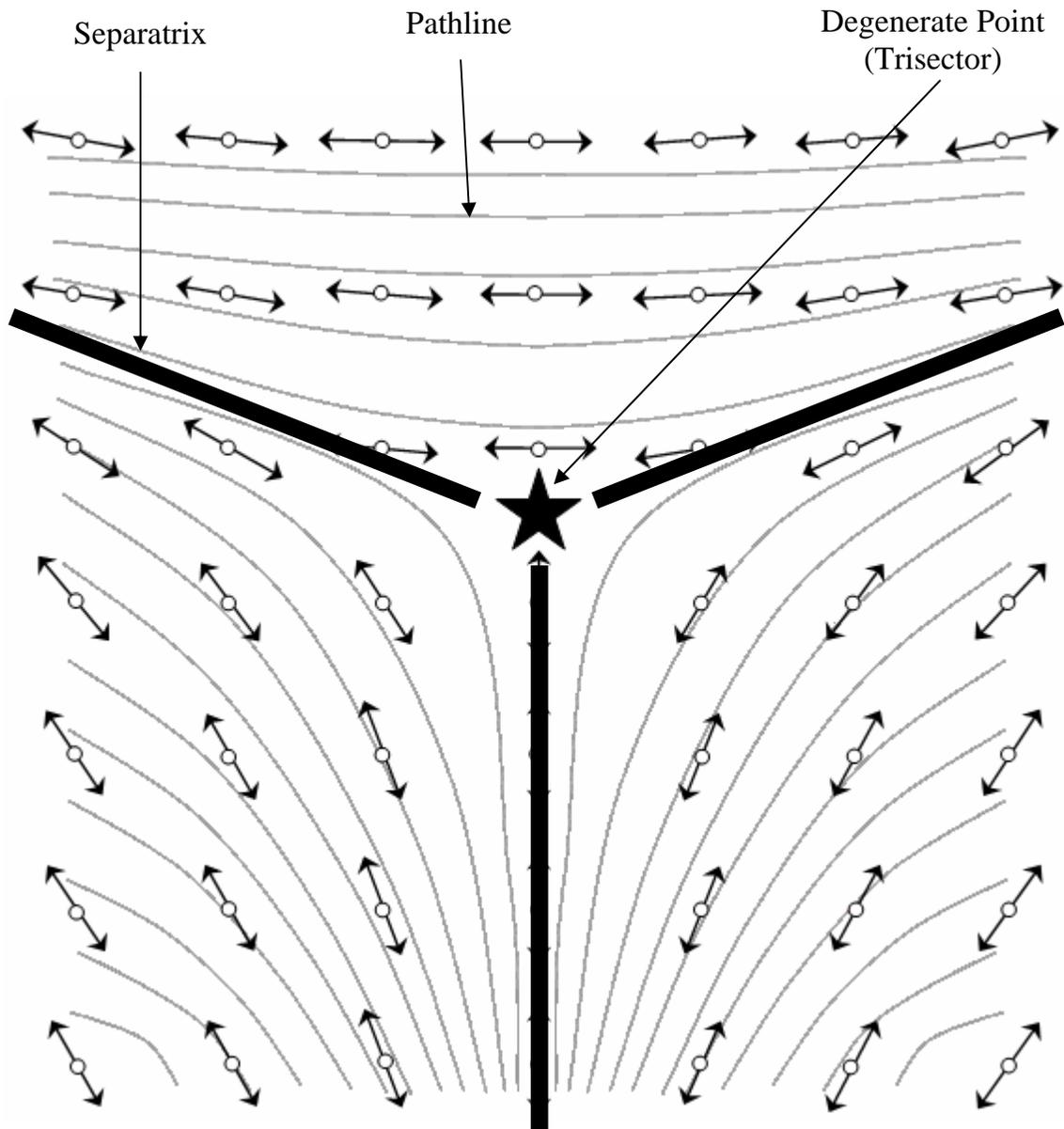


Figure 15 Illustration of a Trisector Degenerate Point.

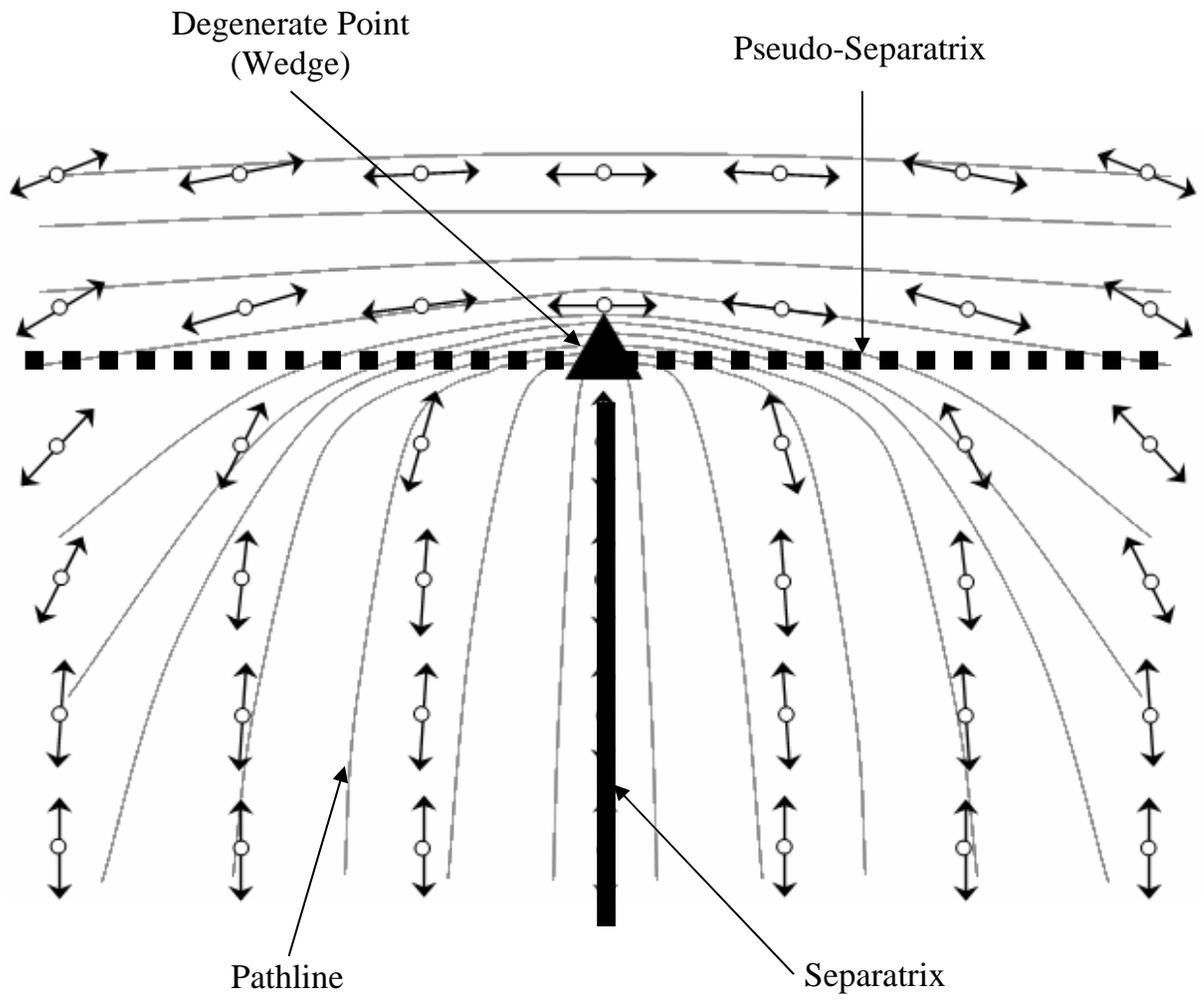


Figure 16 Illustration of a Wedge Degenerate Point.

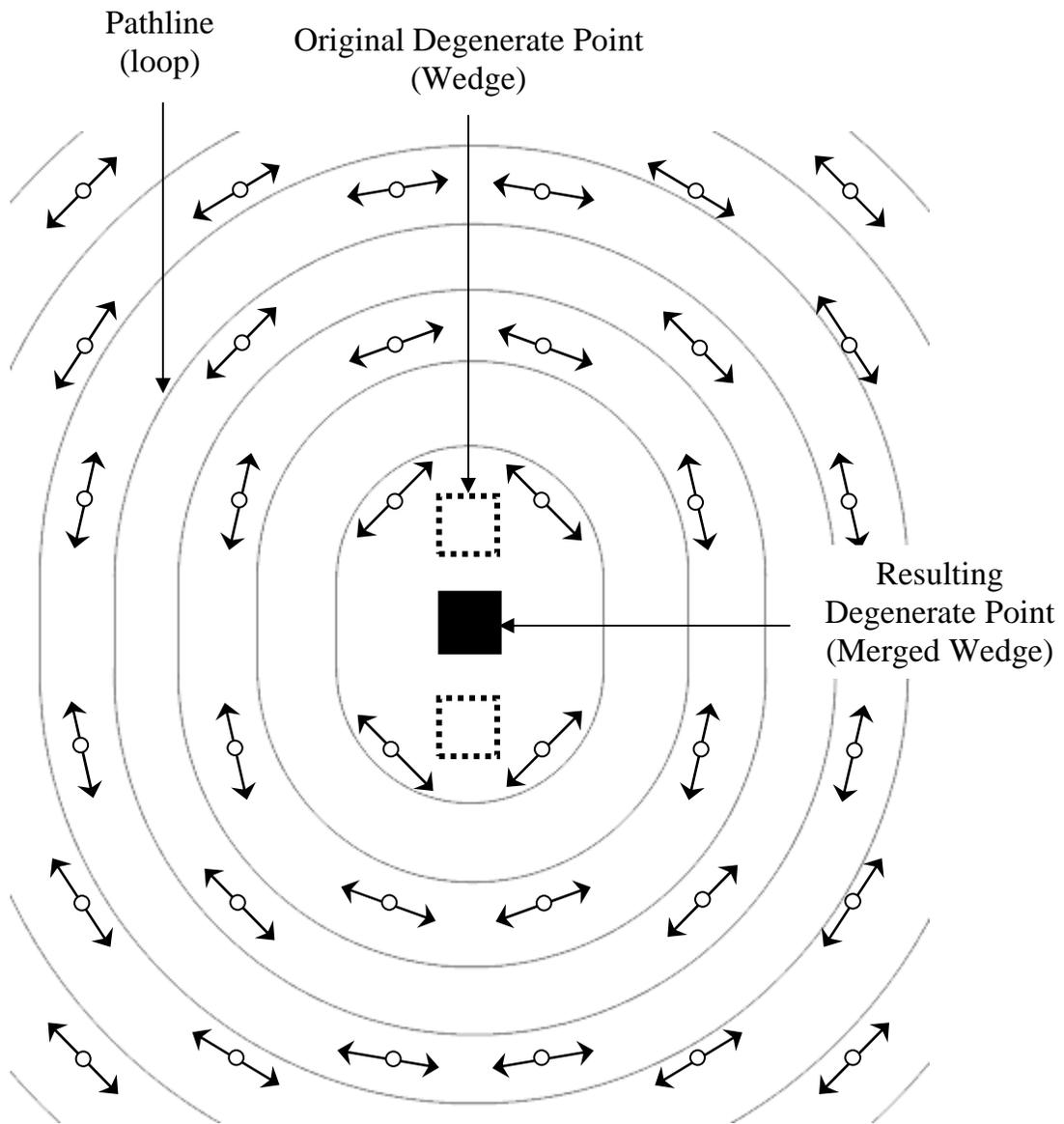


Figure 17 Illustration of a Merged Wedge Degenerate Point.

4.2. Segmentation

The separatrices of degenerate points are pathlines that grow until a border or another degenerate point is reached [27-29]. The reason for segmenting a surface according to the separatrices of degenerate points is that the separatrices form a skeleton of the PFD field that segments the entire surfaces into regions where the pathlines are similar or uniform. At the location near the degenerate points, the flow represented by the surface pathlines present a behavior that can be described as a bifurcation (trisectors), fold (wedges), or looping (merged wedges). Iso-scallop tool paths share the characteristic of having a uniform flow based on the initial tool path. In this section, the segmentation according to the type of degenerate point will be explained.

4.2.1. Trisector Points

As mentioned earlier, trisector points have three separatrices. Since separatrices are actually pathlines that intersect degenerate points, these can be evaluated by placing a seed point and let it flow in both directions. To find the appropriate seed points that will create the separatrices at the trisector degenerate point $\mathbf{P}_{tris}(x, y, z)$, the following method is proposed.

The trisector point $\mathbf{P}_{tris}(x, y, z)$ treated as a CC point and is projected onto the point's feed plane (explained in section 2.1). A two-dimensional circle with a small radius is constructed on the x - y plane around the degenerate trisector point. The circle is made of sampled points P_i . Since there is a unique z coordinate for each x - y coordinate, every point on the projected x - y plane can be assigned a point on the three-dimensional design surface and therefore their preferred direction can be calculated. A vector \mathbf{v}_i with the degenerate point as the origin is built for each of the circle's points and then projected on the feed plane of the trisector point. At

each point P_i , the preferred direction, composed of two opposing vectors $+\mathbf{v}_{pfdi}$ and $-\mathbf{v}_{pfdi}$, is calculated and projected on the feed plane. The three vectors, \mathbf{v}_i , $+\mathbf{v}_{pfdi}$, and $-\mathbf{v}_{pfdi}$, now lie on the feed plane of the degenerate point $\mathbf{P}_{tris}(x, y, z)$ and \mathbf{v}_i is compared with $+\mathbf{v}_{pfdi}$ and $-\mathbf{v}_{pfdi}$. Two vectors are compared using the cosine of the angle β between these two vectors, as explained in section 4.1.2.

For each vector \mathbf{v}_i there will be a positive and negative value of $\cos \beta$ and the negative value is discarded. Since \mathbf{v}_i was projected onto the feed plane, a feed angle ϕ can be assigned to each \mathbf{v}_i . When comparing the values of $\cos \beta$ vs. ϕ , in the case of a trisector there will be three points where $\cos \beta$ equals 1 (or very close to 1). Each of the three feed angles with the $\cos \beta$ equals 1 is assigned a feed vector and therefore a subsequent CC point following the feed vector can be determined. These CC points are used as the seed points for the separatrices. This method is illustrated in Fig. 18.

Each of the separatrices' points are obtained using either a marching method, that determines the preferred direction for each new point evaluated, or using a bilinear interpolation method (explained in section 3.1.) until a border is reached or the evaluated point becomes very close to another degenerate point.

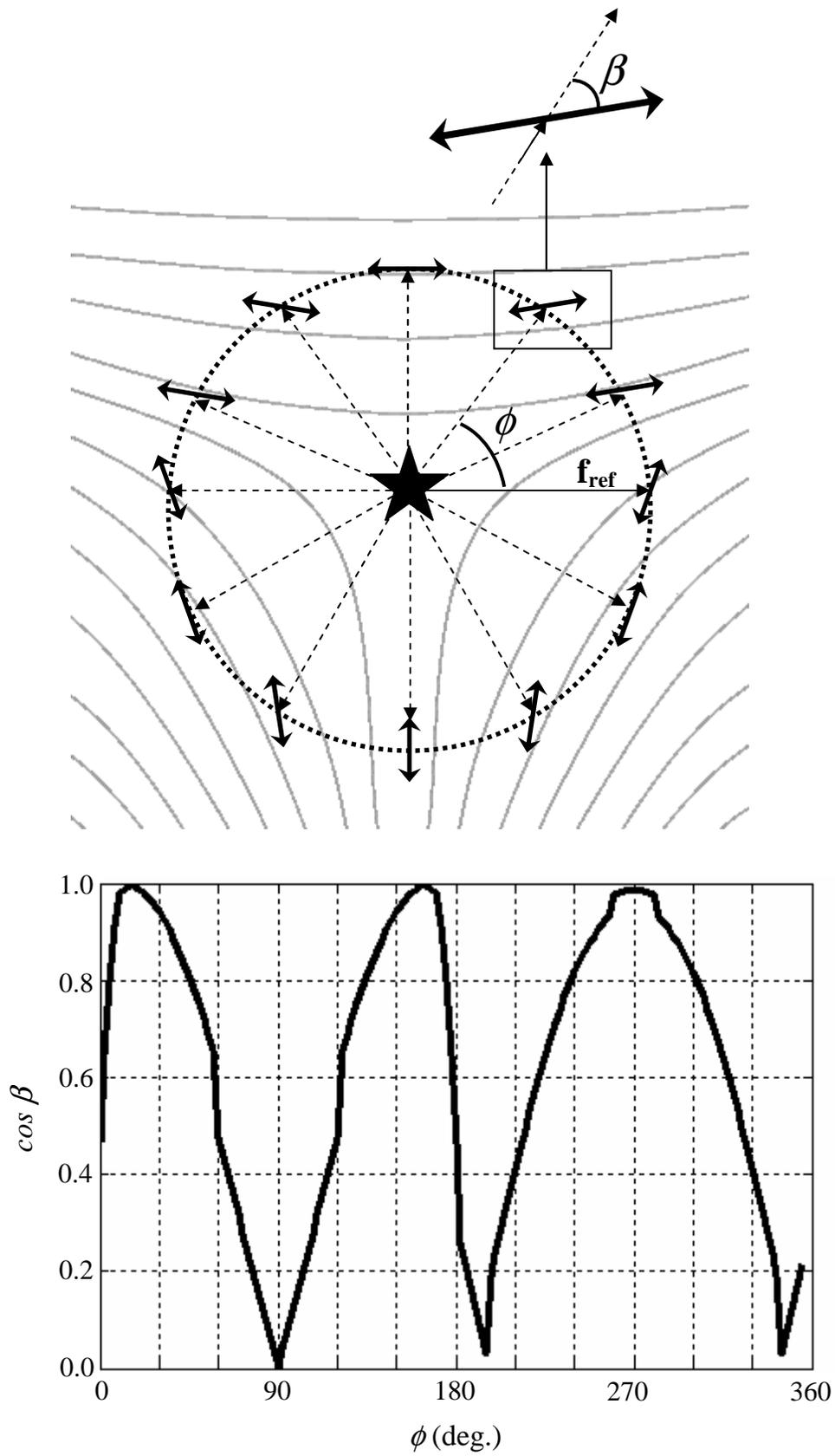


Figure 18 Evaluation of Separatrices in a Trisector Point.

4.2.1. Wedge Points

As pointed out earlier, wedge degenerate points have only one separatrix. The seed point for the wedge separatrix is determined in a similar manner as the seed points for the trisector separatrices. The only difference is that there will be only one point with value for $\cos \beta$ equaling 1 (or very close to 1) when compared with the feed angle ϕ and therefore, there will be only one seed point. The separatrix derived from the seed point is one finite curve on a surface and therefore it cannot segment the surface into different regions. For that reason, in the case of wedge points, it is proposed to add another boundary that will be known as a *pseudo-separatrix*. This special kind of separatrix is also intended to separate the surface into regions with similar flow, as in the case of regular separatrices. The prefix *pseudo* is used because it will not be a pathline.

The *pseudo-separatrix* at the wedge degenerate point $\mathbf{P}_{wedge}(x, y, z)$ is determined as follows. Once the original separatrix of the wedge point $\mathbf{P}_{wedge}(x, y, z)$ is obtained, the three-dimensional tangent vector of this separatrix \mathbf{v}_{tang} is computed at the wedge point $\mathbf{P}_{wedge}(x, y, z)$. The *wedge plane* is obtained by using the tangent vector \mathbf{v}_{tang} as its normal at the wedge point $\mathbf{P}_{wedge}(x, y, z)$. The design surface $\mathbf{S}_{design}(u, v)$ is then intersected by the wedge plane and the resulting curve is the *pseudo-separatrix* of the wedge point $\mathbf{P}_{wedge}(x, y, z)$. If a design surface has only one wedge point, its separatrix and the pseudo-separatrix, that intersect at the wedge point, will divide the surface into three regions. Fig. 19 illustrates the wedge point and its separatrix and pseudo-separatrix.

It is important to note that, since pseudo-separatrices are not pathlines, they can intersect with other separatrices. In that case, segments of pseudo-separatrices should be trimmed at those

intersections especially if these segments reach other regions that were previously determined by other separatrices.

4.2.3 Merged Wedge Points

In a PFD field where there are only two degenerate wedge points and the method to obtain their separatrix and pseudo-separatrice is applied, the result would ideally be a separatrix joining those two points and two pseudo-separatrices. Their surrounding pathlines show a pattern of non-intersecting closed curves, or loops. As the distance between two wedge points approximate zero, the pathline loops take a more circular shape if projected into the x - y plane. If the distance between the two wedge points is infinitely small, then the length of the separatrix would also be nearing zero and the two pseudo-separatrices would unite into a single pseudo-separatrix.

If two degenerate wedge points are merged into one degenerate point at the middle of the original wedge points, and the method described in 3.2.2 is applied, it will be observed that the value of $\cos \beta$ will remain nearly constant at a value approximate to zero as shown in Fig. 20. Because basically there are no maxima when comparing $\cos \beta$ vs. ϕ , there is no preferred direction that might be used to determine a separatrix seed point because a seed point at any preferred direction would just become a point in a loop. For this reason, there is no need to evaluate separatrices for merged wedge point. In the case of a design surface whose PFD field indicate the presence of only one merged point, the surface will not be segmented further and will be treated as a single region. In the region where a merged wedge point or a pair of wedge points exists, it is helpful to consider loop pathlines as closed curves and it also further helps to use a loop as an initial tool path rather than a discontinuous curve. It should be noted that, since a discrete-numerical method is used to create the pathlines, the loops may not close

completely, since errors might accumulate while creating the pathline. In that case, a simple smoothening algorithm is applied so that the pathline creates a closed curve that can be used for further analysis, such as the initial tool path. The algorithm is explained as follows:

Let $\mathbf{C}(t)$ be a finite curve with $\mathbf{P}_{start}(x(t), y(t), z(t))$ and $\mathbf{P}_{end}(x(t), y(t), z(t))$ as its end points and $\mathbf{P}_{mid}(x(t), y(t), z(t))$ is a point in the middle of $\mathbf{C}(t)$. If the distance between its ends is under a threshold, then it is considered a loop. Let $\tilde{\mathbf{C}}(t)$ be the projection of $\mathbf{C}(t)$ into the x - y plane, and contains the projection of $\mathbf{P}_{start}(x(t), y(t), z(t))$, $\mathbf{P}_{end}(x(t), y(t), z(t))$, and $\mathbf{P}_{mid}(x(t), y(t), z(t))$ into $\mathbf{P}_{start}(x(t), y(t))$, $\mathbf{P}_{end}(x(t), y(t))$, and $\mathbf{P}_{mid}(x(t), y(t))$, respectively. A middle point $\mathbf{P}_{s-e}(x, y)$ is created between $\mathbf{P}_{s-e}(x, y)$ and $\mathbf{P}_{start}(x, y)$. The distance between $\mathbf{P}_{s-e}(x, y)$ and $\mathbf{P}_{start}(x, y)$ is projected along the normal vector of $\tilde{\mathbf{C}}(t)$ at $\mathbf{P}_{start}(x, y)$, creating the distance $dnorm_{s-e/start}$. All points on $\tilde{\mathbf{C}}(t)$ between $\mathbf{P}_{mid}(x(t), y(t))$ and $\mathbf{P}_{start}(x(t), y(t))$ are translated in their normal direction in a linear magnitude from zero to $dnorm_{s-e/start}$, that is, the point $\mathbf{P}_{mid}(x(t), y(t))$ is not moved and the rest of the points between $\mathbf{P}_{mid}(x(t), y(t))$ and $\mathbf{P}_{start}(x(t), y(t))$ are translated along their normal gradually until the point $\mathbf{P}_{start}(x(t), y(t))$ is reached, which is translated $dnorm_{s-e/start}$ by along its normal. The same is operation is repeated in the case of $\mathbf{P}_{end}(x(t), y(t))$. Then, since every point in the x - y plane has its unique design surface z value, they are translated into the design surface.

The design surface can be segmented into regions of interest once all degenerate points with their respective separatrices have been identified. An example of a segmented surface is shown in Fig. 21.

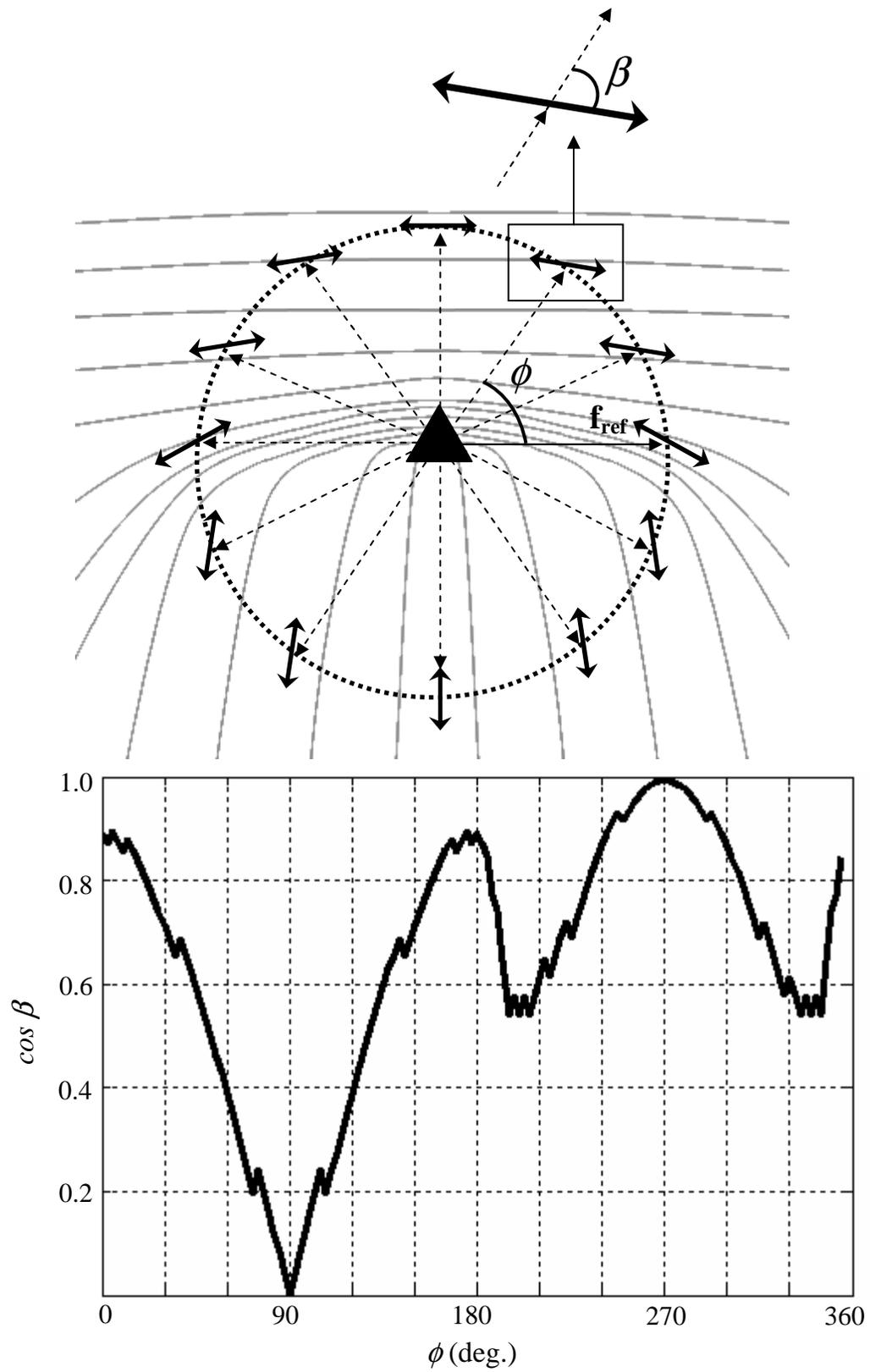


Figure 19 Evaluation of Separatrices in a Wedge Point.

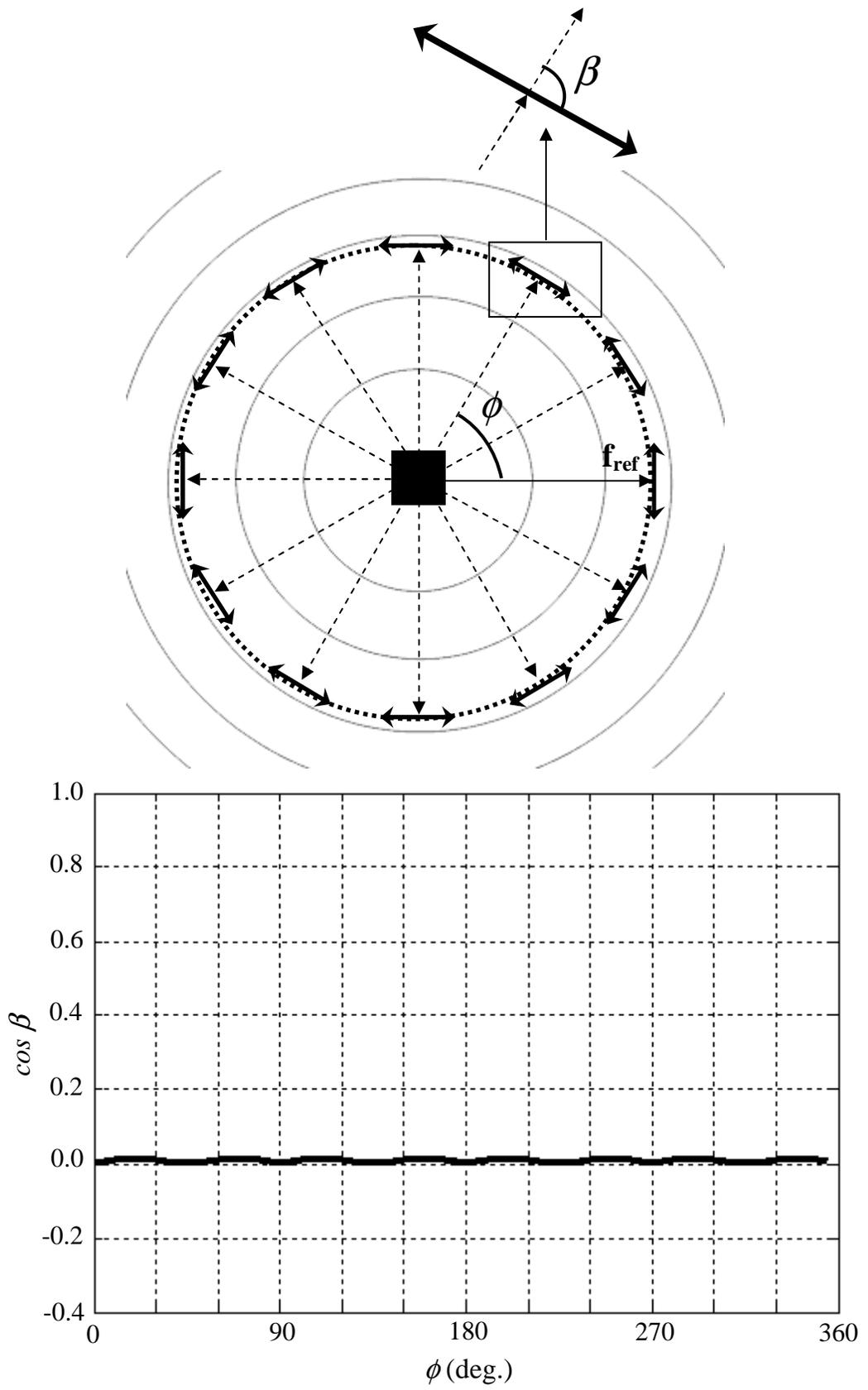


Figure 20 Illustration of the surrounding pathlines of a Merged Wedge Point.

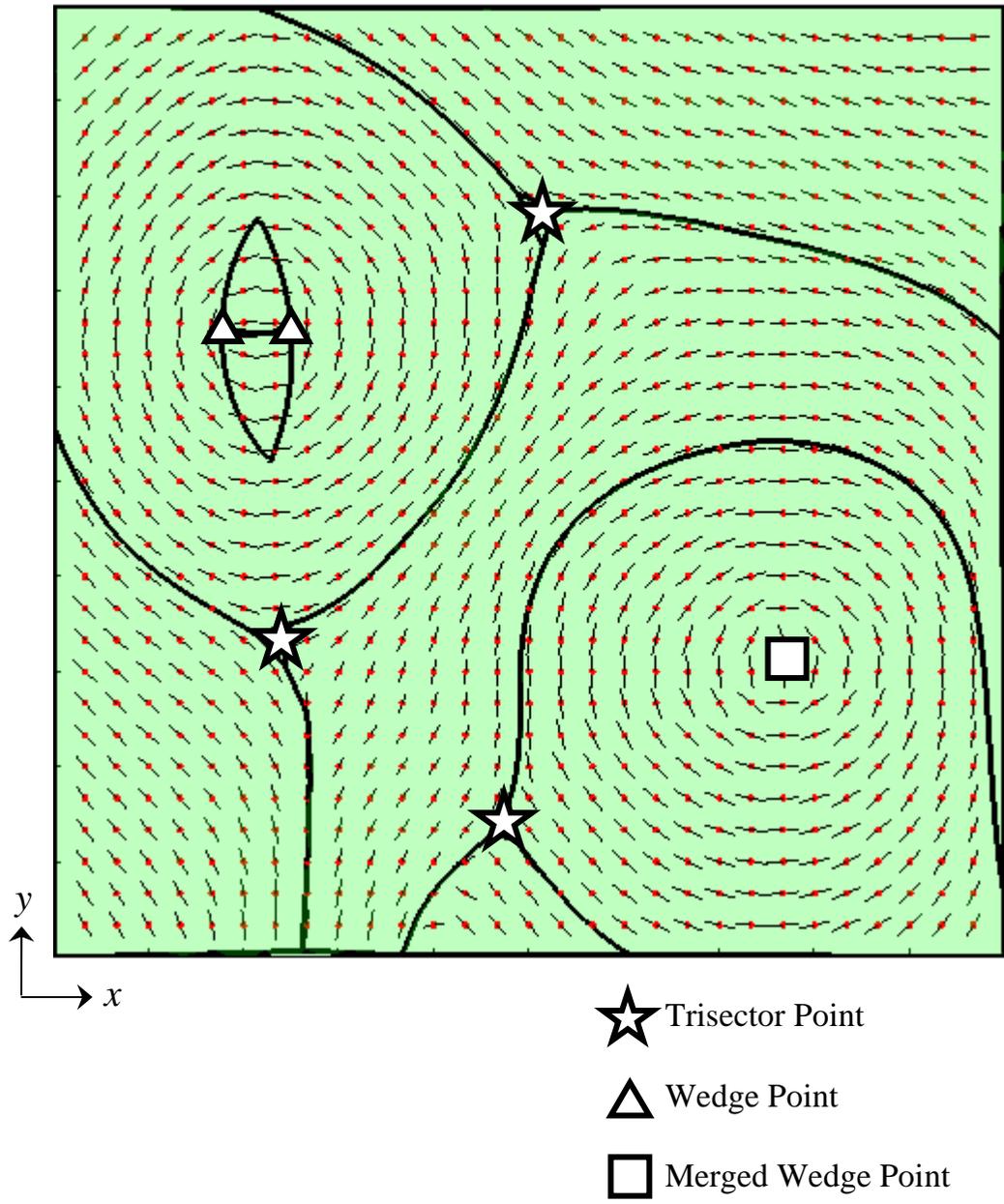


Figure 21 Illustration of a segmented surface and its degenerate points.

4.3. Principal Tool Path Evaluation

Once the design surface has been segmented into regions containing similar or uniform flows, the next step is to cover these regions with machining tool paths using the iso-scallop tool path method. However, this method still requires the input of an initial tool path to generate the adjacent tool paths. In this section, a method to determine the initial tool path based on the region's pathlines is proposed. This initial toolpath, called the *principal tool path*, is chosen in an attempt to reduce the effect of *iso-scallop drift*, and is explained in the following section.

4.3.1. Iso-scallop Tool Path Drift

CC paths defined from the pathlines of the PFD field will seldom share the property of tool paths obtained from the iso-scallop tool path generation, which is that adjacent tool paths will share a common scallop curve. In the iso-scallop tool path generation method, sequential adjacent tool paths are determined based on an initial tool path. The adjacent tool paths are created on both sides of the initial tool path as mentioned in section 3. If an initial tool path is chosen from one of the boundaries of the segmented regions, it will be frequently observed that the adjacent tool paths generated by the iso-scallop method will tend to stop following the pathline pattern of a region as more adjacent tool paths are generated. The direction followed by the adjacent iso-scallop tool paths will gradually be more different than the preferred direction. As a consequence, the actual machining strip width at the generated iso-scallop tool path points will be less than the maximum machining strip width that would be obtained if the preferred direction was followed. The deviation of the feed direction of the generated iso-scallop tool paths from the preferred feed direction is known as *iso-scallop drift*. A method to determine the initial tool path that can generate adjacent tool paths with reduced iso-scallop drift is proposed and explained in the next section.

4.3.2. Principal Tool Path Evaluation Procedure

The purpose of the principal tool path is to serve as the initial tool path whose adjacent iso-scallop tool paths present as little iso-scallop drift as possible. In the iso-scallop tool path generation method, adjacent tool paths are generated from both sides of the initial tool path until a border is reached. The adjacent tool paths are sequential, that is, the next tool path depends on the previous tool path generated. If an initial tool path is chosen from one of the pathlines of the PFD field, generally the more adjacent tool paths are generated from one of the two sides of the initial tool path, the more deviation the generated tool paths will have from the PFD pathlines until a border is reached.

The concept of the principal tool path stems from the need of a pathline whose adjacent tool paths cover the entire region of interest with as little adjacent growth from either side. It is proposed to use a middle pathline of the region to act as the initial tool path. In order to identify the middle pathline from a set of pathlines covering a region, a measurement to compare curves is needed.

A surface region is covered by a set of pathline curves F_n by creating a uniform grid of seed points from which pathlines are generated with the method explained in section 4.1.2. All the pathlines created in a region will act as candidate pathlines for the initial tool path. A sufficient amount of seeds is required so that the principal tool path can represent the middle of a greater number of possible candidate pathlines. The distance between seed points should be small enough so that the region of interest is covered by a sufficiently high number of pathlines.

A pair of curves can be compared using a distance measurement called the Hausdorff distance (d_H). Typically, the distance between two curves **A** and **B** is calculated as the

minimum distance between the set of points in curve **A** and the set of points in curve **B**. In contrast, the Hausdorff distance between curve **A** and curve **B** is the maximum of the minimum distances computed for every combination of the points in curve **A** with the points in curve **B**. The mathematical definition is the following [30]. Let **A** be a curve made of points **a** and **B** a curve made of points **b**:

$$d_H(\mathbf{A}, \mathbf{B}) = \max\{ \max_{\mathbf{a} \in \mathbf{A}} \min_{\mathbf{b} \in \mathbf{B}} d(\mathbf{a}, \mathbf{b}), \max_{\mathbf{b} \in \mathbf{B}} \min_{\mathbf{a} \in \mathbf{A}} d(\mathbf{a}, \mathbf{b}) \} \quad (28)$$

The Hausdorff distance d_H is illustrated in Fig. 22.

Each pathline \mathbf{F}_k is compared with every other pathline \mathbf{F}_m (such that $m=1, 2, \dots, n, m \neq k$) and their respective Hausdorff distance $d_{H\ k,m}(\mathbf{F}_k, \mathbf{F}_m)$ is computed. Then, the mean Hausdorff distance of all pathlines \mathbf{F}_m is calculated and assigned to its corresponding pathline \mathbf{F}_k . The principal toolpath is the pathline \mathbf{F}_k is the one that has the minimum mean Hausdorff distance compared to the rest of the pathlines in the same region. This results in a pathline that lies in the middle of two bordering pathlines if the density of the sample pathlines is such that pathlines cover most of the surface of the region of interest. It should be noted that this procedure can also be applied to closed curves (loops).

Fig. 23 illustrates an example of a principal tool path in a region covered by pathlines. Fig. 24 compares the adjacent iso-scallop tool paths generated from the border of a region and the ones generated from the principal tool path along with the region's pathlines. The iso-scallop tool paths generated using the principal tool path as the initial tool path follow the preferred direction pathlines more closely than the ones generated from the region's border.

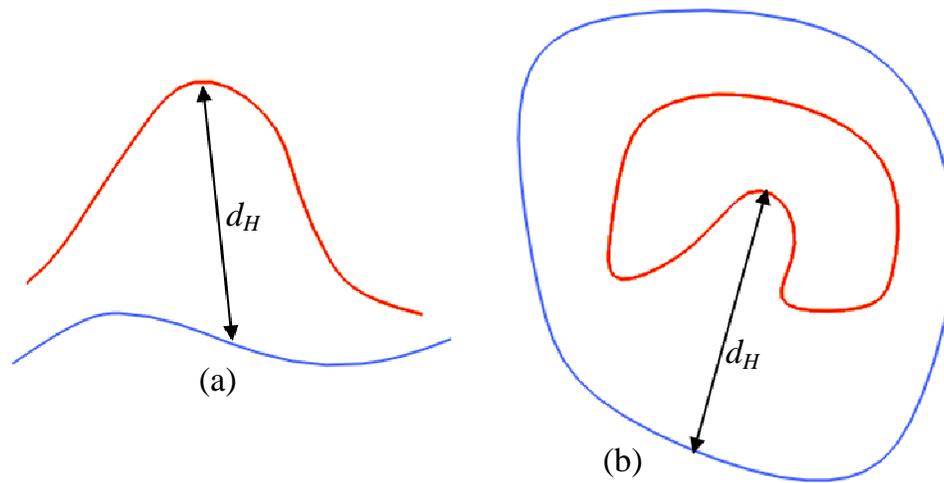


Figure 22 Illustration of the Hausdorff distance d_H in a) pair of open curves, b) pair of closed curves.

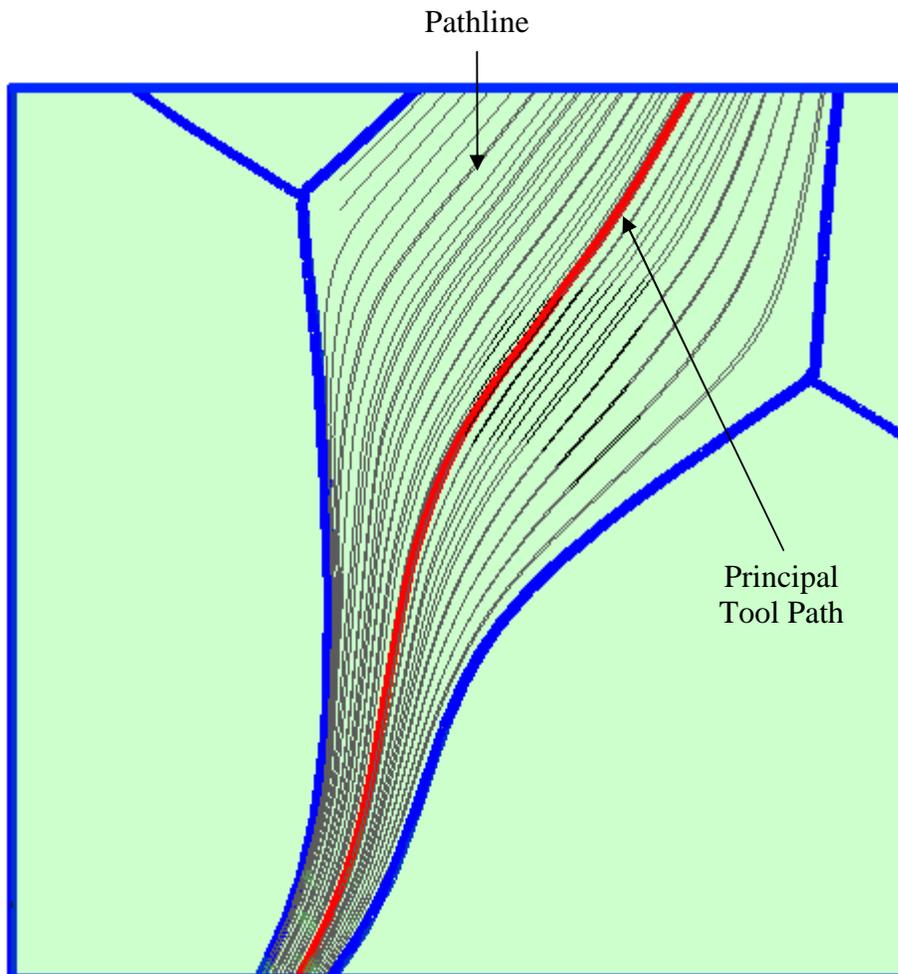
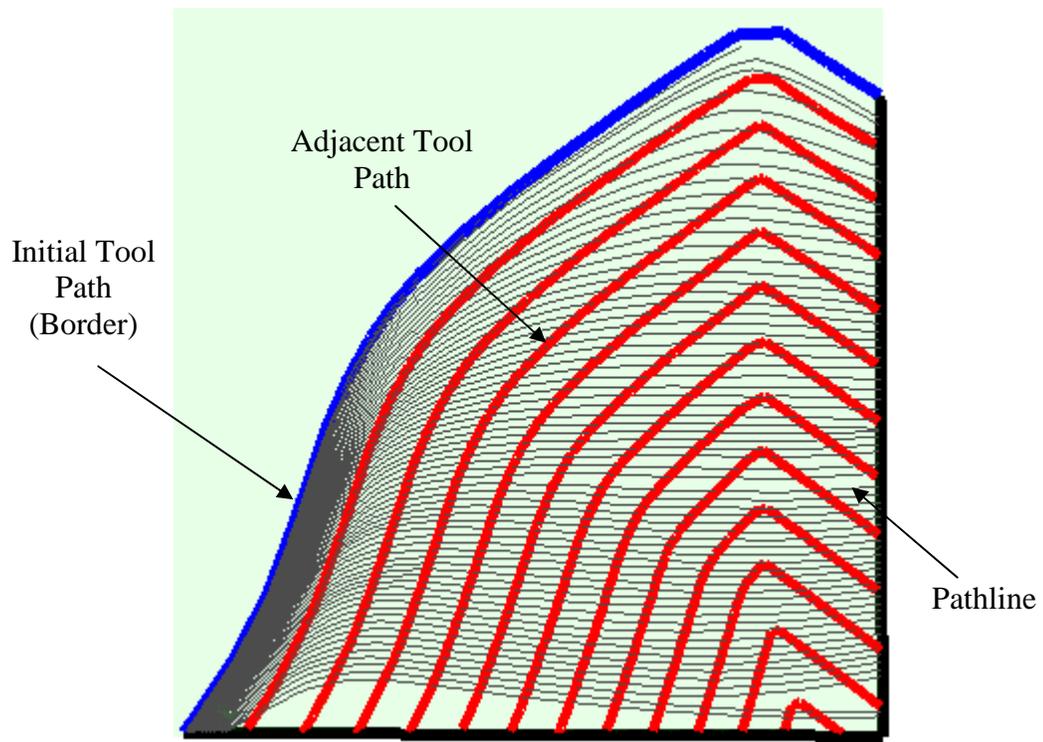
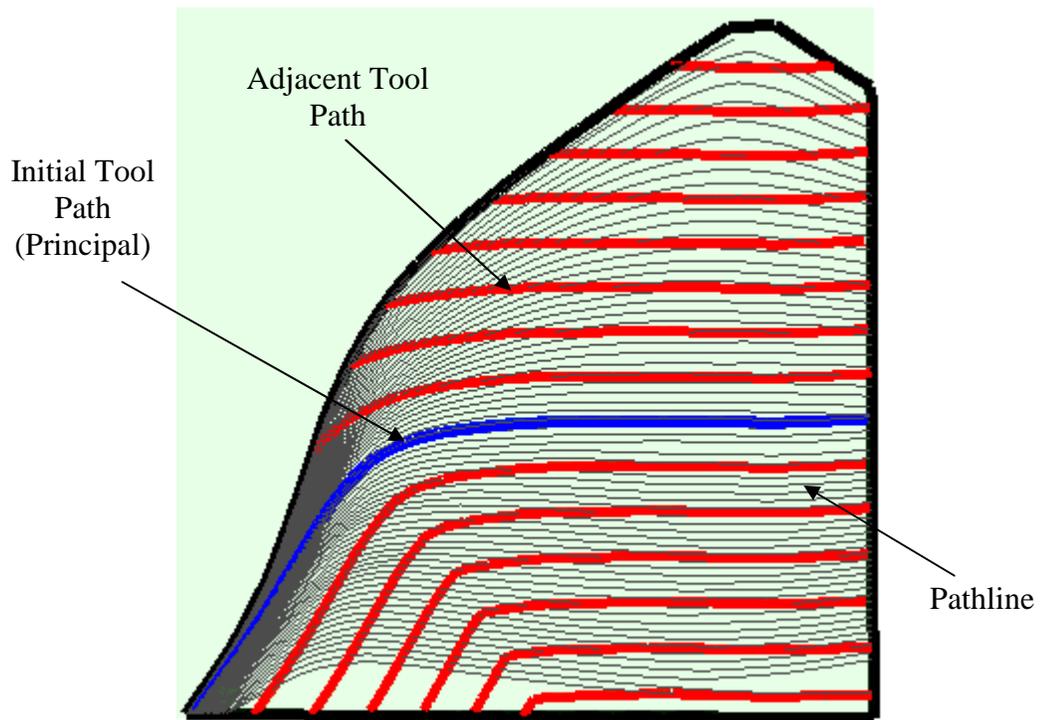


Figure 23 Illustration of the Principal Tool Path.



(a)



(b)

Figure 24 Iso-scallop tool paths compared with pathlines using a) the region border and b) the Principal Tool Path

5. Case Studies and Results Comparison

The proposed method was applied to generate the iso-scallop tool paths on six design surfaces. The ball-end radius r was chosen to be 10 mm and the scallop height h to be 0.2 mm. All six surfaces were generated using third degree NURBS functions using Unigraphics NX 6.0, and then exported into IGES files. The method was implemented using Matlab 2010b.

Case study 1, shown in Figs. 25-28, analyzes a cone-like surface where no degenerate points were identified and therefore, the whole surface was considered as a single region with one principal tool path. This surface does not have visible bifurcations or looping in their pathlines and, similar to a half-cylinder, all pathlines have a similar flow. The proposed method generated a tool path that was 39.6% shorter than the iso-parametric method; 23.3% shorter than the iso-planar method; 9.8% shorter than the conventional iso-scallop method using the u border, and 2.6% shorter than the conventional iso-scallop method using the v border. This last method yielded a tool path that had a similar flow to the overall pathlines in this surface.

Case study 2, shown in Figs. 29-32, analyzes a surface with one trisector degenerate point, segmenting the surface into three regions of interest. Its PFD pathlines show one bifurcation. The proposed method generated a tool path that was 61.1% shorter than the iso-parametric method; 20.4% shorter than the iso-planar method; 6.4% shorter than the conventional iso-scallop method using the u border, and 5.1% shorter than the conventional iso-scallop method using the v border. Both conventional iso-scallop methods have adjacent tool paths that tend to differ from the preferred direction.

Case study 3, shown in Figs. 33-36, analyzes a surface with two wedge degenerate points each having one pseudo-separatrix and both sharing one separatrix. The input surface is segmented

into four regions. The proposed method generated a tool path that was 9.0% shorter than the iso-parametric method; 6.5% shorter than the iso-planar method; 5.1% shorter than the conventional iso-scallop method using the u border, and 1.9% shorter than the conventional iso-scallop method using the v border. Compared to the first two case studies, there are no regions where iso-curves are very dense. It is to be noted that the method using the v border allowed for a larger side step in the vicinity of the bump's left and right sides.

Case study 4, shown in Figs. 37-40, analyzes a surface with one merged wedge degenerate point. Similar to case study 1, the whole surface was considered as a single region and one principal path was computed. The proposed method generated a tool path that was 16.9% shorter than the iso-parametric method; 16.5% shorter than the iso-planar method; 5.4% shorter than the conventional iso-scallop methods using either the u and v border. This case provided an example where loops were used instead of vertical or horizontal tool paths obtained from the conventional iso-scallop methods.

Case study 5, shown in Fig. 41-44, analyzes a surface with all types of degenerate point (three trisectors, two wedges, and one merged wedge). Eight regions were identified and each had its own principal tool path from where the adjacent iso-scallop tool paths were generated. The proposed method generated a tool path that was 37.6% shorter than the iso-parametric method; 37.2% shorter than the iso-planar method; 7.2% shorter than the conventional iso-scallop method using the u border, and 8.2% shorter than the conventional iso-scallop method using the v border. Using the iso-scallop method on either border yield adjacent tool paths that will tend to differ from the preferred directions, the reason being that the surface has a greater number of valleys and mounts and each have their own preferred flow.

The last case study, shown in Figs. 45-48, is a practical case where the surface consisted on the mold of a bicycle seat. One trisector degenerate point was identified along with two wedge points, dividing the surface into five regions. The proposed method generated a tool path that was 48.9% shorter than the iso-parametric method; 42.0% shorter than the iso-planar method; 12.7% shorter than the conventional iso-scallop method using the u border, and 3.3% shorter than the conventional iso-scallop method using the v border.

The results are summarized in Table 1. In every case, the proposed method generated a tool path with a shorter length than the other conventional methods. The largest reduction compared to the conventional iso-scallop method was observed in the practical method, where the proposed method generated a tool path 12.7% shorter than the iso-scallop method using the u border. The case that observed the largest reduction compared to both the iso-scallop u and v border was the fifth case study, with a tool path with a length 7.2% and 8.2% shorter, respectively.

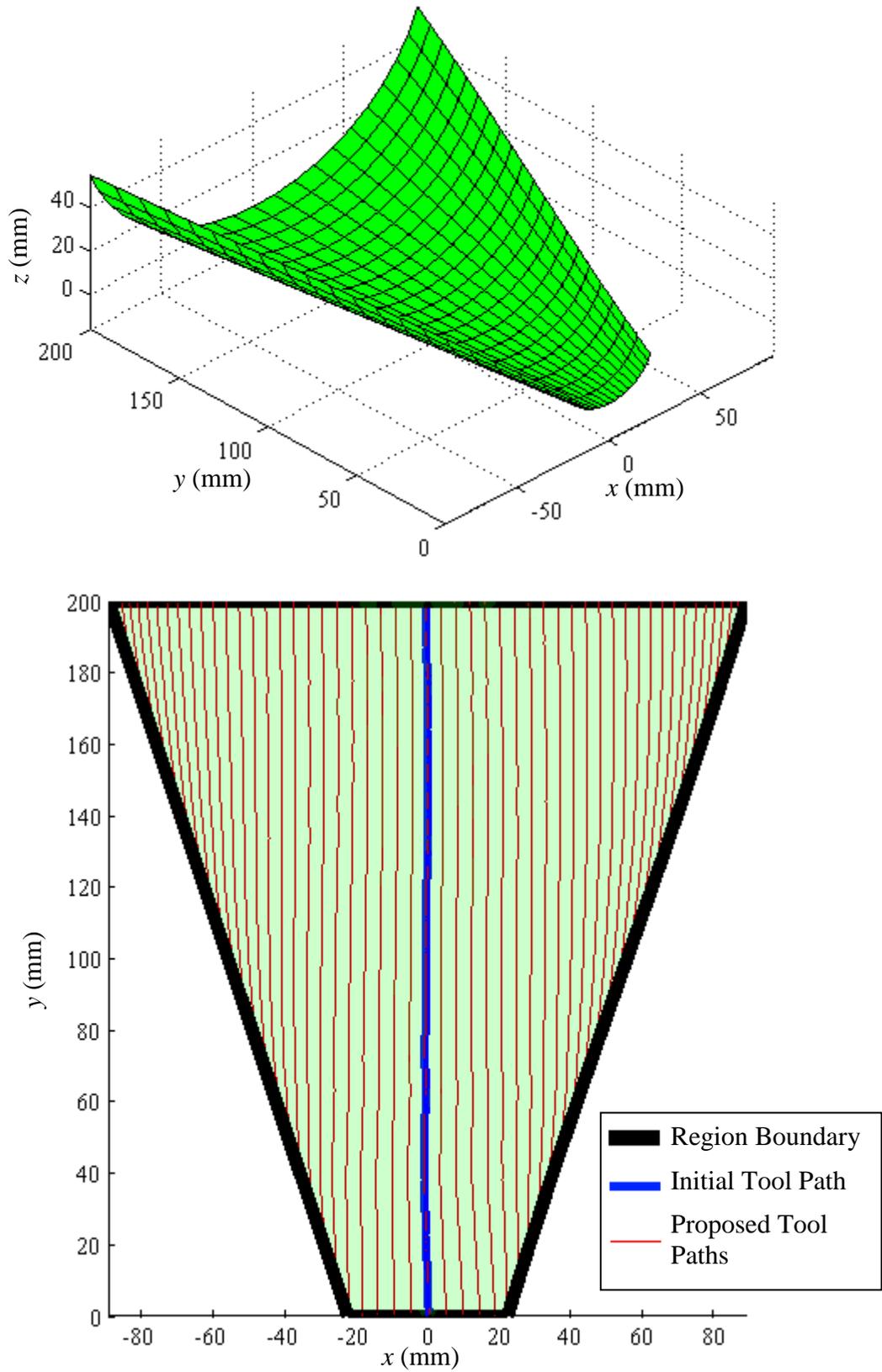


Figure 25 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

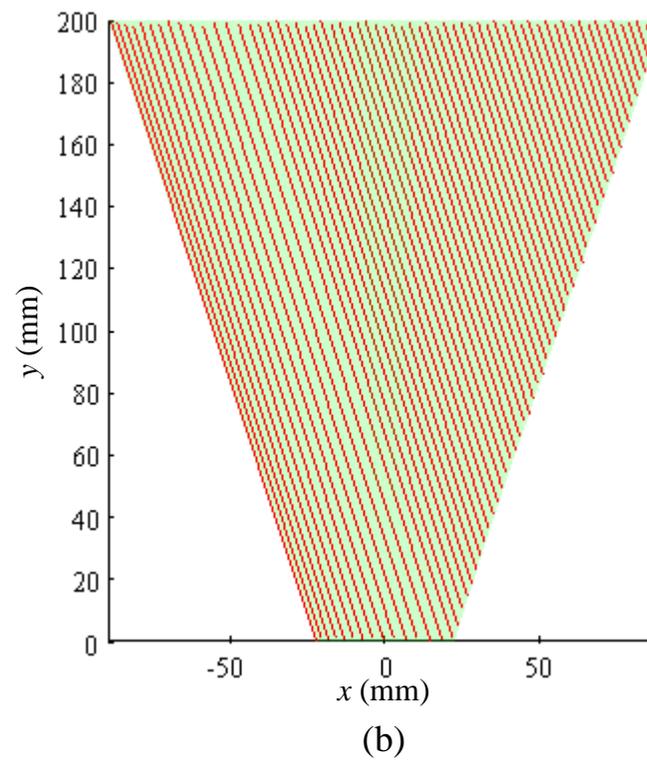
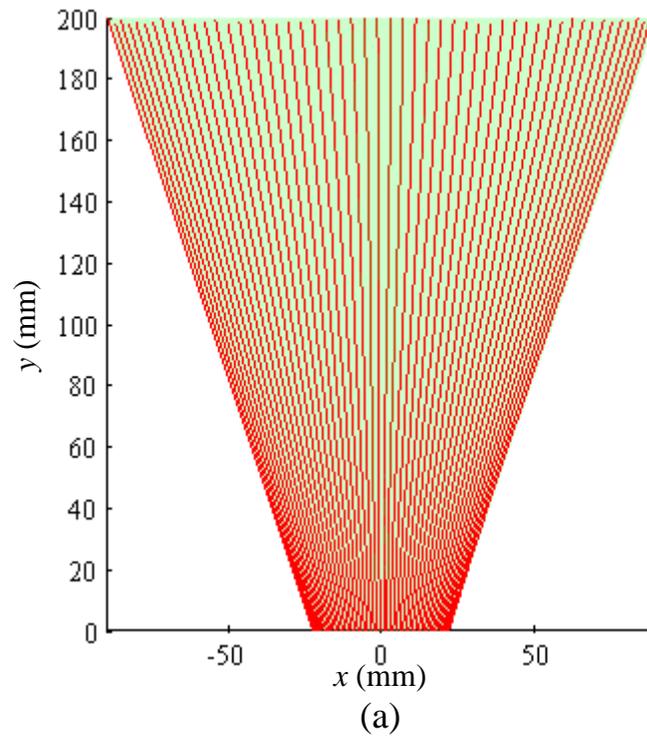


Figure 26 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.

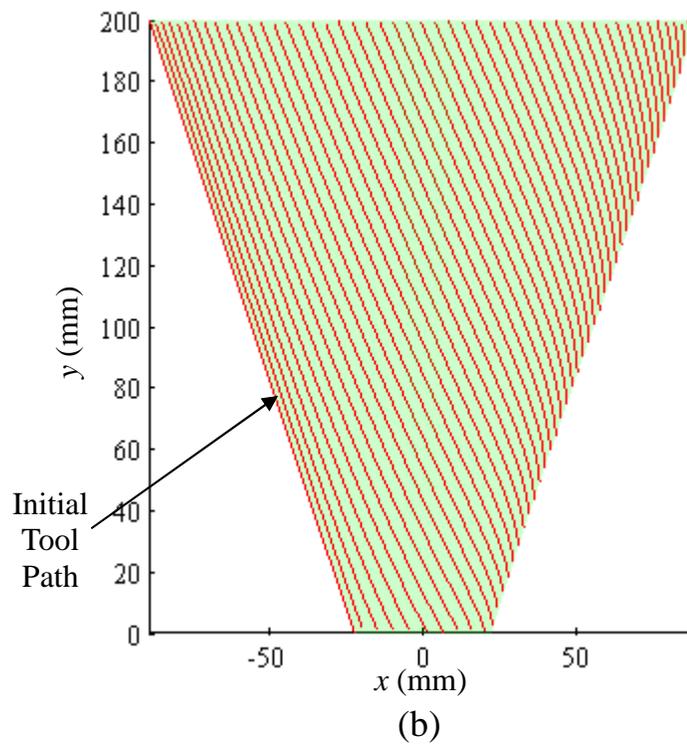
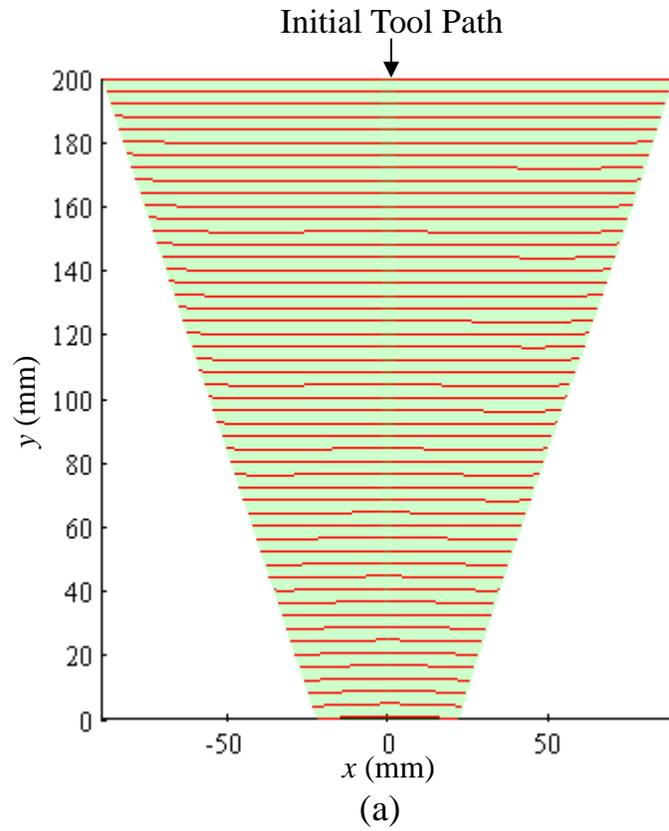


Figure 27 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

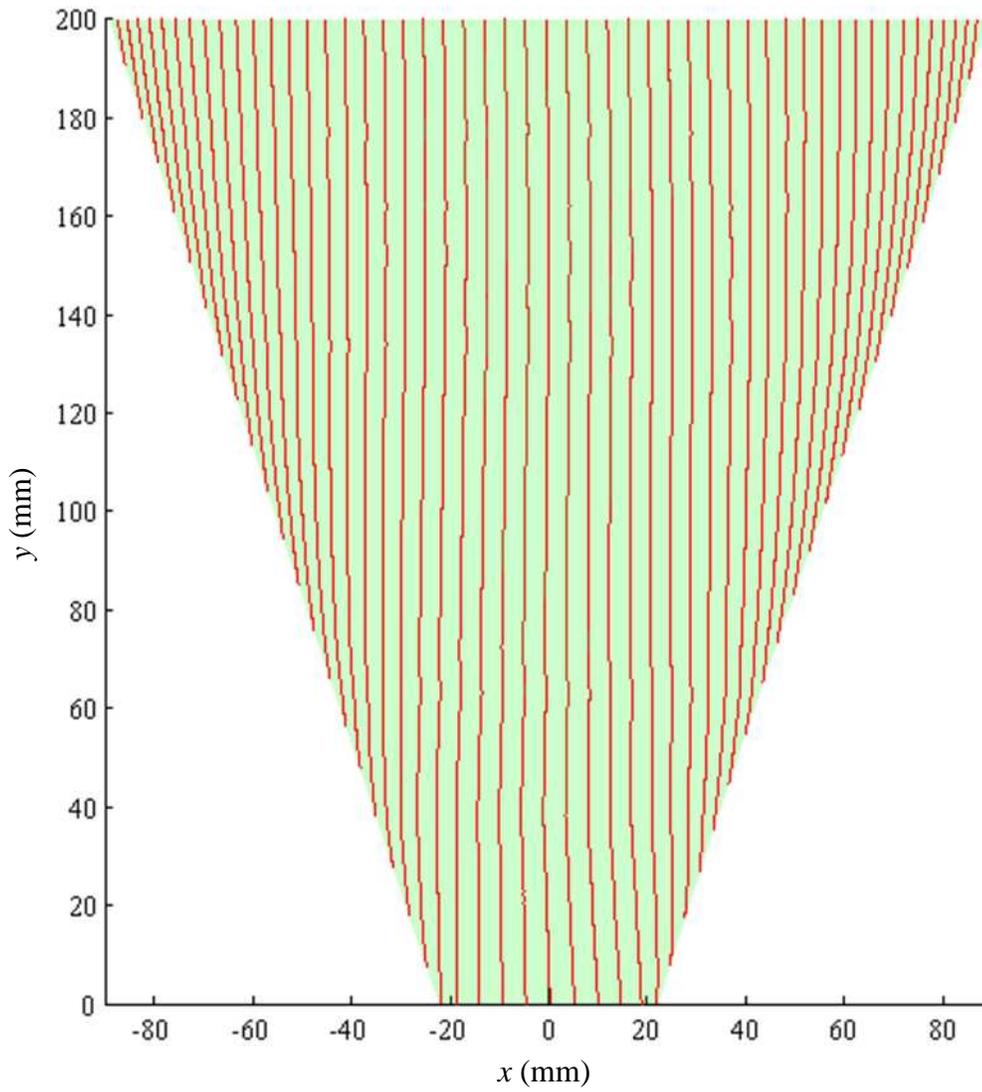


Figure 28 Tool paths of Case Study 1, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

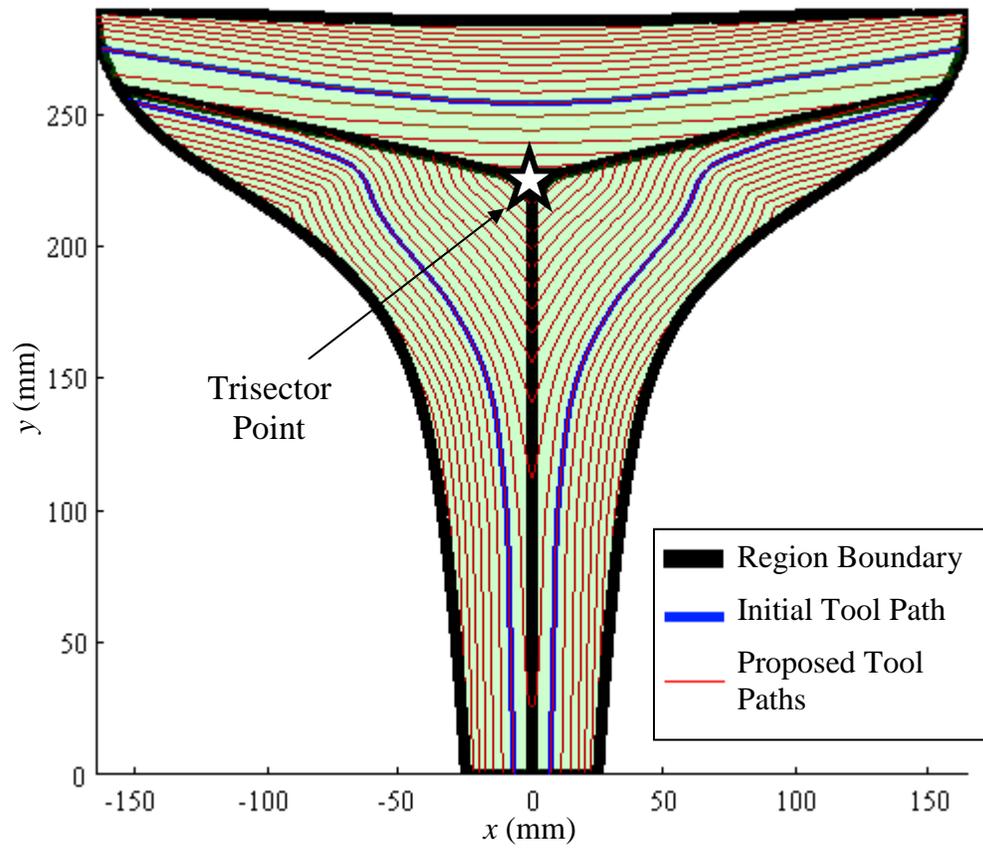
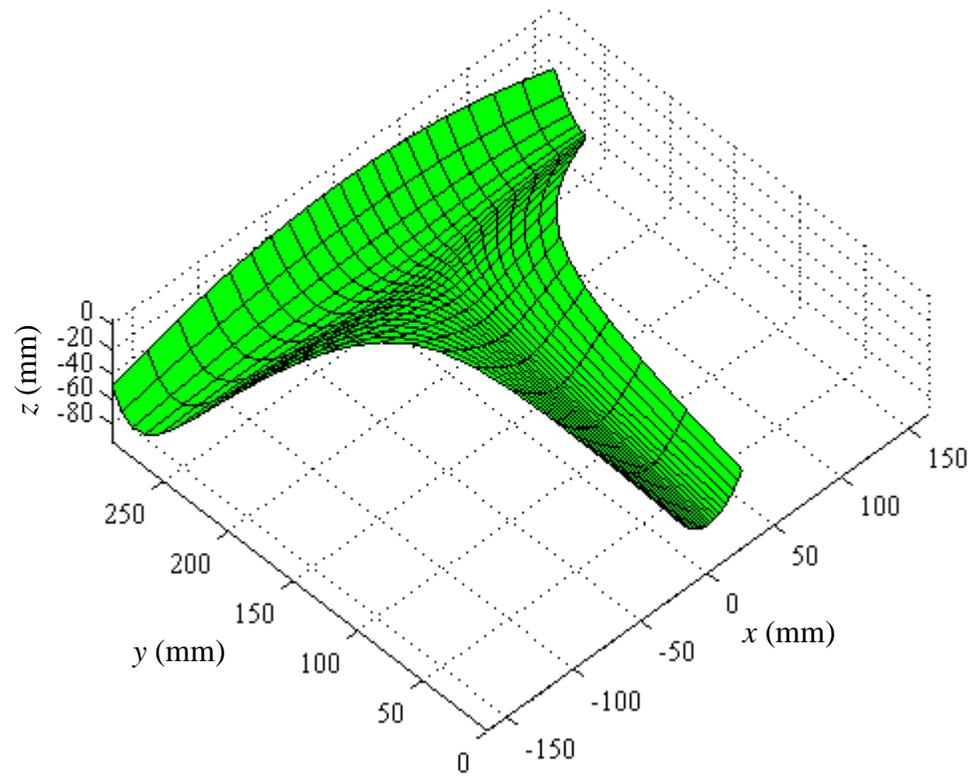


Figure 29 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

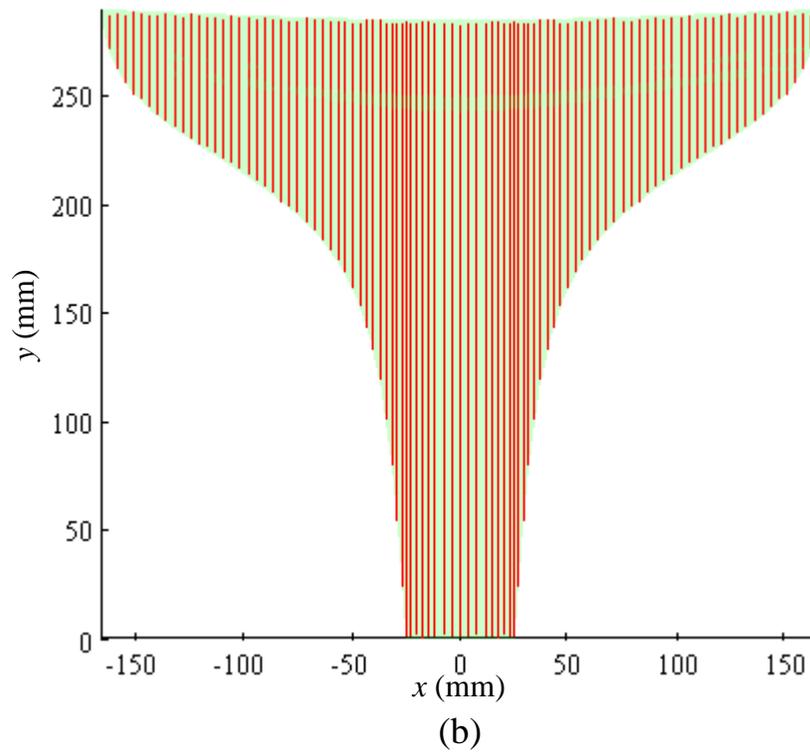
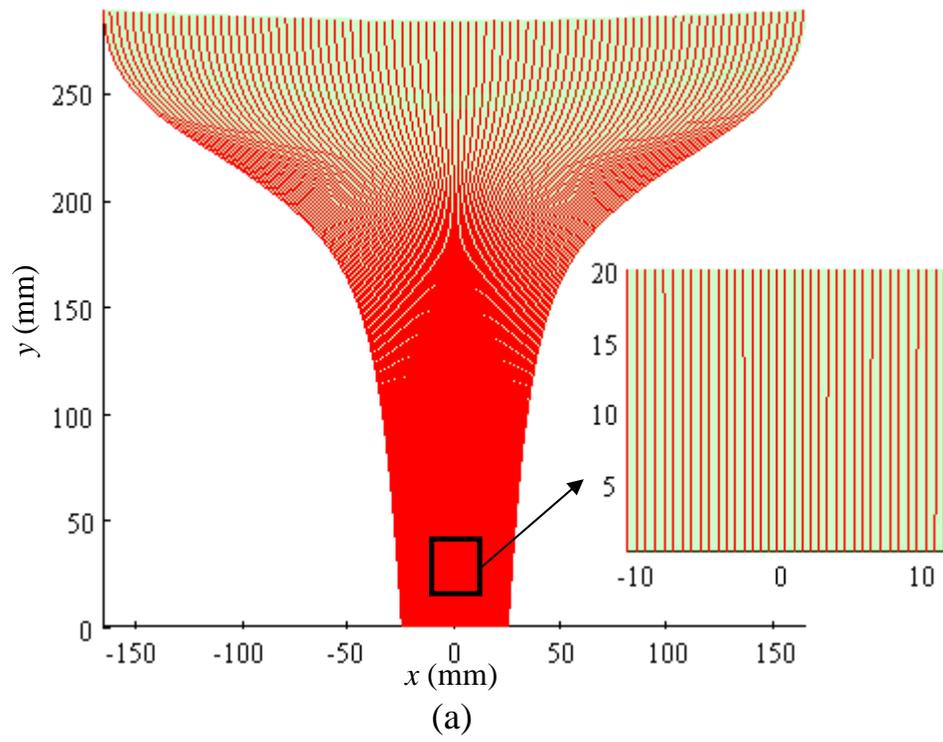


Figure 30 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.

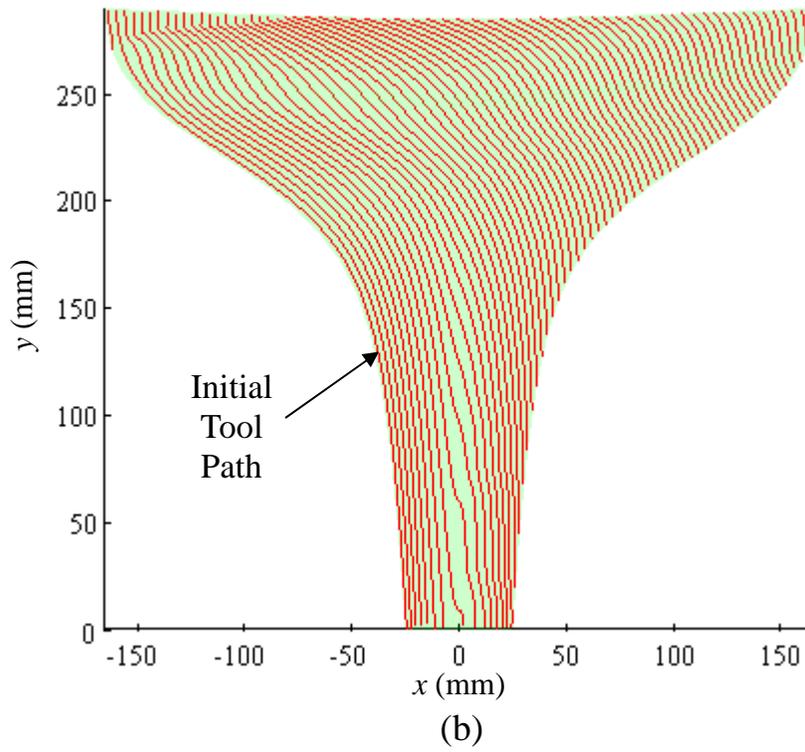
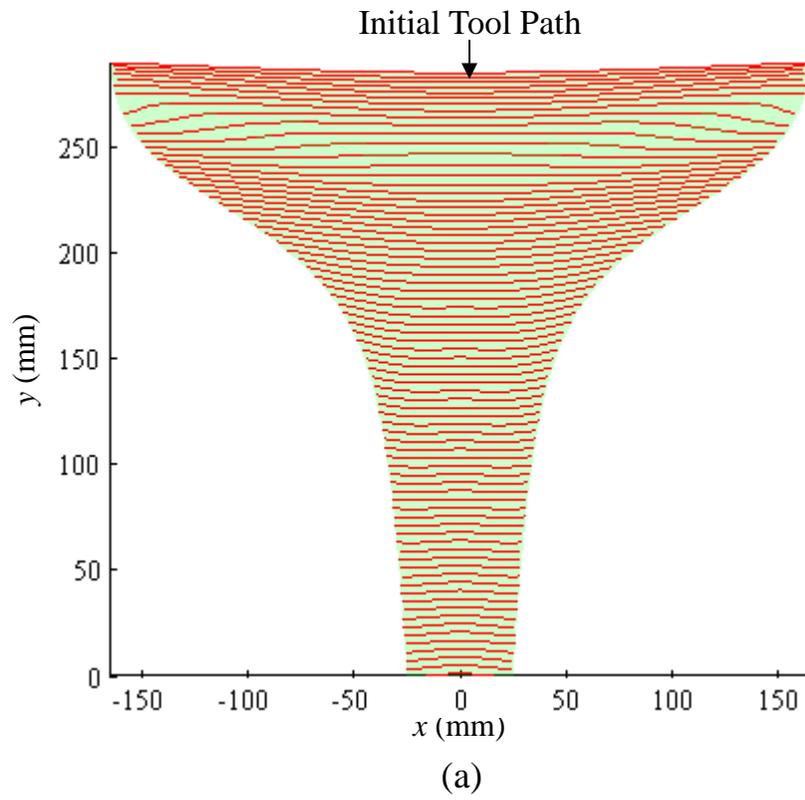


Figure 31 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

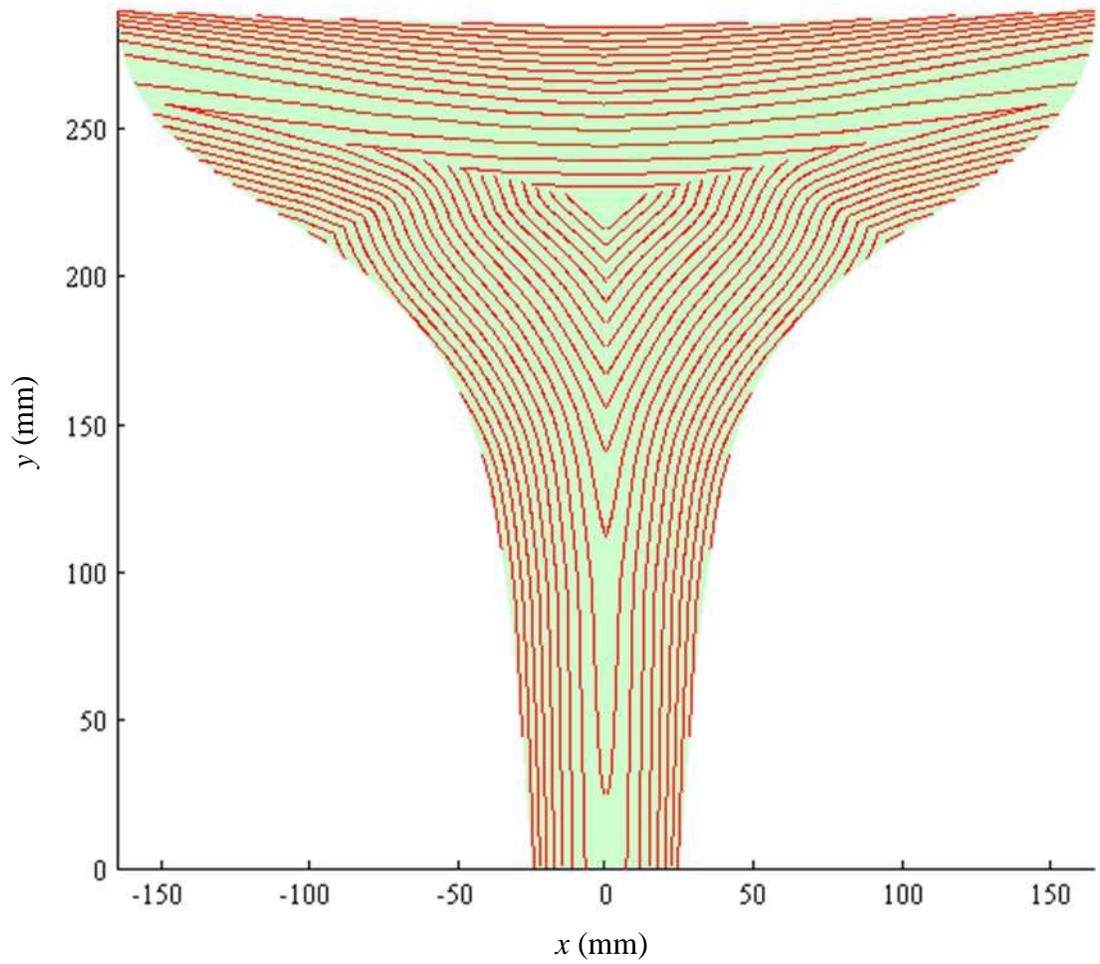


Figure 32 Tool paths of Case Study 2, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

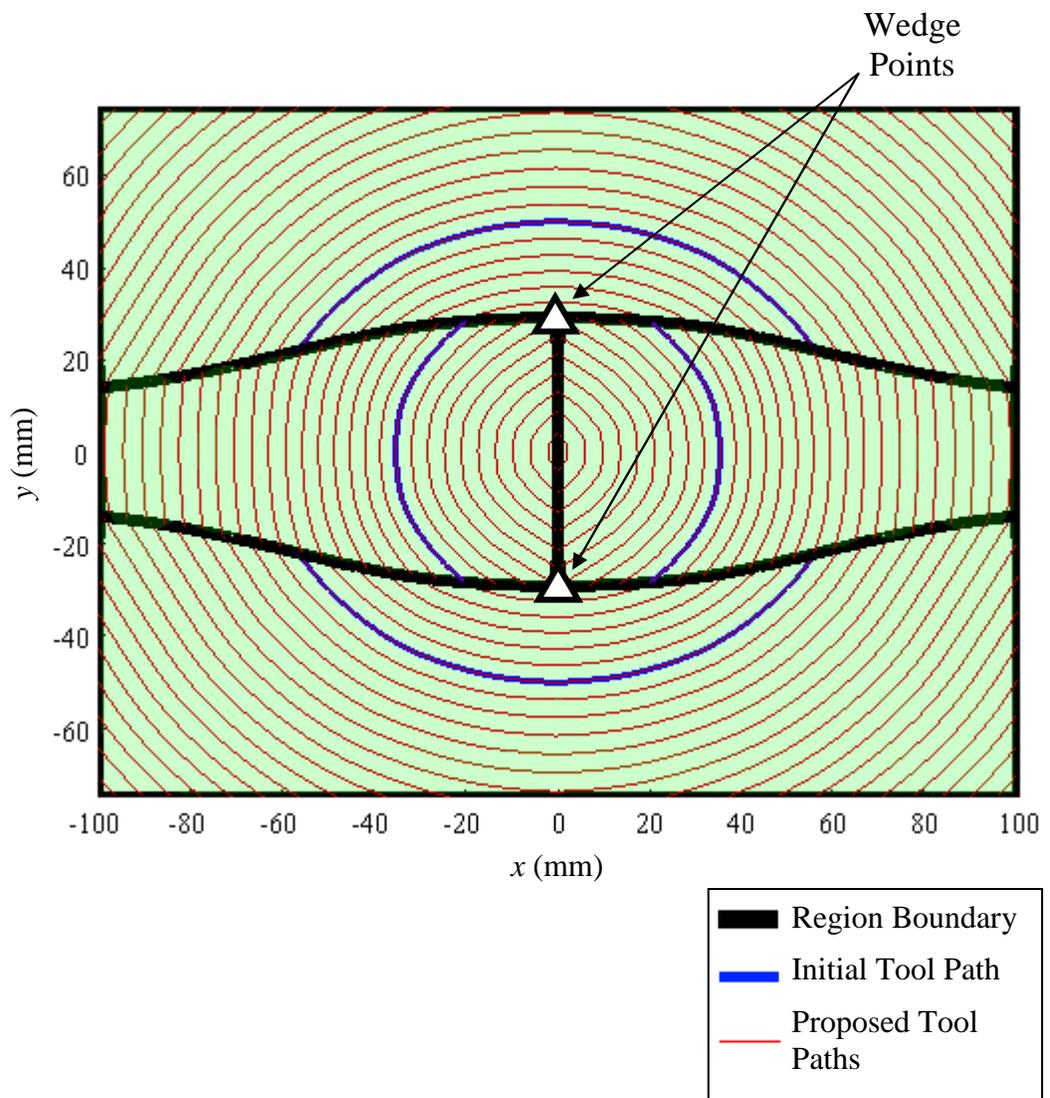
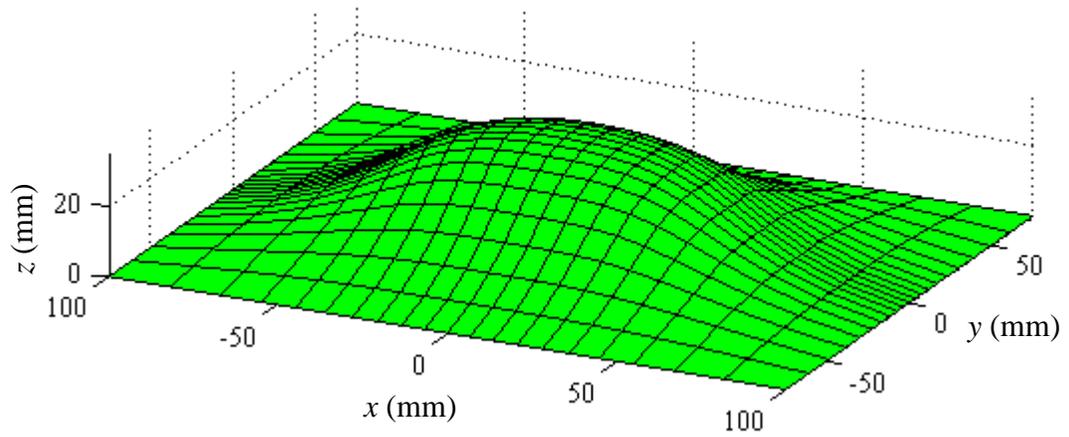


Figure 33 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

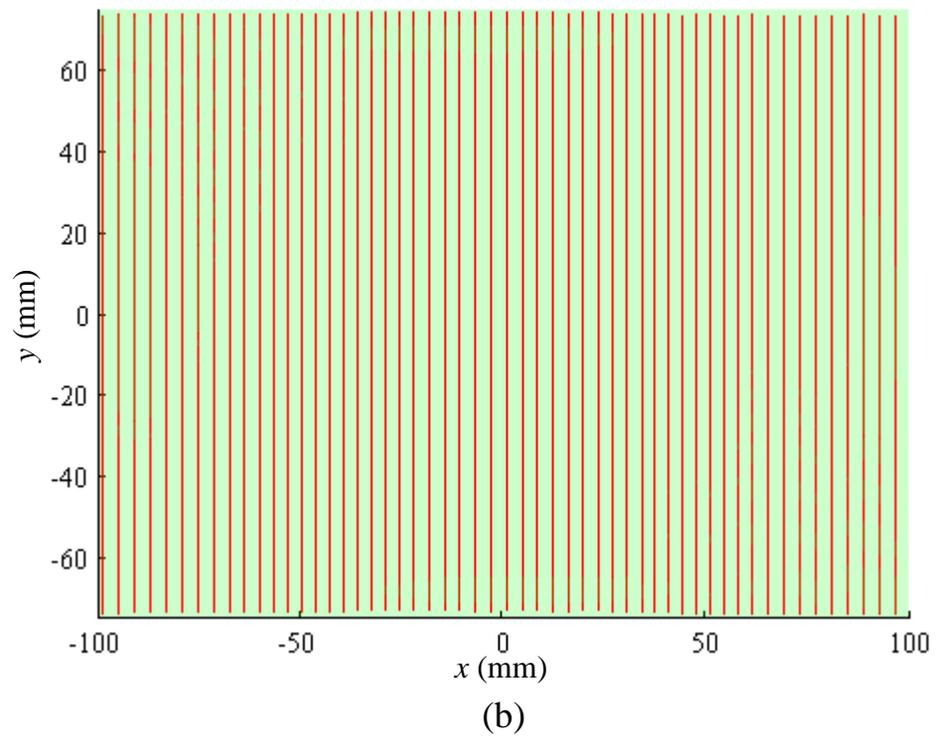
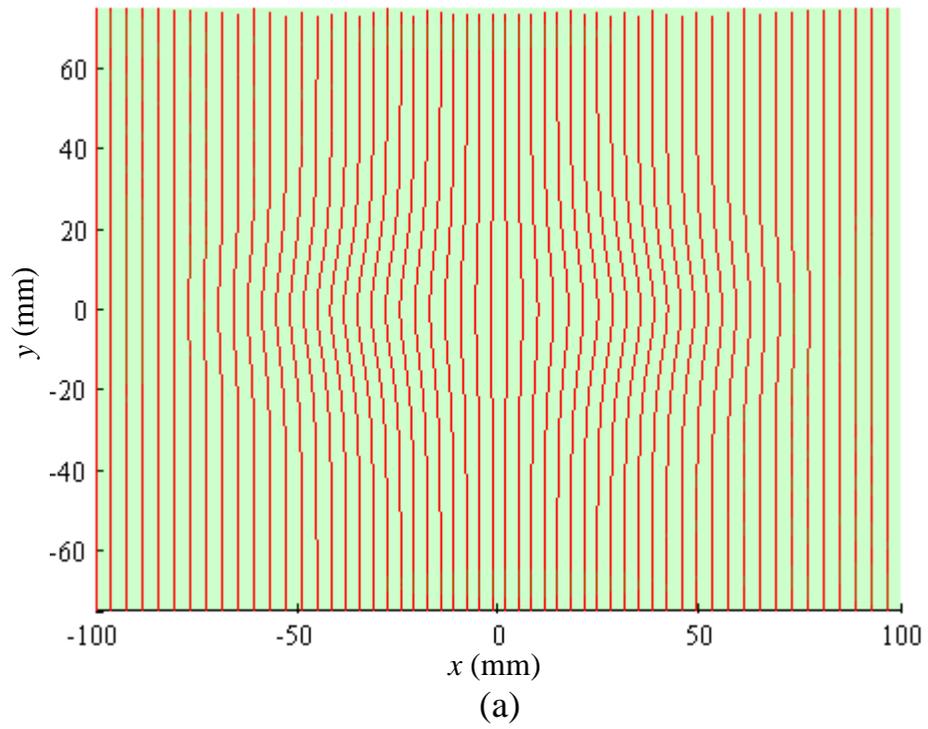


Figure 34 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.

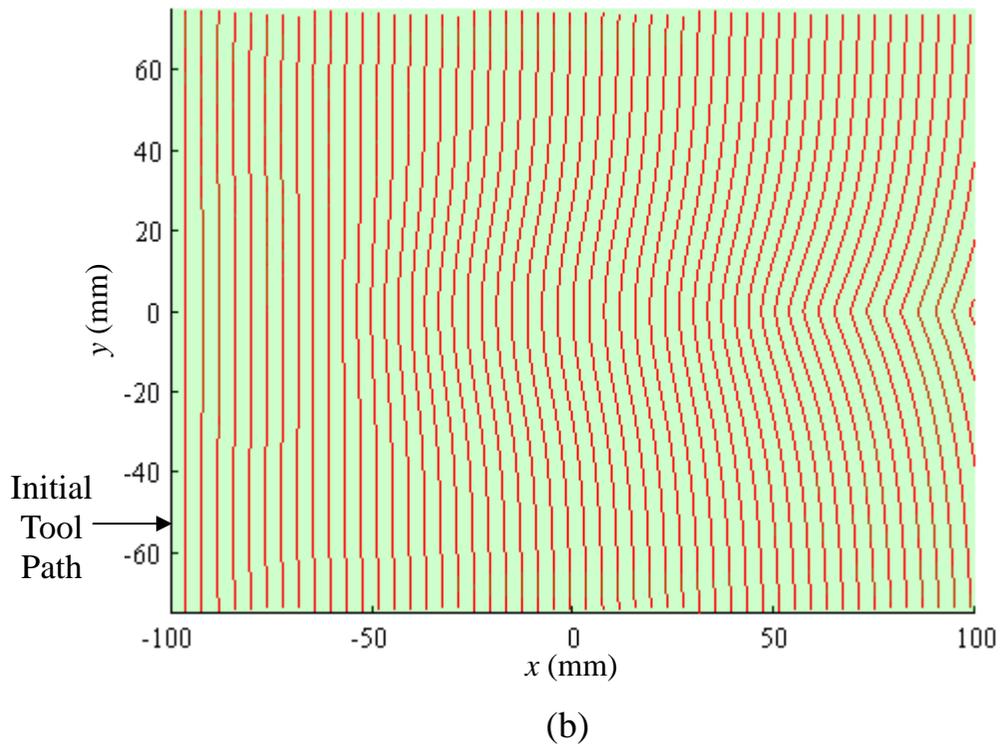
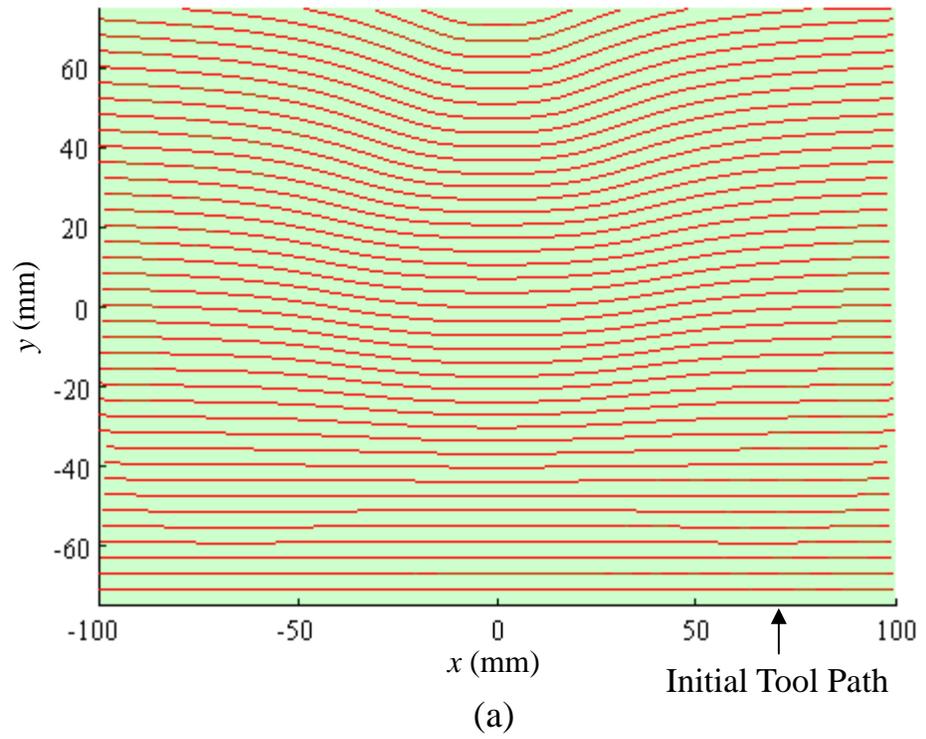


Figure 35 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

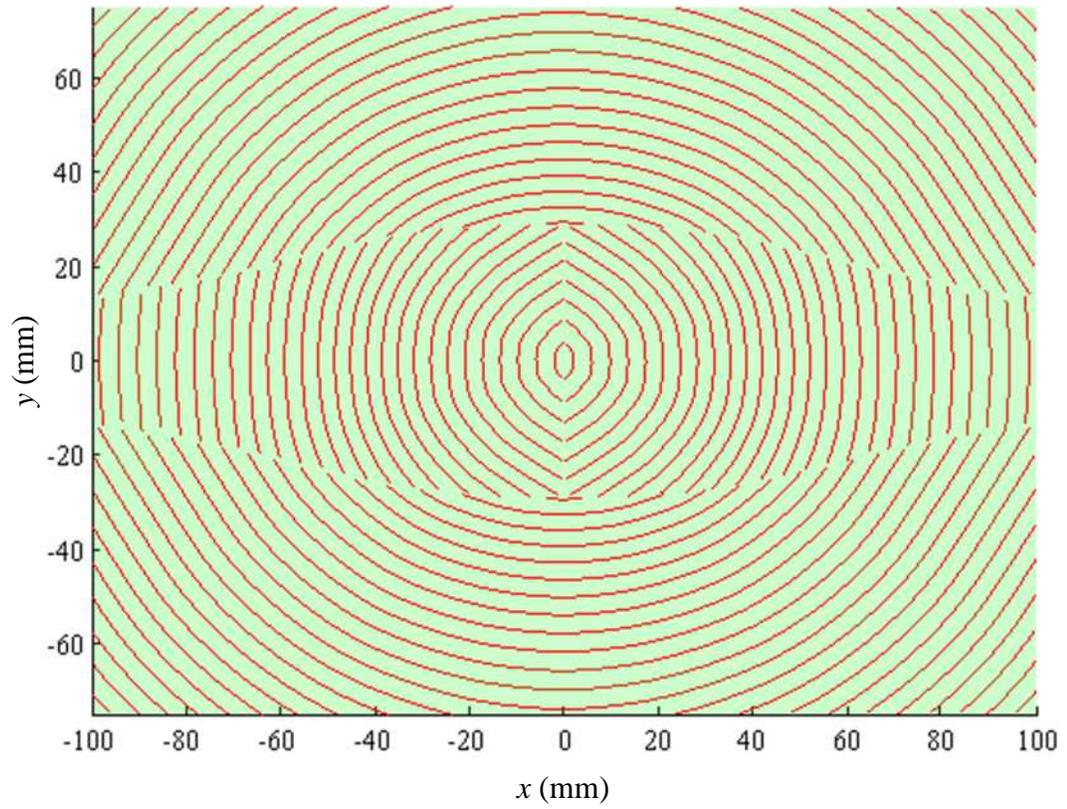


Figure 36 Tool paths of Case Study 3, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

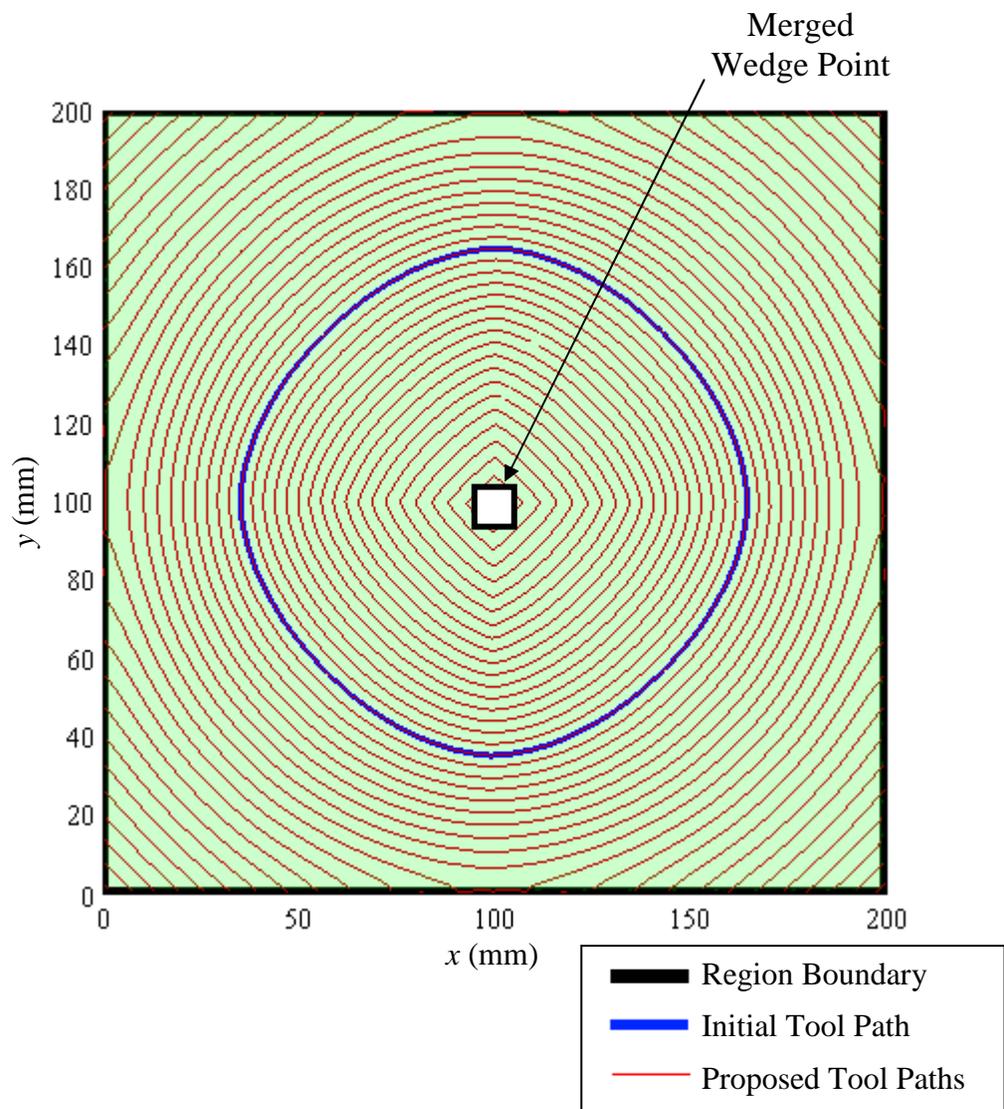
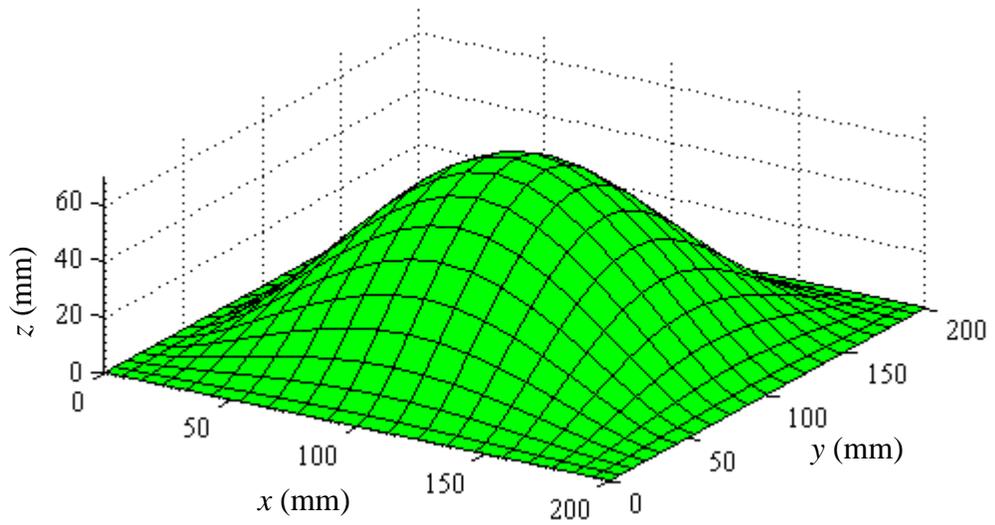


Figure 37 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

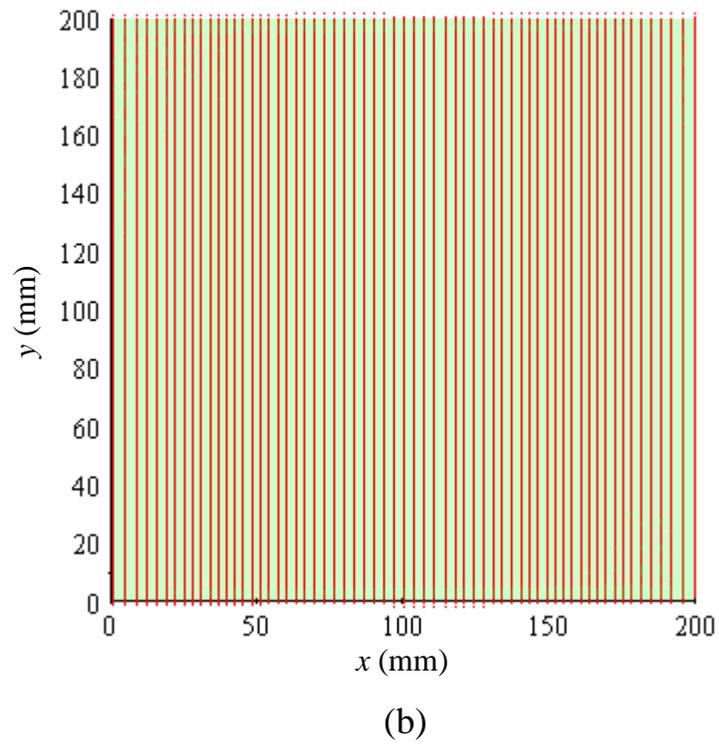
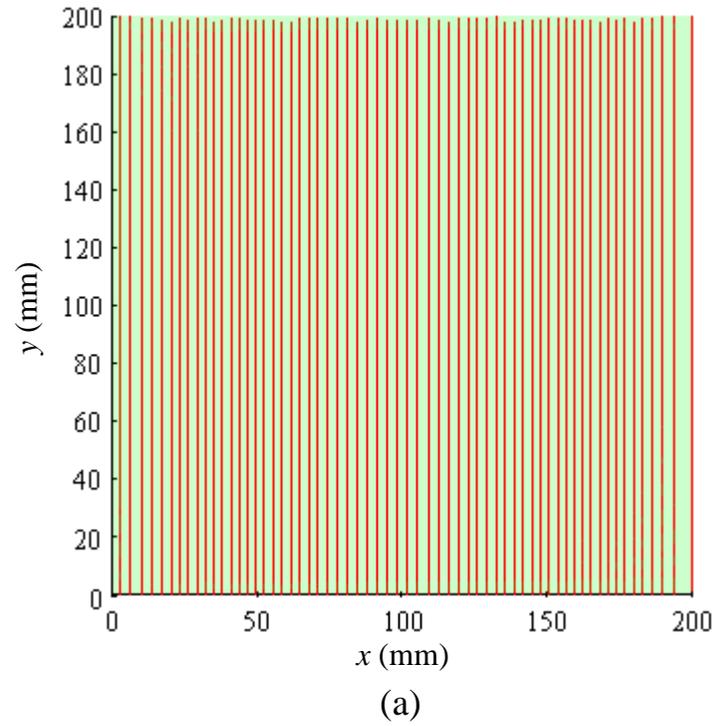


Figure 38 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.

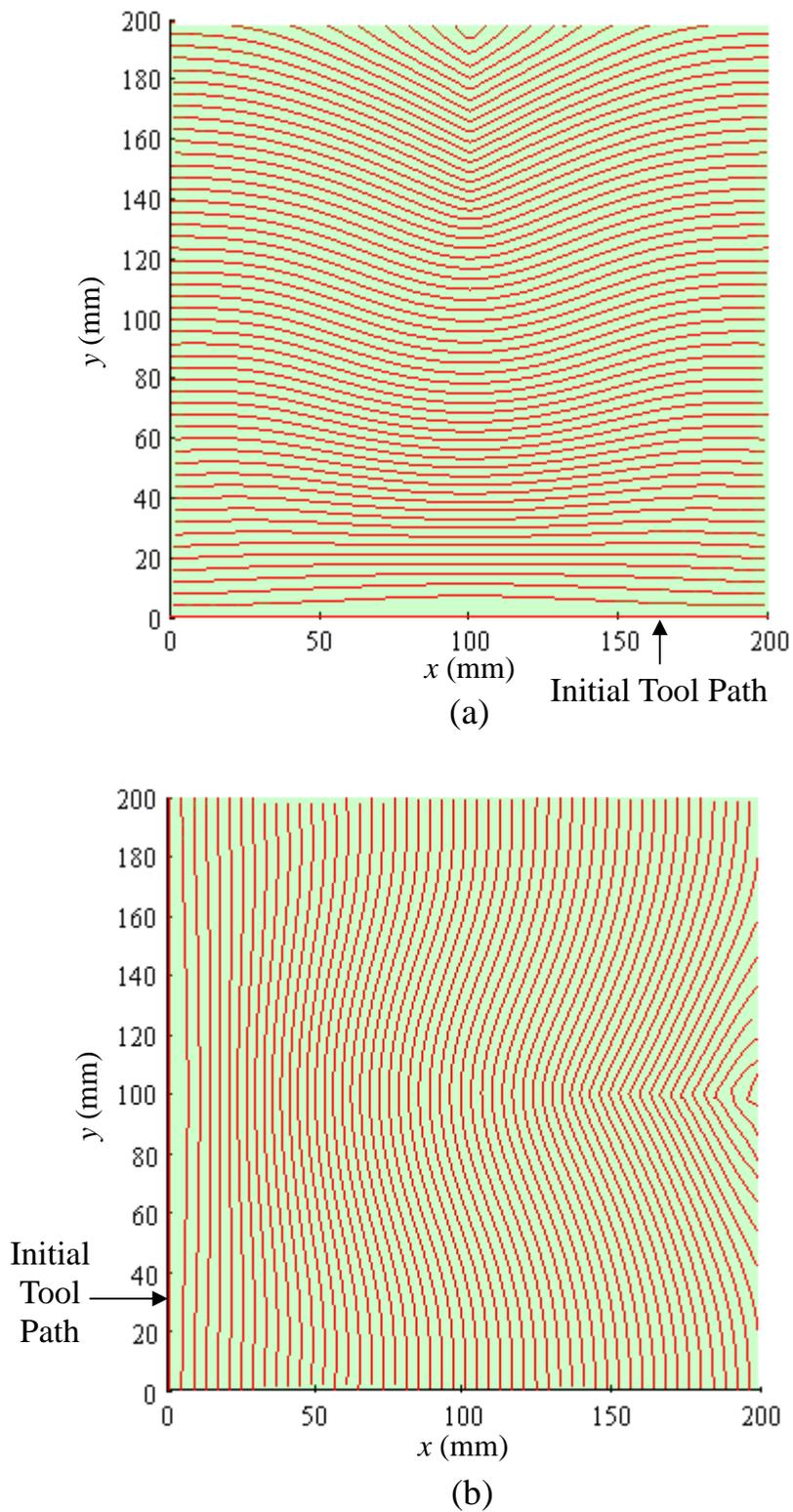


Figure 39 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

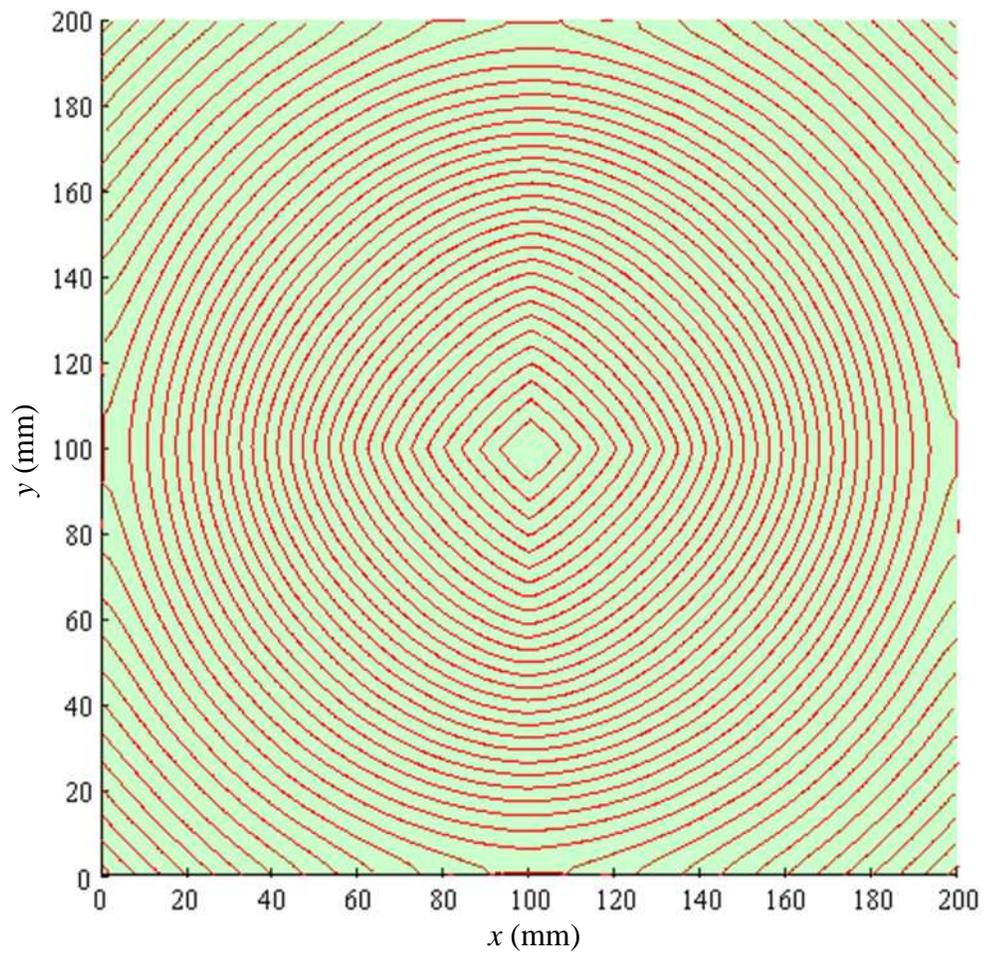


Figure 40 Tool paths of Case Study 4, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

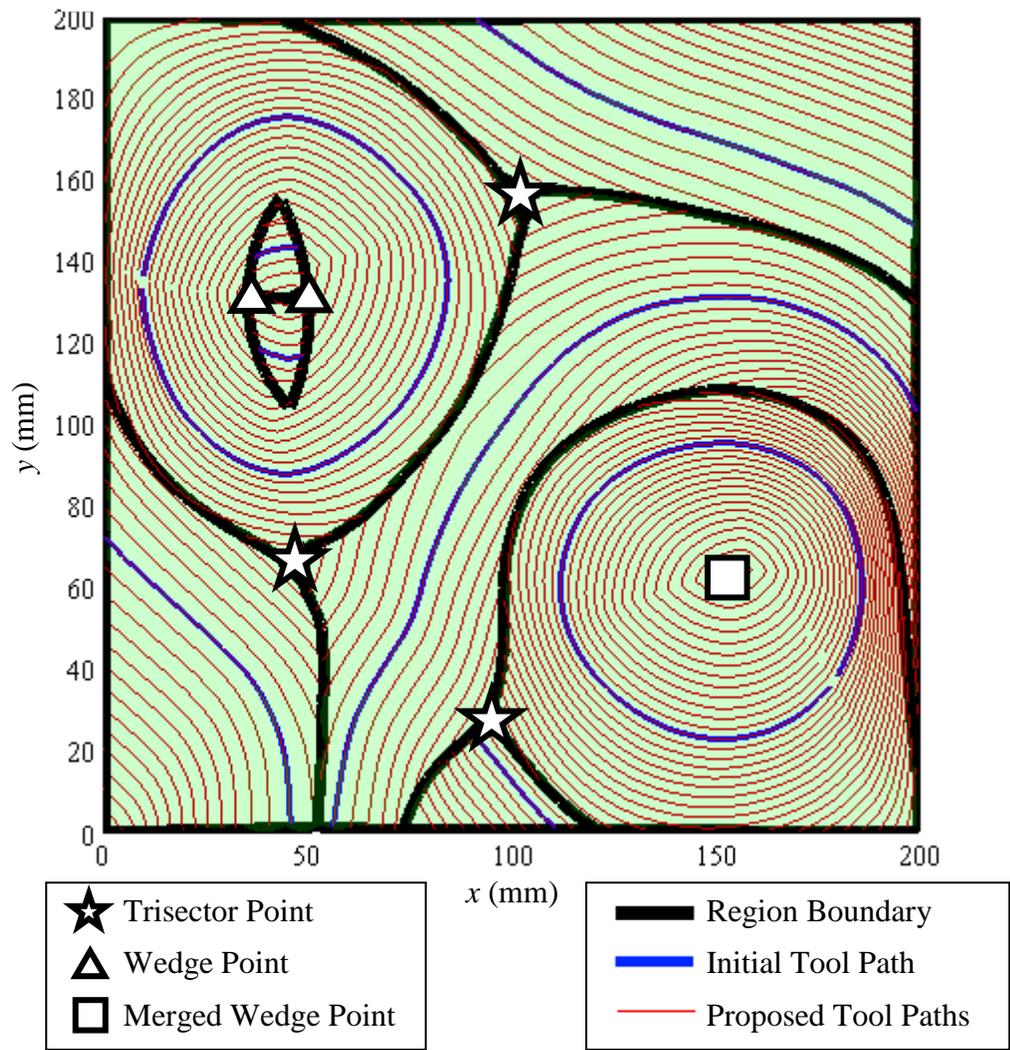
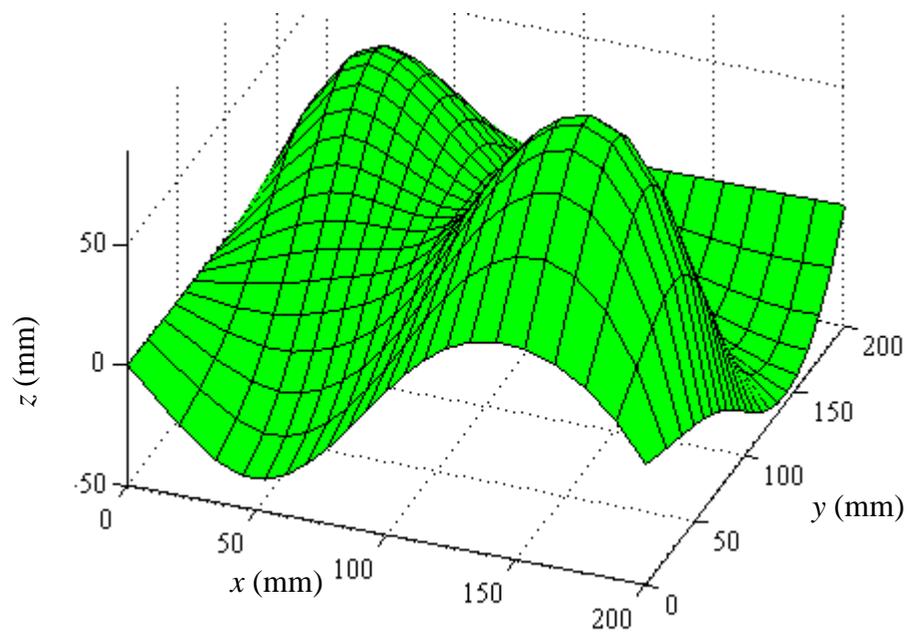


Figure 41 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

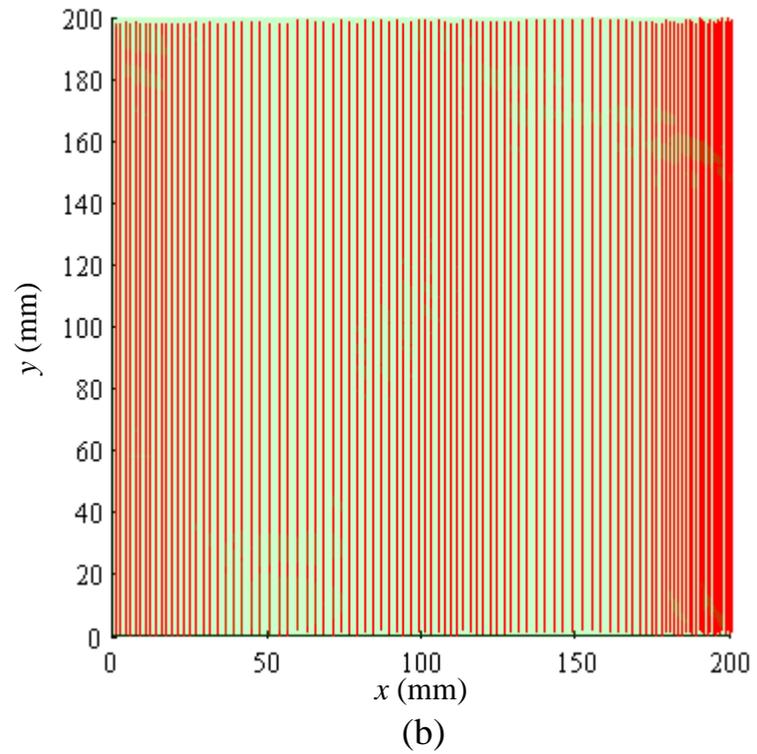
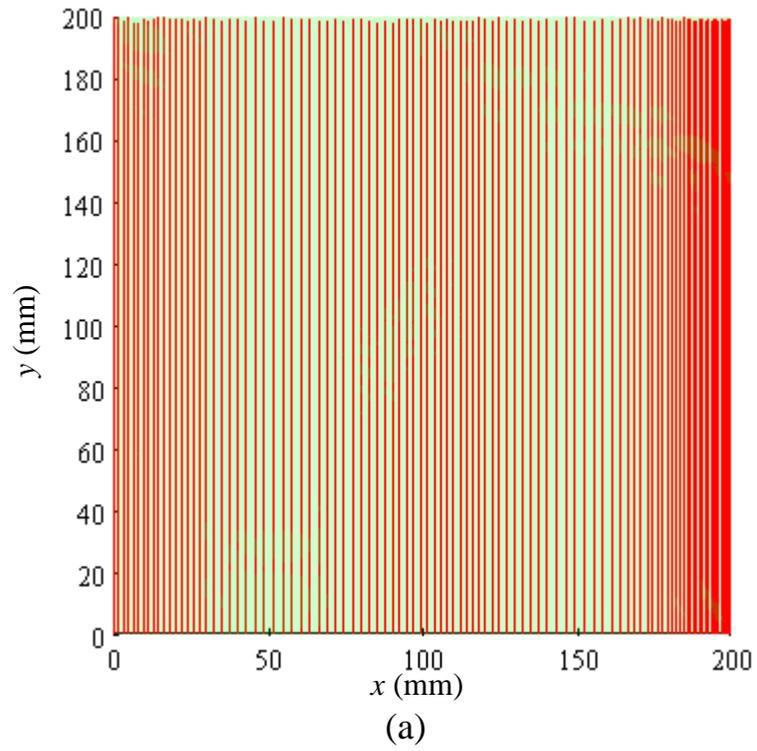


Figure 42 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.

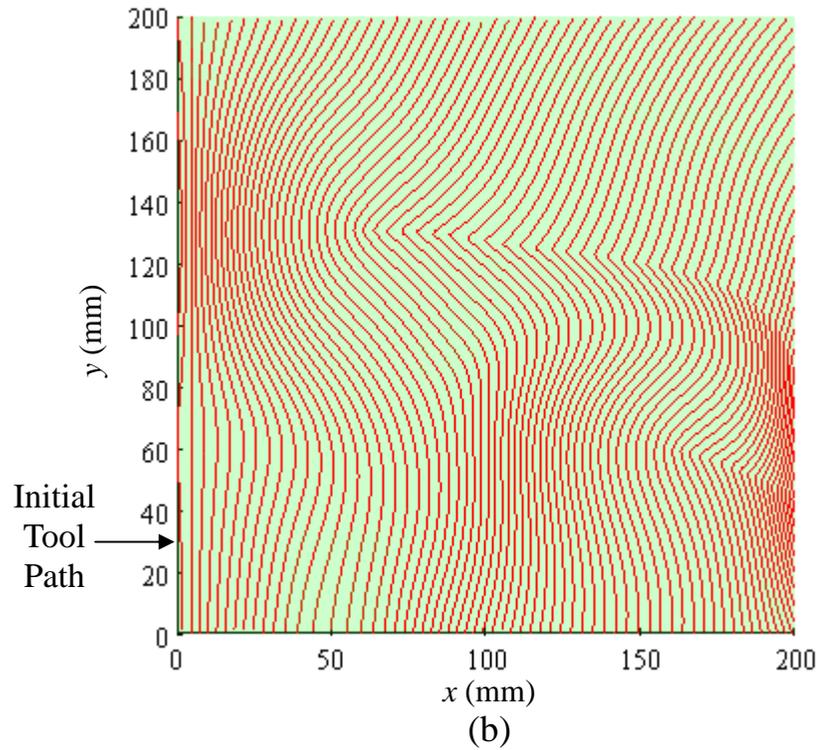
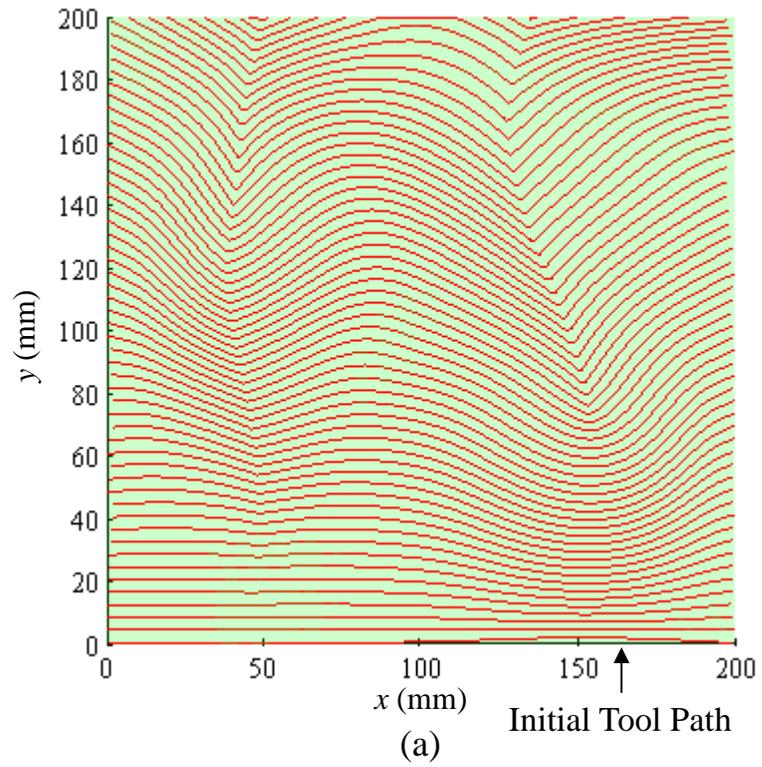


Figure 43 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

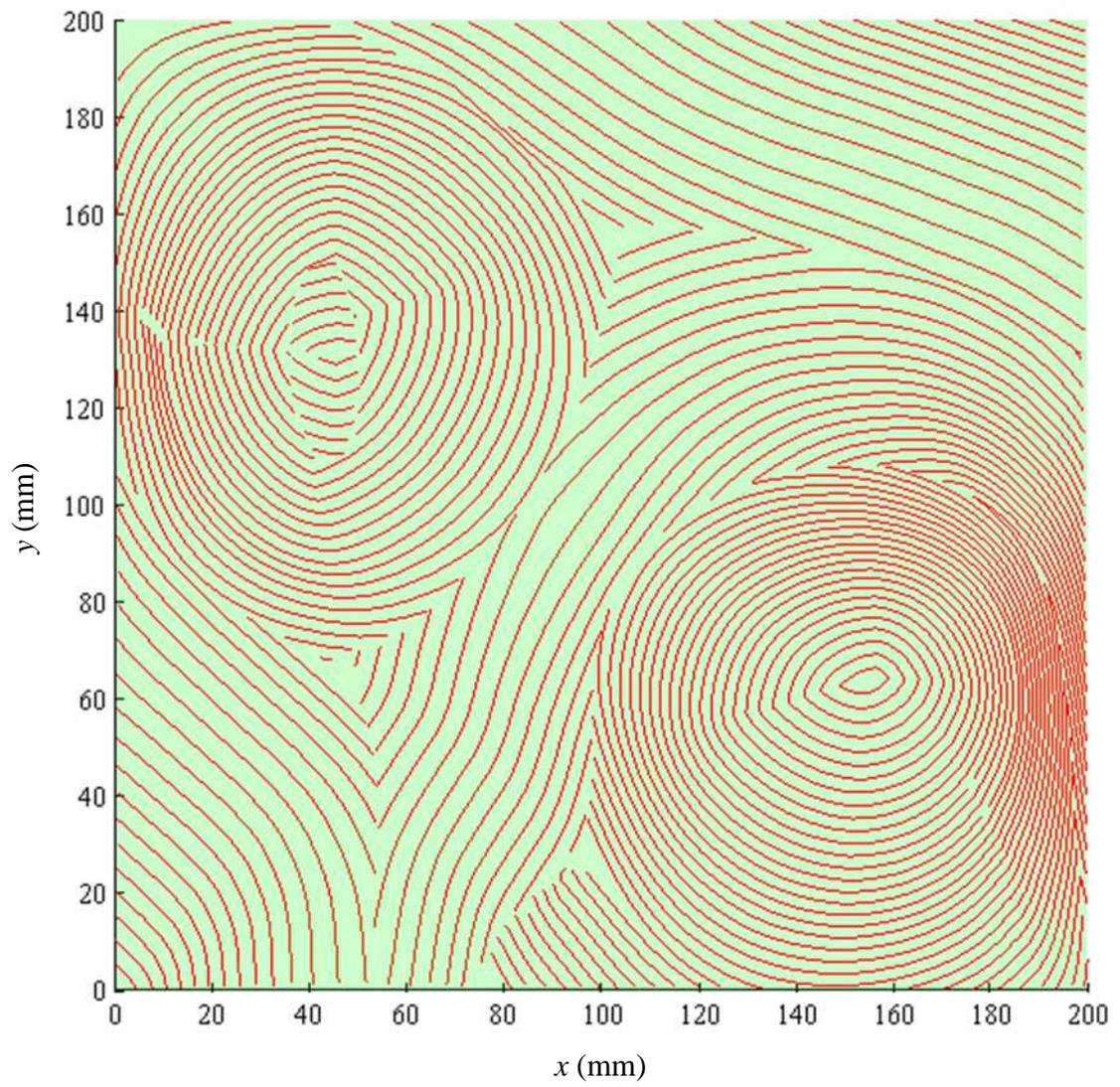


Figure 44 Tool paths of Case Study 5, $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

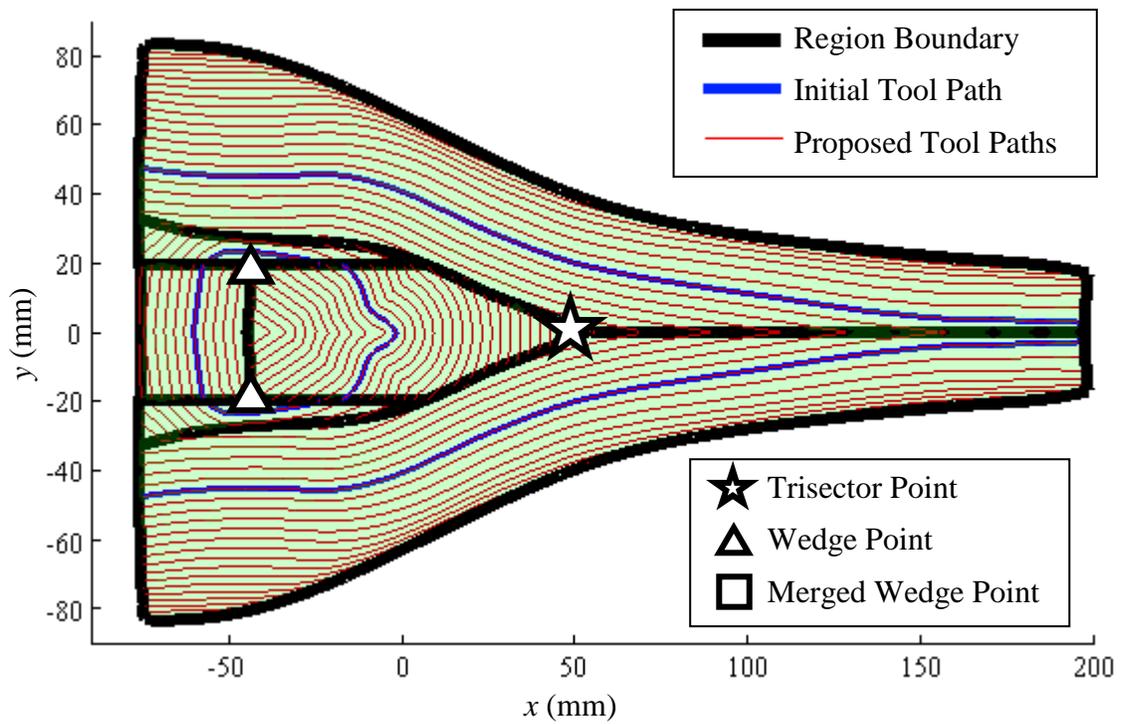
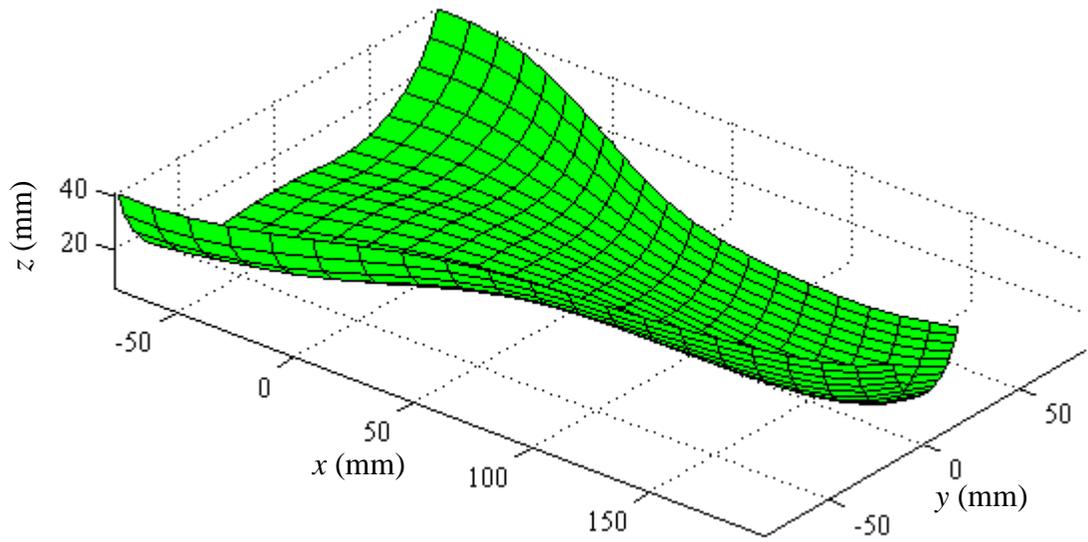
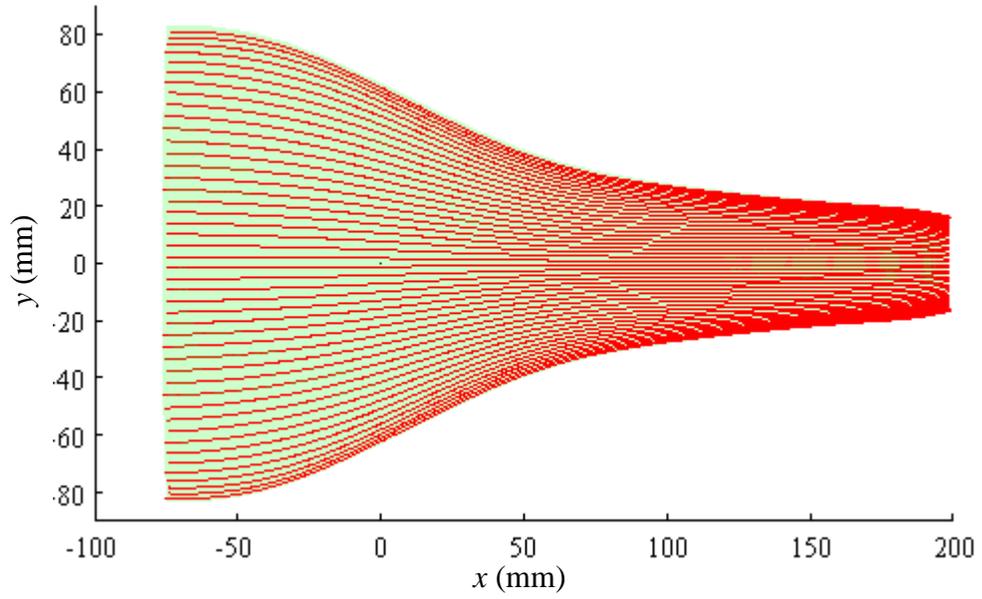
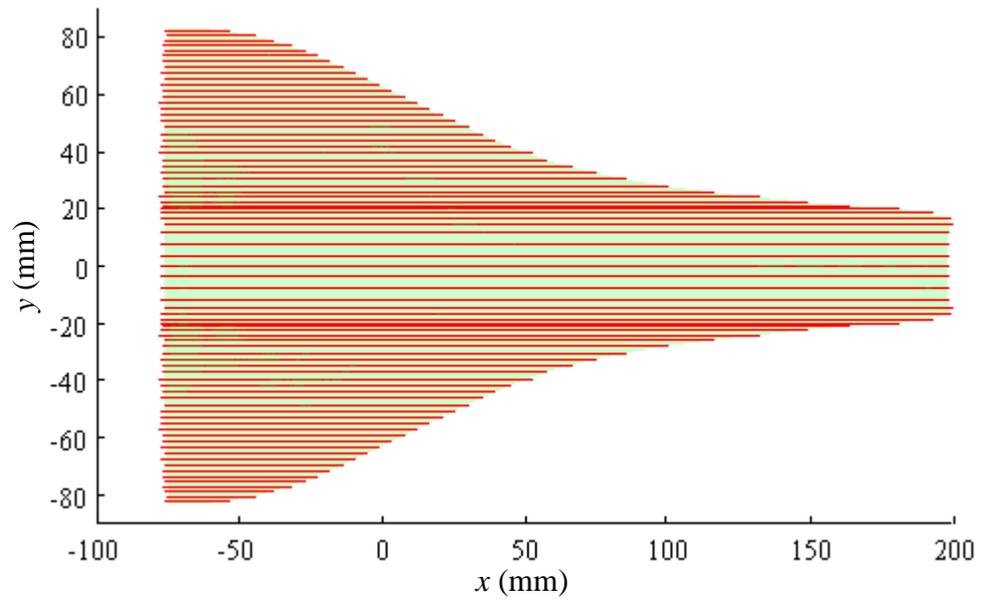


Figure 45 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), proposed method.

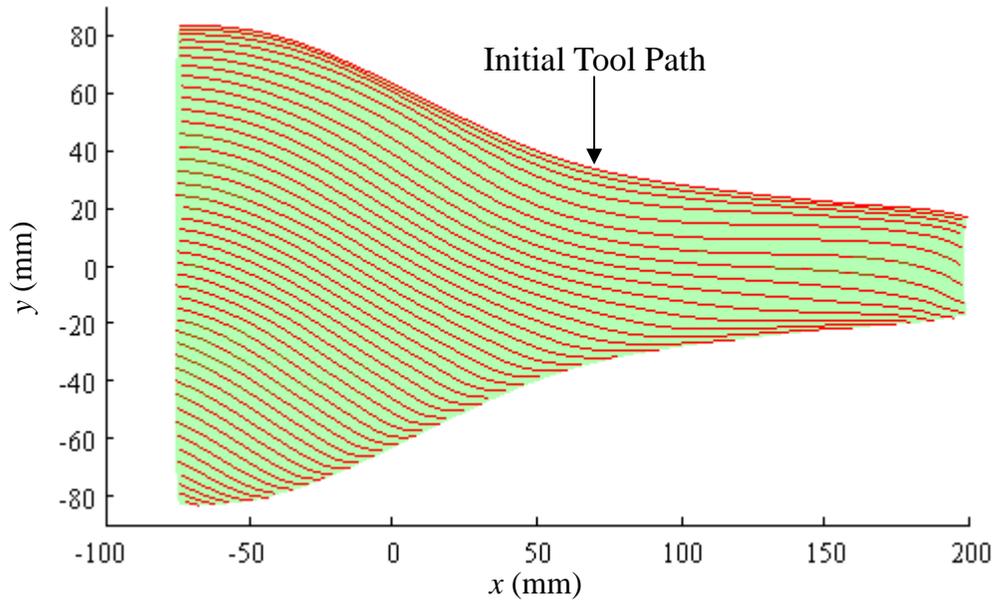


(a)

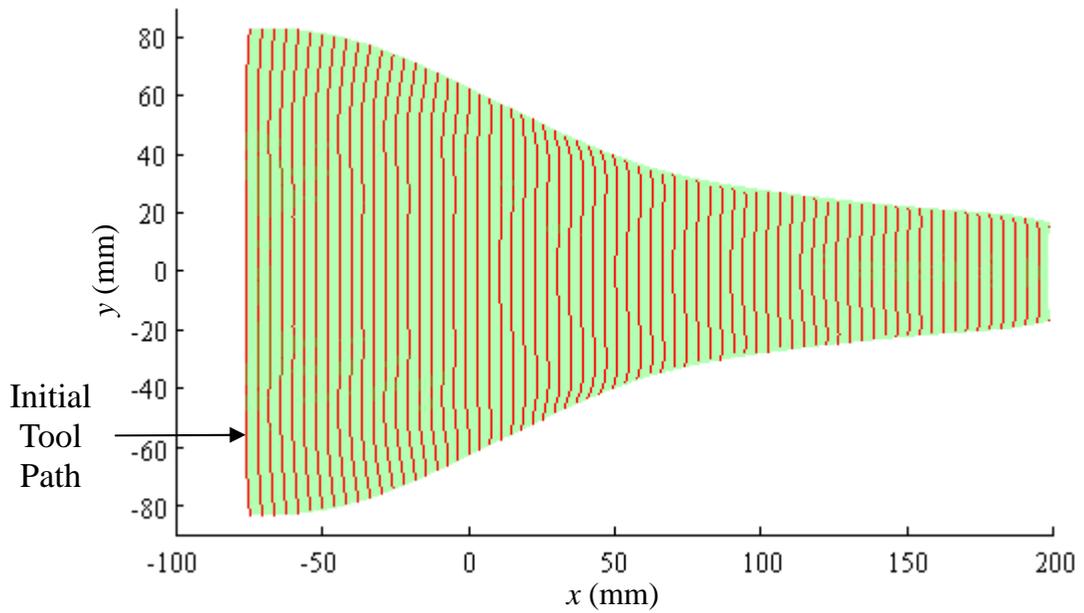


(b)

Figure 46 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-parametric, b) Iso-planar methods.



(a)



(b)

Figure 47 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), a) Iso-scallop u border, b) Iso-scallop v border.

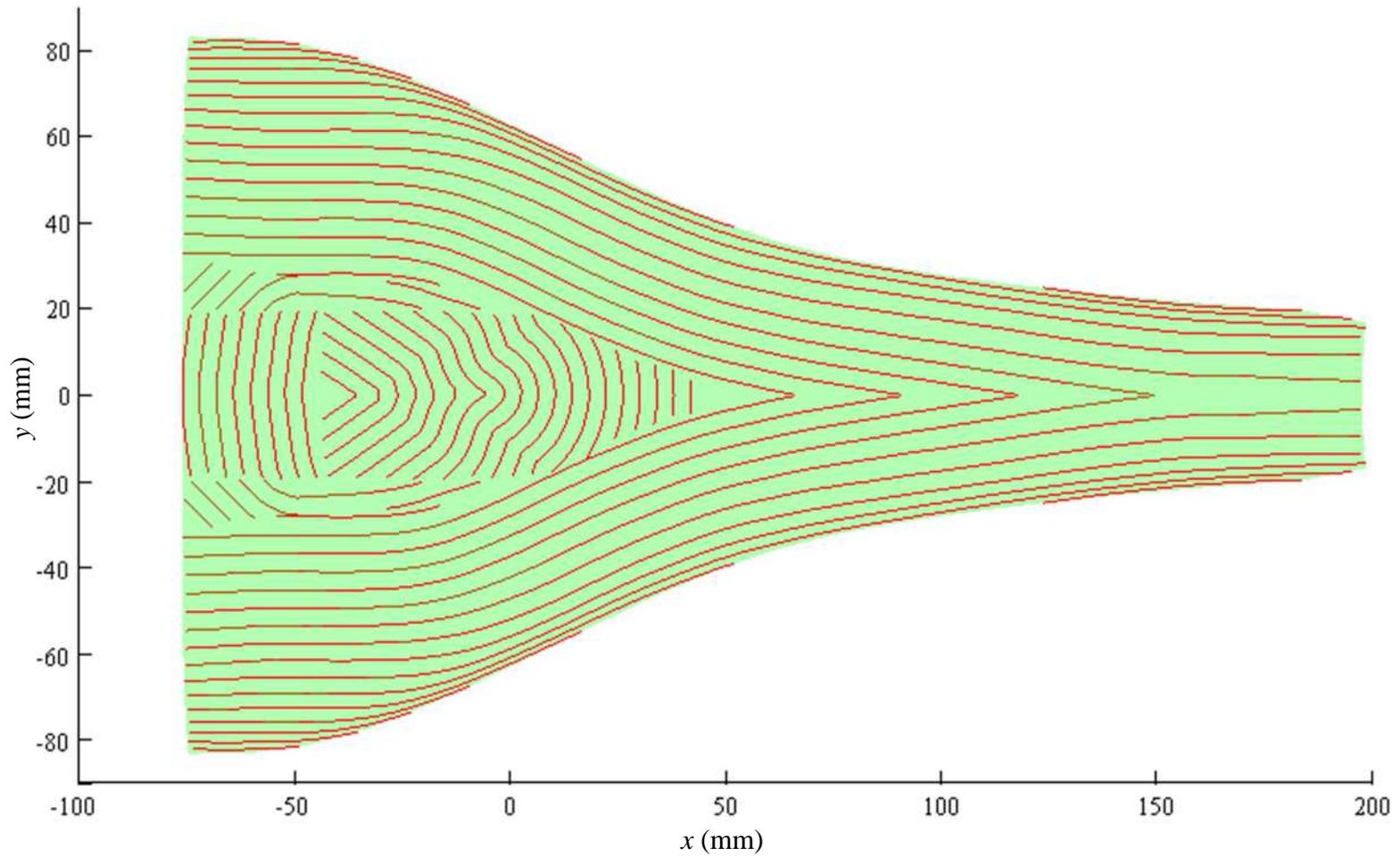


Figure 48 Tool paths of Case Study 6 (practical case), $r = 10$ (mm), $h = 0.2$ (mm), proposed method (tool paths only).

Machining Method	Case Study 1 : No Degenerate Points		Case Study 2 : One Trisector Point		Case Study 3 : Two Wedge Points		Case Study 4 : One Merged Wedge Point		Case Study 5 : Three Trisector, Two Wedge, One Merged Wedge Points		Practical Case Study : Bicycle Seat	
	Tool path length [mm]	Improvement [%]	Tool path length [mm]	Improvement [%]	Tool path length [mm]	Improvement [%]	Tool path length [mm]	Improvement [%]	Tool path length [mm]	Improvement [%]	Tool path length [mm]	Improvement [%]
Iso-parametric	10,424	-39.6%	28,978	-61.1%	8,580	-9.0%	13,722	-16.9%	24,934	-37.6%	18,464	-48.9%
Iso-planar	8,211	-23.3%	14,165	-20.4%	8,352	-6.5%	13,658	-16.5%	24,781	-37.2%	16,269	-42.0%
Iso-scallop, u	6,978	-9.8%	12,049	-6.4%	8,227	-5.1%	12,050	-5.4%	16,770	-7.2%	10,810	-12.7%
Iso-scallop, v	6,461	-2.6%	11,887	-5.1%	7,964	-1.9%	12,050	-5.4%	16,949	-8.2%	9,762	-3.3%
Proposed	6,294	-	11,277	-	7,810	-	11,403	-	15,565	-	9,436	-

Table 1 Results comparison with proposed method, $r = 10$ (mm), $h = 0.2$ (mm).

6. Conclusions

A new method has been proposed in this work to generate iso-scallop tool paths that attempt to follow the preferred feed directions of a sculptured surface. The generated tool path has the shortest overall tool path length for machining a given free-form surface, compared with all existing tool path generation methods. The generated tool paths attempt to follow the preferred feed direction (PFD) at each CC point as closely as possible. This is achieved by:

- a. segmenting the PFD field into regions of similar PFD's based on degenerative point identification and analysis; and
- b. using the principal tool path as the starting tool path for a segmented PFD region in order to reduce PFD drift on the generated iso-scallop tool paths.

As shown from the case studies, this method benefits from surfaces that have a large number of features such as mounts and valleys, where each region may have a preferred flow. This method is also useful for cases where the difference between the maximum and minimum W are notable. Such is the case as case studies 1 and 6, where machining in a single flow may always be contrary to the preferred direction, even if the iso-scallop method is applied.

This method does not require input from the user regarding an initial tool path; hence a sculptured surface will have a single solution using this method. This method is also applicable to the five-axis machining cases because a PFD field can also be generated. Also, this method can be applied for cases where W is not the defining metric to define the preferred direction in a sculptured surface. For example, the CNC machine's kinematics may be considered or the forces applied on the tool.

There are some of the limitations when using the proposed method. This method relies on the resolution chosen for analyzing the surface that defines the size of the grid of the sampled points used to detect the PFD field's degenerate points. Computation speed is lowered if the resolution is too high. On the other hand, degenerate points may not be detected if the resolution is too low. An optimal resolution that is suitable for a general case has not been determined. Machining step resolution also affects the flow of the separatrices and errors may accumulate. Lastly, the method to find the initial tool path for each region depends on the amount of pathlines that cover the region and that is also determined by an input resolution. Finding the middle curve as the principal tool path does not take into account the magnitude of the machining strip width. This value may provide information on finding an initial tool path that may further reduce the overall length of the adjacent tool paths.

References

- [1] Loney GC, Ozsoy TM. (1987) NC machining of free form surfaces. *Computer-Aided Design* 1987;19 2):85-90.
- [2] Elber G, Cohen E. (1994) Toolpath generation for freeform surface models. *Computer-Aided Design* 1994;26 6):490-6.
- [3] Huang Y, Oliver JH. (1994) Non-constant parameter NC tool path generation on sculptured surfaces. *International Journal of Advanced Manufacturing Technology* 1994;9:281-90.
- [4] Suresh K, Yang DCH. (1994) Constant scallop-height machining of free-form surfaces. *ASME Journal of Engineering for Industry* 1994; 116:253-9.
- [5] Lin RS, Koren Y.(1996) Efficient tool-path planning for machining free-form surfaces. *ASME Journal of Engineering for Industry* 1996; 118:20-8.
- [6] Sarma R, Dutta D. (1997) The geometry and generation of NC tool paths. *ASME Journal of Mechanical Design* 1997;119:253-8.
- [7] Feng HY, Li H. (2002) Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer-Aided Design* 2002;34:647-654.
- [8] Tournier C, Duc E. (2002) A surface based approach for constant scallop height tool-path generation. *International Journal of Advanced Manufacturing Technology* 2002;19: 318–24.
- [9] Ding S, Mannan MA, Poo AN, Yang DCH, Han Z. (2003) Adaptive iso-planar tool path generation for machining of free-form surfaces. *Computer-Aided Design* 2003;35:141-153.
- [10] Yoon JH. (2005) Fast tool path generation by the iso-scallop height method for ball-end milling of sculptured surfaces. *International Journal of Production Research* 2005;43:4989–98.
- [11] Yuwen S, Dongming G, Zhenyuan J, Haixia W. (2005) Iso-parametric tool path generation from triangular meshes for free-form surface machining. *International Journal of Advanced Manufacturing Technology* 2006; 28:721–726
- [12] He W, Lei M, Bin H. (2009) Iso-parametric CNC tool path optimization based on adaptive grid generation. *International Journal of Advanced Manufacturing Technology* 2009; 41:538–548
- [13] Chiou CJ, Lee YS (2002). A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Computer-Aided Design* 2002; 34 (5), 357–371.

- [14] Kim T, Sarma SE. (2002) Toolpath generation along directions of maximum kinematic performance; a first cut at machine-optimal paths. *Computer-Aided Design* 2002;34:453-468.
- [15] Giri V, Bezbaruah D, Bubna P, Choudhury AR. (2005) Selection of master cutter paths in sculptured surface machining by employing curvature principle. *International Journal of Machine Tools & Manufacture* 2005;45:1202–1209.
- [16] Agrawal RK, Pratihari DK, Choudhury AR. (2006) Optimization of CNC isoscallop free form surface machining using a genetic algorithm. *International Journal of Machine Tools & Manufacture* 2006; 46: 811–819.
- [17] Quinsat Y, Sabourin L. (2006) Optimal selection of machining direction for three-axis milling of sculptured parts. *International Journal of Advanced Manufacturing Technology* 2006; 27:1132–1139.
- [18] Vijayaraghavan A, Hoover AM, Hartnett J, Dornfeld DA. (2009) Improving endmilling surface finish by workpiece rotation and adaptive toolpath spacing. *International Journal of Machine Tools & Manufacture* 2009;49:89–98.
- [19] Johanna Senatore J, Segonds S, Rubio W, Dessein G. (2012) Correlation between machining direction, cutter geometry and step-over distance in 3-axis milling: Application to milling by zones. *Computer-Aided Design* 2012; 44:1151–1160.
- [20] Lee YS, Ji H. (1997) Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining. *International Journal of Production Research* 1997; 35 (1), 225–252.
- [21] Lee YS. (1998) Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining. *Computer-Aided Design* 1998; 30 (7), 559–570.
- [22] Chen Z, Dong Z, Vickers GW. (2003) Most efficient tool feed direction in three-axis CNC machining. *Journal of Integrated Manufacturing Systems* 2003, vol. 14, pp. 554-566.
- [23] Chen Z, Dong Z, Vickers GW. (2004) A new principle of CNC tool path planning for three-axis sculptured part machining- a steepest-ascending tool path. *ASME Journal of Manufacturing Science and Engineering* 2004, vol. 126, pp.515-523.
- [24] Yoon JH. (2007) Two-dimensional representation of machining geometry and tool path generation for ball-end milling of sculptured surfaces. *International Journal of Production Research* 2007; 45(14):3151–3164.
- [25] Fard MJB, Feng HY (2010). Effective determination of feed direction and tool orientation in five-axis flat-end milling. *Journal of Manufacturing Science and Engineering* 2010; 132, 061011-1-10.

- [26] Delmarcelle T, Hesselink L. (1993) Visualizing second-order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25-33, 1993.
- [27] Delmarcelle T, Hesselink L. (1994) The Topology of Symmetric, Second-Order Tensor Fields. *IEEE Visualization '94 Proceedings*, IEEE Computer Society Press 1994, pp. 140-147.
- [28] Delmarcelle T. The Visualization of Second-Order Tensor Fields. PhD thesis, Standord University, 1994.
- [29] Tricoche X, Scheuermann G. (2003) Topology Simplification of Symmetric, Second-Order 2D Tensor Fields. *Geometric Modeling for Scientific Visualization*, G. Brunnet, B. Hamann, H. Muller, and L. Linsen, eds., pp. 171-184, Springer, 2003.
- [30] Rote G. (1991). Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters* 1991, v. 38, pp. 123-127.
- [31] Piegl L . and Tiller W., *The NURBS Book*, 2nd edn. Springer-Verlag, New York, 1997.

Appendix

Appendix A Bicycle CAD Model Renderings

These rendering images are from the bicycle CAD model used for Case Study 6.



Source: <http://grabcad.com>



Source: <http://grabcad.com>