Proceedings of the Canadian Society for Mechanical Engineering International Congress 2021
CSME Congress 2021
June 27-30, 2021, Charlottetown, PE, Canada

SLICING PROCEDURE FOR NON-PLANAR FUSED FILAMENT FABRICATION

Muhammad Salman Chaudhry
Lassonde School of Engineering
York University
Toronto, Canada
salman01@yorku.ca

Aleksander Czekanski*
Lassonde School of Engineering
York University
Toronto, Canada
alex.czekanski@lassonde.yorku.ca

***Abstract*—In this work, we present a slicing methodology for the generation of an extruder toolpath for 3D printing. The toolpath coordinates are generated by slicing an STL model with a non-planar surface. The methodology together with non-planar slicing algorithm presented were used to slice and fabricate various CAD models with curved layers.**

***Index Terms*—3D printing, toolpath, slicing, G-code**

I. INTRODUCTION

Recently, we have witnessed the evolution of 3D (three-dimensional) printing from its use as a tool for simple rapid prototyping to the delivery of customized solutions in many different areas. The use of 3D printing has been growing across sectors to address complex and specific applications in manufacturing. For instance, in medicine, 3D printing is being employed to fabricate customized prosthetics and orthopaedic devices [1] [2]. 3D bioprinting is even being used to create artificial tissues such as bone, skin, and cartilage [3]. 3D printing or layered manufacturing is also being applied to fabricate functional parts in the aerospace and defence industries [4]. To further reap the benefits of 3D printing as a rapid manufacturing component in many existing technologies and industries, 3D printing itself needs to continue evolving to better suit the challenges and complexities faced in successfully delivering customized solutions. In short, there is a real need to redevelop generic 3D printing technology for highly complex and customized engineering applications.

3D printing is based on the principle of layered manufacturing. Each layer of material is deposited as a thinly sliced cross-section of an object to create a complete 3D structure. Prior to printing, a CAD model for the part is sliced in what is called a slicing procedure. Here, the part's cross-section is realized in each slice and toolpath coordinates are generated [5]. Presently, each slice of the model is planar, meaning that the slices are horizontal. In other words, the slicing surface is the X–Y plane of the build space. These horizontal layer slices result in the staircase effect [6], which affects the overall visual aesthetics of the resulting part. Moreover, the fibre direction with respect to the applied direction of loading can seriously affect the structural integrity of the critical part [7] [**?**] [**?**].

When printing a part in planar layers or slices, these effects can be ignored depending on the final application of the part. However, in specific applications, these effects may carry critical importance. To deliver truly customized solutions with utmost design freedom, 3D printing needs more flexibility in the printing design process. While many of the printing parameters in current 3D printing applications (i.e., slicer software) can be selected in the design process, the resulting part is still fabricated in planar layers. In this work, we address this challenge by presenting a procedure that can be adopted and further expanded for 3D printing in a curved layer (see Figure 1. Using this approach, ready-to-print G-code files based on non-planar slicing surfaces can be generated. The visualized G-code for the toolpath and some actual test results are also shared.

II. LITERATURE REVIEW

Planar deposition of the build layer can result in a structure that is less resilient to directional loading such as tensile stress, leading to delamination. Non-planar sliced parts can take advantage of the varying orientation of the contact surfaces between layers to produce a mechanically resilient structure. The non-planar layer decomposes the axial loading into tensile and shearing components, which increases with increased displacement. The most commonly used slicing techniques involve slices based on uniform layer height. It is evident that the 2.5D contours will result in the loss of the original geometry. One approach that minimizes this loss is adaptive

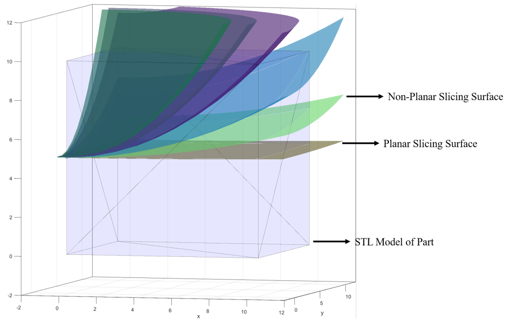

Fig. 1. Illustration for planar versus curved layer.

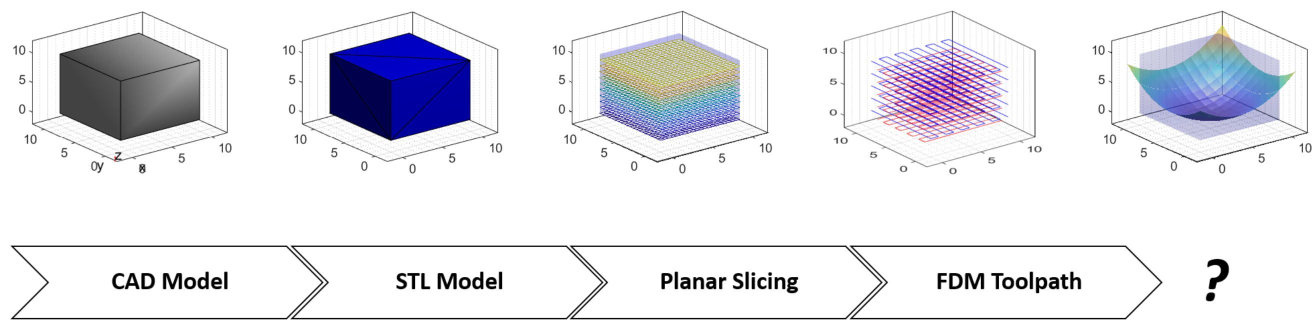

Fig. 2. Planar Printing Process

slicing. In this technique, the layer height of each slice is variable and is calculated based on the cusp height [8] (maximum deviation of a simplified rectangular stair-step profile from the intended object surface) [9] [10]. Adaptive slicing involves changing the layer thickness to better adhere to the geometry of the model in order to improve surface quality of the final product. Another approach to reducing stair-stepping is to slice the object with multi-directional toolpaths. [11] [12]. The object is decomposed into various regions, where each region is sliced and deposited in different orientations. The problem with non-linear slicing has been addressed by some past works [13] [12] [8]. The primary goal of these works was to enhance surface smoothness and reduce the stair-stepping effect in the final printed product. The first curved layer slicing algorithm was presented in [14], in which the surface is approximated parametrically to generate a toolpath. While this work presented only the algorithm, the first curved layer rapid prototyping was demonstrated by [15] [16]. In other works, the combination of planar and curved layers were printed, since not every surface in an object can be sliced using a curved parametric representation [17] [18].

III. Methodology

In this section, the procedure adopted for planar and non-planar toolpath generation is described. A toolpath is a set of instructions that controls the movement of the nozzle and the fused deposition modeling (FDM) printing process. This set of instructions is fed to the FDM machines in G-code format. The process for the realization of a 3D model in the form of G-code commands is known as slicing. There are numerous commercial and open-source slicer software packages to choose from. Most of them have similar functionality, in that the software's slicing capability is limited to planar slices.

A trivial slicing procedure starts with the STL (stereolithography tessellation language) model of the file [19]. This is the triangulated mesh representation approximating the boundary surface of the object (Figure 2). An STL file is an unstructured collection of triangular facets ζ represented by its vertices v and a uni-normal vector n in a 3D Cartesian coordinate system. The STL format of the computer-generated geometry of an object allows for accurate computation of the intersection of geometric entities.

A. Planar Toolpath Generation

To compute the FDM toolpath, the STL model is sliced into many horizontal 2D images of cross-sections. At each slice, the part model intersection with the plane representing a slice is computed to create a 2D polygon. These polygons slices, when stacked and printed with a layer thickness, approximate the 3D geometry of the part. The polygon boundary is created by computing the intersection of each facet ζ with a 2D horizontal plane. This is computationally as simple as the line-plane intersection problem in 3D geometry. The algorithm (1) loops over each facet and checks whether the intersection between the plane and the three boundaries of the triangular facets exists. If the number of points of intersection between the facet and slicing plane is equal to two, then there exists an intersection between the slicing plane and the facet plane in the form of a geometric line bounded by the two points. These line fragments together represent the boundary of the 2D sliced polygon.

In the next step, a suitable trajectory of the FDM tool is computed based on different filling styles ("infill") of the cross-sections. For this study, a 0°/90° infill pattern was employed. This means that each alternate layer of the linear fill (or hatch) of the boundary is rotated 90°.

Algorithm 1: Trivial Mesh Slicing Algorithm

```
Result: Planar G-code
for each plane $P$ in range $0, z_{max}$ do
    init slice of $P$
    for each facet $\zeta$ in $\{\zeta\}$ do
        if $\zeta$ intersects $P$ then
            calculate intersection points
        end
        if intersection points == 2 then
            add new line to slice of p;
        end
    end
end
```

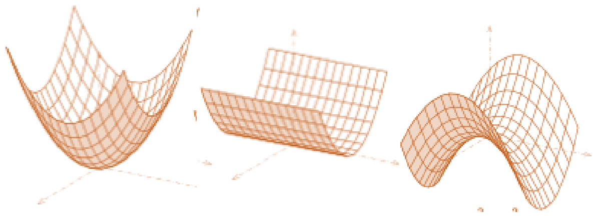
(a) (b) (c)

Fig. 3. (a) Elliptical paraboloid, (b) cylindrical paraboloid, and (c) hyperperbolic paraboloid.

B. Non-Planar Slicing and Toolpath Generation

Slicing with non-planar toolpath generation starts with the definition of the slicing surface. The complexity of the choice of slicing surface revolves around the mathematical representation and computation (intersection) and aesthetic feasibility. The final processed part's visual appeal and smoothness will vary as a result of the geometric characteristics of the surface. Another challenge is approximating and mathematically defining a particular characteristic surface of a part. This study employs quadratic surfaces in the implicit form, which can be solved for the variable z in terms of x and y to obtain the explicit form of the slicing surface (eq. 1). The computation involving quadratic surfaces is simple and well defined in the literature. Later on, this will be extended by incorporating more complex surfaces in parametric form. Here, the surface is defined by parameters A, B, x_{off}, and y_{off}. Particular care is needed when selecting these parameters, and some preplanning is required to ensure the alignment and intersection of the slicing surface with the part's cross-section.

$$S(x,y) = z = A(x - x_{off}) + B(y - y_{off}) - z_{off} \tag{1}$$

The magnitude and polarity of A and B also yield various forms of paraboloids. For example, $B = 0$ yields a cylindrical paraboloid, $A > 0$ and $B > 0$ form an elliptical paraboloid, and $A > 0$ and $B < 0$ result in a hyperbolic paraboloid (see Figure 3).

In the execution of the non-planar algorithm 2, the parabolic surface is offset along the build volume (axis normal to the build plate) of the 3D printing setup and constrained by z_{min} and z_{max}. This sequential offset in the z-direction is equal to the layer height of the slice. At a particular z_{off}, the intersection of each facet ζ with the slicing surface is computed. Each facet is a triangular element defined by three vertices v^i for $i = 1, 2, 3$ and a unit normal. The intersection curve of each ζ and the slicing surface $S(x, y)$ is bounded by points that lie on the ζ boundary. These intersection points are computed by solving a system of equations involving a line in 3D parametric form (eq. 2) and slicing function $S(x, y)$ for a parametric line representing the two vertices of ζ and the slicing function $S(x, y)$. Through substitution, the line surface intersection problem is reduced to a quadratic equation that can be solved to find parameter t. Each facet ζ intersection with the slicing surface can result in either no point of intersections, one point of intersection, or two points of intersection, in which case each point is labelled as the start or end point.

$$\begin{aligned} &v_x^1 + t(v_x^2 - v_x^1) \\ &v_y^1 + t(v_y^2 - v_y^1) \\ &v_z^1 + t(v_z^2 - v_z^1) \end{aligned} \tag{2}$$

$$Jt^2 + Kt^2 + Lt = 0 \tag{3}$$

$$\begin{aligned} J &= A(v_x^2 - v_x^1)^2 + B(v_y^2 - v_y^1)^2 \\ K &= 2A(v_x^1 v_x^2 - (v_x^1)^2 + v_x^2 x_{off} - v_x^1 x_{off}) \\ &\quad + 2B(v_y^1 v_y^2 - (v_y^1)^2 + v_y^2 y_{off} - v_y^1 y_{off}) \\ &\quad - 2(v_z^2 - v_z^1) \\ L &= A((v_x^1)^2 + 2v_x^1 x_{off} + x_{off}^2 \\ &\quad + B((x_y^1)^2 + 2v_y^1 y_{off}) - v_z^1 - z_{off}) \end{aligned} \tag{4}$$

Once all the intersection points on a slicing surface are computed, they are used to define the convex hull boundary of the cross-section on the slicing surface. To do so, we adopt an intuitive *lifting transform* approach inspired by the problem of finding the Delaunay triangulation of a set of points in d-dimensional space. This can be converted to the problem of finding the convex hull of the point when mapped onto hyperparaboloid space of $d+1$ dimensions [20]. In the adopted approach, a *dropping transform* technique is applied, whereby the intersection points x^* of dimension d are projected onto dimension $d - 1$. The main aim in this part of the procedure is to realize the polygon boundary of the cross-section and to create an infill pattern on a defined surface. The computation of a convex hull is much simpler in two dimensions. Once the desired toolpath is generated, it can be mapped onto a 3D surface represented by the slicing function, as illustrated in Figure 4.

Raster Generation

Polygon generation on a slicing surface is followed by the computation of the toolpath to fill the polygon boundary. For

Algorithm 2: Non-Planar Slicing Algorithm

```
Result: Non-Planar G-code
$initializeSurfaceParameters, A, B, x_{off}, y_{off}, z\prime$
for each plane $P$ in the range $z_{min}, z_{max}$ do
    init slice of $P$
    for each facet '$\zeta$' in $\{\zeta\}$ do
        if $\zeta$ intersects $P$ then
            calculate intersection points
        end
        if intersection points == 2 then
            add new line to slice of p;
        end
    end
end
```

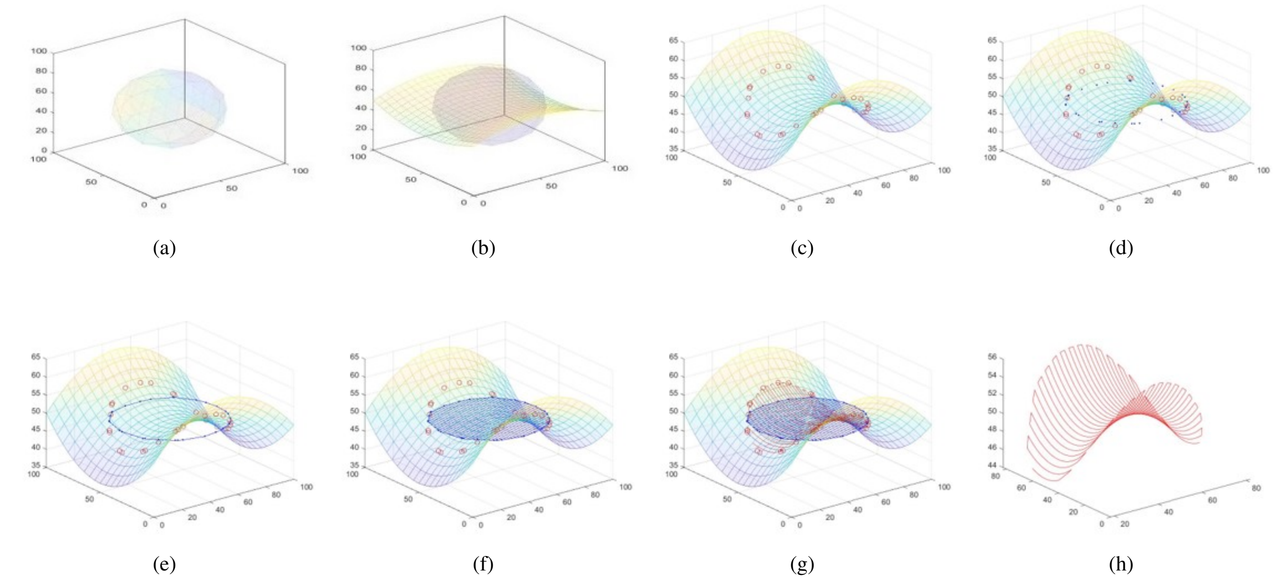

Fig. 4. (a)The slicing process starts with an STL model of the object: (b) at a specific z offset, a predetermined surface patch is realized; (c) at a specific plane offset, the intersection of facets in the STL model is determined; (d) the intersection points on each slice are transformed to two dimensions for toolpath generation; (e) a closed-loop perimeter boundary is determined for the cross-section; (f) the desired toolpath is generated on a 2D slice; (g) the 2D toolpath is projected back onto the predetermined parabolic surface patch to yield a non-planar toolpath; (h) the resulting G-code toolpath on a non-planar surface.

this study, we have taken into account a linear-type raster fill with alternating 0°/90° fibre direction. For a printing nozzle with diameter d_n and layer height h_l , the procedure for this is as follows: for the polygon defined by x^* in a slice, the bounding box is computed (coordinates of the quadrilateral bounding x^*'s), which is split into n vertical lines, where n is the length of the bounding box divided by line width. Here, line width or extrusion width w_e is set to the nozzle diameter $d_n * 1.20$.

For toolpath generation along the infill, the coordinates are generated along these vertical lines. At the intersection of the infill line and polygon boundary, the tool is directed to the next line, filling the the polygon boundary completely. For each alternating slice, the lines are generated along the vertical axis, thus rotating the fibre direction at 90°.

Finally, after generating a toolpath route in a dropped-down projection of the 3D slice, the coordinates are mapped back (*lifting transform*) to the slicing surface, mapping the toolpath in a 3D non-planar slice. However, there is a loss of curvature for curves with a high degree of gradient as a result of mapping to and back from the planar projection plane. This is also true when the travelling coordinates are located at a greater distance. For this reason, a smoothness factor is incorporated, where each millimeter of tool-travelling distance is interpolated to approximate a smooth curve.

Flow Calculations

Each G-code command contains instructions for the position of the nozzle and also the amount of filament to be extruded along the path. A typical G-code command of G0 X1 Y1 Z1 E1 would mean a movement to coordinates $(1, 1, 1)$ while extruding 1 unit of filament. The E value is the length of filament with a diameter d_f needed to fill the volume defined by the print area A_e and path length L_e. In Slic3r software, the print area is approximated by a rectangular center with a semi-circle on either side [**?**]. In this study, the cross-section of the printed fibre or the print area A_e is represented by an ellipsoidal cross-section along the travelling path length of the nozzle. by an ellipsoidal cross-section or extrusion area A_e across the length of the travelling motion. The height of this rectangle is the layer height and its width is the deposition line width.

$$A_e = \pi w_e h_l \tag{5}$$

$$E = \frac{A_e L_e}{\pi(\frac{{d_f}^2}{2})} \tag{6}$$

where E is the length of extrusion and L_e is the path length to be travelled from the current coordinates to the position defined in the G-command. The final G-code output will be implemented according to formatting and the firmware of the environment.

In this section, we presented a series of steps involved in slicing a simple CAD model with a non-planar surface such as a paraboloid. This procedure can be adopted according to the software and hardware interface being implemented in an FDM setup.

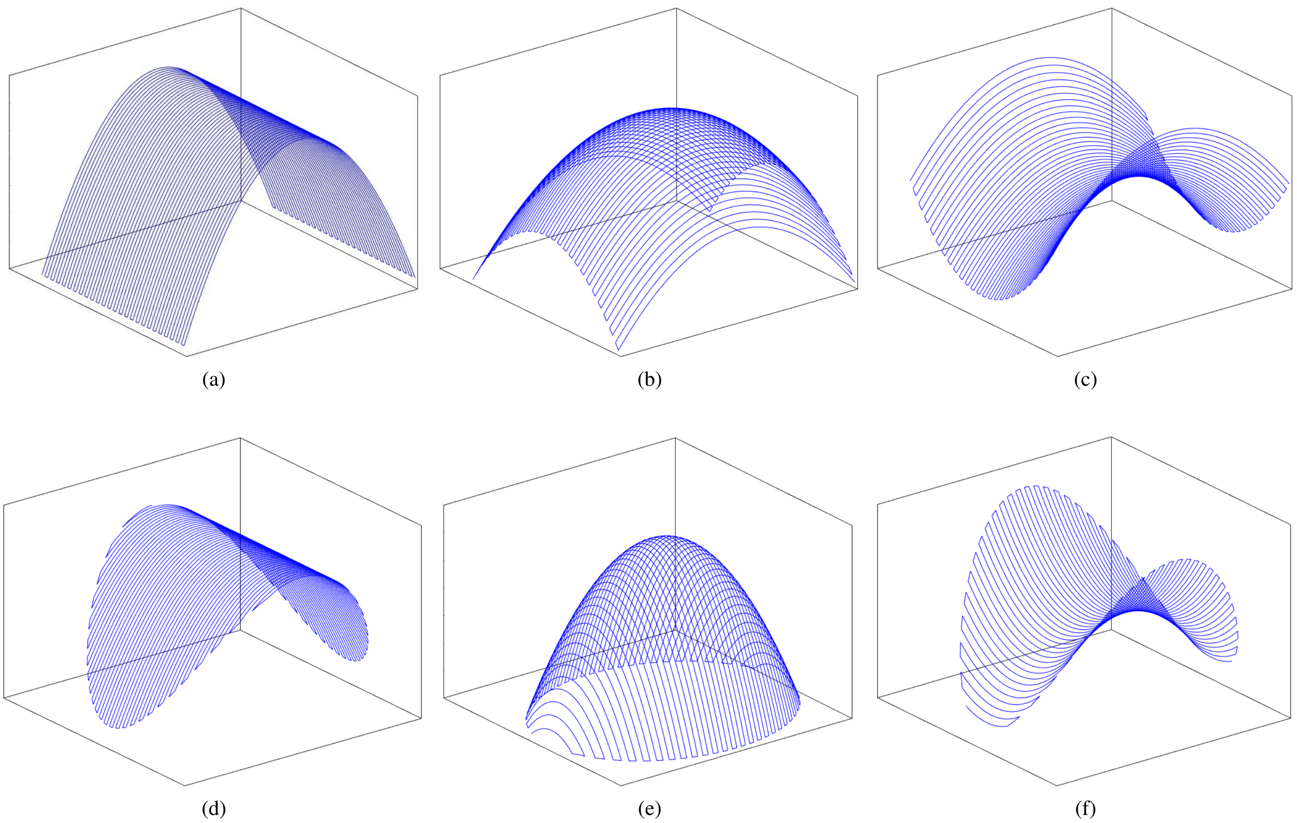

Fig. 5. (a)-(c) shows GCODE tool-path for a rectangular cross section slice, (d)-(f) shows GCODE tool-path for a circular cross section slice.

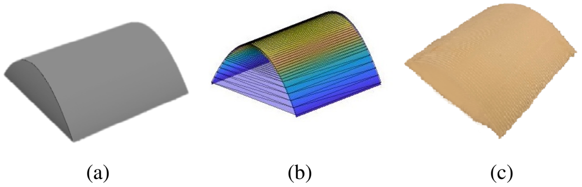

Fig. 6. (a) CAD model for test case 1; (b) visualization of the STL model of test case 1 with superimposed slicing surface; (c) final printed model for test case 1.

IV. Results and Conclusions

In the previous section, a procedure for the generation of non-planar toolpath was presented. In this section, the given procedure is implemented on 3D printing platforms (Cura, Dobot). The slicing of STL models was programmed in MATLAB. It is possible to provide movement coordinates to the tool in 3D space; however, given that all of the FDM machines are built on the planar layer principle, there is a real constraint of the degree of curvature that can be printed owing to possible interactions with the extruder body. This is also the reason that many CAD models can be sliced but only a handful can be printed as of yet. The addition of degrees of freedom in FDM technology and the introduction of more slicing surfaces, as presented in this study, will create many new opportunities and applications in advanced manufacturing.

To demonstrate the applicability of the given procedure, a simple CAD model was sliced with a cylindrical paraboloid surface (Figure 6a). For preplanning, a GUI utility was developed in MATLAB that plotted the STL model and slicing surface in a single plot (Figure 6b). The surface input parameters used for this example are $A = -0.02$ and $B = 0$. This helped to position and align the slicing surface in order to achieve an optimum outlook of the slices, which are then printed on a desktop FDM machine (Dobot Magician) (Figure 6c). Figure 5 illustrates some more examples of the toolpath G-code generated for a square (Figure 5a-c) cross-section and circular (Figure 5d-f) cross-section. For illustration purposes only, one slice is shown from the complete models computed for (Figure 5a-f).

In conclusion, printing by means of a customized slicing function can completely prevent the stair-stepping effect inherent to available 3D printing technology. Moreover, using such a procedure can produce parts with unparalleled visible aesthetics. However, the real-world applications of this approach are limited to 3D printed structures that incorporate paraboloid surfaces, such as a parabolic dish, optical lenses, and architecture. The main aim of this study was to explore the execution of a custom slicing procedure for one particular slicing function. With the addition of more geometrically flexible and mathematically complex slicing functions, the possible applications are boundless.

References

[1] F. Auricchio and S. Marconi, “3d printing: clinical applications in orthopaedics and traumatology,” *EFORT Open Reviews*, vol. 1, pp. 121–127, May 2016.

[2] A. Manero, P. Smith, J. Sparkman, M. Dombrowski, D. Courbin, A. Kester, I. Womack, and A. Chi, “Implementation of 3d printing technology in the field of prosthetics: Past, present, and future,” *International Journal of Environmental Research and Public Health*, vol. 16, p. 1641, May 2019.

[3] D. G. Tamay, T. D. Usal, A. S. Alagoz, D. Yucel, N. Hasirci, and V. Hasirci, “3d and 4d printing of polymers for tissue engineering applications,” *Frontiers in Bioengineering and Biotechnology*, vol. 7, July 2019.

[4] M. Zastrow, “3d printing gets bigger, faster and stronger,” Feb 2020.

[5] C. K. Chua and K. F. Leong, *3D Printing and Additive Manufacturing*. WORLD SCIENTIFIC, Jan. 2017.

[6] W. Oropallo and L. A. Piegl, “Ten challenges in 3d printing,” *Engineering with Computers*, vol. 32, pp. 135–148, June 2015.

[7] M. Somireddy, A. Czekanski, and C. V. Singh, “Development of constitutive material model of 3d printed structure via FDM,” *Materials Today Communications*, vol. 15, pp. 143–152, June 2018.

[8] A. Dolenc and I. Mäkelä, “Slicing procedures for layered manufacturing techniques,” *Computer-Aided Design*, vol. 26, no. 2, pp. 119–126, 1994.

[9] M. Y. Zhou, J. T. Xi, and J. Q. Yan, “Adaptive direct slicing with non-uniform cusp heights for rapid prototyping,” *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 1-2, pp. 20–27, 2004.

[10] J. Tyberg and J. H. Bøhn, “Local adaptive slicing,” *Rapid Prototyping Journal*, vol. 4, no. 3, pp. 118–127, 1998.

[11] D. Ding, Z. Pan, D. Cuiuri, H. Li, N. Larkin, and S. van Duin, “Multi-Direction Slicing of STL Models for Robotic Wire-Feed Additive Manufacturing,” *Solid Freeform Fabrication Symposium*, pp. 1059–1069, 2015.

[12] H. Ming Zhao, Y. He, J. Zhong Fu, and J. Jiang Qiu, “Inclined layer printing for fused deposition modeling without assisted supporting structure,” *Robotics and Computer-Integrated Manufacturing*, vol. 51, no. May 2016, pp. 1–13, 2018.

[13] D. Ahlers, F. Wasserfall, N. Hendrich, and J. Zhang, “3D printing of nonplanar layers for smooth surface generation,” *IEEE International Conference on Automation Science and Engineering*, vol. 2019-Augus, no. September, pp. 1737–1743, 2019.

[14] D. Chakraborty, B. A. Reddy, and A. R. Choudhury, “Extruder path generation for curved layer fused deposition modeling,” vol. 40, pp. 235–243, 2008.

[15] S. Singamneni, A. Roychoudhury, O. Diegel, and B. Huang, “Modeling and evaluation of curved layer fused deposition,” *Journal of Materials Processing Technology*, vol. 212, no. 1, pp. 27–35, 2012.

[16] B. Huang, S. Singamneni, and O. Diegel, “Construction of a curved layer rapid prototyping system: Integrating mechanical, electronic and software engineering,” *15th International Conference on Mechatronics and Machine Vision in Practice, M2VIP’08*, pp. 599–603, 2008.

[17] T. Llewellyn-Jones, R. Allen, and R. Trask, “Curved Layer Fused Filament Fabrication Using Automated Toolpath Generation,” *3D Printing and Additive Manufacturing*, vol. 3, no. 4, pp. 236–243, 2016.

[18] R. J. Allen and R. S. Trask, “An experimental demonstration of effective Curved Layer Fused Filament Fabrication utilising a parallel deposition robot,” *Additive Manufacturing*, vol. 8, pp. 78–87, 2015.

[19] *User Guide to Rapid Prototyping*. Society of Manufacturing Engineers, 2004.

[20] C. Toth, *Handbook of Discrete and Computational Geometry, Third Edition*. Boca Raton, FL: CRC Press, 2017.