

Received: 04 May 2022 / Accepted: 18 June 2022 / Published online: 21 June 2022

*additive manufacturing, slicing,
helix 3D, printer*

Tugdual Le NEEL^{1,2*}
Tarek MESTO^{2,3}
Jean-Yves HASCOET^{1,2}

A NOVEL SLICING STRATEGY FOR CONTINUOUS PRINTING WITH A HELIX 3D PRINTER

Additive manufacturing is an essential solution in the production of parts. Model slicing is an important step of the 3D printing process. The slicing of the layers is the core part of the additive manufacturing because it transforms the 3D model to a 2D profile layer for the printer to manufacture. A novel machine architecture deposits with a helical path. The helical architecture provides a continuous rotation that allows printing continuously without any interruption. Therefore there are no more starting and ending point at each layer. This paper proposes a slicing method compatible with this type of machine. Continuous printing is made as a function of z -level, so at each angle of rotation, the level of z will be incremented. Finally, these disks can be combined as one image to be sent to the ink-jet as a continuous printing. To illustrate this novel slicing methodology a model is sliced.

1. INTRODUCTION

Additive manufacturing (AM) is a new type of rapid manufacturing. Because now, it is progressing into manufacturing. The idea behind additive manufacturing is printing layer by layer of material until it reaches the full model. Additive manufacturing can be split into two categories: direct AM which is printing the product itself, and indirect AM which produces a tool to make the desired part (such as printing molds and cores to be used in casting) [1].

There are two main aspects in additive manufacturing machines, the mechanical system and the software. The software portion has to transform the 3D model into toolpath trajectory code which pilots the machine depending on the machine's architecture. The mechanical system can be of different types to move in the x,y,z coordinates system.

The 3D printing movement can be achieved by several mechanisms. Such as a 2.5D 3D printer that uses a gantry system with an extrusion head or a bed with a binder jet print head [1], or a 5D-axis machine that uses a robot for more complexity.

¹ Ecole Centrale de Nantes, Nantes, France

² Ecole Centrale Nantes, CNRS, GeM, France

³ SATT Ouest Valorisation, Nantes, France

* tugdual.le-neel@ec-nantes.fr

<https://doi.org/10.36897/jme/151119>

Binder jet additive manufacturing is a type of powder bed AM with no fusion-based, thus it has no residual stresses [2]. Unlike the extrusion type of AM, the binder jet has a powder bed that is mostly of sand and newly metallic material [2]. The powder is smoothly laid in a specified thickness layer and the binder is selectively added. The print head attached to the machine jets the binder onto the material to accumulate and form the 3D object.

Binder additive manufacturing can be used as a method for indirect AM. This method is often used in conjunction with a gantry system. The 3-axis of motion moves the tools in the x - y direction, and the bed in the z -axis as to increment the layers. Three main companies focus on the production of sand molds by binder jetting: Voxeljet, ExOne, and Viricast 3D [1]. Only one company, Viricast, uses a six-axis robot as 3D for manufacturing and the others use the cartesian architecture.

Rather than the classical method that deposits a layer by layer by changing the level, and then starts printing the next layer. A novel machine concept was proposed by Hascoët et al. [3] which is designed for continuous printing. The concept shown in Fig. 1 is a rotating arm containing an arrangement of print heads and sand hoppers all along the arm. This arm is mounted onto a lead screw.

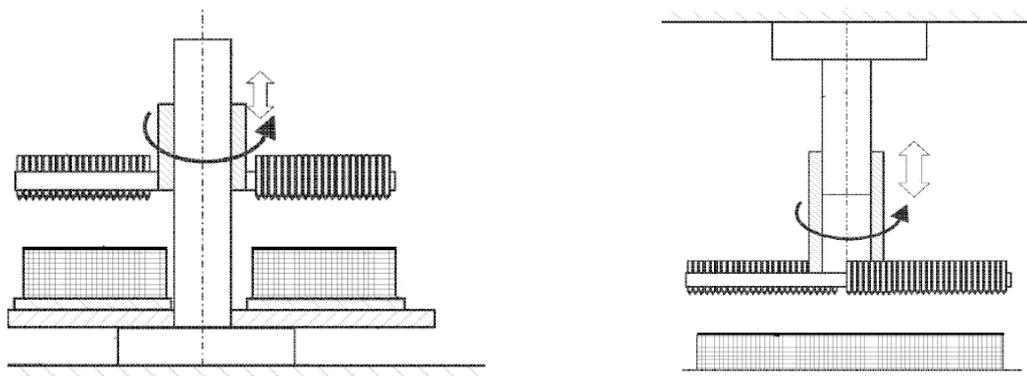


Fig. 1. Machine concept of rotating arm mechanism [3]

The motor turns the lead screw which rotates the arm to deposit the sand and jets the binder selectively to print the part. When the screw rotates, it translates upward creating a helical movement. One full revolution of the arm would print one layer of the model. This concept constantly prints and does not need to stop to increase the level. It is a one-time print.

All the 3D printing techniques have a similar challenge, that is the trajectory or image generating. The novel helix AM also has the challenge to transform a model into a toolpath trajectory. Generating a toolpath trajectory of a model is divided into two challenges: 2D slicing and contour optimization. First challenge is to 2D slice the model as to obtain the edges of the model for a single layer from an STL file format. STL file is a model format. It is one of the formats mostly used for the AM purposes as it is universal [4]. Other alternatives are not discussed here but should be considered including AMF and step-NC formats. The second challenge is dependent on the motion of the machine. For this new concept, a new methodology shall be done to fulfill the helix motion of the machine. This paper aims to make a slicing method to generate a “one-slice” output gradually increasing the helical area.

2. STATE OF ART

Slicing, contour generation, and machine tool path (such as G-code) are the core steps in additive manufacturing. G-code is the language for the machine. It enables it to move its axis and to control the tool to manufacture the part. The G-code contains instructions in X Y Z coordinates with tool deposition instructions.

The most common file used in slicing is STL format. Kirschman et al. have given a simple method for a facet searching and slicing of the triangles from an STL [5]. Tascioglu et al. have developed a similar method based on STL binary format. The model takes into account all types of intersections between the cutting plane and the object's STL meshes [6]. Then a two-point line is stored for each facet slice to facilitate the contour generation by comparing the common points. Jing Hu et al. proposed a method to get the intersection points of the sliced plane and adjust layer thickness for time computation to generate contour [7]. Also, Shafiq M. et al., Bennet et al., and Joshi et al. follow the same steps but each developed a new equation and method to improve and decrease the amount of time for the slicing procedure [3, 8, 9].

Another approach for printing is using a six-axis robot, one such example can be seen in Wire Arc Additive Manufacturing (WAAM). This relies on toolpath generation which at the beginning was a simple 2.5D toolpath [10]. Hascoët et al. have improved the toolpath to manufacture complex parts by adding parameters based on experimental results [11]. Another improvement also done by Hascoët et al. is to improve complex and curve geometry toolpath without any collapsing and defects during printing [12]. All these toolpath methods are used by three to six-axis machines, they are not applicable for the helix motion. The machine used here has a single degree of freedom motion.

Binder jet printers have equipment that projects liquid onto a granular substrate. To solidify or bind the powder on the laid granular substrate, heat or chemical reaction can be used. The 3D model is sliced into 2D layers to guide the machine and project material where it needed. Most slicing procedures are a direct 2D slice of a specific thickness for a layer [3, 5, 13]. Figure 2 illustrates the printing components of a typical binder jet machine that moves in 2.5D. The slices are individually converted into a toolpath and are then combined into one G-code file as one print.

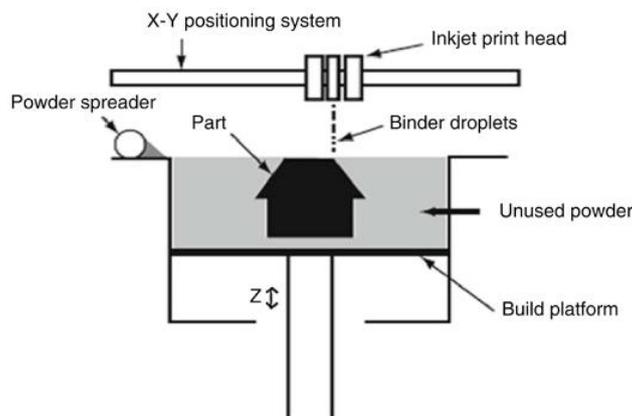


Fig. 2. Cartesian motion of a binder jetting AM machine [2]

The above method is for 2D slicing with print heads that move at a constant z -level. If applying this method to the helix machine, an error will occur due to the ever-increasing height of the layer. The novel slicing method proposed to have this variation in level and high accuracy of 0.4 mm thickness for one layer.

3. HELIX METHOD

The novel machine's mechanism will work in a continuous helical trajectory with no stops or layer shifts. Therefore the slice will be in a helix shape. The printing will take a continuous form in a homogenous way that that is not interrupted by layer shifting. The challenge is how to slice in a helical way, and how to change the conventional 2D slicing method into a helix shape. Figure 3 shows the methodology that is been followed to obtain the toolpath.

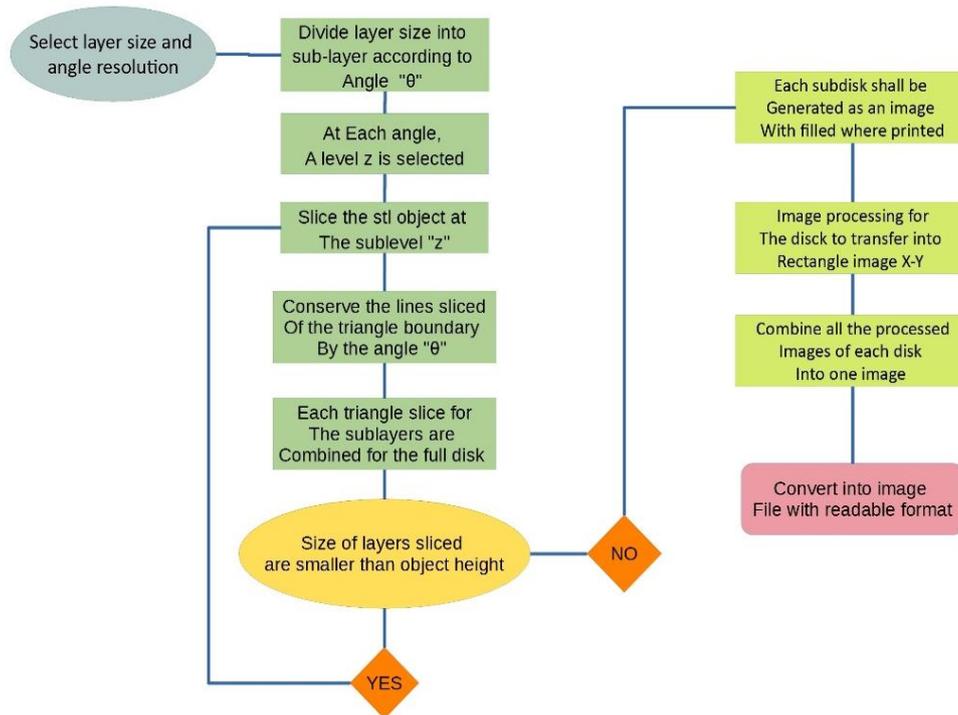


Fig. 3. The slicing methodology and conversion into a printed image

The helix with a surface shape called helicoid; its projection is a circle area Fig. 4. (a) shows the helical line and Fig. 4 (b) the helicoid surface.

The equation of a helix is as follows [14]:

$$\begin{aligned}
 x(t) &= \rho_0 \cos(\omega t + \varphi_0) \\
 y(t) &= \rho_0 \sin(\omega t + \varphi_0) \\
 z(t) &= vt = \frac{S}{2\pi} \omega t
 \end{aligned}$$

where: ω : is the rotational velocity, ρ_0 : is the radius of the helix, φ_0 : the starting phase angle (normally 0).

The three coordinates of the helical object are directly related to the z level, not like the classical 2D slice as the z level does not affect the x -axis, y -axis. At a given z -level exists only some x and y coordinates along the line of size ρ_0 with an angle ωt in each revolution. For that reason, the slicing will not be 2D layered slices, but instead a 3D slice. To do this, and to facilitate the process, multi-steps or multi-slices shall be taken instead of one single slice to obtain the sliced image. Since the binder jetting has a print head that read a 2D image, the generated form of the helical slicing is converted into an image representing the whole model that the printer can read as the procedure and the data shown in the next sections.

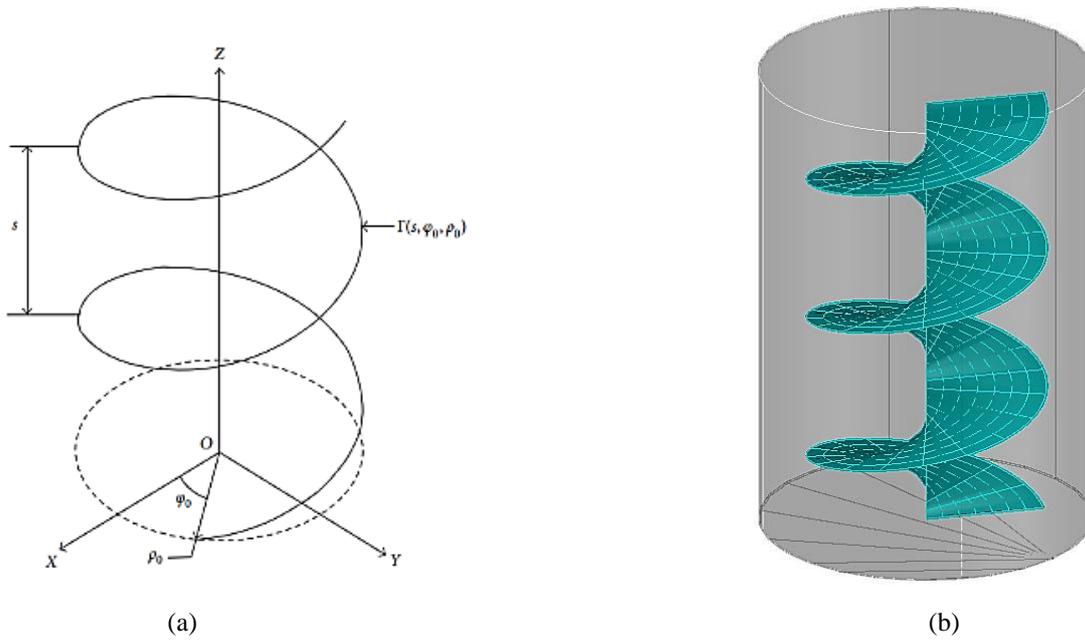


Fig. 4. (a) Helical line shape and, b) helicoid surface shape

The helix has unique x and y coordinates along with one revolution or pitch step, in which the projection of this step will be a circle. This concludes in each pitch or one revolution of a helix turn can be projected into a circle which represents one layer of the object. After the projection of all layers, it is then combined into one rectangular image. Figure 5 shows the projection disk of each layer as pitch “ s ” till “ $n \times s$ ” and one or the n revolution.

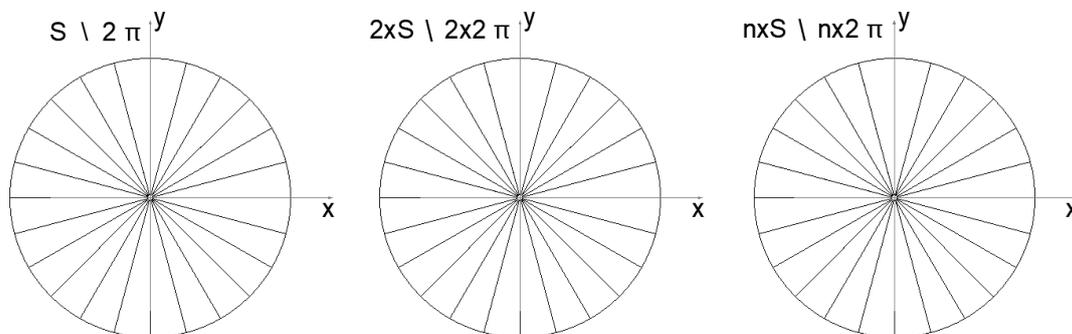


Fig. 5. Projection disk of each revolution (“ s ” the pitch of layer)

Because one revolution is a “layer”, the pitch can be determined as the height of the layer. To obtain an accurate cutting, the layer is obtained by a multi-slicing at a different levels of the helix path. To ensure the accuracy of slicing while decreasing the slicing time, the number of slices for a layer will be specified by angle ωt . This angle will allow slicing the layer into specified triangles at each level and then combining them to obtain a projection disk. The disk is then processed and transformed into a rectangular image, Fig. 6. This last step is because the print head reads a cartesian image, while the machine moves into polar form (radius-angle). The projected disk needs be flattened into cartesian as the y-axis is the radius and the x-axis is the perimeter of the disk, Fig. 6.

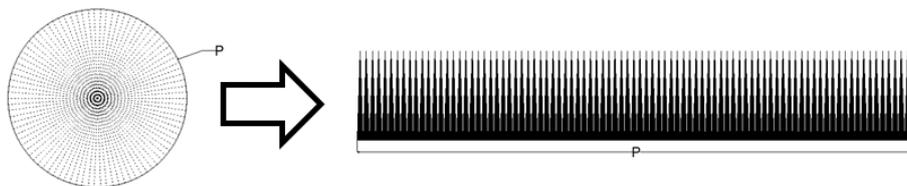


Fig. 6. Transforming the projection disk into a printed image

The machine is rotating and printing selectively. The outer side has the lowest resolution, because it moves faster. On this part the angle θ , that is the fixed angle ωt , can be obtained and several slices per layer. This leads to the following chart, Fig. 7.

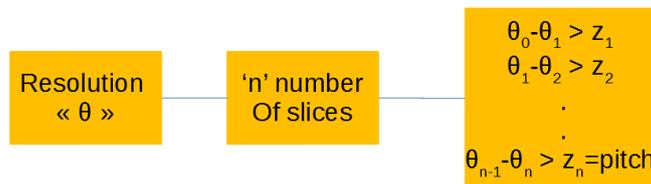


Fig. 7. Chart of slicing angle and height of a layer

The procedure of the projection for each pitch diameter depends on the resolution and the slice at each z level for the interval of angle “ ωt ”.

4. SLICING METHOD

As to illustrate the slicing method a propeller is used, Fig. 8.

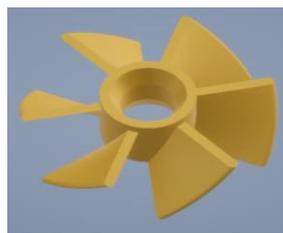


Fig. 8. 3D model of propeller shape

The STL file format is used for rapid prototyping and for this method. It conserves the 3D model by tessellation process in a form of multiple triangular facets of XYZ to represent the surface feature of the object [4].

Each triangular facet has three vertices to identify the facet nodes and geometry. The facet normal is to identify the plane. The outer loop to begin the vertex. And the end facet to indicate the end of the facet. Which from these facets form the object.

The proposed slicing algorithm is to convert each triangular facet into each respective line segment. These line segments are combined into contour lines to make the shape of the boundary for each layer. Ünver et al proposed an algorithm for slicing to check if the plane intersects with the triangle facet by simply searching the 3-points level and comparing them with the z-level and checking which type of cutting is the case, illustrated by Fig. 9 [6].

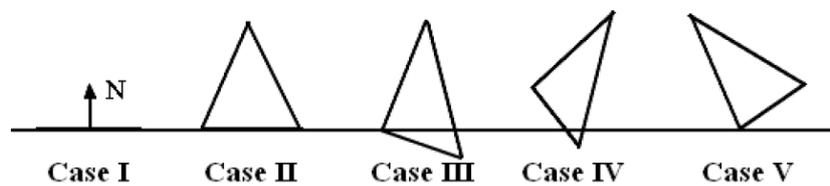


Fig. 9. Intersection cases [4]

After that, if there is a real intersection, the intersected points are calculated and memorized to be used in the contour generation. However, Shafiq et al. [4] proposed an algorithm to reduce the slicing procedure and lower the calculation time. This is used to slice and get the intersection points of each slice faster.

The triangle points or facet vertices be (P_1, P_2, P_3) , and $\bar{u}\bar{v}$ and \bar{w} are the direction vectors of the lines of the facet. Figure 10 The intersection of the plane at level “z” pass through the facet lines by 2 points $P(s_u)$ and $P(s_v)$ and the equation of the point is [4]:

$$P(s) = P_o + s(P_f - P_o)$$

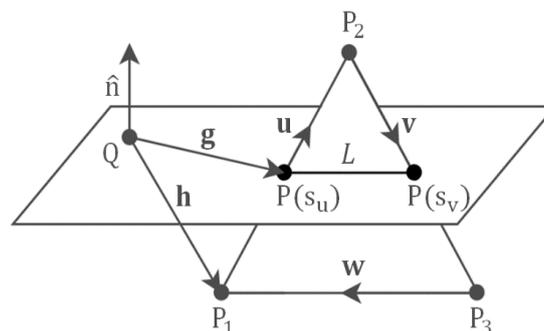


Fig. 10. Intersection between facets and z-plane [4]

With “s” is the slope of the line with the plane. The unit vector of the z-plane given by $\hat{n}(0,0,1)$ where, if there is no intersection between the vector line and the plane, $n.u = n.v = n.w = 0$. This verification is done before proceeding with the following point calculation. For an arbitrary point “Q” on the plane, the vector $Q \Rightarrow P(s)$ is $g = h + S.\bar{u}$

$$h = P_o - Q = P_1 - Q$$

$$g = P(s) - Q \text{ and } \bar{n} \cdot g = 0 = \bar{n} \cdot (h + s \cdot \bar{u}) \Rightarrow \bar{n} \cdot h = -S\bar{n} \cdot \bar{u}$$

$$S = \frac{-\bar{n} \cdot h}{\bar{n} \cdot \bar{u}} = \frac{\bar{n} \cdot (Q - P_o)}{\bar{n} \cdot (P_f - P_o)} \quad [4]$$

With S is S_u and S_v for P_1, P_2 and P_3, P_2 . The Point of the intersection is:

$$P(s_u) = P_1 + s_u(P_2 - P_1) \text{ and } P(s_v) = P_1 + s_v(P_2 - P_1) \quad [4]$$

These two points are combined to set the contour of the sliced object for a 2D slice. This is used for the 2D slice and normal contour generation. In the case of helix slicing, not all points are needed. Instead just the triangle section of the angle for the slicing is used.

For the next step, these points are selected for each triangle to be checked if they are inside the triangle or slice and the intersection of the facet triangle line by z -level, and the line boundary of the triangle. Figure 11 shows the slices of two levels of sub-layers z_1 and z_2 . The triangle OA_1A_2 is the boundary slice and all points inside it are acceptable of points “ P ” and the points “ I ” are the intersection between the lines OA_1 , OA_2 , and the intersection line of the facet with the level z .

To determine if the point is inside the triangle or not, a simple method is used to calculate the triangular area. Calculating the areas of the three triangles is done by substituting one point of a triangle with the point to check. If the summation of the triangles is equal to the total area, then the point is inside; otherwise, it is outside.

Figure 11 illustrates the summation of the triangles $\widehat{QA_1P_9}$, $\widehat{QA_2P_9}$ and $\widehat{A_2A_1P_9}$ is equal to the area of the triangle $\widehat{QA_1A_2}$. For the point sets that intersect the boundary lines of the triangles, an intersection point with the line is created to be included with the points dataset. Repeat for all sublevels and get all points inside the region needed.

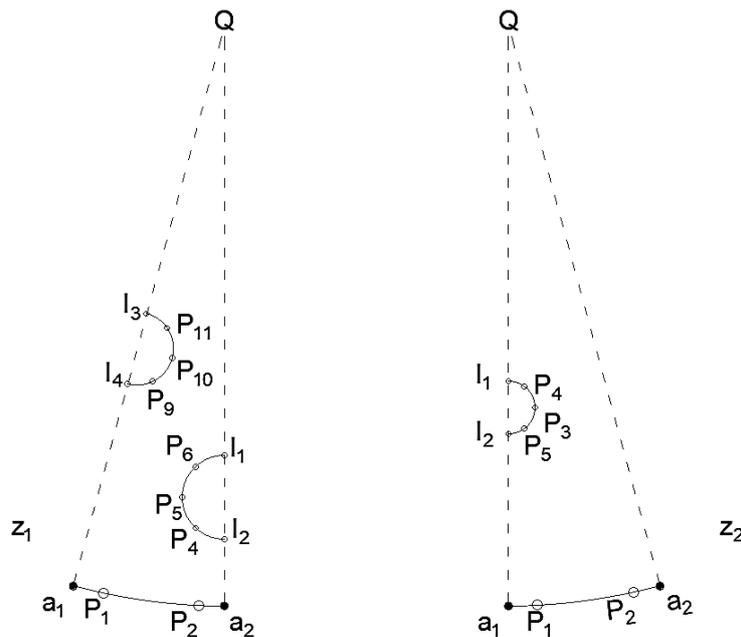


Fig. 11. Triangular slice for each sub-level

In the helix case, where the z level is a function of time and rotation angle, each slice has a shape of a triangle segment with an inner angle corresponding to the resolution angle. These slices form a circle contour in which each slice corresponds to a z -level. Figure 12 is a projected helix-sliced object of one pitch level. Figure 12 shows the triangle slices of a circular cylinder combined into a projection disk, and Fig. 13 shows plotted points of the triangular sliced propeller objects as points before arranging them into a polyline.

Sections A and B showed in Fig. 14 correspond to the intersection of the Z plane with the facet of the STL object. The vertical continuous lines are the line of the facet, and the dashed lines are the boundary of each slice. The points “ P ”s are the intersection with the facets lines and the “ a ” points are the intersection with the boundary of the slice. As in each level, the points needed in the required slice are the intersection with the boundary of the object and the points of the boundary of the slice. In the case where the lines of the facet do not intersect with the boundary of a slice, a point is created to determine that the contour boundary is continuous with the next slice.

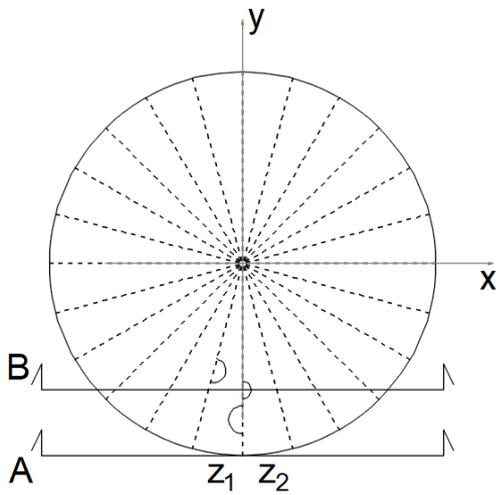


Fig. 12. Projection disk details

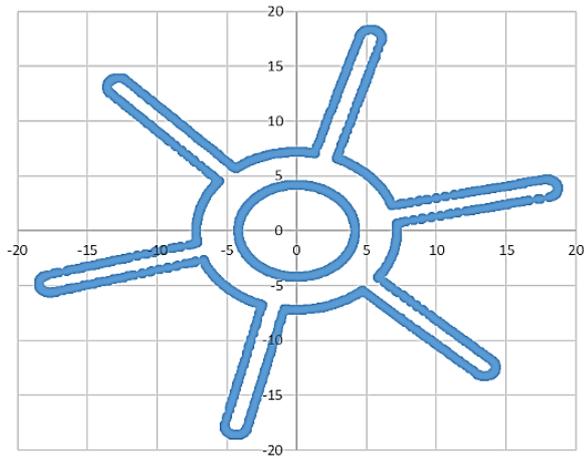


Fig. 13. One pitch data point drawing

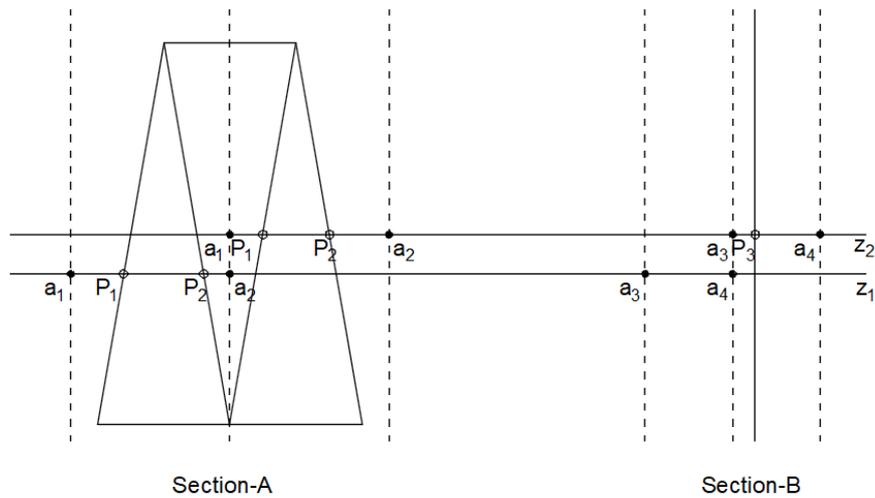


Fig. 14. Intersection for each sub-level

5. CONTOUR GENERATING

After slicing the object and obtaining all points for each slice, they are arranged and combined according to the minimum distances. To obtain this, an initial point “P0” is selected and the distance between this point and another randomly selected point “Pn” is calculated. A loop is done to check the minimum distance with the other points. When the loop calculated all points, it then adds the point with the smallest distance to the contour list and take it as the initial point. Fig. 15 illustrates the propeller and a sliced layer. The red part of the 3D object is conical. This impact can be seen in the layer. The conical slope is transferred into a spiral shape in the slice. Because when layer increases in level, the diameter increase.

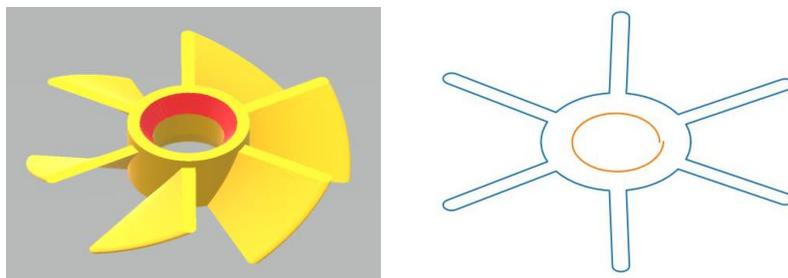


Fig. 15. The contour of the helix sliced object at the start of the inner conic part

Figure 16 shows the 3D plots of two revolutions.

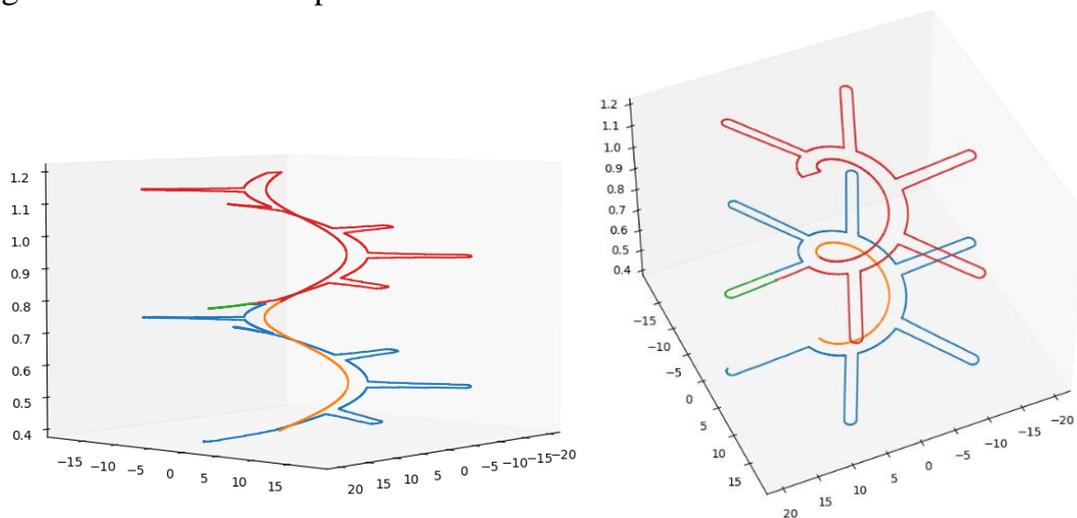


Fig. 16. Propeller helical slice of 0.4 mm and 2 revolution

6. IMAGE FLATTENING

After slicing the model and obtaining a contour, the infill needs to be accounted for using black and white. Black is for printing and the white is to indicate the void. The void is to tell the inkjet not to deposit ink locally. Because the printer moves in a helical shape,

the image boundary is a circle with the radius being the edge of the print. The inkjet head reads cartesian coordinates. In this case, the image shall be flattened from a polar coordinates to an x and y coordinates. Such as when the machine rotates it prints the image having the same resolution. Figure 17 shows the perimeter is the x coordinate of the image and the y coordinate in print radius.

To transform the circular image into a rectangular one, image processing is done in Python to change the pixels coordinates of the image from polar to cartesian. The rest of the image would be empty or white pixels as it indicates to turn off the jets of the print head.

Figure 17 illustrates the order of the pixels to be transferred to flatten for a print image.

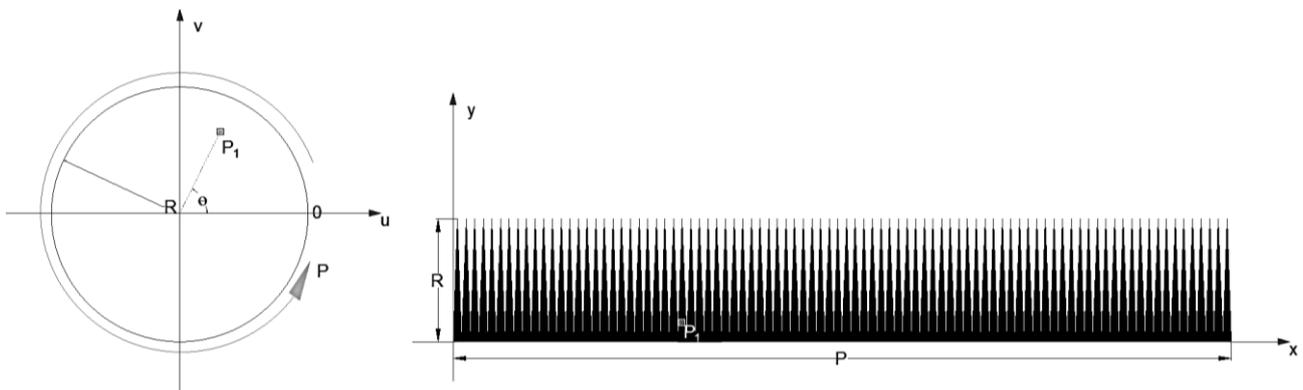


Fig. 17. Coordinates of the original circular image $u-v$ and the $x-y$ for the printer

To show how the calculation of the flattening is realized, Fig. 17 shows the frames of the two images: the sliced one and the flattened one. The real image of the circular disk has a frame “ $u-v$ ”, and its polar (R, θ) from the center of the circle. The flattened image has a plane of an $x-y$ frame from the bottom-left of the image. The relation between these references is polar form the first one and the $x-y$ of the second image as follows:

$$y = R - r$$

$$x = R \cdot \theta$$

where: θ – is the angle from the u -axis as shown in Fig. 17, R – is the max radius of the circle, r – is the radius of the pixel.

For a random pixel of P_1 , the new coordinates are as follows:

$$y = R - \sqrt{u_{p1}^2 + v_{p1}^2}$$

$$x = R \times \tan^{-1} \left(\frac{v_{p1}^2}{u_{p1}^2} \right)$$

To obtain the flattened output, the previous equation is repeated in Python for all pixels in the image and they are combined to get the new image for printing as shown in Fig. 18. Figure 18a shows the original image and 18b the flattened image to be submitted to the printer. Figure 18c is a zoom-on image. The condensed pixels are of the outer radius of the image. The top part indicates the center of the circle which has a lower resolution.

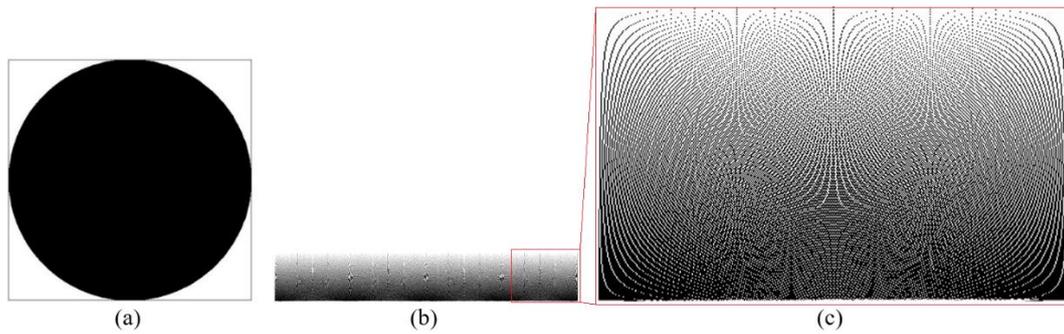


Fig. 18. a) The original circle to be printed, b) flattened image, c) zoom on the pattern

Figure 19 is a slice of the previously shown propeller model. while Fig. 20 is a section with its flattened transformation. This image is a result of multi sections of different levels to form a helical layer. The last step to perform with this image, is to transform it into the x - y coordinates similar to Fig 18. Transforming this image in Python script will lead to the output as shown in Fig. 20, this can be sent to the printer software.

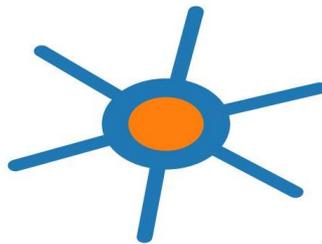


Fig. 19. Propeller disk and its flattened image

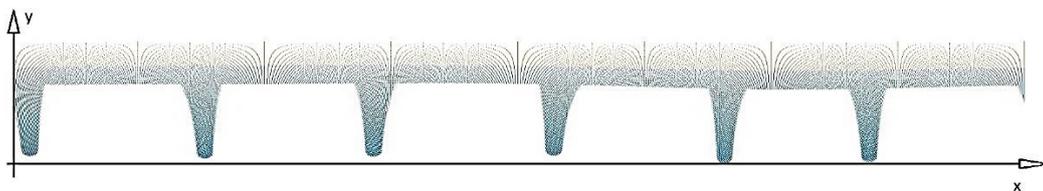


Fig. 20. Flatten transformation of a propeller

7. CONCLUSION

A new slicing method was needed to transform a virtual 3D object into a helical toolpath trajectory. This method has divided the layer into sub-levels, according to the number of slices. This number depends on the print head resolution. Each sub-level is sliced into a triangle shape corresponding to the angular resolution. In the example given for the 3D model propeller, the slice was taken at a level where the inner hole is a conical shape. if the slice was normal 2D the shape would be a circle but in this case, the shape is spiral which proves that the one layer is a combination of different sublevel slices. The novelty of this method is slicing each layer into multiple sections of sub-layers, and recombining these

sections into one layer with different heights. This is to obtain a more precise helical shape with a similar motion to the machine. This method has a massive calculation, so time and memory are needed. To decrease the slicing time, improvements in the algorithm and other strategies could be employed. Furthermore instead of limiting the algorithm to black and white, a gray scale could be used to nuance the deposition.

REFERENCES

- [1] LE NEEL T.A., MOGNOL P., HASCOET J.Y., 2018, *A Review on Additive Manufacturing of Sand Molds by Binder Jetting and Selective Laser Sintering*, Rapid Prototyping Journal, 24/8, Emerald Group Holdings Ltd., 1325–1336, DOI: 10.1108/Rpj-10-2016-0161.
- [2] MIRZABABAEI S., PASEBANI S., 2019, *A Review on Binder Jet Additive Manufacturing of 316l Stainless Steel*, Journal of Manufacturing and Materials Processing, 3/3, Mdpi Multidisciplinary Digital Publishing Institute, DOI: 10.3390/Jmmp3030082.
- [3] HASCOET J.Y., MOGNOL P., Le NEEL T., 2020, *Dispositif Et Procédé Pour La Fabrication Additive*, Institut National De La Propriete Industrielle.
- [4] ADNAN F.A., ROMLAY F.R.M., SHAFIQ M., 2018, *Real-Time Slicing Algorithm for Stereolithography (Stl) Cad Model Applied in Additive Manufacturing Industry*, Iop Conference Series, Materials Science and Engineering, 342/1, DOI: 10.1088/1757-899x/342/1/012016.
- [5] KIRSCHMAN C.F., JARA-ALMONTE C.C., 1992, *A Parallel Slicing Algorithm for Solid Freeform Fabrication Processes*, International Solid Freeform Fabrication Symposium.
- [6] TOPCU O., 2011, *A Method for Slicing Cad Models in Binary Stl Format Deep Learning for Intelligent Manufacturing*, Project Implementation of Metal-Based Microchannel Heat Exchanger in a Micro Refrigeration Cycle, and Numerical and Experimental Investigation of Surface Roughness Effects on Flow Boiling View Project a Method for Slicing Cad Models in Binary Stl Format.
- [7] HU J., 2017, *Study on Stl-Based Slicing Process for 3D Printing*, Solid Freeform Fabrication: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference, Sff, 885–895.
- [8] KING B., RENNIE A., BENNETT G., *An Efficient Triangle Mesh Slicing Algorithm for All Topologies in Additive Manufacturing*, DOI: 10.1007/S00170-020-06396-2/Published.
- [9] ZHANG Z., JOSHI S., 2015, *An Improved Slicing Algorithm with Efficient Contour Construction Using Stl Files*, Int. J. Adv. Manuf. Technol., 5–8, 1347–1362, DOI: 10.1007/S00170-015-7071-9.
- [10] DING D., PAN Z., CUIURI D., LI H., 2014, *A Tool-Path Generation Strategy for Wire and Arc Additive Manufacturing*, Int. J. Adv. Manuf. Technol., 73/1–4, 173–183, DOI: 10.1007/S00170-014-5808-5.
- [11] HASCOET J.-Y., QUERARD V., RAUCH M., 2017, *Interests of 5 Axis Toolpaths Generation for Wire Arc Additive Manufacturing of Aluminum Alloys*, Journal of Machine Engineering, 17/3, 51–65.
- [12] RAUCH M., DORADO J.P., HASCOET J.Y., RUCKERT G., 2021, *A Novel Method for Additive Manufacturing of Complex Shape Curved Parts by Using Variable Height Layers*, Journal of Machine Engineering, 21/3, 80–91, DOI: 10.36897/Jme/138820.
- [13] SHAO M., WEI C., CUI B., LI Y., ZHENG T., 2021, *An Adaptive Slicing Approach for Processing Stl Massive Data Model in Batches Based on Layer Merging*, Journal of Physics: Conference Series, 1884/1. DOI: 10.1088/1742-6596/1884/1/012025.
- [14] LIU R., ZHU H., 2014, *On Helical Projection and Its Application in Screw Modeling*, Adv. Mech. Eng., DOI: 10.1155/2014/901047.