

Quality Implications of Applying 3+2 Axis Toolpaths in 3D Printing

Denys Plakhotnik^{1, 2 a}, Yavuz Murtezaoglu^{1, 2}, Marvin Lachmuth¹, Marc Stautner¹ and Tom Vaneker²

¹ModuleWorks GmbH, Henricistr. 50, 52072, Aachen, Germany

²University of Twente, Drienerlolaan 5, 7522 NB Enschede, the Netherlands

^adenys@moduleworks.com

Keywords: additive manufacturing, 3d printing; FDM; toolpath, 3+2 axis, multi-axis, surface quality

Abstract

In this paper, a 5-axis design of a 3D printer will be presented. The classical serial 3-axis kinematics was upgraded with a rotary-tilt table that introduces two additional rotary axes, and, thus, enabled 5-axis printing. Instead of utilizing the upgraded 3D printer in the continuous 5-axis manner, the authors analyze an intermediate solution – 3+2 axis printing – which can enjoy benefits of both 3-axis and 5-axis printing. In 3+2 printing, material is deposited layer-by-layer, like in 3-axis printing, but not all the layers are oriented in the same direction and may alter their orientation. This approach allows eliminate supports, reduce material usage, and printing time. Furthermore, surface roughness and shape accuracy can be manipulated by changing the orientation of the slicing plane. Analysis of the influence of the slicing plane orientation on the surface quality is the topic of this paper.

Introduction

3d printing is often considered an ultimate technology that can produce any kind of complex geometry. However, this is not true for all processes and all geometries. Additive technologies, especially the layer-based extrusion ones, have several limitations. Most known limitations are the need of support structures and the stair-stepping effect. During extrusion of plastics the plastic itself is in a viscous state. When it is deposited for building a slanted structure, it will have the tendency to flow in the direction of gravity, thus affecting both local surface roughness and geometrical accuracy. To overcome these issues it is generally recommended to add support structures for surfaces that are at an angle of 45 degrees or more. In this case, the CAD/CAM software is to detect such overhangs and plan additional material to be placed beneath. Second, the stair-stepping effect appear due to the fact that each deposited layer has a finite thickness and a certain profile that does not exactly match the shape of fabricated parts. Both issues have severe implications on the surface quality. The support structures are connected to the part material, and they must be removed during postprocessing. The surface areas connected to the support structures are likely to have irregularities like residual material or scarring. The stair-stepping effect causes surface roughness that varies across the part depending on the part's surface curvature and surface normal.

One of the approaches to reduce support structures is to reorient the part as a whole with the goal of minimizing either the amount of support material [1] or surfaces with a critical overhang [2, 3]. Additionally, part geometry can be redesigned to minimize supports [4]. Another approach that aims

at elimination of support structure at all is the use of multi-axis deposition strategies. This can be either fully continuous multi-axis strategies [5, 6] or the 3+2 (indexing) strategy [7, 8]. In the 3+2 strategy, the material is deposited in flat-layers but not all layers are parallel to each other. The build orientation varies from one chunk of layers to another chunk, as shown in Fig. 1. Also Fig. 1 depicts that the stair-stepping effect could significantly be reduced when changing from one build direction to another.

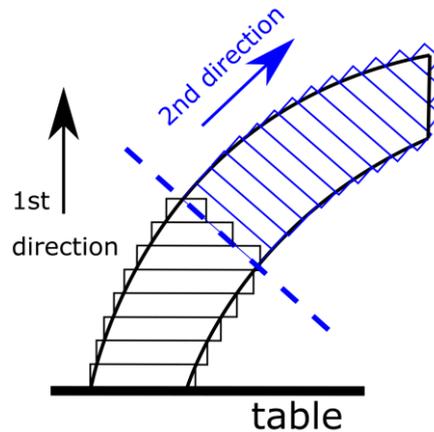


Fig.1 –Principle of the 3+2 printing with 2 build directions

The goal of this paper is to analyze how the surface roughness can be improved by switching from 3d printing to 3+2 printing. It furthermore presents a design of a 3d printer with two additional rotational axes. In order to evaluate the quality implications of 3+2 printing, a novel 3+2 slicing strategy was developed. This strategy decomposes the input geometry into several partitions, while simultaneously avoiding support structures and minimizing the stair-stepping effect. Finally, a quantitative analysis of the stair-stepping effect was performed. It was based on numerical simulation, while future research will compare these results to actual printed specimens.

Algorithmic premises for geometric analysis of meshes

As shown in Fig. 1, 3+2 printing requires decomposition of the part geometry into chunks. In the simplest case, there is one plane that divides the part in two halves. The orientation of that split plane defines the build direction of the second chunk. By varying the position and orientation of the split plane, the second chunk will be sliced differently and therefore optimized to fulfil different criteria.

Assuming that a given part geometry is a triangle mesh (STL file) with n triangles, the algorithm searches for a split plane called E , as shown in Fig. 2 and Fig. 3. The split plane E can be at an arbitrary position and have any orientation, but the first printing direction. The split plane defines two half-spaces (① and ②) located from the opposite sides of it. Half-space ① includes the geometry to be fabricated with the first printing direction, while half-space ② includes the geometry to be fabricated with the second printing direction. The concept of half-spaces is also necessary to analyze mesh triangles for overhang and stair-stepping effect and to identify an optimal position of the split plane.

Fig. 2 and Fig. 3 also depict the first and second half spaces split by plane E and the notation of triangles $T_{1,i}$ and $T_{2,i}$ residing in first and second half spaces.

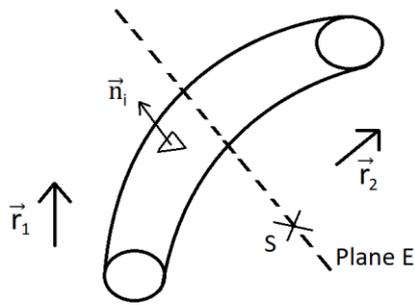


Fig.2 –Definition of the geometries required or used for computation

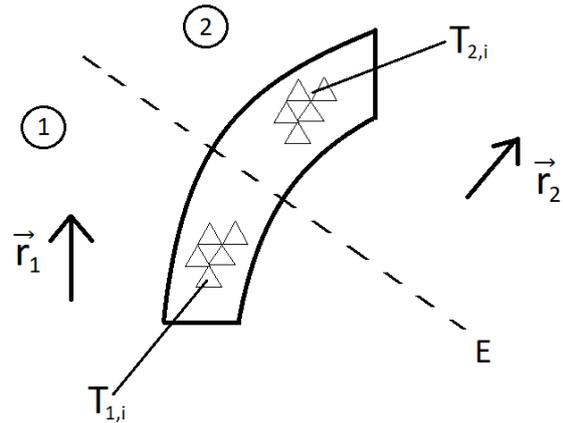


Fig.3 –Definition of triangles after splitting of the space by plane **E**

, where \vec{n}_i : normal vector of triangle i ;

\vec{r}_1 : first printing direction $(0, 0, 1)$;

\vec{r}_2 : second printing direction;

S: point defining the split plane E (plane's normal vector is collinear \vec{r}_2).

Finding intersections between triangles and a plane is computationally expensive. Instead of operating triangles, the algorithm can perform some quick tests on a point cloud. Then, triangle T_i can be represented by a number of points $P_{i,j}$ as shown in Fig. 1.

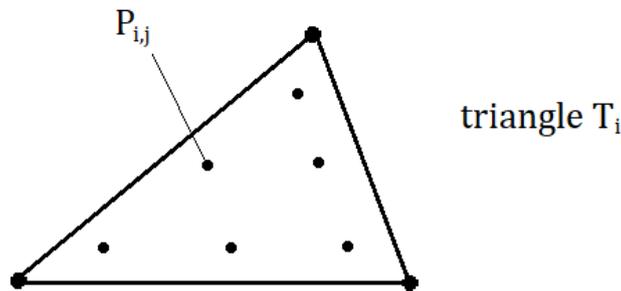


Fig.4 –Points to approximate a triangle

Method - multi-plane decomposition

Let's assume that there is an error function. This function depends on the position of the split planes. The problem to be solved is finding planes $E_{i,j}$ such that the error function is minimized.

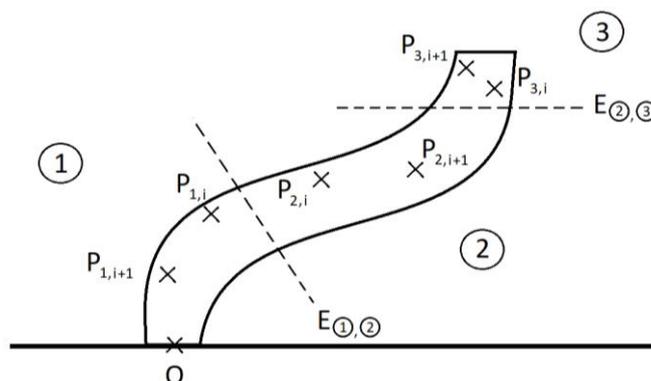


Fig.5 –Notation for multiple planes

Finding the minimum value of the error function for all planes E, E', \dots , is a computationally demanding task. In case of a single split plane E , there are three unknowns. For N subsequent planes, the number of unknowns becomes N times 3. In other words, the error function F , which was initially three-dimensional, becomes N times 3-dimensional. Therefore, the split planes are searched subsequently within a search distance d_s from the previously found plane (or table plane). The search distance d_s can be set to layer thickness multiplied by some factor.

Error function.

Global error functions can be formulated in many ways. Let's assume the global error function F is the total sum of all error functions for all triangles, therefore, a sum of local error functions $F_{m,i}$ in each point $P_{m,i}$ belonging to m^{th} "half-plane".

Best printing results are achieved if printing direction and triangle normal are orthogonal to each other. Deviation from the ideal angle of 90° can be a relevant characteristic for defining the error function. This scheme was considered in [9] to predict scallop height for the stair-stepping effect, as shown in Fig. 6.

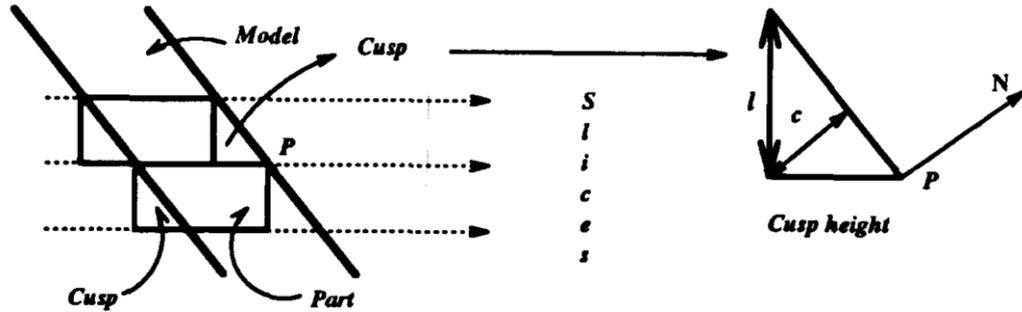


Fig.6 –Staircase effect and cusp height calculation [9]

This allows us to design an error function $F_{1,i}$:

- If $\vec{n}_{1,i}$ and \vec{r}_1 are parallel or almost parallel (within a given threshold), then error function $F_{1,i}$ equals 0.
- If $\vec{n}_{1,i}$ and \vec{r}_1 are orthogonal or almost orthogonal (within a given threshold), then error function $F_{1,i}$ equals 0.
- If neither of the conditions above is met, then there is a weighted error function, given by Eq. 1, depending on the angle between $\vec{n}_{1,i}$ and \vec{r}_1 .

$$\begin{aligned} \alpha_{1,i} < \alpha_{threshold} &\Rightarrow F_{1,i} = \alpha_{1,i}A_{1,i} \\ \alpha_{1,i} > \alpha_{threshold} &\Rightarrow F_{1,i} = (\alpha_{threshold} + K(\alpha_{1,i} - \alpha_{threshold}))A_{1,i} \end{aligned} \quad (1)$$

, where

K is a penalty coefficient.

$A_{1,i}$ is the area of the triangle.

Our global error function F is a function of plane E . The goal, without the boundary conditions, is to minimize the total error, as given in Eq. 2.

$$\text{minimize}(F(E(S, \varphi, \theta)) \quad (2)$$

, where

\mathbf{S} is a position defining plane E position;

φ and θ are the Euler angles defining plane E orientation.

Eq. 2 can be formulated as three-dimensional problem given in Eq. 3.

$$\begin{aligned} & \text{minimize}(F(d, \varphi, \theta) \\ & \text{subject to } \varphi \in [0^\circ, 360^\circ] \mid \theta \in [0^\circ, 180^\circ] \mid d \in [d_{min}, d_{max}] \end{aligned} \quad (3)$$

If there are n points $\mathbf{P}_{i,j}$ evenly distributed over the surface of its triangle, each point can be associated to an equal fraction of the triangle's surface area A'_i , as given in Eq. 4.

$$A'_i = \frac{A_i}{n} \quad (4)$$

Basically, we do not use the triangle mesh but use points $\mathbf{P}_{i,j}$ instead, where each $\mathbf{P}_{i,j}$ has an according normal $\vec{\mathbf{n}}_{i,j}$, which is still the triangle normal coming from $\vec{\mathbf{n}}_i$ of triangle T_i , where $\mathbf{P}_{i,j}$ is derived from. Furthermore, A'_i , which defines zone area for every $\mathbf{P}_{i,j}$, comes from triangle T_i 's area A_i divided by the number of inserted points $\mathbf{P}_{i,j}$. Since mesh triangles vary in their size, it is not consistent to define the number of points being constant for all triangles. Instead of n , a user-provider global value ΔA , which defines the target for area size for inserted points. The actual A'_i is unlikely to meet ΔA , because the areas of triangles tend to vary and n_i , the number of inserted points per triangle, is integer. Thus, the number must be rounded to the nearest integer, but not less than 0, as given in Eq. 5.

$$n_i = \max\left(1, \left\lceil \frac{A_i}{\Delta A} \right\rceil\right) \quad (5)$$

This approximation simplifies the task of calculating the error function F , as in Eq. 6.

$$F_{1,j} = K(\alpha_{1,i} - \alpha_{threshold})A_{1,i}/n_i \quad (6)$$

Algorithm pseudo code. Eventually, the algorithm can be presented by the pseudo code in Fig. 7.

```

Read through the triangle mesh geometry.
Convert triangles into points.
Set first printing direction to z -axis.
For loop using  $\varphi$ 
    For loop using  $\theta$ 
        For loop using d steps
            Calculate error function for  $(\varphi, \theta, d)$ 
Choose  $\varphi, \theta, d$  with minimal error function value, calculate current printing direction
and split plane. Remove all points in the half space below the split plane

```

```

Stop if no points are left.
Set previous printing direction to current printing direction.
Go to Step 4.

```

Fig.7 –Algorithm's pseudo code

Validation

A model of a 3d printer equipped with a rotary-tilting table was developed, as shown in Fig. 8. The printer model can be loaded together with generated toolpaths into ModuleWorks CAM software for visualization and analysis, as shown in Fig. 9. The proof the 3+2 optimization is provided by using geometric results, where split planes and according subsections are highlighted with toggling colors.

For a given part, two parameters varied. The first parameter was the number of desired slices (1,2, 3, 4, and 5); this parameter defined the search distance as part size divided by the number. The second parameter was the angular granularity (1 and 5 degrees) defining the angular steps for iterating φ and θ of the split planes. Smaller angular steps increase the number of iterations and, therefore, increase the computation time. Fig. 9- Fig. 12 show several decomposition results.

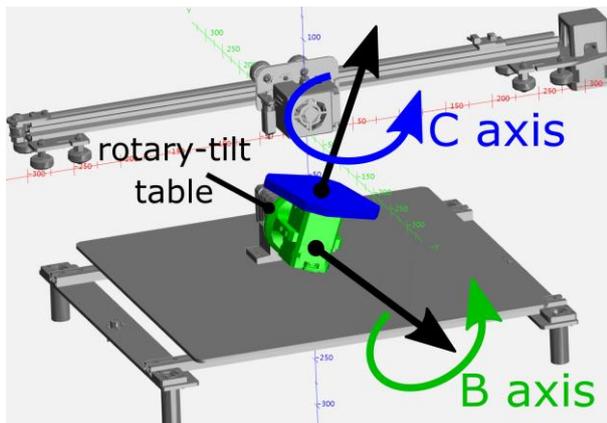


Fig.8 –Kinematics of rotary axes

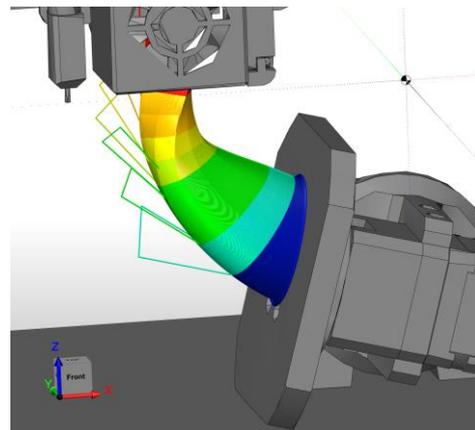
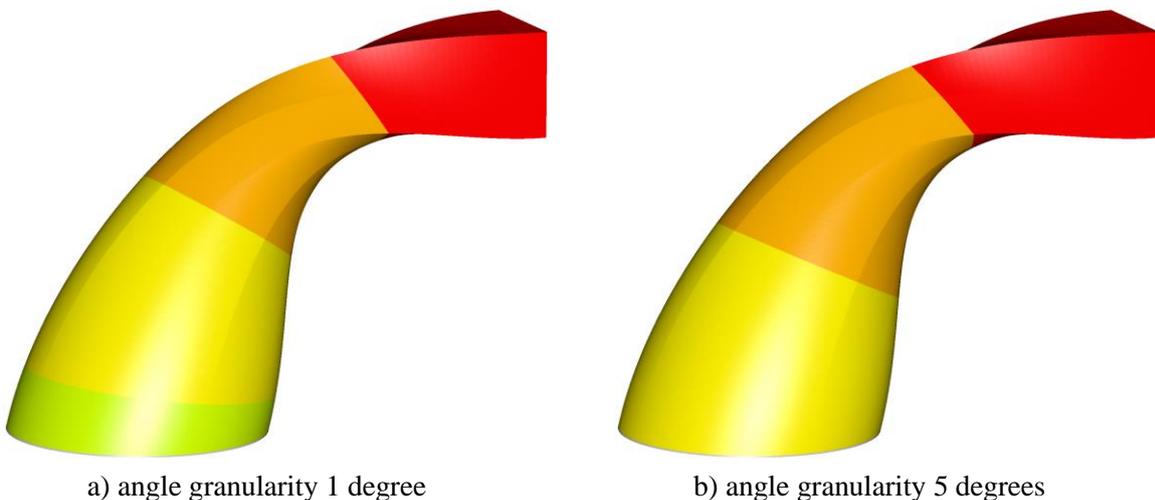


Fig.9 –Backplot of 5-axis toolpath in ModuleWorks CAM software

It is noticeable that the smaller granularity (1 degree) resulted in bigger number of subsections. Also, the difference to the 5 degrees granularity decomposition can be spotted as a relatively small chunk attached to the table.



a) angle granularity 1 degree

b) angle granularity 5 degrees

Fig.10 –Decomposition with Desired Slices = 1

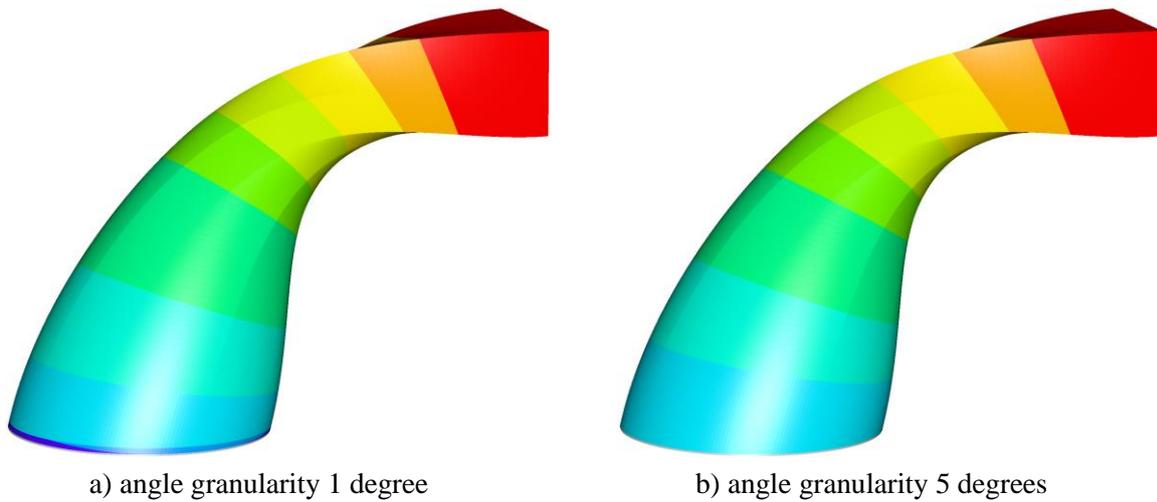


Fig.11 –Decomposition with Desired Slices = 3

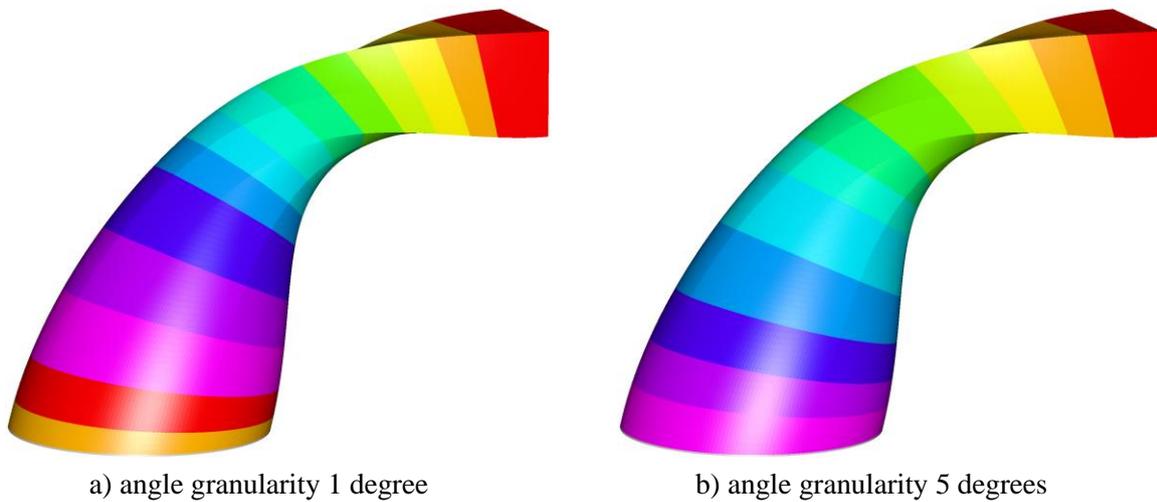


Fig.12 –Decomposition with Desired Slices = 5

Besides the global error function can be evaluated. The global error function sums up error functions of individual slices. Fig. 13 shows two graphs presenting how the global error function changes depending on the number of desired slices and, therefore, on the search distance, which decreases, whilst the number of desired slices increases. The general trend shows that the global error function tends to decrease with the decreased search distance. This can be explained by applying smaller subsections that are better aligned to the surface curvature to produce a smaller error. However, the function appears to be not always downwards monotone. There are jumps back at the right side of the charts. It seems that the reverse direction in the trend was due to discrete nature of the surface representation. The algorithm iterates over non-smooth triangulated surfaces with discrete steps, which can almost surely never meet an optimum. Thus, these upwards variations in graphs (less than 1%) can be attributed to the numerical issues during computations.

Understanding of the global error function is not intuitive. The value of the global error function represents some metrics that cannot be directly mapped on the existing measures describing surface quality. The scallop height depends on the layer thickness and the orientation of the surface relative to the build direction. For the sake of simplicity, layer thickness can be omitted, and the stair stepping cusp height can be represented as a fraction of the layer height. Fig. 14- Fig. 18 show the distribution of different cusp heights changes with the increase of the number of partitions of the test part. All

charts share the “red” histogram of the cusp distribution of the test parts without any subdivision. Then, the number of partitions increases from four to 32. The results for partition numbers above 16 are not shown because of marginal differences. In Fig. 19, 20 baskets of cusp heights with the step of 5% of the layer thickness are depicted with black bars. In the consequent illustrations (Fig. 14- Fig. 18), grey bars depicting a histogram of the previous partitioning have been added in order to improve comprehension of the presented information. As it can be seen, applying of the decomposition algorithm progressively decrease high magnitude scallops. Four partitions have been enough to eliminate cusps above 65% of the layer thickness. Increasing the number of partitions to 32 helped to push this value down to 40% of the layer thickness. Four partitions led to a distribution skewed to the right side, while the larger numbers of partitions showed no clear trend. Besides, the average cusp height decreased progressively outwards the value of 0.377 for the non-partitioned one, with a local bump at 26 partitions, as shown in Fig. 19.

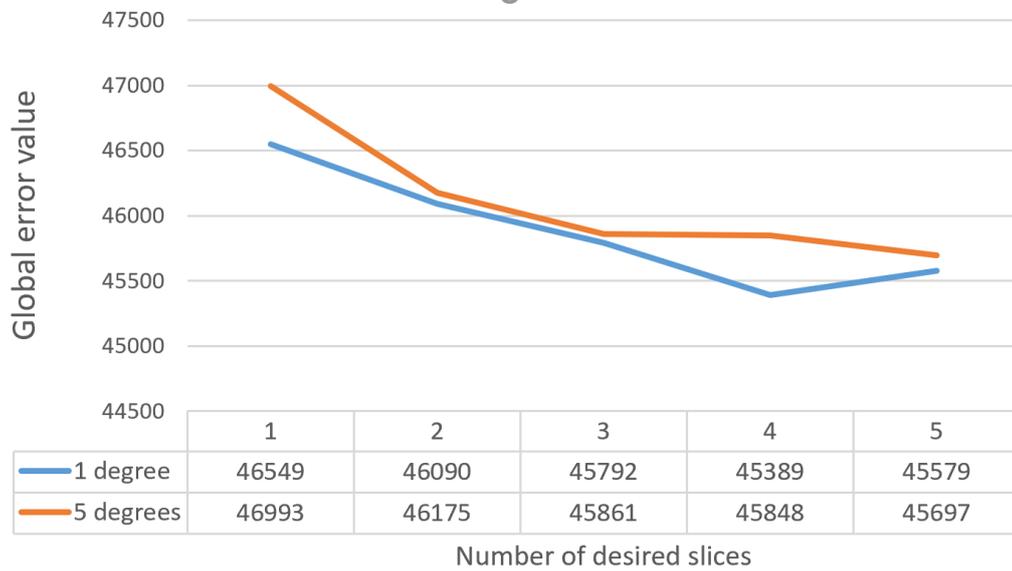


Fig.13 –Global error function for different trials

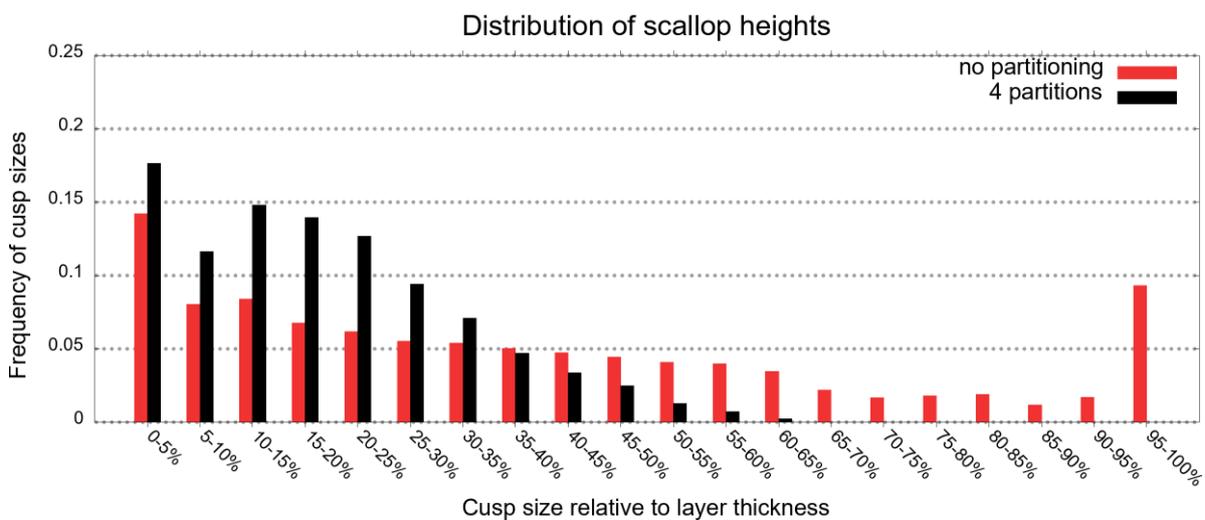


Fig.14 –Distribution of cusp sizes for the part consisting of 4 partitions

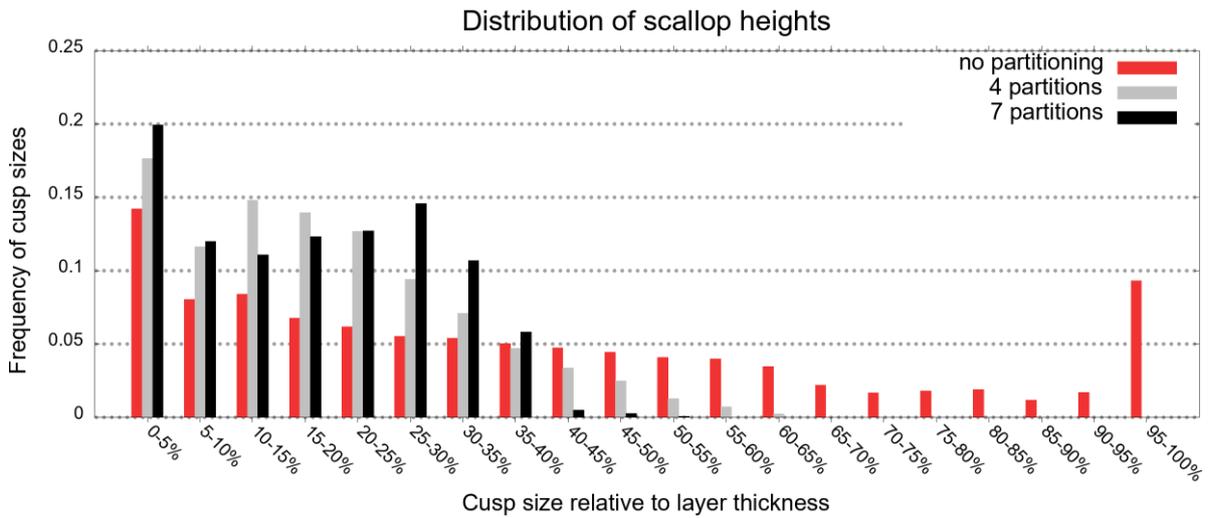


Fig.15 –Distribution of cusp sizes for the part consisting of 7 partitions

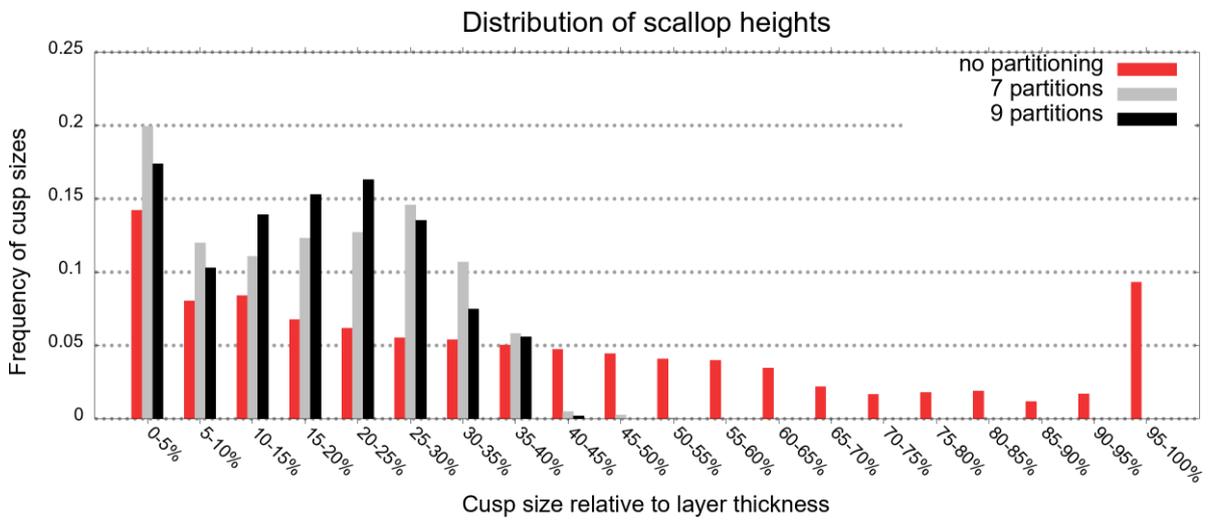


Fig.16 –Distribution of cusp sizes for the part consisting of 9 partitions

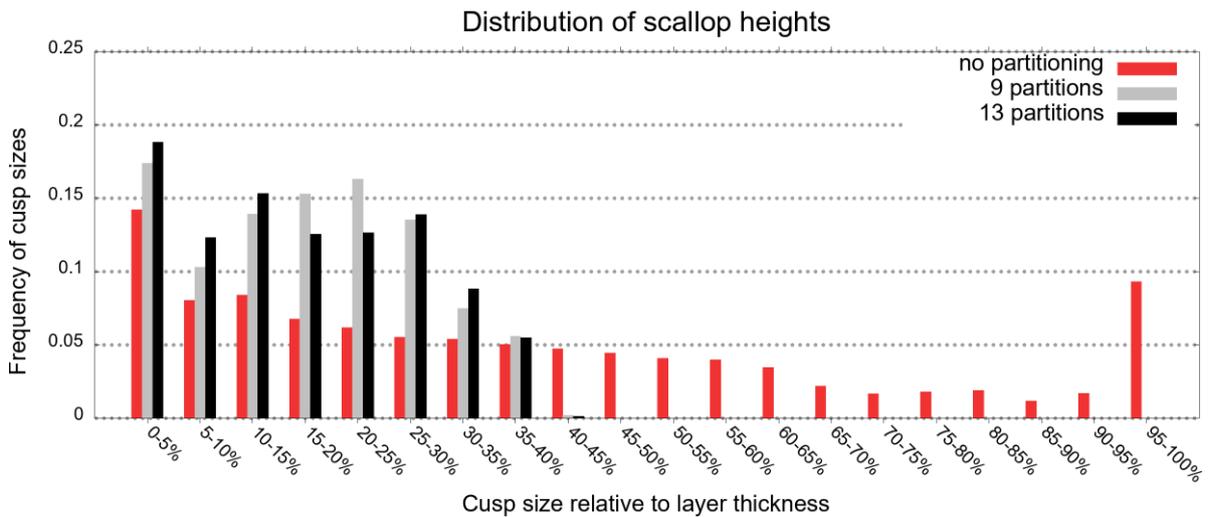


Fig.17 –Distribution of cusp sizes for the part consisting of 13 partitions

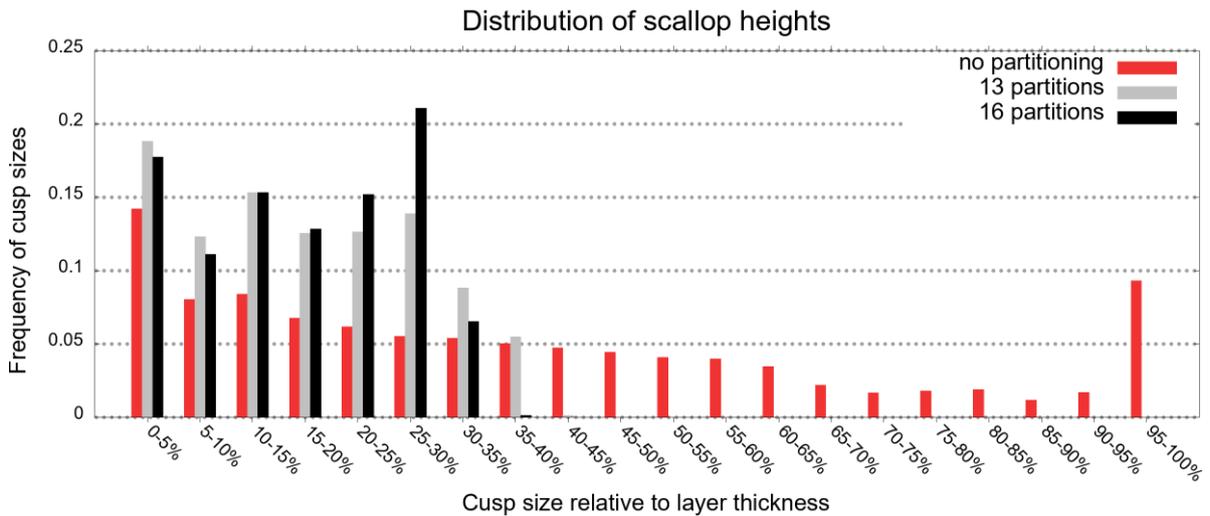


Fig.18 –Distribution of cusp sizes for the part consisting of 16 partitions

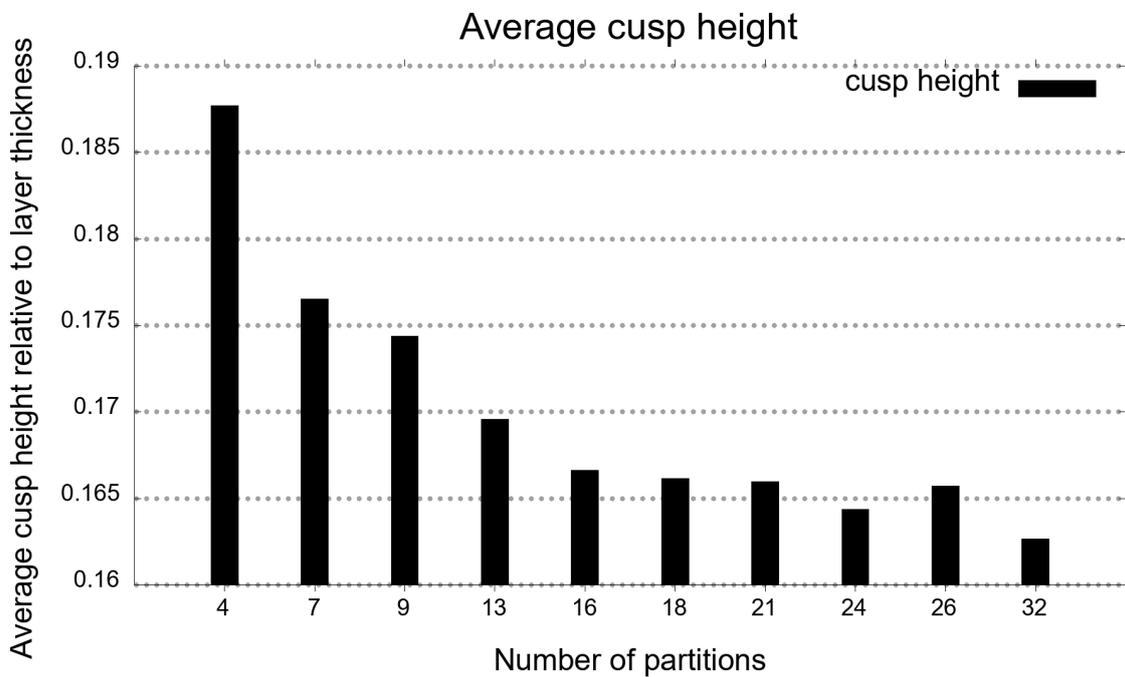


Fig.19 –Average cusp height (0.377 for the non-partitioned one)

Conclusions

Multi-axis 3d printing has been proved to improve cusp height. In the presented use case, 16 build directions was sufficient to substantially reduce the average cusp height along with elimination of extreme values (>50% of the layer thickness) with already 7 build directions. Further increase of the number of partitions lowers the average cusp height but some decompositions can also be against the trend. Ultimately, distribution of the cusp sizes results in cusp values skewed to smaller values, whilst the most extreme cusp heights are completely eliminated. The algorithm's performance is efficient enough to run the test cases within a few minutes on an ordinary PC. The future work will include measurement of the surface and comparison with the computed values.

References

- [1] Ezair, B., Massarwi, F. and Elber, G. Orientation analysis of 3D objects toward minimal support volume in 3D-printing. *Computers & Graphics*. 51, (2015), 117–124.
- [2] Zwier, M. P. and Wits, W. W. Design for Additive Manufacturing: Automated Build Orientation Selection and Optimization. *Procedia CIRP*. 55, (2016), 128–133.
- [3] Wang, C. and Qian, X. Simultaneous optimization of build orientation and topology for additive manufacturing. *Additive Manufacturing*. 34, (2020), 101246.
- [4] Langelaar M. Combined optimization of part topology, support structure layout and build orientation for additive manufacturing. *Structural and Multidisciplinary Optimization*. 57(5), (2018), 1985-2004.
- [5] Dai, C., Wang, C. C. L., Wu, C., Lefebvre, S., Fang, G. and Liu, Y.-J. Support-free volume printing by multi-axis motion. *ACM Transactions on Graphics*. 37, 4 (2018), 1–14.
- [6] Coupek, D., Friedrich, J., Battran, D. and Riedel, O. Reduction of Support Structures and Building Time by Optimized Path Planning Algorithms in Multi-axis Additive Manufacturing. *Procedia CIRP*. 67, (2018), 221–226.
- [7] Murtezaoglu, Y., Plakhotnik, D., Stautner, M., Vaneker, T. and van Houten, F. J. A. M. Geometry-Based Process Planning for Multi-Axis Support-Free Additive Manufacturing. *Procedia CIRP*. 78, (2018), 73–78.
- [8] Wulle, F., Coupek, D., Schäffner, F., Verl, A., Oberhofer, F. and Maier, T. Workpiece and Machine Design in Additive Manufacturing for Multi-Axis Fused Deposition Modeling. *Procedia CIRP*. 60, (2017), 229–234.
- [9] Dolenc, A. and Mäkelä, I. Slicing procedures for layered manufacturing techniques. *Computer-Aided Design*. 26, 2 (1994), 119–126.