Improvements to the tree support software tool

This document presents the improvements and changes to the tree support generating software tool since it´s presentation in June 2023 at the Lasers in Manufacturing Conference (LiM) in Munich/Germany. For further information the reader is kindly referred to the paper "Development of a tree-support software module for PBF-LB/M" which can be found in the proceedings of the conference under the following weblink:

https://www.wlt.de/2023/proceedings-zur-lasers-manufacturing-konferenz-lim-2023-jetzt-online

The novel software module designed to generate tree-based support structures for controlled additive manufacturing processes, specifically powder bed fusions of metals. The primary objective is to reduce material consumption, energy costs, and printing times while ensuring efficient support for complex components, heat dissipation, stress absorption, and prevention of geometric overhangs and defective production.

The software module was developed using Python, with libraries such as numpy, trimesh, pythonocc, and OpenCascade Technology modeling kernel.

The development process began by investigating different support structures and evaluating two well-known botanical algorithms: L-Systems and Space Colonization. However, both algorithms were found to have weaknesses; Space Colonization produced branches that required excessive material, while L-Systems lacked control over the distribution of contact points with the component surface.

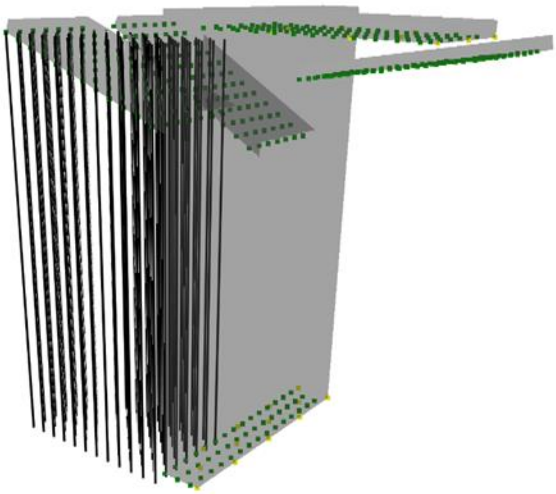
To overcome these issues, a new alternative algorithm was developed that builds a tree-shaped support structure from contact points on the component surface to the trunk. The approach is advantageous as it places the "trunk" in the middle under the treetop without requiring calculation of the starting point in advance. This approach results in short processing time of under 2 minutes for most components.

The software developed distinguishes between the calculation of points (contact points, branch points, intersection points with the geometry) and the generation of geometry for the branches and trunks. The points are calculated based on tessellated data, and the geometry based on the points is generated as a solid body.
In the first step, the STEP file to be processed is tessellated to generate the triangles required for the following calculations. The alignment of the surface normals of the triangles is then checked and corrected if necessary to ensure that the normals are aligned uniformly. In the next step, the contact points of the branches on the component surface are generated.
Two methods were employed to determine the component surfaces requiring support: triangles described in STL format and ray tracing. The first method involves comparing the angle between the surface normal of each triangle and the Z-axis with the maximum permissible overhang angle. Triangles where the surface normal exceeds this angle are considered as surfaces needing support.

The second method, ray tracing, calculates the angle at which a ray hits the component while generating evenly distributed contact points on the component surface for tree structure calculation. The advantage of using ray tracing is that it can be hardware-accelerated by graphics cards, significantly speeding up the calculation process. However, this method's efficiency depends on the graphics card used, and installing the required libraries on various computers failed.

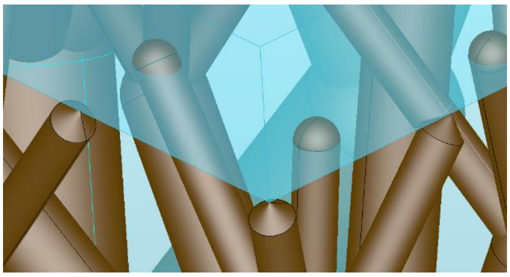
Consequently, the calculation of overhangs was implemented based on the surface normals of the triangles in the tessellated model.

To generate the contact points (leaves) on the component surface a non-hardware-accelerated ray tracing method was used. An evenly distributed grid of points is projected to the component surfaces. The points are calculated separately for each partial surface of the component.



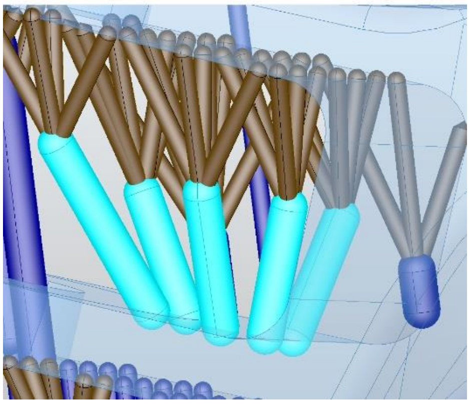
The software module also detects and considers overhang edges and individual points requiring support structures during the manufacturing process. To distinguish between edges of triangles due to the component curvature and "real" component edges, a simple filter function was used. This function compares the angle between two neighboring triangles provided by trimesh library; any edge with an angle greater than 20° between the triangular surfaces is treated as a component edge. The next step examines whether the edge is concave or convex; concave edges are not considered. In the final step, the angle of the edge to the Z-axis is determined. If the angle exceeds the OVERHANG_ANGLE_EDGE, support points are created on the edge at a distance of GRID_STEP. These additional points are assigned to the closest partial surface. Each of these partial point clouds is divided into clusters with the maximum crown diameter. A separate tree is calculated for each of these clusters.

The algorithm described in the LIM paper is used to calculate the branch points. After calculating the branch point, it is checked whether this point is inside or outside the component. If the point is inside the component (see picture below), the intersection point of each branch with the component is calculated and the branch ends with a hemisphere at the intersection point with the component.



When generating the geometry for the first level of branches, a distinction is made between the points on the surface and the additional points on edges and points. For the points on the surface, a hemisphere is generated at the end of the branch to fill any gaps between the branch and the surface. For the additional points, a cone is generated to ensure better removability after printing.

If the length of the branches/trunks exceeds a predefined length, the radius is automatically increased. For trunks whose length exceeds a second limit, the radius is further increased, but the



trunk is not created as a cylinder, but as an elongated cone, as shown in yellow in the last picture of this document, to further increase stability. These limits are part of the manufacturing restrictions and were determined in a parameter study by iLAS.

The behavior of the software module can be controlled by nearly 30 parameters which are stored in a config file. Some of these parameters are mentioned earlier in the text
After the calculations have been completed, the support structures can be exported in STL and/or STEP format, either with or without a component. A unique selling point compared to other support generators is the output of solid bodies in STEP format, which ensures easy further processing of the support structures in CAD/CAM programs.

The software module has proven to be equivalent or more powerful than commercial solutions available during the project. Its main advantage lies in the use of solid bodies and the export of tree structures in STEP format, making them easily accessible for subsequent processes. The software's efficiency, adaptability, and compatibility with various CAD systems make it a valuable tool for optimizing support structures in controlled additive manufacturing processes.