Lasers in Manufacturing Conference 2023

Development of a tree-support software module for PBF-LB/M

Jan Hünting[a], José Manuel Crego Lozares[a], Maria Isabelle Maiwald[a], Jochen Michael[b], Claus Emmelmann[a], Tim Röver[a]

[a] *Hamburg University of Technology, Institute of Laser and System Technologies, Harburger Schloßstr. 28, Hamburg 21079, Germany*
[b] *Cenit AG, Industriestr. 52-54, 70565 Stuttgart, Germany*

Abstract

By combining simulation and generative design, a software tool was developed that generates individual support structures in the form of trees. The objective is that even users with little experience can manufacture components with efficient support structures using PBF-LB/M. The key element of the method is the use of tree structures whose shape is generated by algorithmic, biological growth using botanical methods. Topology optimization, which has already proven itself in the design of optimal support structures, was used to optimize the support structures of three benchmark parts. The generated trees were then compared to commercially available tree-like supports and block supports using Finite element analysis to simulate the PBF-LB/M process. The simulation also included a stress relief heat treatment and the removal of the part from build plate and supports. Based on the knowledge gained from the numerical assessment of the generated tree supports, the created tool allows the user to generate highly material-usage efficient supports with block support-comparable capabilities without requiring any previous experience of the subject.

Keywords: Support structure; Additive manufacturing; Waste reduction; Generative design; Tree support; PBF-LB/M

1. Introduction

Over the past few years, additive manufacturing (AM) has become one of the leading options for manufacturing. Contrary to conventional subtractive methods, AM relies on layer-by-layer production [1] which allows to produce geometrically complex designs [2]. Among the different AM technologies available, powder bed-based production of metallic components by laser powder bed fusion of metals (PBF-LB/M) [3] has taken over most of the research community's focus [4]. The combination of complex designs with excellent material properties achieved through PBF-LB/M surpasses conventional manufacturing processes and has proven to be successful in fields of application such as aerospace [5], medical [6], and energy [7] sectors.

The manufacturing process of components using PBF-LB/M begins with the application of a powder layer of metallic material onto the building platform, which is melted with the help of a laser. After the material has

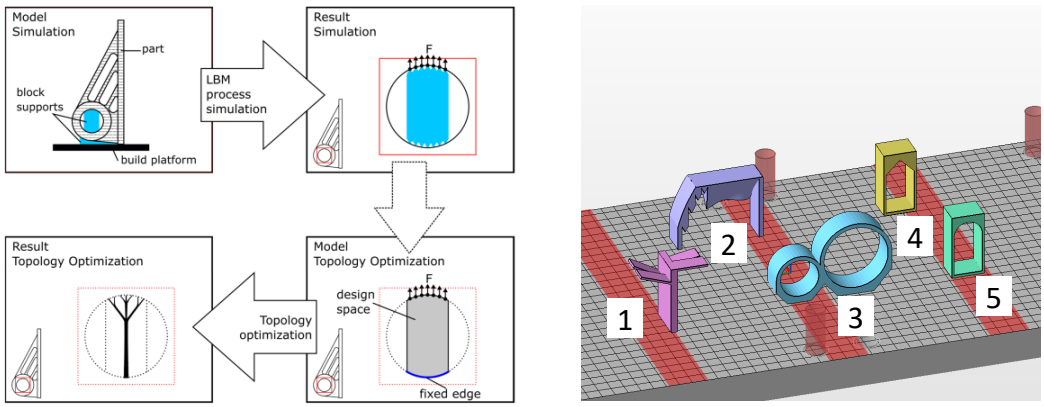

Fig. 1. (a) Schematic of generation of tree-like topology optimized support structures by 2D topology optimization reprinted from [8]*; (b) screenshot of benchmark parts 1, 2, 3, 4 and 5 on build platform reprinted from* [9].

solidified, the build platform is lowered, and the process is repeated layer by layer until the component is built. However, contractions of the molten material cause residual stresses that can lead to cracking, recoater interference, and component failure due to layer separation from the build platform [10].

In order to obtain the parts, elements known as support structures are often needed to act as fixed structures for the already solidified regions such as overhanging structures to support them and remain balanced throughout the print job. In addition to these two functions, support structures help avoid residual stresses and distortions by redistributing and dissipating the heat that is generated during the process. As a result, these elements play a key role when manufacturing components using PBF-LB/M technology [11].

Nonetheless, the printing of the support structures causes additional material and energy consumption as well as extended printing times which ultimately leads to increased overall costs [11]. Nowadays designs for support structures are not optimized for the part to be printed. In the individual and small series production of components, this leads either to oversizing of the support structures or to misprints. Among the main tools used to overcome this problem, Topology Optimization (TO) has become one of the most interesting approaches due to its high material exploitability and low distortions that can be achieved [12].

Specifically, tree support structures are highly resource-efficient and can be described using approaches from algorithmic botany. Algorithmic botany has developed methods to describe biological growth through algorithms. Preliminary work has shown that the optimal structures have a tree-like geometry [13]. As seen in Fig. 1 (a), Bartsch developed an algorithm-based TO for part-specific support design aiming to advance today's additive manufacturing toward first-time-right production. The resulting 2D TO validated for a series of benchmark parts shown in Fig. 1 (b), with Ti6Al4V as the use-case material, demonstrated that the support structures can be tailored for the parts without the need of user experience [14].

To our knowledge, few attempts on 3D tree-like support structures optimized for the PBF-LB/M process have been reported in literature. This study presents a new software tool that can generate 3D tree support structures inspired by TO and presents its capabilities by comparing it to commercially available supports. Residual stresses, geometrical distortion, and materials usage were the main parameters that were evaluated.

2. Related work: *2D topology optimization as starting point for the tree generation*

In the thesis of Bartsch it was investigated if the digitalization of support design and removal in PBF-LB/M processes has the potential to decrease support costs and lead to a first-time-right production [14]. Therefore,

an automated, load-based approach for support design was developed. The base of this approach were the results of topology optimizations with realistic load cases, received from process simulations, as inputs.

The material used in the thesis was the titanium alloy Ti-6Al-4V because of its popularity in PBF-LB/M processes and its high melting point leading to a high sensitivity regarding residual stresses. A comprehensive material model, including thermo-physical, optical and mechanical properties, was derived for use in further calculations and experiments.

Systematic mechanical TO studies have been carried out to elaborate design rules for the creation of 2D tree supports. The goal was to minimize the support costs, mainly by reducing the support volume, while maintaining a predefined structural integrity. For easier support removal and reduced computational effort, the topology optimizations were accomplished in two dimensions only. The contours of the 2D trees were determined, the topologies were drawn in manually and the evaluation parameters were measured. The final parameters can be seen in Fig. 4 (a).

In the next step the Space Colonization algorithm was chosen for the support design and was implemented using the Grasshopper plugin of the Rhino 6 CAD software. A rectangular grid of points was created and projected on the part geometry. After that, all the points are deleted which are located outside of the design domain or on surfaces which do not violate the angle restriction. The stresses at these support-part-interface points (anchor points) were interpolated from the simulation results and the trunk was created in the centre of loads of the tree. Afterwards, the Space Colonization algorithm was applied to obtain the topology of the tree. The forces which have to be compensated at all anchor points were calculated using the yield strength of the material and the stresses which were obtained from the process simulation. In a last step these forces were used to calculate the widths of all branches which were then used to finally model the whole 3D topology of the tree.

Then a benchmark test consisting of five different parts was developed to evaluate the quality of the developed methodology for generation of 2D tree supports concerning economical and technical evaluations. Each of the parts exhibit at least one type of common geometrical feature, including curved and straight edges and surfaces as well as material accumulations and free bar elements to provoke overheating and distortion, respectively. The benchmark test was executed, comparing cone, block and 2D tree supports in two different hatch distances. It was shown that both variants of the newly developed 2D tree supports required significantly less support volume than the cone and block supports, while being nearly as good in dissipating heat as the cone supports and considerably better than the block supports. The economical benchmark test showed, that the 2D tree supports exhibit low manufacturing costs because of their reduced support volumes but mandatory additional steps in support design increase the design costs. The author stated that increasing the lot sizes would lead to significantly reduced costs in 2D tree supports compared to both standard supports.

3. Method

The first part of this section gives an overview on the development of the tree generating code, stating the used software and libraries as well as mentioning the different approaches of botanical algorithms that were tested. The second part focuses on the validation of the programmed code using a Finite element analysis including a stress relieving heat treatment to compare the generated tree supports with two commercially available support structures.

3.1. Development of the tree generation code

The intention of the development was to use freely available libraries to avoid license conflicts. Therefore, development of the tree generation tool was done with Python [15], mainly using numpy [16], trimesh [17]

and pythonocc [18]. The trimesh library was used to analyse the mesh representation of the part. The data structures of trimesh are basing on numpy, so it was used for most of the calculations. The geometry and mesh creation as well as the STEP export use pythonocc, based on OpenCascade [19] Technology modeling kernel.

For the development of a 3D tree generation tool, the algorithm used to develop the trees is essential. Therefore, two well-known botanical algorithms, L-Systems [20] and Space Colonization [21], were analysed. Both algorithms try to describe the biological growth of trees. The L-Systems algorithm is starting at a given point and generates branches according to specific rules, until a predefined boundary is reached. The Space Colonization algorithm also starts at a given point and tries to reach predefined endpoints where the growth of a branch stops. These behaviours and the requirements led to the disqualification of both algorithms.

For both algorithms, the starting point must be calculated before the tree generation starts, so the complete part has to be analysed at the beginning. The L-Systems algorithm tries to reach a boundary surface and generates more or less random contact points with this surface. So, the position and the distance between theses points cannot be predicted. The Space Colonization algorithm increases the thickness of the already created branches with each iteration. At the end, often there are some points which are difficult to reach, but the algorithm tries to reach them nevertheless by increasing the diameter of the branches. In the end this leads to a massive clump instead of a tree. A “tree” generated with Space Colonization in a preliminary study is shown in Fig. 2 (a). The green points are reached, and the red points are unreached after 100 iterations. Therefore, L-Systems and Space Colonization were rejected as solutions for this study.

A simpler algorithm was investigated instead. The input for the algorithm is a point cloud with uniformly distributed points on the surface of the part. Starting with the first point, a search for the four closest neighbours with a k-d tree [22] is carried out. The centre point of the polygon, defined by the five points, is moved downward below the lowest of the five points. Then the angles between the Z-axis and the connection line from the lowered centre point to the polygon points are checked with the predefined support angle value. If the connection line angles are higher than the support angle value, the centre point is moved downward until the angles fall below the support angle value. The connection lines between the lowered centre point and the polygon points form the first branches. The iteration is started again with the next five points until all points on the surface are consumed. Then the iteration starts a new layer of branches with the lowered centre points until only one point remains. This point is used to build the trunk. With each new layer of branches the radius of the branches is increased.

The result of top-down tree algorithm is shown in Fig. 2 (b). A short, animated video of the simulation can be found in Appendix A.1. The advantages of this algorithm are that all points are consumed, the trunk is close to the centre of the surface and the angle of the branches can be controlled.

After the successful development and implementation of a preliminary version of the tree generation code, the implementation of additional features, such as detection of overhanging surfaces, edges and points as well as the integration of process simulation results, a STEP export function and many others, began.

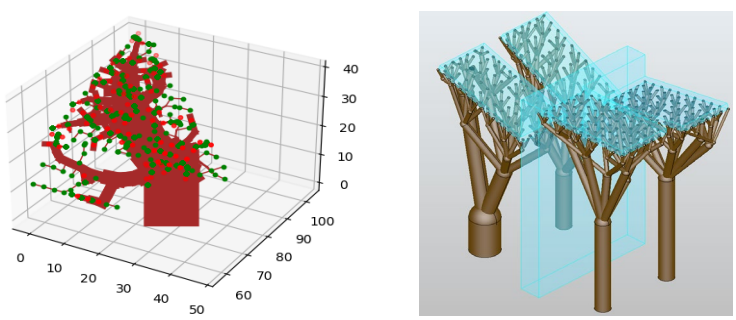

Fig. 2. (a) Space Colonization “tree”; (b) top-down tree.

3.2. Validation using an AM simulation tool

For validation of the generated tree supports, PBF-LB/M with stress relief heat treatment Finite element analysis was performed on three out of five of the benchmark parts suggested by Bartsch [14] (Parts 1, 3 and 5) using Netfabb Ultimate Simulation Local 2023 software and comparing the results with Block Supports (BS) and commercially available tree supports from Netfabb (NFS).

The BS and NFS cases, shown in Fig. 3 (a) and (b) respectively, were created in Autodesk Netfabb Ultimate 2023 and then simulated in the Finite Element Analysis (FEA) module. The volume occupied by the block supports was 12 %. The NFS were obtained imposing an overhang angle of the 35° for all surfaces that required support.

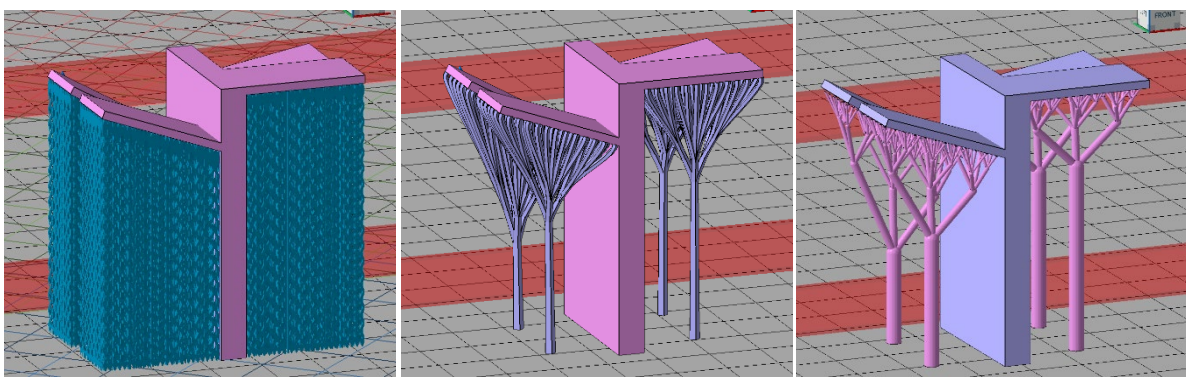

Fig. 3. Supports on Part 1; (a) Block Support (BS); (b) Netfabb Tree support (NFS); (c) BEST Support (BESTS).

The benchmark parts evaluated with the BEST support (BESTS), shown in Fig. 3 (c), were loaded into the FEA module differentiating between part and support.

The building platform selected for the simulation was that of a SLM500 machine with four lasers and the material properties for Ti6Al4V were obtained from Netfabb Ultimate Simulation Local library. Laser parameters for the simulation were chosen as follows: Laser power was fixed to 240 W with a beam diameter D = 0.15 and heat absorption efficiency of 40 %. Travel speed was set to 1200 mm/s with a layer thickness of t = 0.06 and a hatch spacing of h = 0.1 mm.

The element size of the finite element meshes were chosen to be 0.3 mm with a padding tolerance of 0.01 mm. This mesh configuration allowed to obtain good sensitivity of results with an average simulation time of 45 min associated to each simulation. For all simulations, the support-structure-failure-criteria parameter was activated and fixed to a value of 1050 MPa. After the PBF-LB/M process simulation, the simulation continued with a standard stress-relief heat treatment for Ti6Al4V at T = 700 °C for t = 2 hours. Following the heat treatment, the build plate and supports removal were simulated for each part in this order.

4. Results

The design parameters of the developed code are presented and explained in the following section. It is shown which values were modified. Afterwards the implementation and functionality of the tool to generate the tree support structures is expounded. The last subsection discusses the assessment of the obtained results from the simulations regarding the number of failed elements, the maximum displacements and the material usage of all three tested support structures.

4.1. Final design parameters for tree supports

Bartsch found several parameters to evaluate the two-dimensional support trees, which were created in her thesis [14] and are shown in Fig. 4 (a).

The three-dimensional tree supports developed in this work can be adjusted by manipulating up to ten parameters of the developed code. Fig. 4 (b) shows all relevant parameters. Some of these parameters stayed unvaried, like the maximum overhang angle of each branch with respect to the Z-axis (θ = 54°) and the initial radius of the support branches at the anchor points (r_o = 0.2 mm).

The parameters used to modify the topology of the supports are the number of subbranches *K*, the grid step *G* and the radius factor r_f. By changing *K*, the number of subbranches that combine into a branch or trunk is chosen. An increased *K*-value leads to less layers of branches. *K*-values greater than 5 will result in the excessive elongation of first-level branches, thereby increasing the height of the tree and substantially rise the risk of generating a trunk below the build platform. Because of this and the fact that high values of *K* also result in unwanted thick trunks, the optimal *K*-value was found to be 5. For the surfaces which require support, the grid step *G* defines the distance between the anchor points generated from the code. Higher values of *G* mostly decrease the used support volume due to less branches being generated. To avoid branches and trunks with unnecessarily high radii and to add another possibility to control the support structure volume, a radius factor r_f is implemented. It can be used to modify the thickness of all branches and trunks besides the first layer of branches which is in direct contact with the part. The results presented in this publication were obtained using *G* = 2 mm and r_f = 0.85 as well as the already mentioned values for *K*, θ and r_o. The mesh size and the grid step have an additional impact on the accuracy and the computational time of the code.

It should be mentioned that none of the parameters implemented in the code has a direct matching counterpart from Bartsch´s 2D evaluation parameters. Nevertheless, some of these parameters like the crown width, the crown height, the branch width and the branch length can be indirectly influenced by adjusting the values of *G*, K, r_f, r_o and θ.

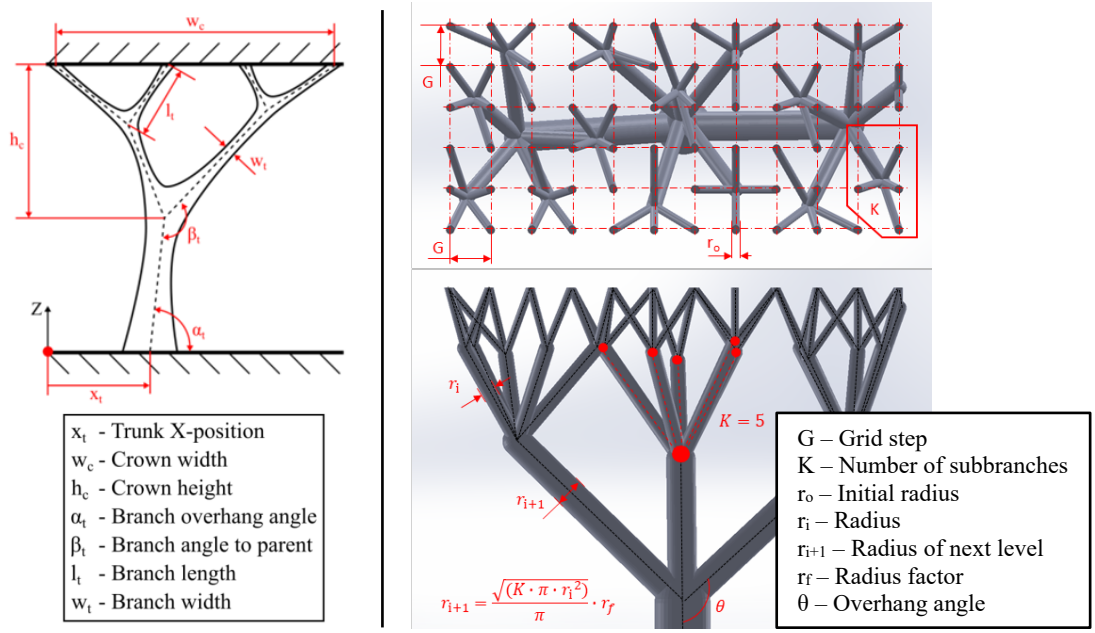

Fig. 4. (a) Tree evaluation parameters in 2D [14]; (b) tree adjustment parameters in 3D.

4.2. Implementation of tree support generation tool

The process shown in Fig. 5 (a) gives an overview of the tree generation process. It starts with the import of the STEP file of the 3D model. From this solid representation a mesh representation, composed of triangles,

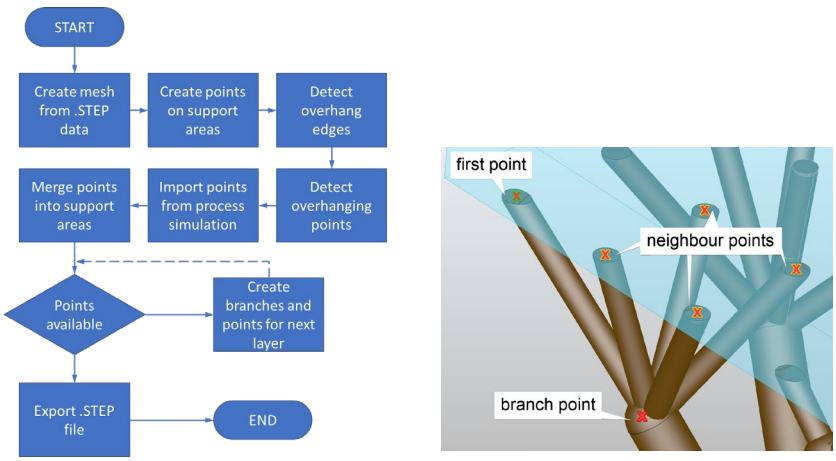

Fig. 5. (a) Process of the tree generation; (b) branch details.

is created. The solid representation of the part is kept in the background in order to output it at the end together with the tree as a STEP file. For the following calculations, only the mesh is used. The angle of the triangles surface normal is used to collect the triangles of interest for the support structures.

Connected triangles are used to create sub-meshes. For each of these sub-meshes a separate tree is generated as shown in Fig. 3 (c). With the ray tracing functions provided by trimesh [17], uniformly distributed points are created on these sub-meshes. In the next process steps, overhanging edges and overhanging points are detected. On the edges, uniformly distributed points are created also. As an optional step, points from external tools, e.g. process simulation, can be imported. These additional points are merged into the point clouds of the sub-meshes. Then an iteration is started for each sub-mesh to create the branches with the algorithm described in chapter 3.1. The main output of the algorithm are points which are used to generate the solid geometry of the branches with OpenCascade [19]. Each branch is built up from a cylinder which starts at the lower branch point and ends at the point on the surface in the first iteration. Then a sphere is generated at the top of the branch to ease the connection to the surface and to avoid gaps due to the different angles of the branches. In a last step the branches are cut with the solid representation of the part to get the correct intersections between part and branches. An exemplary result is shown in Fig. 5 (b).

At the end, the 3D trees are visualized together with the part. The user can export the trees with or without the part as a STEP or STL file. The STEP file can be used to manipulate the trees manually in a CAD system or to post process the part for NC programing.

4.3. Assessment of tree supports

Finite element analysis of the three parts showed an expected decrease of residual stresses in each part from 9.3e2 MPa to 1.32e-3 MPa overall, leaving a value around 34 MPa in the more complex regions of the parts (e.g. the 30°, 45°, 60° and 90° connections from overhanging surfaces to the main segment in Part 1). Even though the heat treatment provides a near-nullification of the residual stresses, a considerable deformation to the parts is induced during this process. The build plate removal through wire EDM during the

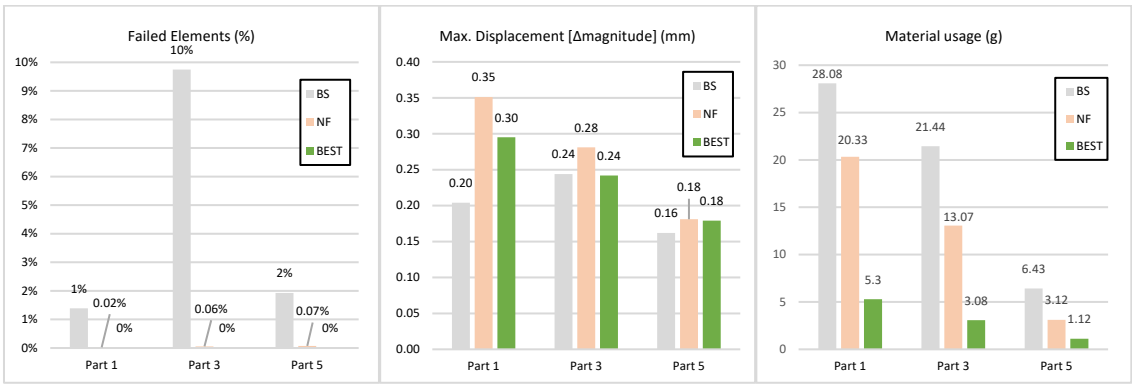

Fig. 6. Evaluation parameters; (a) Percentage of failed support elements; (b) maximum geometrical displacement; (c) material usage.

simulation dramatically affects the dimensions of the parts, which on average will experience an additional deformation in the Z-axis. This increase was more noticeable in Part 3 (P3) that has a bigger base area in contact with the build plate. The removal of the supports also induces an additional deformation in X, Y and Z locally in the contact region and overall in the Z-axis.

Although all parts were simulated successfully, the structure-type results in Fig. 6 (a) showed that the yield strength of Ti6Al4V was exceeded in some of the finite elements associated to the support structures during the simulations.

The amount of failed support elements is low but could lead to catastrophic failure during the process and could represent a risk when manufacturing the part. In this aspect, BS performed the worst while BESTS did not fail in any element. Further validation of this statement could be left for future works as these results are rather unexpected for widely used BS.

Displacements in X, Y and Z were obtained from the Netfabb simulation for each of the parts at each stage of the whole process. The values shown in Fig. 6 (b) represent the compound displacement achieved at the most critical point for each part. The BS elements mostly provide expectedly low distortion values for the parts analysed in the Z-axis. The maximum displacement obtained is 0.24 mm for Part 3 (P3) in the Z-axis. Fig. 6 (c) shows the amount of material used for each of the simulations. The amount of support material used for BS is the highest, using 596 % more material than BESTS when used for P3.

The results provided by the NFS show a higher displacement compared to BS. The displacement values for P1 and P5 respectively are 72 % and 11,7 % higher than BS. The material usage for NFS is reduced in all parts

Fig. 7. Printed benchmark parts with BEST supports; (a) Part 1; (b) Part 3; (c) Part 5.

when compared with BS, particularly for P3 (39 %) and P5 (51 %).

The BESTS show the most efficient use of material overall using 71 % less material on average with respect to NFS, being able to use only 1.12 g for P5. The displacement obtained when using BESTS is up to 14 % lower than NFS and even 1 % lower than BS in the case of P3.

In Fig. 7 the evaluated benchmark parts printed with the BEST supports can be seen.

Supplementary material regarding the obtained results can be found in Appendix A.1.

5. Conclusion and Outlook

In the present study we have successfully developed a tree-support software module for PBF-LB/M and validated it by FEA. The main conclusions are:

- A powerful tool capable of generating 3D tree support structures for complex parts has been developed successfully. FEA suggested that these support structures are efficient in terms of part fixation, stress relief and material usage.
- The developed code allows the user to manipulate up to 10 parameters of tree generation, allowing trees to be generated based on the user's preferences and the print job's requirements. The parameters can be adjusted according to the user's interest and the final goal of the print-job to obtain trees that perform well on the conservative side or are lightweight but fragile in order to save material.
- The BEST tree support tool can save the tree and the part as solid in the STEP format. Therefore, the user can modify the tree structure in a CAD system or use the tree data for NC programing.
- The generated trees for the selected benchmark parts were generated using the parameters that were found to be the most promising after the main runs. The best results were obtained using $K = 5$, $G = 2$ mm and $r_f = 0.85$.
- The displacements obtained from the BESTS have proven to be at the same level as the BS who are intended for minor displacement. However, the BESTS showed a reduction in material usage of up to 76 % and 85 % (both in P3) as well as an average reduction of 71 % and 83 % along all Parts when compared to NFS and BS respectively. Nevertheless, the BESTS sufficiently supported the expected mechanical loads and therefore showed a high potential for cost reduction.
- Netfabb simulation was able to provide displacements in X, Y and Z. A validation of these obtained values could be assessed with a 3D scanner.
- Three out of five benchmark parts from the previous work of Bartsch [14] were tested, leaving two parts for the future scope.

The BEST tree support code could represent a huge advantage in AM projects towards a more efficient use of material resources as well as its capability to be used in bigger parts. In particular, the overall costs of batch and serial production could be highly reduced by the implementation of this tool. In future works the developed code should be applied to a demonstrator part with complex features and should be manufactured. Analysis of geometrical accuracy of the part should be done to verify the results of the study at hand.

Acknowledgments

This research was funded by the Federal Ministry of Education and Research (BMBF) as part of the research project “Bäume als effiziente Stützstrukturen in der additiven Fertigung” (BEST); grant numbers 02P20E241 and 02P20E240.

The authors would like to thank Prathveraj Shetty (iLAS, TUHH) for carrying out the majority of the FEA as well as the generation of the trees.

Furthermore, the authors would like to thank Michael Schwartz (CENIT AG) for valuable discussions throughout the work on this study.

References

[1] A. Gebhardt, 'Generative Fertigungsverfahren', in *Generative Fertigungsverfahren*, Carl Hanser Verlag GmbH & Co. KG, 2013, p. I–XXIV. doi: 10.3139/9783446436527.fm.
[2] 'Fundamentals of Laser Powder Bed Fusion of Metals | ScienceDirect'. https://www.sciencedirect.com/book/9780128240908/fundamentals-of-laser-powder-bed-fusion-of-metals (accessed May 05, 2023).
[3] 'DIN EN ISO/ASTM 52911-1:2020-05, Additive Fertigung – Konstruktion – Teil 1: Laserbasierte Pulverbettfusion von Metallen (ISO/ASTM 52911-1:2019); Deutsche Fassung EN_ISO/ASTM 52911-1:2019', Beuth Verlag GmbH. doi: 10.31030/3060962.
[4] I. Gibson, D. Rosen, and B. Stucker, 'Additive Manufacturing Technologies'. New York, NY: Springer New York, 2015. doi: 10.1007/978-1-4939-2113-3.
[5] B. Blakey-Milner *et al.*, 'Metal additive manufacturing in aerospace: A review', *Mater. Des.*, vol. 209, p. 110008, Nov. 2021, doi: 10.1016/j.matdes.2021.110008.
[6] M. Salmi, 'Additive Manufacturing Processes in Medical Applications', *Materials*, vol. 14, no. 1, p. 191, Jan. 2021, doi: 10.3390/ma14010191.
[7] C. Sun, Y. Wang, M. D. McMurtrey, N. D. Jerred, F. Liou, and J. Li, 'Additive manufacturing for energy: A review', *Appl. Energy*, vol. 282, p. 116041, Jan. 2021, doi: 10.1016/j.apenergy.2020.116041.
[8] K. Bartsch, I. Herzog, C. Emmelmann, and F. Lange, 'A Novel Approach to Support Structures Optimized for Heat Dissipation in SLM by Combining Process Simulation with Topology Optimization'. 2019.
[9] K. Bartsch, J. Ohrenberg, and C. Emmelmann, 'Benchmark parts for the evaluation of optimized support structures in Laser Powder Bed Fusion of metals', *Procedia CIRP*, vol. 94, pp. 254–259, Jan. 2020, doi: 10.1016/j.procir.2020.09.048.
[10] 'Thermo-Mechanical Modeling of Additive Manufacturing - 1st Edition'. https://www.elsevier.com/books/thermo-mechanical-modeling-of-additive-manufacturing/gouge/978-0-12-811820-7 (accessed May 05, 2023).
[11] J. Jiang, X. Xu, and J. Stringer, 'Support Structures for Additive Manufacturing: A Review', *J. Manuf. Mater. Process.*, vol. 2, no. 4, Art. no. 4, Dec. 2018, doi: 10.3390/jmmp2040064.
[12] F. Mezzadri, V. Bouriakov, and X. Qian, 'Topology optimization of self-supporting support structures for additive manufacturing', *Addit. Manuf.*, vol. 21, pp. 666–682, May 2018, doi: 10.1016/j.addma.2018.04.016.
[13] 'Clever Support: Efficient Support Structure Generation for Digital Fabrication - Vanek - 2014 - Computer Graphics Forum - Wiley Online Library'. https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.12437 (accessed May 05, 2023).
[14] K. Bartsch, 'Digitalization of design for support structures in laser powder bed fusion of metals'. in *Light Engineering für die Praxis*. Cham: Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-22956-5.
[15] G. van Rossum and F. L. Drake, 'Python 3 Reference Manual: Python Documentation Manual', Release 3.0.1 [Repr.]. in Python documentation manual / Guido van Rossum; Fred L. Drake [ed.], no. Pt. 2. Hampton, NH: Python Software Foundation, 2010.
[16] C. R. Harris *et al.*, 'Array programming with NumPy', *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.
[17] Dawson-Haggerty et al., 'trimesh'. Apr. 11, 2023. [Online]. Available: https://trimsh.org
[18] T. Paviot, 'pythonocc (7.7.0)'. in Zenodo. Dec. 07, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7471333
[19] 'Open Cascade'. https://www.opencascade.com/
[20] P. Prusinkiewicz, J. Hanan, A. Lindenmayer, F. D. Fracchia, and K. Krithivasan, 'Lindenmayer systems, fractals', and plants. in *Lecture notes in biomathematics*, no. 79. New York Berlin Heidelberg: Springer, 1989.
[21] A. Runions, B. Lane, and P. Prusinkiewicz, 'Modeling Trees with a Space Colonization Algorithm', in *Eurographics Workshop on Natural Phenomena*, Czech Republic: The Eurographics Association, 2007, p. 8 pages. doi: 10.2312/NPH/NPH07/063-070.
[22] J. L. Bentley, 'K-d trees for semidynamic point sets', in *Proceedings of the sixth annual symposium on Computational geometry - SCG '90*, Berkley, California, United States: ACM Press, 1990, pp. 187–197. doi: 10.1145/98524.98564.

Appendix

A.1. Supplementary data

DOI of data set: https://doi.org/10.15480/336.5207
Handle of data set: http://hdl.handle.net/11420/15472