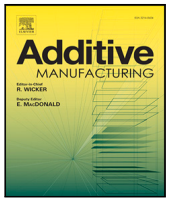
Additive Manufacturing 94 (2024) 104426

Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma



Research paper

Looping: Load-oriented optimized paths in non-planar geometry

Johann Kipping *, Doran Nettig, Thorsten Schüppstuhl

Institute for Aircraft Production Technology, Hamburg University of Technology, Denickestr. 17, Hamburg 21073, Germany

ARTICLE INFO

Keywords:
Continuous
Path planning
Non-planar
Load-oriented
Multi-axis

ABSTRACT

Effective material utilization in the additive manufacturing of lightweight components is of increasing importance. The Looping (Load-oriented optimized paths in non-planar geometry) method presented in this work enables the translation of desired material orientations into suitable manufacturing instructions. The desired material orientations are derived from the principal stress directions that would manifest for an isotropic material. By employing non-planar slicing, these orientations can be followed by the deposited material beads. The novel path planning algorithm combines load-orientation and path continuity. While this can be beneficial for load-oriented printing in general, it is an especially significant step for load-oriented printing of continuous fiber reinforced polymers. The ability to follow desired material orientations with continuous paths shows particularly high potential for highly anisotropic fiber reinforced polymers. The algorithms are implemented and demonstrated in a complete process chain. However, challenges remain in the optimization of the orientation and manufacturing system for fiber reinforced polymers, which are not the focus of this work. For this reason, the process chain is realized for a neat polymer. In this context, the developed method is computationally evaluated with respect to layer height, unfilled areas, manufacturing time, geometric accuracy, and physical fabrication. The continuous and load-oriented path planning algorithm is tested against a continuous contour parallel approach and planar slicing through tensile testing. The investigations show an applicability of the process chain to successfully produce complex parts with the desired load-oriented paths. The proposed algorithm shows an increase in mechanical performance compared to the contour parallel approach highlighting its potential for non-planar printing. However, it is also found that limitations of the non-planar manufacturing process still limit its potential to surpass optimally oriented planar printing for the investigated geometry.

1. Introduction

The conventional path planning approach for FFF is based on planar algorithms. The part is sliced with planes horizontal to the build plate to obtain the surfaces on which path planning is done [1]. As the inter-layer adhesion represents the weakest bonding [2–4], 2.5D slicing can result in distinct stiffness losses if the optimal material orientations are out-of-plane [5]. The inability to align the print- and load-direction becomes increasingly problematic as fiber reinforced polymers (FRP) with improved, yet even more anisotropic material properties are being adopted into additive manufacturing (AM) processes [6–8].

One solution for this shortcoming is non-planar slicing. With this method, the constituting layers are not flat and parallel to the print bed, but can assume arbitrary manifold shapes [9]. The optimal material orientations can include turbulent and divergent regions [10]. These regions oftentimes result in slices with very high curvatures, which cannot be manufactured due to collisions and self intersections [11,12]. Bridging the gap between achieving optimal material orientations and part manufacturability remains as one of the most challenging problems in the field of non-planar slicing for FFF. Most state of the art approaches for non-planar slicing obtain the layers as isosurfaces of a scalar field [13–15]. The method by which this scalar field is computed is crucial for the achievable print quality and mechanical performance of the printed part. Popular choices are the geodesic distance as a scalar field [16–18] or the vector field of maximum principal stress directions, from which the scalar field is computed by solving a Poisson or similar equations [11,19–21]. These vector fields are obtained by finite-element-analysis (FEA). One of the major challenges described in the literature is the orientational ambiguity, divergence and turbulence of these vector fields. These properties lead to lower adequacy to the stress flow, increased computational complexity and numerical errors of the developed methods. The slicing method from Zhang et al. [12] assigns different weights to enable to balance the 3 most significant advantages of non-planar printing. These are support-free printing,

* Corresponding author.
E-mail address: johann.kipping@tuhh.de (J. Kipping).

https://doi.org/10.1016/j.addma.2024.104426
Received 3 January 2024; Received in revised form 25 July 2024; Accepted 9 September 2024
Available online 24 September 2024

surface quality and load-orientation. The model is deformed and layers are generated using adaptive slicing.

In addition to the slicing, the path planning has to take the desired material orientations into account as well. The most advanced approaches project the orientation field onto the slices and solve a Poisson equation to obtain a scalar field. The isolines of this field can be connected into a load-oriented path. Publications that have followed this approach include [11,18,20,21] for the non-planar case and [22] for planar paths. Other possibilities to achieve load-orientation are wave projection for infill patterns [23] and graph-based approaches [24] where the amount of possible orientations is severely limited. Furthermore, some approaches combine topology optimization with tracing the stress directions, which results in uneven spacing [25].

A challenge posed if such methods are to be applied to printing FRPs is the necessity for a fiber cutting procedure for discontinuities in the nozzle's path. The cutting mechanism has to be integrated into the print head, which is achieved by industrial solutions with a knife further up the filament path [26,27]. The consequence is a minimum fiber length of around 40 mm for both systems, making it impossible to fill small regions with fiber. Furthermore, discontinuities generally decrease the mechanical performance and print quality of FRPs as the length of the fibers themselves has a great impact on strength [28,29]. A further decrease can result from the excessive resin build-up with uncertain fiber position when cutting the fiber and lifting the nozzle [30]. In addition to addressing these issues, non-retraction paths reduce print-time by eliminating the need for travel motions [30]. In this work, the notion of continuity is used in the sense stated above: A continuous path for a given surface is a non-retraction path that covers the surface in a single stroke. The problem of complete coverage of surfaces is closely related to the traveling-salesman-problem and many variants have been proven to be NP-hard [31]. If retraction and tracing across already covered areas is not allowed, the method of connecting adjacent lines followed in [11,18,22] cannot generally yield a continuous path for any non-convex geometry. This is due to isolated areas representing dead ends. These correspond to leafs in the connection graph of neighboring lines, similar to the "bottlenecks" described in [31]. For contour parallel methods these are the maxima of the distance transform from the contour and for direction parallel approaches they correspond to the non-convex regions which cannot be connect without tracing back. An example of this are the upper and lower rightmost areas of the surface in Fig. 6 and the lower left area of the surface in Fig. 7. The reasons stated above clearly show, that a custom path planning solution enabling load-orientation in conjunction with continuity is highly desirable.

The most prominent approaches that achieve global continuity use space-filling curves. Specifically, connected Fermat Spirals [32,33] and Hilbert curves [34] were used. A method that generates a space-filling curve with graph theory employs a stochastic optimization method to avoid self intersection and establish load-orientation [35]. Other graph based methods achieve full continuity by constructing Eulerian graphs [30,36], but generally suffer from low resolution. Other graph based methods include traveling back on edges to achieve continuity [37]. Like the go-and-back method, where the path is generated by constructing the path with classical methods and double the spacing [38]. This creates two lines for every path, one of which can be retraced to close the path to full continuity. A similar method was applied for Honeycomb patterns, which additionally allow to dynamically adjust infill density [39]. Without increasing intra-layer continuity, process planning techniques can decrease the number of discontinuities by finishing several layers of branching sections before continuing with the other sections [40,41].

Several approaches exist that combine continuity and load-orientation. Most importantly, Fang et al. demonstrate a method in a recent publication, where continuous fiber loops are incorporated in load-oriented slices [21]. The proposed method shows a pronounced increase in stiffness and strength. The printed parts exhibit very low fiber volume fraction, however, as only some layers were filled with fibers, which are also spaced far apart inside the layer. Wang et al. propose a planar algorithm, where continuity is achieved by allowing for material deposition outside the perimeter in the form of loops, requiring additional post processing step of cutting away the excess material [25]. Placing material outside the perimeter allowed to accurately represent the desired material orientations in the planned path, as no compromise had to be made to ensure continuity. Bedel et al. construct closed loops by applying a stochastic optimization method to a graph-based approach to form a space-filling load-oriented curve [35]. However, their algorithm struggles to create long continuous segments. The only claim to a non-planar approach besides [21] is made by Li et al. [20]. They severely smooth the orientation field until only parallel lines remain. Additionally, the method is required to have an even number of intersection points with the boundary contours, to be able to compute an Eulerian path, which is the basis for their continuous path planning. An even number of intersection points is only guaranteed for the case of parallel lines, making their approach inapplicable for load-orientation.

This paper presents several improvements in continuous and load-oriented 3D-printing. These are shown on a complete process chain from virtual to physical geometry. An overview of the computation steps as well as their in- and outputs can be seen in the flowchart in Fig. 1. The desired material orientations are derived from an FEA of a volumetric part assuming isotropic material properties, which is described in Section 2.2. The results are then passed on to the slicing and path planning algorithms for the generation of manufacturing instructions. The algorithms for the load-oriented non-planar slicing, support generation, and the continuous and load-oriented path planning are presented and discussed in Section 2.3 – Section 2.5. Finally, the parts are manufactured on a robotic 3D-printing machine and the approach is evaluated using tensile testing, which is described in Section 3 along with the evaluation of the computational results. The slicing and path planning algorithms in particular show great potential for the printing of FRPs. Yet, especially the optimization of material orientations and the manufacturing system still pose significant challenges to realize a complete and optimal process for such materials. For that reason the complete process chain is demonstrated and evaluated using neat polymer.

2. Materials & methods

2.1. Materials

Fiber reinforced polymers represent the material class where the highest benefit of load-oriented slicing and path planning is expected, due to their high degree of anisotropy. As mentioned in Section 1, a highly dynamic and local variation of layer height in conjunction with a high flexibility in layer orientation is necessary in order to fabricate parts with non-planar layers. The ability to achieve such layer height variations and orientation flexibility together has not been successfully implemented for FRP printing systems. The material choice is therefore still limited to neat polymers by the manufacturing process. The tests are conducted with PLA as the material for the parts. For the FEA the material is assumed to have a Young's modulus of 3800 MPa and a Poisson ratio of 0.35. PVA is used as the support material, which can be removed by soaking the printed part in water. The parts have to be dried afterwards, to remove any water absorbed by the PLA. The PLA is sourced from Easyprint in the color red and the support material is PrimaSelect™ PVA+ in natural color.

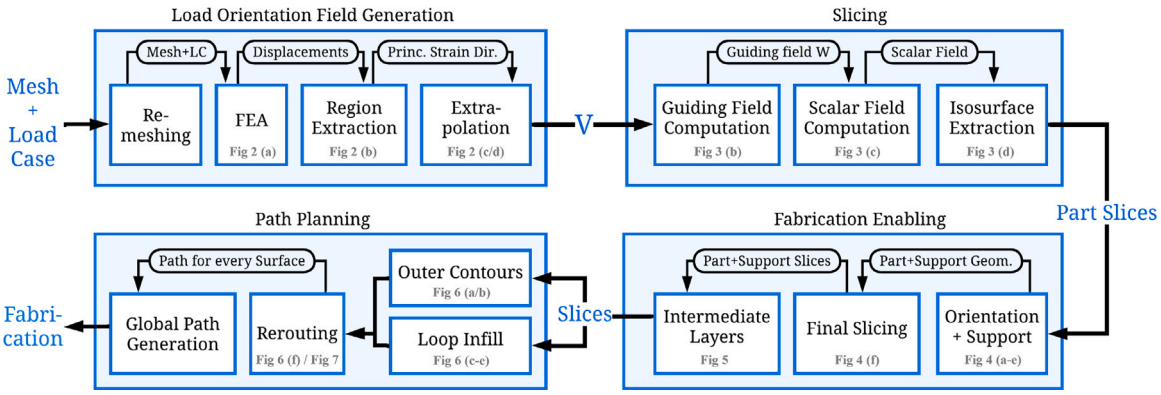


Fig. 1. Flowchart illustrating the computation pipeline described in this work. Input to the pipeline is the mesh representing the object and the load case, output are the fabrication instructions.

2.2. Finite element analysis

The goal of the slicing and path planning algorithms presented in the later sections is to enable taking desirable material orientations into account. The algorithms therefore require a description of these desired material orientation as input. Ideally, the desired material orientations would be generated through the complex modeling and optimization of the anisotropic and in the case of FRPs inhomogeneous material. The optimization could even go so far as to model the printing process itself and iteratively take the results of slicing and path planning into account. Such modeling and optimization approaches are however not part of the scope of this work. Instead, like other works that focus on load-oriented slicing and/or path planning [11,18,19,22,25], the complexity in finding desired material orientations is strongly reduced. The slicing and path planning algorithms do not depend on the specific method used to generate the material direction field though. They could therefore be combined with a more complex optimization in the future, which will become increasingly important to fully utilize the potential of FRPs. As a first step it is assumed that a significant improvement can already be achieved by aligning the material with the load paths that would manifest if the part was made from an isotropic material, especially when considering printing with a neat polymer. Therefore, a homogeneous, isotropic and elastic material is assumed as a basis. The desired material orientations are then generated by aligning the printed fiber strands with the expected maximum principal stress directions under a given load case.

Input to the computation pipeline can be any mesh. The FEA starts with the generation of an appropriate mesh $\mathcal{T}$ by remeshing and tetrahedralization of the input mesh to ensure consistent edge lengths and tetrahedra volumes. Allowing arbitrary meshes, makes the pipeline flexible, as quadrilateral meshes are also very common in structural optimization and triangular surface meshes are most often used for 3D-printing applications. Topology optimization can be applied in generating the input geometry to reduce part mass. In the next step, loads, boundary conditions and the relevant material properties are defined on the mesh surface. From the mesh and material information, a global stiffness matrix K is constructed based on the element stiffness matrices. With the stiffness matrix K, the loads f and other boundary conditions, the system of equations

$$Ku = f \tag{1}$$

can be solved for u. The vector field u describes the displacements between the nodal locations of the undeformed reference configuration and the deformed current configuration resulting from loads and other boundary conditions. An example of a field u can be seen in Fig. 2 (a). From the displacement field the strains of the material can be calculated and linked to the stresses in the material by a constitutive law. For simplicity a linear elastic model is used in the calculations in this work.

Up to this point the method presented in this work coincides with those used in previous works [11,18,19,22].

The state-of-the-art approaches for FEA based slicing obtain the principal stresses $[\sigma_{max}, \sigma_{mid}, \sigma_{min}]$ and stress directions $[\tau_{max}, \tau_{mid}, \tau_{min}]$ for every element $e \in \mathcal{T}$ by computing the eigendecomposition of the stress tensor. The resulting vector fields form the basis for constructing a guiding field and a load-orientation field. The load-orientation field is then used in the load-oriented path planning. Critical regions are then extracted by using a metric calculated from the absolute values of the principal stresses. In this last step the current approaches come with two major drawbacks: If the absolute value of the principal stresses is used like in [11,18,22], no distinction is made between tension and compression regions. Axial loading of the material beads in tension is very beneficial for stiffness and strength. However, axial loading in compression might lead to strength losses due to buckling and delamination, especially for composite materials [42]. A further issue is that the fields of principal stress directions have ambiguous orientations. The directions $\vec{\tau_{max}}$ are therefore inhomogeneously oriented, meaning that the dot product of vectors of neighboring elements can be negative. To achieve manufacturable surfaces from the ambiguously oriented material directions further mitigation measures are necessary. These measures include flooding algorithms, reorienting vectors in critical regions, filtering and Cartesian rectification [11,18–20,22]. All of these methods require, however, manipulation and smoothing of the field that inherently leads to a loss of information, which in turn lowers the load-orientation. The method used in [11] has an especially high computational complexity of $O(n^2)$ with n being the number of critical regions. Currently, no method capable of generating manufacturable surfaces from ambiguously oriented direction fields for arbitrary part geometries is known to the authors that does not include costly reorientation measures.

To avoid these costly reorientation measures and to distinguish regions of tension and compression, the approach proposed in this work differs from the state-of-the-art. A suitable field and metric can be found in inspecting the intermediate steps of the conventional calculation pipeline. Note that in the following a Lagrangian approach is chosen. Alternatively, an Eulerian approach could be taken as well. As stated, most state-of-the-art approaches including the one presented in this work utilize a linear elastic material model. However, to not limit the method more than necessary, linear elasticity can be viewed as a special case of the Saint Venant–Kirchhoff material, which in turn is a special case of a hyperelastic material. The discussion is therefore carried out to be applicable to any homogeneous, isotropic and hyperelastic material, which includes linear elasticity as one particular choice. For the assumed homogeneous, isotropic and hyperelastic material, the second Piola–Kirchhoff stresses S can be calculated as an isotropic tensor function of the symmetric and positive definite right Cauchy–Green deformation tensor C as

$$S = G(C). \tag{2}$$

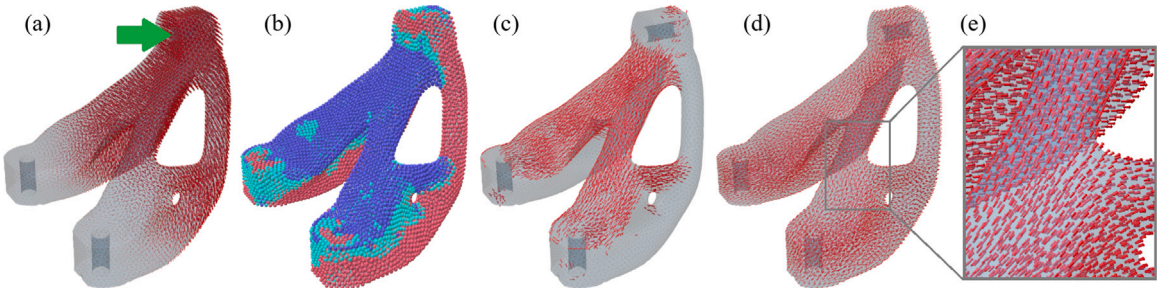


Fig. 2. Computation of guiding vector field for an exemplary part. (a) Displacement vector field u, scaled for visibility. Applied load indicated by green arrow, fixed boundary condition at bottom holes. (b) Regions for the computation of vector fields; Compressive regions in red, tensile regions in blue and cyan, blue is the selected region $\mathcal{R}_t$. (c) Maximum principal direction vector field of U selected by maximum principal value of E. (d) Resulting extrapolated vector field V. (e) Zoomed in area of the vector field V, the vector field has homogeneous orientation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

C is calculated as the square of the deformation gradient tensor F.

$$C = F^2 \tag{3}$$

The right polar decomposition of F allows for its decomposition into the orthonormal rotation tensor R and the positive definite symmetric right stretch tensor U. Using the property $R = R^\top$, C can therefore be identified with U, according to the following equation:

$$C = F^2 = F^\top F = U^\top R^\top R U = U^\top U = U^2 \tag{4}$$

It can be shown, that the eigenvectors of $G(C)$ are identical to the eigenvectors of C [43] (p.194). The square of a symmetric positive definite matrix has the same eigenvectors as the matrix, so C has the same eigenvectors as U. Together with Eq. (2), it can therefore be concluded that the eigenvectors of S and U are identical. By definition, the strain energy density function of a hyperelastic material can only increase with an increase in strain. This monotonic relationship between stress and strain implies that the maximum principal strain and stress directions are aligned as well. In conclusion, for an isotropic, homogeneous and hyperelastic material, the principal directions of stress, strain and stretch are all identical up to orientation. Additionally, which of these directions corresponds to the maximum of strain or stress respectively is identical as well. Utilizing this property, the approach proposed in this work does not use the absolute value of the principal stress directions as the desired material directions. Instead, the stretch field U is used to obtain the load-orientation vector field V. Different from strain or stress this field is directionally unambiguous and therefore serves as a homogeneously oriented surrogate for the principal stress directions. Although well suited for V, U is less suited for the region selection. While the eigenvalues of U have to be positive and are small in domains of compression, the eigenvalues of the Lagrangian finite strain tensor E are positive in tension and negative in compression domains. The distinct sign difference makes the region differentiation easier in E, which is why it is chosen for the region selection. The stresses could be chosen as a metric as well, but as was shown this would require an additional calculation step. The eigenvalues of E will be denoted as $[\epsilon_{max}, \epsilon_{mid}, \epsilon_{min}]$. Only the regions of maximum positive strain are used for defining the layer orientations. Consequently, the regions where compression represents the largest absolute value of the principle strain are taken out of the optimization. The algorithm thus focuses on the regions of tensile strain. The region defined by these conditions is denoted as

$$\mathcal{R}_t = \{e \in \mathcal{T} | \epsilon_{max}(e) > 0 \ \wedge \ \epsilon_{max}(e) > k|\epsilon_{min}(e)|\}. \tag{5}$$

The parameter k represents a threshold value to avoid including regions where $\epsilon_{max}(e)$ is too small, which can result in turbulent vectors. It is chosen as $k = 2$ for all computations in this work. An example of such a region can be seen in Fig. 2 (b). The tensile domain is displayed as cyan and blue, while the region defined by the factor k is the blue area. It can be clearly seen how only the regions are included where a clear distinction can be made towards the tensile loading of the cells. Part (c) of Fig. 2 shows the vectors $\vec{\tau_{max}}$ extracted from the field. Similar to previous approaches, V is computed by extrapolating the eigenvectors $v_{max}(e)$ belonging to $\epsilon_{max}(e)$ for $e \in \mathcal{R}_t$ into $\mathcal{T} \setminus \mathcal{R}_t$. This final field V can be seen in Fig. 2 (d).

In summary this work utilizes the same general computation pipeline for the FEA as similar works. However, in choosing a different field and metric for the orientation field and region selection, the contributions of this work on are the following: The use of the stretch field U as the basis for computing the load-orientation field V removes the necessity for additional costly reorientation steps in the computation pipeline, which could otherwise limit the possible part complexity severely. Additionally using the maximum principal strains instead of the absolute principal stresses for the region selection focuses the region selection on the regions of tensile strain, with the added benefit of removing the need for stress computation.

2.3. Field computation

Having computed the desired material orientations V in the region $\mathcal{R}_t$ with the methods described above, this information is utilized to compute paths for the nozzle to create a load-oriented and optimized part. As mentioned in Section 1, the process starts by slicing the part with surfaces on which the nozzle-paths can be computed. These non-planar surfaces are able to adapt to the out-of-plane material orientations. The adaption is achieved through computation of a scalar field by solving a Poisson equation with a prior computed vector field W on a tetrahedral mesh. The isosurfaces of this scalar field are then extracted as the non-planar slices used for path planning.

In order to compute the slices, the first step is the extrapolation of the $v_{max}(e)$ for $e \in \mathcal{R}_t$ into $\mathcal{T} \setminus \mathcal{R}_t$ to yield the vector field V. V can be computed by minimizing the Dirichlet energy

$$E_D(\mathbf{u}) := \int_\Omega \mathbf{u}\Delta\mathbf{u} \, dA, \tag{6}$$

which is a measure of deviation from the constant function, expressing a notion of smoothness. To solve the minimization for a set boundary conditions, the Laplace problem

$$\begin{aligned} \Delta\mathbf{u} &= 0 \quad \text{on } \Omega \\ \mathbf{u} &= v_{max} \quad \text{on } \partial\Omega \end{aligned} \tag{7}$$

is solved. For the discrete case $\Omega \mathrel{\hat{=}} \mathcal{T} \setminus \mathcal{R}_t$, $\partial\Omega \mathrel{\hat{=}} \mathcal{R}_t$ and $u \mathrel{\hat{=}} v$ on $\mathcal{R}_t$. Further detail on how this is solved on a tetrahedral mesh can be found in [19]. Having computed the extrapolation, the homogeneously oriented vector field V is obtained. The performed extrapolation from the selected region of tensile stresses $\mathcal{R}_t$ into the rest of the part domain $\mathcal{T} \setminus \mathcal{R}_t$ is visualized in Fig. 2 (c) to (d). The thereby obtained field V is used as an input for the computation of slices and shown again in Fig. 3 (a).

For the computation of the governing vector field W that is serving as the basis for computing the slices, the requirements for this field are stated again in the following. The governing vector field W has to be:

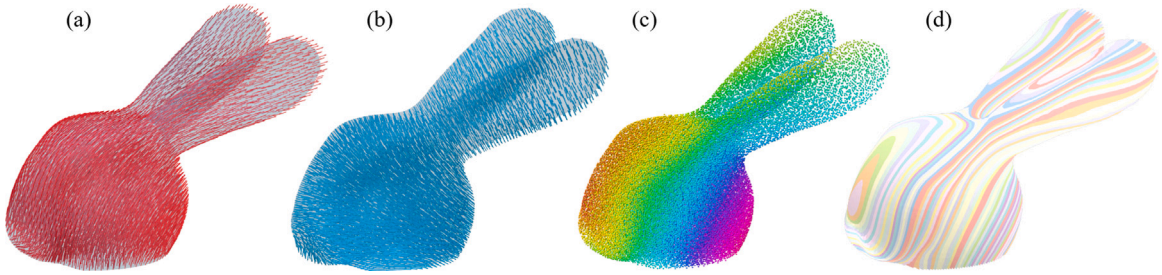


Fig. 3. Computation process of the fields and slices for an exemplary part. (a) Load-oriented vector field V. (b) Guiding vector field W which approximately represents the surface normals of the slices. (c) Scalar field ψ_{tet} computed from W. (d) Slices that constitute the part, on which the path planning is done.

1. Orthogonal in every point to V to ensure $v \in V$ is as close to the tangent space of the isosurfaces as possible to prevent lossy projection.
2. Minimally turbulent and divergent in order to minimize surface curvature and irregularities in layer height.
3. Directionally homogeneous to enable a meaningful Poisson solution.

The first two criteria are achieved by simply computing a mean direction of V, taking the cross product with a unit direction, resulting in a vector $\boldsymbol{\mu} = \overline{V} \times \vec{e}_1$ that has the property of lying orthogonal to the mean of vectors in V. Constructing $\boldsymbol{\mu}$ is done to now have a consistent direction with which the optimized vector field W can be computed by taking the cross product to yield

$$W_i = \frac{1}{|\boldsymbol{\mu}|} \boldsymbol{\mu} \times V_i \quad . \tag{8}$$

Taking the mean direction minimizes the turbulences of the resulting field, as $\boldsymbol{\mu}$ and V_i are unlikely to align in any region. Alignment would result in a bad conditioning, which means fast changes in direction in W for small changes in distance as $|\boldsymbol{\mu} \times V_i|$ goes to zero and the resulting direction is decided by minor deviations in alignment. This method of computing W, resulting in a field with less turbulence, constitutes another improvement in this work. To summarize, the degree of freedom that results from setting only the optimal material orientation of the fibers, but needing two vectors to describe the tangent vectors on the local isosurface is exploited to increase manufacturability in a similar way to [18].

The third criterion is already fulfilled by the measures described in Section 2.2. The field W fulfilling all the listed criteria and computed in the way described above can be seen in Fig. 3 (b) for the bunny test part.

As W describes the orientation of the slices through their normal vectors, to compute the slices a scalar field is created whose gradient minimizes the difference to W and extract the isosurfaces at specific values. This is accomplished by solving the Poisson equation obtained by taking the divergence in the following manner

$$W = \nabla \psi_{tet} \quad \Rightarrow \quad \nabla \cdot W = \Delta \psi_{tet} \; . \tag{9}$$

As the Laplace operator can be represented as a linear combination on the nodes, resulting in a system of linear equations, this Poisson equation can be solved efficiently on tetrahedral meshes when using an interpolation function on the nodes. The result from such a computation for the bunny test part can be seen in Fig. 3 (c) where the scalar field values are displayed as color on the meshes nodes. The slices that result from extracting isosurfaces from this scalar field can be seen in Fig. 3 (d).

To achieve a higher alignment of W and $\nabla \psi_{tet}$ every vector in W is multiplied by weighting factor $w(e)$ for element e in Eq. (9). Employing weighting factors increases the influence of elements where higher stresses are expected by scaling the vectors before the computation of the scalar field. In this work, the weighting factors are set as follows

$$w(e) = max(0, min(p_{90}, \epsilon_{max}(e))) + 1 \tag{10}$$

with p_{90} being the 90th percentile of ϵ_{max}. The maximum resulting weighting factor is 2 for the region of highest maximum principal strain. The weighting introduced here increases the possibility for layer distances being greater than the maximum h_{max} achievable by the hardware. To still achieve valid manufacturing instructions the introduction of intermediate layers is necessary. The generation of these intermediate layers, of support geometry and further fabrication enabling methods are described in detail in the next section.

2.4. Fabrication enabling and support structures

To enable the fabrication without collision of the print head with the build plate the part has to be reoriented. The method used to ensure printability in this regard is explained in the first part of this section. In order to create the non-planar layers from the flat built plate, support structures are oftentimes used in planar printing to uphold overhanging structures. As stated in the introduction, non-planar printing can eliminate the need for supports, while sacrificing load-orientation. As this work is focused on the latter, support structures are not avoided to not further constrain the possible layer orientations. The second part of this section is concerned with the generation of the support structure geometry. As mentioned in the previous section, intermediate layers are necessary to ensure the hardware restrictions are met. The method used to insert these layers is presented in the final subsection.

2.4.1. Part orientation

The starting point for the part reorientation measure is the observation that minimal deviation from printing vertically on the built plate is desired to minimize the reorientation effort and reduce the risk for collision with the build plate. Therefore, the simple way to choose part orientation implemented in this work is to compute the average normal direction of all layers once the part has been sliced for the first time. The slices are then transformed by rotating the averaged normal into the positive z-direction. Additionally, a user defined buffer height h_{buff} is established between the lowest part of the reoriented model and the build plate by translation.

2.4.2. Support geometry

The necessity of support structures has already been pointed out. Their construction represents an increase in manufacturing time, which makes it desirable to minimize the necessary support. A requirement that arises from non-planar printing is the compatibility of part layers and support layers. For the general case it is impossible to construct planar support layers before the non-planar part layers, as this would lead to reachability issues. Constructing them in parallel is not possible, because individual non-planar layers would need to be cut into small sections printed successively, which in turn would make continually placing material beads in the load direction impossible. This motivates the past approaches to extrapolate the vector and by extensions the scalar fields into the previously computed support geometry. The algorithm proposed in this work builds on the same idea.

Starting the computation of support geometry, elements of the outer surface of $\mathcal{T}$ denoted by $\partial\mathcal{T}$ are examined for requiring support. Areas

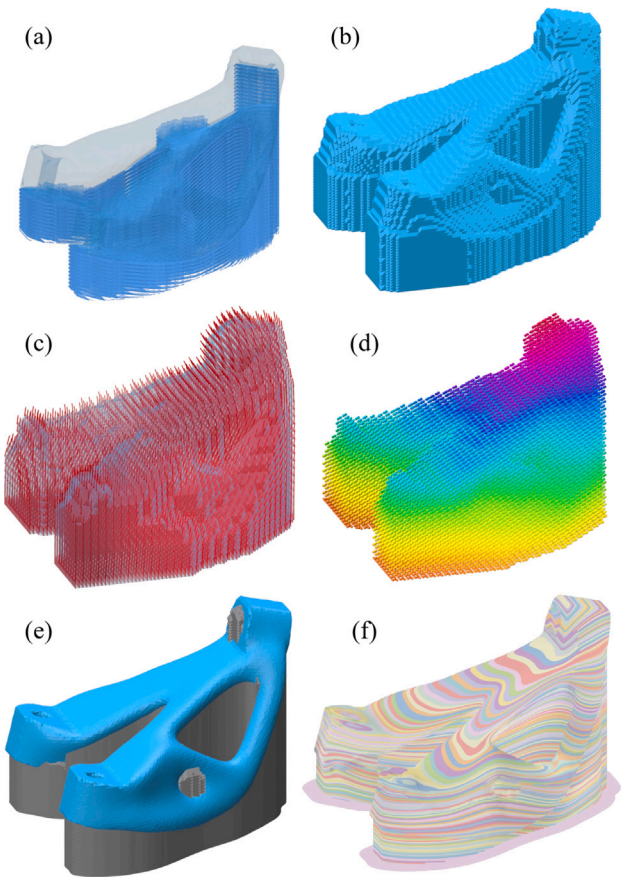


Fig. 4. Computation process of the support structure and slices. (a) Point cloud P. (b) Compound part and support mesh $\mathcal{M}$ computed from point cloud P and the part mesh $\mathcal{T}$. (c) Extrapolated guidance vector field W. (d) Scalar field ψ_{tet} in $\mathcal{M}$. (e) Part and support geometry $\mathcal{P}$ and $\mathcal{S}$. (f) Slices for part and support geometry extracted as isosurfaces from the scalar field ψ_{tet} in $\mathcal{P}$ and $\mathcal{S}$ respectively.

which need support are identified by defining the surface $\partial\mathcal{T}_{supp} = \{e \in \partial\mathcal{T} \mid n_e \cdot [0,0,-1]^T > 0.5\}$. $\partial\mathcal{T}_{supp}$ therefore includes those triangles whose normal n_e exceeds a value of 0.5 in the dot product with the negative z unit vector, meaning a wall angle of over 45° is present. This wall angle has been observed to be an adequate threshold value. The points belonging to the triangles in need of support are iteratively transformed down by a set value to form a point cloud as follows

$$P = \{p - i \cdot t_p \cdot \vec{e_z} \mid p \in \partial\mathcal{T}_{supp}, \quad i \in \{1, 2, \dots, n_p\}\}. \tag{11}$$

The translational interval was chosen as $t_p = 0.5$ mm and n_p was set such that the highest z-coordinate of the point cloud ended up below the build plate. The point cloud P can be seen in Fig. 4 (a), with points below the X-Y plane being cut off. Using this point cloud P, the compound part and support mesh $\mathcal{M}$, shown in Fig. 4 (b), is created. First a tetrahedral grid $\mathcal{G}$ with a high spatial resolution is created, which spans the bounding box of $\mathcal{T}$ including the buffer space below the part. Then $\mathcal{M}$ is obtained by extracting the cells from $\mathcal{G}$ that lie inside $\partial\mathcal{T}$ and in proximity to P.

Next, W is extrapolated from $\mathcal{T}$ into $\mathcal{M}\backslash\mathcal{T}$ by again solving equation (7) with $\partial\Omega$ now being the already computed vector field in $\mathcal{T}$ and a build plate region $\mathcal{B}$ where all vectors are set as $[0,0,1]^T$. $\mathcal{B}$ is extracted from $\mathcal{M}$ by extracting all points with a z-coordinate below $1/2h_{buff}$. The into the support domain extrapolated guidance vector field W is displayed in Fig. 4 (c).

Having computed W, the same process as in Section 2.3 with equation (9) and weighting from Eq. (10) is followed to obtain the scalar field ψ_{tet} in $\mathcal{M}$, which can be found in Fig. 4 (d).

The last step before slicing is to separate $\mathcal{M}$ into the part geometry $\mathcal{P}$ and the support geometry $\mathcal{S}$ by computing

$$\begin{aligned} \mathcal{P} &= \mathcal{M} \cap \mathcal{T} \\ \mathcal{S} &= \mathcal{M} \cap proj(extr(\partial\mathcal{T})) \setminus \mathcal{T} \end{aligned} \tag{12}$$

with *proj* being a shorthand for the operation of projecting a mesh onto the X-Y plane and remeshing into a valid mesh and *extr* denoting the operation of extruding along the Z-axis. The result of the partition into part and support geometry can be seen in Fig. 4 (e), where the part and support geometry are shown in different colors.

2.4.3. Slicing

$\mathcal{P}$ and $\mathcal{S}$ now contain information on the field ψ_{tet} from which the slices can be extracted as isosurfaces. Due to the weighting process, the field values do not correspond to the unit measure, which is corrected for by the measures described in the following and algorithm 1. To define the slicing intervals correctly, the field is first sliced naively with the interval arbitrarily set as $l_{int} = h_{max}$. The parameters h_{min} and h_{max} denote the minimum and maximum possible layer heights for the printing hardware that is used. The initial slicing with $l_{int} = h_{max}$ just serves as a precomputation in order to obtain slices whose distance roughly lies in the correct range. After computing the maximum distance d_{max} of a slice to its predecessor over all slices using the method described by algorithm 2, the new interval is then set as $l_{new} = h_{max}/d_{max} \cdot l_{int}$. This scaling is done in order to have the maximum distance be roughly equal to h_{max} when slicing again with this new interval. The slices are then obtained by extracting isosurfaces $\mathcal{S}_h$ from the resulting scalar field ψ_{tet} at the values $h \in \{x | x = i \cdot l_{new}, x \leq max(\psi_{tet}) - l_{new}, i \in \mathbb{N}\}$. This computation can be done very efficiently as the Laplace–Beltrami operator can be represented in the form of a sparse symmetric positive semi definite matrix. The resulting isosurfaces $\mathcal{S}_h$ can be seen in Fig. 4 (f).

Algorithm 1 Slicing

Require: $\mathcal{P}, \psi_{tet}, \mathcal{L}_{init}$
$l_{int} \leftarrow h_{max}$
$\mathcal{S}_h \leftarrow extract\_slices(\mathcal{P}, l_{int})$
$d_{max} \leftarrow max(compute\_layer\_height(\mathcal{S}_h, \mathcal{L}_{init}))$
$l_{new} \leftarrow \frac{h_{max}}{d_{max}} \cdot l_{int}$
$\mathcal{S}_h \leftarrow extract\_slices(\mathcal{P}, l_{new})$

The layer height computation that is necessary for the slicing process and setting the extrusion speed is done by iterating over all isovalues h, while merging the already checked layers into the object $\mathcal{L}_s$. Conventional approaches for such distance computations use raycasting along the surface normals or simply computing the minimum Cartesian distance to the lower layers for every element of the surface. Both approaches can lead to inaccurate results. For raycasting, taking the surface normals can result in larger distances if the surfaces are angled, or the rays miss $\mathcal{L}_s$ entirely. The Cartesian distance increases linearly in overhang areas, which results in over-extrusion in these regions. In this work, both methods are combined in the following way, which is also described in algorithm 2. Rays are cast along the surface normals and checked for intersection with $\mathcal{L}_s$. In these areas, minimal Cartesian distance is computed and saved as the layer height. For regions where no intersection is found, the value found for the closest neighboring surface element is chosen. Taking the value of the closest neighbor guarantees the continuation of equal layer height in overhang regions. To be able to accurately compute the layer height for the lowest isovalues, $\mathcal{L}_s$ is initialized with a model of the build plate in case of support surfaces. For the part surfaces, $\mathcal{L}_s$ is initialized as all support surfaces unified. This initialization is represented by the input of $\mathcal{L}_{init}$ in the algorithms.

Algorithm 2 Layer height computation

Require: $\mathcal{S}_h, \mathcal{L}_{init}$
  $\mathcal{L}_s \leftarrow \mathcal{L}_{init}$
  $d \leftarrow [\,]$
  for $\mathcal{T} \in \mathcal{S}_h$ **do**
    $d_h \leftarrow min\_cartesian\_distance(\mathcal{T}, \mathcal{L}_s)$
    $hits \leftarrow ray\_intersect(\mathcal{T}, \mathcal{L}_s)$
    $d_h[\neg hits] \leftarrow d_h[closest\_neighbors(\neg hits)]$
    $d \leftarrow append(d, d_h)$
    $\mathcal{L}_s \leftarrow merge(\mathcal{L}_s, \mathcal{T})$
  end for
  return d

Achieving a maximum distance of h_{max} through the method described above is part of the novel approach presented in this work. Ensuring that h_{max} is not exceeded by the slicing guarantees its compliance with the upper hardware limitation. On the other hand, this method almost always leads to the lower limit being exceeded, the remedy for which is the topic of the next section.

2.4.4. Intermediate layers

The generation of intermediate layers is proposed by Fang et al. in [11]. In their case, these surfaces fill gaps that result from taking the lower bound of the hardware as minimum distance, as this leads to the maximum distance being exceeded in many regions. They are generated by finding areas with too high distance and extracting new slices from the field to fill these areas.

As mentioned above, in this work a new method is presented that immediately creates slices that do not exceed the upper limit, but results in many slices whose distance is too small. In the following, the method by which this issue is solved is presented. Switching the order of operations in such a way can evidently result in a more even distribution of layer distances as the upper limit is taken into account from the beginning and not after the fact, which unnecessarily limits the range where intermediate layers can be inserted.

Cutting away the layers is done by iterating over all the layers while merging the already handled layers into one object $\mathcal{L}_c$ to check the distance to all lower layers concurrently. An example of this distance computation can be seen in Fig. 5 (a), where the distance to $\mathcal{L}_c$ is displayed in a color range. The iteration starts with $\mathcal{L}_c$ equal to the lowest layer. Every iteration, the triangles of the current isosurface $\mathcal{S}_h$ whose distance to $\mathcal{L}_c$ is smaller than h_{min} are marked, which can be seen in Fig. 5 (b). The process of cutting the layers by h_{min} is prone to small holes that propagate through the part, resulting in highly perforated slices and low quality parts. This can also be seen as small separate red spots in Fig. 5 (b). As a mitigation measure a method similar to the opening operator from morphology is applied. The opening operator first consists of applying erosion followed by dilation on the data. The erosion removes small holes but also decreases the outer boundary of the data. Applying dilation in the same extent expands the outer boundary near to its original state, but the holes stay closed. Exemplary results from the erosion and following dilation can be seen in Fig. 5 (c) and (d) respectively. For the marked triangles, the erosion is achieved by setting a triangles status to unmarked if not all neighboring triangles with a coincident node are marked. The dilation is then done in reverse, so if any triangle with coincident nodes is marked the current triangle is marked as well. These operations are successively carried out $n_{open} = \frac{1}{2} \cdot d_{hole}/l_{edge}$ times respectively, with l_{edge} being the average edge length in the current isosurface and d_{hole} being the desired minimum diameter of holes in the final slice. After all n_{open} operations, all marked triangles are removed from the surface mesh, resulting in the areas exceeding the hardware bound h_{min}, which can be seen in Fig. 5 (e) for the example with the *Toptopt* part. An important observation is, that deviations can propagate through the part, resulting in rifts or uneven splits. To mitigate these issues and promote slice integrity, every other layer has to be left out in a first pass. A second pass is then done to remove any further areas exceeding the lower limit for layer height. The exact part geometry is retained during this process. Areas where the surfaces are cut away are filled by the increased layer height of the other layers, so there are no subtractive changes to the geometry. At the same time the process only defines regions on the existing geometry to be cut away, avoiding deviations in an additive sense.

A step to ensure bed adhesion and printability is the insertion of a planar base layer at the height $h_{max} - h_{min}$. The material of the base layer is therefore printed directly on the build plate. All protruding support layers are then cut away at the height $h_{max} + h_{min}$, so that the support layers terminate right before reaching the base layer. This method drastically lowers the necessary buffer space h_{buff} which in turn reduces printing time and material waste, because the bottom support slices do not have to reach planarity but can start at an angle. The planar base layer is extended by a certain value, in order to create a brim, to increase bed adhesion.

2.5. Looping algorithm

On the freeform manifold surfaces resulting from all the computations described in the previous sections, a path for the nozzle has to be generated to be able to fabricate the part using FFF. This path usually consists of a number of outer contours with the rest of the slice being filled with a pattern which is referred to as infill. An example for such a path can be seen in Fig. 6 (f). Both the number of outer contours and infill pattern have an impact on the load capabilities of the manufactured part. The Looping algorithm presented in this paper represents the unification of these outer contours with a load-oriented infill pattern, that guarantees continuity for a path planned on a freeform surface.

Creating such load-oriented and continuous paths is achieved by constructing loops that follow the optimal material orientations and are connectable by a rerouting procedure that guarantees continuity. This includes the connection to the outer contours. The algorithm can be split up into the following 4 steps, which are executed for every surface in every slice:

1. Generate the outer contour parallel paths and extract the infill area.
2. Obtain the load-oriented loops.
3. Build the rerouting graph and convert it to a rerouting tree.
4. Reroute according to the rerouting tree to obtain a globally continuous path.

These steps are explained in detail in the following sections. Prior to these steps, the slice obtained by extracting the isosurface has to be split into its contiguous subsurfaces. For each of these contiguous surfaces an operator similar to the opening operator from morphology is applied again, which is described in more detail in [33]. This is done to remove thin sections, which would hinder the generation of a continuous path if their width is smaller than $2 \cdot w_{path}$ with w_{path} being the path spacing. At this point, excess material is created by the travel movements between the subsurfaces for branching geometries. Furthermore, the surfaces are remeshed by isotropic remeshing to avoid numerical problems, as the triangles created by the slicing operation can have arbitrarily small areas.

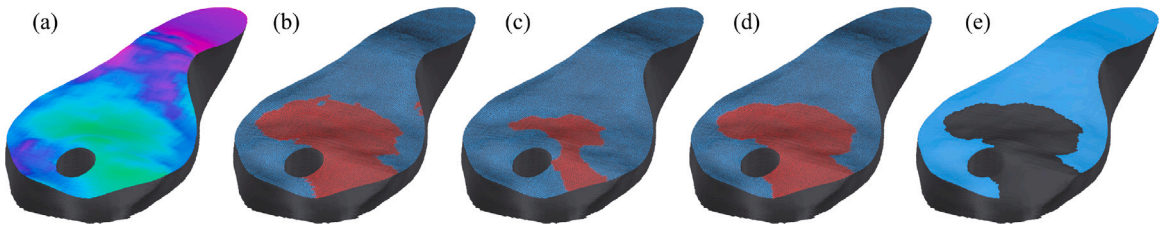


Fig. 5. Opening operator for cutting layers when adjusting layer height to be in the manufacturable range for the lower layers of the *Topopt* part. (a) Surface with distance to rest of the merged layers $\mathcal{L}_c$. Distance indicated by color, with increasing value from green to purple. Already printed layers $\mathcal{L}_c$ displayed in black. (b) Area under threshold to be cut away. (c) Surface with erosion applied to data. (d) Surface with opening operator applied. (e) Final cut surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.5.1. Outer contours

The outer contours form the shell of the part and constitute the visible surface. They have a significant impact on the load capabilities of the fabricated part, as most load cases result in high compressive or tensile stresses in these outer areas. The outer contours are obtained for a manifold triangulated surface S by computing the geodesic distance ϕ_{out} from the boundary of the surface ∂S and extracting isolines at values given by the number of outer contours n_{cont} and the path spacing w_{path}. The geodesic distance function ϕ is computed by the heat method [44]. This method solves the eikonal equation

$$|\nabla\phi| = 1 \quad with \quad \phi|_\gamma = 0\ . \tag{13}$$

With $\gamma = \partial S$ to compute ϕ_{out}, the isolines are then given by $\{c_i\}_{i=1,\dots,n_{cont}}$ with

$$c_i = \{x_p | \phi_{out}(x_p) = (i - 0.5) \cdot w_{path}\}. \tag{14}$$

If the surface has m holes, ∂S will have $m-1$ discontinuities, which separate the isocurves into contiguous subcurves $s_{i,j}$ with $i \in \{1, \dots, n_{cont}\}$, $j \in \{1, \dots, m\}$. A result for an example surface can be seen in Fig. 6 (a), with the value of ϕ_{out} being displayed as colors and the extracted outer contours in black. With these outer contours collected in the ordered set C, the infill area can be computed by cutting away the surface for $\phi_{out} \leq (n_{cont} + 0.5) \cdot w_{path}$ creating a new triangulated surface S_{infill}, that can be seen as the blue surface in Fig. 6 (b). With this operation it is possible that the contiguous surface S is split in multiple contiguous subsurfaces, that have to be isolated and for whom the infill generation algorithm described in the next section has to be executed individually.

2.5.2. Load-oriented loops

The central part of the presented algorithm is the novel infill pattern, guaranteeing continuity and load-orientation. The loops making up this infill pattern are later connected with each other and the outer contours, making up the path for the surface S_{infill}. Connecting contours and infill pattern is implemented using the first-in-spiral-out (FISO) algorithm, whose aim is achieving minimal curvature through connecting all loops as early as possible when traversing them. The initial step in order to obtain the loops is identical to the discontinuous load-oriented path planning as described in [11], with the difference of doubling the sampling interval to $2 \cdot w_{path}$. The first step of the loop generation is very similar to the slicing, but instead of W, V is used as the basis for computing the scalar field. V is then projected onto the surface. The scalar field ψ_{tri} is computed from these vectors by again solving the Poisson equation, this time for triangular meshes.

$$V^\perp = \nabla\psi_{tri} \ \Rightarrow\ \nabla \cdot V^\perp = \Delta\psi_{tri}\ , \quad V^\perp = V \times N\ . \tag{15}$$

The isolines of ψ_{tri} are then extracted at intervals of twice the desired path spacing as

$$l_i = \{x_p | \phi_{lin}(x_p) = (i - 0.5) \cdot 2w_{path}\} \tag{16}$$

with $i \in \{1, \dots, n_{load}\}$ and the number of discontinuities being $n_{load} = \lceil \frac{max(\psi_{tri}) - w_{path}}{w_{path}} \rceil$. These isolines form the ordered set L.

In the next step, the geodesic distance ϕ_{lin} from these lines on the domain of the surface S_{infill} is computed by again solving the eikonal equation from Eq. (13) with $\gamma = \cup\{l_i\}_{i=1,\dots,n_{load}}$. To allow for the loops to slightly protrude out of the infill area the computation is done on the surface S. Allowing a slight overlap of contour and infill can increase the adhesion between them and decrease the gap to the outer contours. The surface is then cut off at $2w_{path}$, to decrease the impact of numerical errors. The distance field on the cut off infill area is displayed in Fig. 6 (d). The subcontours of the isocontour at value χ from this new scalar field are in the form of loops if $\chi < \frac{1}{2}w_{path}$. These loops are the starting point for the rerouting procedure and are indexed by their corresponding isoline i and its subindex j as $o_{i,j}$ forming the ordered set O.

2.5.3. Rerouting graph

To be able to connect neighboring loops into a continuous path without self intersection, instruction on how to connect the loops have to be generated. These instructions exist in form of a tree T, where nodes represent the loops, subcurves or contours and the edges represent a connection that should be made during the rerouting procedure. This tree T is generated from a graph G that includes all possible connections, by extracting its minimum spanning tree. G is similar to a Reeb graph, which is utilized and further described in [45]. A Reeb graph is a topological graph, that represents the progression of the level sets of a topological object. Similarly, when neighboring isolines differ in their number of contiguous subcurves, this is reflected in a branching or merging of the Reeb-like graph G used in this work. To clear nomenclature, even though a Reeb graph has graph in its name, the object G and all subsequent mentions of graphs are of a different class of mathematical object, the abstract graphs from graph theory.

To create the Reeb-like graph, an iteration is done over the isolines with increasing isovalue. To find which subcurves of the next isoline to connect to the current subcurves, the shortest geodesic connection between the endpoints of each subcurve is computed. If ψ_{tri} is monotonically increasing along this geodesic, the nodes corresponding to these subcurves are connected in the graph. Checking the monotonic increase is done to prevent connecting subcurves that are separated by holes, which would cause the geodesic to wrap around the hole, resulting in it backtracking along ψ_{tri}. Merely inspecting proximity can lead to incorrect connections, which should be avoided. The graph generation procedure is shown in algorithm 3.

Before iterating over the previously generated isolines, the lines in the ordered set L are reoriented such that they have their endpoints closest to the boundary. In algorithm 3 the number of isolines is denoted with n_{iso} and the number of contiguous subcurves for the isoline with index i is denoted by $n_{sub}(i)$.

An example for a result of the generation of the Reeb-like graph is shown in Fig. 7 (a) for a surface with the isolines l_i and the corresponding Reeb-like graph. In order to connect the outer contours from the ordered set C to this newly generated graph, the respectively closest innermost outer contour from C is connected with every connected component from G. The edge weight for the connection is taken to

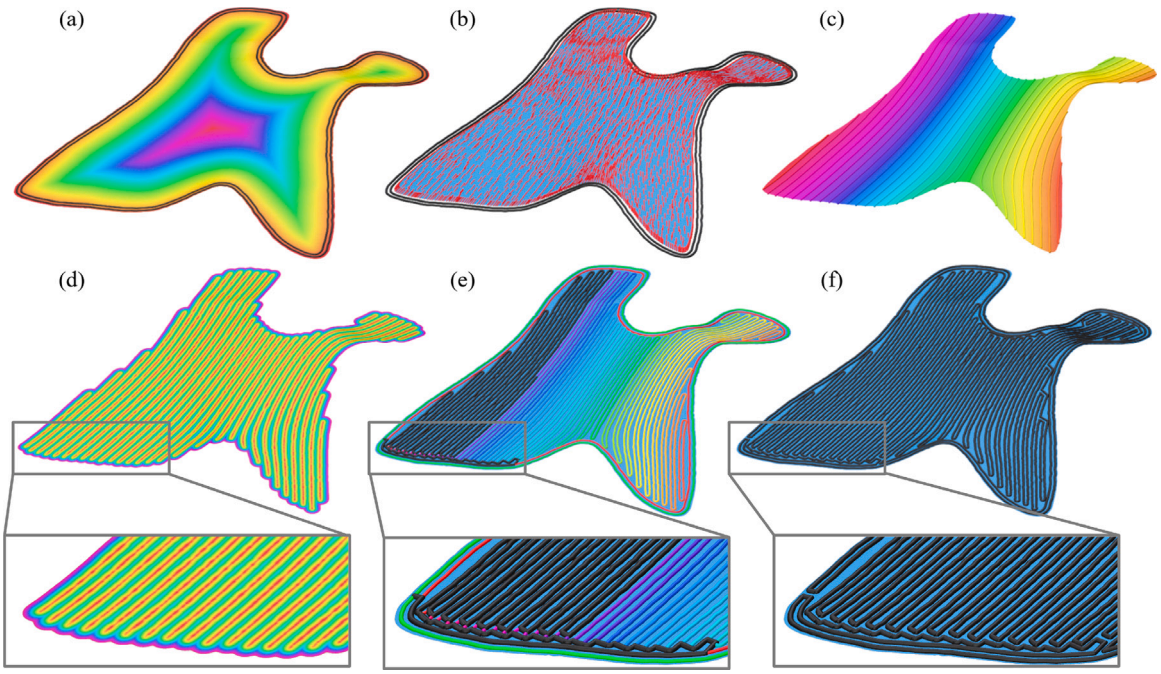


Fig. 6. Computation of the continuous path for an example surface 6 (a) Geodesic field ϕ_{out}, values displayed in color; outer contours displayed in black. (b) Outer contours in black, infill surface S_{infill} in blue, projected vector field V as red arrows. (c) Scalar field ψ_{tri} and isolines l_i on infill surface. (d) Geodesic distance ϕ_{iin} from isolines on cut off infill area. (e) Partial execution of the rerouting algorithm, colors indicating the isovalue of the loops to be connected to the continuous path. (f) Final continuous path formed by connecting the loops. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 3 Generate Reeb-like

Require: $L, S_{infill}, \psi_{tri}, n_{iso}, n_{sub}(i)$
  $G \leftarrow Graph()$
  for $i < n_{iso}$ **do**
    for $j < n_{sub}(i)$ **do**
      for $k < n_{sub}(i+1)$ **do**
        $l_1 \leftarrow L[i,j]$
        $l_2 \leftarrow L[i,k]$
        $g(t) \leftarrow geodesic_{S_{infill}}(l_1(0), l_2(0))$
        if $\{t | \frac{d}{dt}\psi_{tri}(g(t)) < 0\} \neq \emptyset$ **then**
          $G.add\_edge((i,j),(i,k), weight = 0)$
        end if
      end for
    end for
  end for

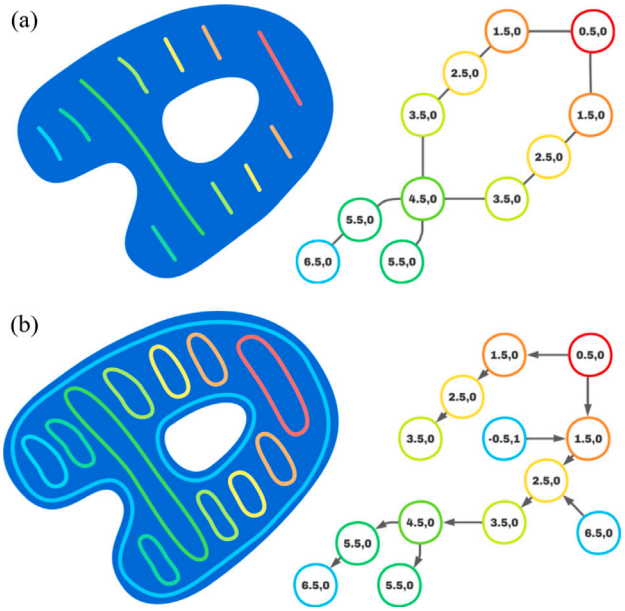


Fig. 7. Schematic of the creation of the rerouting tree. (a) Isocontours on the scalar field ψ_{tri} and the Reeb-like graph. (b) Loops, which are isocontours of the geodesic distance field ϕ to the isolines, the outer contours and the rerouting tree.

be the minimum distance. As a next and final step, the outer contours from C have to be connected among themselves, with the weight for the edge again being the minimal distance between contours. Similar to the FISO algorithm, only contours are connected that are neighbors in their isovalue. After all connections have been included in the graph, the minimum spanning tree of G is computed. The resulting tree contains only those connections, where rerouting is possible. Furthermore, the edges become directional, as a root is set when computing the rerouting tree. A result from the computation of a tree from the Reeb-like graph is illustrated in Fig. 7 (b). The loops in the ordered set O are shown on an example surface and the rerouting tree is shown with the nodes representing the outer contours added.

2.5.4. Rerouting procedure

With the loops in O, the outer contours in C, which are also in the form of contiguous loops, and the rerouting tree T, computing the continuous path is only a matter of interconnecting the elements from O and C. The loops O are broken and connected to C via two lines in the distance of w_{path}. The rerouting process is equivalent to the FISO-method with the difference of using the load-oriented loops instead of contour parallel loops. A detailed explanation can be found in [33]. In short, a recursive function is applied to iterate over the rerouting tree T, similar to depth-first search. As mentioned previously, edges in the rerouting tree represent a connection to be made by the rerouting procedure. Pairs of rerouting segments are found by traversing along

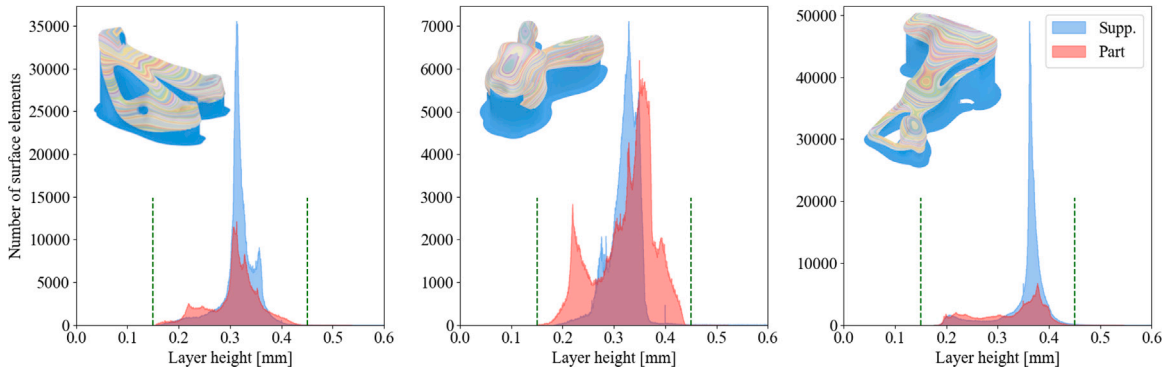


Fig. 8. Distribution of layer height for 0.6mm nozzle. Number of surface elements plotted with 1000 bins. Maximum and minimum layer height input by user displayed as green dotted lines. From left to right the three exemplary parts *Topopt*, *Bunny* and *Yoga*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the contours. The traversal always starts at the previous rerouting position. After ensuring the compatibility in length and orientation of these segments, the endpoints of the segments are joined. This forms two connections between the contours, one for traveling forth, one for coming back. After completing a full traversal of the rerouting tree, all loops and outer contours are connected for the current subsurface, forming the continuous path.

The procedures described above are done for every subsurface in every slice. These subpaths are then connected into a globally continuous path by connecting the subpaths with travel movements. Additionally, a line is extruded along the bounds of the part as a skirt. After these procedures, the final manufacturing instructions are generated by a post processor from the globally continuous path.

3. Results

3.1. Parts

To increase comparability, the parts from [11] are used for the experimental evaluation. To decrease manufacturing times and allow for accelerated testing, they are scaled by a factor of 0.5 in all dimensions. The three parts named *Topopt*, *Bunny* and *Yoga* are provided by the authors of [11] and exhibit a wide range of geometric properties, making them well suited for testing the algorithms proposed in this paper. They can be seen in Figs. 10 and 11. The load cases assumed in this work are matched to the ones displayed in [11]. As no information on the exact replication is provided, the load cases are deducted from the figures in the paper. For the *Topopt*, these are fixed boundary conditions in the bottom holes and load is applied along the axis of the single top hole. This results in tension along the frontal section and in compression along the rear struts. For both the *Bunny* and *Yoga* models, the load is applied in the downwards direction at the topmost surface of the part. For both models, fixed boundary conditions are employed at the bottom surface of the part, yet only restricting in the Z-Direction.

3.2. Computational results

The implementation is done using Python and could be successfully deployed on multiple machines. Numpy [46] and scipy [47] are used for mathematical computations, together with the VTK [48] wrapper pyvista [49] for simplicial complexes and nets. For the heat method potpourri3d [50] is used and networkx [51] for graph theory. Most of the figures in this paper are generated using polyscope [52]. For the computation of the Laplace–Beltrami operator in Section 2.4.3, the Lapy package for python [53] is used. The computational complexity for the slicing procedure could be severely reduced in comparison to [11] by eliminating the computation of critical regions with complexity $O(2^2)$. For path planning the complexity is not increased compared to the FISO algorithm as both use the same rerouting strategy which has the highest complexity due to its recursive nature. However, computation time will be negatively affected as the calculation of the loops is more time intensive than simply taking the isocontours of the geodesic field from the boundary as it is done in FISO.

Two of the biggest influencing factors on part quality are the layer height and unfilled or under-filled areas. Guidetti et al. [18] and Fang et al. [11] have both done investigation in this direction. In the following section these two properties are analyzed and compared. The layer height has a significant impact on printability and the surface quality of the part, while unfilled areas can lead to distinct losses in stiffness and strength as less material is deposited.

Starting with the layer height, uniformity tends to decrease manufacturing defects, yet full uniformity would require planar slicing. In conclusion, the limits of the hardware should be fully exploited, but never exceeded. In this way printability is achieved for the parts, while fully utilizing the permissible range of layer heights. Distance computation is done in the same way as explained in Section 2.4.3.

The results for the algorithms presented in this work can be seen in Fig. 8. Typical choices for the lower and upper limit for layer height are $0.25d$ and $0.75d$ respectively, with d being the nozzle diameter. These are indicated in Fig. 8 as dotted green lines. What can clearly be deduced is, that the method introduced in 2.3 to guarantee a maximum layer height of $0.75d = 0.45$ mm in this case has achieved the desired effect. The lower limit of $0.25d = 0.15$ mm is not exceeded either. Another observation is, that the large spikes indicate a layer height around which most values lie. This spike is suspected to be indicative of part quality, as less variation in layer height puts less strain on the hardware. However, if only a single spike would be observed there would be no variation in layer height, which would correspond to uniformity and limited adaptability to the load orientation field. Thus, the exploitation of the whole permissible range, seen in the graphs as the width of the distribution, confirms the utilization of load-oriented slicing.

The authors of [11] use a histogram of the number of waypoints for a range of layer heights to measure their distribution. As a result their data is highly dependent on point spacing and resampling uniformity. To avoid such a dependency, the values on the triangular surface elements are used in this work. The size of these elements is adequately uniform due to isometric remeshing, which can be seen for an example surface in Fig. 5 (b)–(d). Element size uniformity is assumed to have a positive impact on the data quality, as all areas of the surface are represented equally in the histograms. In direct comparison with Figure 12 from [11], one can see that both the limit of $0.75d$ set in this work and the limit of $0.8d$ set in their work is exceeded by their results.

Further analysis of the distribution, like the location of spikes or the uniformity in the interval $[0.25d, 0.75d]$ will not be included in this work as more evidence and investigation is required to draw meaningful conclusions.

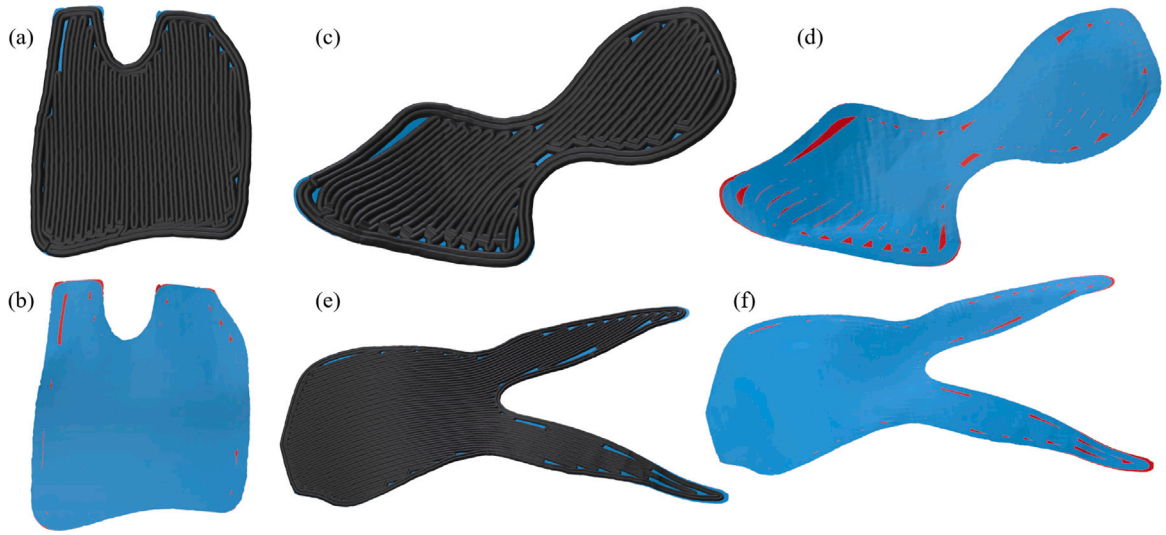


Fig. 9. Visualization of unfilled areas for example surfaces of all three example parts. Path displayed in black, unfilled areas in red. Path (a) and unfilled areas (b) for *Topopt* part; path (c) and unfilled areas (d) for *Yoga* part; path (e) and unfilled areas (f) for *Bunny* part. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Path spacing has been chosen as 0.65mm in this work to allow for a positive extrusion factor with a 0.6mm nozzle. Guidetti et al. [18] and Fang et al. [11] both compute and analyze the line spacing. In neither of the studies, a clear mathematical indication of the method by which the spacing is computed is provided. A simple approach that might have been used in the papers would be the computation of minimal distance to all neighboring path lines, excluding the immediately connected segments. The most important symptom of irregular line spacing is over- or under-filling. This effect cannot be adequately represented by the minimal distance computation, as under-filled areas would not be included. To be able to observe under-filling, the path-to-surface distance has to be computed instead of path-to-path. A visual representation of the path-to-surface distance can be seen in Fig. 9. The under-filled areas are shown in red. What can be seen is, that the path planning algorithm does result in some areas with clear under-filling.

The under-filled regions partly originate from the geodesic distance computation leading to the rounding of corners. Yet, they mainly result from the interplay of continuity and load-orientation. Similar to the slicing procedure, arbitrarily small path widths or intermediate path segments would be necessary to completely fill the surface. However, the first is not possible due to the limitations of the manufacturing system and the second would break the requirement of continuity, as singular lines would have to be placed without the possibility to close the loop. A possible solution would be to use a repulsive curves approach [54], which would be expected to slightly impede the degree of load-orientation, but significantly reduce under-filling. Additionally, varying the width of the deposited bead could reduce the voids. Varying the width could negatively effect the range of possible layer heights due layer height and line width being inversely related for a constant extrusion and movement speed. As the proposed path planning inside of S_{infill} can be viewed as an infill pattern, slight under-filling is not inherently problematic. For the manufacture of FRP parts these air pockets can lead to stress concentrations, which should definitely be avoided.

3.3. Physical fabrication

One method to check the performance of slicing and path planning is the fabrication of test specimen. Especially for additive manufacturing this is a highly appropriate measure, as parts can be produced quickly and cheaply. As mentioned in Section 2.1, PLA is used for the manufacture of physical specimen that are evaluated regarding the effect of the path planning. The ability for load-oriented continuous printing becomes increasingly important when manufacturing parts with fiber reinforced polymers, so an evaluation of the developed algorithms regarding these materials is planned in future work. The printing machine used for the physical fabrication is introduced in more detail in Section 3.3.1. Choosing the process parameters displayed in Table 1 is informed by standard parameters for printing PLA and PVA, with some adjustments to increase bed adhesion and part quality. For now, the lower and upper limits are derived from standard practice and previous observations while printing. A more in depth identification of more optimal printing parameters should be element of further study, especially for FRP. For the fabrication of these parts a 0.6mm nozzle is chosen as a conventional nozzle size combining manufacturing efficiency with high resolution and minimal oozing.

Table 1
Process parameters used for fabrication of all parts.

Path spacing	PLA	PVA	Bed	$[h_{min}, h_{max}]$	Max. print speed
0.65mm	195 °C	215 °C	70 °C	[0.15,0.45]	1000 $\frac{mm}{min}$

The parts themselves have been published by the authors of [11]. All parts that are publicly available were manufactured on the printing machine. One specimen for each printed geometry before and after support removal is shown in Fig. 10. A first observation is, that the manufacturing instructions generated with the presented method could be verified to yield printable parts. Furthermore, the PVA support material can be removed without residue, even for undercuts. The dynamic layer height variation could be realized in the printing process without producing major over- or under-filling. When inspecting the surface quality, the stair step effect can clearly be seen. This can in part be attributed to load-orientation being the focus in this work and not surface quality like other non-planar methods. Additionally, the manufacturing system, like other 5-axis printing system, exhibits lower stiffness and an increased backlash due to the higher number of axes. Other visible defects are in the rear strut in the *Topopt* part, where the layers seem to have imperfect bonding and the clearly visible missing arm part in the *Yoga* model, where the support material was removed during generation, resulting in the material not being deposited correctly. The former error can be fixed by decreasing the layer height and the latter one can be fixed by choosing a smaller nozzle. Another observation is, that clearly less support material is needed when comparing to Figure 14 in [11].

Fig. 10. Results of the physical fabrication using a multi-axis 3D-printer. From left to right with and without support material: *Topopt, Bunny, Yoga*.

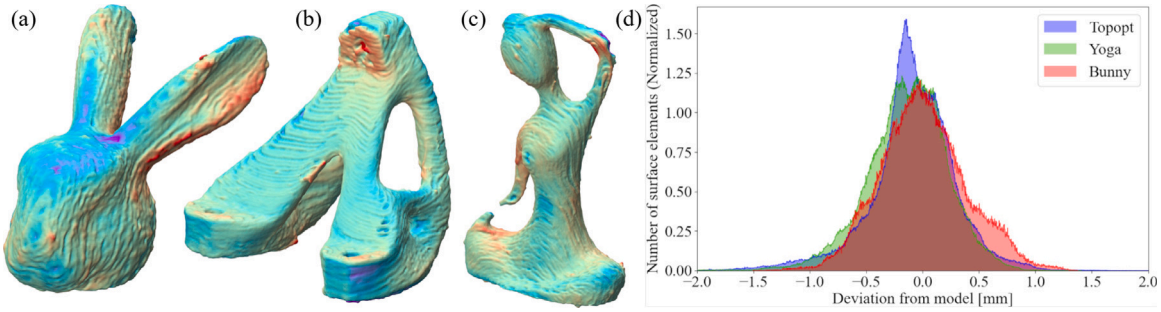


Fig. 11. Evaluation of geometrical accuracy. (a-c) Deviation from the target model on scans of the printed parts. (d) Histogram of the surface elements by deviation from the target geometry.

Table 2
Part properties and user-defined parameters used for fabrication of the different parts with looping algorithm.

Part	Dim. [mm]	Vol. [ml]	# Tet	Time
Topopt	61 x 54 x 49	22.76	70 505	7.2h
Bunny	41 x 61 x 63	18.4	60 375	5.1h
Yoga	52 x 31 x 73	17.13	52 446	8.2h

The parts' properties, parameters and print times are shown in Table 2. It is difficult to draw conclusions about the different print times, as the parts were scaled by a factor of 0.5, although the nozzle diameter was decreased to 0.6mm from 1mm in comparison to [11] as well. The difference in print times can also be attributed to the difference in support geometry, slicing and printing machine.

To test to which degree the proposed method is capable of producing geometrically accurate parts, the EinScan-SP 3D-Scanner is used to obtain a detailed scan of the printed parts. After meshing, the scanned STL is aligned with the target geometry mesh by use of the iterative closest point method. For each surface element of the scanned mesh, the smallest distance to the target geometry is computed. The results of this computation can be seen in Fig. 11. On the left side, the colors indicate the deviation in relation to the geometry. What is clearly visible here is the stair-step effect and some general deformation, that could be due to thermal distortion or the connection between the support and the part. On the right side, the deviation from the target geometry is shown as a histogram, normalized to represent a density by dividing by the total number of elements. The bulk of deviations are under 1 mm, this result is to be further improved by decreasing backlash in the printing machine and better temperature management. Choosing a smaller nozzle could additionally reduce the spillage and formation of excess material that seems to be the cause of some of the larger deviations.

3.3.1. 3D printing machine

The system used to manufacture the parts is derived, presented and analyzed in [55]. In the following, the methodical derivation is given and further details are provided. As can be seen in Fig. 12 the printer uses a robotic manipulator to move the build plate in 6 degrees of freedom. The stationary print heads allow for fast switching, constant gravitational direction and reduced complexity of wiring and material delivery. The material output is synchronized in real time with the built plate's movement through a high speed programmable logic controller incorporated into the robot controller in conjunction with a stepper motor driver. The inverse kinematics are computed online on the robot controller with an advanced NC-Kernel. Singularity and collision avoidance, a major challenge for robotic AM, is guaranteed up to high nozzle tilt angles through the mathematically derived design and layout of the machine. [55]. The instructions are read in GCode, which are generated by a post processor of the slicer presented in this work. A simulation integrated into the software framework ensures the ability to inspect the robot movements prior to fabrication.

3.4. Experimental results

The purpose of the experimental evaluation is to further validate the developed path planning algorithm. The performance of parts printed with the developed Looping algorithm is compared to parts using a classical FISO approach with the same slices as basis for the path planning and the optimized planar orientation from [11], where the path planning was chosen as FISO as well and layer height was set as 0.35mm. The slicing direction is directly matched to the results titled "optimized" from [11] which is also used in this work to refer to this orientation. The tests serve only as an initial validation. A quantitative in-depth analysis will be the topic of further research. For every algorithm three specimen where fabricated. The part properties after fabrication can be seen in Table 3. The masses where determined for the dry parts, as after the removal of the support material by submersion the parts absorb water which has to be removed. One thing to note is the lower mass of the third Looping specimen, which did affect the performance of the part. The cause of this deviation cannot be determined by optical inspection of the part. The mean density is determined by division of the mean mass of the specimen by the target volume. What can be seen is that the densities of the parts are roughly consistent over the algorithms, with the density of the planar specimen being slightly lower than that of the non-planar parts. The deviation is believed to be due to the lag or imperfect tuning in the controller that deals with slowing the material flow during reorientation movements. If this were the case, it would results in more

Table 3
Part properties after fabrication for the Topopt part, for the different algorithms. Every Algorithm was tested three times.

Algorithm	Masses	SD Mass	Mean density	Time	Strength	Stiffness
Looping	[23.4 g, 24.7 g, 22.7g]	1.08 g	1.05 [g/cm^3]	7.2 h	1561.5 N	234.66 N/mm
FISO	[23.8 g, 23.9 g, 24.0g]	0.1 g	1.05 [g/cm^3]	7.32h	1314.8 N	200.88 N/mm
Opt. planar	[23.5 g, 23.4 g, 23.3g]	0.1 g	1.03 [g/cm^3]	5.27 h	2326.3 N	382.31 N/mm

Fig. 12. Robotic multi-axis printing machine used in this work for the physical fabrication of the parts. Further detail in [55].

material being deposited for the non-planar methods. Comparing the densities of the manufactured parts to the density of the filament which is 1.22 [g/cm^3], a clear reduction in density due to the voids in the parts can be noted. The manufacturing times seen in Table 3 show the increase due to reorientation measures in manufacturing time when employing the non-planar slices.

The tests are carried out using a ZwickRoell Z010 universal testing machine. The printed, cleaned and dried parts are mounted to a fixture as seen in Fig. 13. The mounting and connection to the moving part of the testing machine replicate the load case from [11], which is also used in the FEA and described in more detail in Section 3.1. The parts are then deformed at a constant speed of 0.4 mm/s. The traveled distance and force are continuously recorded during the test. The deformation is stopped after failure of the parts, which can be easily identified both visually and from the measured force, because of the brittle failure mode.

As the experimental evaluation is not the focus of this work the sample size and experimental design matrix are kept small. Only the *Topopt* part is investigated. For the Looping, FISO and Planar parts three specimen are manufactured and tested.

The resulting force–displacement curve can be seen in Fig. 13. The kink in the graph is most likely a result of the final setting of the part on the mounting surface. Apart from that, all parts display an almost purely elastic deformation followed by a brittle failure of the complete structure.

The measured mean maximum force of the FISO parts is 1314.8 N. The one of the Looping parts lies 18.76 % higher at 1561.5 N and

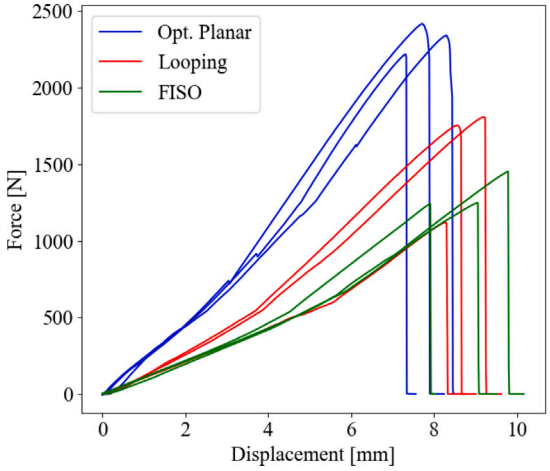


Fig. 13. Tensile tests of the *Topopt* example part to compare the developed slicing and path planning algorithm with a contour parallel algorithm and planar slicing. Mean strength for Planar 2326.3N, Looping 1561.5N and FISO 1314.8N. Increase of almost 20% for the Looping path planning, with planar slicing still achieving the highest stiffness and strength.

planar is the highest with 2326.3 N. The mean measured slope after the kink is calculated to be 382.31 N/mm for the planar parts, 234.66 N/mm for the Looping parts, 16.82 % higher than the 200.88 N/mm of the FISO parts. Therefore, the mean mechanical performance of the parts produced using the Looping algorithm exceeds that of the FISO parts both in stiffness and strength for the given load case. Yet, the performance of the parts with planar slicing still exceeds that of the non-planar parts.

The specimen after testing can be seen in Fig. 14. The Looping parts consistently failed around the mounting hole with a mostly interlaminar failure mode. For the FISO parts the failure consistently occurred slightly lower in the neck section of the part with an intralaminar failure. The first two parts manufactured with the optimized planar slicing failed in the truss sections, with the last specimen breaking around the lower mounting holes. For all planar parts intralaminar failure can be observed.

All of the obtained data for the analysis of layer height, unfilled areas and the experimental evaluation on the testing machine is provided in the supplementary material. Furthermore, the scanned and target geometries for the evaluation of geometrical accuracy are provided as well.

4. Discussion and conclusion

In this work, a contiguous method for generating manufacturing instructions for the additive manufacture of load-oriented parts is presented. By starting with the finite-element-analysis and ending with the actual fabrication step, a seamless coverage of all steps is guaranteed. A focus was the balance of load-orientation and the ability to actually manufacture the parts. With its future potential in mind, this balance includes the specific requirements of printing fiber reinforced polymers. The major improvements over previous works include:

- Full continuity in path planning together with load-orientation removing the necessity of a cutting apparatus when printing FRPs.

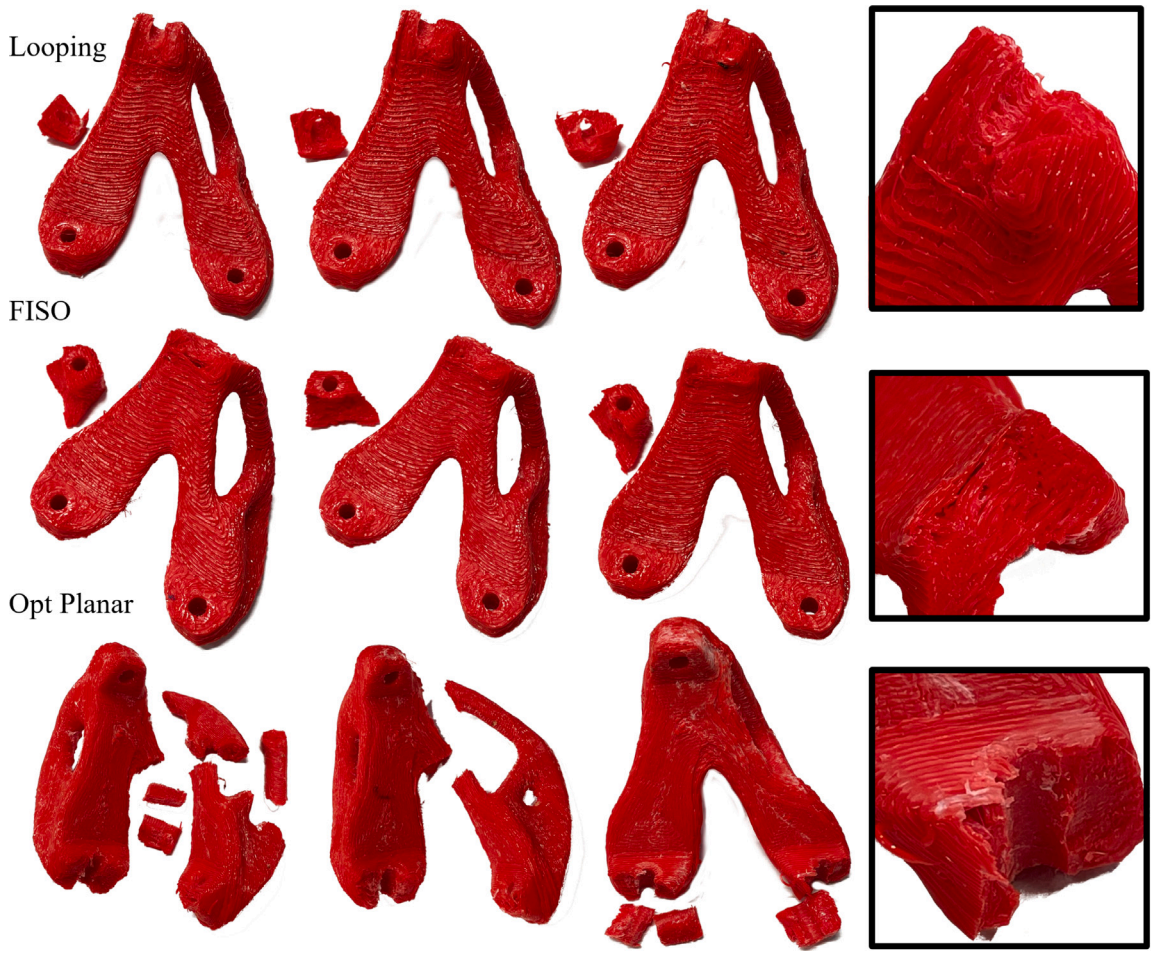


Fig. 14. Fracture surfaces after the tensile tests of the *Topopt* example part. Top row: Looping specimen. Middle row: FISO specimen. Bottom row: Optimized planar specimen. On the right the fracture surfaces are shown in more detail.

- Removing the complex vector field homogenization steps through careful choice of which fields to use from the FEA.
- Improved guiding field calculation with the utilization of the cross product to generate the field from the optimal material orientations.
- Field weighting scaled by the influence of elements in the load analysis.
- Improved support geometry generation and intermediate layer generation.

These improvements are implemented and the method was evaluated in the scope of this work regarding computational results and physical fabrication.

The experimental results showed an increase in strength when comparing the Looping and FISO algorithm. This is believed to be due to the effectiveness of load-orientation in path planning, as the Looping algorithm lead to interlaminar failure and the FISO algorithm resulted in intralaminar failure for the same slices. A lower strength was achieved with non-planar slicing in comparison to planar slicing in an optimized direction. Clear superiority of non-planar printing has been demonstrated before [11]. The part used was geometrically identical up to scaling to the part investigated in Section 3.4. The results are difficult to compare as the strength was not mass normalized. A more recent publication showed superiority over planar printing for the same geometry, but did not consider its optimal planar orientation [21]. Comparing their non-planar, fiber reinforced part with the optimized planar neat PLA part investigated in this work, the latter shows an almost three times (2.96 ×) higher specific strength. Although some variation is expected due to the type of PLA, machine and specific print parameters these results seem to be at least not clearly implying a general superiority of non-planar over optimized planar printing. Another recent work showed the superiority of 2.5D parts printed on a conventional printer over non-planar printing on a 5-axis machine for PLA [18]. The current inferiority of the non-planar parts could be attributed to process induced imperfections, as they can lead to over-filling, voids and less homogeneous material distribution in the part in general. Therefore, to utilize the potential of non-planar load-optimized slicing and path planning further improvements of the manufacturing process will be necessary in the aspects of flow-rate control, reorientation movements and speed control. Further investigation will be required to find out why the unambiguous and substantial improvements by non-planar slicing reported in [11] could not be reproduced. The observed lower strength of the Looping algorithm compared to the planar algorithm is furthermore believed to be affected by the gaps discussed in making up a large part of the total slice area in the narrow parts around the top mounting holes where the Looping parts failed. For the planar parts, the simultaneous failure at several points speaks for a homogeneous stress distribution achieved by the topology optimization and better material distribution in the part.

Limitations to the presented method include the increased manufacturing time due to the complex motions necessary during fabrication and the pronounced stair-step effect when aligning the slices with the optimal material orientations. Additionally, supporting structures are necessary when applying the presented method. These currently require an additional print head, as water soluble material is used. Varying the layer height results in an inhomogeneous fiber volume fraction when

using FRPs, which increases the difficulty of modeling the part and predicting failure. However, this effect has been reduced in comparison with the state of the art where the fibers are often deposited even more irregularly.

Future work should cover three main objectives. Firstly, improvements in the ability to close gaps that remain after the path planning step of the algorithm presented in this work. One possible solution could be adjusting the extrusion width adaptively. Alternatively an approach similar to [54] could be employed, where an energy is minimized to move the already present path into the gaps. Secondly, to decrease the manufacturing time, the support material could be printed using a partial infill, which would require an infill pattern for non-planar slicing, that is compatible with the slicing method presented in this work. Lastly, the focus of future work should lie on applying the developed algorithms to fiber reinforced polymers. To this end, studies into structural optimization, and manufacturability challenges like layer height variation and the hardware limits of fiber coextrusion should be done.

In conclusion it can be said, that the proposed continuous path planning method shows great potential for the load-oriented manufacture of parts with anisotropic materials. Improvements in the computation and layer height distribution are achieved by the developed method. An increase in load bearing capabilities is shown by the experiments for the path planning. However, the optimized planar specimen still show a higher stiffness and strength than the non-planar parts. This is believed to have its cause in process induced imperfections, which should be improved in future works. The continuity of the algorithm and the increased influence of non-planar slicing for a higher degree of anisotropy positions the developed method as a promising approach for manufacturing parts from fiber reinforced polymers.

Funding

This research was funded through the I3 program (Interdisciplinary, Innovative, Engineering (german: Ingenieurwissenschaften)) of the Hamburg University of Technology.

CRediT authorship contribution statement

Johann Kipping: Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Doran Nettig:** Writing – review & editing, Methodology, Investigation, Formal analysis. **Thorsten Schüppstuhl:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data acquired in this work is provided in the supplementary material.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.addma.2024.104426.

References

[1] Ian Gibson, David W. Rosen, Brent Stucker, Mahyar Khorasani, David Rosen, Brent Stucker, Mahyar Khorasani, Additive Manufacturing Technologies, vol. 17, Springer, 2021.

[2] Samia Adil, Ismail Lazoglu, A review on additive manufacturing of carbon fiber-reinforced polymers: Current methods, materials, mechanical properties, applications and challenges, J. Appl. Polym. Sci. (ISSN: 0021-8995) 7 (2023) e53476, http://dx.doi.org/10.1002/app.53476.

[3] Chelsea S. Davis, Kaitlyn E. Hillgartner, Seung Hoon Han, Jonathan E. Seppala, Mechanical strength of welding zones produced by polymer extrusion additive manufacturing, Addit. Manuf. (ISSN: 2214-8604) 16 (2017) 162–166, http://dx.doi.org/10.1016/j.addma.2017.06.006, URL https://www.sciencedirect.com/science/article/pii/S2214860416303116.

[4] Xia Gao, Shunxin Qi, Xiao Kuang, Yunlan Su, Jing Li, Dujin Wang, Fused filament fabrication of polymer materials: A review of interlayer bond, Addit. Manuf. (ISSN: 2214-8604) 37 (2021) 101658, http://dx.doi.org/10.1016/j.addma.2020.101658, URL https://www.sciencedirect.com/science/article/pii/S2214860420310307.

[5] Ziyang Zhang, Denizhan Yavas, Qingyang Liu, Dazhong Wu, Effect of build orientation and raster pattern on the fracture behavior of carbon fiber reinforced polymer composites fabricated by additive manufacturing, Addit. Manuf. (ISSN: 2214-8604) 47 (2021) 102204, http://dx.doi.org/10.1016/j.addma.2021.102204, URL https://www.sciencedirect.com/science/article/pii/S2214860421003651.

[6] Fuda Ning, Weilong Cong, Jingjing Qiu, Junhua Wei, Shiren Wang, Additive manufacturing of carbon fiber reinforced thermoplastic composites using fused deposition modeling, Composites B (ISSN: 1359-8368) 80 (2015) 369–378, http://dx.doi.org/10.1016/j.compositesb.2015.06.013, URL https://www.sciencedirect.com/science/article/pii/S1359836815003777.

[7] Ka Zhang, Wuxiang Zhang, Xilun Ding, Multi-axis additive manufacturing process for continuous fibre reinforced composite parts, Procedia CIRP (ISSN: 22128271) 85 (34) (2019) 114–120, http://dx.doi.org/10.1016/j.procir.2019.09.022.

[8] Pedram Parandoush, Dong Lin, A review on additive manufacturing of polymer-fiber composites, Compos. Struct. (ISSN: 0263-8223) 182 (2017) 36–53, http://dx.doi.org/10.1016/j.compstruct.2017.08.088, URL https://www.sciencedirect.com/science/article/pii/S0263822316329063.

[9] Pooyan Nayyeri, Kourosh Zareinia, Habiba Bougherara, Planar and nonplanar slicing algorithms for fused deposition modeling technology: a critical review, Int. J. Adv. Manuf. Technol. (ISSN: 1433-3015) 119 (5) (2022) 2785–2810, http://dx.doi.org/10.1007/s00170-021-08347-x.

[10] Dustin R. Jantos, Klaus Hackl, Philipp Junker, Topology optimization with anisotropic materials, including a filter to smooth fiber pathways, Struct. Multidiscip. Optim. (ISSN: 1615-1488) 61 (5) (2020) 2135–2154, http://dx.doi.org/10.1007/s00158-019-02461-x.

[11] Guoxin Fang, Tianyu Zhang, Sikai Zhong, Xiangjia Chen, Zichun Zhong, Charlie C.L. Wang, Reinforced FDM: Multi-axis filament alignment with controlled anisotropic strength, ACM Trans. Graph. (ISSN: 0730-0301) 39 (6) (2020) http://dx.doi.org/10.1145/3414685.3417834.

[12] Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, Charlie C.L. Wang, S3-slicer: A general slicing framework for multi-axis 3D printing, ACM Trans. Graph. (ISSN: 0730-0301) 41 (6) (2022) http://dx.doi.org/10.1145/3550454.3555516.

[13] Yi-Rong Luo, Robert Hewson, Matthew Santer, Spatially optimised fibre-reinforced composites with isosurface-controlled additive manufacturing constraints, Struct. Multidiscip. Optim. (ISSN: 1615-1488) 66 (6) (2023) 130, http://dx.doi.org/10.1007/s00158-023-03586-w.

[14] Chengkai Dai, Charlie C.L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, Yong-Jin Liu, Support-free volume printing by multi-axis motion, ACM Trans. Graph. (ISSN: 0730-0301) 37 (4) (2018) http://dx.doi.org/10.1145/3197517.3201342.

[15] Yujie Shan, Dongming Gan, Huachao Mao, Curved layer slicing based on isothermal surface, Procedia Manuf. (ISSN: 2351-9789) 53 (2021) 484–491, http://dx.doi.org/10.1016/j.promfg.2021.06.081, URL https://www.sciencedirect.com/science/article/pii/S2351978921001025.

[16] Yujie Shan, Yiyang Shui, Junyu Hua, Huachao Mao, Additive manufacturing of non-planar layers using isothermal surface slicing, J. Manuf. Process. (ISSN: 1526-6125) 86 (2023) 326–335, http://dx.doi.org/10.1016/j.jmapro.2022.12.054, URL https://www.sciencedirect.com/science/article/pii/S1526612522009173.

[17] Yamin Li, Dong He, Xiangyu Wang, Kai Tang, Geodesic distance field-based curved layer volume decomposition for multi-axis support-free printing, 2020, arXiv:2003.05938.

[18] Xavier Guidetti, Efe C. Balta, Yannick Nagel, Hang Yin, Alisa Rupenyan, John Lygeros, Stress flow guided non-planar print trajectory optimization for additive manufacturing of anisotropic polymers, Addit. Manuf. (ISSN: 2214-8604) 72 (2023) 103628, http://dx.doi.org/10.1016/j.addma.2023.103628, URL https://www.sciencedirect.com/science/article/pii/S2214860423002415.

[19] Johann Kipping, Thorsten Schüppstuhl, Load-oriented nonplanar additive manufacturing method for optimized continuous carbon fiber parts, Materials (ISSN: 1996-1944) 16 (3) (2023) http://dx.doi.org/10.3390/ma16030998, URL https://www.mdpi.com/1996-1944/16/3/998.

[20] Yamin Li, Dong He, Shangqin Yuan, Kai Tang, Jihong Zhu, Vector field-based curved layer slicing and path planning for multi-axis printing, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 77 (2022) 102362, http://dx.doi.org/10.1016/j.rcim.2022.102362, URL https://www.sciencedirect.com/science/article/pii/S0736584522000503.

[21] Guoxin Fang, Tianyu Zhang, Yuming Huang, Zhizhou Zhang, Kunal Masania, Charlie C.L. Wang, Exceptional mechanical performance by spatial printing with continuous fiber, 2023, arXiv:2311.17265.

[22] Xiangjia Chen, Guoxin Fang, Wei-Hsin Liao, Charlie C.L. Wang, Field-based toolpath generation for 3D printing continuous fibre reinforced thermoplastic composites, Addit. Manuf. (ISSN: 2214-8604) 49 (2022) 102470, http://dx.doi.org/10.1016/j.addma.2021.102470, URL https://www.sciencedirect.com/science/article/pii/S2214860421006205.

[23] Tong Liu, Shangqin Yuan, Yaohui Wang, Yi Xiong, Jihong Zhu, Lu Lu, Yunlong Tang, Stress-driven infill mapping for 3D-printed continuous fiber composite with tunable infill density and morphology, Addit. Manuf. (ISSN: 2214-8604) 62 (2023) 103374, http://dx.doi.org/10.1016/j.addma.2022.103374, URL https://www.sciencedirect.com/science/article/pii/S2214860422007631.

[24] Lingwei Xia, Sen Lin, Guowei Ma, Stress-based tool-path planning methodology for fused filament fabrication, Addit. Manuf. (ISSN: 2214-8604) 32 (2020) 101020, http://dx.doi.org/10.1016/j.addma.2019.101020, URL https://www.sciencedirect.com/science/article/pii/S2214860419315659.

[25] Ting Wang, Nanya Li, Guido Link, John Jelonnek, Jürgen Fleischer, Jörg Dittus, Daniel Kupzik, Load-dependent path planning method for 3D printing of continuous fiber reinforced plastics, Composites A (ISSN: 1359-835X) 140 (2021) 106181, http://dx.doi.org/10.1016/j.compositesa.2020.106181, URL https://www.sciencedirect.com/science/article/pii/S1359835X2030419X.

[26] Markforged, Industrial 3D printing, 2023, URL https://markforged.com/.

[27] Anisoprint, Continuous fiber 3D printing for manufacturing of optimal composites, 2023, URL https://anisoprint.com/.

[28] L.G. Blok, M.L. Longana, H. Yu, B.K.S. Woods, An investigation into 3D printing of fibre reinforced thermoplastic composites, Addit. Manuf. (ISSN: 2214-8604) 22 (2018) 176–186, http://dx.doi.org/10.1016/j.addma.2018.04.039, URL https://www.sciencedirect.com/science/article/pii/S2214860417305687.

[29] Yaohui Wang, Guoquan Zhang, Huilin Ren, Guang Liu, Yi Xiong, Fabrication strategy for joints in 3D printed continuous fiber reinforced composite lattice structures, Compos. Commun. (ISSN: 2452-2139) 30 (2022) 101080, http://dx.doi.org/10.1016/j.coco.2022.101080, URL https://www.sciencedirect.com/science/article/pii/S2452213922000250.

[30] Guoquan Zhang, Yaohui Wang, Jian He, Yi Xiong, A graph-based path planning method for additive manufacturing of continuous fiber-reinforced planar thin-walled cellular structures, Rapid Prototyp. J. (ISSN: 1355-2546) 29 (2) (2023) 344–353, http://dx.doi.org/10.1108/RPJ-01-2022-0027.

[31] Esther M. Arkin, Sándor P. Fekete, Joseph S.B. Mitchell, Approximation algorithms for lawn mowing and milling, Comput. Geom. (ISSN: 0925-7721) 17 (1) (2000) 25–50, http://dx.doi.org/10.1016/S0925-7721(00)00015-8, URL https://www.sciencedirect.com/science/article/pii/S0925772100000158.

[32] Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, Baoquan Chen, Connected fermat spirals for layered fabrication, ACM Trans. Graph. (ISSN: 0730-0301) 35 (4) (2016) http://dx.doi.org/10.1145/2897824.2925958.

[33] Johann Kipping, Zsolt Kállai, Thorsten Schüppstuhl, A set of novel procedures for carbon fiber reinforcement on complex curved surfaces using multi axis additive manufacturing, Appl. Sci. (ISSN: 2076-3417) 12 (12) (2022) http://dx.doi.org/10.3390/app12125819.

[34] Alexios Papacharalampopoulos, Harry Bikas, Panagiotis Stavropoulos, Path planning for the infill of 3D printed parts utilizing Hilbert curves, Procedia Manuf. (ISSN: 2351-9789) 21 (2018) 757–764, http://dx.doi.org/10.1016/j.promfg.2018.02.181, URL https://www.sciencedirect.com/science/article/pii/S235197891830221X.

[35] Adrien Bedel, Yoann Coudert-Osmont, Jonàs Martínez, Rahnuma Islam Nishat, Sue Whitesides, Sylvain Lefebvre, Closed space-filling curves with controlled orientation for 3D printing, Comput. Graph. Forum (2022) http://dx.doi.org/10.1111/cgf.14488, URL https://inria.hal.science/hal-03185200.

[36] Kohei Yamamoto, Jose Victorio Salazar Luces, Keiichi Shirasu, Yamato Hoshikawa, Tomonaga Okabe, Yasuhisa Hirata, A novel single-stroke path planning algorithm for 3D printers using continuous carbon fiber reinforced thermoplastics, Addit. Manuf. (ISSN: 2214-8604) 55 (2022) 102816, http://dx.doi.org/10.1016/j.addma.2022.102816, URL https://www.sciencedirect.com/science/article/pii/S2214860422002160.

[37] Yuming Huang, Guoxin Fang, Tianyu Zhang, Charlie C.L. Wang, Turning-angle optimized printing path of continuous carbon fiber for cellular structures, Addit. Manuf. (ISSN: 2214-8604) 68 (2023) 103501, http://dx.doi.org/10.1016/j.addma.2023.103501, URL https://www.sciencedirect.com/science/article/pii/S2214860423001148.

[38] Yuan Jin, Yong He, Guoqiang Fu, Aibing Zhang, Jianke Du, A non-retraction path planning approach for extrusion-based additive manufacturing, Robot. Comput.-Integr. Manuf. (ISSN: 0736-5845) 48 (2017) 132–144, http://dx.doi.org/10.1016/j.rcim.2017.03.008, URL https://www.sciencedirect.com/science/article/pii/S0736584516302654.

[39] Yamin Li, Shangqin Yuan, Weihong Zhang, Jihong Zhu, A new continuous printing path planning method for gradient honeycomb infill structures, Int. J. Adv. Manuf. Technol. (ISSN: 1433-3015) (2023) http://dx.doi.org/10.1007/s00170-023-11065-1.

[40] Chanyeol Yoo, Samuel Lensgraf, Robert Fitch, Lee M. Clemon, Ramgopal Mettu, Toward optimal FDM toolpath planning with Monte Carlo tree search, 2020, arXiv:2002.01631.

[41] Fanchao Zhong, Yonglai Xu, Haisen Zhao, Lin Lu, As-continuous-as-possible extrusion-based fabrication of surface models, ACM Trans. Graph. (ISSN: 0730-0301) 42 (3) (2023) http://dx.doi.org/10.1145/3575859.

[42] Masahito Ueda, Yunosuke Tasaki, Chikara Kawamura, Kenji Nishida, Masanori Honda, Koichi Hattori, Toshiaki Miyanaga, Tetsuya Sugiyama, Estimation of axial compressive strength of unidirectional carbon fiber reinforced plastic considering local fiber kinking, Composites C (ISSN: 2666-6820) 6 (2021) 100180, http://dx.doi.org/10.1016/j.jcomc.2021.100180, URL https://www.sciencedirect.com/science/article/pii/S266668202100075X.

[43] Ray W. Ogden, Non-Linear Elastic Deformations, Ellis Horwood Limited, 1984.

[44] Keenan Crane, Clarisse Weischedel, Max Wardetzky, The heat method for distance computation, Commun. ACM (ISSN: 0001-0782) 60 (11) (2017) 90–99, http://dx.doi.org/10.1145/3131280, URL http://doi.acm.org/10.1145/3131280.

[45] Ioanna Mitropoulou, Mathias Bernhard, Benjamin Dillenburger, Nonplanar 3D printing of bifurcating forms, 3D Print. Addit. Manuf. (ISSN: 2329-7662) 9 (3) (2021) 189–202, http://dx.doi.org/10.1089/3dp.2021.0023.

[46] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, Travis E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.

[47] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R.J. Nelson, Eric Jones, Robert Kern, Eric Larson, C.J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.

[48] Will Schroeder, Ken Martin, Bill Lorensen, The Visualization Toolkit–An Object-Oriented Approach to 3D Graphics, fourth ed., Kitware, Inc., 2006.

[49] C. Bane Sullivan, Alexander Kaszynski, PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK), J. Open Source Softw. 4 (37) (2019) 1450, http://dx.doi.org/10.21105/joss.01450.

[50] Nicholas Sharp, Potpourri3d, 2022, https://github.com/nmwsharp/potpourri3d (Accessed: 2022-05-02).

[51] Aric A. Hagberg, Daniel A. Schult, Pieter J. Swart, Exploring network structure, dynamics, and function using networkx, in: Proceedings of the 7th Python in Science Conference (SciPy2008), 2008, pp. 11–15.

[52] Nicholas Sharp, et al., Polyscope, 2019, www.polyscope.run.

[53] Martin Reuter, Franz-Erich Wolter, Niklas Peinecke, Laplace–Beltrami spectra as 'Shape-DNA' of surfaces and solids, Comput. Aided Des. (ISSN: 0010-4485) 38 (4) (2006) 342–366, http://dx.doi.org/10.1016/j.cad.2005.10.011, Symposium on Solid and Physical Modeling 2005. URL https://www.sciencedirect.com/science/article/pii/S0010448505001867.

[54] Chris Yu, Henrik Schumacher, Keenan Crane, Repulsive curves, ACM Trans. Graph. 40 (2) (2021).

[55] Johann Kipping, Doran Nettig, Zsolt Kallai, Thorsten Schueppstuhl, A robotic printer for nonplanar additive manufacturing of carbon fiber reinforced polymers, in: ISR Europe 2023; 56th International Symposium on Robotics, 2023, pp. 205–212.