Reinforced FDM: Multi-Axis Filament Alignment with Controlled Anisotropic Strength

GUOXIN FANG, Delft University of Technology, The Netherlands
TIANYU ZHANG, The Chinese University of Hong Kong, China
SIKAI ZHONG, Wayne State University, U.S.A
XIANGJIA CHEN, The Chinese University of Hong Kong, China
ZICHUN ZHONG, Wayne State University, U.S.A
CHARLIE C.L. WANG*, The University of Manchester, United Kingdom

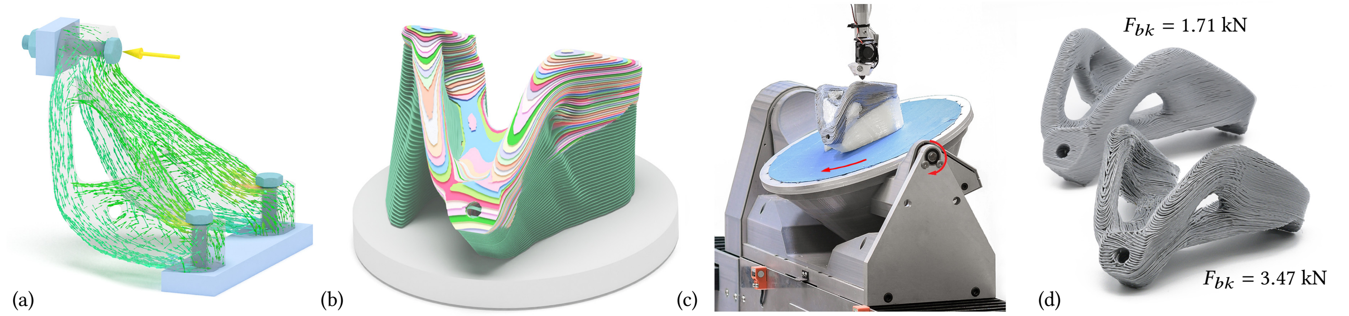


Fig. 1. Given a 3D model under certain loading (a), our method decomposes the model into curved layers (b) that optimizes the anisotropic strength of the 3D printed object while incorporating the manufacturing constraints. Compatible curved layers of supporting structures are also constructed for fabricating models with large overhang. Toolpaths are optimized to supervise the fabrication of curved layers on a dual-material multi-axis 3D printer (c) using fused deposition of filaments. (d) Compared to the planar layer-based 3D printing with optimized printing direction, the objects fabricated by our method can withstand up to 203% loads in the physical tensile tests.

The anisotropy of mechanical strength on a 3D printed model can be controlled in a multi-axis 3D printing system as materials can be accumulated along dynamically varied directions. In this paper, we present a new computational framework to generate specially designed layers and toolpaths of multi-axis 3D printing for strengthening a model by aligning filaments along the directions with large stresses. The major challenge comes from how to effectively decompose a solid into a sequence of strength-aware and collision-free working surfaces. We formulate it as a problem to compute an optimized governing field together with a selected orientation of fabrication setup. Iso-surfaces of the governing field are extracted as working surface layers for filament alignment. Supporting structures in curved layers are constructed by extrapolating the governing field to enable the fabrication of overhangs. Compared with planar-layer based Fused Deposition Modeling (FDM) technology, models fabricated by our method can withstand up to 6.35× loads in experimental tests.

CCS Concepts: • **Computing methodologies → Shape modeling**; **Mesh geometry models**.

Additional Key Words and Phrases: reinforcement, anisotropic strength, multi-axis motion, 3D printing

ACM Reference Format:
Guoxin Fang, Tianyu Zhang, Sikai Zhong, Xiangjia Chen, Zichun Zhong, and Charlie C.L. Wang. 2020. Reinforced FDM: Multi-Axis Filament Alignment with Controlled Anisotropic Strength. *ACM Trans. Graph.* 39, 6, Article 1 (December 2020), 15 pages. https://doi.org/10.1145/3414685.3417834

1 INTRODUCTION

3D printing is a process of joining materials to make objects directly from 3D digital models, which has motivated a lot of innovations in many areas. Filament-based *Fuse Deposition Modeling* (FDM) is the most widely adopted 3D printing process as it can fabricate more types of materials than other 3D printing technologies. In the current practice of 3D printing, materials are usually accumulated layer upon layer in planes along a fixed direction. This reduces the complexity of a system in both software and hardware as well as the cost of equipment. When applying this simplification to FDM, the weak adhesion between neighboring planar-layers of filaments

[*] Corresponding author: changling.wang@manchester.ac.uk (Charlie C. L. Wang)

Authors' addresses: Guoxin Fang, Department of Sustainable Design Engineering, Delft University of Technology, the Netherlands; Tianyu Zhang; Xiangjia Chen, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China; Sikai Zhong; Zichun Zhong, Department of Computer Science, Wayne State University, U.S.A; Charlie C.L. Wang, Department of Mechanical, Aerospace and Civil Engineering, The University of Manchester, United Kingdom.

0730-0301/2020/12-ART1 $15.00
https://doi.org/10.1145/3414685.3417834

always leads to an easy-to-delaminate problem. The anisotropy of mechanical property – strong along the axial directions of filaments but weak in other directions – can be observed on all models fabricated by FDM (ref. [Ahn et al. 2002]).

Three methods are usually conducted in industry to strengthen 3D printed parts [Gibson et al. 2009]: 1) changing the designed geometry [Stava et al. 2012; Zhou et al. 2013], 2) optimizing the processing setup (e.g., printing orientation [Umetani and Schmidt 2013], infill percentage and structure [Lu et al. 2014; Wang et al. 2013; Zhang et al. 2015]), and 3) applying chemical or thermal post-treatment. Most existing approaches in the literature of computational design and fabrication are developed along the first two threads, and have not considered the anisotropy of mechanical properties introduced by the process of FDM. The method of finding an optimized printing orientation [Umetani and Schmidt 2013] cannot be generalized to models with complex shape, which may have stress extremities along different directions in different regions (e.g., the Topo-opt model shown in Fig.1). In this paper, we propose a computational framework that is able to strengthen 3D printed models by specially designed layers and toolpaths taking advantage of the anisotropy.

1.1 Our Method

The challenge of how to decompose a solid into a sequence of strength-aware and collision-free working surfaces is tackled by computing an optimized governing field together with a selected orientation of fabrication setup. The working surface layers for multi-axis 3D printing are determined as iso-surfaces extracted from the governing field. A field extrapolation based method is developed to generate curved layers for fabricating the supporting structures in regions with large overhang. After that, a field-based method is developed to generate toolpaths on each curved layer. In summary, we make the following contributions:

- We introduce a field-based optimization framework to generate curved layers so that the anisotropy of fused filaments can be well controlled in multi-axis 3D printing to reinforce the mechanical strength of 3D printed models.
- When optimizing the field for layer decomposition, we enable the collision-free printing by 1) selecting a 'best' setup orientation and 2) updating the field in the regions with collision.
- We extrapolate the optimized field to generate curved layers of supporting structures for enabling the fabrication of large overhang in multi-axis 3D printing.
- We propose a field-based scheme to generate optimized toolpaths on curved layers for aligning filaments along designed directions.
- We physically fabricate prototypes by a FDM 3D printer with 5-axis motion, and we conduct tensile / compression experiments to verify the enhanced mechanical strength on these prototypes.

All steps of our framework are computed in a domain of tetrahedral mesh, which is naturally inherited from the *Finite Element Analysis* (FEA) for computing the stress distribution of a given model under certain loading.

1.2 Related Work

1.2.1 *Structural analysis and optimization for 3D printing.*

To improve the mechanical strength of 3D printed objects, many approaches have been developed in the literature to optimize both shape and topology of 3D models. For example, a lightweight structural analysis solver was developed in [Stava et al. 2012] to detect the areas with high structural stress, which were later repaired by hollowing, thickening and strut insertion. Langlois et al. [2016] presented a stochastic structural analysis method to predict all observed failure cases for a model subject to real-world loading conditions. Special structures [Cui et al. 2020; Lu et al. 2014; Wang et al. 2013; Zhang et al. 2015] and infills [Wu et al. 2018] are computed through numerical iterations to find optimized material distribution for strengthening stiffness while reducing the weight of a part.

In most of these approaches, FEA is employed in the loop of optimization to change the shape and topology of a model. Zhou et al. [2013] addressed the robustness issue to optimize a model in the worst-case considering not well-defined loads. Similarly, Schumacher et al. [2018] also presented a structural optimization approach to deal with the worst-case weakness in tension. Asymmetric strength of material under tension and compression has been considered in their approach. None of these approaches consider the anisotropy of mechanical properties introduced by the process of FDM 3D printing, not to mention the method to control such anisotropy for reinforcing 3D printed models as what is proposed in this paper. In our approach, we mainly focus on the scenarios with well-defined loads.

The anisotropy of mechanical property has been employed in some preliminary study to reinforce models. The orthotropic infill and microstructure are studied and modeled to better enhance the behavior of homogenization-based topology optimization [Groen and Sigmund 2018; Groen et al. 2019]. A fast cross-sectional structure analysis is employed to find an optimal 3D printing direction [Umetani and Schmidt 2013]. Ulu et al. [2015] presented another solution to find a best orientation by maximizing the *safety factor* defined by FEA. However, the reinforcement is limited by the conventional material deposition in planar layers.

1.2.2 *Curved layer slicing and multi-axis printing.*

In contrast to the traditional 3D printer where the orientation of the nozzle is fixed and only planar motion is applied, systems using multi-axis motion have caught a lot of attention in recent years (e.g., [Keating and Oxman 2013; Pan et al. 2014; Peng et al. 2016]). Relatively simple printing tasks with small components were demonstrated on these hardware systems.

To further explore the functionality in multi-axis 3D printing, advanced algorithms of toolpath generation have been invented. In literature, the first attempt at using non-planar layers in 3D printing was made a decade ago in an approach called the *Curved Layer Fused Deposition Modelling* (CLFDM) [Chakraborty et al. 2008], which provides the function of FDM fabrication using a toolpath with dynamically changed z-values within individual layers. Effort has been made to realize such 3-axis motion on a delta style parallel robot [Allen and Trask 2015; Llewellyn-Jones et al. 2016]. A model can be fabricated by producing a double-curved skin layer on top of a sandwich structure with static-z printed layers so that the staircase effect can be removed to improve the surface quality. Recently, Etienne et al. [2019] presented a method to generate curved layers to minimize the staircase effect by computing an optimized deformation for the given model represented by a tetrahedral mesh. By the inverse map of the deformation, cross-sections obtained from 2D slicing are converted into the curved layers for 3D printing. In contrast, we generate curved working surface directly in the tetrahedral mesh as iso-surface.

In order to fully utilize the six *degree-of-freedom* (DOF) introduced by multi-axis motion, an algorithm was proposed by Wu et al. [2016] to compute collision-free toolpaths to extrude materials for a wire-frame model (i.e., edges of a triangular mesh). A naive ordering of edges can lead to a configuration that certain edges cannot be approached. To tackle this challenge, a global planning is conducted on a directed graph. Another relevant research of 6-DOF in wire-frame 3D printing was proposed by Huang et al. [2016], which considers stability constraints in the manufacturing process together with collision-free constraints. Both approaches have included the time-consuming step of collision-detection in the loop of planning. As a result, only web-like models with very small number of primitives can be considered (e.g., less than 1k struts). Differently, Dai et al. [2018] developed a convex-front governed approach that can always ensure collision-free working surfaces for material accumulation. The printing model was represented by voxels and the curved layers were first extracted from grids formed by the centers of voxels, and then trimmed by the input model using Boolean operation. The robustness of their algorithm and the surface quality of extracted layers are significantly influenced by the artifacts caused by the voxel representation. In the method of Xu et al. [2019], the artifacts on the boundary of curved layers were voided by directly extracting iso-lines of a growing field computed on the mesh surface. Nevertheless, none of these multi-axis 3D printing approaches have considered the mechanical anisotropy introduced by using different toolpaths of filament alignment.

1.2.3 Toolpath generation. Generating collision-free toolpaths for multi-axis 3D printing is strongly relevant to the accessibility problem in multi-axis *Computer Numerical Control* (CNC) machining, which has been studied for more than two decades. The visibility map is employed in [Elber 2014] to study the accessibility of multi-axis milling. Recent work further improves the performance of 5-axis milling by computing a gouge-free toolpath while also optimizing the dynamic behavior of machines [Kim et al. 2015]. The computation of accessibility is complicated and time-consuming. The problem of collision-avoidance is resolved in this paper by finding a good orientation of fabrication setup and locally updating the governing field for layer generation.

Researchers have started to consider the influence of toolpaths on the mechanical strength of 3D printed models. An implicit function based toolpath generation algorithm for 3D printing has been developed in [Steuben et al. 2016]. Their toolpaths are governed by stress-fields on 2D slices for fabrication. However, the toolpaths are not directly optimized according to structural analysis. Hornus et al. [2020] introduced a method to reduce the lack-of-filament region in each planar layer by locally adjusting the width of tool-paths. Principal stress lines are employed for generating toolpaths that can strengthen 3D printed structures [Tam and Mueller 2017]. Nevertheless, the computation in these approaches is still limited on a two-manifold (i.e., a planar cross-section or a surface). Differently, our work proposed in this paper focuses on generating specially designed toolpaths inside a three-manifold (i.e., a larger searching space in a volume).

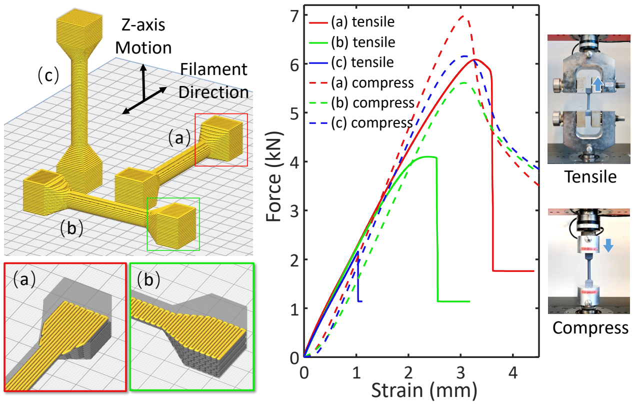


Fig. 2. Anisotropic mechanical behavior can be clearly observed on 3D printed specimens with filaments aligned along different directions. We can observe the enhancement of the breaking force with 176% (in tensile tests) and 27% (in compression tests) when aligning filaments along the direction of loading. All material samples were tested using same setup and loading condition as show in the right.

The toolpaths for aligning filaments in each curved layer have significant influence on the quality of 3D printed objects. With the help of boundary distance-field, Zhao et al. [2016] presented a method to generate connected toolpaths as contour-parallel Fermat spirals which can reduce the amount of discontinuity. The concept is later extended to compute milling toolpaths for 3D surfaces [Zhao et al. 2018]. However, the anisotropic strength of filament has not been considered yet. Moreover, the material filling rate of Fermat spiral is low near the medial-axis regions. This will lead to a relatively weak mechanical strength [Fernandez-Vicente et al. 2016]. While generating stress-oriented toolpaths in critical regions, we employ a field-based hybrid strategy to generate directional-parallel and contour-parallel [Jin et al. 2013] toolpaths for ensuring good filling rate in other regions. In the community of CNC machining, vector-fields have been employed to generate toolpaths for different objectives (e.g., energy-efficiency [Pavanaskar et al. 2015]).

2 ANISOTROPIC STRENGTH

In this section, we study the anisotropic property of 3D printed models and present the strategy of filament alignment for controlling the anisotropic strength.

The anisotropy of mechanical strength on models fabricated by FDM has been studied by experiments in prior work [Ahn et al. 2002; Tam and Mueller 2017]. The fractographic analysis using *scanning electron microscope* (SEM) images [Riddick et al. 2016] has shown that the weak adhesion between neighboring layers and also the incompletely filled area between filaments [Xie et al. 2020] are the major reasons for tensile failure. This anisotropy in yield

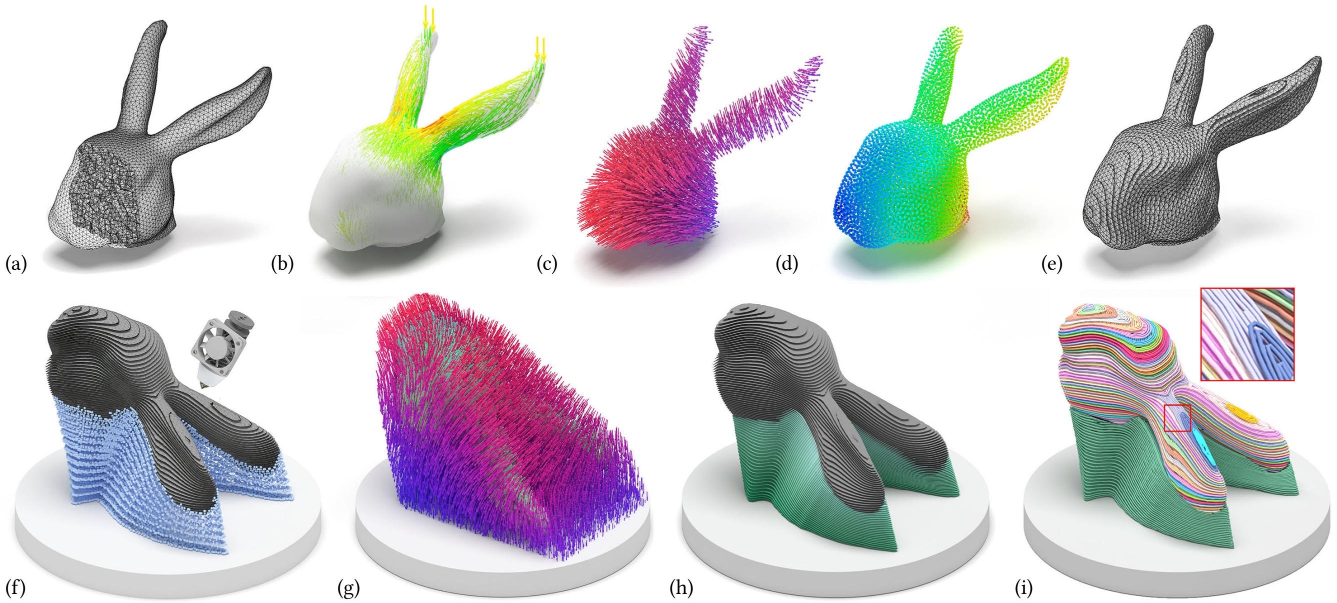


Fig. 3. An overview of our computational framework for generating curved layers and toolpaths with controlled anisotropic strength: (a) a Bunny-head model $\mathcal{H}$ is represented by a tetrahedral mesh $\mathcal{T}$, (b) principal stresses with values are visualized by colors, (c) a vector-field $\mathbf{V}(\mathbf{x})$ is optimized according to the principle of reinforcement and the fabrication constraints, (d) a scalar-field $G(\mathbf{x})$ is obtained by enforcing $\nabla G(\mathbf{x})$ to follow $\mathbf{V}(\mathbf{x})$, (e) preliminary curved layers are generated by extracting the iso-surfaces from $G(\mathbf{x})$, (f) an orientation of fabrication is determined by considering the accessibility of printer head and regions with large overhangs are detected by a sampling based method, (g) a vector-field $\bar{\mathbf{V}}(\mathbf{x})$ is extrapolating $\mathbf{V}(\mathbf{x})$ for supporting structure, (h) final curved layers are extracted from the governing fields for 3D printing, and (i) toolpaths are generated for curved layers according to the principal stresses.

strength is also demonstrated in our experiment with tensile and compression tests on different specimens printed along different directions (see Fig. 2). When the filaments are aligned along the direction of tensile load (the sample (a)), the breaking force is 176% higher than the case under transverse load (the sample (c)) and 51.6% higher than the in-layer tearing (the sample (b)) in tensile tests. Meanwhile, the breaking force under compression along the axial direction of filaments is about 27.6% higher than applying transverse compression.

We apply the following principle to design multi-axis toolpaths for reinforcing 3D printed models in FDM.

Principle of Reinforcement: *Aligning the axial direction of filaments along the maximal principal stresses in local regions can provide overall strength improvement of a part under given loads.*

In this paper, we follow this principle to slice every input model into curved layers and further generate the toolpaths on every layer. The whole computational pipeline can finally provide a process for depositing filaments along the directions of principal stresses. In practice, we release the requirement a bit by only applying the principle in selected critical regions to ensure the manufacturability.

3 OVERVIEW

To tackle the challenge of generating strength-aware and manufacturable working surfaces inside a given model $\mathcal{H}$, we propose a field-based computational pipeline conducted on the tetrahedral mesh representation $\mathcal{T}$ of the solid which consists of two parts: 1) field optimization for generating strength-aware layers (Fig. 3(a)-(e)) and 2) computation for enabling fabrication (Fig. 3(f)-(i)).

Field-Optimization for Slicing The main idea for the slicing algorithm of our reinforced FDM framework is to compute an optimized scalar-field $G(\mathbf{x})$ inside $\mathcal{H}$ according to the stress analysis, where both the distribution of stresses and the scalar-field $G(\mathbf{x})$ are stored on the tetrahedral mesh $\mathcal{T}$ (see Fig.3(a) and (b)). After optimization, the iso-surfaces $\{S_i\}_{i=1,\cdots,n}$ with $S_i = \{\mathbf{x} \mid G(\mathbf{x}) = d_i\}$ will be extracted from $G(\mathbf{x})$ to serve as working surfaces for toolpath generation. Each surface is represented as a triangular mesh (Fig. 3(e)). Objective of strengthening is derived to enforce the orientation of working surfaces by optimizing the gradient $\nabla G(\mathbf{x})$ of $G(\mathbf{x})$ according to the principle of reinforcement (Section 2). Moreover, objectives for fabrication are also derived by controlling $\nabla G(\mathbf{x})$ – see the details presented in Section 4.1.

Directly optimizing the objective functions defined on $\nabla G(x)$ will lead to a problem with many local minima, the computation of which is very slow and hard to converge. We propose to first compute an optimized vector-field $\mathbf{V}(\mathbf{x})$ with reference to the objectives defined on $\nabla G(\mathbf{x})$ (see Fig. 3(c) and Section 4.2). The field $G(\mathbf{x})$ for curved layer slicing is later determined by minimizing the difference between $\nabla G(\mathbf{x})$ and $\mathbf{V}(\mathbf{x})$ (Section 4.3). To generate compatible vector-fields that lead to manufacturable working surfaces, constraints derived from the principle of reinforcement are only applied to the highly stress regions.

Fabrication Enabling Although the smoothness of local surface has been considered when generating the governing field $G(\mathbf{x})$, the global accessibility of the printer head is not guaranteed. Given a set of curved layers, the accessibility is strongly coupled with the orientation of a model's setup in 3D printing as the bottom part is restricted by the build platform of hardware (see Fig. 3(f)). We define a conservative metric that is able to efficiently evaluate the accessibility of all layers along different setup orientations, and search one with optimal accessibility (Section 5.1.1). For models where layers remain inaccessible (not too many according to our experimental tests), the working surfaces at collided regions will be 'flattened' by adjusting the vector-field progressively (Section 5.1.2). This may sacrifice the anisotropic strength in relevant regions; however, we argue that the manufacturability has higher priority.

Additionally, the fabrication of regions with large overhangs is another difficulty to overcome in our framework. After detecting the regions which need support by a sampling based method, we compute a tetrahedral mesh $\bar{\mathcal{T}}$ as an envelope of these sampled regions (see the blue regions in Fig. 3(f)). The constraint $\bar{\mathcal{T}} \cap \mathcal{T} = \mathcal{M}$ is imposed for mesh generation, where $\mathcal{M}$ indicates the boundary surface of $\mathcal{T}$. With the help of this compatibility, we extrapolate the vector-field $\mathbf{V}(\mathbf{x})$ in $\bar{\mathcal{T}}$ as $\bar{\mathbf{V}}(\mathbf{x})$ (Fig. 3(g)) to result in a scalar-field $\bar{G}(\mathbf{x})$ for generating the curved layers of supporting structures (see Fig. 3(h) and Section 5.2). By employing the strategy of thickness-controlled slicing (Section 5.3), toolpaths optimized according to the principle of reinforcement can then be generated on curved layers (see Fig. 3(i) and Section 5.4).

4 FIELD-BASED SLICING

In this section, we introduce the method to generate curved layers for multi-axis 3D printing by following an optimized scalar-field $G(\mathbf{x})$. After defining the objective functions for optimization, a vector-field $\mathbf{V}(\mathbf{x})$ is introduced as an intermediate to compute the desired gradient of $G(\mathbf{x})$. In consequence, the governing field $G(\mathbf{x})$ can be efficiently determined from $\mathbf{V}(\mathbf{x})$ by solving a least-squares problem. Curved layers for FDM fabrication are obtained by extracting the iso-surfaces of $G(\mathbf{x})$.

4.1 Objectives of Optimization

Goals and constraints for optimizing the governing field $G(\mathbf{x})$ are analyzed here to derive objective functions for optimization.

4.1.1 Anisotropic strength. Given user-specified loads on the model $\mathcal{H}$, we use commercial FEA software (e.g. *Abaqus*) to compute the stress distribution and output the 3×3 stress tensor $\sigma(e)$ for every tetrahedral element $e \in \mathcal{T}$ at the centers $\mathbf{x}_e$. Consistently, all constraints related to stresses are also imposed at the centers of tetrahedra in our formulation. Notice that the slopes of the material curves shown in Fig. 2 are similar to each other, which indicates similar stiffness of 3D printed models along different printing directions although the experiment shows strong anisotropy in yield strength. Therefore, linear-FEA with isotropic material property (Young's modules as $2,346.5MPa$, Poisson's ratio as 0.37[1]) is applied at this stage to compute the stress distribution.

Eigenvalue decomposition is then applied to each stress tensor to determine three principal stresses $[\sigma_1, \sigma_2, \sigma_3]$ sorted by their absolute values as $|\sigma_1| > |\sigma_2| > |\sigma_3|$. The eigenvector associated with the maximal principal stress σ_1 is defined as the maximal stress direction $\tau_{\max}$. Similarly, the eigenvector associated with the minimal principal stress σ_3 is defined as the $\tau_{\min}$. $\tau_{\max}(e)$ and $\tau_{\min}(e)$ are stored in every tetrahedron $e \in \mathcal{T}$ to supervise the field optimization for layer decomposition and toolpath generation.

In order to borrow the anisotropy of yield strength in FDM, we aim at generating toolpaths to follow the critical orientation $\tau_{\max}$. In our pipeline, curved layers are obtained by extracting the iso-surfaces $\{S_i\}$ of the scalar-field $G(\mathbf{x})$. Ideally, we need to make the curved layer tangential to the critical orientation $\tau_{\max}$. As a result, toolpaths on each curved layer can be generated to follow the critical orientations. This requirement can also be formulated as letting the surface normal of each layer, which is in fact the gradient of $G(\mathbf{x})$, be perpendicular to $\tau_{\max}$ and be parallel to $\tau_{\min}$. Considering $\tau_{\min} \perp \tau_{\max}$, we only need to impose the objective of letting $\nabla G(\mathbf{x})$ be parallel to $\tau_{\min}$ as it also ensures $\nabla G(\mathbf{x}) \perp \tau_{\max}$. This also imposes the directions of first two principal stresses being tangential to the surface-layers for further optimization in toolpath generation.

By selecting some critical regions (the method of selection will be discussed below), the elements in critical regions are represented as a set $\mathcal{T}^* \subset \mathcal{T}$. The requirements of $\tau_{\max}$ and $\tau_{\min}$ in these regions can be achieved by minimizing the following objective function

$$E_s = \sum_{e \in \mathcal{T}^*} V_e \|\nabla G(\mathbf{x}_e) \times \tau_{\min}(e)\|^2, \tag{1}$$

where $\mathbf{x}_e$ is the center of a tetrahedral element, the volume of element e is employed as the weight V_e, and the field values of every nodes are considered as variables to be optimized. $G(\mathbf{x})$ inside each element is defined as a piecewise linear function interpolating the node values.

4.1.2 Layer thickness. As working surfaces are generated by extracting iso-surfaces of $G(\mathbf{x})$, the distance between two neighboring working surfaces S_i and S_{i+1} varies from place to place. According to the limitation of currently available hardware, the thickness of each layer that can be 3D printed has a limited range $[t_{\min}, t_{\max}]$. This should be controlled when optimizing $G(\mathbf{x})$. For two points $\mathbf{p}$ and $\mathbf{q}$ that are both inside $\mathcal{T}$, it indicates to deposit material at $\mathbf{p}$ earlier than at $\mathbf{q}$ when $G(\mathbf{p}) < G(\mathbf{q})$. The gradient of $G(\mathbf{x})$ indicates the speed of 'growing' materials. When the norm of $\nabla G(\mathbf{x}_e)$ is optimized to be nearly a constant, we can control the layer thickness to be as uniform as possible.

The control of layer thickness is therefore achieved by minimizing the following objective function

$$E_t = \sum_{e \in \mathcal{T}} V_e (\|\nabla G(\mathbf{x}_e)\| - c)^2, \tag{2}$$

where c is a target thickness of each layer – e.g., $c = (t_{\min} + t_{\max})/2$.

4.1.3 Harmonic of gradients. Highly curved or even closed iso-surfaces of $G(\mathbf{x})$ will be formed in the regions with large variation in gradients. Caused by the potential collision between the printer head and the already fabricated layers, the curved layers generated

[1] Material property from technical data sheet of PLA filament

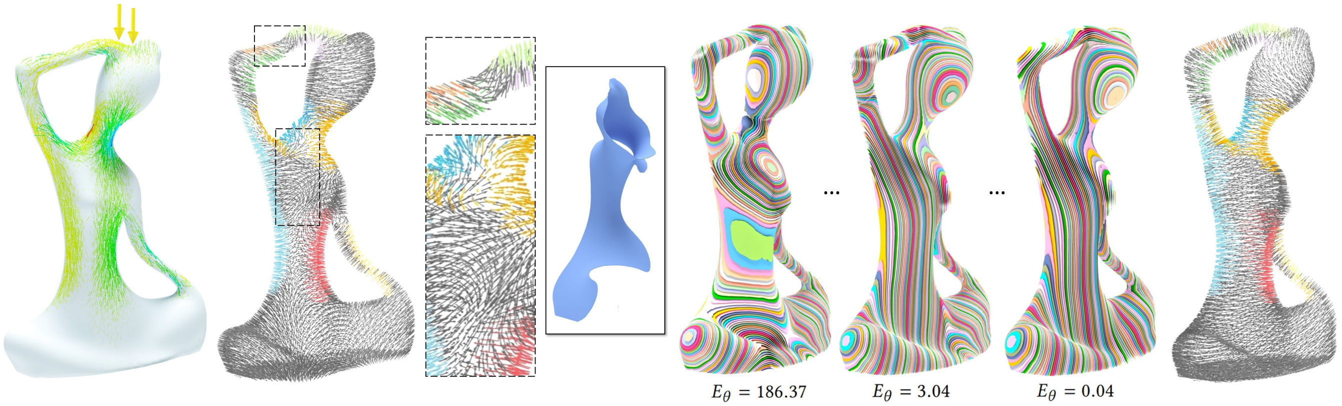


Fig. 4. The orientations of vectors in the segmented critical regions (displayed in different colors) have significant influence on the compatibility of $\mathbf{V}(\mathbf{x})$ (as indicated $\nabla G(\mathbf{x})$), where incompatibility will result in iso-surfaces not able to be printed – see the blue surface. The greedy algorithm for alternating orientations of vectors in the segmented critical regions can effectively improve $\mathbf{V}(\mathbf{x})$'s compatibility – therefore the quality of $G(\mathbf{x})$'s iso-surfaces. Progressive results with their corresponding values of E_θ (Eq. (6)) are also given.

from iso-surfaces in these regions are not printable (see the blue iso-surface shown in Fig. 4). Therefore, the objective function below is proposed to impose harmonic variation on gradients in neighboring tetrahedra.

$$E_h = \sum_{(e_a,e_b)\in\mathcal{N}} A_{a,b}\|\nabla G(\mathbf{x}_{e_a}) - \nabla G(\mathbf{x}_{e_b})\|^2, \quad (3)$$

where $\mathcal{N}$ denotes the pairs of neighboring tetrahedra, and the area of face shared by e_a and e_b is used as the weight $A_{a,b}$.

Ambiguity: The objective function of anisotropic strength Eq. (1) is not well defined as there are two possible directions for $\nabla G(\mathbf{x}_e)$ as $\pm\tau_{\min}$. This leads to a lot of local minima in the space of solution. Adding the harmonic energy Eq. (3) can somewhat but not completely resolve this problem. Moreover, the direction of principal stresses determined by eigenvalue decomposition is less robust in regions with relatively isotropic stress tensor – i.e., $\tau_{\min}$ is given ambiguously.

Critical Regions: Enforcing the direction of filament alignment everywhere inside the solid $\mathcal{H}$ can easily result in a governing field $G(\mathbf{x})$ hard to meet the requirements of layer thickness and harmonic on its gradient. Therefore, two factors are considered when selecting the critical regions – i.e., a tetrahedron will be added into $\mathcal{T}^*$ only when: 1) $|\sigma_1| > k_1|\sigma_{max}|$ and 2) $|\sigma_2| > k_2|\sigma_3|$. k_1 and k_2 are two parameters that can be tuned by users, where k_1 is employed to control the percentage of regions to be reinforced and k_2 is used to ensure the certainty of σ_{min} generated in eigenvector decomposition (i.e., regions with $|\sigma_2| \approx |\sigma_3|$ are avoided). In our framework, $k_1 = 0.1$ and $k_2 = 3.0$ are chosen by experiment.

4.2 Vector-Field Based Optimization

To determine an optimal solution by considering all the factors as discussed above, an intuitive solution is to minimize the weighted objectives as $w_sE_s + w_tE_t + w_hE_h$. As a non-linear optimization problem, it's hard to be effectively solved. Meanwhile, how to choose the weight for each item is tricky. To overcome the difficulty in optimization, we introduce a vector-field $\mathbf{V}(\mathbf{x})$ as an intermediate to approximate $\nabla G(\mathbf{x})$. $\mathbf{V}(\mathbf{x})$ is represented in a discrete form by storing one vector for each element $e \in \mathcal{T}$.

The major difficulty for computing a vector field with minimized E_s and E_h comes from the τ_{min} ambiguity and the smoothness requirement of the field to be determined. To handle similar field optimization problem, Arora et al. [2019] imposed a non-linear optimization based method. The ambiguity of frame alignment is translated to 'truncated' frame-tensors matching with the objectives, which are optimized together with the Lapacian smoothness term. This method is effective but if directly applied to our vector field alignment problem, it may converge to local minimum as the design space is more flexible. Recently, a field optimization solution was introduced in [Gil-Ureta et al. 2020] to ensure a convex energy that can be sufficiently optimized. However, their formulation is mainly for thin-layer structures.

When we are able to give the deterministic value of every vector $\mathbf{v}_e$ in the critical region, vectors in free regions can be determined by minimizing the harmonic energy similar to Eq. (3). To avoid solving the optimization problem with too many variables, we conduct a strategy to first segment the vectors in $\mathcal{T}^*$ into sub-regions $\{\mathcal{T}_j^*\}$ and then assign a consistent orientation of $\tau_{\min}$ to every element in the same sub-region. That can be formulated as

$$\arg\min_{\mathbf{v}_e} E_w = \sum_{(e_a,e_b)\in\mathcal{N}} A_{a,b}\|\mathbf{v}_{e_a} - \mathbf{v}_{e_b}\|^2, \quad (4)$$

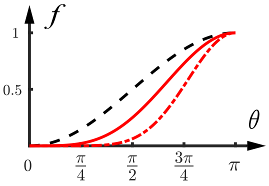
$$s.t. \quad \mathbf{v}_e = \rho_j\tau_{\min}(e) \quad (\forall e \in \mathcal{T}_j^*, j = 1, 2, ...m). \quad (5)$$

The least-squares solution of $\partial E_w/\partial \mathbf{v}_e = 0$ ($\forall e \in \mathcal{T} \setminus \mathcal{T}^*$) can be efficiently determined by imposing the boundary conditions in Eq. (5). However, we are still facing the difficulty of ambiguity as discussed above – i.e., $\mathbf{v}_e$ ($\forall e \in \mathcal{T}^*$) could have two options as $\rho_j = \pm 1$. An algorithmic scheme is developed below to determine the sign of $\tau_{\min}$ while considering the compatibility of all neighboring vectors in the field.

The segmentation is conducted by applying a flooding algorithm. Two neighboring elements e_a and e_b are considered as belonging to the same sub-region if

$$|\tau_{\min}(e_a)\cdot\tau_{\min}(e_b)| > \alpha,$$

where α is the variable used to control the number of sub-regions and the performance of segmentation. Choosing a smaller α could incorrectly group elements with large directional variation of $\tau_{\min}$ into the same cluster, and a larger value may result in too many sub-regions. $\alpha = 0.75$ was selected according to the results of experiments. When adding a new element into an already flooded region, we reverse the orientation of its $\tau_{\min}$ if it is not consistent with other elements in the region. Suppose m sub-regions $\{\mathcal{T}_j^*\}_{j=1,\cdots,m}$ can be formed (e.g., as displayed in different colors in Fig. 4). Now all elements in a sub-region can have multiple $\tau_{\min}$ with consistent orientation. However, incompatibility of vectors similar to vortex flow in fluid mechanics (see the zoom-views of Fig. 4) can still be found. We need to optimize the orientation compatibility between sub-regions $\{\mathcal{T}_j^*\}$.



First of all, we need a function $f(\hat{\mathbf{v}}_a, \hat{\mathbf{v}}_b)$ that is monotonic to θ and will only be activated when θ is nearly $\pi/2$ (i.e., $\hat{\mathbf{v}}_a$ and $\hat{\mathbf{v}}_b$ tend to be incompatible). In the right figure, we compare different possible options: $f = \frac{1}{4}\|\hat{\mathbf{v}}_a - \hat{\mathbf{v}}_b\|^p$ (black curve) and $f = \frac{1}{2^p}(1 - \hat{\mathbf{v}}_a \cdot \hat{\mathbf{v}}_b)^p$ with $p = 2$ (red solid) and $p = 4$ (red dash). According to the analysis, we define the objective for optimizing the compatibility between neighboring vectors as

$$E_\theta = \sum_{(e_a,e_b)\in\mathcal{N}} A_{a,b}(\|\mathbf{v}_{e_a}\|\|\mathbf{v}_{e_b}\| - \mathbf{v}_{e_a}\cdot\mathbf{v}_{e_b})^4. \qquad (6)$$

A greedy algorithm is employed to update the values of $\{\rho_j\}$ by repeatedly reversing the sign of ρ_j in the region that gives the largest decrease of $\Delta E_\theta(\rho_j) = E_\theta(\rho_j) - E_\theta(-\rho_j)$. The compatibility of the vector-field can be significantly improved by optimizing the orientation of $\tau_{\min}$ (see Fig. 4 for an illustration).

4.3 Governing Field and Slicing

After determining the optimized vector-field $\mathbf{V}(\mathbf{x})$, we can compute the governing field by solving the following minimization problem to determine the field values $\{g_i\}$ on all nodes as

$$\{g_i\} = \arg\min \sum_{e\in\mathcal{T}} w_e\|\nabla G(\mathbf{x}_e) - \hat{\mathbf{v}}_e\|^2, \qquad (7)$$

where $\hat{\mathbf{v}}_e = c\mathbf{v}_e/\|\mathbf{v}_e\|$ is used to impose the objective of layer thickness. When conducting a linear interpolation function to present the field function $G(\mathbf{x})$ inside every element, the gradient $\nabla G(\mathbf{x}_e)$ is in the form of a linear combination of field values on the four nodes of an element e. Therefore, Eq. (7) is in a least-squares form and can be solved efficiently.

Given an iso-value d, the iso-surface $G(\mathbf{x}) = d$ can be extracted as a triangular mesh from the tetrahedra mesh $\mathcal{T}$. This is an application of the marching tetrahedra algorithm [Treece et al. 1999]. After finding the minimal and the maximal values of $G(\mathbf{x})$ on all vertices as $g_{\min}$ and $g_{\max}$, we incrementally determine the iso-values d_i for all curved layers S_i according to the requirements of layer thickness. More details will be discussed later in the following section.

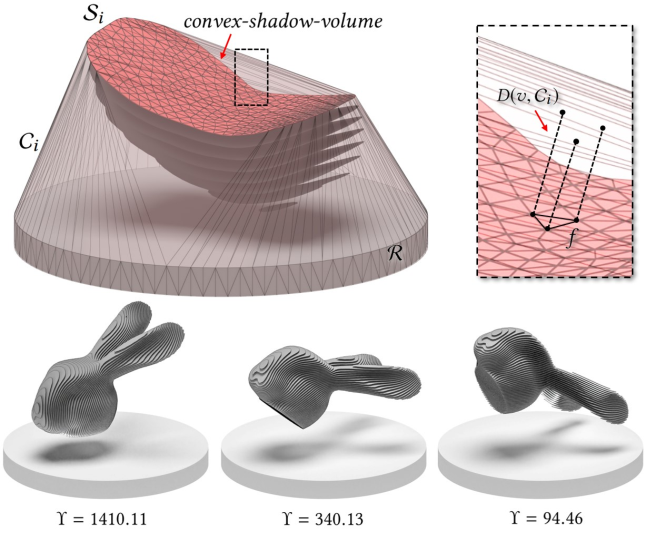


Fig. 5. (Top) An illustration of how to evaluate the *convex-shadow-volume* of a working surface $\mathcal{S}_i$. (Bottom) Different values of the convex-shadow-volume Υ for the Bunny head model in different setup orientations.

In our framework, we minimize the objectives for stress alignment, thickness control, and harmonic requirement by solving two least-squares optimization problems. This strategy can efficiently find a good balance of multiple objectives. When determining the scalar field $G(\mathbf{x})$ by solving Eq. 7, the values of E_s and E_h may be slightly increased which is caused by the difference between $\nabla G(\mathbf{x})$ and $\mathbf{V}(\mathbf{x})$. However, this step can significantly minimize the value of E_t to guarantee the nearly uniform thickness of layers.

5 FABRICATION ENABLING

After being able to generate the set of curved layers $\{\mathcal{S}_i\}$, algorithms are developed in our framework to enable the fabrication of each curved layer by using a multi-axis FDM 3D printer. The problems to be solved include the accessibility optimization, the support generation, the layer thickness control and the toolpath generation.

5.1 Accessibility Optimization

For every curved layer $\mathcal{S}_i$, we need to ensure its *accessibility* – i.e., can be touched by a printer head while not colliding with the platform $\mathcal{R}$ and any other already printed layers $\mathcal{S}_k$ ($\forall k < i$). This is achieved by applying the following two schemes.

5.1.1 Find best setup orientation. The setup orientation of a model has significant influence to the accessibility of its working surfaces (i.e., layers in 3D printing), which has been proved by the prior research in 5-axis machining [Hu et al. 2013; Radzevich and Goodman 2002]. Among all possible setup orientations of a model $\mathcal{H}$ placed on top of the 3D printer's platform $\mathcal{R}$, an orientation that maximizes the summed accessible area of all its working surfaces

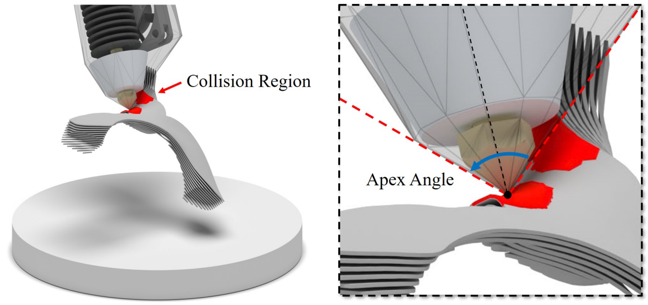


Fig. 6. An illustration for collision detection and the bounding volume of a specially designed printer head with small apex angle.

is to be determined. Although the location of $\mathcal{H}$ above $\mathcal{R}$ can also affect its accessibility, it is less sensitive than changing orientations. Only orientation is considered in our approach, and the position of $\mathcal{H}$ is fixed to let its center above the center of $\mathcal{R}$ and the whole model slightly higher than the platform $\mathcal{R}$.

It is very time-consuming and therefore impractical to directly compute the area of accessible regions according to a given orientation of fabrication setup. Inspired by the concept of advancing convex-front to ensure accessibility in material deposition [Dai et al. 2018], we define an easy-to-compute *convex-shadow-volume* as an indirect metric for quickly evaluating the accessibility. When conservatively considering the shape of a printer head as flat-end, every working surface $\mathcal{S}_i$ should be located on the boundary of convex hull C_i of $\mathcal{R}$ and $\mathcal{S}_j$ $(\forall j \leq i)$. A triangle $f \in \mathcal{S}_i$ falls into the *shadow* region behind the convex-front of material accumulation if it is inside C_i. The deeper f falls into the shadow, the harder it is to be touched in a collision-free way. This can be evaluated by the approximated shadow volume as $\frac{1}{3}A(f)\sum_{v \in f} D(v, C_i)$ with $D(v, C_i)$ returning the distance between a vertex v to the convex-hull C_i (see the illustration in Fig. 5). Therefore, the convex-shadow-volume for a working surface is defined as the sum of all triangles' shadow volumes. The total convex-shadow-volume for all working surfaces $\{\mathcal{S}_i\}_{i=1,\cdots,n}$ is

$$\Upsilon = \frac{1}{3}\sum_{i=1}^{n}\sum_{f \in \mathcal{S}_i}\sum_{v \in f} A(f) D(v, C_i), \tag{8}$$

which is employed for searching the best setup orientation.

We randomly sample points on the Gaussian sphere to serve as candidates of setup orientations, and the one giving the smallest Υ is selected. After rotating a model to this 'best' orientation, we need to check if the node with the minimal value of $G(\mathbf{x})$ is closer to the platform than the node with the maximal value $g_{\max}$. When this is not the case, we adjust the orientation of fields by reversing all the vectors in the vector field $\mathbf{V}(\mathbf{x})$ and assigning $G(\mathbf{x}) = g_{\max} - G(\mathbf{x})$. Finally, the accessibility of each working surface $\mathcal{S}_i$ can be verified by placing the 3D model of the printer head to the sample points on $\mathcal{S}_i$ for detecting the collision. Surface normal of every sample point is used as the orientation of the printer head. To accelerate the collision detection, a convex bounding volume of the printer head is employed in our implementation.

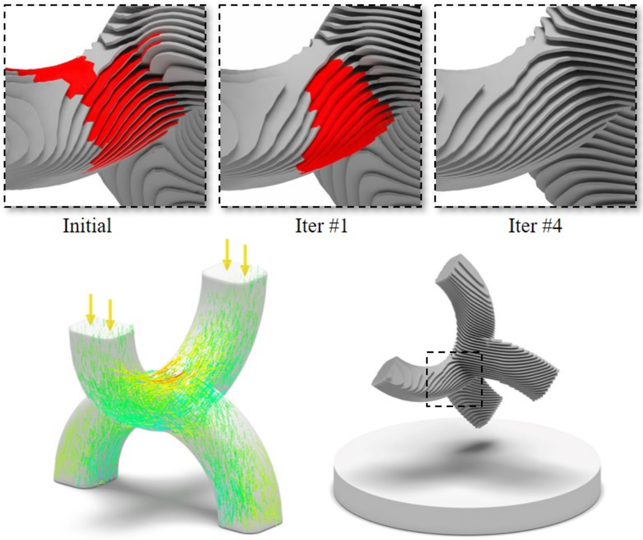


Fig. 7. Field relaxation is applied to the C^2-model for improving its accessibility: the stress distribution under loading (bottom-left), the curved layers generated by our algorithm (bottom-right), and the progressive results of relaxation (top row).

5.1.2 *Field relaxation for accessibility.*

When using a specially designed printer head, the bounding volume can have a cone shape with a very small apex angle (see Fig. 6). This is very helpful to allow the printer head to go into deep shadow regions without collision – i.e., a better accessibility comparing to the general 3D printer head with nearly a flat end-shape. After selecting the best setup orientation and using this printer head, accessibility of all working surfaces is already enabled on many models with complex shapes (e.g., the *Topo-opt* model in Fig. 1, the *Bunny head* model in Fig. 3, and the *Yoga* model in Fig. 4). However, this is not guaranteed – especially when dealing with models having sharp change of principal stresses. As shown in Fig. 7, the symmetric stress field distribution of the C^2-model makes it difficult to fabricate some of the complex curved layers even after selecting the best setup orientation. On the other hand, the relative size of a model w.r.t. the 3D printer head will also influence the accessibility. To deal with these cases, we slightly sacrifice the anisotropic strength of a model to ensure the accessibility by using progressive field relaxation.

All the tetrahedra adjacent to the points resulting in collision are first determined and stored in a set $\mathcal{T}_{col}$. Then, a smooth weighting field $\omega(\cdot)$ is computed on all tetrahedra as

$$\min_{(e_a,e_b)\in\mathcal{N}} (\omega_{e_a} - \omega_{e_b})^2, \tag{9}$$
$$s.t. \quad \omega_e = \eta \ (\forall e \in \mathcal{T}_{col}), \quad \omega_e = 0 \ (\forall e \in \mathcal{T}_{away}).$$

$\mathcal{T}_{away}$ defines the set of tetrahedra in $\mathcal{T}$ that are n-rings away from $\mathcal{T}_{col}$ (e.g., $n = 5$ is used according to experiment). Considering flat layers parallel to the platform are always printable, we relax the vector-field $\mathbf{V}(\mathbf{x})$ by

$$\mathbf{v}_e = \omega_e \mathbf{v}_e + (1 - \omega_e)\hat{\mathbf{z}} \tag{10}$$

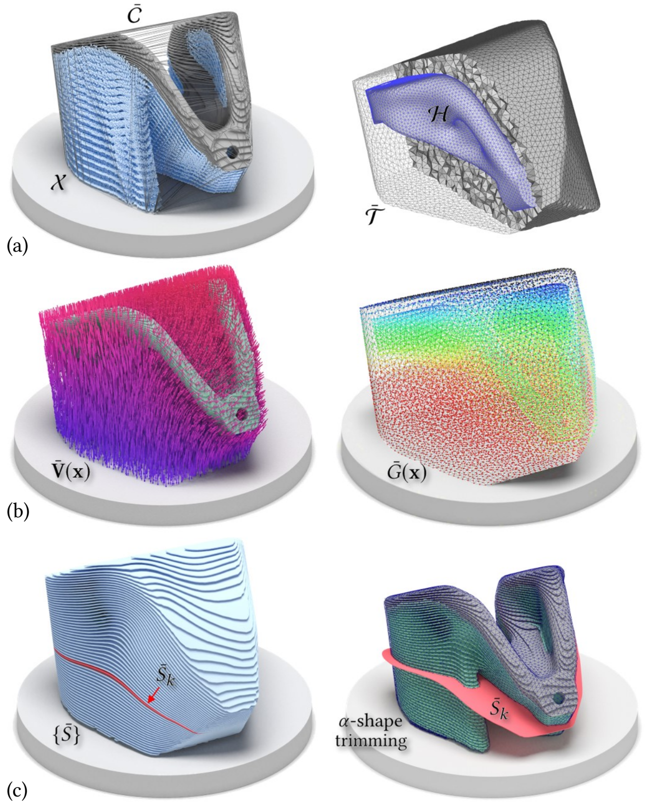


Fig. 8. An illustration of support generation: (a) a set of sample points $\mathcal{X}$ for indicating the region of support and the tetrahedral mesh $\bar{\mathcal{T}}$ generated in the convex-hull $\bar{C} = C(\mathcal{X} \cup \mathcal{H})$, (b) a new vector-field $\bar{\mathbf{V}}(\mathbf{x})$ and the corresponding scalar-field $\bar{G}(\mathbf{x})$ computed in $\bar{\mathcal{T}}$ by extrapolating $\mathbf{V}(\mathbf{x})$, and (c) extracting iso-surfaces of $\bar{G}(\mathbf{x})$ as $\{\bar{S}_k\}$ and trimming them by the α-shape of $\mathcal{X}$.

with $\hat{\mathbf{z}} = (0, 0, -1)$. After each relaxation, we conduct the method presented in Section 4.3 to generate the updated $G(\mathbf{x})$ and the new curved layers. The relaxation is repeatedly applied to the vector field $\mathbf{V}(\mathbf{x})$ until all working surfaces become fully accessible.

Note that, this field-relaxation method can always generate printable layers at the end of iteration as it converges to the fully planar layers. The speed of relaxation is controlled by the factor η. Using a large value will 'flatten' the curved layers very quickly. $\eta = 0.7$ is experimentally selected in our tests.

5.2 Support Generation for Overhang

We now tackle the other problem of fabrication – regions with large overhang. Although the multi-axis motion has been used to print 3D models in a support-free way [Dai et al. 2018], incorporating the sequence of material deposition determined by their approach in our framework will tremendously change the alignment direction of toolpaths. As a consequence, the anisotropic strength cannot be well controlled any more. Therefore, we adopt the method to add supporting structures by using water-soluble materials.

Methods of adding supporting structures for planar 3D printing [Dumas et al. 2014; Vanek et al. 2014] cannot be applied here as it is hard to make the planar layers of support be compatible with the curved layers of $\mathcal{H}$. We seek the help of extrapolating $G(\mathbf{x})$ in regions that need additional support to generate curved layers for supporting structures. There are three steps in our algorithm presented in the following subsections.

5.2.1 *Detection.*

Assuming the printer head is kept orthogonal to a working surface $\mathcal{S}_i$ during material deposition, we can detect the overhang regions by ray-intersecting. Specifically, starting from any point $\mathbf{p} \in \mathcal{S}_i$, we shot a ray along the opposite direction of surface normal at $\mathbf{p}$. When this ray intersects the platform $\mathcal{R}$ or any layer $\mathcal{S}_{k\ (k<i-1)}$ printed earlier than the previous one $\mathcal{S}_{i-1}$, supporting structures are needed. After sampling the intersected rays with distance c – i.e., the desired layer thickness, we can obtain a set of samples $\mathcal{X}$ that indicate the region needs support (see Fig. 8(a)).

5.2.2 *Meshing and extrapolation.*

We compute the convex hull $\bar{C} = C(\mathcal{X} \cup \mathcal{H})$ to generate a tetrahedral mesh $\bar{\mathcal{T}}$ in $\bar{C}$ as an envelope of the sampled region $\mathcal{X}$ and $\mathcal{H}$. The constraint $\bar{\mathcal{T}} \cap \mathcal{T} = \mathcal{M}$ is imposed for mesh generation so that the nodes and faces from $\bar{\mathcal{T}}$ and $\mathcal{T}$ are compatible with the boundary surface mesh $\mathcal{M}$ of $\mathcal{H}$ (see Fig. 8(b) for an illustration).

We then extrapolate the vector field $\mathbf{V}(\mathbf{x})$ and thereafter the governing field $G(\mathbf{x})$ in the newly generated tetrahedral mesh $\bar{\mathcal{T}}$. We first determine the vector $\mathbf{v}_e$ for each $e \in \bar{\mathcal{T}}$ by solving the problem $\arg\min_{\mathbf{v}_e} E_w$ with E_w defined in Eq. (4) while imposing the boundary condition: $\mathbf{v}_{e_a} = \mathbf{v}_{e_b}$ when $e_a \in \bar{\mathcal{T}}$, $e_b \in \mathcal{T}$ and $(e_a, e_b) \in \mathcal{N}$. Meanwhile, vectors in the elements intersected with (or next to) the platform $\mathcal{R}$ are enforced to be $\hat{\mathbf{z}} = (0, 0, -1)$ so that there is no collision between the printer head and $\mathcal{R}$ when printing the first layer. Similarly, the governing field $\bar{G}(\mathbf{x})$ can be obtained by solving Eq. (7) while fixing the field values on all nodes of $\mathcal{T}$ (see Fig. 8(b)).

5.2.3 *Slicing and trimming.*

When the same iso-values d_i is employed in $\bar{G}(\mathbf{x})$, the curved layer $\bar{S}_i$ that is compatible with $S_i \in \mathcal{H}$ can be extracted in the tetrahedral mesh $\bar{\mathcal{T}}$. The overhang on S_{i+1} (if there is any) will be fully supported by $\bar{S}_i$ and S_i together. In our implementation, we compute the α-shape of all points in $\mathcal{X}$ [Edelsbrunner and Mücke 1994] and use the polygonal mesh of the α-shape to further trim all $\{\bar{S}_i\}$ (see Fig.8(c)). As a result, the unnecessary regions are removed and the fabrication time can be further reduced.

5.3 Layer Thickness Control

The hardware of extruder in printer head allows to dynamically control the rate of filament extrusion, which enables the flexibility of fabricating layers with non-uniform thickness [Etienne et al. 2019]. However, there is still a limited range of layer thickness that can be stably realized as $[t_{\min}, t_{\max}]$. When generating curved layers by extracting iso-surfaces from $G(\cdot)$ and $\bar{G}(\cdot)$, we need to control the layer thickness. This is implemented by first applying the *extraction* step that satisfies the required minimal distance $t_{\min}$ between neighboring iso-surfaces and then conducting the *refinement* step to insert partial layers among those layers with distance larger than $t_{\max}$.

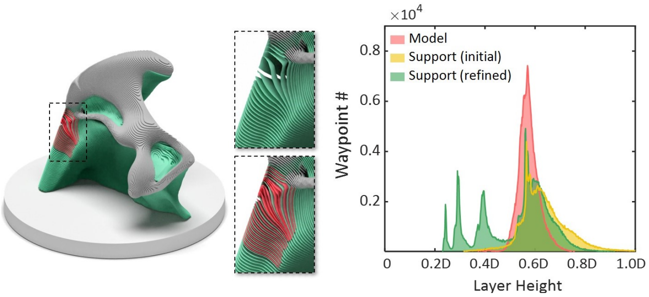


Fig. 9. The thickness variation of layers generated on the *Yoga* model before vs. after applying the (partial) refinement.

Given $g_{\min}$ and $g_{\max}$ as the minimal and the maximal field values in $G(\cdot)$, we incrementally determine the iso-values $\{d_i\}$ for surface extraction to ensure the requirement on $t_{\min}$.

(1) Define the first layer of $\mathcal{H}$ as $G(\mathbf{x}) = d_1$ with $d_1 = g_{\min} + c/2$.
(2) For every d_{i+1}, we initially define it as $d_{i+1} = d_i + c$. The minimal distance $D_{\min}$ from the current layer of iso-surface to the previous layer is checked, and the value of d_{i+1} is adjusted until $D_{\min} \geq t_{\min}$.
(3) The above step is repeatedly applied until $d_i > g_{\max}$, and the last layer is added as $d_n = g_{\max} - \epsilon$ with $\epsilon = 10^{-5}$ being used to avoid numerical degeneration. Note that, the layers of support are extracted in $\bar{G}(\mathbf{x})$ by using the same set of $\{d_i\}$.
(4) For the supporting structure defined in $\bar{G}(\mathbf{x})$ with field value less than $g_{\min}$, we can determine the layers in a similar method by descending iso-values starting from $g_{\min} - c/2$ until reaching the minimal value of $\bar{G}(\mathbf{x})$.

With the help of the magnitude control of $\nabla G(\cdot)$ imposed in Eq. (7), this algorithm can generate curved layers with a very small range of thickness variation. See the histogram of the *Yoga* model shown in Fig. 9 as an example, the variation of layer thickness falls in a very narrow region. However, the above layer extraction algorithm does not control the maximal distance between neighboring layers, which is solved by inserting additional layers at the necessary regions.

When the distance between two layers $G(x) = d_i$ and d_{i+1} is larger than $t_{\max}$ in some regions, the corresponding tetrahedra in these regions are stored in a set $\mathcal{T}_{ref}$. An additional (partial) layer can be inserted by extracting a new iso-surface $G(\mathbf{x}) = \frac{1}{2}(d_i + d_{i+1})$ as a triangular mesh surface in the tetrahedra of $\mathcal{T}_{ref}$. Triangles with distance to $G(x) = d_i$ or d_{i+1} less than $t_{\min}$ are removed from this new mesh surface. If the distance between this new surface to $G(x) = d_i$ and d_{i+1} is still larger than $t_{\max}$, another round of refinement is needed. In practice, the thickness requirement on hardware has $t_{\max} > 2t_{\min}$, which prevents forming the endless loop of layer refinement.

5.4 Toolpath Generation

The outcome of our computational framework is the toolpaths for every curved layer $\mathcal{S}_i$, where each toolpath is represented by a set of consecutive waypoints $\{(\mathbf{c}_k, \mathbf{n}_k, h_k)\}$. $\mathbf{c}_k$ and $\mathbf{n}_k$ indicate the position as a contact point and the orientation of printer head in the coordinate system of parts, and h_k defines the rate of material extrusion at $\mathbf{c}_k$. A hybrid strategy of directional-parallel and contour-parallel toolpaths is applied in our approach.

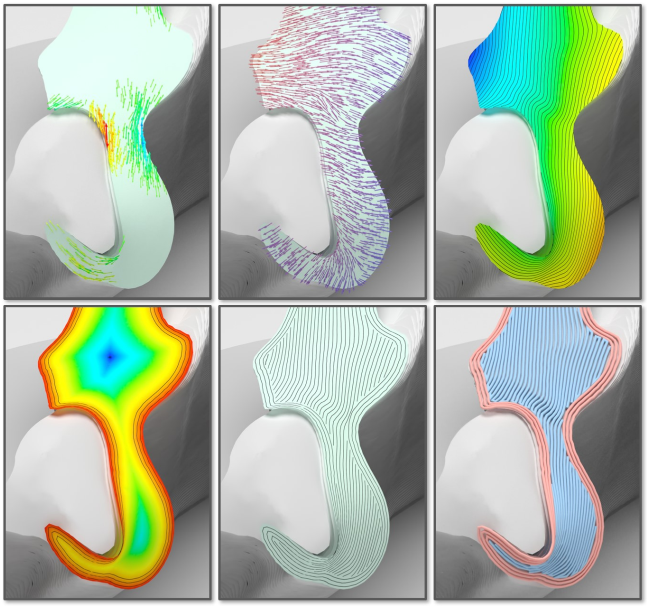


Fig. 10. A hybrid strategy is employed to generate toolpaths on each curved layer. (Top) From left to right, the projected vectors in the critical region, the optimized vector field $\mathbf{w}(\cdot)$, and the scalar field $P(\cdot)$ with its iso-curves as directional-parallel toolpaths. (Bottom) From left to right, the boundary distance field $B(\cdot)$ with its iso-curves as contour-parallel toolpaths, the directional-parallel toolpaths trimmed by $B(\cdot)$, and the finally connected toolpaths.

For each curved layer $\mathcal{S}_i$ represented by a triangular mesh surface, stress-oriented toolpaths are generated in the critical regions (i.e., $\mathcal{S}_i \cap \mathcal{T}^*$) together with directional-parallel toolpaths in other regions by using the iso-curves of a scalar field $P(\cdot)$. The field values of $P(\cdot)$ stored on the triangular nodes can be obtained by the methodology similar to the generation of $G(\cdot)$ in Section 4. Specifically, we first compute a vector-field $\mathbf{W}(\cdot)$ to indicate the gradient of $P(\cdot)$, where a vector $\mathbf{w}_f$ is defined on every triangle $f \in \mathcal{S}_i$. Again, this is formulated as an optimization problem

$$\arg\min_{\mathbf{w}_f} \sum_{(f_a, f_b) \in \mathcal{N}_{\mathcal{S}_i}} L_{a,b} \|\mathbf{w}_{f_a} - \mathbf{w}_{f_b}\|^2, \tag{11}$$

$$s.t. \quad \mathbf{w}_f = \hat{\mathbf{v}}_f \quad (\forall f \in \mathcal{S}_i \cap \mathcal{T}^*), \tag{12}$$

where $\mathcal{N}_{\mathcal{S}_i}$ is the set of neighboring triangles on $\mathcal{S}_i$, and $L_{a,b}$ is the length of an edge shared by f_a and f_b. $\hat{\mathbf{v}}_f$ is obtained by normalizing $\pm\tau_{\max}(e) \times \hat{\mathbf{n}}_f$ $(\forall f \in e)$, where $\hat{\mathbf{n}}_f$ is the normal of f and the sign is determined by the segmentation-based optimization as presented in Section 4.2. For those layers with $\mathcal{S}_i \cap \mathcal{T}^* = \emptyset$, the face f farthest to the boundary of $\mathcal{S}_i$ is employed in Eq. (12) to serve as anchor. After that, we determine $P(\cdot)$ as piecewise linear functions of node values $\{p_j\}$ by solving

$$\{p_j\} = \arg\min \sum_{f \in \mathcal{S}_i} A(f) \|\nabla P(\mathbf{x}_f) - \hat{\mathbf{w}}_f\|^2, \tag{13}$$

where $\mathbf{x}_f$ is the center of a triangular face f, and $\hat{\mathbf{w}}_f = c_w \mathbf{w}_f / \|\mathbf{w}_f\|$ is used to impose the demand on uniform width c_w between neighboring toolpaths. The stress-oriented and directional-parallel toolpaths can then be extracted on the triangular mesh of $\mathcal{S}_i$ as iso-curves, where the iso-values is incrementally changed by c_w from curve to curve. An illustration for these steps of toolpath generation is given in the top row of Fig. 10.

The contour-parallel toolpaths are generated with the help of a boundary distance field $B(\cdot)$ computed on the mesh surface of $\mathcal{S}_i$. Given n_c as a user specified number of contours, iso-contours $B(\mathbf{x}) = (k + 0.5)c_w$ are extracted with $k = 1, \cdots, n_c$ for contour-parallel toolpaths. For the directional-parallel toolpaths generated from $P(\cdot)$, the portions with boundary distance less than $n_c c_w$ are trimmed off (see the middle of bottom row in Fig. 10).

Lastly, the directional-parallel and the contour-parallel toolpaths are connected by the following method:

(1) For every endpoint $\mathbf{q}$ on each directional-parallel toolpath, we search and connect it to an endpoint $\mathbf{p}$ of the other directional-parallel toolpath if $\|\mathbf{pq}\| < \frac{3}{2}c_w$. This step is repeated until no more endpoint from directional-parallel toolpaths can be further connected.

(2) For every remaining endpoint $\mathbf{q}$, if there is a contour-parallel toolpath within the distance of $\frac{3}{2}c_w$, we break the contour-parallel toolpath at the closest point of $\mathbf{q}$ and connect $\mathbf{q}$ to this closest point.

An example result of this algorithm can be found in Fig. 10. After connecting the fragments of iso-curves into long toolpaths, we uniformly re-sample each toolpath to generate the positions of waypoints $\{\mathbf{c}_k\}$. The surface normal at $\mathbf{c}_k$ is employed as the orientation $\mathbf{n}_k$. The feedrate h_k is determined by the distance between $\mathbf{c}_k$ to the other curved surface layers (i.e., $\mathcal{S}_{i-1}$ and $\mathcal{S}_{i+1}$).

6 RESULTS AND DISCUSSION

6.1 Hardware

The physical fabrication of models presented in this paper is conducted on a multi-axis FDM 3D printer, the motion system of which is modified from a 5-axis CNC machine. As shown in Fig. 1, two rotational axes can be conducted on the printer's platform in addition to the x, y, z-axial motions given on the printer head. Moreover, the machine is equipped with two printer heads where the second one is used for printing water-soluble materials for supporting structures (see Fig. 14 for objects before and after removing the supporting structures).

The orientation of the printer head is kept along the z-axis to ensure the quality of material adhesion with the help of gravity. The waypoints on toolpaths generated by our approach are given in the coordinate system of parts. The corresponding motion in the configuration space needs to be determined by *inverse kinematics* (IK). Unlike the 6-DOF motion system employed in robot-assisted 3D printing, there is no redundancy in kinematics. As a consequence, singularity may not be avoided on some waypoints. Moreover, the solution determined by IK in the configuration space may have large variation on one rotational motor between two neighboring waypoints with small normal variation. Local perturbation is applied to the normals of waypoints to overcome the problem when necessary. For the cases that cannot be resolved by local perturbation, we make a break of toolpath at the corresponding point.

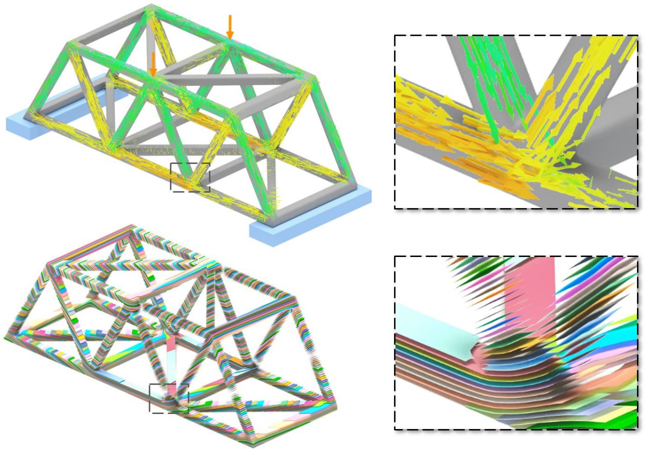


Fig. 11. Bridge example: computational result of curved layers for following the distribution of principal stresses.

Nozzles used in the printer heads of our system have the diameter $D = 1.0$mm. The eSUN 1.75mm PLA+ and Raise3D 1.75mm PVA filaments are employed to print models and supports respectively. According to [Etienne et al. 2019] and also our experimental tests, the variation of layer-thickness that can be effectively realized on this hardware setup falls in the range of $[0.2D, 0.8D]$ – i.e., $t_{\min} = 0.2$mm and $t_{\max} = 0.8$mm are employed in our computation.

6.2 Computational Results

We implemented our computational framework in C++. Source code and datasets of this work are available to the public[2]. All the computational experiments are obtained on a desktop PC with an Intel (R) Core TM i7-9700K CPU (6 cores @ 3.6GHz) + 32GB RAM, running Windows 10. The numerical library Eigen [Guennebaud et al. 2019] is employed as the solver of linear equations, and the PQP library [Gottschalk et al. 1996] is used for computing the point-to-surface distances. High-quality isotropic tetrahedral meshes are generated using a particle-based method [Zhong et al. 2019, 2018] for both the initial model and the region of supporting structures.

We tested our approach on a variety of models. The first example is the *Topo-opt* model in Fig. 1, discretized in 70.5k tets. The second and third models are Bunny head (60.3k tets) and Yoga (52.4k tets), shown in Figs. 3 and 4 respectively. We also tested on models with relatively regular shape but widely used in mechanical engineering: the C^2-model (46.5k tets) shown in Fig. 7 and the Bridge model (100.4k tets) shown in Fig. 11. All these models have large overhang regions after determining a setup orientation with optimal accessibility; therefore, curved layers for supporting structures are also generated by our computational framework, which are shown in the corresponding figures and Fig. 12.

Computational statistics on these models are given in Table 1. It can be found that the computation of all examples can be finished in around 8-16 minutes. The major bottleneck of our computation is

[2] https://github.com/GuoxinFang/ReinforcedFDM

Table 1. Computational statistics of our computational pipeline of reinforced FDM with multi-axis motion.

		Solid	Time (sec.) of Field-Opt. Slicing			Time (sec.) of Fabrication Enabling					Support	Total
Model	Fig.	#tets	Field $\mathbf{v}(\cdot)$	Field $G(\cdot)$	Layers	Re-Ort.	Relax.†	Support‡	Slice★	Toolpath	#tets	(sec.)
Topo-opt	1	70, 505	44.6	5.3	4.1	69.5	-	58.2	301.1	134.0	178.022	616.8
Bunny head	3	60, 375	10.4	1.1	3.1	41.9	-	55.4	270.2	87.6	162, 843	469.7
Yoga	4	52, 446	28.4	3.6	3.5	124.2	-	84.5	523.2	100.9	142, 183	868.3
C^2-model	7	46, 547	18.6	2.4	5.8	71.1	21.3	78.6	218.2	37.5	152, 799	453.5
Bridge	11	100, 420	67.2	5.2	9.1	40.2	-	189.3	420.3	211.0	394, 834	942.3

† The relaxation of vector-field is only applied to a model when necessary. The reported time includes the steps of vector-field update, governing field generation, layer extraction, and collision detection, which are applied in iterations.
‡ The time of support generation includes the sampling-based detection, the generation of a new tetrahedral mesh, and the computation of vector-field and scalar-field in the new volume mesh.
★ The time of slicing reported here includes the time of iso-surface extraction, adaptive refinement, the computation of α-shape, and the trimming of curved layers for supporting structures.

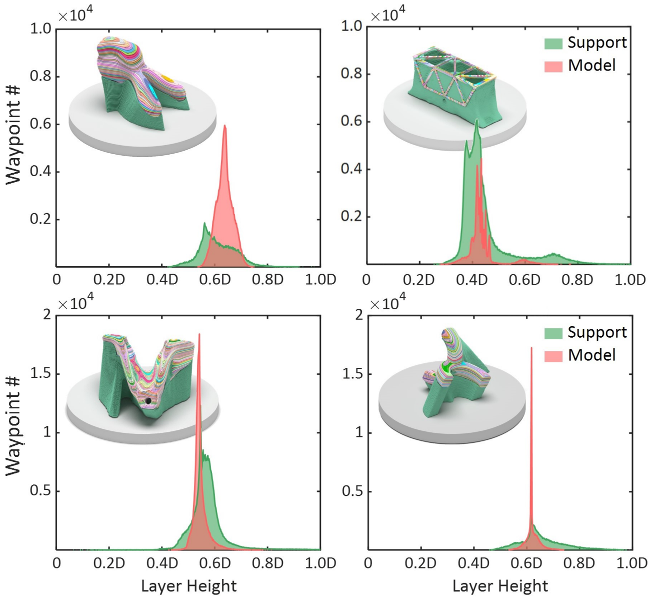


Fig. 12. The histogram of distance variation between curved layers on example models shown in this paper.

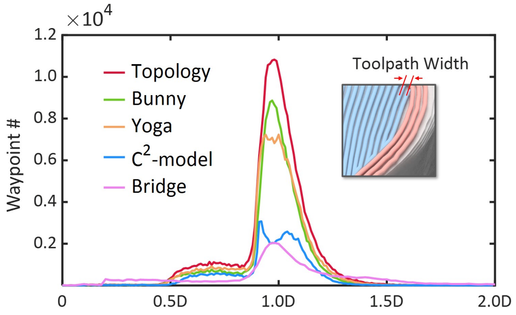


Fig. 13. The histogram of distance variation between toolpaths on example models shown in this paper.

Table 2. Statistics of physical fabrication.

	Conventional FDM			Multi-Axis FDM		
	Planar Layer #		Fab.	Curv. Layer #		Fab.
Model	Solid	Supt.	Time†	Solid	Supt.	Time†
Topo-opt	140	104	11.8 h	100	137	33 h
Bunny	131	77	7.1 h	100	89	14.5 h
Yoga	178	145	8.3 h	130	78	13.8 h
C^2-model	139	76	6.3 h	120	84	10.0 h

† The reported time (unit: hours) of fabrication includes the time for printing solid, the time for printing support and also the time for switching between two printer heads.

the step of slicing – around half of the total computing time is spent in this step. Our further study shows that the reconstruction of α-shape and trimming takes about $87-93\%$ of the slicing time. Besides, the step of toolpath generation is also relatively time-consuming as the governing fields need to be computed for all layers and the IK need to be solved for all waypoints, the amount of which is very large (e.g., around 730k waypoints in total for the Topo-opt model).

In order to ensure the good quality of fabrication, the variation of layer thickness should be controlled. We visualize the distributions of layer thicknesses for all the examples in Fig. 12. Specifically, we evaluate the minimal distances from all waypoints to the other layers represented as triangular meshes and plot the distribution of distances as a histogram. It is found that the variation of distances is relatively small after controlling the magnitude of $\nabla G(\cdot)$ in Eq. (7) and using the adaptive slicing algorithm. All the thicknesses fall in the range of $[t_{\min}, t_{\max}]$.

Besides, the variation of distances between neighboring toolpaths should also be controlled. For each waypoint $\mathbf{c}_k$ on a curved layer, we compute the distance from $\mathbf{c}_k$ to all toolpaths on the same layer except the portion directly connecting to $\mathbf{c}_k$. The distances computed on all waypoints are displayed as a distribution in Fig. 13 for every model. Again, by controlling the magnitude of $\nabla P(\cdot)$ in Eq. (13), the variation of distances between toolpaths generated by our approach is very small as more than 97.9% falling in the range of $1 \pm 0.5D$ where the nozzle diameter of the printer head is used as the standard width D.

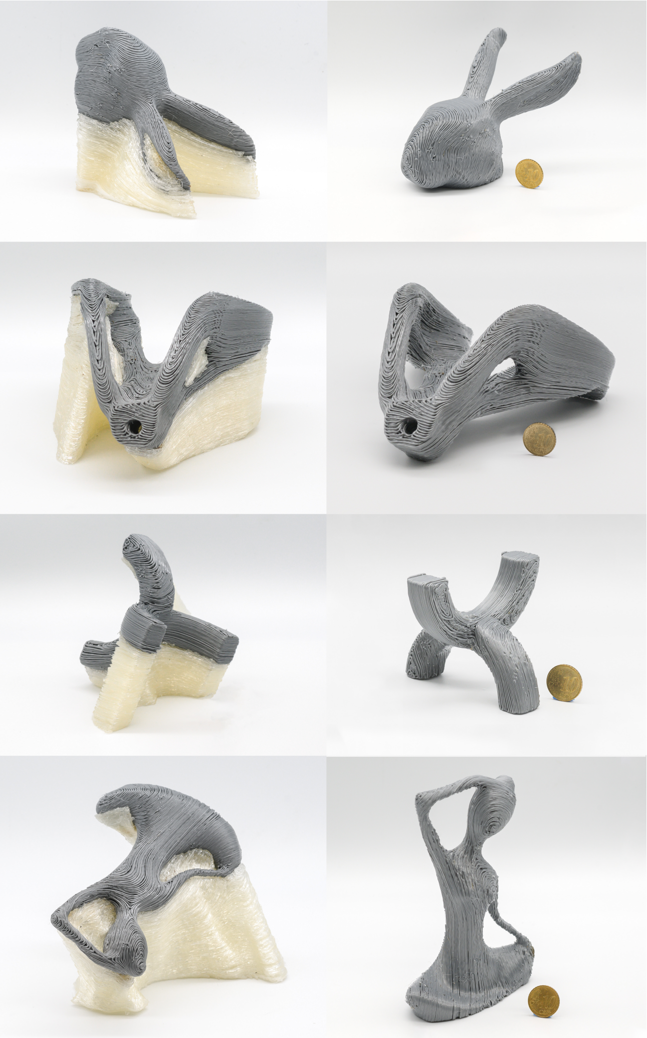
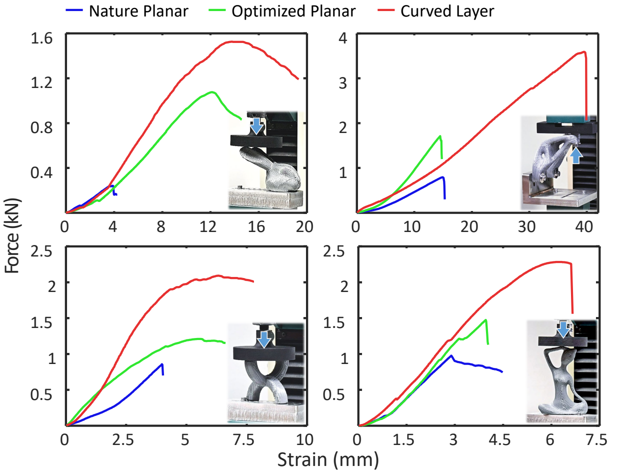


Fig. 14. Physical fabrication results (before and after removing the supporting structures) for Bunny head, Topo-opt, C^2-model and Yoga.



Model	Force F_{bk} at Failure Point (Unit: kN)		
	Planar Layers		Optimized Curved Layers
	Natural Ort.	Optimized Ort.	
Bunny head	0.24	1.07	1.52
Topo-opt	0.79	1.71	3.47
C^2-model	0.86	1.21	2.09
Yoga	0.98	1.47	2.28

Fig. 15. Comparison of tensile / compression tests conducted on Bunny head, Topo-opt, Yoga, and C^2-model. The breaking force F_{bk} in each test is captured at the peak of the strain-force curve. For planar-layer based 3D printing, models are fabricated with the natural orientation and also the orientation optimized by the method of [Ulu et al. 2015].

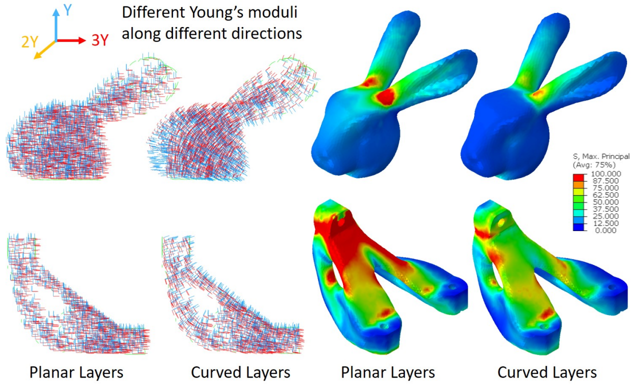


Fig. 16. Behavior verification of the optimized directions for filament alignment by using FEA software, Abaqus. The anisototric material property is specified at the element-level as shown in the left – different Young's moduli are assigned long different directions (i.e., $3Y$, $2Y$ and Y respectively with $Y = 782MPa$).

6.3 Physical Experiments and FEA Verification

Physical models have been fabricated by using the toolpaths generated by our framework on a multi-axis FDM 3D printer. The statistics of model fabrication are given in Table 2. For a given model, the fabrication time and the number of layers are reported for both the conventional planar-layer based FDM and the multi-axis FDM using curved layers. The fabrication time of planar-layer based FDM is also evaluated on the same machine by using 2.5D toolpaths. In general, the numbers of planar-layers and curved layers are comparable for the same model when using the same target layer thickness. However, the fabrication time of multi-axis FDM is longer. This is mainly caused by the complicated motion involved in multi-axis FDM. The results of fabrication are given in Fig. 14.

Tensile tests are conducted to verify the reinforcement of models fabricated by our method on a Tinius Olsen H5KS machine with specially designed fixture made by stainless steel. The comparisons are given on three models – the Topo-opt, the Bunny head, the Yoga and the C^2-model. As shown in Fig. 15, the breaking force of models fabricated by our method becomes 450% (Topo-opt), 635% (Bunny head), 132% (Yoga) and 243% (C^2-model) comparing to the models fabricated by planar-layer based FDM using the natural orientation. Moreover, the comparison is also given to the Topo-opt model and the Bunny head model by printing with optimized orientation in planar-layer based FDM [Ulu et al. 2015]. It is found that the breaking force can still be enhanced by 103% (Topo-opt), 42% (Bunny head), 55% (Yoga) and 73% (C^2-model), respectively – see also the strain-force curves and the statistics shown in Fig. 15.

On the other hand, FEA simulation with anisotropic material property is also conducted to evaluate the behavior of filament orientation computed by our method. Anisotropic behavior of different orientations of filaments is modeled at the element-level. For each tetrahedron, we assign the strongest modulus direction parallel to the filament orientation and the weakest modulus along the normal direction of a layer. The simulation results are identical to the physical test of overall strength enhancement can be observed (see Fig. 16). The FEA simulation (by using Abaqus software) for the bunny head model shows an overall reduction by 32.8% and 12.5% on maximal and average principal stresses after applying the anisotropic material orientation obtained from our optimized toolpaths for filament deposition. The simulation conducted on the Topo-opt model also verifies the performance of our method – with 46.1% and 20.3% reduction on the maximal and the average principal stresses respectively.

6.4 Limitation

The major limitation of our current approach is that the speed of fabrication is slower than the conventional planar-layer based FDM process. High-speed and stable rotation motions are difficult to be realized on motors employed in our hardware, the cost of which is only about 3 – 4 times of a high-end conventional FDM printer. Large momentum caused by the high-speed rotational motion of the printing platform and the printed object requires the system to use motors with large payload, which usually is hard to provide precise motions. Therefore, we have to sacrifice the speed to gain better precision. Meanwhile, more significant artifacts of staircase can be observed on models printed by our method than that in planar-layer based FDM. Although the main goal of our work is to strengthen the mechanical property, it also possible to switch to the objective function presented in [Etienne et al. 2019] to optimize the outer surface quality by curved printing.

Our ray-based method to determine the overhang regions needs supporting structures, which may generate rays having no intersection with any prior layers or the platform. When this occurs, we apply some local perturbation to adjust a ray's orientation until it can intersect with the platform. This is another limitation of our approach. It is worthy to investigate a better method for the support generation. Using optimized design (e.g., [Dumas et al. 2014]) can significantly reduce the time and material used to fabricate the supporting structures.

In our current formulation, only static loads are considered in FEA to generate the guidance of filament alignment. Nevertheless, even when applying fixed loads, the stress distribution could change after an object has already started to deform. Moreover, the loads applied to an object could change from time to time. It is important to incorporate the worst-case analysis (e.g., [Schumacher et al. 2018; Zhou et al. 2013]) for supervising the toolpath generation in our future work.

7 CONCLUSION

As a benefit of multi-axis 3D printing, we are allowed to design the paths of filament alignment inside an object so that the anisotropic strength of filaments can be controlled to reinforce the 3D printed object's strength under loads. A computational pipeline is introduced in this paper for generating curved layers and toolpaths according to the distribution of principal stresses. The most challenging part is how to incorporate the manufacturing constraints while optimizing the filament alignment.

We introduced field-based methodology to compute curved layers and toolpaths by extracting the iso-surfaces and iso-curves from the governing scalar-fields, which are optimized according to both the demand of mechanical strength and the constraints of manufacturing. To ease the computation of optimization, we first determine an optimized vector-field as an indication of the governing field's gradients. The final scalar-fields are computed from the normalized vector-fields. By this strategy, we convert the nonlinear optimization problem into a few linear optimization problems that can be solved efficiently and effectively. By extrapolating the governing field into the regions which need support, curved layers for supporting structures are generated. Moreover, algorithms are also developed to ensure the accessibility and control the variation of layer-thickness. Physical experiments have been conducted to verify the reinforced mechanical strength on objects fabricated according to the toolpaths generated by our framework.

The results of our experimental tests are very encouraging. Models fabricated by our method can withstand loads in 1.42 – 6.35× when comparing to the models fabricated by planar-layer based FDM. At present, we only consider the reinforcement of 3D printed models according to certain loads. We plan to explore the possible extension by considering the worst-case analysis in the future.

ACKNOWLEDGMENTS

The authors would like to thank the support from HKSAR RGC General Research Fund (GRF): CUHK/14202219 and National Science Foundation (NSF) under Grant Numbers IIS-1816511, OAC-1657364, OAC-1845962, and OAC-1910469. Guoxin Fang is supported in part by the China Scholarship Council.

REFERENCES

Sung-Hoon Ahn, Shad And, Paul Wright, Michael Montero, Dan Odell, and Shad Roundy. 2002. Anisotropic material properties of fused deposition modeling ABS. *Rapid Prototyping Journal* 8 (10 2002). https://doi.org/10.1108/13552540210441166

R. Allen and R. Trask. 2015. An experimental demonstration of effective Curved Layer Fused Filament Fabrication utilising a parallel deposition robot. *Additive Manufacturing* 8 (2015), 78–87.

Rahul Arora, Alec Jacobson, Timothy R. Langlois, Yijiang Huang, Caitlin Mueller, Wojciech Matusik, Ariel Shamir, Karan Singh, and David I.W. Levin. 2019. Volumetric Michell Trusses for Parametric Design & Fabrication. In *Proceedings of the 3rd ACM Symposium on Computation Fabrication (SCF '19)*. ACM, New York, NY, USA, 13.

Debapriya Chakraborty, B. Aneesh Reddy, and A. Roy Choudhury. 2008. Extruder Path Generation for Curved Layer Fused Deposition Modeling. *Comput. Aided Des.* 40, 2 (feb 2008), 235–243. https://doi.org/10.1016/j.cad.2007.10.014

Qiaodong Cui, Timothy Langlois, Pradeep Sen, and Theodore Kim. 2020. Fast and Robust Stochastic Structural Optimization. *Computer Graphics Forum* 39 (05 2020), 385–397. https://doi.org/10.1111/cgf.13938

Chengkai Dai, Charlie C. L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-Free Volume Printing by Multi-Axis Motion. *ACM Trans. Graph.* 37, 4, Article Article 134 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201342

Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. 2014. Bridging the Gap: Automated Steady Scaffoldings for 3D Printing. *ACM Trans. Graph.* 33, 4, Article Article 98 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601153

Herbert Edelsbrunner and Ernst P. Mücke. 1994. Three-Dimensional Alpha Shapes. *ACM Trans. Graph.* 13, 1 (Jan. 1994), 43–72. https://doi.org/10.1145/174462.156635

Gershon Elber. 2014. Accessibility in 5-axis milling environment. *Computer-Aided Design* 26, 11 (2014), 796–802.

Jimmy Etienne, Nicolas Ray, Daniele Panozzo, Samuel Hornus, Charlie C. L. Wang, Jonàs Martínez, Sara McMains, Marc Alexa, Brian Wyvill, and Sylvain Lefebvre. 2019. CurviSlicer: Slightly Curved Slicing for 3-Axis Printers. *ACM Trans. Graph.* 38, 4, Article Article 81 (July 2019), 11 pages. https://doi.org/10.1145/3306346.3323022

Miguel Fernandez-Vicente, Wilson Calle, Santiago Ferrandiz, and Andres Conejero. 2016. Effect of Infill Parameters on Tensile Mechanical Behavior in Desktop 3D Printing. *3D Printing and Additive Manufacturing* 3, 3 (2016), 183–192. https://doi.org/10.1089/3dp.2015.0036

Ian Gibson, David W. Rosen, and Brent Stucker. 2009. *Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing* (1st ed.). Springer Publishing Company, Incorporated.

Francisca Gil-Ureta, Nico Pietroni, and Denis Zorin. 2020. Reinforcement of General Shell Structures. *ACM Trans. Graph.* 39, 5, Article 153 (June 2020), 19 pages. https://doi.org/10.1145/3375677

S. Gottschalk, M. C. Lin, and D. Manocha. 1996. OBBTree: A Hierarchical Structure for Rapid Interference Detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, New York, NY, USA, 171–180. https://doi.org/10.1145/237170.237244

Jeroen P. Groen and Ole Sigmund. 2018. Homogenization-based topology optimization for high-resolution manufacturable microstructures. *Internat. J. Numer. Methods Engrg.* 113, 8 (2018), 1148–1163. https://doi.org/10.1002/nme.5575

Jeroen P. Groen, Jun Wu, and Ole Sigmund. 2019. Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill. *Computer Methods in Applied Mechanics and Engineering* 349 (2019), 722 – 742. https://doi.org/10.1016/j.cma.2019.02.031

Gael Guennebaud, Benoit Jacob, and et al. 2019. Eigen v3.3. http://eigen.tuxfamily.org.

Samuel Hornus, Tim Kuipers, Olivier Devillers, Monique Teillaud, Jonàs Martínez, Marc Glisse, Sylvain Lazard, and Sylvain Lefebvre. 2020. Variable-Width Contouring for Additive Manufacturing. *ACM Trans. Graph.* 39, 4, Article 131 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392448

Pengcheng Hu, Kai Tang, and Chen-Han Lee. 2013. Global obstacle avoidance and minimum workpiece setups in five-axis machining. *Computer-Aided Design* 45, 10 (2013), 1222 – 1237. https://doi.org/10.1016/j.cad.2013.05.007

Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. FrameFab: Robotic Fabrication of Frame Shapes. *ACM Trans. Graph.* 35, 6, Article Article 224 (Nov. 2016), 11 pages. https://doi.org/10.1145/2980179.2982401

G.Q. Jin, W.D. Li, and L. Gao. 2013. An Adaptive Process Planning Approach of Rapid Prototyping and Manufacturing. *Robot. Comput.-Integr. Manuf.* 29, 1 (Feb. 2013), 23–38. https://doi.org/10.1016/j.rcim.2012.07.001

Steven Keating and Neri Oxman. 2013. Compound Fabrication: A Multi-Functional Robotic Platform for Digital Design and Fabrication. *Robot. Comput.-Integr. Manuf.* 29, 6 (Dec. 2013), 439–448. https://doi.org/10.1016/j.rcim.2013.05.001

Yong-Joon Kim, Gershon Elber, Michael Bartoň, and Helmut Pottmann. 2015. Precise Gouging-Free Tool Orientations for 5-Axis CNC Machining. *Comput. Aided Des.* 58, C (Jan. 2015), 220–229. https://doi.org/10.1016/j.cad.2014.08.010

Timothy Langlois, Ariel Shamir, Daniel Dror, Wojciech Matusik, and David I. W. Levin. 2016. Stochastic Structural Analysis for Context-Aware Design and Fabrication. *ACM Trans. Graph.* 35, 6, Article 226 (Nov. 2016), 13 pages. https://doi.org/10.1145/2980179.2982436

T. Llewellyn-Jones, R. Allen, and R. Trask. 2016. Curved layer fused filament fabrication using automated tool-path generation. *3D Printing and Additive Manufacturing* 3, 4 (2016), 236–243.

Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-Last: Strength to Weight 3D Printed Objects. *ACM Trans. Graph.* 33, 4, Article Article 97 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601168

Yayue Pan, Chi Zhou, Yong Chen, and Jouni Partanen. 2014. Multitool and Multi-Axis Computer Numerically Controlled Accumulation for Fabricating Conformal Features on Curved Surfaces. *Journal of Manufacturing Science and Engineering* 136, 3 (03 2014). https://doi.org/10.1115/1.4026898

Sushrut Pavanaskar, Sushrut Pande, Youngwook Kwon, Zhongyin Hu, Alla Sheffer, and Sara McMains. 2015. Energy-efficient vector field based toolpaths for CNC pocketmachining. *Journal of Manufacturing Processes* 20 (2015), 314 – 320. https://doi.org/10.1016/j.jmapro.2015.06.009

Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI 16)*. Association for Computing Machinery, New York, NY, USA, 887–896. https://doi.org/10.1145/2858036.2858106

Stephen P. Radzevich and Erik D. Goodman. 2002. Computation of Optimal Workpiece Orientation for Multi-axis NC Machining of Sculptured Part Surfaces . *Journal of Mechanical Design* 124, 2 (05 2002), 201–212. https://doi.org/10.1115/1.1468634

Jaret C. Riddick, Mulugeta A. Haile, Ray Von Wahlde, Daniel P. Cole, Oluwakayode Bamiduro, and Terrence E. Johnson. 2016. Fractographic analysis of tensile failure of acrylonitrile-butadiene-styrene fabricated by fused deposition modeling. *Additive Manufacturing* 11 (2016), 49 – 59. https://doi.org/10.1016/j.addma.2016.03.007

Christian Schumacher, Jonas Zehnder, and Moritz Bächer. 2018. Set-in-Stone: Worst-Case Optimization of Structures Weak in Tension. *ACM Trans. Graph.* 37, 6, Article Article 252 (Dec. 2018), 13 pages. https://doi.org/10.1145/3272127.3275085

Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. 2012. Stress Relief: Improving Structural Strength of 3D Printable Objects. *ACM Trans. Graph.* 31, 4, Article Article 48 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185544

John C. Steuben, Athanasios P. Iliopoulos, and John G. Michopoulos. 2016. Implicit Slicing for Functionally Tailored Additive Manufacturing. *Comput. Aided Des.* 77, C (Aug. 2016), 107–119. https://doi.org/10.1016/j.cad.2016.04.003

Kam-Ming Mark Tam and Caitlin T Mueller. 2017. Additive Manufacturing Along Principal Stress Lines. *3D Printing and Additive Manufacturing* 4, 2 (2017), 63–81. https://doi.org/10.1089/3dp.2017.0001

G.M. Treece, R.W. Prager, and A.H. Gee. 1999. Regularised marching tetrahedra: improved iso-surface extraction. *Computers Graphics* 23, 4 (1999), 583 – 598. https://doi.org/10.1016/S0097-8493(99)00076-X

Erva Ulu, Emrullah Korkmaz, Kubilay Yay, O. Burak Ozdoganlar, and Levent Burak Kara. 2015. Enhancing the Structural Performance of Additively Manufactured Objects Through Build Orientation Optimization. *Journal of Mechanical Design* 137, 11 (10 2015). https://doi.org/10.1115/1.4030998 111410.

Nobuyuki Umetani and Ryan Schmidt. 2013. Cross-Sectional Structural Analysis for 3D Printing Optimization. In *SIGGRAPH Asia 2013 Technical Briefs (SA '13)*. Association for Computing Machinery, New York, NY, USA, Article Article 5, 4 pages. https://doi.org/10.1145/2542355.2542361

J. Vanek, J. A. G. Galicia, and B. Benes. 2014. Clever Support: Efficient Support Structure Generation for Digital Fabrication. In *Proceedings of the Symposium on Geometry Processing (SGP '14)*. Eurographics Association, Goslar, DEU, 117–125. https://doi.org/10.1111/cgf.12437

Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. 2013. Cost-Effective Printing of 3D Objects with Skin-Frame Structures. *ACM Trans. Graph.* 32, 6, Article Article 177 (Nov. 2013), 10 pages. https://doi.org/10.1145/2508363.2508382

Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. 2018. Infill Optimization for Additive Manufacturing – Approaching Bone-like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (February 2018), 1127–1140. https://doi.org/10.1109/TVCG.2017.2655523

Rundong Wu, Huaishu Peng, François Guimbretière, and Steve Marschner. 2016. Printing Arbitrary Meshes with a 5DOF Wireframe Printer. *ACM Trans. Graph.* 35, 4, Article Article 101 (July 2016), 9 pages. https://doi.org/10.1145/2897824.2925966

Fubao Xie, Lufeng Chen, Zhaoyu Li, and Kai Tang. 2020. Path smoothing and feed rate planning for robotic curved layer additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 65 (2020), 101967. https://doi.org/10.1016/j.rcim.2020.101967

Ke Xu, Yingguang Li, Lufeng Chen, and Kai Tang. 2019. Curved layer based process planning for multi-axis volume printing of freeform parts. *Computer-Aided Design* 114 (2019), 51 – 63. https://doi.org/10.1016/j.cad.2019.05.007

Xiaolong Zhang, Yang Xia, Jiaye Wang, Zhouwang Yang, Changhe Tu, and Wenping Wang. 2015. Medial Axis Tree-an Internal Supporting Structure for 3D Printing. *Comput. Aided Geom. Des.* 35, C (May 2015), 149–162. https://doi.org/10.1016/j.cagd.2015.03.012

Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2016. Connected Fermat Spirals for Layered Fabrication. *ACM Trans. Graph.* 35, 4, Article Article 100 (July 2016), 10 pages. https://doi.org/10.1145/2897824.2925958

Haisen Zhao, Hao Zhang, Shiqing Xin, Yuanmin Deng, Changhe Tu, Wenping Wang, Daniel Cohen-Or, and Baoquan Chen. 2018. DSCarver: Decompose-and-Spiral-Carve for Subtractive Manufacturing. *ACM Trans. Graph.* 37, 4, Article Article 137 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201338

Sikai Zhong, Zichun Zhong, and Jing Hua. 2019. Surface reconstruction by parallel and unified particle-based resampling from point clouds. *Computer Aided Geometric Design* 71 (2019), 43 – 62. https://doi.org/10.1016/j.cagd.2019.04.011

Zichun Zhong, Wenping Wang, Bruno Lévy, Jing Hua, and Xiaohu Guo. 2018. Computing a High-Dimensional Euclidean Embedding from an Arbitrary Smooth Riemannian Metric. *ACM Trans. Graph.* 37, 4, Article 62 (July 2018), 16 pages. https://doi.org/10.1145/3197517.3201369

Qingnan Zhou, Julian Panetta, and Denis Zorin. 2013. Worst-Case Structural Analysis. *ACM Trans. Graph.* 32, 4, Article Article 137 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461967