

More Stiffness with Less Fiber: End-to-End Fiber Path Optimization for 3D-Printed Composites

Xingyuan Sun
Princeton University
Princeton, NJ, USA
xs5@princeton.edu

Geoffrey Roeder
Princeton University
Princeton, NJ, USA
roeder@princeton.edu

Tianju Xue
Hong Kong University of Science and
Technology
Hong Kong, China
cetxue@ust.hk

Ryan P. Adams
Princeton University
Princeton, NJ, USA
rpa@princeton.edu

Szymon Rusinkiewicz
Princeton University
Princeton, NJ, USA
smr@princeton.edu

ABSTRACT

In 3D printing, stiff fibers (*e.g.*, carbon fiber) can reinforce thermo-plastic polymers with limited stiffness. However, existing commercial digital manufacturing software only provides a few simple fiber layout algorithms, which solely use the geometry of the shape. In this work, we build an automated fiber path planning algorithm that maximizes the stiffness of a 3D print given specified external loads. We formalize this as an optimization problem: an objective function is designed to measure the stiffness of the object while regularizing certain properties of fiber paths (*e.g.*, smoothness). To initialize each fiber path, we use finite element analysis to calculate the stress field on the object and greedily “walk” in the direction of the stress field. We then apply a gradient-based optimization algorithm that uses the adjoint method to calculate the gradient of stiffness with respect to fiber layout. We compare our approach, in both simulation and real-world experiments, to three baselines: (1) concentric fiber rings generated by Eiger, a leading digital manufacturing software package developed by Markforged, (2) greedy extraction on the simulated stress field (*i.e.*, our method without optimization), and (3) the greedy algorithm on a fiber orientation field calculated by smoothing the simulated stress fields. The results show that objects with fiber paths generated by our algorithm achieve greater stiffness while using less fiber than the baselines—our algorithm improves the Pareto frontier of object stiffness as a function of fiber usage. Ablation studies show that the smoothing regularizer is needed for feasible fiber paths and stability of optimization, and multi-resolution optimization helps reduce the running time compared to single-resolution optimization.

CCS CONCEPTS

• **Applied computing** → **Computer-aided manufacturing**.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SCF '23, October 8–10, 2023, New York City, NY, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0319-5/23/10.

<https://doi.org/10.1145/3623263.3623356>

KEYWORDS

3D Printing, Continuous Fiber, Path Planning, End-to-End Differentiable, Optimization

ACM Reference Format:

Xingyuan Sun, Geoffrey Roeder, Tianju Xue, Ryan P. Adams, and Szymon Rusinkiewicz. 2023. More Stiffness with Less Fiber: End-to-End Fiber Path Optimization for 3D-Printed Composites. In *Symposium on Computational Fabrication (SCF '23)*, October 8–10, 2023, New York City, NY, USA. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3623263.3623356>

1 INTRODUCTION

Additive manufacturing has revolutionized the ability to fabricate three-dimensional objects of high geometric complexity, with a variety of applications including in healthcare, automotive, and aerospace industries [Shahrubudin et al. 2019]. However, the increasing flexibility in manufacturing has outstripped our ability to produce designs that optimally take advantage of 3D printers. This has motivated research on *computational fabrication* pipelines that augment human specification of goals with computational optimization of designs that best realize those goals, for problems ranging from ensuring structural integrity through controlling appearance and fine-tuning the fabrication process [Attene et al. 2018].

In this work, we address the problem of producing structurally-sound parts that are capable of bearing nontrivial load. We aim to exploit the capabilities of devices such as the Markforged Mark Two [Markforged 2023a], which is based on conventional fused deposition modeling (FDM) using thermoplastic nylon, but augments this with the ability to extrude and deposit continuous fibers. Options for the latter include carbon fiber, Kevlar, fiberglass, and HSHT (High Strength High Temperature) fiberglass, all of which offer the ability to selectively strengthen printed parts with respect to tensile loads. In effect, this process creates fiber-reinforced plastic (FRP) composites [Kabir et al. 2020], but with the ability to control fiber placement to achieve specific tradeoffs in strength, weight, and cost.

The optimization of fiber layout is similar to problems traditionally considered in computational fabrication, such as topology optimization (*i.e.*, removing material from certain regions) and spatially-varying assignment of different materials. Systems for

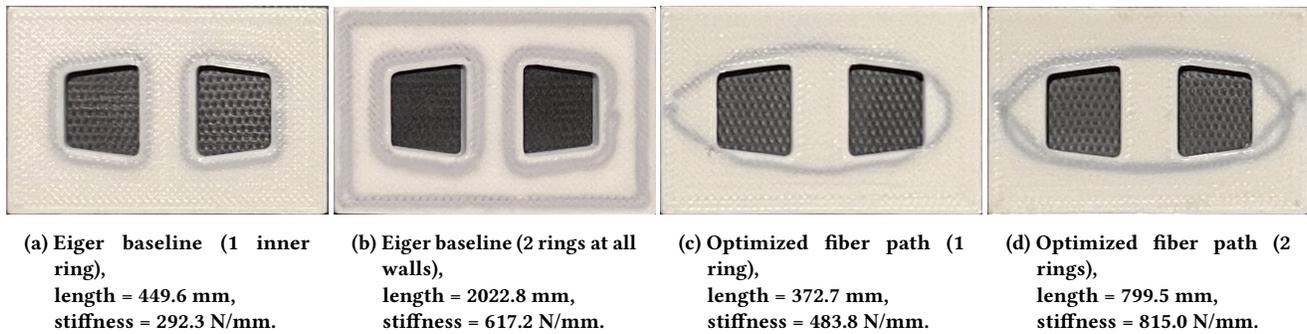


Figure 1: Planned and 3D printed fiber paths with fiber lengths and average stiffness measured over four batches annotated, for a part with external tension applied between two holes. (a) (b): Concentric fiber rings generated by the Eiger baseline only consider geometry. (c) (d): Our optimized fiber paths, tuned for the applied loads, yield greater stiffness at lower fiber lengths.

these latter tasks are typically based on Eulerian analysis and optimization, in which a quantity (density, material choice, *etc.*) is determined for each location in space (*e.g.*, on a voxel grid). Similarly, almost all existing methods for optimizing carbon fiber composites focus on the spatially-varying fiber direction field, then use variants of greedy extraction or ODE solvers to extract the fiber paths themselves [Schmidt et al. 2020; Wang et al. 2021].

In contrast, we are inspired by a Lagrangian point of view: we characterize the strength of the part as a function of the fiber path, compute gradients with respect to changes in path coordinates, and optimize the fiber path directly using gradient descent. This strategy is based on the adjoint method [Cao et al. 2003; Errico 1997], commonly used for PDE-constrained optimization, and exploits modern systems for automatic differentiation [Griewank and Walther 2008], which have evolved considerably in recent years to support a range of machine learning and general optimization problems. Our end-to-end optimization approach has the benefit of focusing directly on the final goal—maximizing stiffness with respect to external loads—rather than on indirect objectives such as minimizing strain throughout the object.

We incorporate our gradient descent-based optimization into a complete system that addresses three key challenges: (1) solving for the stress field of the object given external loads, (2) computing an optimization objective and its gradient based on the stress field, and (3) providing a good initialization of fiber layout for our local optimizer. To address the first challenge, we model the composite material using the linear elastic model, and approximately solve the PDE using the finite element method. Without loss of generality and for the sake of simplicity, we model the composite material in two dimensions under the assumption of in-plane stress (*i.e.*, we only consider laminates). We also simplify the problem by considering Dirichlet (fixed-displacement) boundary conditions. To address the second challenge, we design an objective function based on total strain energy given the boundary conditions: under the assumption of linear elasticity, maximizing this energy is equivalent to maximizing the object’s stiffness. We regularize the objective to ensure that the optimized fiber paths are feasible. Finally, to address the last challenge, we initialize each fiber path by greedily following the directions of maximum tensile stress (or perpendicular to the direction of maximum compressive stress). We further use a

multi-resolution approach inspired by multigrid methods, to reduce the running time of the optimization.

We show designs produced by our method on a number of illustrative case studies, demonstrating that our method yields higher stiffness with less fiber as compared to baseline paths produced by the Eiger software by Markforged [2023b]. We compare our results to greedy extraction based on either the stress field or optimized fiber direction field, as well as other ablations including omitting regularization or multi-resolution optimization. We print our designs (see Figure 1), using the method of Sun et al. [2021] to compute fiber extruder paths that compensate for fiber stiffness. Finally, we test our printed parts to verify that our method matches the predicted stiffness in the real world (subject to inherent print-to-print variations in material strength).

2 RELATED WORK

2.1 Fiber orientation optimization in 3D printing

A task that is similar to fiber path planning is fiber orientation optimization, where researchers discretize space into elements and optimize fiber orientations in them. Additional steps, such as greedy extraction, ODE solvers, or geometric methods, must be performed to produce fiber paths from the orientation field. Thus fiber orientation optimization can serve as the first step of fiber path planning, which we will discuss in § 2.2. See Hu [2021] for a survey (called “free material optimization”). The most common approach for orientation optimization is to set density and orientation as design variables and optimize an objective such as compliance [Chu et al. 2021; da Silva et al. 2020; Jung et al. 2022] or the Tsai–Wu failure criterion [Ma et al. 2022] with a gradient-based optimizer. To address the checkerboard pattern issue (periodicity of the orientation variables), researchers usually use filtering [Andreassen et al. 2011] to smooth the design variable field (*e.g.*, through a weighted average of neighboring elements). Another choice of design variable is the lamination parameters: Shafiqfard et al. [2019] and Demir et al. [2019] proposed to first optimize lamination parameters, search for the best fitting fiber orientations from the optimized lamination parameters, and then perform an optimization on the orientation field while considering manufacturing constraints (*e.g.*,

curvature). There are also iterative variants. For example, Caivano et al. [2020] proposed iterating between calculating the principal stress direction and updating the material distribution until convergence. While mainly concentrating on orientation optimization, some approaches do ultimately generate fiber paths. For example, Fedulov et al. [2021] first optimized density and orientation and then used third-party software for printing trajectory generation; Schmidt et al. [2020] performed density and orientation optimization and generated streamlines using the 4th-order Runge-Kutta integrator for visualization.

2.2 Fiber path planning in 3D printing

A variety of path planning algorithms have been proposed for continuous fiber-reinforced plastics—see Zhang et al. [2020] for a survey. The most common approach is to first perform an optimization (topology, orientation, *etc.*), and then extract fiber paths from the result. As discussed, orientation optimization is one choice of the optimization (*i.e.*, use density and orientation as the design variables), but there are different methods for path extraction. Wang et al. [2021] proposed to “walk” in the field along with the stress direction and consider the angle turned in every move to produce smoothed paths. Papapetrou et al. [2020] described three methods for path extraction: the offset method and the EQS (Equally-Space) method use the geometry of the optimized layout, and the streamline method fits the orientation field with streamlines. Safonov [2019] proposed to alternate between topology optimization and fiber orientation updates using an evolutionary heuristic method.

There are also more potential choices for the design variable. For example, one choice is to only optimize the density. Li et al. [2021] performed topology optimization of material density (without orientation) using regularizers that force the fiber material to form lines. However, they did not extract fiber paths explicitly at the end, so it is unclear whether the fibers are directly printable. Li et al. [2020] and Chen and Ye [2021] proposed to lay fibers along with the load transmission trajectories. Almeida Jr. et al. [2019] proposed to perform the SIMP (Solid Isotropic Material with Penalization) method first, designed the fiber pattern manually, and then used a genetic algorithm to determine the number of fiber rings/paths that would minimize compliance (defined as mass divided by stiffness). Sugiyama et al. [2020] proposed to calculate the stress field and update fiber paths so that they follow the direction of maximum principal stress, repeating this process until convergence. Apart from these two-stage approaches, there are also end-to-end approaches based on genetic algorithms. For example, Yamanaka et al. [2016] modeled fiber paths as streamlines and optimized them directly using a genetic algorithm.

In summary, most existing works perform fiber planning in two stages (topology/orientation optimization followed by path extraction). In contrast, our method performs an end-to-end optimization of the fiber layout, maximizing the regularized object stiffness via a gradient-based optimizer.

2.3 PDE-constrained optimization

Also related to the problem of optimizing geometry to maximize stiffness is the area of PDE-constrained optimization, in which an optimization problem is subjected to physical constraints expressed

via partial differential equations (PDEs) [De los Reyes 2015]. There are two common types of algorithms to solve PDE-constrained optimization problems: *all-at-once* and *black-box* [Herzog and Kunisch 2010]. *All-at-once* treats both the design variable and the state variable as independent variables, so the method may not satisfy the constraints before the optimization finishes. A common *all-at-once* algorithm is SQP (sequential quadratic programming) [Boggs and Tolle 1995]. A disadvantage of the *all-at-once* approach is the dimension of the state variable can be very large, which makes the optimization costly. *Black-box* solves the problem in reduced form, by treating the design variable as the only independent variable, so that a gradient-based optimizer can be applied (*e.g.*, gradient descent, Newton’s method). We formalize the fiber path planning task as a PDE-constrained optimization problem and use the *black-box* approach, specifically the adjoint method, to solve it.

3 METHOD

The pipeline of our method is shown in Figure 2. Starting from a goal (a shape with some external loads), we first simulate the stress field using the finite element method (§ 3.1). We apply a greedy fiber extraction algorithm by “walking” in the stress field, and then downsample the greedy path (§ 3.2). We build and optimize an objective function based on the object’s stiffness and regularity conditions of the fiber paths, using the adjoint method to calculate the gradients of the objective (§ 3.3). These steps can be repeated until a desired number of fiber paths are extracted and optimized. We then perform a coarse-to-fine optimization by upsampling and optimizing the fiber paths a specified number of times (§ 3.4).

3.1 Simulation

In this subsection, we describe how we solve the stress field given a shape, some external loads, and a specified fiber layout. We denote the body as Ω , the stress tensor as σ , the strain tensor as ϵ , the displacement vector as \mathbf{u} , and the stiffness tensor as \mathbf{C} . The linear elastic model can be written as

$$\begin{aligned} -\nabla \cdot \sigma &= f, \\ \epsilon &= \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top), \\ \sigma &= \mathbf{C} : \epsilon, \end{aligned} \quad (1)$$

where the colon is the dot product. f is the body force and we set it to 0. For certain regions on the boundary of Ω (*i.e.*, $\partial\Omega$), the value of \mathbf{u} is given as input (*i.e.*, Dirichlet boundary condition). For the remaining regions, we have $\sigma \cdot \mathbf{n} = T$ (*i.e.*, Neumann boundary condition), where \mathbf{n} is the outward unit normal vector, and T is the tractive force which we set to 0.

The constitutive equations $\sigma = \mathbf{C} : \epsilon$ can also be written in a matrix product form; under the assumption of in-plane stress, we have

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \frac{E_1}{1-\nu_{21}\nu_{12}} & \frac{E_1\nu_{21}}{1-\nu_{21}\nu_{12}} & 0 \\ \frac{E_2\nu_{12}}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{bmatrix}, \quad (2)$$

where E_1 and E_2 are Young’s moduli, ν_{12} and ν_{21} are the Poisson’s ratios, and μ is the shear modulus. For simplicity, we assume both plastic and fiber are isotropic materials, and they have different Young’s moduli E_{plastic} and E_{fiber} and identical Poisson’s ratio ν .

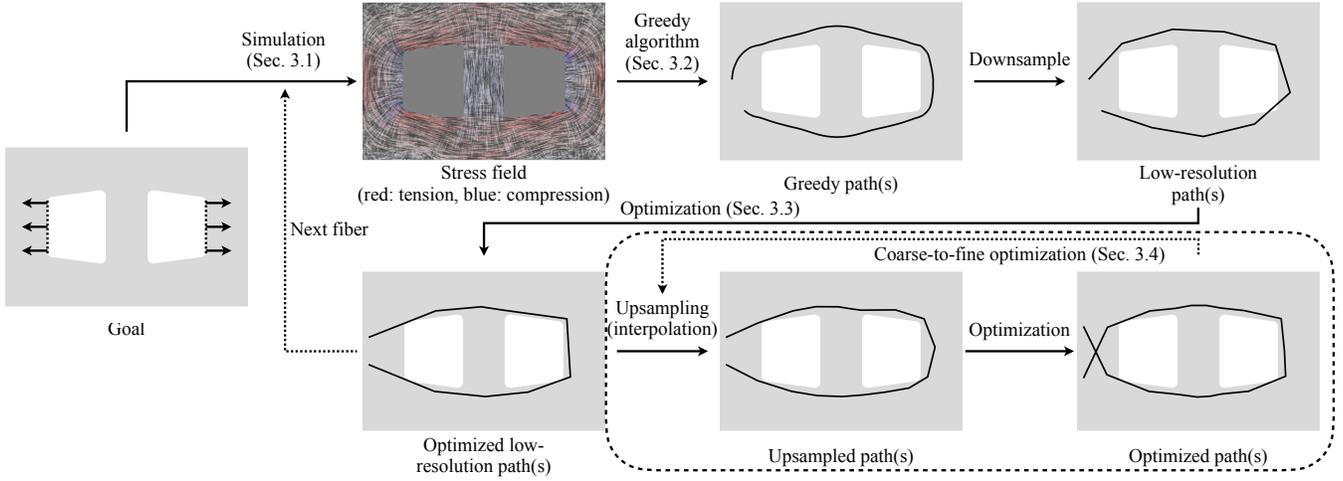


Figure 2: We repeatedly use the finite element method to calculate the stress field of the object (§ 3.1), extract a new fiber path by greedily “walking” on the stress field (§ 3.2), optimize the downsampled fiber path with an objective function designed to maximize stiffness and regularize fiber paths to be manufacturable (§ 3.3), and finally upsample and optimize all the fiber paths several times to perform coarse-to-fine optimization (§ 3.4).

The next issue is to calculate the Young’s modulus field. Consider a laminate of height h_{object} , with some layers filled with just plastic and others containing both plastic and fiber. We assume that all layers with fiber, adding up to a total height of h_{fiber} , have identical fiber paths, and omit plastic where fiber is present. The set of fiber paths is denoted as P , and every path p in it is a sequence of vertices on the fiber path. For a point $x \in \Omega$, for the purpose of differentiability, we define its “soft” Young’s modulus as

$$E(x) := E_{\text{plastic}} \cdot \alpha_{\text{plastic}}(x) + E_{\text{fiber}} \cdot \alpha_{\text{fiber}}(x), \quad (3)$$

where

$$\alpha_{\text{fiber}}(x) := \sum_{p \in P} \exp\left(-\left(\frac{\text{dis}(p, x)}{w_{\text{fiber}}/2}\right)^2\right) \cdot h_{\text{fiber}}, \quad (4)$$

where $w_{\text{fiber}} = 0.9$ mm is the width of the fiber, $\text{dis}(\cdot, \cdot)$ measures the distance between a point and a path, and

$$\alpha_{\text{plastic}}(x) := h_{\text{object}} - \min(\alpha_{\text{fiber}}(x), h_{\text{fiber}}). \quad (5)$$

We allow fiber paths to overlap in this setting, as even in real prints from the Markforged Mark Two, we do not observe any problems. We then have $\mu(x) = \frac{E(x)}{2(1+\nu)}$. Finally, we solve the PDE in Equation 1 using FEniCS with DOLFIN [Logg and Wells 2010] by solving its first-order condition on linear triangular finite elements. Figure 2 visualizes an example of the calculated stress field, using line integral convolution [Cabral and Leedom 1993].

3.2 Greedy fiber path extraction

In this subsection, we describe how we greedily extract a fiber path from a stress field along the directions of maximum tensile stress or perpendicular to the direction of maximum compressive stress. With the stress tensor σ calculated from § 3.1, we first calculate the

stress on plastic:

$$\sigma_{\text{plastic}} := \sigma \cdot \frac{E_{\text{plastic}} \cdot \alpha_{\text{plastic}}}{E_{\text{plastic}} \cdot \alpha_{\text{plastic}} + E_{\text{fiber}} \cdot \alpha_{\text{fiber}}}. \quad (6)$$

For the stress matrix at any point $x \in \Omega$, we can calculate its eigenvalue with the largest absolute value $\lambda(x)$ and the corresponding eigenvector $\mathbf{v}(x)$. We then build a scalar field with $|\lambda(x)|$ and randomly sample a starting point x_0 with the field as sampling weights. From the starting point, we walk in both directions along with $\pm\mathbf{v}(x_0)$ (or perpendicular to $\mathbf{v}(x_0)$ if $\lambda(x_0)$ is negative) at a fixed step size of 0.5 mm. If we walk outside Ω or within 1.3 mm to $\partial\Omega$ (number measured from prints from Eiger), we retry at most 19 times with a random rotation uniformly sampled between $-\pi/12$ to $\pi/12$. The algorithm stops when a preset length limit is reached, or we cannot walk in both directions even after retries.

We then downsample the extracted fiber path by keeping 1 of every 20 vertices. We iterate every subsequence of the downsampled path and select the one that minimizes the objective function we will define in Section 3.3. We repeat this process 10 times (sampling 10 starting points) and keep the one that minimizes the objective function.

3.3 Gradient calculation and optimization

In this subsection, we describe how we design an objective function and minimize it using a gradient-based optimizer. We denote the optimized strain energy in Equation 1 as U , and the set of fiber paths as P . The objective $\mathcal{L}(P)$ is defined as

$$-U + \sum_{p \in P} \left(w_{\text{lap}} \cdot \mathcal{L}_{\text{lap}}(p) + w_{\text{min}_l} \cdot \mathcal{L}_{\text{min}_l}(p) + w_{\text{bdy}} \cdot \mathcal{L}_{\text{bdy}}(p) \right), \quad (7)$$

where w_{lap} , $w_{\text{min_len}}$, and w_{bdy} are hyper-parameters. The Laplacian regularizer \mathcal{L}_{lap} penalizes non-smooth fiber paths:

$$\mathcal{L}_{\text{lap}}(p) := s(P)^3 \cdot \sum_{i=2}^{|p|-1} \left\| p_i - \frac{p_{i-1} + p_{i+1}}{2} \right\|^2, \quad (8)$$

where $s(P)$ is a count of the total number of segments in all fiber paths (*i.e.*, $\sum_{p \in P} |p| - |P|$). The reason to apply the $s(P)^3$ multiplier is because the Laplacian regularizer is sensitive to upsampling, which we discuss in § 3.4, and this multiplier keeps our Laplacian regularizer scale-invariant. The minimum-length regularizer $\mathcal{L}_{\text{min_l}}$ penalizes infeasibly-short fiber paths:

$$\mathcal{L}_{\text{min_l}}(p) := \max \left(l_{\text{min}} - \sum_{i=2}^{|p|} \|p_i - p_{i-1}\|, 0 \right)^2, \quad (9)$$

where l_{min} is the minimum fiber length that can be printed by the 3D printer. The boundary regularizer \mathcal{L}_{bdy} penalizes fiber paths outside Ω or too close to $\partial\Omega$:

$$\mathcal{L}_{\text{bdy}}(p) := \sum_i \max(d_{\text{min}} - \text{dis}(p_i, \Omega), 0)^2, \quad (10)$$

where $\text{dis}(p_i, \Omega)$ measures the distance from p to $\partial\Omega$ (positive for $p_i \in \Omega$, negative otherwise) and d_{min} is the lower limit of distance from fiber to the boundary.

The next step is to calculate $\frac{d\mathcal{L}(P)}{dP}$. Here we apply the adjoint method. Denote the first-order condition of Equation 1 as $F(\mathbf{u}, P) = 0$. By the implicit function theorem (under proper regularity conditions) \mathbf{u} can be thought of a function of P , and the derivative $\frac{d\mathbf{u}}{dP}$ is well-defined. Taking the derivative of F with respect to P , we have

$$\frac{dF}{dP} = \frac{\partial F}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dP} + \frac{\partial F}{\partial P} = 0, \quad (11)$$

which leads to

$$\frac{d\mathcal{L}(P)}{dP} = - \frac{\partial \mathcal{L}(P)}{\partial \mathbf{u}} \left(\frac{\partial F}{\partial \mathbf{u}} \right)^{-1} \frac{\partial F}{\partial P} + \frac{\partial \mathcal{L}(P)}{\partial P}. \quad (12)$$

We implement this end-to-end differentiation automatically using dolfin-adjoint [Mitusch et al. 2019] and PyTorch [Paszke et al. 2019]. We use the BFGS implementation in SciPy [Virtanen et al. 2020] to minimize $\mathcal{L}(P)$, and again we iterate every subsequence of the optimized path and select the one that minimizes $\mathcal{L}(P)$. We can repeat the steps in § 3.1, § 3.2, and § 3.3 several times to extract multiple fiber paths.

3.4 Coarse-to-fine optimization

To speed up the optimization, we perform multigrid optimization. As described in § 3.2, we initially downsample all the fiber paths. Then, for every fiber path p , we insert midpoints between every p_i and p_{i+1} by B-spline interpolation, using SciPy, and optimize all the fiber paths. This process can be repeated several times to generate the final fiber paths for 3D printing.

4 FABRICATION AND EXPERIMENTAL SETUP

In this section, we describe how we manufacture real 3D prints and measure their position-load curves. We use a Markforged Mark Two printer with nylon as the plastic material and carbon fiber as



Figure 3: A 3D printed part being tested on a universal testing machine (Instron 600DX), with square nuts in the trapezoidal holes. The machine moves at a speed of 20 mm/min and stops when the object breaks or by a manual stop.



Figure 4: Nine different fiber layouts printed for moduli calculation (left to right, top to bottom): no fiber path, 1 to 3 inner rings, 1 to 3 outer rings, 1 to 2 rings for all walls.

the reinforcing fiber material. We print laminates with a height of 2 mm and 16 layers, of which the 4th, 7th, 10th, and 13th layers are fiber layers. All layers without fiber and regions in fiber layers without fiber are filled with nylon (solid fill).

For the 2D shape, we use a 46 mm \times 30 mm rectangle with two rounded isosceles trapezoidal holes, the same shape as shown in Figure 2. The isosceles trapezoids have two sides of 11 mm and 14 mm and a height of 11 mm, with every corner smoothed by an arc with a radius of 1 mm. We will reuse this shape in § 5, § 6.3, and § 7.

To measure the position-load curve of a print, we insert two square nuts into both its holes and apply tension to them using a universal testing machine (Instron 600DX), as shown in Figure 3. The machine is programmed to move at a speed of 20 mm/min until the object breaks or by a manual stop when we believe enough data is collected. A position-load curve is recorded for every print.

5 MODULUS CALCULATION

In this section, we describe how we determine the effective Young's moduli of nylon and carbon fiber. We print composites with different (baseline) fiber layouts, measure their stiffness, then optimize for moduli such that their stiffness in simulation best matches the real-world measurements.

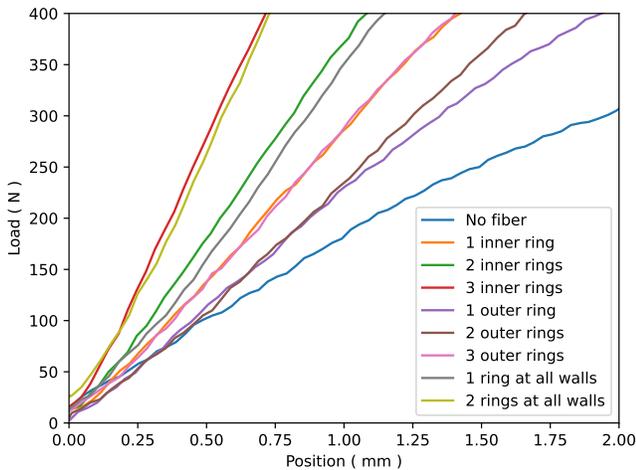


Figure 5: Position-load curves recorded from the testing machine. The beginning parts of the curves are noisy due to parts not being perfectly vertical, etc., and a too-large load can cause the object to buckle, violating our in-plane stress assumption. We therefore use the middle parts of the curves, with loads between 150 N and 300 N, to calculate the stiffness.

5.1 3D prints for testing

We use Eiger to generate nine different layouts of carbon fiber paths: no fiber path, 1 to 3 inner rings, 1 to 3 outer rings, 1 to 2 rings for all walls. To reduce the bias introduced by the non-uniformity of the material, we print all of them in one batch, as shown in Figure 4. Due to the variability of the printing process, we print three batches of these nine prints and pick the batch with the best printing quality.

5.2 Stiffness measurement

As described in Section 4, we test the prints and record their position-load curves (Figure 5). Note that the beginning of every curve can be noisy as the part is not perfectly vertical, etc. Additionally, a large load can cause the part to buckle out of the 2D plane, which violates our in-plane stress assumption. We therefore measure the position change between a load of 150 N and a load of 300 N for every print, and calculate the stiffness by dividing load change (150 N) by position change (in mm). The results are shown in Figure 6, marked as “X”. Note that there is a factor of 0.5 when converting stiffness in N/mm to energy in N·mm at 1 mm displacement (e.g., a stiffness of 500 N/mm corresponds to having strain energy of 250 N·mm at 1 mm displacement).

5.3 Simulation and modulus calculation

For each measured data point, we apply Dirichlet boundary conditions corresponding to 1 mm displacement on the two shorter sides of the two holes on the rectangle. We calculate the strain energy of the object, then do a grid search for the values of the moduli of nylon and carbon that minimize the sum of squared distances between measured and simulated stiffness. The search yields moduli of 0.40 GPa for nylon and 20.1 GPa for carbon, with results shown

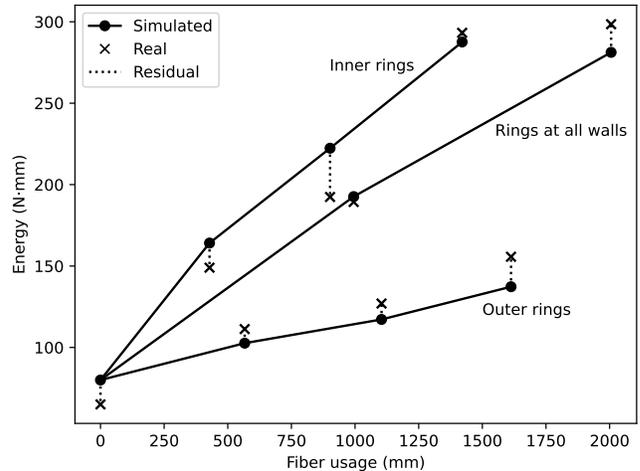


Figure 6: We calibrate the moduli of nylon and carbon fiber using nine prints with three different types of fiber layouts: inner rings, outer rings, and rings at all walls. The solid lines connect datapoints sharing the same fiber layout strategy. The real results are marked as “X”, the simulated results are marked as small solid circles, and the residuals are shown as dotted lines. The energy numbers are calculated at 1 mm displacement.

in Figure 6. As we can see, the simulation results mostly match the real results, with small residuals relative to the energy.

6 EXPERIMENTS

In this section, we present detailed evaluations of the performance of our method in both simulation and real experiments on four case studies (§ 6.1–§ 6.4), then show several additional results in § 6.5. We start with two simple shapes—a rectangle and a “plus” shape—then move to more complex shapes: rectangles with two and four holes (Figure 7). The first baseline we use is *concentric* fiber rings from Eiger, which have three different types: *inner*, *outer*, and *all walls*. For the next two case studies, to better illustrate the effectiveness of our algorithm on complex shapes and loads, we include two additional baselines: (1) *greedy*, simplifying our algorithm by removing all the optimization components and directly generating results using the greedy algorithm; (2) *field-opt-greedy*, similar to *greedy* but with an additional step of field optimization before running the greedy algorithm. The latter baseline, intended to represent the approach of previous work on fiber orientation optimization (see § 2.1), optimizes a vector field that aligns to the stress direction, with a smoothing regularizer. Additional details about the field optimization can be found in the appendix. We refer to the results from our method as *optimized*. For all the experiments (unless otherwise specified), we use the BFGS optimizer and limit the maximum number of iterations to 500 and a gradient tolerance of 3×10^{-9} . The objective function is set with $w_{\text{lap}} = 1 \times 10^{-8}$, $w_{\text{min}_1} = 1$, and $w_{\text{bdy}} = 1$.

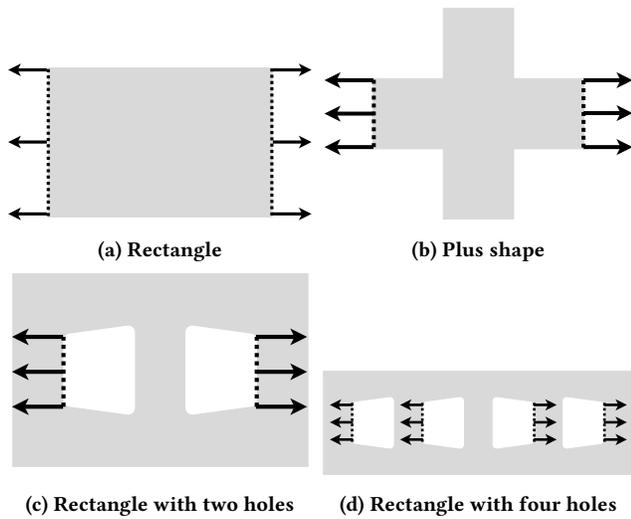


Figure 7: Four shapes we use in our case studies. (a) and (b) are relatively simple shapes, and the loads are applied on the two sides. For (c), a rectangle with two holes, tension is applied on the two shorter sides of the holes. (d) is designed to be a multi-functional rectangle with four holes, and the user can choose one hole from the left two holes and another hole from the right two holes to apply tension.

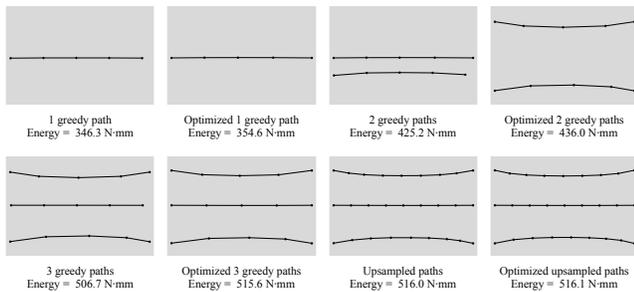


Figure 8: Step-by-step visualization of how our method extracts three fiber paths, optimizes them, and performs coarse-to-fine optimization on the *rectangle* shape. In the first row, we extract the first fiber path, optimize it, extract the second fiber path, and optimize both paths. The two paths curve and move up and down after the optimization, respectively. In the second row, we extract a third fiber path, optimize all three paths, upsample them, and finally optimize them. The energy numbers are calculated at 1 mm displacement.

6.1 Case 1: rectangle

In this case study, we show how our algorithm works step by step on a rectangle (45 mm × 30 mm), with tension applied to its two shorter sides (Figure 7a). As we can see in the first row in Figure 8, we first greedily extract a fiber path and optimize it. When we add and optimize a second fiber path, the two paths separate and curve. In the second row, we extract a third greedy path and optimize the three paths together. Finally, we double the number of points of

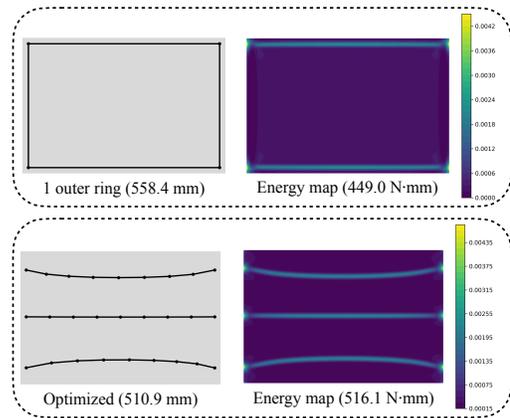


Figure 9: Fiber paths and energy maps of *outer* and *optimized* at 1 mm displacement on the *rectangle* shape. We use less fiber while achieving higher energy, as the baseline lays vertical fibers that are much less useful than horizontal fibers.

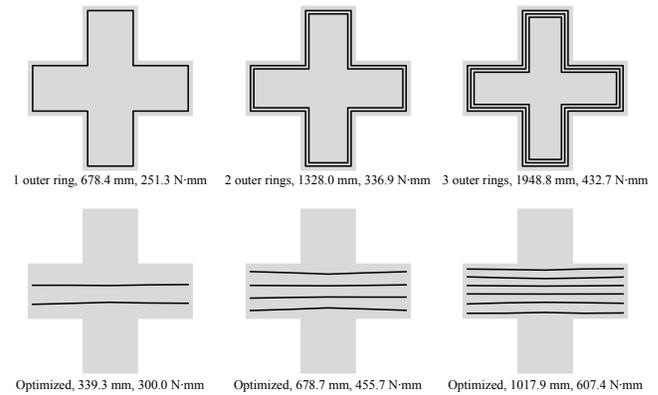


Figure 10: Fiber paths, lengths, and strain energy at 1 mm displacement of *outer* and *optimized* on the *plus* shape. With the help of optimization, fiber paths automatically distribute themselves uniformly in the space as we increase the number of fiber paths. By laying slightly bending fibers in horizontal directions, we save fiber while increasing the energy, compared to *outer*, which lays fiber in unrelated regions.

both fiber paths and optimize the three paths together. The last step does not help much since the task is relatively simple. As shown in Figure 9, for a fixed displacement of 1 mm, our algorithm uses less fiber while achieving higher energy in simulation, compared to 1 *outer* concentric fiber ring, which lays fiber in vertical directions that are much less useful than fibers in horizontal directions.

6.2 Case 2: “plus” shape

As shown in Figure 7b, we use a “plus” shape whose edges are all of length 15 mm, and we apply tension to two sides of the shape. We compare fiber paths of *outer* and *optimized* in Figure 10, with three solutions from each strategy. As we can see, *outer* lays fibers in regions of low relevance to the loads applied, in contrast

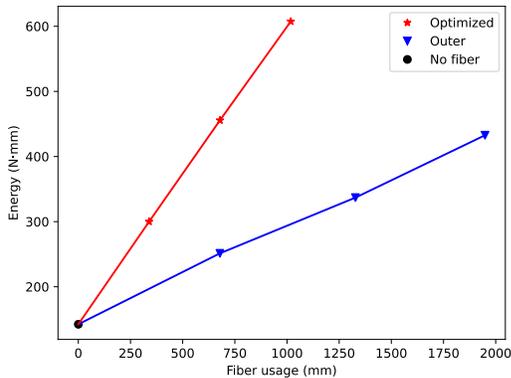


Figure 11: Energy-fiber usage plot of *outer* and *optimized* at 1 mm displacement on the *plus* shape. Our method improves over the Pareto front of *outer* by laying fibers according to the external loads.

Table 1: Real (measured) stiffness of *greedy* (*g*), *field-opt-greedy* (*f*), and *optimized* (*o*) on the *rectangle with two holes* shape, measured between 150 N and 300 N (4 batches). *Optimized* performs consistently better than the baselines when using a similar or less amount of fiber.

Stiffness (N/mm)	Solution 1			Solution 2		
	<i>g</i>	<i>f</i>	<i>o</i>	<i>g</i>	<i>f</i>	<i>o</i>
Batch 1	490.1	574.6	625.5	745.0	741.3	992.6
Batch 2	584.3	692.0	756.0	801.0	741.3	985.2
Batch 3	483.5	485.0	656.7	671.6	603.4	953.5
Batch 4	491.7	481.7	603.0	670.0	670.9	970.4
Average	512.4	558.3	660.3	721.9	689.2	975.4
Length (mm)	400.0	400.0	372.7	800.0	800.0	799.5

to *optimized* which prioritizes regions of high relevance to the loads. We also observe that the optimization process automatically distributes fiber paths uniformly as we extract more fiber paths. Based on our simulation, for a fixed displacement at 1 mm, the fiber paths of *optimized* improve upon the Pareto front of *outer*, as shown in Figure 11.

6.3 Case 3: rectangle with two holes

As shown in Figure 7c, we also tested a rectangle (46 mm × 30 mm) with two rounded isosceles trapezoid holes, with external forces applied to the two sides of the holes.

Planned fiber paths and simulation results. The fiber paths generated from all methods are shown in Figure 12. We set the maximum greedy fiber path length so that fiber lengths of *greedy*, *field-opt-greedy*, and *optimized* are comparable. As we can see, the baseline methods use only geometric information; both *greedy* and *field-opt-greedy* generate similar fiber paths along stress directions, but paths from *field-opt-greedy* are smoother; *optimized* wraps fiber

paths tightly around the holes while aligned with stress direction, yielding larger strain energy when using a similar amount of fiber.

Real experiment results. To evaluate the quality of fiber paths, we perform real-world experiments by applying tension to 3D prints on a universal testing system (600DX from Instron). Due to the limited space on the printer bed, two sets of comparisons are performed separately: (1) *inner*, *outer*, and *all walls* vs. *optimized*; (2) *greedy* and *field-opt-greedy* vs. *optimized*. We thus printed eight batches, four for each set of comparisons. Again, as in Section 4, we measure the stiffness of a print by calculating the slope of its position-load curve, picking two points that have loads of 150 N and 300 N. The results of *inner*, *outer*, and *all walls* vs. *optimized* are shown in Figure 13. As we can see, our algorithm consistently provides significantly higher stiffness than the concentric baselines when using a similar or lower amount of fiber. Note that the fiber lengths may have slight discrepancies between simulation and real-world experiments since they are from different path generation algorithms (one from our re-implementation of Eiger, another from Eiger directly). The results of *greedy* and *field-opt-greedy* vs. *optimized* are shown in Table 1. Again, our algorithm consistently improves the stiffness over the two baselines while using a similar or lower amount of fiber.

6.4 Case 4: rectangle with four holes

As shown in Figure 7d, we also tested a rectangle (84 mm × 28 mm) with four rounded isosceles trapezoid holes. We design the shape to be multi-functional—if we label the holes from 1 to 4 from left to right, we assume the user uniformly chooses one of the four settings: 1) hole 1 and hole 3; 2) hole 1 and hole 4; 3) hole 2 and hole 3; 4) hole 2 and hole 4. To support this multi-functional shape, we simulate all four cases and calculate the average strain energy.

The fiber paths from all the methods are shown in Figure 14. Again, both *greedy* and *field-opt-greedy* produce fibers along stress directions with fiber paths from *field-opt-greedy* being slightly smoother. *Optimized* lays the first fiber over all holes and lays the second fiber around the middle two holes, due to the multi-functional nature of the shape. The energy-fiber usage plot is shown in Figure 15, where *optimized* improves upon the Pareto front of every baseline.

6.5 Results on additional shapes

In this subsection, we provide results from our method and baselines on several additional shapes. The shape designs are inspired by sketches from SketchGraphs [Seff et al. 2020], a large-scale dataset of sketches of real-world CAD models, as well as shapes from existing works [Ma et al. 2022; Shafiqfard et al. 2019]. We use a Laplacian regularizer weight $w_{\text{lap}} = 5 \times 10^{-7}$, and the results are shown in Figure 16, with every dotted line a Dirichlet boundary condition. For the first shape, all methods use a similar amount of fiber but *optimized* achieves much higher energy than others. For the second shape, *optimized* uses a similar amount of fiber as *concentric*, less fiber than *greedy* and *field-opt-greedy* but achieves higher energy. For the third shape, *optimized* achieves comparable energy as *concentric* but saves approximately 70% of fiber. Compared to *greedy* and *field-opt-greedy*, *optimized* achieves much higher energy

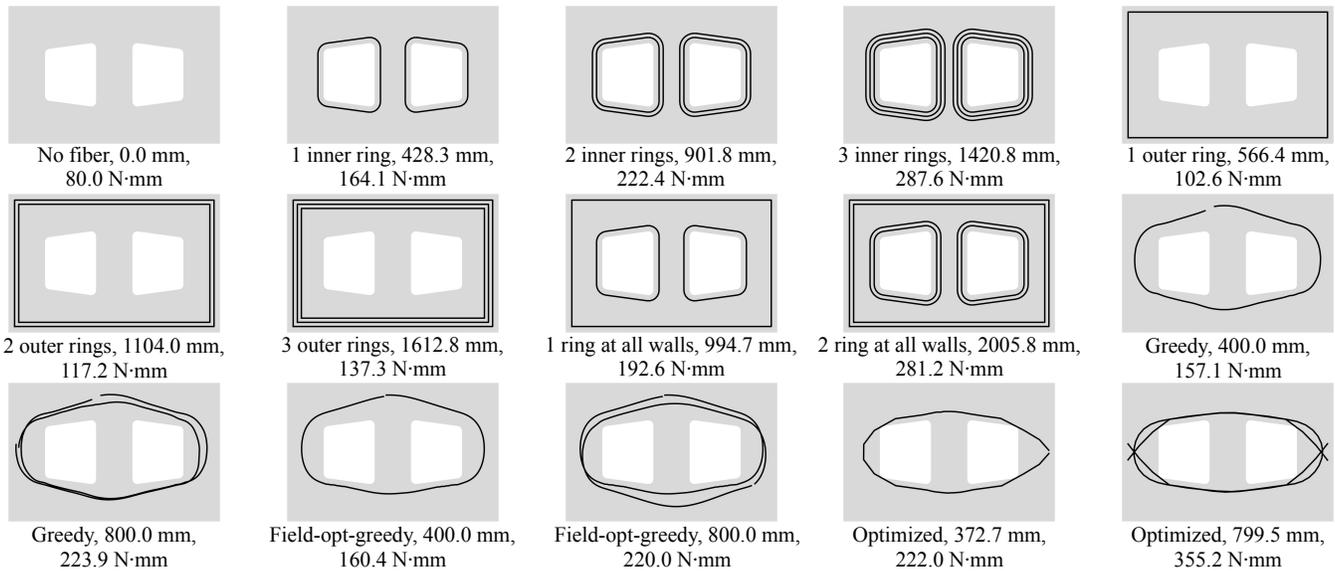


Figure 12: Fiber paths, lengths, and strain energy at 1 mm displacement of *inner*, *outer*, *all walls*, *greedy*, *field-opt-greedy*, and *optimized* on the *rectangle with two holes* shape. *field-opt-greedy* provides similar but smoother paths compared to *greedy*, and *optimized* provides more effective fiber paths. Note that there is a factor of 2 when converting the strain energy in N-mm at 1 mm displacement to stiffness in N/mm which we will use in real experiments (e.g., strain energy of 250 N-mm at 1 mm displacement corresponds to having a stiffness of 500 N/mm).

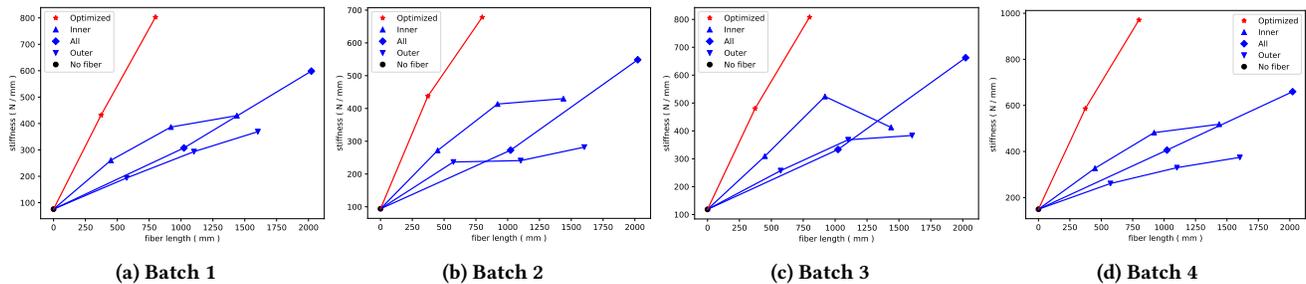


Figure 13: Real stiffness-fiber length plots of *inner*, *outer*, *all walls*, and *optimized* on the *rectangle with two holes* shape, measured between 150 N and 300 N (4 batches). By laying fibers tightly around the holes, *optimized* consistently performs better than all others.

while using slightly more fiber. For the fourth and fifth shapes, *optimized* uses less fiber or comparable fiber as other baselines while achieving significantly higher energy.

7 ABLATION STUDIES

Our algorithm without optimization has been studied in Section 6 as the *greedy* baseline. In this section, we study the effects of removing two other components of our method: the Laplacian regularizer and the multi-resolution optimization, using the shape *rectangle with two holes* (Figure 7c).

7.1 Ablation study of the Laplacian regularizer

As both the minimum-length regularizer and the boundary regularizer are intuitively necessary for fiber paths to be long enough

for printing purposes and within the object boundary, we study the effect of removing the Laplacian regularizer from the optimization. We run our algorithm with the same hyper-parameter setting except for $w_{lap} = 0$. We extract one fiber path and upsample for one time. As shown in Figure 17, the optimizer successfully optimizes the low-resolution path as the number of points is still small (Figure 17a), but introduces jagged results with more degrees of freedom (Figure 17b), demonstrating the need for some form of regularization.

7.2 Ablation study of multi-resolution optimization

In this subsection, we study how the multi-resolution approach speeds up the optimization process. For the multi-resolution case,

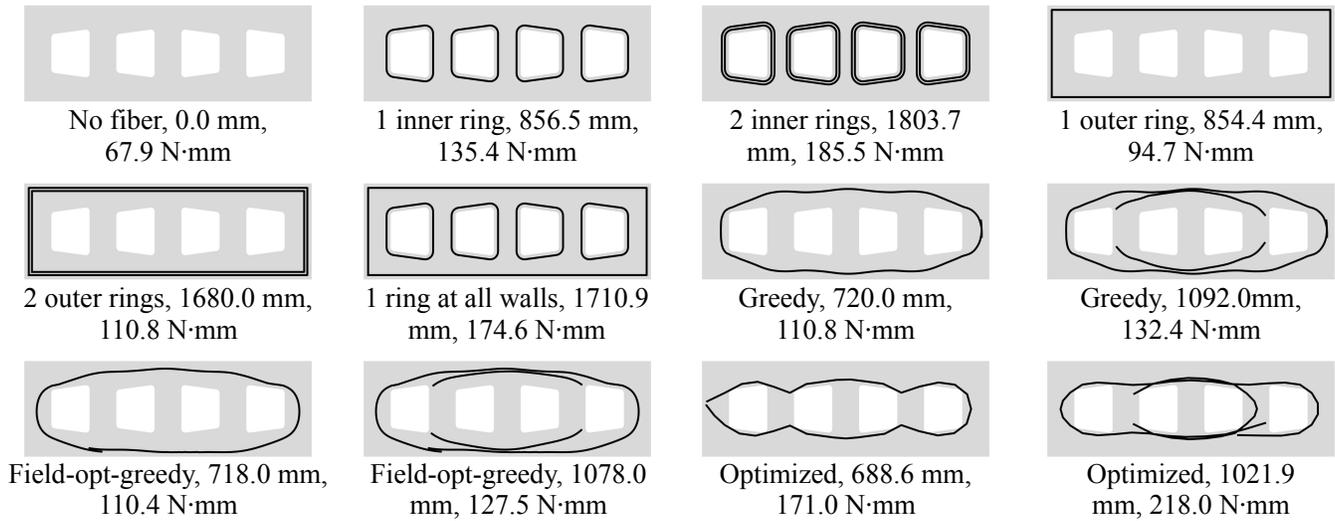


Figure 14: Fiber paths, lengths, and strain energy at 1 mm displacement of *inner, outer, all walls, greedy, field-opt-greedy, and optimized* on the multi-functional *rectangle with four holes* shape. Similarly, *greedy* and *field-opt-greedy* lay fibers along stress directions, and *field-opt-greedy* provides slightly smoother fiber paths. *Optimized* lays the first fiber over all the holes, and the second fiber around the middle two holes, with paths tightly around the holes.

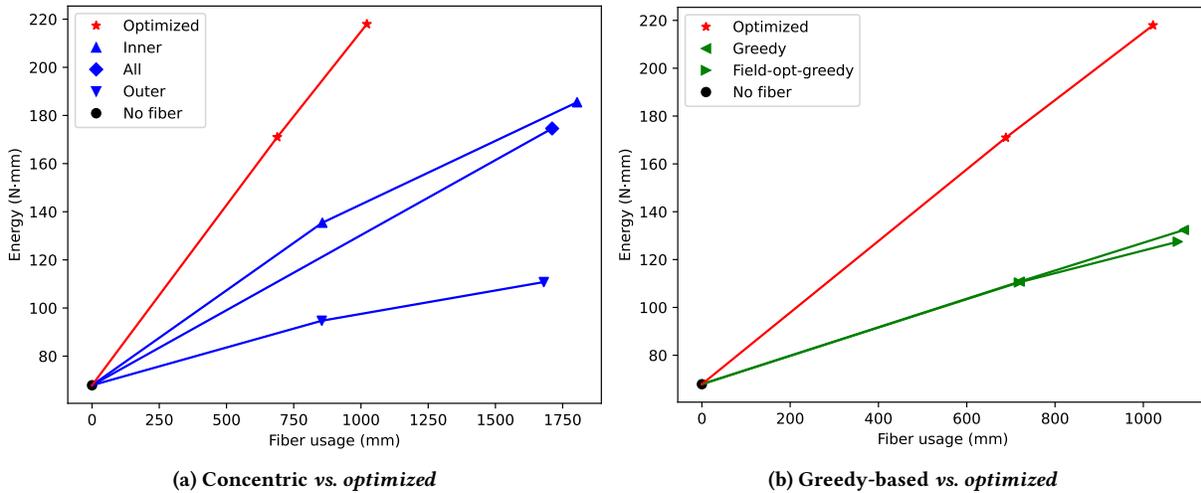


Figure 15: Energy-fiber usage plot (at 1 mm displacement) of all methods on the *rectangle with four holes* shape. The comparison between concentric fiber rings and *optimized* is shown on the left, and the comparison between greedy-based baselines and *optimized* is shown on the right. *Optimized* improves the Pareto front of all the baselines by laying fibers tightly around the holes.

we extract one fiber path, downsample its resolution by a factor of 20, optimize it, and upsample and optimize it three times, with every optimization limited to 100 iterations. For the single-resolution case, we also extract one fiber path, downsample its resolution by a factor of 2, optimize it and limit the maximum number of optimization iterations to 400. For a fair comparison, we use the same random seed for both cases when sampling starting points of the greedy path extraction algorithm. As shown in Figure 18, both cases get

similar fiber paths with similar strain energy, but multi-resolution optimization reduces the running time by approximately 40%.

8 DISCUSSION

In this work, we studied the task of fiber path planning in 3D printing for given external loads, aiming at maximizing the stiffness. We proposed an end-to-end optimization approach that optimizes regularized object stiffness directly to the fiber layout, rather than an intermediate fiber orientation field, with the help of the adjoint

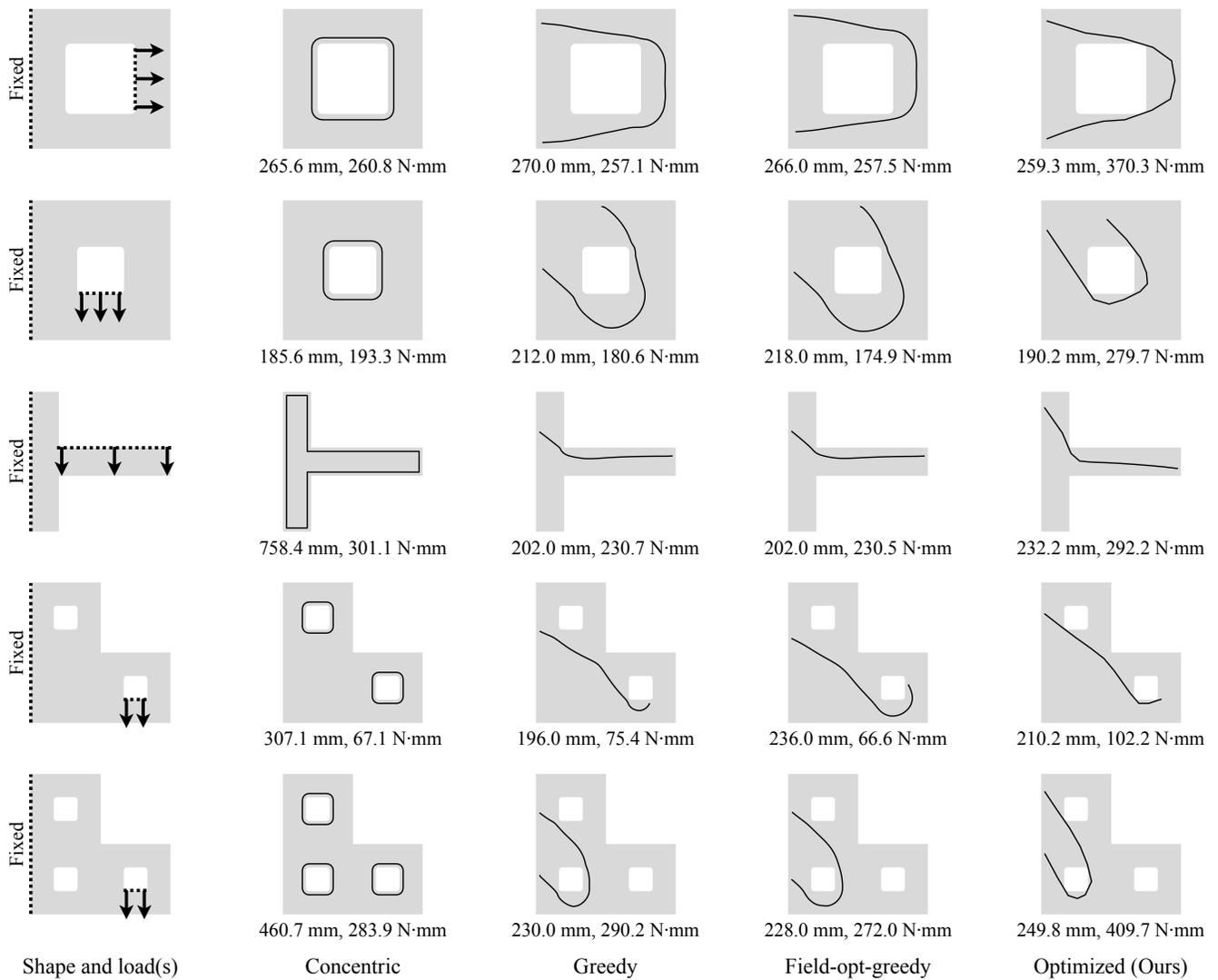
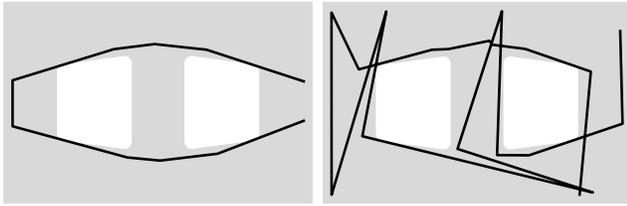


Figure 16: Fiber paths and energy at 1 mm displacement of all methods on additional shapes. Every dotted line indicates a Dirichlet boundary condition. For the first shape (from top to bottom), *optimized* achieves significantly higher energy while using slightly less fiber than all baselines. For the second shape, *optimized* achieves higher energy while using a similar amount of fiber as *concentric* and less fiber than *greedy* and *field-opt-greedy*. For the third shape, *optimized* saves approximately 70% of fiber usage while achieving similar energy as *concentric*. It also achieves much higher energy than *greedy* and *field-opt-greedy* while using slightly more fiber. For the last two shapes, compared to other baselines, *optimized* achieves significantly higher energy while using a less or comparable amount of fiber.

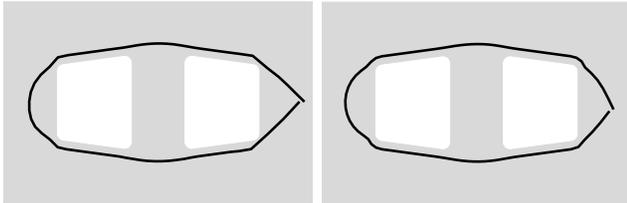
method and automatic differentiation. We perform planning by extracting fiber paths using a greedy algorithm that lays fiber paths along stress directions, followed by coarse-to-fine optimization. To apply our method, we first measure the effective moduli of plastic and fiber by manufacturing and testing real 3D prints. We then study our method with three baselines on four case studies and several additional shapes. The first baseline is concentric fiber rings from Eiger, a leading digital manufacturing software package developed by Markforged. The second baseline is our method with the optimization part removed, producing fiber paths from the

greedy path extraction algorithm. The third baseline includes a fiber field optimization part which smooths the stress field before using it in the greedy algorithm. We demonstrated that, both in simulation and real experiments, our method could generate shorter fiber paths while achieving greater stiffness (*i.e.*, we improved the Pareto front). We also studied the effects of removing the Laplacian regularizer and the multi-resolution optimization, showing the Laplacian regularizer is necessary for the optimization to be stable and multi-resolution optimization helps reduce the running time.



(a) Optimized low-resolution path (b) Optimized upsampled path

Figure 17: Optimization results with the Laplacian regularizer disabled. As shown on the left, the optimizer successfully optimizes the low-resolution path. It fails to optimize the fiber path after upsampling, as shown on the right.



(a) Multi-res, 293 s, 195.8 N·mm (b) Single-res, 485 s, 190.0 N·mm

Figure 18: Running time and the strain energy at 1 mm displacement for both single-resolution and multi-resolution optimization. In this case, we save approximately 40% of running time by multi-resolution optimization.

We would also like to mention some limitations of our method. First, our simulation simplifies the task by assuming linear elasticity, restricting to in-plane stress, and treating both plastic and fiber as isotropic materials with different Young's moduli and identical Poisson's ratio. Lifting these assumptions would introduce greater mathematical complexity, but would require no conceptual changes to our approach. Additionally, the planning is not performed in real time. For example, to plan fiber paths for the shape *rectangle with two holes*, our method uses 10 minutes and 18 minutes to generate the two studied solutions, respectively. Relative to the time required to design and print a part, this represents only a small increase. In addition, the hyper-parameters may have to be tuned when the task changes. For example, if we switch to a much larger shape, the scale of strain energy and the lengths of fiber paths will change. We may have to adjust the weight of the Laplacian regularizer, balancing the optimization stability and the variety of fiber paths, though this is usually easy to tune in a few tries. Lastly, as our optimizer is gradient-based, the optimization may be trapped in a local minimum. Thus a good initialization is important for our method, and we may have to sample greedy paths several times to obtain a good one.

ACKNOWLEDGMENTS

We would like to thank Amit Bermanno and members of the Princeton Laboratory for Intelligent Probabilistic Systems for valuable discussions and feedback, Joseph Vocaturro and the Princeton Department of Civil and Environmental Engineering for providing

the universal testing machine, as well as Markforged. This work is partially supported by the Princeton School of Engineering and Applied Science, as well as the U. S. National Science Foundation under grants #IIS-1815070, #IIS-2007278, and #OAC-2118201.

REFERENCES

- José Humberto S Almeida Jr., Lars Bittrich, Tsuyoshi Nomura, and Axel Spickenheuer. 2019. Cross-section optimization of topologically-optimized variable-axial anisotropic composite structures. *Composite Structures* 225 (2019), 111150.
- Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S Lazarov, and Ole Sigmund. 2011. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization* 43, 1 (2011), 1–16.
- Marco Attene, Marco Livesu, Sylvain Lefebvre, Thomas Funkhouser, Szymon Rusinkiewicz, Stefano Ellero, Jonàs Martínez, and Amit Haim Bermanno. 2018. Design, Representations, and Processing for Additive Manufacturing. *Synthesis Lectures on Visual Computing* 10, 2 (June 2018), 1–146.
- Paul T Boggs and Jon W Tolle. 1995. Sequential quadratic programming. *Acta numerica* 4 (1995), 1–51.
- Brian Cabral and Leith Casey Leedom. 1993. Imaging Vector Fields Using Line Integral Convolution. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. 263–270.
- R Caivano, A Tridello, D Paolino, and G Chiandussi. 2020. Topology and fibre orientation simultaneous optimisation: A design methodology for fibre-reinforced composite components. *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications* 234, 9 (2020), 1267–1279.
- Yang Cao, Shengtai Li, Linda Petzold, and Radu Serban. 2003. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. *SIAM journal on scientific computing* 24, 3 (2003), 1076–1089.
- Yuan Chen and Lin Ye. 2021. Topological design for 3D-printing of carbon fiber reinforced composite structural parts. *Composites Science and Technology* 204 (2021), 108644.
- Sheng Chu, Mi Xiao, Liang Gao, Yan Zhang, and Jinhao Zhang. 2021. Robust topology optimization for fiber-reinforced composite structures under loading uncertainty. *Computer Methods in Applied Mechanics and Engineering* 384 (2021), 113935.
- Andre Luis Ferreira da Silva, Ruben Andres Salas, Emilio Carlos Nelli Silva, and JN Reddy. 2020. Topology optimization of fibers orientation in hyperelastic composite material. *Composite Structures* 231 (2020), 111488.
- Juan Carlos De los Reyes. 2015. *Numerical PDE-constrained optimization*. Springer.
- Eralp Demir, Pouya Yousefi-Louyeh, and Mehmet Yildiz. 2019. Design of variable stiffness composite structures using lamination parameters with fiber steering constraint. *Composites Part B: Engineering* 165 (2019), 733–746.
- Ronald M Errico. 1997. What is an adjoint model? *Bulletin of the American Meteorological Society* 78, 11 (1997), 2577–2592.
- Boris Fedulov, Alexey Fedorenko, Aleksey Khaziev, and Fedor Antonov. 2021. Optimization of parts manufactured using continuous fiber three-dimensional printing technology. *Composites Part B: Engineering* 227 (2021), 109406.
- Andreas Griewank and Andrea Walther. 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM.
- Roland Herzog and Karl Kunisch. 2010. Algorithms for PDE-constrained optimization. *GAMM-Mitteilungen* 33, 2 (2010), 163–176.
- Zheng Hu. 2021. A review on the topology optimization of the fiber-reinforced composite structures. *Aerospace technic and technology* 3 (2021), 54–72.
- Tae-hoon Jung, Jaewook Lee, Tsuyoshi Nomura, and Ercan M Dede. 2022. Inverse design of three-dimensional fiber reinforced composites with spatially-varying fiber size and orientation using multiscale topology optimization. *Composite Structures* 279 (2022), 114768.
- SM Fijul Kabir, Kavita Mathur, and Abdel-Fattah M Seyam. 2020. A critical review on 3D printed continuous fiber-reinforced composites: History, mechanism, materials and properties. *Composite Structures* 232 (2020), 111476.
- Hang Li, Liang Gao, Hao Li, Xiaopeng Li, and Haifeng Tong. 2021. Full-scale topology optimization for fiber-reinforced structures with continuous fiber paths. *Computer Methods in Applied Mechanics and Engineering* 377 (2021), 113668.
- Nanya Li, Guido Link, Ting Wang, Vasileios Ramopoulos, Dominik Neumaier, Julia Hofele, Mario Walter, and John Jelonnek. 2020. Path-designed 3D printing for topological optimized continuous carbon fiber reinforced composite structures. *Composites Part B: Engineering* 182 (2020), 107612.
- Anders Logg and Garth N Wells. 2010. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)* 37, 2 (2010), 1–28.
- Guowei Ma, Wenwei Yang, and Li Wang. 2022. Strength-constrained simultaneous optimization of topology and fiber orientation of fiber-reinforced composite structures for additive manufacturing. *Advances in Structural Engineering* (2022), 1369433221088946.
- Markforged. 2023a. Carbon Fiber Composite 3D Printer: Markforged Mark Two. <https://markforged.com/3d-printers/mark-two>.

- Markforged. 2023b. Eiger 3D Printing Software. <https://markforged.com/software>.
- Sebastian K Mitusch, Simon W Funke, and Jørgen S Dokken. 2019. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software* 4, 38 (2019), 1292.
- Vasileios S Papapetrou, Chitrang Patel, and Ali Y Tamijani. 2020. Stiffness-based optimization framework for the topology and fiber paths of continuous fiber composites. *Composites Part B: Engineering* 183 (2020), 107681.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- Alexander A Safonov. 2019. 3D topology optimization of continuous fiber-reinforced structures via natural evolution method. *Composite Structures* 215 (2019), 289–297.
- Martin-Pierre Schmidt, Laura Couret, Christian Gout, and Claus BW Pedersen. 2020. Structural topology optimization with smoothly varying fiber orientations. *Structural and Multidisciplinary Optimization* 62, 6 (2020), 3105–3126.
- Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. 2020. SketchGraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design. In *ICML 2020 Workshop on Object-Oriented Learning*.
- Torkan Shafiqhfar, Eralp Demir, and Mehmet Yildiz. 2019. Design of fiber-reinforced variable-stiffness composites for different open-hole geometries with fiber continuity and curvature constraints. *Composite Structures* 226 (2019), 111280.
- Nurhalida Shahrudin, Te Chuan Lee, and Rhaizan Ramlan. 2019. An overview on 3D printing technology: Technological, materials, and applications. *Procedia Manufacturing* 35 (2019), 1286–1296.
- Kentaro Sugiyama, Ryosuke Matsuzaki, Andrei V Malakhov, Alexander N Polilov, Masahito Ueda, Akira Todoroki, and Yoshiyasu Hirano. 2020. 3D printing of optimized composites with variable fiber volume fraction and stiffness using continuous fiber. *Composites Science and Technology* 186 (2020), 107905.
- Xingyuan Sun, Tianju Xue, Szymon Rusinkiewicz, and Ryan P Adams. 2021. Amortized Synthesis of Constrained Configurations Using a Differentiable Surrogate. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.
- Ting Wang, Nanya Li, Guido Link, John Jelonnek, Jürgen Fleischer, Jörg Dittus, and Daniel Kupzik. 2021. Load-dependent path planning method for 3D printing of continuous fiber reinforced plastics. *Composites Part A: Applied Science and Manufacturing* 140 (2021), 106181.
- Yusuke Yamanaka, Akira Todoroki, Masahito Ueda, Yoshiyasu Hirano, Ryosuke Matsuzaki, et al. 2016. Fiber line optimization in single ply for 3D printed composites. *Open Journal of Composite Materials* 6, 04 (2016), 121.
- Leen Zhang, Xiaoping Wang, Jingyu Pei, and Yu Zhou. 2020. Review of automated fibre placement and its prospects for advanced composites. *Journal of Materials Science* 55, 17 (2020), 7121–7155.

A APPENDIX: FIELD OPTIMIZATION

Given a stress field σ , we would like to find a fiber field $\boldsymbol{v} : \Omega \rightarrow \mathbb{R}^2$ such that (1) its direction is aligned with σ ; (2) it is smooth. We solve \boldsymbol{v} by minimizing an objective function that reflects both properties:

$$\mathcal{L}(\boldsymbol{v}; \sigma) := \alpha_{\text{stress}} \cdot \mathcal{L}_{\text{stress}}(\hat{\boldsymbol{v}}; \sigma) + \alpha_{\text{smooth}} \cdot \mathcal{L}_{\text{smooth}}(\hat{\boldsymbol{v}}), \quad (13)$$

where α_{stress} and α_{smooth} are hyper-parameters, and

$$\hat{\boldsymbol{v}}(x, y) := \boldsymbol{v}(x, y) / \|\boldsymbol{v}(x, y)\| \quad (14)$$

is the normalized \boldsymbol{v} , as the objective function should be invariant regardless of the length of $\boldsymbol{v}(x, y)$. Note that the objective function should also be invariant if we randomly flip some $\boldsymbol{v}(x, y)$'s, which needs some special handling, as we will discuss below.

Consistent with σ . For a specific point $(x, y) \in \Omega$, we calculate the tension in the stress field σ along $\hat{\boldsymbol{v}}(x, y)$, which is

$$\hat{\boldsymbol{v}}(x, y)^\top \sigma(x, y) \hat{\boldsymbol{v}}(x, y). \quad (15)$$

We then integrate it over Ω and get

$$\mathcal{L}_{\text{stress}}(\hat{\boldsymbol{v}}; \sigma) := - \iint_{\Omega} \hat{\boldsymbol{v}}(x, y)^\top \sigma(x, y) \hat{\boldsymbol{v}}(x, y) dx dy, \quad (16)$$

where the negative sign indicates we would like to maximize the tension along the field direction.

Smoothness. We penalize the squared Frobenius norm of the gradient of $\hat{\boldsymbol{v}}$:

$$\mathcal{L}_{\text{smooth}}(\hat{\boldsymbol{v}}) := \iint_{\Omega} \|\nabla \hat{\boldsymbol{v}}(x, y)\|_F^2 dx dy. \quad (17)$$

Note that the penalty should be invariant to flips of $\hat{\boldsymbol{v}}(x, y)$'s, so we handle this invariance when calculating the finite difference:

$$\begin{aligned} \|\nabla \hat{\boldsymbol{v}}(x, y)\|_F^2 := & \min \left(\left\| \frac{\hat{\boldsymbol{v}}(x+h, y) - \hat{\boldsymbol{v}}(x, y)}{h} \right\|^2, \left\| \frac{\hat{\boldsymbol{v}}(x+h, y) + \hat{\boldsymbol{v}}(x, y)}{h} \right\|^2 \right) \\ & + \min \left(\left\| \frac{\hat{\boldsymbol{v}}(x, y+h) - \hat{\boldsymbol{v}}(x, y)}{h} \right\|^2, \left\| \frac{\hat{\boldsymbol{v}}(x, y+h) + \hat{\boldsymbol{v}}(x, y)}{h} \right\|^2 \right), \end{aligned} \quad (18)$$

where h is the step size.

In the experiments, we set α_{stress} to 1 and α_{smooth} to 0.02. We use the BFGS optimizer with a gradient tolerance of 1×10^{-6} and set the maximum number of iterations to 100.