DTU Library

Synthesis of Frame Field-Aligned Multi-Laminar Structures

Stutz, Florian Cyril; Olsen, Tim Felle; Groen, Jeroen Peter; Trung, Tuan Nguyen; Aage, Niels; Sigmund, Ole; Solomon, Justin; Bærentzen, Jakob Andreas

Synthesis of Frame Field-Aligned Multi-Laminar Structures

FLORIAN CYRIL STUTZ* and TIM FELLE OLSEN*†, Technical University of Denmark, Denmark
JEROEN PETER GROEN, Technical University of Denmark, Denmark
TUAN NGUYEN TRUNG, Technical University of Denmark, Denmark
NIELS AAGE, Technical University of Denmark, Denmark
OLE SIGMUND, Technical University of Denmark, Denmark
JUSTIN SOLOMON, Massachusetts Institute of Technology, United States of America
JAKOB ANDREAS BÆRENTZEN, Technical University of Denmark, Denmark
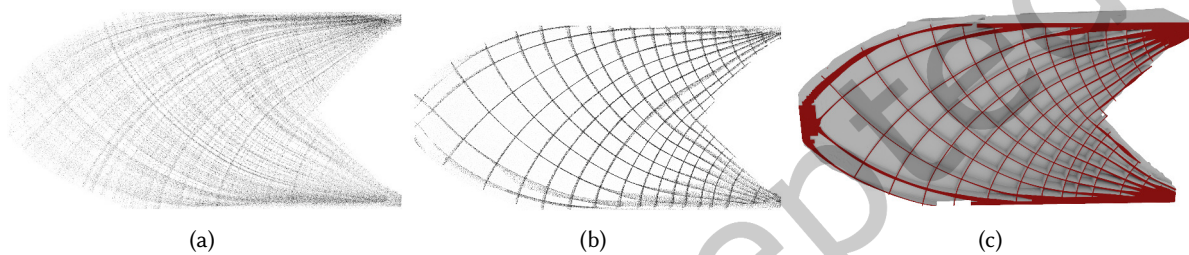
Fig. 1. Given a frame field we can generate a set of optimal laminations aligning exactly with the field orientations (Subfigure a). Using a novel optimization energy that only needs local orientation awareness, we can create a well-spaced subset of these laminations (Subfigure b). We then proceed to create near optimal, highly stiff multi-laminar structure as a volumetric solid (Subfigure c).

In the field of topology optimization, the homogenization approach has been revived as an important alternative to the established, density-based methods. Homogenization can represent microstructures at length scales decoupled from the resolution of the computational grid. The optimal microstructure for a single load case is an orthogonal rank-3 laminate.

*Both authors contributed equally to this research.
†Corresponding author: tife@dtu.dk.

Authors' addresses: Florian Cyril Stutz; Tim Felle Olsen, Technical University of Denmark, Department of Applied Mathematics and Computer Science, Richard Petersens Plads 324, Kgs. Lyngby, Denmark, 2800; Jeroen Peter Groen, Technical University of Denmark, Department of Mechanical Engineering, Niels Koppels Allé 404, Kgs. Lyngby, Denmark, 2800; Tuan Nguyen Trung, Technical University of Denmark, Department of Applied Mathematics and Computer Science, Richard Petersens Plads 324, Kgs. Lyngby, Denmark, 2800; Niels Aage, Technical University of Denmark, Department of Mechanical Engineering, Niels Koppels Allé 404, Kgs. Lyngby, Denmark, 2800; Ole Sigmund, Technical University of Denmark, Department of Mechanical Engineering, Niels Koppels Allé 404, Kgs. Lyngby, Denmark, 2800; Justin Solomon, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 77 Massachusetts Ave, Cambridge, MA, United States of America, 02139; Jakob Andreas Bærentzen, Technical University of Denmark, Department of Applied Mathematics and Computer Science, Richard Petersens Plads 324, Kgs. Lyngby, Denmark, 2800.

Initially, we investigate where singularities occur in orthogonal rank-3 laminates and show that the laminar parts of the structures we seek are unaffected by the singularities. Based on this observation, we propose a method for generating multi-laminar structures from frame fields that describe rank-3 laminates. Rather than establishing a parametrization of the domain, we compute stream surfaces that align with the frame fields and solve an optimization problem in order to find a well-spaced collection of such stream surfaces. Since our method does not rely on a parametrization, we also do not need a combing of the frame fields to generate this collection. Finally, we provide a method for synthesizing multi-laminar structures from a stream surface collection. This method produces a volumetric solid for each surface and combines these to form the output.

We demonstrate our method on several frame fields produced by the homogenization approach to topology optimization.

1 INTRODUCTION

In recent years, *topology optimization* [Bendsøe and Sigmund 2004] has emerged as an important tool in digital modelling and fabrication. By minimizing compliance, for example, topology optimization can produce mechanical structures that are stiffer than what a human designer would usually be able to achieve, using only a specified amount of material. More generally, topology optimization algorithms can directly optimize for structures, extremizing various quality measures of fabricated objects.

Density-based approaches for topology optimization employ a straightforward minimization over parameters that represent element-wise material density and, as such, operate directly on a volumetric shape representation. Unfortunately, large-scale topology optimization problems are very computationally demanding with this type of approach, albeit feasible in many contexts [Aage et al. 2017; Baandrup et al. 2020; Liu et al. 2018].

The *homogenization-based* approach to topology optimization offers an alternative wherein the material is represented in terms of homogenized microstructures [Bendsøe and Kikuchi 1988]. The optimal microstructure for single-load case stiffness optimization, which we also employ, is the rank-3 microstructure, a 3-scale lamination system with three orthogonal lamination directions. Using the orientation and lamination thicknesses obtained during topology optimization, we can realize a physical structure from the homogenization solution at a finite length scale in a process called *de-homogenization*. In practice, this two-step procedure yields finely resolved structures at a much lower computational cost than density-based methods [Bendsøe and Sigmund 2004; Geoffrey-Donders 2018; Geoffroy-Donders et al. 2020; Groen and Sigmund 2018; Groen et al. 2020; Pantz and Trabelsi 2008].

Clearly, one could also choose truss-based microstructures for homogenization-based topology optimization, resulting in a final structure consisting of trusses as done by Wu et al. [2019]. However, a truss carries load only in the direction of the truss itself, while a sheet can carry load along two directions. In practice, this means that closed wall structures are up to three times stiffer than truss-based Michell structures [Sigmund et al. 2016]. Consequently, our goal is to construct closed wall structures. Resent work have been done to combine the closed wall structures with truss based reinforcement [Gil-Ureta et al. 2020].

The specific problem we address is the following. Assuming the lamination orientations are given by a frame field, we seek a set of surfaces such that each surface aligns everywhere with one of the frame orientations. The surfaces should be approximately evenly spaced, and the spacing should correspond to a choice of length scale; up to three surfaces might intersect at any point.

If we can find a 3D parametrization of the domain such that the gradients of the coordinate functions are everywhere aligned with the frame field, the surfaces are simply constant coordinate surfaces pulled back from the parametrization domain. Unfortunately, the frame field might be far from integrable, and there are few if any robust approaches that can handle such cases. While recent work either does not account for singular curves

[Groen et al. 2020] or modifies the frame field at the cost of structural performance to promote integrability [Arora et al. 2019], we take a different route which does not require a parametrization of the domain.

In Section 4.2, we investigate the occurrence of singularities in frame fields obtained from homogenization-based topology optimization and relate the location of the singular curves to the underlying density and layer-thickness fields.

Based on this investigation, we propose a novel de-homogenization method that is oblivious to singularities in fields from mechanical homogenization-based topology optimization. Compared to the most similar method by Groen et al. [2020], our approach is faster, yet produces results that are much cleaner (i.e. without spurious small holes) and of similar mechanical performance.

As a core part of our method, we efficiently find a large superset of surfaces that align with input frame fields using a stream surface tracing approach which alleviates the need for consistent labeling of the input field. Additionally, we align strictly to the exact input field, and the method neither performs nor requires a smoothing of the input field. Given the computed superset, we select an evenly-spaced subset by posing a robust binary optimization problem which we solve efficiently through relaxation.

This scheme constitutes the main contribution of the paper. The tracing is discussed in Section 4.1 and the selection procedure in Section 4.3.

Given a set of stream surfaces, we further provide a method for the synthesis of output shapes. In topology optimization, we usually need a manufacturable solid as the output. In Section 5.2 we describe a method which produces a volumetric solid from a collection of surfaces based on their distance fields and the underlying layer-thickness field.

2 RELATED WORK

In recent years, density-based topology optimization has been used to find optimal mechanical structures in various fields. In the area of compliance minimization, giga-scale finite element models have been applied [Aage et al. 2017; Baandrup et al. 2020; Liu et al. 2018]. While such large-scale topology optimization makes the benefits of topology optimized structures very apparent, it also relies on supercomputers and/or is not applicable in real time, which is one of the key steps towards the goal of incorporating topology optimization in the everyday engineering design process.

Density-based topology optimization methods such as SIMP or RAMP [Sigmund and Maute 2013] were designed to directly produce single-scale mechanical structures. However, earlier work, specifically the ground-breaking work by Bendsøe and Kikuchi [1988], modelled material as having an infinitesimal microstructure — as opposed to being locally characterized only by density. Materials consisting of so-called rank-N microstructures have been shown to be theoretically and computationally optimal, while circumventing the problem that density-based topology optimized structures depend on the size of the chosen finite element mesh [Avellaneda 1987; Francfort et al. 1995; Sigmund and Maute 2013]. Work in additive manufacturing has sparked new interest in these microstructures [Panetta et al. 2015; Schumacher et al. 2015]

The process of going from the results of homogenization based topology optimization to high-resolution structures is called *de-homogenization*. It was introduced by Pantz and Trabelsi [2008], who combined homogenization-based topology optimization with field integration, as done in quad-meshing, to de-homogenize 2D examples whose orientation fields do not contain singularities. They later expanded their approach to structures with singularities of index $\pm 1/2$ lying in void regions by punching out holes around these singularities [Pantz and Trabelsi 2010]. Groen and Sigmund [2018] revisited this method and simplified the approach, while introducing additional parameters for more control of the de-homogenized structure. Their approach was limited to singularity-free fields and has since been ported to 3D [Groen et al. 2020]. Geoffrey-Donders [2018] proposed a method for de-homogenizing structures in 2D with singularities of index $\pm 1/2$ without the need of punching

holes based on the work by Hotz et al. [2010]. Stutz et al. [2020] expanded the approach by Groen and Sigmund [2018] to incorporate examples with singularities of index ±1/4.

All of the above papers indicate a strong relationship to quad-dominant meshing in 2D and hex-dominant meshing in 3D. Depending on the examples, a deterioration of the hexahedra is desirable as is the anisotropy resulting thereof. However, as shown in [Groen and Sigmund 2018; Stutz et al. 2020], spurious singularities can occur. In 3D, orientations of the microstructures are not unique, a problem for the de-homogenization that can to a certain degree be circumvented by regularization [Groen et al. 2020].

Approaches for truss-structures have been presented for singularity-free fields in [Arora et al. 2019; Larsen et al. 2018] and in [Wu et al. 2019] for fields containing singularities.

Field-based quad-meshing and hex-meshing is most often done by combing fields and integrating to find scalar functions with integer-jump conditions, where the combed field are differently labelled [Bommes et al. 2009; Kälberer et al. 2007; Nieser et al. 2011]. A lot of research for field-based hex-meshing focuses on achieving pure-hex meshes [Huang et al. 2011; Palmer et al. 2019; Ray et al. 2016; Solomon et al. 2017]. These methods focus on the field design part of the hex-meshing pipeline with the main goal to achieve as many hexahedral elements as possible. Thus, these methods minimize a smoothness energy while ensuring that at the surface one direction of the octahedral frame is well-aligned with the surface normal [Huang et al. 2011]. As a natural effect, hex-meshes extracted from such a model tend to have minimized anisotropy and minimized deterioration of the hexahedral elements.

For de-homogenization and hex-dominant meshing of homogenization-based topology optimization results, it is of importance to note that the fields are typically *prescribed* (rather than optimized during the meshing procedure) and cannot be changed to obtain more smoothness without reducing the mechanical performance of the obtained structure [Stutz et al. 2020]. Approaches like Kälberer et al. [2007] and Nieser et al. [2011] are promising for de-homogenization but contain a major pitfall since fields arising from the homogenization method often have singularities of higher indices (±1/2 in 2D) or have significant divergence at mechanical boundary conditions. Such higher indices imply a greater rotational speed and typically integration based methods for de-homogenization must enforce alignment to the fields with a penalization approach [Groen and Sigmund 2018; Groen et al. 2020; Stutz et al. 2020]. This penalization weight trades off structural alignment with spacing of the structural members and implicitly introduces anisotropy. If the alignment weight is chosen too small, the resulting parametrization will not align well with the underlying field as it tries to create unit-length gradients. If the alignment weight is chosen too large, the gradient of the parametrization will become zero and result in stretched out iso-contours [Stutz et al. 2020]. These problems might be mitigated by introduction of additional optimization terms, which has so far not been deeply investigated. It is important to note that anisotropy is desired and of the utmost importance for the mechanical performance.

In field-based hex-dominant meshing as done by Gao et al. [2017], the isotropy of the desired hexahedra is a key ingredient of the algorithm. This is due to the optimization, which trades off the regularity of the hexahedra and their alignment to the underlying field. An expansion to anisotropic hex-dominant meshing might be achieved, if the desired hex-edge length was known beforehand and not only given implicitly.

Ni et al. [2018] have a promising approach to solve for vertex position of a tetrahedral mesh, which is similar to Gao et al. [2017]. The nature of the approach is aimed at producing vertices of a hex-mesh with a prescribed isotropic edge-length. Note that Gao et al. [2017] and Ni et al. [2018] create tetrahedra where the hexahedra do not align with the field, which could cost dearly in terms of mechanical performance, when used for de-homogenization, since the resulting structure would not align with the load path at all in these regions. Recently, polycube methods have advanced the hex-meshing field, but since methods like Guo et al. [2020] and Livesu et al. [2020] do not rely on fields they are not applicable to de-homogenization.

We use the stream surfaces to synthesize a volumetric solid. Beyond the scope of the current paper, one might also employ the stream surfaces to create a hexahedral mesh as the dual of the structure obtained as the

intersection of the stream surfaces by drawing inspiration from the notion of the *spatial twist continuum* (STC) which was introduced by Murdoch et al. [1997]. Work in this vein includes the approach of Takayama [2019] which expands on the 2D work of Campen et al. [2012]; Campen and Kobbelt [2014] and relies on user-defined (as opposed to frame field aligned) implicit surfaces as an input to a hex meshing approach. The work by Calderan et al. [2020] does align to frame fields but requires the user to guide the process of hexahedral meshing.

Campen et al. [2016] create a foliation as a means of finding a bijective parametrization of a 3D shape. While there is a clear similarity between the notion of a stream surface and a transversal section of a leaf of a foliation of a 3-manifold [Milnor 1970], their aim is to create a bijective map entailing strong conditions on the direction field whereas we take the frame field as is.

Finally, it should be mentioned that stream surfaces are often used as visualization tools in fluid dynamics [Hultquist 1992; Machado et al. 2014].

3 INPUT FRAME FIELDS

This section discusses how frame fields are generated from homogenization-based topology optimization and presents novel research on the type of singular curves and locations occurring for these frame fields.

3.1 Homogenization-Based Topology Optimization

Both density-based and homogenization-based approaches to topology optimization operate on continuum representations of the material. However, in density-based approaches, the material is considered isotropic at the microscopic level. Homogenization-based topology optimization, on the other hand, uses orthotropic microstructures that vary in shape and orientation, and the optimization aligns these microstructures with the principal stresses [Pedersen 1989]. Thus, while homogenization-based topology optimization relies on more parameters to describe the material than density-based approaches, the length scale of the microstructure is considered infinitely small, and, hence, these parameters are in the form of continuous fields. Compared to density-based approaches, the output from homogenization-based topology optimization must be de-homogenized to obtain a manufacturable design [Pantz and Trabelsi 2008]. This decoupling of the topology optimization and shape generation allows us to choose a length scale for de-homogenization, making it possible to control the level of detail of the resulting shape. The lower bound on the compliance (assuming this is the optimized quantity) of the de-homogenized final shape is given by the continuum representation produced by the homogenization-based optimization. Consequently, the aim is to produce a finite length scale geometry whose compliance is as close as possible to this solution.

Two examples of microstructures are depicted in Figures 2a and 2b. The rectangular hole microstructure was considered for topology optimization by Bendsøe and Kikuchi [1988] and the rank-2 material with orthogonal layers was considered by Bendsøe [1989]. The rectangular hole microstructure can be rotated, and the size of the hole can be changed for both directions independently. The rectangular hole microstructure is a single-scale approximation of the multiscale rank-2 material in Figure 2b. The multiscale rank-2 materials have been shown to be optimal for two-dimensional problems with a single strain tensor by Avellaneda [1987]. The rank-2 microstructure is also orientated and the relative thickness of its layers can vary independently. The three-dimensional equivalent to the rank-2 microstructure is called a rank-3 microstructure with orthogonal layers. This microstructure is optimal for three-dimensional problems with a single load case.

In this paper, we rely on fields that describe the orientation and thickness of the layers of a rank-2 microstructure in 2D and orthogonal rank-3 microstructure in 3D. The orientations are described as 4-direction fields in two dimensions and as octahedral fields in three dimensions. The fields we employ were obtained by solving compliance minimization problems using the homogenization approach. Specifically, they were obtained using the method of Groen et al. [2020], and we refer to their paper for details.

A two dimensional illustration of the homogenization result for a double clamped cantilever beam is shown in Figure 2c. This result provides the theoretically lowest compliance, obtained for an infinitely fine lamination. Lamination angles and widths vary from element to element in the optimized structure. The goal of de-homogenization is to produce a structure with globally connected and continuously varying lamination with controlled and finite periodicity. If successful, and the periodicity is fine enough, de-homogenization provides a near-optimal design without any further finite element analysis.

3.2 Regularization and Singularities

A crucial part of the homogenization-based topology optimization is to find the optimal rotations of the microstructures, since microstructures have high stiffness in their principal directions but low in shear. Thus regularization of the orientations during the topology optimization will influence the resulting performance of the mechanical structure since more material needs to be allocated to strongly regularized regions [Stutz et al. 2020]. If regularization of the orientation fields is done after the topology optimization, either actively as discussed in Arora et al. [2019] or by not enforcing high enough penalization weights for an integrative method as discussed by Groen and Sigmund [2018] and Stutz et al. [2020], the resulting structure will not align well to the optimal microstructure orientation. Such non-optimally aligned regions may cause a dramatic loss of performance of the structure, leading to failure cases [Groen and Sigmund 2018; Stutz et al. 2020]. Therefore, the motivation of this paper is to robustly find structures that adhere to the local orientation of the microstructure as closely as possible outside of void or fully solid regions.

Another concern is singularities which may arise in the fields obtained from homogenization-based topology optimization. In two dimensions there are three reasons why singularities might occur [Stutz et al. 2020];

- Singularities in the underlying stress field will lead to singularities in the layer-normal fields since the microstructure aligns to the principal stress directions.
- Regularization inflicted on the layer-normal fields during the topology optimization will break up singularities with a higher index in the stress fields into multiple singularities of lower index in the layer-normal fields.
- In regions where the microstructure is completely solid or void, singularities can be introduced by noise. In solid regions the microstructure becomes isotropic and the optimal orientation becomes non-unique. In void regions the microstructure is not present and an optimal orientation of the microstructure is therefore non-existing.

Stutz et al. [2020] show that, in two dimensions, singularities in topology optimized layer-normal fields must occur in completely solid or void regions.

Our de-homogenization method is the first three-dimensional approach that takes singularities explicitly into account, thereby significantly reducing the risk of failing structures. In the following we present an investigation of the occurrence of singularities in fields from the homogenization approach. These observations help us develop our singularity-aware de-homogenization method.

At the outset, the two-dimensional observations of Stutz et al. [2020] do not hold in three dimensions. Firstly, microstructure orientations in three dimensions are not unique due to in-plane stress; this can cause spurious singularities to appear. These singularities can be handled with a low amount of regularization, as shown by Groen et al. [2020]. Secondly, as shown in Figure 3, singularities can occur in stress fields, even when the microstructures are not completely solid. If we consider Figure 3a we see a field describing a singularity with index 1. Stutz et al. [2020] observed that because the rotational velocity of the field increases towards infinity at the singularity, the topology optimization process fills the region around the singularity with material to account for the spinning stress field at the singularity. On the other hand, when we embed the fields from Figure 3a in three dimensions, as shown in Figure 3b, the optimizer can choose to fill the newly introduced orthogonal layer with

material and not assign any material to the two layers carried over from 2D. Furthermore, we observe that this third layer does not have to be completely solid but can have any arbitrary layer-thickness, e.g. 50%. In this case we refer to the microstructure as being *transversely isotropic* since the microstructure is isotropic in one plane (the yellow one) but anisotropic perpendicular to this plane. The option to cut out singularities and later on fill them with material, will inevitably lead to excessive use of material in three dimensions. The example described in Figure 3 is, to the best of the authors' knowledge, the only singularity in three dimensions that occurs outside of fully solid or entirely void regions, and we will refer to such singularities as *transverse singularities*.

The following thoughts can explain this. First, if all layer normals change direction at a location outside the void, for example, around a source, then the region would need to be filled with material by the optimizer to be made isotropic. Second, non-zero stress directions will always be perpendicular to a layer normal, with non-zero layer thickness, meaning that stresses must always be transferred within a solid slab or plate. This will always align a stress field's singular curve with a layer normal outside of fully solid or entirely void regions. This leaves us only with fields as shown in Figure 3b, where of course, the indices of the singularities can be different. Third, consider for a moment that the red or blue layer would be non-zero. Then their layer-normals would rotate infinitely fast at the singular curve, and thus the optimizer would fill the region completely with material to make the microstructure isotropic at the singular curve. Hence we conclude that the only singular curve not embedded into complete solid or void can be seen in Figure 3b, where the red and blue layer thicknesses are zero.

With this knowledge in hand, we are in a position to design an approach that can account for singularities in three dimensions without explicit computation. Our stream surface generation method is designed to steer the expansion of stream surfaces based on the underlying layer-thickness. For a transverse singular curve, the only active layer has a near constant normal, see for example Figure 3b. This nearly constant normal makes it very easy to create the stream surfaces orthogonal to the transverse singularity. Of course these findings are also relevant for de-homogenization methods based on integration [Arora et al. 2019; Groen et al. 2020]. However, compared to integration-based approaches that rely on a single combing, our approach yields large benefits in terms of robustness since a wrongly chosen layer-normal in the highly rotational areas close to a singular curve would only affect a single stream surface and not the complete structure. Further, as explained in detail in Section 4.2, we are able to design a method that does not need any knowledge or computation of the singular curves.

4 COMPUTING COLLECTIONS OF STREAM SURFACES

The overarching idea of our method is to compute a large set of surfaces, $\mathcal{S}$, which align with the frame field and then find a well-spaced selection of these, $\mathcal{S}_{opt}$, to get a representation of the multi-laminar structure that we seek. In this section, we discuss how we find and select these aligned surfaces using stream surface tracing. In Section 5, we will discuss how the final output is computed from this representation.

In engineering, a *streamline* is simply a curve that is everywhere tangential to a vector field [Hultquist 1992]. A stream surface is the generalization to 3D, i.e. a surface whose normal is everywhere aligned with one of the vectors of the input frame field.

Unfortunately, we cannot rely on the frame field being combed, and hence, we do not have consistent labeling of the vectors in the frame. This is handled by simply finding the frame vector best aligned with the estimated normal of the next point that we compute when expanding a stream surface. It is also worth noting that we generally wish to stop stream surface tracing when the stream surface would otherwise exit a given bounding shape. Thus, we assume a known mask or layer thickness in the following.

4.1 Tracing Stream Surfaces

We start by tracing stream surfaces to create the set $\mathcal{S}$. The stream surfaces are traced independently, starting from random seed points in the domain. Rather than constructing a surface connectivity, we compute a point cloud for each stream surface. The points are placed using a method similar to the technique for *Poisson Disk Sampling* (PDS) introduced by Bridson [2007], except that our points are placed on a surface in 3D and are not filling the entire 3D domain.

We initialize each surface with a single seed point $\mathbf{p}_0$ and with two of the three frame vectors at $\mathbf{p}_0$. The first vector is our desired surface normal $\mathbf{N}$ at the seed point, and the second vector is perpendicular to $\mathbf{N}$ and describes our rotational origin $\mathbf{D}$. New points are now generated in an annulus centered on the seed point and perpendicular to the surface normal. Uniformly distributed random variables control the rotation angle from $\mathbf{D}$ and distance from $\mathbf{p}_0$. The annulus has an inner radius of r, which is the minimum distance allowed between points. The outer radius is set to $2r$ in accordance with Bridson's algorithm [Bridson 2007]. Each time we generate a new point, we check if it is too close to any previously generated points of the stream surface, using a lookup grid for efficiency. This generation process is visualized in Figure 4. When a new point is accepted, it is added to a queue of points used to further expand the surface.

To mitigate drift, we employ the fourth order Runge-Kutta method (RK4) [Chapra 2012]. Starting from a previously determined point, $\mathbf{p}_0$, with normal $\mathbf{N}$, we search in direction $\mathbf{d}_0$ with step length Δ. We need a parallel transport operator $\mathcal{P}_{\mathbf{x}}^{\mathbf{y}} : \mathbb{R}^3 \to \mathbb{R}^3$ to transport the initial direction $\mathbf{d}_0$ at $\mathbf{x}$ onto the tangent plane estimated at a given point $\mathbf{y}$ with a normal defined by the field. The RK4 method combines partial steps through a weighted sum, to estimate the new point. The full update can be described by,

$$\mathbf{k}_1 = \Delta \cdot \mathcal{P}_{\mathbf{p}_0}^{\mathbf{p}_0}(\mathbf{d}_0) \ , \qquad\qquad \mathbf{k}_2 = \Delta \cdot \mathcal{P}_{\mathbf{p}_0}^{\mathbf{p}_0 + \frac{\mathbf{k}_1}{2}}(\mathbf{d}_0) \ ,$$

$$\mathbf{k}_3 = \Delta \cdot \mathcal{P}_{\mathbf{p}_0}^{\mathbf{p}_0 + \frac{\mathbf{k}_2}{2}}(\mathbf{d}_0) \ , \qquad\qquad \mathbf{k}_4 = \Delta \cdot \mathcal{P}_{\mathbf{p}_0}^{\mathbf{p}_0 + \mathbf{k}_3}(\mathbf{d}_0) \ ,$$

$$\mathbf{p}_n = \mathbf{p}_0 + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \ .$$

While this method is relatively precise, some drift is still unavoidable. To improve precision, we compute the position of $\mathbf{p}_n$ from all points $\mathbf{p}_i$ inside the sphere with radius $2r$ centered at $\mathbf{p}_n$. To update the position estimate we use the vector $\mathbf{p}_n - \mathbf{p}_i$ projected into the tangent plane and normalized as $\mathbf{d}_0$ and define $\Delta = ||\mathbf{p}_n - \mathbf{p}_i||$. These new estimates are averaged to produce the new point $\mathbf{p}$.

To further improve alignment of the surface to the normals, we recompute the positions of all points from their neighbors using the same scheme as for the initial placement.

We discard a point if it is too close to a neighbor, i.e. at distance $< r$. This could still allow for "spiraling" surfaces caused by drift, i.e. surfaces which, having returned almost to their starting point, are at some distance from said starting point in the normal direction. Therefore, we also look at all neighbors within $4r$. If any of these neighbors, when projected onto the tangent plane defined by $\mathbf{p}$ and the associated normal, are closer to $\mathbf{p}$ than r we also discard the point. We also do not expand stream surfaces into regions where the corresponding layer has a layer-thickness of zero. No material would be assigned in these regions by the volumetric synthesis described in Section 5.2. Finally, we do not expand into fully solid regions, since these areas will be filled with material anyway by the volumetric synthesis.

It is clear that stream surfaces are subject to some drift. We experienced that this can be a problem when a stream surface closes up after it has been traced around a round object. In our initial approach, the position of each new point was computed based on only one previous point, and we sometimes observed gaps in the surface when it closed up as illustrated in Figure 5. However, with our robust tracing method that computes each new

point based on several points and also realigns in post processing, this is not an issue. Pseudocode for the robust tracing is given in Algorithm 1.

Algorithm 1 Stream surface creation

Inputs: Frame field, lamination thicknesses, list of seed points, list of probe points.

1: **for** each seed point $\mathbf{p}_s$, **do**
2:      Initialize queue Q with $\mathbf{p}_s$ and PDS grid with desired radius.
3:      **while** Q is not empty, **do**
4:          Set $\mathbf{p}_0$ to front of Q.
5:          Compute 30 new points $\mathbf{p}_i$ using weighted RK4.
6:          **if** $\mathbf{p}_i$ is a valid point, **then**
7:              Save $\mathbf{p}_i$ to point-cloud S_s, PDS grid and Q.
8:          **end if**
9:      **end while**
10:      **for** every point $\mathbf{p}$ in S_s, **do**
11:          Re-estimate $\mathbf{p}$ using weighted RK4.
12:      **end for**
13:      Save S_s to the set of surfaces $\mathcal{S}$.
14: **end for**

4.2 Singularities

As discussed in Section 3.2, singular curves relate to the currently traced stream surface in one of two ways. Either the singular curve directly affects the surface normal of the currently traced stream surface or it affects the normals of surfaces perpendicular to the currently traced surface (transverse singular curve).

We need only be concerned with the former type when tracing stream surfaces since the latter kind of singularity does not affect the surface normal on which our stream surface depends, i.e. we can trace a stream surface even if it is impinged by a transverse singularity.

Moreover, as stated in Section 3.2 singular curves either lie in fully solid or void regions or must be transverse singular curves. Motivated by this we can prevent our results from being affected by the first two type of singular curves by stopping the tracing of stream surfaces if the layer-thickness becomes zero or if the region's density is solid.

Note that to avoid mechanical issues, 5% is often the smallest allowed layer thickness, and above 95% density is considered solid. Recall that for singularities, not in complete solid or void (transverse singular curves), only a single layer has a non-zero thickness as depicted in Figure 3b. Here we generate the layer traversing the singular curve perpendicular due to its non-zero layer-thickness. This allows our approach to handle singular curves implicitly by only generating the relevant stream surfaces near a singular curve.

With this in place, we now have a method to create the set of stream surfaces $\mathcal{S}$ and proceed to selecting an optimal subset.

4.3 Energy for an Optimization-Based Subselection Approach

We will now take the set of surfaces $\mathcal{S}$ that we have created in the previous sections and continue by finding a well-spaced subset $\mathcal{S}_{opt}$. We will compute $\mathcal{S}_{opt}$ by optimizing over binary variables $\mathbf{w}$ that will be assigned to the stream surfaces. However, before we can define our optimization problem, we need to define the contribution of each stream surfaces to the optimization energy. For simplicity and consistency with the figures, we will

describe this procedure in two dimensions. The algorithm works the same in three dimensions, and we will explain essential details for the implementation inline on an ongoing basis.

First let γ denote the desired average spacing in the set $\mathcal{S}_{opt}$. As an aid, we define the projection of a point $\mathbf{x} \in \mathbb{R}^2$ onto a streamline $s \subset \mathbb{R}^2$ as $\mathbf{x}_p = \arg\min_{\mathbf{x}_s \in s} ||\mathbf{x} - \mathbf{x}_s||$. We can then define a binary indicator for a given streamline s by

$$\bar{I}_s : \mathbb{R}^2 \rightarrow \{0, 1\} \ ,$$
$$\bar{I}_s(\mathbf{x}) = \begin{cases} 1, & \text{if } ||\mathbf{x} - \mathbf{x_p}|| \leq \frac{\gamma}{2}, \\ 0, & \text{else.} \end{cases} \tag{1}$$

This simple indicator is shown in Figure 6b for the two streamlines following orthogonal field directions in Figure 6a. For our application to 4-direction fields, we need to distinguish between the two orthogonal field directions locally. Therefore, we choose for every $\mathbf{x} \in \Omega$, two orthogonal directions from the 4-direction field at random and assign them to 2-direction fields f_1 and f_2. This assignment of the orthogonal directions to f_1 and f_2 allows us to define a function $F_s : \mathbf{x}_s \mapsto \{1, 2\}$ that indicates for every point $\mathbf{x}_s \in s$ if the streamline follows the local label of field f_1 or field f_2. In three dimensions, we use the normal of the stream surface as a field identifier. We now expand the indicator to create a separate response for each frame direction:

$$I_s : \mathbb{R}^2 \rightarrow \{0, 1\} \times \{0, 1\} \ ,$$
$$I_s(\mathbf{x}) = [I_{s,1}(\mathbf{x}), I_{s,2}(\mathbf{x})]^\top \ ,$$
$$I_{s,d}(\mathbf{x}) = \begin{cases} 1, & \text{if } F_s(\mathbf{x}_p) = d \wedge ||\mathbf{x} - \mathbf{x}_p|| \leq \frac{\gamma}{2} \ , \\ 0, & \text{else,} \end{cases} \tag{2}$$

Note that there is no need for consistency of the field labels f_1 or f_2 in a neighborhood, i.e. no combing is needed, as shown in Figure 7. This makes the indicators very simple to implement and the approach very robust. The response of the indicator defined in Equation 2 can be seen in Figure 11. Having defined the indicator we can now formulate a binary optimization problem,

$$\underset{\mathbf{w} \in \{0,1\}^{n_\mathcal{S}}}{\text{minimize}} \ E(\mathbf{w}) = \int_\Omega \left\| \sum_{s \in \mathcal{S}} w_s I_s(\mathbf{x}) - [1, 1]^\top \right\|_{L_1} d\mathbf{x} \ , \tag{3}$$

where we refer to the optimization variables w_s as weights and $n_\mathcal{S} = |\mathcal{S}|$. If we were to use the indicators from Equation 1 the selected streamlines would all follow the same lamination direction, since the optimizer would penalize crossing streamlines. This can be seen in Figure 8. If we use the same set of streamlines but use the indicators defined in Equation 2 for the optimization we obtain both laminations as can be seen in Figure 9.

One could define the optimization energy in Equation 3 as a least-squares problem ($||\cdot||_{L^2}$) instead of the least absolute deviation ($||\cdot||_{L^1}$). However, the least-squares version promotes rejection of streamlines rather than allowing some overlap in converging regions. This rejection of streamlines is clearly highlighted when comparing the solutions using L^1- and L^2-norms, as shown in Figure 10. Details on the solution of the minimization problem in Equation 3 are discussed in Section 4.6. We now continue to estimate the variable $n_\mathcal{S}$.

4.4 Number of Streamlines $n_\mathcal{S}$ in the Covering Set of Streamlines $\mathcal{S}$

To solve the minimization problem in Equation 3 we need to know how large the number of streamlines $n_\mathcal{S} = |\mathcal{S}|$ provided to the optimizer needs to be.

First, we need to define the desired average spacing γ of the streamlines in S_{opt}. Then the cardinality of S_{opt} can be approximated by,

$$\left|S_{opt}\right| = \frac{n_x}{\gamma} + \frac{n_y}{\gamma} \quad , \tag{4}$$

where n_x and n_y are the dimensions of the design space in x, respectively y direction. Note that the cardinality of S_{opt} grows in linear dependence to the dimensions of the design space, since streamlines are one dimensional objects. This means that doubling all dimensions of the design space will only lead to a doubling of the cardinality of S_{opt}. This also holds true in three dimensions, here due to the two-dimensionality of stream surfaces. We further need to define the error ε by which a streamline should deviate on average from its optimal position. We denote this as a fraction of the optimal average spacing γ, e.g. $\varepsilon = 0.1$ would allow a streamline to be placed in a band of 0.2γ width around its optimal location. We can then derive the cardinality of S by:

$$n_S = |S| = \frac{1}{\varepsilon}\left|S_{opt}\right| = \frac{1}{\varepsilon}\left(\frac{n_x}{\gamma} + \frac{n_y}{\gamma}\right) \quad . \tag{5}$$

As with S_{opt}, we note that the cardinality of S grows linear with the dimensions of the design space. We also note that the cardinality of S grows linear in dependence to the desired error ε, meaning that reducing ε by a factor k will increase the cardinality of S only by a factor k. Both these observations are again valid in two dimensions as well as in three dimensions.

We have now computed how many stream surfaces we need to provide to the minimization problem in Equation 3 to obtain good results.

4.5 Resolution of the Energy

To solve the minimization problem in Equation 3 the only thing that remains is to discretize the energy E on a pixel grid, where we refer to a single pixel as a probe point. To efficiently subselect streamlines, we need to know the resolutions of the discretized energies, i.e. the number of probe points needed to differentiate streamlines in the set S. This number depends on the desired error ε and the desired average spacing γ. Each streamline should activate the probe points lying in a band of width γ around the streamline. Two streamlines that are more than $\varepsilon \cdot \gamma$ apart should activate a different set of probe points. This implies that the number of probe points needed can be computed by,

$$n_p = \frac{n_x}{\varepsilon \cdot \gamma} \cdot \frac{n_y}{\varepsilon \cdot \gamma} = \frac{1}{\varepsilon^2}\left(\frac{n_x}{\gamma} \cdot \frac{n_y}{\gamma}\right) \quad . \tag{6}$$

Here we note that the number of probe points grows quadratically in two dimensions, meaning doubling both dimensions of the design space will increase the number of probe points needed by a factor of four. Respectively, the number of probe points grows cubically in three dimensions. Note, however, that the subselection is only a fraction of the time spent on the whole approach as can be extracted from Table 1.

We have now discretized the energy E of the minimization problem in Equation 3 and are now ready to solve it.

4.6 Subselection using a Relaxed Approach to Binary Programming

Solving the minimization problem in Equation 3 can be done by using integer linear programming. However, the underlying problem is likely NP-hard due to the binary constraints. This makes a direct solve of the problem formulated in Equation 3 infeasible. To solve the least absolute deviations problem, we relax the optimization variables w_s to be continuous in the interval $[0, 1]$ instead of the discrete set $\{0, 1\}$. This leads to the following

convex linear program, which can be solved in polynomial time:

$$\underset{\mathbf{w} \in [0,1]^{n_S}}{\text{minimize}} \; E(\mathbf{w}) = \int_\Omega \left\| \sum_{s \in S} w_s I_s(\mathbf{x}) - [1, 1]^\top \right\|_{L^1} d\mathbf{x} \; . \tag{7}$$

We solve the relaxed problem in Equation 7 with an interior point method and then fix weights that have been set to either 0 or 1. Subsequently, we solve a binary program with the remaining weights (typically < 5% of the original weights) using a branch and cut algorithm. A branch and cut algorithm splits the original problem into sub-problems and uses cutting planes to cut away parts of the possible solution space until an optimal integer solution is found for a sub-problem. If that solution is better than a relaxed solution of a second sub-problem, the second sub-problem does not need to be solved. This is done iteratively until the algorithm converges. For details, we refer to Padberg and Rinaldi [1991]. We use the implementation provided in CVX [Grant and Boyd 2014].

Note that the high number of binary weights chosen in the relaxed problem is due to the indicator having binary values. If we were to base the energy on a signed distance function instead, we would almost exclusively receive non-binary weights as a result from the relaxed problem in Equation 7 since the optimizer would try to trade off contributions of different streamlines.

The observation in Section 4.4 that the computational burden of the problem in Equation 3 grows linearly in the amount of stream surfaces has an important practical use. Spiralling stream surfaces, which can occur due to heavy noise in the topology optimized fields and are described in Section 4.2, will cover more space than non-spiralling surfaces. They are therefore chosen less by the optimizer when the number of surfaces in S increases.

We have now found a well-spaced set of laminar surfaces S_{opt} and can continue to the generation of output structures.

5 OUTPUT GENERATION

The stream surface tracing and selection procedure described above produces a set of stream surfaces, S_{opt}, each represented as a point cloud. In itself, this representation is useful for visualization. However, our end goal is to provide methods for synthesizing output structures. Here we present a method that transforms each stream surface into a functional representation (implicit) with varying thickness. These functional representations are then sampled on a voxel grid, and we compute the union to obtain the final volumetric solid.

5.1 Post-Processing the pointclouds

When constructing the initial set of surfaces, we do not need a particularly high density of points in each surface. We only need enough to be able to compute the activation of probe points. However, a high density of points will provide a smoother and more precise volumetric solid.

It is crucial that our post-processing still adheres to the field and follows the initial surface. Therefore, our up-sampling is a continuation of the generation procedure. We initialize a new grid for our Poisson Disk Sampling procedure with a much smaller allowed point distance. We then add all of the original points of a selected stream surface to the grid. Subsequently, we add all original points to a queue and restart the point generation. This way, we fill in additional points between the original points since we have chosen a smaller distance for the Poisson Disk Sampling. Any new position is generated as the average of estimates from neighboring points, hence the super-sampled surface will still be following the field closely.

5.1.1 Triangulation of the pointclouds. In order to improve the precision of our computation of volumetric solids (see Section 5.2) we compute a triangulation of each pointcloud. Our triangulation borrows ideas from Gopi et al. [2000] and uses an existing implementation of 2D Delaunay triangulation by Shewchuk [1996].

Our pointclouds have a couple of properties which simplify the triangulation significantly. All pointclouds have relatively uniform spacing between points, they are generated such that new points are never closer than a given distance to any other points. We know our desired surface normal at each point, since we follow the frame field.

Like Gopi et al. [2000] we compute a local 2D Delaunay triangulation around each point and compare the meshes from nearby points. The local Delaunay triangulation is done for the projection of all points within 3× the super sampling point distance onto the estimated tangent space of the center point.

If an edge is mutually Delaunay for all nearby points, then we keep it. As described by Gopi et al. [2000], we might see holes in the mesh from edges not being Delaunay for some of the nearby points. In this case we can run a simple hole closing method for small holes. In our case holes with less than 10 edges are closed.

5.2 Volumetric Solids

We now have a list of triangulated surfaces, evenly spaced and aligned to the field. However, such a list of surfaces is not useful for manufacturing or analysis of mechanical performance. Therefore, we synthesize a volumetric solid by giving each surface a thickness matching the material distribution given by the homogenization.

To construct the volumetric model, we build an implicit representation of each thickened surface and unify these. Then we add blocks to represent the solid regions where surfaces are not traced. Finally, the surface of the mechanical structure is computed as an iso-contour from the implicit representation.

We compute the functional solid for each stream surface, s, from its triangle mesh representation using

$$V_s(\mathbf{x}) = \mathrm{ss}\left(\frac{d_s(\mathbf{x})}{\tau(\mathbf{x})}\right), \tag{8}$$

where $d_s(\mathbf{x}) = \min_{\mathbf{x}_s \in s} ||\mathbf{x}_s - \mathbf{x}||$ is the distance to the stream surface, $\tau(x)$ is the thickness interpolated to $\mathbf{x}_s = \arg\min_{\mathbf{x}_s \in s} ||\mathbf{x}_s - \mathbf{x}||$ from the thicknesses at the corners of the triangle containing $\mathbf{x}_s$ to the point itself, and

$$\mathrm{ss}(t) = \begin{cases} 1 - 3t^2 + 2t^3, & t \in [0, 1] \ , \\ 0, & \text{elsewhere} \ . \end{cases} \tag{9}$$

Given a functionally represented solid for each stream surface, the volumetric solid corresponding to a stream surface collection is simply the union of the solids for each of the stream surfaces. The union is computed as the maximum over all stream surfaces, and these values are stored in a 3D voxel grid,

$$V[\mathbf{x}] = \max_s(V_s(\mathbf{x})) \ . \tag{10}$$

It is important to note that, in addition to the functional solids, we add blocks of material where the underlying homogenization result has a material density above the solid threshold.

Finally, we compute a triangle mesh of the boundary of the volumetric solid using iso-surface polygonization with the iso-value 0.5 [Ju et al. 2002].

6 IMPLEMENTATION AND RESULTS

Our implementations are in C++ and Matlab. C++ was used for the stream surface tracing and synthesis of volumetric solids. Matlab was used for the selection of surfaces. This code employs the CVX package [Grant and Boyd 2008, 2014] and the Mosek solvers [MOSEK ApS 2021].

The generation of stream surfaces and the synthesis of volumetric solids have been parallelized using MPI and the native threading facilities of C++, respectively. The stream surface tracing and the subselection were executed on a node equipped with two Intel Xeon E5-2650 v4 processors [DTU Computing Center 2021]. The volumetric

Table 1. Here we show the statistics for the different steps in our pipeline. The first three blocks of rows show relevant statistics for the initial point sampling of stream surfaces, the selection of optimal stream surfaces, and the super-sampling of the selected stream surfaces. The fourth block reports statistics for generating the volumetric solids. We then report in the fifth block the overall runtime of our approach. In addition to our statistics, we have included the Homogenization time and provide De-homogenization both for our and Groen et al. [2020] for direct comparison. Note that neither we nor Groen et al. [2020] provide a final mesh or FE-based post processing as part of de-homogenization timing.

	Cantilever (1 layer)	Cantilever (3 layers)	Electrical Mast	L-Shaped Beam	Torsion Sphere	Torsion Sphere Heavy realignment
$|\mathcal{S}|$ (Number of stream surfaces)	480	480	240	240	480	400
Generating $\mathcal{S}$ (walltime)	00:06:04	00:04:49	00:02:31	00:24:51	00:42:06	05:39:52
Average points per surface	1 762	1 277	1 004	1 980	7 197	8 593
Subselection (walltime)	00:00:57	00:00:50	00:01:20	00:00:30	00:03:33	00:01:40
$|\mathcal{S}_{opt}|$	25	42	40	56	11	4
Super-sampling (walltime)	00:03:28	00:05:27	00:05:39	00:22:48	00:31:58	01:30:23
Average points per super-sampled surface	3 609	3 027	2 995	2 015	30 608	22 650
Output volume dimension	$960 \times 480 \times 480$	$960 \times 480 \times 480$	$480 \times 480 \times 1440$	$960 \times 960 \times 480$	$720 \times 720 \times 720$	$720 \times 720 \times 720$
Volumetric generation (walltime)	00:39:58	00:54:02	00:47:35	00:38:28	00:42:44	00:59:44
Total walltime	00:50:27	01:06:08	00:57:24	01:26:37	02:00:21	08:11:39
Homogenization	07:48:00	09:54:00	22:20:00	43:30:00	40:21:00	40:21:00
De-homogenization ours	00:10:29	00:11:06	00:09:30	00:48:09	01:17:37	07:11:55
De-homogenization by Groen et al. [2020]	00:45:39	–:–:–	00:54:49	01:08:06	01:19:44	01:19:44

generation was executed on a single Intel Core i7. An overview of the statistics, including computation time, is shown in Table 1.

For the evaluation of our results, we used the TetWild software kindly made available by Hu et al. [2018] as well as the commercial application COMSOL.

6.1 Parameters

In our implementation, we use several parameters. Some of these can be chosen by the user, and others are derived. In the following we discuss both types and values used in our experiments, see Appendix A for a full list of values.

As described in Section 4.1, the creation of surfaces is controlled by a sampling distance. For all of our examples, we base the value of this on the input voxel side length, v. The sampling radius was chosen to be v for the surface creation, and we super sample with a radius of $\frac{v}{2}$.

Since the length scale is relative to the scale of the design, we provide the length scale as the number of surfaces along the shortest dimension of the input volume containing the homogenization solution in the following discussions, i.e.

$$N_s = \frac{\min_i(n_i)}{\gamma} \quad . \tag{11}$$

We generally use $N_s = 10$ which corresponds to a value of $\gamma = 4.8$ except in the case of the Torsion Sphere for which $\gamma = 7.2$. In a Spacing Experiment, we show the effect of varying N_s.

The total number of surfaces that should be traced is given by Equation 5. However, this is a conservative number that assumes the entire domain is used which is not true for the Electrical Mast and the L-Shaped Beam. For the Electrical Mast, only the top third of the domain is used in combination with a thin region, and the

L-Shaped Beam also uses a partial domain. In both cases, the computed $n_S = 500$, and in both cases we only use $n_S = 240$ to account for the reduced utilization. A full description of the design domains and boundary conditions can be found in [Groen et al. 2020]. Finally, we employ $n_S = 480$ for all the spacing experiments, and this number is too small for $N_s = 15$ and $N_s = 20$. However, the subselection is still satisfactory which indicates that the computed values are probably conservative in some cases.

Because a certain resolution is needed to resolve a volumetric surface, the required resolution depends on the thickness of the stream surfaces. Figure 18 highlights the issues that arise if we use too low resolution.

The surface thickness cannot be controlled directly, since we must maintain the volume fraction: if we want more surfaces, they must be thinner. In practice, the input fields provide the fraction of material occupied by a given lamination orientation at each point in the volume, and we use that to compute the thickness. To compute the required resolution, we use μ, the minimal material thickness, and the length scale given as γ, the desired spacing discussed in Section 4.5. Specifically, the number of voxels, n_i^r, required along a given axis, i, of the volume that should hold the final synthesized solid is given by

$$n_i^r(\tau^r) = \frac{\tau^r \cdot n_i}{\mu \cdot \gamma} \quad , \tag{12}$$

where τ^r is the minimal surface thickness measured in voxel units relative to the volumetric structure and n_i describes the number of input voxels in dimension i.

As the minimal surface thickness for the volumetric structure, we use $\tau^r = 1.5$ voxel units. It can be shown that for thicknesses less than 1 voxel unit, spurious holes can arise when the surface is parallel to the voxel grid. Using a thickness of 1.5 gives a reasonable margin.

In all of our experiments, we use a voxel grid for the output with a resolution that is ten times that of the input grid. For most of the experiments this resolution exceeds that given by Equation 12. However, this makes it easier to compare structures based on the same input fields.

6.2 Missing Structural Members and Field Alignment

As discussed in Section 2 our method aims to circumvent the problem of missing structural members due to enforcement of alignment to the input field when using an integrative method to create a parametrization. As discussed in [Groen and Sigmund 2018; Groen et al. 2020; Stutz et al. 2020], alignment of the final structure to the input field needs to be enforced by a constraint when adapting integrative approaches as Bommes et al. [2009]; Kälberer et al. [2007]; Nieser et al. [2011]. This is done by enforcing the parametrization to be orthogonal to the second (and third) normal direction. However, if this alignment is too strict, the gradient of the parametrization may become almost zero in large regions. This, in turn, can then lead to overly thick structural members or to missing structural members especially around singularities as discussed by Stutz et al. [2020]. Our approach creates well-aligned structures before selecting a subset, eradicating the problem since we cannot suffer from vanishing gradients, as we do not integrate the fields. We show an example in Figure 12. Note that the same behavior can be observed in three dimensions.

The structure shown in Figure 12a has been obtained by de-homogenizing a 320×80 layer-normal field by an integrative approach proposed by Stutz et al. [2020]. Note how there are missing structural members above and underneath the singularity. The structure has a compliance $C = 26.46$ and a volume fraction of $V = 0.275$. For comparison of structures with slightly varying volume we calculate the compliance-volume product $C \cdot V = 7.30$. Stutz et al. [2020] report compliance-volume products of 7.05, 7.48, 7.63, and 21.39 for different alignment weights at the same resolution. Here 7.05 is their best performing structure at an intermediate alignment weight, and 21.39 is a failure case.

Figure 12b shows the structure created by our approach also using a 320×80 layer-normal field. Note that our approach yields a structure with evenly spaced structural members. The structure has a compliance $C = 27.67$

and a volume fraction of $V = 0.269$. The compliance-volume product for this structure is $C \cdot V = 7.44$. Note that this value is only 5.5% worse than Stutz et al. [2020] best value. Moreover, with our approach we do not risk a failure case due to bad alignment or zero gradients in a parametrization.

6.2.1 Field alignment in 3D. Our method aims at providing a strong alignment to the underlying field. Figure 13 shows the deviation from the field. The deviation is computed as the difference between the angle-weighted surface normal at each vertex in the triangulation and the nearest frame direction to that normal, interpolated at the vertex position.

For all our test problems it is clear that the vast majority of our surface points exhibit a very high degree of alignment. The mean deviation is within a few degrees and we have a very narrow standard deviation. We do, however, have a few outliers as can be seen by the tails of the histograms. Upon closer inspection of the surfaces it is clear that the outliers are primarily located on the surface edges and regions with layer thicknesses very close to the minimal cutoff.

In Section 4.1 we describe a method for realigning the generated points. The realignment was used on the torsion sphere example, first 10 iterations and later we ran 100 iterations. As shown in Table 1 the recomputation is an expensive process to run. Even just running the 10 iterations takes significantly longer than any of the other data sets.

While the torsion sphere is a challenging experiment, we have few points with high deviation as seen in Figure 14a. Running 10 iterations of the realignment does yield some reduction in the number of outliers, but after 10 iterations we get diminishing returns from further realignment.

6.3 Volumetric Structures from Topology Optimized Fields

We ran various input fields from topology optimization through our pipeline. The fields were generated by the method proposed by Groen et al. [2020]. For problem formulations of the homogenization-based topology optimization and a description of the load cases, we refer to Geoffroy-Donders et al. [2020]; Groen et al. [2020]. The timings of the field generation and the de-homogenization are reported in Table 1, where we see that the topology optimization dominates over our de-homogenization approach. In Figure 15a we see a quarter of an electrical mast as proposed in [Geoffroy-Donders et al. 2020]. The fields generated for the electrical mast example contain spurious singularities in fully solid regions and the void due to the microstructure being isotropic (solid) or non-existent (void). Nevertheless, we produce very smooth surfaces, since our stream surfaces do not need to expand into solid or void regions. Groen et al. [2020] make use of the fact that singularities only arise in fully solid or void regions by combing the fields in intermediate regions first, such that the spurious singularities cannot create seams in the combed field that extend into the intermediate regions. However, their approach yields no control or guarantee over how much singularities influence designs since they still rely on the orientations in solid and void for integration, although they use relaxation for such elements.

Figures 15b, c & d show the three dimensional version of Michell's cantilever. For loading cases and problem formulation we refer to Geoffroy-Donders et al. [2020]; Groen et al. [2020]. We compute de-homogenization results for two versions. In Figure 15b, we depict a solution for the cantilever where we enforce that either all three layers have a layer thickness of more than 5% or that all layer thicknesses are zero. Such a design, albeit heavier, is less sensitive to imperfections in the mapping due to the transverse stiffening effect of the additional layers and is indirectly better for resistance against buckling [Clausen et al. 2016]. Note that due to all three layers being enforced to have non-zero layer widths outside of the void, the microstructure orientation becomes unique in this example. Spurious singularities only arise in solid and void regions. A cut section through the structure is shown in the bottom image in Figure 15b.

In Figure 15c, we show the second version of the cantilever that we consider. These input fields have been created without any enforcement on the layer-thicknesses and correspond to the cantilever in Groen et al. [2020].

We compare our results with theirs, first on a visual level in Figures 15c & d and then in terms of compliance and volume in Table 2.

In the leftmost images of Figures 15c & d we see the full de-homogenized structures. The two structures are very similar. Note that Groen et al. [2020] use some additional expensive fine-scale evaluation to remove unused excess material, i.e. low strain-energy elements. This puts their structure at a slight advantage over ours, since we do not incorporate such a post-processing step for our structure in Figure 15c. The middle images of Figures 15c & d show a detail and the right images show horizontal cuts through the structures.

On the bottom right, in Figures 15e & f we show our version of the torsion sphere example proposed in Groen et al. [2020] that was based on Michell's famous torsion sphere [Michell 1904]. Note that since we use optimal rank-3 microstructures, we do not get a truss structure, but a stiffer layer structure [Sigmund et al. 2016]. The torsion sphere has a singularity that connects the two boundary conditions, similar to a towel being wrung. This singular curve passes through the solid region at one boundary condition, then through the void and the solid region at the opposite boundary condition. Note that this is again not a problem for our algorithm since we neither need to expand into void nor solid regions. Groen and Sigmund [2018] include the singular curve in their integration of the field without any special measures, since they are relaxing the parametrization in the void and fully solid regions. Our method produces three high-quality shells that align well with the input field.

The L-Shaped Beam depicted in Figure 16 is also an example tackled by Groen et al. [2020]. This design represents both bending and torsion moments. This example is made exceedingly difficult by the thin walls around the hollow end of the structure. In order to better describe this shape, we added an additional surface. This new surface represent the outer isocontour of material. The indicator functions are computed for this isocontour and then we force the selection of this surface by locking it in the optimizer.

Enforcing an outer surface layer is consistent with principal stresses being aligned with or perpendicular to unloaded design boundaries, which at least for the three active layer case means that one layer should cover the surface. And does not damage the mechanical performance of the experiment, as can be seen in Table 3.

6.4 Finite Element Analysis

Groen et al. [2020] evaluated their Cantilever design on a regular grid using $960 \times 480 \times 480 = 221\,184\,000$ finite elements with trilinear shape functions, whereas we use the TetWild software provided by Hu et al. [2018] to obtain a tetrahedral mesh for our evaluation. These tetrahedral meshes were then analysed in the commercial finite element software COMSOL Multiphysics [AB 2021] using quadratic shape functions.

All our results are evaluated this way, except the torsion sphere, which is evaluated using code supplied to us by Groen et al. [2020], for a direct comparison.

We compare the compliance volume products to [Groen et al. 2020]. Note, however, that the compliance and volume of the optimal structure is also known from the original topology optimization. Thus, we can also compare the ratio of the compliance volume products of the de-homogenized results to the original homogenization outputs, i.e.

$$\mathcal{F}_{CV} = \frac{C_s \cdot V_s}{C_h \cdot V_h}.$$

We see in Table 2 that our cantilever performs 2.3% percent worse than the best performing result of Groen et al. [2020]. However, it is clear from Table 3 that, on average, we perform on par with Groen et al. [2020] across all models. In only a single case do we have a result which is slightly outside the range reported by Groen et al. [2020].

Removing non-load carrying regions, as done by Groen et al. [2020], would decrease the volume without affecting the compliance, thereby improving the overall mechanical performance. However, most of the models would likely not benefit significantly. The electrical mast seems to have several dangling parts, but this model is

Table 2. Comparison of our results for the cantilever example with the best performing results obtained by Groen et al. [2020]. We use the following abbreviations: V_s = volume of the de-homogenized structure, C_s = compliance of de-homogenized structure, V_h = volume of the homogenization-based topology optimization solution, C_h = compliance of the homogenization-based topology optimization solution, $\mathcal{F}_{CV} = (C_s \cdot V_s)/(C_h \cdot V_h)$ = compliance-volume fraction. Note that Groen et al. [2020] evaluated their design on a grid of $960 \times 480 \times 480$ finite elements with trilinear shape functions, where as we use the method proposed by Hu et al. [2018] to obtain tetrahedral meshes for evaluation in COMSOL Multiphysics [AB 2021].

Cantilever	Groen et al.	Our approach	Our approach	Our approach
DOF	$667 \cdot 10^6$	$1.9 \cdot 10^6$	$2.8 \cdot 10^6$	$3.6 \cdot 10^6$
C_h	226.68	228.45	228.45	228.45
V_h	0.1000	0.1000	0.1000	0.1000
C_s	243.31	215.65	215.85	215.98
V_s	0.1021	0.1188	0.1187	0.1187
$C_s \cdot V_s$	24.845	25.62	25.62	25.64
$\mathcal{F}_{CV}$	1.0960	1.1214	1.1215	1.1222

only one quarter of a symmetrical design, and the seemingly dangling parts would meet their counterparts on the symmetry conditions. Certainly, the biggest change would be to the single layer cantilever in Figure 4.1(c) where the extreme vertical plate exhibits two "ears" that do not quite meet up. These would likely by removed by such a procedure.

The final example, i.e. the torsion sphere, is analyzed using the same code and voxel representation as done by Groen et al. [2020]. We use a regular grid of 570^3 finite elements with trilinear shape functions. The computation is done across 3200 cores on a HPC system consisting of 555 nodes each with two AMD EPYC 7351 CPUs (16 cores) and 128GB RAM. However, without the element removal step. This is done partly because the application of the torsion load and necessary passive solid regions are difficult to reconstruct in COMSOL Multiphysics, and partly because this allows us to the compare our results directly to those from Groen et al. [2020]. As can be seen from Table 3, analyzing the torsion sphere using the same tool as Groen et al. [2020] leads to similar results as seen for all of our other validation studies. This clearly demonstrates the validity of our proposed geometry extraction method.

Note also that all other validations than the one conducted for the torsion sphere were carried out on a big desktop computer using COMSOL Multiphysics rather than on hundreds of compute nodes.

6.5 Spacing experiment

An important feature of the homogenization based topology optimization is that we get a field on a coarse scale and can then de-homogenize to a much finer scale without recomputing the FEA.

This feature was tested with the experiment illustrated in Figure 17. In this figure we show the two cantilevers, de-homogenized with surface spacing N_s ranging from 5 to 20 surfaces. Our images clearly show the increasingly

Table 3. General overview of our mechanical performance. We compare the compliance-volume fractions of our de-homogenization and state-of-art by Stutz et al. [2020] for the 2D and Groen et al. [2020] for the 3D examples. Note that we compare to state-of-art values without post processing, since we did none. Our torsion sphere is evaluated on a 576×576×576 finite element grid with trilinear shape functions, code provided by [Groen et al. 2020]. The rest is evaluated using tetrahedral meshes [Hu et al. 2018] and commercial finite element software using quadratic shape functions [AB 2021].

Experiment	$C_h \cdot V_h$	$C_s \cdot V_s$	$\mathcal{F}_{CV}$	$\mathcal{F}_{CV}$ State-of-Art
2D center loaded clamped beam	6.55	7.44	1.14	$1.08 - 1.16$
Cantilever	22.85	25.64	1.12	$1.09 - 1.15$
Cantilever 3 layers	24.95	25.44	1.02	N/A
Electrical mast	9.81	10.80	1.19	$1.10 - 1.16$
L-shaped beam	58.49	63.44	1.08	$1.05 - 1.11$
Torsion sphere	1.40	1.98	1.41	$1.27 - 1.69$

fine structures, and in Table 3 we show that the mechanical performance does indeed converge toward the theoretical optimum as we refine the structure.

The compliance volume product is seen to first decrease and then increase. In principle, the product should get lower and lower with more and more layers. However, two factors work against this. 1) More layers require more and smaller elements to resolve feature sizes and hence smaller stiffness and larger compliance. This effect may partially be alleviated by ensuring same element size for all cases. 2) More layers results in physically thinner layers that are more sensitive to flaws in the mapping (as well as manufacturing errors). The latter is alleviated with the result where all three layers are above the minimum thickness or zero. Here, the additional layers were expected to provide transverse stiffness and hence the structure becoming less sensitive to mapping flaws. The compliance volume product correspondingly keeps decreasing with more layers.

As can be seen in Table 4 the time spent generating, selecting and super sampling the surfaces are quite stable, even under changes to the desired spacing, and thereby the complexity of the final output. This stability allows us to investigate a number of spacings quite fast to get a level of detail which match the desired output. After this has been achieved we can start the expensive volumetric generation.

A significant benefit of our method is that we do not need to tune a parameter if the desired level of detail changes. In [Groen et al. 2020] a minimal length-scale parameter is tuned to set a lower bound for plate thickness. However, we decouple finding the stream surfaces and synthesizing the solid. Hence, we are able to choose the output resolution based on the observed minimal thickness of the stream surfaces.

7 DISCUSSION AND FUTURE WORK

In this paper, we have introduced a novel method for creating multi-laminar structures that align to frame fields. The main challenge lies in the fact that even though we can easily make local structures that align with the frame field, we cannot easily assemble these into a global structure with continuous and connected layers. One way to approach this is through the introduction of a parametrization of the domain. Indeed, the previous methods of

Table 4. Mechanical performance and timings for the spacing experiment. Note that the time to generate the volumetric structure increases with the complexity of the structure.

Experiment	N_s	$\mathcal{F}_{CV}$	De-homogenization Time	Volumetric generation time
Cantilever	5	1.144	00:09:41	00:24:48
Cantilever	10	1.140	00:10:23	00:40:09
Cantilever	15	1.126	00:12:18	00:49:04
Cantilever	20	1.136	00:12:36	01:03:18
Cantilever 3 layers	5	1.066	00:08:06	00:31:56
Cantilever 3 layers	10	1.018	00:11:08	00:53:55
Cantilever 3 layers	15	1.014	00:11:46	01:19:40
Cantilever 3 layers	20	1.007	00:11:16	02:43:00

which we are aware require a parametrization of the domain. This is however *only* straight forward to compute in the guaranteed absence of singularities in the frame field.

Singularities are not a problem for the particular application that we are concerned with, namely de-homogenization, simply because the singularities only arise in solid regions, void regions, or they form curves perpendicular to the active laminations. In the latter case, there is clearly no problem, and in both former cases we would also stop stream surface tracing as the structures we seek cease to be laminar in solid and void regions. In the sense that our code does not have to explicitly take singularities into account, our approach is actually oblivious to singularities. In contrast, a parametrization based approach needs to explicitly deal with singularities by introducing seams and clearly cannot align perfectly to a non-integrable field. For the application of de-homogenization this translates into the pitfall that the parametrization modifies the resulting mechanical structures negatively. Moreover, a practical challenge when computing a parametrization is that the frame field must be combed – i.e. there must be a consistent labeling of the frame vectors. This can severely influence the robustness of an integration-based approach.

Compared to Groen et al. [2020], our method tends to be faster and produce results whose mechanical performance are within the range of what is reported by Groen et al. While Groen et al. produce models with better compliance volume products in the best cases, their method also exhibits considerable variation. In contrast, our experiments generating cantilevers at four different spacings ($N_s \in \{5, 10, 15, 20\}$) show that our method produces results whose mechanical properties vary only slightly, and the variation is easy to interpret as a function of N_s. It also bears mentioning, that our results are without any kind of post processing, yet they are relatively clean, exhibiting fewer spurious parts and tiny holes than the surfaces presented by Groen et al. [2020].

Admittedly, there are limitations to our approach. At present, we compute the thicknesses of the structural members based on the assumption that spacing between surfaces does not vary. However, the spacing does vary which leads to tapering structural members. As we can see from our compliance values, this is not a major concern, but we envision that a few steps of shape optimization on the de-homogenized structure could further increase the performance of the structures by removing excess material and adapting local thicknesses.

Additionally, structures only a few elements wide can be difficult to capture. To combat this, we will look into seeding surfaces in regions where surfaces are not already present. This will increase the amount of communication needed in an MPI setting, but could improve the initial stream surface set.

Manufacturing the volumetric structures may pose a few challenges. We produce closed wall structures since these are best from a mechanical point of view, but this choice entails that powder based 3D printers cannot be used. On the other hand, fused filament fabrication type printers should work. Such printers would likely build internal scaffolding which could be difficult or impossible to remove, but if we use short length scales, we don't expect this to be an issue except in the case of models such as the torsion sphere due to the large internal cavity.

Finally, we have focused on applying the method to fields arising from single load compliance minimization in this paper, but topology optimization is replete with problems where the proposed method might be applicable, hinting at future application areas. Moreover, the method could also be applied to frame fields of provenance unrelated to topology optimization, albeit singularities would probably need to be handled explicitly for such data.

ACKNOWLEDGMENTS

(a) Rectangular hole microstructure.

(b) Rank-2 microstructure with orthogonal layers.

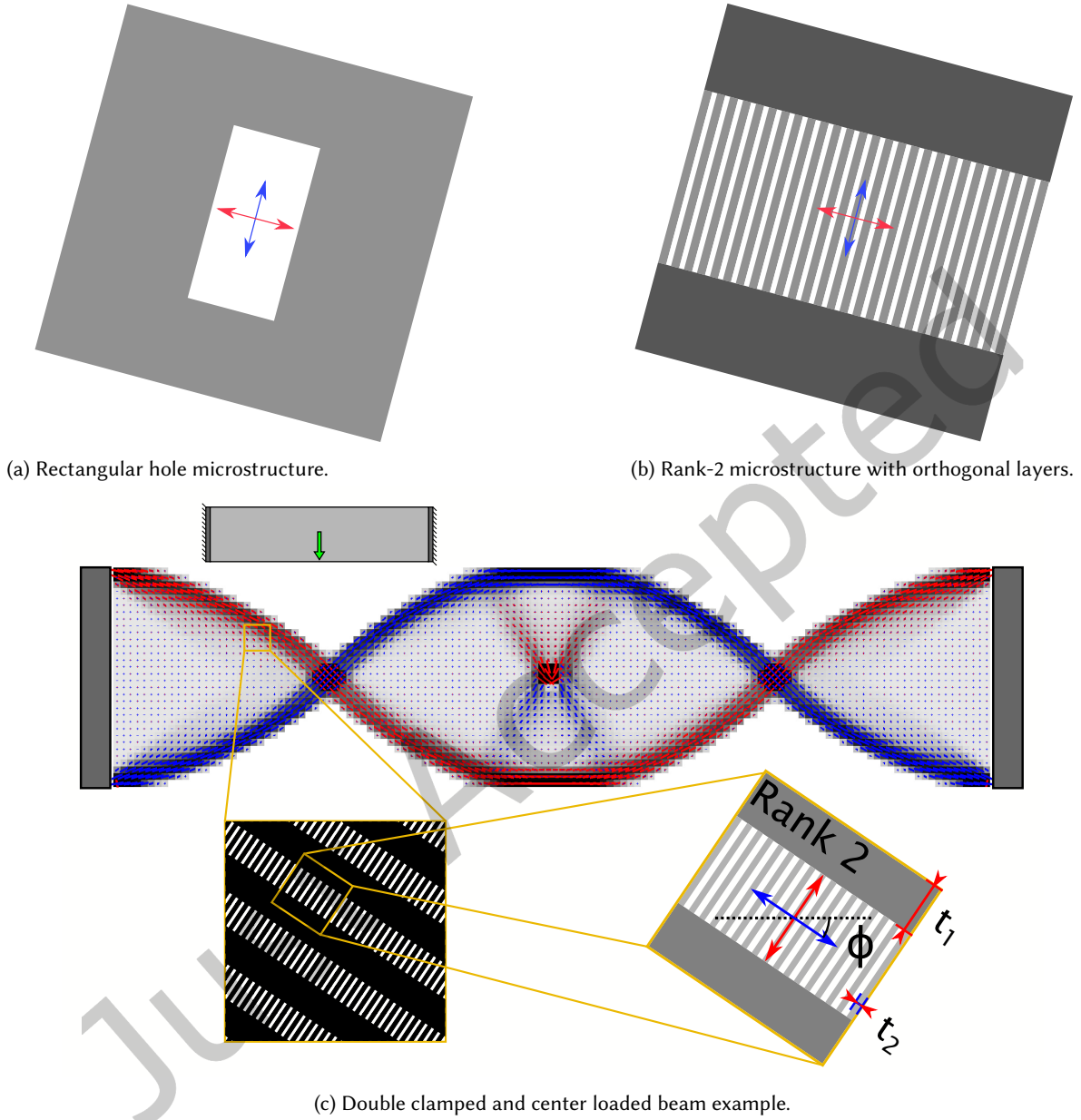(c) Double clamped and center loaded beam example.

Fig. 2. Figures a) and b) depict common anisotropic microstructures in two dimensions. On the left the rectangular hole microstructure introduced for topology optimization in Bendsøe and Kikuchi [1988]. On the right the rank-2 microstructure with orthogonal layers first used for topology optimization in Bendsøe [1989]. c) Double clamped and center loaded beam homogenization result based on rank-2 microstructure parametrization. The figure highlights the rotation of the microstructure ϕ and the thickness of the individual layers t_i.
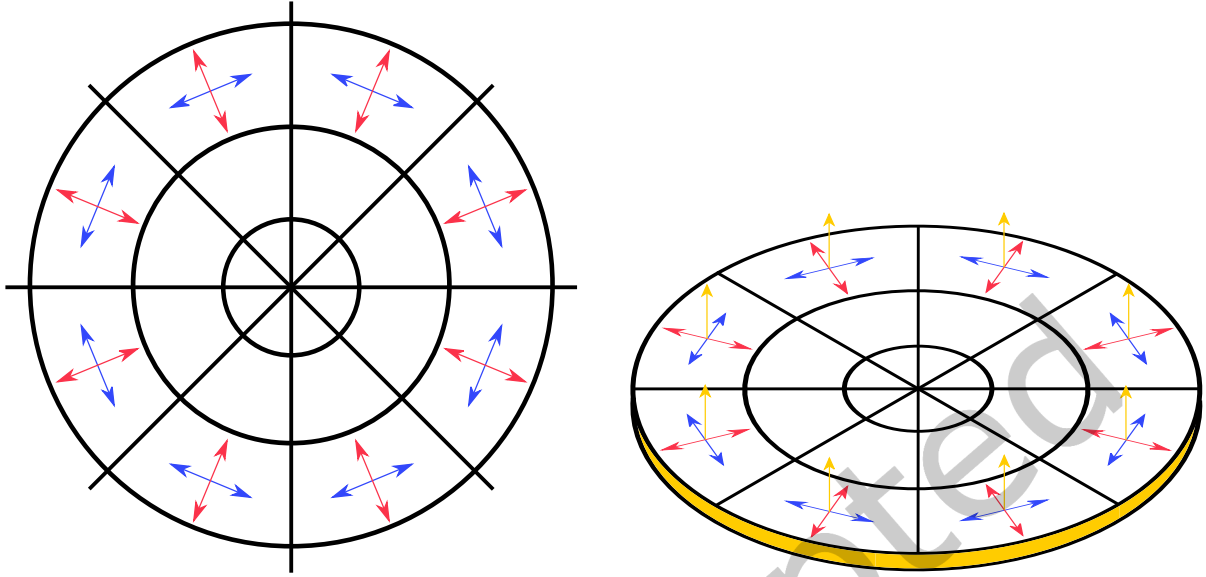
(a) Two-dimensional example of a singularity with index 1.

(b) The same singularity as in Figure 3a but now embedded in an orthogonal layer (in yellow) in three dimensions.

Fig. 3. On the left we see an example of a singularity with index 1 in two dimensions. It is clear that the rotational speed increases the closer we get to the singular point. Stutz et al. [2020] have shown, that the optimizer has an incentive to put such singularities in an either fully void or fully solid region, since there would always be shear forces acting on any non-solid microstructure at the singular point. On the right we see the same field and singularity embedded into an orthogonal layer (in yellow) in three dimensions. Note how the optimizer now can choose to fill the yellow layer with material and completely ignore the red and blue field, while still creating a stiff structure. Moreover, this singularity does not have to be in a completely solid region since the relative layer thickness of the yellow layer can be lower than 100 percent. In this case we refer to the microstructure as *transversely isotropic* since the microstructure is isotropic in one plane (the yellow one) but anisotropic perpendicular to this plane.
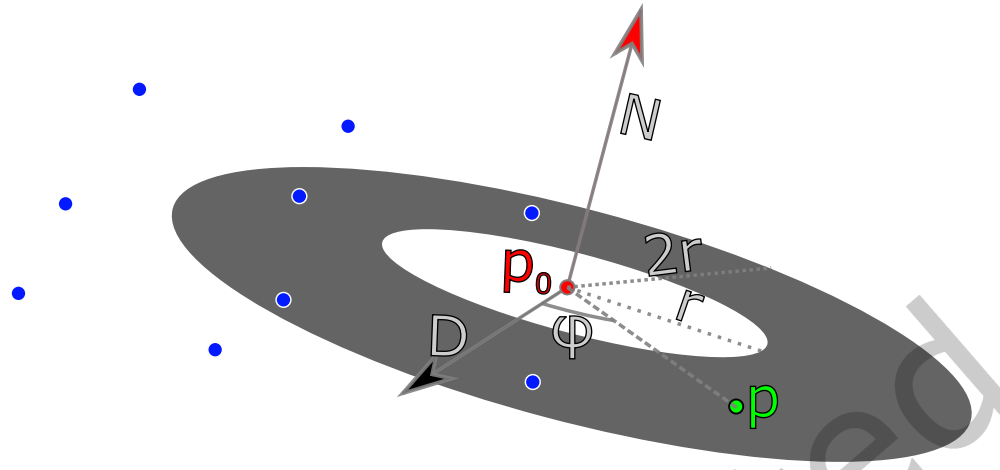
Fig. 4. Outline of how a new point **p** is generated at a random position inside the annulus around $\mathbf{p}_0$ oriented perpendicularly to the desired surface normal **N**.
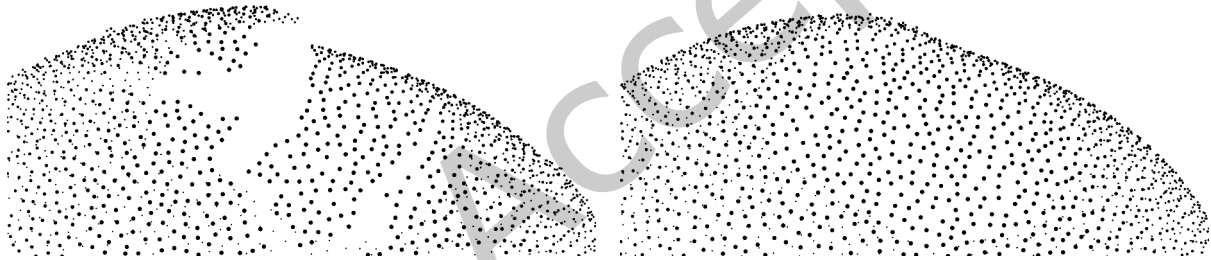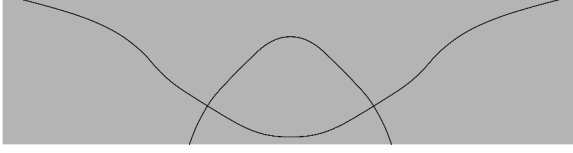

Fig. 5. Here we see the effect of drifting. Small deviations in point position and interpolation of the field over long distances lead to a crack in the surface. In the left image each point is computed using only a single preceding point. The right image shows the weighted scheme we employ.

(a) Two streamlines following two different orthogonal fields.

(b) Total response of the indicator $\bar{I}_s$ for the streamlines in Figure 6a. Green and blue elements activate the indicator once, orange elements activate the indicator twice.

Fig. 6. On the left, we see two streamlines following orthogonal field directions. On the right, we see the sum of the indicator $\bar{I}_s$ from Equation 1 for the two streamlines. Here the contributions of the streamlines are colored in blue and green in regions with value one. The orange highlighted regions are elements where both streamlines create an indicator response and subsequently the total value equals two. An optimizer would try to minimize the amount of these orange elements since it tries to minimize overlapping streamline-energies. This version of the indicator is blind for the fact that the two streamlines follow different fields. In order to be able to space out both family of streamlines correctly, we need to split the indicator as shown in Equation 2 and Figures 7 and 11.
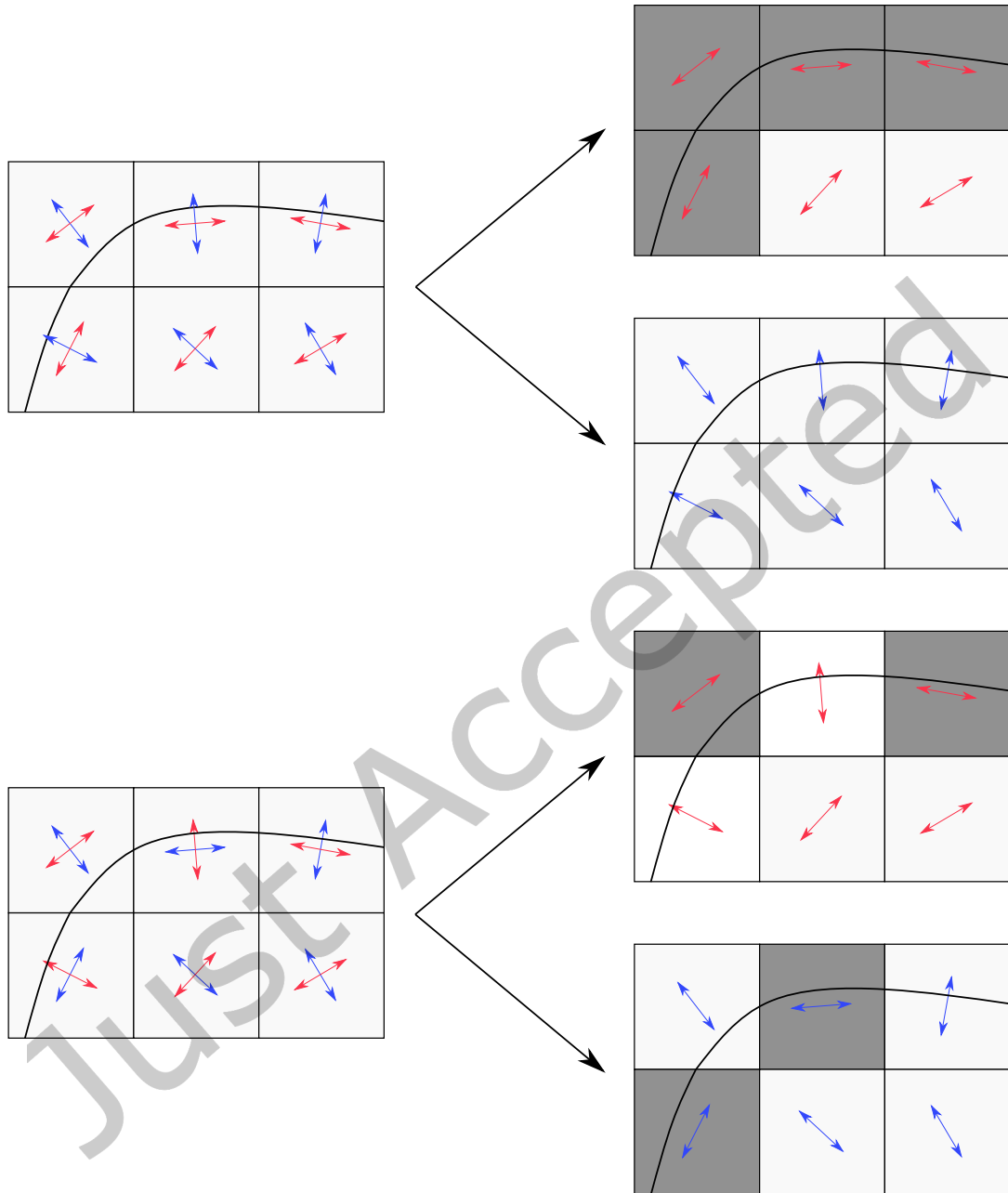
Fig. 7. Example of the indicator response created from a streamline traced in the same field but differently labeled. On the left, we see a streamline traced in a combed (above) and uncombed (below) version of a 4-direction field. On the right, we see the corresponding indicators for label "red" and label "blue", which are local labels. Note how the streamline activates only one of the two indicators for each element, as indicated by the gray coloring.
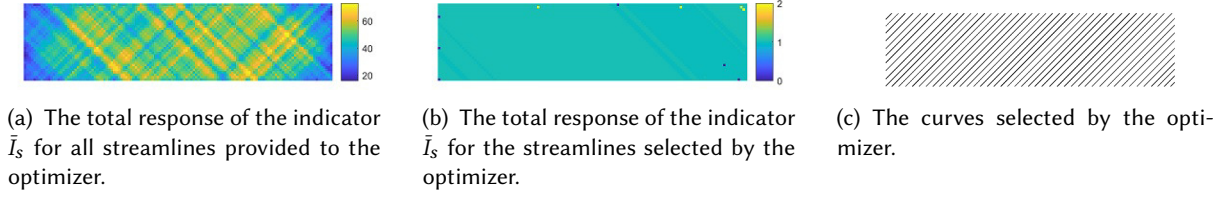
(a) The total response of the indicator $\bar{I}_s$ for all streamlines provided to the optimizer.

(b) The total response of the indicator $\bar{I}_s$ for the streamlines selected by the optimizer.

(c) The curves selected by the optimizer.

Fig. 8. Optimization results using the simple indicator defined in Equation 1.

(a) The response of the indicators $I_{s,1}$(top) and $I_{s,2}$(bottom) for all streamlines provided to the optimizer.

(b) The response of the indicators $I_{s,1}$(top) and $I_{s,2}$(bottom) for the streamlines selected by the optimizer.

(c) The curves selected by the optimizer.
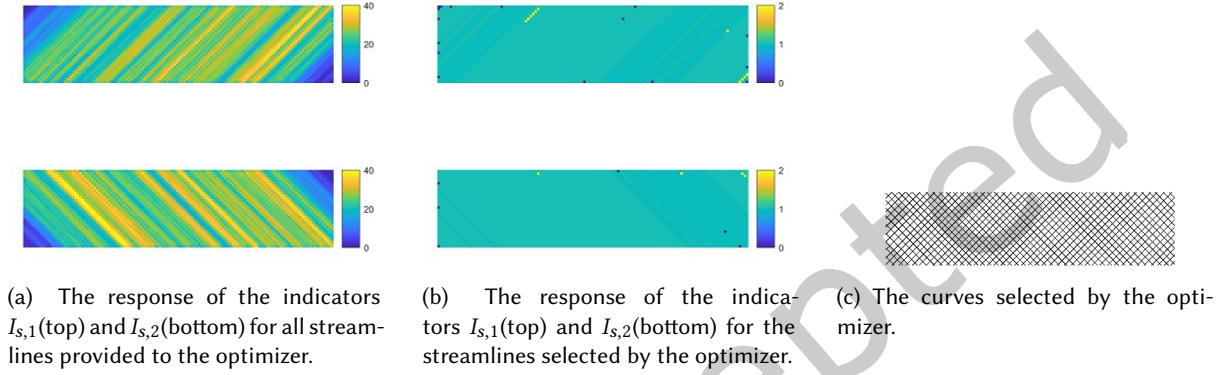
Fig. 9. Optimization results using the expanded indicator defined in Equation 2.

(a) The total response of the indicators $I_{s,1}$(top) and $I_{s,2}$(bottom) when using an L^1-norm for selecting streamlines.

(b) The total response of the indicators $I_{s,1}$(top) and $I_{s,2}$(bottom) when using an L^2-norm for selecting streamlines.

(c) The curves selected by the optimizer when using an L^1-norm.

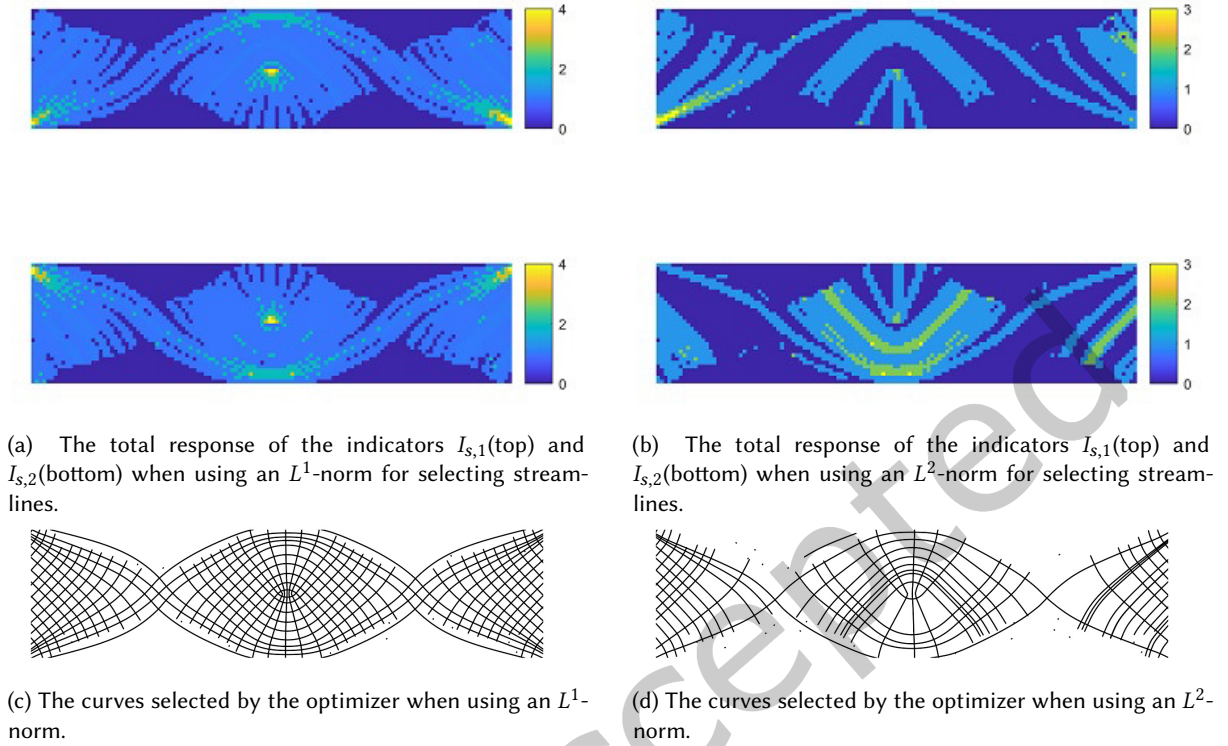(d) The curves selected by the optimizer when using an L^2-norm.

Fig. 10. Streamline selection results obtained when providing an L^1-norm or an L^2-norm to the optimizer. Note the missing streamlines that occur for the L^2-norm results due to multiply covered regions being penalized too harsh. Each line is bounded by local material properties and the design domain. This lead to some lines that span large parts of the domain and tiny lines only connecting the edges of a single valid element. This can give the illusion of continuous lines with a disconnect, however this is actually two independent lines. A clear example of this is the top and bottom most lines in Figure 10c that seem to be disconnected at the center.

(a) Indicator response $I_{s,1}$ for a combed field.

(b) Indicator response $I_{s,1}$ for an uncombed field.

(c) Indicator response $I_{s,2}$ for a combed field.

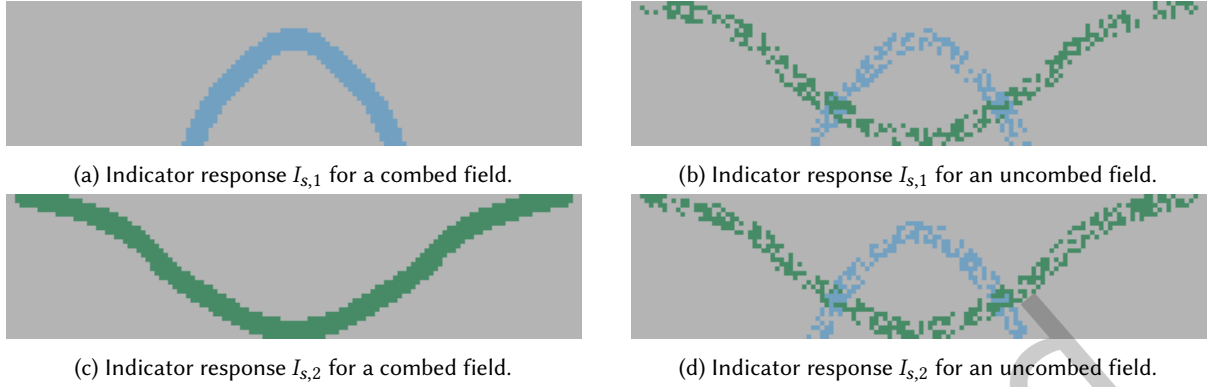(d) Indicator response $I_{s,2}$ for an uncombed field.

Fig. 11. Responses of the split indicator from Equation 2 for the streamlines depicted in Figure 6a. The indicator responses are shown for a combed and an uncombed version of the underlying field. Note how in a combed field the streamlines are separated by the indicators. In an uncombed field, the contributions of the streamlines split to both indicators. However, it is clear that the contributions of a streamline to the two indicators form a disjoint union. Regions with value two (highlighted in orange in Figure 6) do no longer exist.
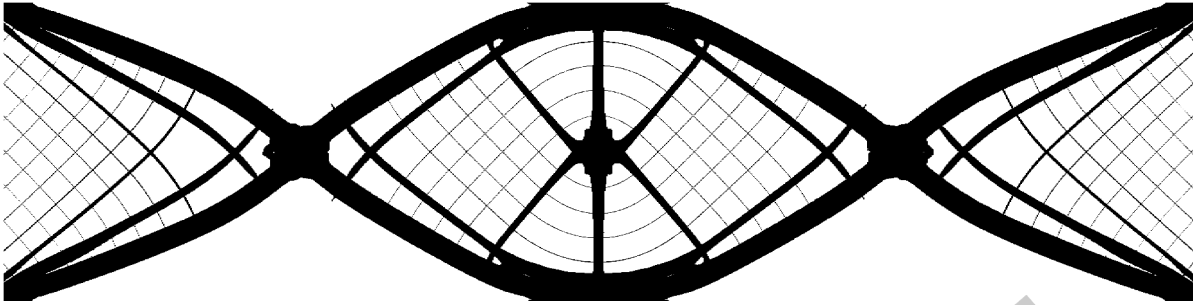
(a) De-homogenization using an integrative approach proposed in Stutz et al. [2020] yielding missing structural members around the central singularity, where the gradients have become almost zero. A resolution of 320×80 is used as an input mesh. The structure has a compliance-volume product of $C \cdot V = 7.30$. Stutz et al. [2020] report Compliance-Volume products of 7.05, 7.48, 7.63 and 21.39 for different alignment weights, where 7.05 is their best performing structure at an intermediate alignment weight and 21.39 is a failure case.
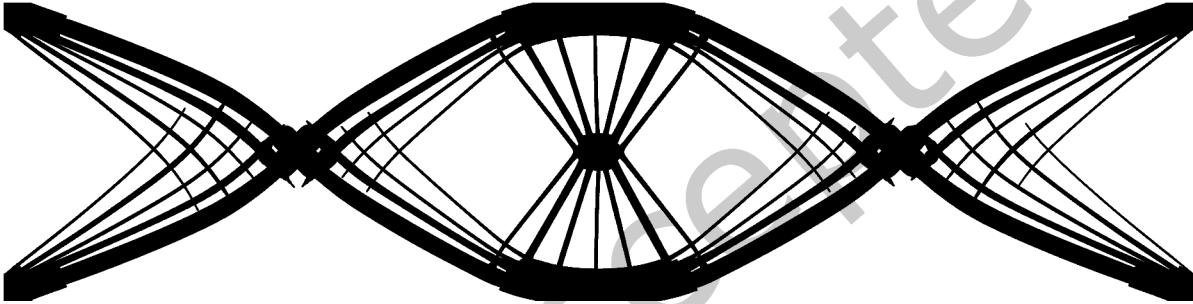
(b) Our approach yields a structure with evenly spaced structural members for the same 320×80 input field as used above. The compliance-volume product for this structure is $C \cdot V = 7.44$ and is therefore a mere 5.5% worse than Stutz et al. [2020] best value. However, with our approach we do not risk a failure case.

Fig. 12. Comparison between an integrative approach based on Stutz et al. [2020] yielding missing structural members and our approach which creates evenly spaced structural members.

(a) The classic cantilever.

(b) A cantilever where all 3 layers are forced to be active.

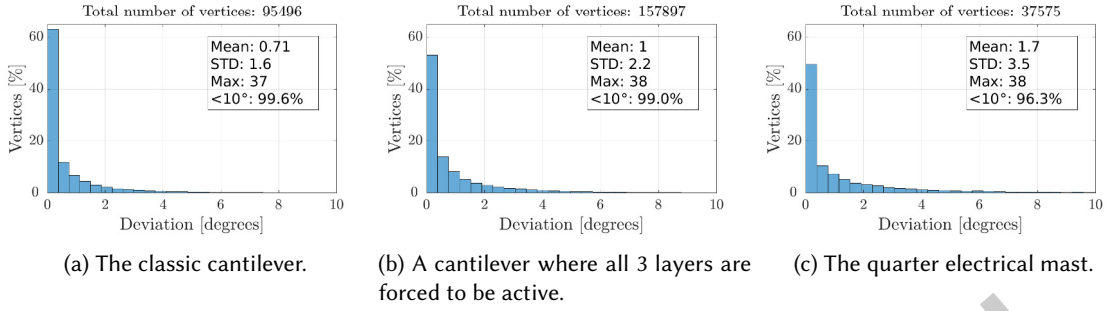(c) The quarter electrical mast.

Fig. 13. These histograms show the pointwise deviation from the frame field. This deviation is computed as the angular difference between the frame field and the surface normal of the triangulation.

(a) 0 iterations.

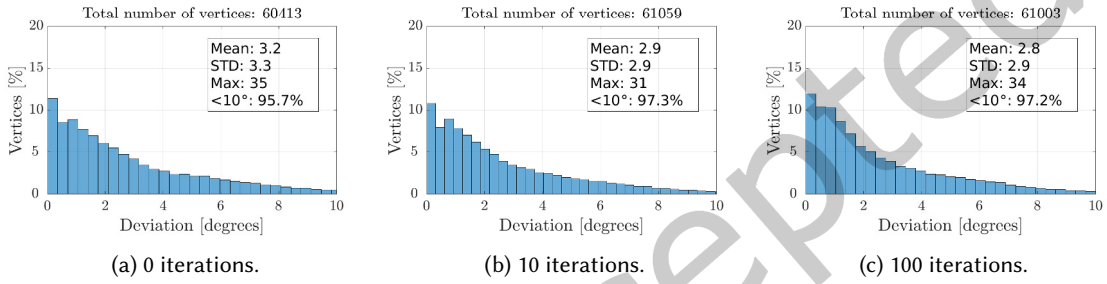(b) 10 iterations.

(c) 100 iterations.

Fig. 14. Alignment experiment using the torsion sphere, highlighting the effect of varying the number of iterations. Re-alignment is done before super-sampling. This causes differences in local point spacing which, in turn, affect the number of points inserted by the super-sampler.
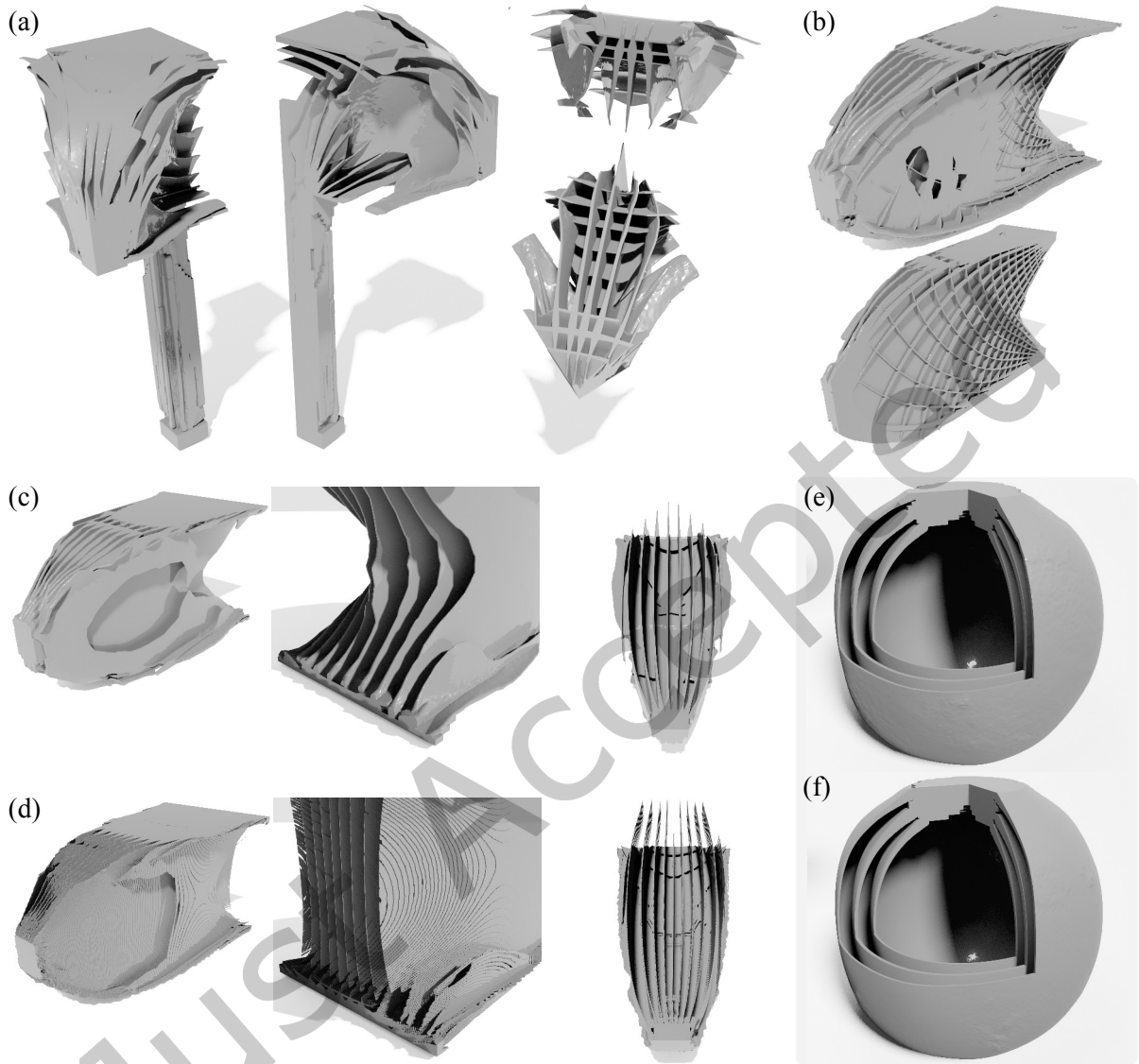
Fig. 15. This figure show five topology optimization examples created with our method from Section 5.2. In (a) we see the electrical mast example introduced in Geoffroy-Donders et al. [2020]. We show here three views: front, side, and cut open to reveal the interior of the structure. In (b) we show a cantilever produced from a homogenization solution where all three layers have been forced to be active. In (c) and (d) we show the results for a cantilever. Our method was used to produce the results in (c) whereas (d) shows the results produced using the method due to Groen et al. [2020]. From left to right, the three images in each row show the entire structure a close up and a view from above with the top cut away. In (e,f) we show the torsion sphere example introduced in Groen et al. [2020], where the boundary conditions with torsion applied are located in the top and the bottom. Note that we cut out an eighth to reveal the interior laminations of the torsion sphere. In (e) 10 iterations of the re-estimation procedure discussed in Section 4.1 whereas 100 iterations were used in (f).
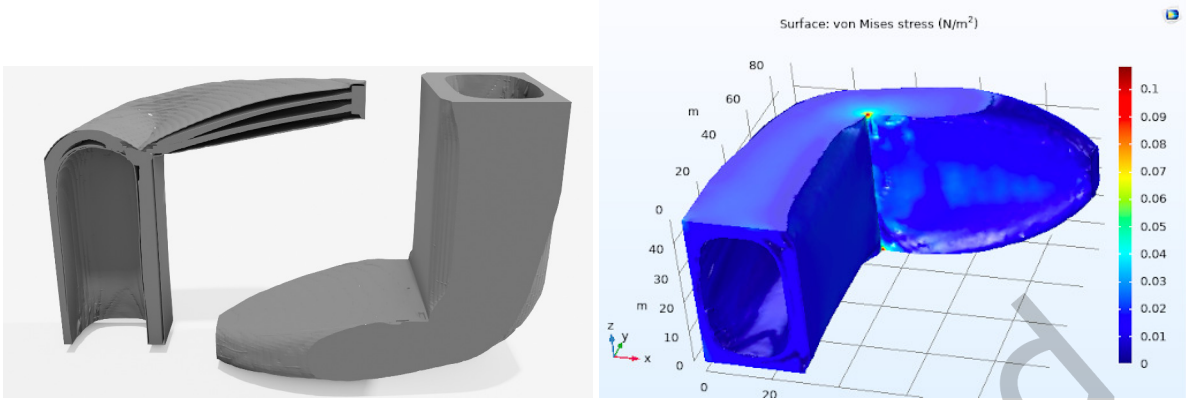
Fig. 16. This image shows the L-shaped beam de-homogenized. A cut-through version of the model is shown left, and the complete model is in the middle. On the right the Von Misses stresses are illustrated with COMSOL.

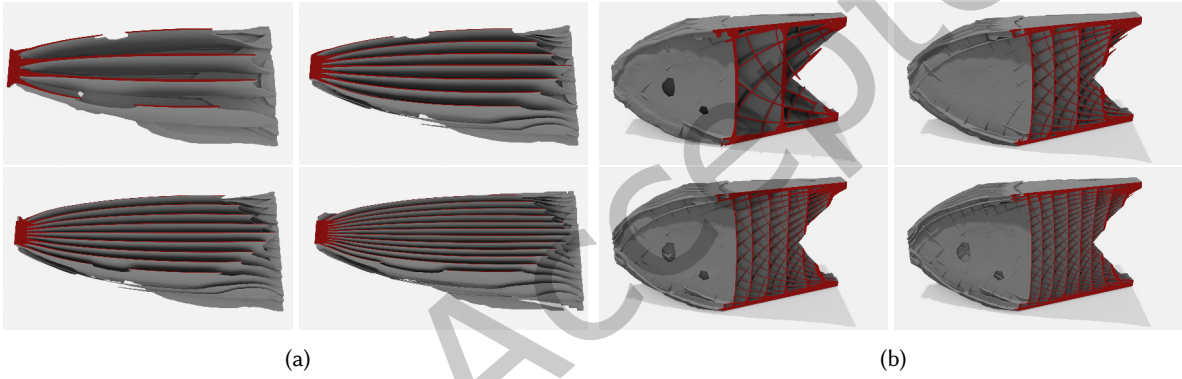(a)   (b)

Fig. 17. Cross sections of models from the Spacing Experiment where the cut surfaces are shown in red. In (a) the cantilever with at least one active layer, and in (b) the cantilever with three active layers. In both images, from left to right and top to bottom, the cantilever is shown with a surface spacing (N_s) that corresponds to 5, 10, 15 and 20 surfaces. Note that the cantilever with three active layers and a surface spacing $N_s = 20$ is computed on a grid which is 1.3 times finer along each dimension.
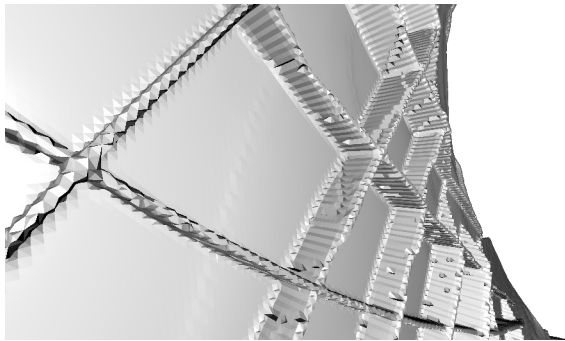
(a) Cantilever constructed from a $960 \times 480 \times 480$ voxel grid.
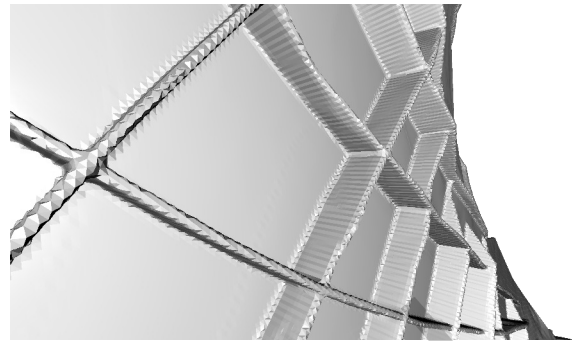
(b) Cantilever constructed from a $1248 \times 624 \times 624$ voxel grid.

Fig. 18. When surfaces have very small amounts of material associated with them and we want a high number of surfaces, we get very thin structures. This requires a more precise voxel grid to resolve, and for the Cantilever with 3 layers and $N_s = 20$ surfaces, we get artifacts at the resolution shown in (a).

A PARAMETERS

Table 5. Full parameter overview complementing Section 6.1. Here we list constants and parameters used for our experiments.

Experiment	Constants					Theoretical Parameters		Actual Parameters	
	$n_x \times n_y \times n_z$	γ	μ	ε	τ^r	$n_{\mathcal{S}}$	$n_x^r \times n_y^r \times n_z^r$	$n_{\mathcal{S}}$	$n_x^r \times n_y^r \times n_z^r$
Cantilever	$96 \times 48 \times 48$	4.8	0.05	0.1	1.5	400	$600 \times 300 \times 300$	480	$960 \times 480 \times 480$
Cantilever 3 layers	$96 \times 48 \times 48$	4.8	0.05	0.1	1.5	400	$600 \times 300 \times 300$	480	$960 \times 480 \times 480$
Electrical	$48 \times 48 \times 144$	4.8	0.05	0.1	1.5	500	$300 \times 300 \times 900$	240	$480 \times 480 \times 1440$
L-Shaped Beam	$96 \times 96 \times 48$	4.8	0.05	0.1	1.5	500	$600 \times 600 \times 300$	240	$960 \times 960 \times 480$
Torsion Sphere	$72 \times 72 \times 72$	7.2	0.05	0.1	1.5	300	$300 \times 300 \times 300$	400	$720 \times 720 \times 720$
Cantilever $N_s = 5$	$96 \times 48 \times 48$	9.6	0.05	0.1	1.5	200	$300 \times 150 \times 150$	480	$960 \times 480 \times 480$
Cantilever $N_s = 10$	$96 \times 48 \times 48$	4.8	0.05	0.1	1.5	400	$600 \times 300 \times 300$	480	$960 \times 480 \times 480$
Cantilever $N_s = 15$	$96 \times 48 \times 48$	3.6	0.05	0.1	1.5	600	$900 \times 450 \times 450$	480	$960 \times 480 \times 480$
Cantilever $N_s = 20$	$96 \times 48 \times 48$	2.4	0.05	0.1	1.5	800	$1200 \times 600 \times 600$	480	$960 \times 480 \times 480$
Cantilever 3 layers $N_s = 5$	$96 \times 48 \times 48$	9.6	0.05	0.1	1.5	200	$300 \times 150 \times 150$	480	$960 \times 480 \times 480$
Cantilever 3 layers $N_s = 10$	$96 \times 48 \times 48$	4.8	0.05	0.1	1.5	400	$600 \times 300 \times 300$	480	$960 \times 480 \times 480$
Cantilever 3 layers $N_s = 15$	$96 \times 48 \times 48$	3.6	0.05	0.1	1.5	600	$900 \times 450 \times 450$	480	$960 \times 480 \times 480$
Cantilever 3 layers $N_s = 20$	$96 \times 48 \times 48$	2.4	0.05	0.1	1.5	800	$1200 \times 600 \times 600$	480	$1248 \times 624 \times 624$

REFERENCES

Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. 2017. Giga-voxel computational morphogenesis for structural design. *Nature* 550, 7674 (2017), 84–86.

COMSOL AB. 2021. COMSOL Multiphysics®. https://www.comsol.com.

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, 13 pages.

Marco Avellaneda. 1987. Optimal bounds and microgeometries for elastic two-phase composites. *SIAM J. Appl. Math.* 47, 6 (1987), 1216–1228.

Mads Baandrup, Ole Sigmund, Henrik Polk, and Niels Aage. 2020. Closing the gap towards super-long suspension bridges using computational morphogenesis. *Nature Communications* 11, 1 (2020), 1–7.

Martin P Bendsøe. 1989. Optimal shape design as a material distribution problem. *Structural optimization* 1, 4 (1989), 193–202.

Martin Philip Bendsøe and Noboru Kikuchi. 1988. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering* 71, 2 (1988), 197–224.

Martin P. Bendsøe and Ole Sigmund. 2004. *Topology Optimization.* Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-05086-6

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)* 28, 3 (2009), 77.

Robert Bridson. 2007. Fast Poisson disk sampling in arbitrary dimensions. *ACM SIGGRAPH* (08 2007). https://doi.org/10.1145/1278780.1278807

Simon Calderan, Guillaume Hutzler, and Franck Ledoux. 2020. Dual-Based User-Guided Hexahedral Block Generation Using Frame Fields. *Proceedings of the 28th International Meshing Roundtable* (2020). https://doi.org/10.5281/ZENODO.3653430

Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual loops meshing. 31 (2012), 1–11. https://doi.org/10.1145/2185520.2185606

Marcel Campen and Leif Kobbelt. 2014. Dual strip weaving. 33 (2014), 1–10. https://doi.org/10.1145/2661229.2661236

Marcel Campen, Cláudio T. Silva, and Denis Zorin. 2016. Bijective maps from simplicial foliations. 35 (2016), 1–15. https://doi.org/10.1145/2897824.2925890

Steven Chapra. 2012. *Applied numerical methods with MATLAB for engineers and scientists.* McGraw-Hill, New York.

Anders Clausen, Niels Aage, and Ole Sigmund. 2016. Exploiting Additive Manufacturing Infill in Topology Optimization for Improved Buckling Load. *Engineering* 2, 2 (2016), 250–257. https://doi.org/10.1016/J.ENG.2016.02.006

DTU Computing Center. 2021. DTU Computing Center resources. https://doi.org/10.48714/DTU.HPC.0001

Gilles Francfort, François Murat, and Luc Tartar. 1995. Fourth-order moments of nonnegative measures on S2 and applications. *Archive for Rational Mechanics and Analysis* 131, 4 (1995), 305–333. https://doi.org/10.1007/bf00380913

Xifeng Gao, Wenzel Jakob, Marco Tarini, and Daniele Panozzo. 2017. Robust Hex-Dominant Mesh Generation Using Field-Guided Polyhedral Agglomeration. *ACM Trans. Graph.* 36, 4, Article 114 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073676

Perle Geoffrey-Donders. 2018. *Homogenization method for topology optmization of struc-tures built with lattice materials.* Ph. D. Dissertation.

Perle Geoffroy-Donders, Grégoire Allaire, and Olivier Pantz. 2020. 3-d topology optimization of modulated and oriented periodic microstructures by the homogenization method. *J. Comput. Phys.* 401 (2020), 108994. https://doi.org/10.1016/j.jcp.2019.108994

Francisca Gil-Ureta, Nico Pietroni, and Denis Zorin. 2020. Reinforcement of General Shell Structures. *Acm Transactions on Graphics* 39, 5 (2020), 3375677. https://doi.org/10.1145/3375677

M. Gopi, S. Krishnan, and C. T. Silva. 2000. Surface reconstruction based on lower dimensional localized Delaunay triangulation. *Computer Graphics Forum* 19, 3 (2000), 467–478. https://doi.org/10.1111/1467-8659.00439

Michael Grant and Stephen Boyd. 2008. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura (Eds.). Springer-Verlag Limited, 95–110. http://stanford.edu/~boyd/graph_dcp.html.

Michael Grant and Stephen Boyd. 2014. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. http://cvxr.com/cvx.

J.P. Groen and O. 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 (online since April 2017).

Jeroen P. Groen, Florian C. Stutz, Niels Aage, Jakob A. Bærentzen, and Ole Sigmund. 2020. De-homogenization of optimal multi-scale 3D topologies. *Computer Methods in Applied Mechanics and Engineering* 364 (2020). https://doi.org/10.1016/j.cma.2020.112979

Hao-Xiang Guo, Xiaohan Liu, Dong-Ming Yan, and Yang Liu. 2020. Cut-enhanced PolyCube-maps for feature-aware all-hex meshing. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 106–1.

Ingrid Hotz, Jaya Sreevalsan-Nair, Hans Hagen, and Bernd Hamann. 2010. Tensor Field Reconstruction Based on Eigenvector and Eigenvalue Interpolation. In *Scientific Visualization: Advanced Concepts*, Hans Hagen (Ed.). Dagstuhl Follow-Ups, Vol. 1. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 110–123. https://doi.org/10.4230/DFU.SciViz.2010.110

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral meshing in the wild. *ACM Trans. Graph.* 37, 4 (2018), 60–1.

Jin Huang, Yiying Tong, Hongyu Wei, and Hujun Bao. 2011. Boundary aligned smooth 3D cross-frame field. In *ACM transactions on graphics (TOG)*, Vol. 30. ACM, 143.

J. P. M. Hultquist. 1992. *Constructing stream surfaces in steady 3D vector fields.* 171–178 pages. https://doi.org/10.5555/949685.949718,10.5555/949685

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. 2002. Dual Contouring of Hermite Data. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 339–346. https://doi.org/10.1145/566570.566586

Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. QuadCover-Surface Parameterization using Branched Coverings. In *Computer graphics forum*, Vol. 26. Wiley Online Library, 375–384.

S. D. Larsen, O. Sigmund, and J. P. Groen. 2018. Optimal truss and frame design from projected homogenization-based topology optimization. *Structural and Multidisciplinary Optimization* 57, 4 (2018), 1461–1474. https://doi.org/10.1007/s00158-018-1948-9

Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-Band Topology Optimization on a Sparsely Populated Grid. 37, 6, Article 251 (2018), 14 pages. https://doi.org/10/gg55hf

Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. 2020. LoopyCuts: Practical Feature-Preserving Block Decomposition for Strongly Hex-Dominant Meshing. *ACM Trans. Graph.* 39, 4, Article 121 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392472

Gustavo M. Machado, Filip Sadlo, and Thomas Ertl. 2014. Image-based streamsurfaces. *Brazilian Symposium of Computer Graphic and Image Processing* (2014), 6915327. https://doi.org/10.1109/SIBGRAPI.2014.30

Anthony George Maldon Michell. 1904. LVIII. The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8, 47 (1904), 589–597. https://doi.org/10.1080/14786440409463229 arXiv:https://doi.org/10.1080/14786440409463229

J Milnor. 1970. Foliations and foliated vector bundles, mimeographed notes. *Institute for Advanced Study, Princeton* (1970).

MOSEK ApS. 2021. *MOSEK Fusion API for C++ manual, Version 9.2.*

Peter Murdoch, Steven Benzley, Ted Blacker, and Scott A Mitchell. 1997. The spatial twist continuum: A connectivity based method for representing all-hexahedral finite element meshes. *Finite elements in analysis and design* 28, 2 (1997), 137–149.

Saifeng Ni, Zichun Zhong, Jin Huang, Wenping Wang, and Xiaohu Guo. 2018. Field-Aligned and Lattice-Guided Tetrahedral Meshing. *Computer Graphics Forum* 37, 5 (aug 2018), 161–172. https://doi.org/10.1111/cgf.13499

Matthias Nieser, Ulrich Reitebuch, and Konrad Polthier. 2011. Cubecover-parameterization of 3d volumes. In *Computer graphics forum*, Vol. 30. Wiley Online Library, 1397–1406.

Manfred Padberg and Giovanni Rinaldi. 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review* 33, 1 (1991), 60–100.

David Palmer, David Bommes, and Justin Solomon. 2019. Algebraic Representations for Volumetric Frame Fields. *arXiv preprint arXiv:1908.05411* (2019).

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *Acm Transactions on Graphics* 34, 4 (2015), 135. https://doi.org/10.1145/2766937

Olivier Pantz and Karim Trabelsi. 2008. A post-treatment of the homogenization method for shape optimization. *SIAM Journal on Control and Optimization* 47, 3 (2008), 1380–1398.

Olivier Pantz and Karim Trabelsi. 2010. Construction of minimization sequences for shape optimization. In *2010 15th International Conference on Methods and Models in Automation and Robotics*. IEEE, 278–283.

Pauli Pedersen. 1989. On optimal orientation of orthotropic materials. *Structural optimization* 1, 2 (1989), 101–106.

Nicolas Ray, Dmitry Sokolov, and Bruno Lévy. 2016. Practical 3d frame field generation. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 233.

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to control elasticity in 3D printing. *Acm Transactions on Graphics* 34, 4 (2015), 136. https://doi.org/10.1145/2766926

Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, Ming C. Lin and Dinesh Manocha (Eds.). Lecture Notes in Computer Science, Vol. 1148. Springer-Verlag, 203–222. From the First ACM Workshop on Applied Computational Geometry.

Ole Sigmund, Niels Aage, and Erik Andreassen. 2016. On the (non-)optimality of Michell structures. *Structural and Multidisciplinary Optimization* 54, 2 (2016), 361–373. https://doi.org/10.1007/s00158-016-1420-7

Ole Sigmund and Kurt Maute. 2013. Topology optimization approaches. *Structural and Multidisciplinary Optimization* 48, 6 (01 Dec 2013), 1031–1055. https://doi.org/10.1007/s00158-013-0978-6

Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary element octahedral fields in volumes. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 28.

F.C. Stutz, J.P. Groen, O. Sigmund, and J.A. Bærentzen. 2020. Singularity Aware De-Homogenization for High-Resolution Topology Optimized Structures. *Structural and Multidisciplinary Optimization* 62 (2020), 2279–2295. https://doi.org/10.1007/s00158-020-02681-6

Kenshi Takayama. 2019. Dual sheet meshing: An interactive approach to robust hexahedralization. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 37–48.

Jun Wu, Weiming Wang, and Xifeng Gao. 2019. Design and optimization of conforming lattice structures. *IEEE transactions on visualization and computer graphics* 27, 1 (2019), 43–56.