We introduce an efficient method for designing shell reinforcements of minimal weight. Inspired by classical Michell trusses, we create a reinforcement layout whose members are aligned with optimal stress directions, then optimize their shape minimizing the volume while keeping stresses bounded.

We exploit two predominant techniques for reinforcing shells: adding ribs aligned with stress directions and using thicker walls on regions of high stress. Most previous work can generate either only ribs or only variable-thickness walls. However, in the general case, neither approach by itself will provide optimal solutions.

By using a more precise volume model, our method is capable of producing optimized structures with the full range of qualitative behaviors: from ribs to walls, and smoothly transitioning in between. Our method includes new algorithms for determining the layout of reinforcement structure elements, and an efficient algorithm to optimize their shape, minimizing a non-linear non-convex functional at a fraction of the cost and with better optimality compared to standard solvers.

We demonstrate the optimization results for a variety of shapes, and the improvements it yields in the strength of 3D-printed objects.
structures form, in the limit of low volumes, classical Michell structures and can be obtained by solving a convex optimization problem. It is also well understood for pure bending problems for plates, i.e., the special case of flat shells with loads orthogonal to the surface; in this case, it also reduces to a different convex problem.

The situation is far more complex for the reinforcement of shells embedded in three dimensions. For these shells, the weight-optimal structure may be locally either beam-like, forming ribs aligned with stress directions, or membrane-like, forming variable-thickness walls with no perforation [Sigmund et al. 2016]. The first case typically corresponds to bending-dominated regions while the second to areas dominated by in-plane forces. The optimal local structure is determined by the surface shape, the supports, and the loads.

In this general case, the problem is no longer convex and cannot be optimally solved either by methods that assume that the result is only a variable thickness shell or by Michell-truss type methods.

In this paper, we propose a novel efficient computational method for constructing optimized reinforcement structures for shells, naturally producing a full range of behaviors spanning the space between variable-thickness shell and rib-reinforcement. Our approach can be applied to reinforcing any types of free-form manufactured objects: 3D printed plastic or metal shapes, as well as structures produced by casting.

We partition the problem into three steps (Figure 1): (1) determine the field of (approximately) optimal stress directions; (2) construct the skeleton of the reinforcement structure that follows these directions, forming polygonal (predominantly) quad cells aligned with the field; (3) optimize how material is distributed inside the cells.

Main contributions.

- For computing the field of optimal stress directions, we developed a generalization of Hencky-Prandtl nets which takes bending into account and can still be solved by minimizing a convex energy.
- For material distribution optimization, we use a low-parametric structure model for cells to efficiently optimize the distribution of the material. As the global optimization problem is fundamentally non-convex (we discuss the reasons on Section 3), to solve it we propose an efficient global/local method which shows stable and fast convergence behavior.

We validate our approach by optimizing several 3D shapes. This evaluation shows that our method can handle shells with arbitrary curvature, and successfully transitions between membrane- and bending-dominated regions, obtaining the expected optimal sub-structures. We demonstrate that, by optimizing jointly for bending and compression/tension dominated regimes, we obtain lighter structures than previous work.

2 RELATED WORK

Our work builds on the ideas from classical structure design for 2D elasticity and plates, with the key ones originating the work of Michell [1904].

We complement these fundamental ideas with quadrangulation techniques which can be reinterpreted as a way to transition from an infinite continuum of field-aligned beams to its discretization. We use a variation of Bommes et al. [2009], but any conforming, field-aligned method could be used (e.g., [Aigerman and Lipman 2015; Campen et al. 2015; Ebke et al. 2016; Kälberer et al. 2007; Myles et al. 2014a]). The optimization method for computing the optimal strain field can be viewed as a specialized cross-field optimization method. Similarly to the recently proposed method of Knöppel et al. [2015], it has the advantage of being convex.

We refer to Vaxman et al. [2016] for a complete overview of the related work on field design and to Bommes et al. [2013] for quadrangulation.

Shell Optimization. The closest works to ours are the recent works Kilian et al. [2017] and Li et al. [2017], with which we share a number of ideas. The former describes an elegant connection between curvature and Michell trusses and optimizes the surface shape so principal stress and curvature directions coincide. Only membrane forces are considered, and the volume approximation they use is valid for narrow beams (see Section 3). Similar to our work, Li et al. [2017] keeps the shell surface fixed. This work considers a network of ribs, aligned with stress lines, and minimizes their volume; similarly to Kilian et al. [2017], this work also uses a narrow-beam approximation for the volume, and always produces a thin-beam structure. The cross-section shape of individual beams is optimized, which produces additional weight reduction. We discuss differences to these works in more detail in Section 8.

On the other extreme, Zhao et al. [2017] considers the optimization of variable shell thickness, while keeping the topology fixed, which is suboptimal for bending reinforcement. Our work aims to bridge the gap between these extremes.

The approach of Pietroni et al. [2015] aligns a network of beams to an input stress field. Another recent related work, Jiang et al. [2017] considers structures made out of beams with a small number of distinct cross-sections. Both methods are suitable for architectural design; instead, we focus on applications, like 3D printing, which allow greater flexibility of structures.

Structural Optimization. The literature on structural optimization is quite extensive, and there is no chance that we can do justice
to all of it. The main types of approaches found in the literature include topology optimization methods (homogenization, SIMP, or ESO-based), analytic methods for optimal structures directly based on Michell-type theories, and methods based on shape derivatives (using an explicit or implicit evolving surface representation). Important books, which include reviews of many other works are Rozvany [1976], Allaire [2002], Bendsoe and Sigmund [2004], as well as recent reviews, Munk et al. [2015] and Sigmund and Maute [2013].

The most prevalent methods in topology optimization of structures are based on SIMP-type methods (see Bendsoe and Sigmund [2004] for a review), which relax the problem to optimizing a density over a domain, which is then converted to a structure by thresholding. This approach has many advantages, including simplicity of implementation [Sigmund 2001], connection to homogenization theory, flexibility in integrating functionals, and ease of scalable implementation [Aage et al. 2015; Wu et al. 2016]. Nevertheless, the result will typically depend on initialization: for the complex topology to emerge, the domain needs to be discretized using a fine grid. The parameters of the result (e.g., the sparsity of the structure, or minimal thickness) need to be controlled indirectly through algorithm parameters. Finally, the result is a voxelized structure, which then needs to be converted in some way to a form more suitable for manufacturing. In comparison, our method directly produces solutions based on a globally optimal field (in low-volume limit) and a beam skeleton for the optimized structure, which can be directly adjusted by the user in a variety of ways (e.g., converted to a spline-based CAD model if desired). We compare in more detail in Section 8.

Ground structure methods are among the oldest methods for optimizing the topology of truss/beam structures. These methods start with a structure consisting of a large number of redundant beams and optimize it to determine the cross-sections, which automatically eliminates some of the beams. Recent examples of applying these type of methods include Sokół [2011], and Zegard and Paulino [2014, 2015]. Compared to our approach, ground structure methods have to restrict the directions of beams to a small set, which affects both optimality and flexibility of the design. The larger the initial set of beams, the closer they may approximate the optimal result. In computer graphics, the ground structure method was used early in Smith et al. [2002] for truss structure design. Panetta et al. [2015] used a version of a ground structure method to obtain initial topologies for computing microstructures with prescribed material properties, followed by shape optimization.

To a great extent, our work was inspired by the beautiful structures explored in the literature on analytic or semi-analytic structure design, e.g. Rozvany [2012], which includes many examples of exact problem solutions, such as Hencky-Prandtl nets.

Our goal is to use this type of ideas in the general setting of curved surface domains, taking advantage of the optimality criteria and insights into the structure of the solutions. A concise exposition of the theory underlying Michell-type optimal layouts can be found in Strang and Kohn [1983]. We note that the application of Michell-type structures in 3D is only appropriate for certain problem settings: e.g., Sigmund et al. [2016] observed that, with no lower-bound constraints on shell thickness, variable thickness shells are likely to emerge as a solution. While their analysis is limited to compliance minimization, our experiments show it also applies to weight minimization (see Figures 23, 24, and 25).

Shape-derivative based optimization techniques (e.g., Allaire and Jouve [2008]) can obtain very good results when one needs to improve an existing design, by evolving the shape to a local minimum. However, while level-set methods of this type allow for topology changes, the result does vary considerably depending on the starting point. In contrast, our goal is to obtain a starting point that is close to the global optimum, as long as the desired structure has a relatively low volume.

More recently, homogenization-based topology optimization has been used to create high-resolution manufacturable structures (e.g. Geoffroy-Donders et al. 2017, 2018; Groen and Sigmund 2017). Several fields corresponding to structure parameters are optimized for a coarse mesh and later projected into a high-resolution mesh creating complex microstructures. This approach was applied to generate structures for 3D volumes [Geoffroy-Donders et al. 2018]; applying this technique to thin shells would require using a very fine resolution, as the dimension in the normal direction of the shell needs to be resolved, making the approach computationally expensive. It may also create impossible to manufacture fine-scale structures with void/solid alteration along the normal direction of the shell.

Digital fabrication. The works closest to ours in this domain are Li and Chen [2010], Tam et al. [2015], and Tam [2015]. These methods are based on constructing structures from stress lines on surfaces which, while different from the optimal fields we compute, are often a close approximation. The overall pipeline of the method of Li and Chen [2010] is similar to ours: this work starts with a field, and construct trusses following the field by tracing lines from supports to loads. The method is limited to planar elasticity and demonstrated only for relatively simple structures.

Tam et al. [2015] uses FDM to add material directly along the principal stress lines, on 2.5D surfaces. The main problems they solve are stress line generation and selection. They minimize strain energy subject to a maximal total print length (i.e max material) and consistent maximum spacing between lines.

In 3D, Arora et al. [2019] builds volumetric Michell trusses by creating a stress-aligned 3D texture parametrization and extracting a truss structure from it. The stress field corresponds to the deformation of the initial mesh filled with solid material, not taking into account the redistribution of stress resulting from removing material. The cross-section area of all trusses are assumed to be the same; as a consequence, the volume of the resulting structure is sub-optimal.

Applying topology optimization (SIMP and ground structure methods) to 3D printing applications is discussed in Zegard and Paulino [2016].

3 MOTIVATING EXAMPLES

To motivate our method, we start with simple examples of qualitatively different behavior of optimized structures. With these examples, we demonstrate that in general, for in-plane loading, using a thicker surface is significantly more efficient than using narrow
protruding ribs, and this solution cannot be obtained when using the convex volume approximations used in previous work.

The two key behaviors of optimal shell-like structures, observed in special-case analytic solutions and topology optimization (cf. [Sigmund et al. 2016]), are the formation of discrete narrow ribs protruding from the surface in bending-dominated cases (most forces are perpendicular to the surface), and relatively smooth variation in shell thickness in the pure tensile/compression case (in-plane forces), as shown in Figure 3.

![Image](Fig. 3. (a) optimal structure for a standard cantilever test case with variable thickness, cf. [Sigmund et al. 2016] (b) bending plate optimization, subject to uniform vertical loads, resulting in a rib structure qualitatively consistent with analytic results; (c) an optimized pipe structure, subject to internal pressure, exhibiting a mix of behaviors.

These two behaviors are not observed in the simplified models of beam networks approximating a surface: in a typical network, beams do not expand in the direction parallel to the surface, to merge into a variable-thickness shell optimal in such cases. It turns out that this is due to the qualitatively inaccurate volume computation with the volume of the beam network approximated by the sum of individual beam volumes. We now consider two simple examples showing why this is the case.

First, we consider optimization of a single horizontal beam of width \( w \) and height \( h \), clamped at one end, and loaded at the other, at an angle \( \alpha \) to the beam direction (Figure 4, left). This example clarifies optimal behavior when there is stress in only one direction, both for bending (\( \alpha = \pi/2 \)) and tension (\( \alpha = 0 \)). The second example involves two intersecting beams (Figure 4, right). It is a simple model for a piece of a surface where there is stress in two directions.

For a single beam with an end load \( (F \cos \alpha, F \sin \alpha) \), the maximal stress is proportional to \( F \cos \alpha/(wh) + F \sin \alpha/(h^2w) \), the sum of the membrane stress along the beam and the bending stress. We optimize the beam volume \( V(w,h) = whl \) keeping this stress constant \( \sigma_0 \). Eliminating \( w \) using the constraint, we get \( V = (lF/\sigma_0)(\cos \alpha + \sin \alpha/y) \), where \( y = h/l \).

When \( \sin \alpha \neq 0 \), the optimal solution maximizes the relative thickness \( y \). In contrast, when forces act along the beam (\( \alpha = 0 \)), the volume value is fixed by the constraint on stress and the choice of \( y \) makes no difference. We conclude that, for single beams, the solution can always be taken to be "thick and narrow" beams (we refer to them as ribs), with maximal possible thickness. The situation is more complicated for surfaces. In the second example, we approximate the surface locally by beams aligned with the perpendicular stress directions (Figure 4, right). For simplicity, we assume the forces, widths \( w \), lengths \( l \), and thicknesses \( h \) to be the same for both beams. If we view the intersecting beams individually and approximate the total volume as \( V = 2whl = 2wyh^2 \), then the reasoning above applies to each beam: for in-plane forces (\( \alpha = 0 \)) we get \( wy = F \), and the optimal volume is \( V = 2Fh^2 \), independent of the choice of \( h \). However, even a small bending component will prioritize maximal \( h \) solutions, so both beams will be thick and narrow.

Considering beams in separation ignores the fact that the intersection area of the beams is counted twice: this part of material is performing "double work", supporting loads in two directions along two beams. The correct combined volume of two beams is given by

\[
V'(w,y) = 2why^2 - w^2yl,
\]

assuming the same beam width and thickness for both. As before, for in-plane forces (\( \alpha = 0 \)), we have the constraint \( wy = F \). Replacing \( y \), the functional \( V' \) can be expressed then as \( V'(w) = Fl(2l - w) \).

From this expression, it is clear the minimum volume is obtained maximizing \( width \), as opposed to \( thickness \). For a maximal width \( l \), the optimal volume is \( Fh^2 \). In comparison, if we use a large relative thickness \( y \), then optimal \( w = Fl/y \approx \epsilon \) is close to zero, and the volume \( V'(\epsilon) \) is close to \( 2Fh^2 \), two times higher than optimal \( V'(l) \).

More generally, a combination of bending and membrane forces are required to keep an arbitrarily shape structurally stable (Figure 5). In this case, two intersecting beams with an out-of-plane load in addition to in-plane, there is an optimal trade-off between \( w \) and \( h \) minimizing the volume.

![Image](Fig. 5. Loaded shells dominated by bending and membrane forces, and a mix of these. Support nodes are highlighted in green while loads are shown in red. Loads on (a) and (b) include, respectively, external and internal pressure.)

If we further impose constraints on maximal and minimal surface thickness, even in membrane-dominated areas, ribs would form, because the optimal solid shell there would be too thin. A general...
optimization method should be able to smooth between grillage-like structures for bending dominated areas and “thin and wide” structures for the rest of the surface.

We conclude, from these examples, that to reinforce a shell in a manner close to optimal for arbitrary loads and shell shape, both solid variable-thickness and rib-like structures may be required in different areas of the surface, and for these to emerge, in a beam-based optimization problem, a non-convex volume function accounting for beam intersection areas has to be used.

4 PROBLEM FORMULATION

We start with a description of the variable-thickness perforated surface shell structure that we use to model surface reinforcement, and the optimization problem we aim to solve.

4.1 Parameters for reinforced shell structure

Our input is an initial shell \( M \) of thickness \( h_x \geq 0 \) (constant per triangle) represented by a triangle mesh, with a vector of external forces \( f \) applied to its vertices, and a set of fixed vertices (supports).

We aim to compute a reinforcement structure, added to the initial shell, which we call a perforated shell of variable thickness \( M^p \) (Figure 6). \( M^p \) consists of a partition \( P \) of the input surface into polygonal faces (typically quads), corresponding to 3D cells, and an extruded shape for each cell, consisting of blocks as described below. The blocks for sequences of cells may form rib-like structures, if the blocks are tall and narrow, or can fill the cells completely which corresponds to the variable-thickness solid shell case.

Given an approximate user target for cell size, our goal is to optimize the edge orientation of the cell boundaries, and the thicknesses and widths of blocks forming each cell to minimize the weight while maintaining an upper bound on stresses (calculated using a beam model). \( M^p \) can be viewed as the reinforcement structure for \( M \).

To simplify our problem, we split all polygonal cells into triangular subcells. We refer to the additional edges inserted in this way as diagonals. We treat these in a special way in the optimization and, in the end, ensure that the triangular subcells can be merged back into the original polygonal cell (Section 7).

4.2 Elastic deformation discretization

We model the perforated shell structure as a beam network: for each interior edge, there are two beams, corresponding to the blocks of incident cells along the edge.

**Notation.** The beam network consists of a set of beams \( E \) that are joined together at nodes (Figure 8). For a node \( i \), \( N(i) \) is the set of indices of nodes connected to it, and the vector \( e_{ij} \) connecting nodes

![Fig. 7. Geometric parametrization of triangular cell. Left: perspective view of the three blocks; Center: top view showing widths; Right: triangle sides \( l_i \) and respective triangle heights \( a_i \).](image)

\begin{align*}
V(y, h) &= A \left( (2 - y_1) y_1 h_1 + (2 - y_2) y_2 h_2 \right. \\
&\quad \left. + (2 - y_3) y_3 h_3 \right) \\
&= A \left( (2 - y_1) y_1 h_1 + (2 - y_2) y_2 h_2 \right. \\
&\quad \left. + (2 - y_3) y_3 h_3 \right).
\end{align*}

This expression can be written as \( V(y, h) = \max_{(i, j, k)} V_{ijk}(y, h) \), where \( V_{ijk}(y, h) \) is given by (1) for arbitrary permutation \((i, j, k)\) of \((1, 2, 3)\). This expression for the volume is useful for the optimization method in Section 7.

The out-of-plane heights \( h_i \) can be constrained not to exceed a user-defined value \( h_{\text{max}} \), and the normalized widths \( y_i \) are constrained so that the trapezoidal areas do not overlap:

\begin{align*}
y_1 + y_2 + y_3 &\leq 1, \quad y_i \geq 0, \quad 0 \leq h_i \leq h_{\text{max}}, \quad \text{for } i = 1, 2, 3.
\end{align*}
\[\varepsilon^m[ij] = (u_i - u_j) \cdot \hat{e}_{ij} / l_{ij},\]

where \(\hat{e}_i\) and \(\hat{e}_j\) are the normals meeting at nodes \(i\) and \(j\), and \(\Delta n\) denotes a linearized change of the normal. The normal change, in turn, is expressed in terms of the displacements \(u_\ell, \ell \in N(i)\) of the incident vertices.

This leads to the expression for a scalar bending strain on beams:

\[\varepsilon^b[ij] = (D^b_{ij} u)^T u\]  

with the expressions for \(D^b_{ij}\) given in Appendix D.

At a distance \(z\) from the middle surface of a beam, the strain is given by \(e^m + z e^b\), where we omit the beam index. Based on the standard Bernoulli beam assumptions, after integration over \(z \in [-h/2, h/2]\) the total beam energy can be expressed as

\[\frac{1}{2}(w h u)^T (D^m)^T u + \frac{1}{6} w h^3 l u^T (D^b)^T (D^b)^T u,\]

which leads to the following expression for the stiffness matrix of the beam system:

\[K^B = wh (D^m)^T (D^m) + \frac{1}{6} wh^3 l (D^b)^T (D^b)^T.\]  

Given the expression for strain \(e^m + z e^b\), clearly both strain and stress are maximal on one of the surfaces, i.e., for \(z = h\) for a given beam. This leads to the following stress constraint:

\[|(D^m_{ij} + h_{ij} D^b_{ij})^T u| < \sigma_0.\]  

The stiffness of the reinforced system combines the stiffness of the beams and shell. We form the shell stiffness matrix \(K^S\) using the elastic discretization described in Section 6. The combined stiffness matrix \(K\) is given by \(K = K^B + K^S\).

Optimization problem. We use index \(c\) for triangular cells. Let \(w\) be the vector of width parameters of all cells, \(y\) the vector of corresponding normalized widths \(w_i/\alpha_i\), \(h\) the vector of all thickness parameters, \(H\) the diagonal matrix with thicknesses on the diagonal, \(u\) the displacements, \(f\) the forces, and \(1\) be the vector of all ones.

We now formulate our optimization problem:

\[\min_{h, y, b} \sum_{c \in cells} V(h_c, y_c) \quad s.t. \quad K(h, y, \mathcal{P})u = f,\]

\[|(D^m + D^b H)u| \leq \sigma_0, \quad 0 \leq h_c \leq h_{\max}, \quad y^c_i \geq 0,\]

\[y^c_1 + y^c_2 + y^c_3 \leq 1, \quad \text{for all cells } c, i = 1, 2, 3,\]  

where the absolute value in the stress constraint is taken element-wise, and the minimum is taken over all partitions \(\mathcal{P}\) of \(M\). The cell volume \(V\) is defined in (1), the stress constraint comes from (6), and the last three constraints correspond to (2). Additionally, we enforce the same thickness on two sides of all cell diagonals.

As optimization over all possible partitions into cells is an intractable problem, combining combinatorial and continuous aspects, we use an heuristic to decide the partition first using beam continuum approximation (Section 6). Once \(\mathcal{P}\) is fixed, we optimize with respect to \(w\) and \(h\) only.

While the above formula volume differs only by a seemingly simple quadratic term from the simplest approximation, this completely changes the behavior of the problems, and, in particular, the behavior of the solvers. The problem no longer reduces to convex by a change of variables as it is the case for the simplest formula (cf., e.g., Hemp [1973]), and a different type of solvers need to be applied. In our experiments, commonly used general purpose non-convex solvers converge very slowly and often fail to make progress. Our solution is described in Section 7.

5 OVERVIEW OF THE APPROACH

Our pipeline for solving the optimization problem consists of the steps listed below.

1. Field optimization. Compute a per-triangle cross-field on surface using stress-based optimization (Section 6). This field corresponds to an idealized system of densely spaced thin beams (beam continuum) with directions chosen to minimize weight. The problem is formulated in terms of displacements and the desired cross-field is obtained from the symmetric strain tensors. This requires solving a convex optimization problem with inequality constraints.

2. Quadrangulation. Create a quad-domain mesh aligned to this cross-field with a user-controlled spacing of edges. This step is done using a version of mixed-integer quadrangulation [Bommes et al. 2009], although any quad layout method with field alignment can be used instead (Section 7). The faces of the mesh will correspond to the cells of the perforated shell \(MP\).
(3) **Cell optimization.** Optimize shape parameters of the cells (Section 7). We introduce a substructure for each cell, with a small number of control parameters defining its shape (widths \( w \) and thicknesses \( h \) of rectangular beams along each side). We derive the optimal material distribution by efficiently solving a nonlinear, non-convex problem minimizing the total volume of all cells with respect to \( w \) and \( h \), while keeping stresses below a user-defined maximum. To make the problem tractable, we defined an efficient local-global optimization method.

(4) **Final geometry construction.** Finally, we derive the final geometry of \( M_p \) according to optimized widths and thicknesses. We obtain a triangle mesh by performing a sequence of boolean operations between meshes representing the beams. The final watertight and manifold mesh can be directly used for 3D printing or decomposed into elements for FEM analysis.

In the following sections, we describe the details of the steps, in the order in which they are applied to produce the final result; however, the key step is the third one (cell optimization), as it has the most impact on the optimality of the result, as demonstrated in the evaluation (Section 8).

6 **WEIGHT-MINIMIZING FIELD OPTIMIZATION**

In this section, we describe our method for constructing a field of directions on the surface which approximates optimal directions for weight minimization with bounded stress. The cells in our construction will be aligned with these directions.

The key idea is to solve a version of the beam weight minimization in the limit case. We assume that there is a continuum of infinitely thin and infinitely close beams in two orthogonal directions forming the surface. The density of the beams and their orientations are the optimization variables. The idea of using this type of continua goes back to Michell [1904]; in contrast to the standard Michell continua formulations, which takes only tension into account, we use both tension and bending. We describe first the classical theory of Michell continua, leading to a convex problem, and then extend it to the case of shells with bending forces, preserving convexity.

6.1 **Michell continua**

Here, we briefly review the classical solution, following Strang and Kohn [1983]. The best directions are known to be the principal stress directions on the original input shell, although these fields are optimized [Kohn 1983]. The best directions are known to be the principal stress directions on the original input shell, although these fields are optimized [Kohn 1983]. The best directions are known to be the principal stress directions on the original input shell, although these fields are optimized [Kohn 1983]. The best directions are known to be the principal stress directions on the original input shell, although these fields are optimized [Kohn 1983].

The force balance for a plate or a shell with no bending is given by the standard equations in terms of in-plane stress tensor \( \sigma \), strain \( \varepsilon \), possibly varying elasticity tensor \( E(p) \), and external force density \( f \):

\[
\text{div} \sigma = f, \quad \sigma = E(p) : \varepsilon; \quad \varepsilon = \frac{1}{2} (\nabla u + \nabla^T u),
\]

where \( A : B = \sum A_{ijkl} B_{kl} \) for a 4-tensor and a 2-tensor.

A Michell continuum is an idealization of a beam network, characterized, at every point, by beam densities \( p_1 \) and \( p_2 \) in two directions (Figure 9). In other words, how many beams cross a unit-length segment along one of the coordinate directions. In the limit of small thicknesses, the total fraction of a small area covered by trusses at a point \( p \) is \( p_1(p) + p_2(p) \). The total volume of the trusses in the continuum can be defined as \( \int_{\Omega} p_1 + p_2 dA \). Note that this approximation of area covered by trusses suffers from the same flaw pointed out in Section 3.

\[
\begin{align*}
\text{Fig. 9. Michell continua example. In the limit of small thicknesses } d, \text{ every point is characterized by two orthogonal beam densities, } p_1 \text{ and } p_2. \\
\end{align*}
\]

The optimal trusses have to be oriented along stress directions, and be critically stressed, i.e., all (non-averaged) stresses on the trusses are equal to maximal stress \( \sigma_0 \). This leads to the relationship between \( p_1 \) and corresponding averaged principal stress: \( \lambda_i(\sigma) = p_i(\sigma_0), i = 1, 2 \), where \( \lambda_i(\sigma) \) denotes the \( i \)-th singular value.

Then, we obtain the following optimization problem for the volume, formulated entirely in terms of stresses:

\[
\text{minimize } \int_{\Omega} |\lambda_1(\sigma)| + |\lambda_2(\sigma)| dA, \text{ subject to } \text{div} \sigma = f. \quad (9)
\]

This problem is known to be convex [Strang and Kohn 1983] (although it is more difficult than the linear programming formulation for a truss network). Note that principal stress directions are not fixed and are determined by the optimization. We use these directions as the field for orienting beams in \( M_p \).

The problem (9) has a simple dual (Appendix B) of the form

\[
\text{maximize } \int_{\Omega} f^T u dA \text{ subject to } |\lambda_i(\varepsilon)| \leq \varepsilon_0, i = 1, 2, \quad (10)
\]

where \( \varepsilon \) is the strain of the optimal solution (Figure 10). The dual problem is significantly easier to deal with in the case of continua. We note that here we neglect the overlap volume of trusses discussed in Section 3; while it could be included as \( -p_1 p_2 \) term, this would immediately make the problem non-convex, and the benefit of more precise field optimization in terms of volume reduction is minor (Section 8).

\[
\begin{align*}
\text{Fig. 10. Field optimization: } (a) \text{ Design domain } \Omega \text{ and boundary conditions; } (b) \text{ Primal solution, squared densities } q(p_1 + p_2); (c,d) \text{ Dual solutions, displacements } u \text{ and strain } \varepsilon \text{ eigen-values and -vectors.}
\end{align*}
\]
6.2 Continuum optimization with bending

Next we generalize problem (10) to include bending forces.

If the thickness of the shell remains fixed, one can add bending to the functional with relative ease without changing the convexity of the problem. We set the shell thickness in this case to half of the maximal allowable thickness; while the resulting field is suboptimal, as we show experimentally in Section 8, inaccuracy in the beam direction has less effect on the overall weight reduction, compared to width/thickness optimization of beams.

We make the standard assumption of planar stress for the shell, i.e., no stresses are active in the direction perpendicular to the shell surface. The strain at a distance \(z\) from the midline of the shell is given by (Figure 11)

\[
\varepsilon(z) = \varepsilon^m + z\varepsilon^b,
\]

where \(\varepsilon^b\) is the bending strain tensor, equal to the linearization of the change in the shape operator \(\nabla\hat{n}\).

Consistently with the Michell continuum, we seek to minimize the total weight of a beam continuum, bounding the stress everywhere by \(\sigma_0\). We observe that the eigenvalues of \(2 \times 2\) symmetric matrix \(A\), using the substitutions \(a = (A_{11} + A_{22})/2\), \(b = (A_{11} - A_{22})/2\), and \(c = A_{12}\), are of the form \(a \pm \sqrt{b^2 + c^2}\). Note that these are respectively convex and concave functions of the argument, and, therefore, reach their maxima (respectively minima) on the boundary of the shell, for \(z = \pm h/2\). For this reason, it is sufficient to bound eigenvalues of stress (or strain) only for \(z = h/2\) and \(z = -h/2\), to guarantee the bounds elsewhere.

In the case of bending, the dual problem formulated in terms of displacements has a simpler form relative to the primal problem:

\[
\text{maximize} \quad \int_{\Omega} f^T \text{u} \, dA \quad \text{subject to} \quad |\lambda_i(\varepsilon^m \pm \frac{h}{2}\varepsilon^b)| \leq \sigma_0, \quad i = 1, 2, \ldots \quad (11)
\]

Note that now there are two sets of constraints, corresponding to two surfaces of the shell.

The strain tensors \(\varepsilon(h/2)\) and \((-h/2)\) can be interpreted as defining cross-fields on the surface. We use the angular average of these fields to align the cells.

6.3 Field discretization and optimization

Finally, we describe a discretization of the convex problem defined above, and how to use the resulting field to build a mesh.

As a first step, we solve a discrete version of the problem (11), which yields displacements \(\text{u}\) at vertices. From these displacements, we compute the per-triangle strain field eigenvectors, forming a cross-field on the surface, i.e., an assignment of 4 unit vectors, aligned with perpendicular principal strain directions, to each triangle.

\[
\varepsilon^b_T = \sum_{i=1,2,3} \frac{\theta_i}{2A_l} \varepsilon_{ijk}^+ (\varepsilon^b_T)^T
\]

where \(\theta_i\) are linearized changes in the angles between normals of adjacent triangles.

Discretization of the optimization problem. The optimization problem (11) has a relatively simple discretization that can be readily plugged in into a cone program solver, e.g., MOSEK [ApS 2015].

We assume that the surface is given as triangle mesh, \(M = (V, E, F)\), and the same notation is used for edge vectors and vertices as we used for beam networks.

The variables in the problem are displacements, which we discretize using standard piecewise-linear functions on the surface, with the vector of unknowns \(\text{u}\) (we use non-bold letters for high-dimensional vectors including all components of corresponding three-dimensional quantities).

The two quantities that need to be discretized are membrane and bending strains; we define these per triangle.

If \(\varepsilon_{ij}\) are the vectors along triangle edges, for a triangle \(T\), we have the following expression for the strain, computed as \(\frac{1}{2}(\nabla \text{u} + \nabla^T \text{u})\):

\[
\varepsilon^m_T = \frac{1}{4A_T} \sum_{i=1,2,3} \varepsilon_{ijk}^+ \varepsilon_{i}^T + \varepsilon_{j}^T \varepsilon_{jk}^T.
\]
Similarly to $\epsilon^m$, we can write $\epsilon^b = B^b u$. Then the discrete problem takes the form

$$
\text{maximize } f^T u \text{ subject to, for all } T, \{\lambda_i(B^T_i u + h B^b_i u)\} \leq \epsilon_0, i = 1, 2.
$$

where $f$, similarly to the beam case, denotes the vector of per-vertex forces, and $u$ is the vector of all vertex displacements. We use eigenvalue formulas defined in Appendix A for (19) to convert the problem to a convex cone problem, which we solve using the MOSEK solver [ApS 2015].

Detecting field zones. While the output of the previous step defines a tensor for each triangle, not all of these are meaningful. In some cases (if the triangle is not deformed at all, or deformed negligibly) the strain is zero. More generally, some points may have isotropic strains of the form $kl$, where $k$ is a nonzero constant, for which all vectors are eigenvectors, so the cross-field is not defined uniquely on this triangle. For general fields, such points are usually isolated. However, for the fields corresponding to the solution of the problem we are considering, the situation is different. There are four possible regimes (see, e.g., Strang and Kohn [1983]). Specifically, the possibilities include

1. $\lambda_i(\epsilon) = -\lambda_j(\epsilon) = \epsilon_0$, principal strains are critical and have opposite directions; this corresponds to well-defined two orthogonal beam families;
2. $\lambda_i(\epsilon) = \lambda_j(\epsilon) = \pm \epsilon_0$, principal strain are critical and have the same direction; in this case, stresses (which are dual variables to the inequality constraints) are large but beam directions are not well defined;
3. $|\lambda_i(\epsilon)| < |\lambda_j(\epsilon)| = \epsilon_0$, only one strain is critical; this corresponds to a single family of beams.
4. $|\lambda_i(\epsilon)| < \epsilon_0$, in this case, stresses are both zero, which means there are no beams in this area.

Fig. 13. Zones of an optimal strain field. (1) Two orthogonal directions, (2) no preferred direction, (3) one direction, (4) no beams. The crossfield directions are well-defined only for regions 1, 3.

Completing the field. To complete the field on the whole surface, which is needed for a complete structure, we use cross-field constrained optimization procedure of Bommes et al. [2009]. In this algorithm, the cross-field is encoded by a per-triangle angle, with respect to a reference direction $\beta_i$ in each triangle. The angles on salient triangles are fixed. On the remaining triangles, these are found by a greedy solve of a mixed-integer problem minimizing the energy

$$
E = \sum_{\text{edges}(ij)} (\beta_i - \beta_j + k_{ij} \frac{\pi}{2} + \kappa_{ij})^2
$$

where the summation is over all dual edges connecting triangles $i$ and $j$, $\kappa_{ij}$ is the angle between reference directions in triangles, and $k_{ij}$ is an integer unknown accounting for the fact that cross-field values represented by angles $\beta + k\pi/2$ are the same.

In the resulting field, defined by the angles $\beta_i$ for all triangles, and integers $k_{ij}$ for all edges, one can easily compute per-vertex field index and identify field singularities, which become irregular (valence different from 4) vertices of the quad mesh at the next step. We refer to Bommes et al. [2009] for details of the index computation.

6.4 Construction of the quad-dominant network
In general, there may be no optimal beam spacing (in the low-volume limit, the finer the structure, the lower the optimal volume can be for a given stress). For this reason, the beam spacing is defined by a user-specified parameter $H$. The most direct approach for constructing a quad mesh aligned with a field would be to trace it. However, while it was shown [Myles et al. 2014b; Ray and Sokolov 2014] that this approach can be implemented robustly, in general it requires T-joints (i.e. beams joining other beams in the middle), and it is in general hard to ensure uniform spacing over the whole mesh. We choose a more conservative approach based on constructing a conforming quadrangulation, without T-joints, using a version of the mixed-integer quadrangulation algorithm [Bommes et al. 2009] at this step.

While the method does not guarantee perfect alignment of the parametric lines to the input field, it minimizes the deviation in least-squares sense. We refer to Bommes et al. [2009] for further details.

7 OPTIMIZING CELL GEOMETRY
Given an input quad-dominant mesh, split into triangles, we aim at deriving the optimal width and thickness for each edge. As we previously stated, to obtain a structure close to optimal, the edges need to follow the directions we derived in the field optimization step. While our material distribution optimization method works for any mesh, the further the edges deviate from their optimum directions, the greater would be the total weight. The main idea of our algorithm is to use a local-global iteration, solving per-triangle concave problems for each cell, for which the solution is guaranteed to be on the boundary on the constraint domain. This yields a rapidly converging efficient algorithm.

7.1 Optimization algorithm
We introduce a domain-decomposition-style algorithm for solving the problem (7). We observe that in the optimization problem (7),
all constraints except $Ku = f$ are localized, i.e., each constraint uses variables related to one cell. Moreover, the functional itself is a sum of local volume terms $V^c(w^c, h^c)$. $Ku = f$ expresses the equation of force balance, i.e., that the sum of beam forces at each node is equal to the external force at this node. Our approach is to fix individual beam forces to their values for current values of geometry parameters $w$ and $h$, and then solve for an update to $w$ and $h$ as a set of local volume-minimization problems with variables $w^c, h^c$, replacing the global constraint $Ku = f$ with local constraints requiring that individual beam forces remain the same.

We start with an outline and then elaborate on how the local step optimization problems are solved.

Initially, we assign sufficiently high values to $w$ and $h$, to ensure that max stress constraints are satisfied.

- **Global step.** The global step is just the standard solution of the elastic equilibrium problem, for fixed cell parameters: Solve $Ku = f$, for fixed $K$ defined by $w$ and $h$. Compute beam forces as described above.
- **Local step.** The local step is the key part of our algorithm. Recall that an important feature of optimal structures is that their magnitude is equal to $\sigma m + h a b$. This leads to the critical stress constraint $\sigma m + ha b = \sigma 0$. Using expressions for $g m$ and $g h$ above, which we keep fixed at the local step, this is equivalent to

$$g m = |f m|/l = g m, \quad \frac{1}{6} \sqrt{\gamma a b} = |g h|/l = g h.$$  

(15)

**Local optimization problem.** The stress in the block, under our assumptions, reaches its maximal value at the top or bottom, where its magnitude is equal to $\sigma m + ha b$. This leads to the critical stress constraint $\sigma m + ha b = \sigma 0$. Using expressions for $g m$ and $g h$ above, which we keep fixed at the local step, this is equivalent to

$$g m + \frac{6}{h} a b = \sigma 0 a y,$$

where we have switched to the variable $y = w/a$ introduced in (7), where $a$ is the corresponding cell triangle height. Without loss of generality, we assume $a y = 1$, which can be achieved by scaling all forces. The complete local problem in variables $h^c, w^c$, $i = 1, 2, 3$ is:

$$\min_{y^i, h^c} V(y^c, h^c) \quad \text{s.t.} \quad \frac{g b}{h} + \frac{6}{h} a b = \sigma 0 a y_i$$

(16)

$$\quad 0 \leq h^c \leq h_{\text{max}}, \quad y^e_i \geq 0, \quad \text{for } i = 1, 2, 3$$

$$\quad y^e_1 + y^e_2 + y^e_3 \leq 1,$$

where the volume $V$ is given by (1), and the last three constraints by (2).

By eliminating the stresses, we arrive at a single constraint per block relating $w$ and $h$, which we express as follows:

$$y = (6g h z^2 + g a z)/a.$$  

(17)

where $z = h^{-1}$ is a new variable we introduce to simplify the expressions. This allows us to eliminate all variables $w^c$ from the local optimization problem, leaving only three variables $z^e$, with constraints $0 \leq z_{\text{min}} \leq z^e_i, i = 1, 2, 3$, and $z_{\text{min}} = (h_{\text{max}})^{-1}$.

We say that a cell is filled if the equality $y^e_1 + y^e_2 + y^e_3 = 1$ is satisfied, i.e., the blocks completely fill the cell.

Without loss of generality, we assume that for the solution $z_1 \leq z_2 \leq z_3$; in practice, six problems corresponding to six permutations of $(1, 2, 3)$ need to be solved and minimal solution picked.

**Proposition 1.** The function $V(z^c)$ is a concave function of $z_1$. As a consequence, its minima are reached on the boundary of the constraint domain, specifically, it is reached at one of the five types of configurations:

(1) all three blocks have maximal thickness: $z^e_i = z_{\text{min}}, i = 1, 2, 3$;
(2) the cell is filled, i.e. $y^e_1 + y^e_2 + y^e_3 = 1$, and no inequality constraint reaches equality;
(3) the cell is filled, and two thicker beams have equal thickness, $z^e_1 = z^e_2$;
(4) the cell is filled, and two thinner beams have equal thickness, $z^e_2 = z^e_3$;
(5) the cell is filled, and the thickest beam has maximal thickness $z^e_1 = z_{\text{min}}$.

In the first case, the solution is completely determined. In the second case, there are four possibilities: no inequality constraint is active (a 2-variable unconstrained optimization problem needs to be solved, e.g., parametrized by $z^e_1, z^e_2$); the other three cases
define one-parametric families of solutions, and one-dimensional unconstrained optimization needs to be performed to find exact values, as we explain below. These families can be parametrized by, e.g., \( z_i^c = (b z_i^c)^{-1} \), with the values of the remaining \( z_i^c \) and \( y_i^c \) determined from the active constraints. The proposition is proved in Appendix C.

This behavior of \( V \) is in stark contrast to the low-volume formulation ignoring common areas of beam-like parts of the structure: one can see that in three cases out of four, it creates a completely filled cell.

**Solving the optimization problem.** Proposition 1 leads to an efficient algorithm for the local step.

Observe that the constraint \( y_i^c + y_i^c + y_i^c = 1 \) has the form

\[
\sum_i g_i^m z_i^c + 6 g_i^b (z_i^c)^2 = 1,
\]

i.e., it is quadratic in \( z_i \). This allows us to reduce the problem to a set of unconstrained optimization problems in one or two variables.

1. Compute \( g_i^b \), \( g_i^m \), \( i = 1, 2, 3 \), from (15) for current displacements.
2. Evaluate \( V(z) \), for the case 1 solution with \( z_i^c = z_{min} \).
3. For each permutation of (1, 2, 3) solve three one-dimensional optimization problems, minimizing \( V(z^c) \), for each of the cases 3-5, and the two-dimensional problem for case 2 of proposition Prop 1. In each case 3-5, substitute the active constraint for \( z_i^c \) into (18), yielding a quadratic equation in two remaining free variables, one of which is \( z_i^c \). Solve it to express the other variable in terms of \( z_i^c \), and solve a one-dimensional optimization problem for \( V(z_i^c) \). This yields a set of solution candidates; the minimal solution is guaranteed to belong to it.
4. Pick the minimal solution from the set of solutions obtained for all possible permutations and cases on the previous step.
5. Update \( w_i^c \) using formula \( w_i^c = (6 g_i^b z_i^c + g_i^m z_i^c) \), and recompute the global stiffness matrix \( K \).

The convergence behavior of the method is considered in Section 8.

**Handling polygonal cells and postprocessing.** There are three factors not considered in the solution method above: (1) possible inconsistency of thickness values across diagonal edges inside triangulated polygonal cells; (2) the coherence of block widths and thicknesses along the edge lines of the quad mesh, approximating the optimized stress lines (Figure 14). While jumps in thickness/width along these lines do not affect the stresses in our simplified model, in practice, these are likely to lead to localized stress concentrations close to jumps, and they are aesthetically objectionable. (3) Stress values may slightly exceed the maximal stress after a final global step.

We experimentally observe that many of the candidate solutions have close values, especially in areas with no predominant stress direction.

We address (1) primarily in the process of optimization, at step 4, we pick a minimal candidate solution with lowest block thickness on the diagonal, which may not be the most optimal one, as long as it does not deviate above a threshold. Once the optimization is complete, for each subcell of a polygonal cell we increase the lowest block thickness to the maximal minimal thickness over the whole cell. In addition, for each cell, we store a number of candidate solutions with the smallest volumes.

We address (2) in a post-processing step using stored candidate solutions: for each non-diagonal edge of a cell, we find its continuation edges along quad mesh edge line in both directions, and choose the candidate solution closest in width to the average of the previous and next edge widths along the edge line.

To address (3), we find all blocks with stress exceeding \( c_0 = 1 \), and increase their thickness and width, while maintaining constraints, to decrease the stress to the bound. This process is repeated iteratively until convergence. We note that all additional steps are designed to ensure that the final result satisfies stress constraints: in all cases, we never decrease the amount of material in cells, so while the resulting solution may be suboptimal, it always satisfies stress constraints. In practice, the effect of these alterations on the resulting weight is small.

Alternatively, to alleviate these issues, one could optimize for the width and thickness per-vertex instead of per-edge. To be more precise, we would require two widths/thicknesses per-vertex since two beams crossing at a vertex may be different. While this approach is, in principle, possible, in practice is much more expensive: for vertex-based cells, the volume function is not likely to be concave, so a general nonconvex solver needs to be used. Our experiments showed that, as the number of variables and constraints increases, a general solver is likely to get stuck in local minima (see Section 8 Figure 21).

**Inflation.** Once we have performed the weight optimization, we have one value of thickness and one value of width for each half-edge of the optimized polygonal mesh. If we simply extrude the solid block that matches thickness and width for each half edge cell, we end up in the situation illustrated in Figure 14.a, where there are visible discontinuities between adjacent blocks, causing possible structural discontinuities. Instead, for each continuous stream of quad edges, we generate a unique solid block that interpolates thickness and width along its length (Fig. 14.b). In detail, given a sequence of aligned edges, we first derive a thickness value for each vertex by averaging the thickness from its adjacent half edges. Similarly, we interpolate widths, but this time we derive two different values for each vertex, one for each side of the sequence. We then define a tangent vector for each vertex as the cross product between its...
normal and its direction along the edge sequence (obtained by aver-
aging the direction of the two incident edges). Then, having defined
a proper reference frame, a thickness, and a width for each vertex,
we have all the information to extrude a proper volumetric block.
We perform a boolean operation to merge all the blocks together
to a manifold watertight mesh using the approach of Zhou et al.
[2016].

8 EVALUATION

Topology optimization. To validate our approach vs. a general-
purpose topology optimization method, we solve a similar problem
with topology optimization code [Aage et al. 2015] by restricting
the volume of the material to a cylinder of fixed small thickness h,
and choosing the volume grid resolution to be half of the cylinder
thickness, leaving little room for shell shape variation: with this
grid resolution, Aage et al. [2015] can only generate beams (or walls)
of thickness h and h/2. To make the comparison fair, we configure
our optimization to have a minimum thickness close to the maxi-
mum thickness (near-constant thickness). Default parameters were
used in Aage et al. [2015]. We observe that for small target volume
fractions, as expected, Aage et al. [2015] generates substructures
similar to the beam structures we construct (Figure 15). For large
volume fractions, both methods result in variable-thickness shells.
However, due to the limitations on the thickness mentioned above
(large minimum thickness), the thickness variation is minimal. We
note that topology optimization requires high resolution (with mul-
tiple cells fitting in the thickness direction of the shell) to achieve
more variation in shell-thickness.

Figure 16 shows the results of SIMP topology optimization vs.
our method, with compliance as a function of the volume fraction.
While our algorithm does not have a target volume fraction, we
try to match these using different load magnitudes. To measure
compliance for our results, we use HyperWorks FEA analysis, with
a unit load. For SIMP, we use the code provided by Andreassen et al.
[2011] with a square grid of 200 × 200 pixels, and the compliance re-
ported by this software. The main reason not to use HyperWorks for
SIMP is that it would require triangulating the result and smoothing
its sharp corners, so the result will be inaccurate. Instead, we veri-
fied that the compliance reported by both systems were the same
for various shapes and load cases. In the plot, we observe similar
behaviors for both methods. We note that in our case we need to
choose the beam spacing parameter: when this parameter is chosen
to be too coarse, the performance deteriorates.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure16}
\caption{Volume fraction vs. compliance energy for a standard example, a
double truss, SIMP topology optimization vs. our method.}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure17}
\caption{Comparison of volumes obtained using different fields for the shell
structure. Images (a-d) show the quad-mesh obtained with the different
fields: (a) optimized stress directions from solving (14), (b) stress direc-
tions from solving elasticity, (c) MIQ field constructed using the smoothing
method of Bommes et al. [2009], and (d) constructed rotating (a) by 45
degrees.}
\end{figure}

The role of beam direction. In Figure 17, we explore the depen-
dence of the role of beam direction in structure optimality, by com-
paring structure volume for several fields, in addition to our opti-
mized field (Eq. 14). As a “worst-case” baseline we use the cross-field
at a maximal distance from the original field (d); as one can see,
the field makes a significant difference when it is very far away
from the optimal directions, i.e., the choice of directions matters.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure15}
\caption{a) cylinder beam structure obtained using topology optimization
with a target volume of 0.375. The pixelated effect is caused by the voxel
grid; b) structure obtained using our method with a maximum thickness of
0.05, minimum thickness of 0.04, no support shell ($h_s = 0.00$).}
\end{figure}
Reinforcement of General Shell Structures

To evaluate the effect of the choice of diagonals in conversion from a quad mesh to a triangle mesh, we compare the optimized volumes for the shorter and longer diagonal choices. For the pipe example (Figure 18-left), the volumes are equal to 14.6% and 16.9% of the original volume, respectively. Choosing shorter diagonals (our default) results in better-shaped triangles, which we have observed to require less material to keep the stress low. If instead of flipping all diagonals, we flip only interior diagonals, using the shortest-diagonal on the ring near the boundary, we obtain a volume of 15.9% (Figure 18-right).

Convergence and dependence on the starting point. We have observed that our algorithm almost invariably converges in several iterations, and yields the expected behavior of solutions in the extreme cases (ribs in the case of bending-dominant shells, and wide-and-thin beams in high-tension/low-bending areas). The plots in Figure 19 show the volumes at each iteration for a basic and more complex problem. Note that sometimes the optimal value is approached from below: the local step overshoots the volume reduction and the stress exceeds the maximal allowed level. Nevertheless, the method recovers reliably.

Comparison to other optimization methods. We also compare our method to two general-purpose constrained optimization methods, SLSQP [Kraft 1988] and Ipopt, a barrier interior point method [Byrd et al. 2000]. In this setup, we used an approximation of the beams volume that is smoother (it uses average instead of max) and simple lower and upper bounds on the width and thickness. As a starting point, we have used the solution of the convex problem with volume ignoring the overlaps. Somewhat surprisingly, these methods were not able to change this initial solution by much after a few hundred iterations; although moving in the right direction, in terms of values, it may differ by a factor up to two from the optimal solution.

While the SLSQP solutions exhibited oscillatory behavior, alternating decreasing the functional with decreasing constraints violations, the interior point method solutions mostly stayed close to the initial values. Figure 21 shows comparative results with Ipopt for a small cantilever test case. We observe that Ipopt converges to a volume almost an order of magnitude larger. When Ipopt is initialized with the solution of the convex problem with simplified

---

Fig. 18. Effects of the choice of diagonal directions: on the left, the original diagonals computed by our method, choosing the shorter diagonal; on the right, the diagonals of the first boundary ring are preserved and internal ones are flipped.

Similarly, the curvature field (c), unrelated to stress directions, produces relatively high values of the volume. On the other hand, the difference between the optimized field and the stress field (a and b), while present, is not large in most cases. This experimentally justifies using, e.g., Li et al. [2017], the elasticity stress field instead of the optimized field; however, the additional expense of using an optimized field is minimal, so there is effectively no penalty for this improvement.

The plots on Figure 20 show the volume and stress histogram vs. iteration count for a complex model. Using our default initialization, all stresses start with values below $\sigma_0$, by construction, and during the optimization they converge to values near $\sigma_0$. As mentioned on Section 7.1, during the optimization some stress values may exceed $\sigma_0$. These are resolved during the post-process (after iteration 80 for this example), without significantly increasing the total volume.

Fig. 19. Left: Volume convergence for our method, for the boat model, using different starting points. Right: similar plots for the bending plate.

Fig. 20. Convergence plots for the Pipe model (figure 29d). On the left, the volume vs. iteration count; On the right, the evolution of the histogram of the stress where volume percentage is measured with respect to the initial volume. At iteration 81, we start the post-process that ensure stresses are below $\sigma_0 = 1$.

Fig. 21. Volume vs. iteration step for a small cantilever example (408 beams) using Ipopt vs. our method.

While the SLSQP solutions exhibited oscillatory behavior, alternating decreasing the functional with decreasing constraints violations, the interior point method solutions mostly stayed close to the initial values. Figure 21 shows comparative results with Ipopt for a small cantilever test case. We observe that Ipopt converges to a volume almost an order of magnitude larger. When Ipopt is initialized with the solution of the convex problem with simplified
volume functional (green), the optimization fails to find a better solution. Using a different initialization (orange: maximum width and thickness) does not provide better results. In contrast, our method converges to a much better solution in only a few iterations.

Effects of constraints on structure parameters. Figure 22 shows the effect of increasing maximal allowable thickness in a bending scenario, with the structure moving from fully solid, to a structure of narrow but tall beams (if the bound is increased to infinity, in principle, any bending force can be realized by a zero-volume infinitely thin rib).

Similarly, Figure 23 shows the effect of increasing minimal thickness in a tensile/compression scenario. In this case, tall beams appear when increasing the lower bound because widths decrease to compensate for the increase of thickness. The usage of beams in tensile/compression scenarios, while common, is often a consequence of constraints on minimum element size and are sub-optimal in the absence of these constraints. On the torsion cylinder experiment, the effect of using a minimal thickness of 0.04 vs. 0.00 increases the volume by 60%.

Comparisons to related work. A direct apples-to-apples comparison of different methods for optimization for shells is extremely difficult for a number of reasons, most importantly, because of different models used (e.g., all approaches use a variation of a beam model, and the specific choice has a significant impact on stress estimation). Another important reason includes a strong dependence of the results on the maximal width/thickness/aspect ratio constraints. We summarize qualitative differences from the most closely related works Kilian et al. [2017] and Li et al. [2017].

The key difference from Kilian et al. [2017] is that it changes the shape of the surface, focuses on the case when bending forces can be neglected, and uses the narrow-beam volume approximation. Kilian et al. [2017] approach might be appropriate for the targeted type of architectural applications, combining surface shape design with structural constraints, but in many cases the surface cannot be modified arbitrarily. Similarly, form-finding methods for structural shape optimization [Veenendaal and Block 2012] apply a minimal modification to the input shape such that the final form can transfer their loads purely through axial or in-plane forces. In the optimal case, no out-of-plane stress is present in the final shape, which may require a significant change to the surface: e.g., for gravity loads, resulting shapes need to have a very specific form for bending loads to be eliminated. For most examples we consider, decreasing out-of-plane forces, for given loads, without a drastic change on the input shape is impossible: e.g., it would be impossible to keep the surface of the shelf or the bottom of a boat flat.

In comparison, we deal with a shell of fixed surface shape, take bending forces into account and use a more precise volume approximation, which leads to a non-convex problem.

The method of Li et al. [2017] is closer to ours: they construct a quadrangulation following a field to determine beam directions, and also take bending into account in the elasticity model. However, rather than using an optimized stress field as we do, they use the stress field of the non-optimized shell. Although the fields are typically close, in some cases, the optimized field provides an advantage (Figure 17).

More importantly, Li et al. [2017] uses a narrow-beam approximation for the volume which has a major effect on the efficiency of the reinforced structure. To measure this effect, we compare the results of our optimization using different volume formulas as functional (Figure 24). The results show that using the simpler volume approximation can increase the volume used by up to 44%.

As explained on Section 3, the convex volume approximation biases the solution against using variable-thickness shells. To demonstrate the effects of this, we compare the stiffness of two reinforcements, obtained using different methods, with the same material cost. The shell and rib-reinforcement structure (Figure 25b) where provided by the authors of Li et al. [2017], while the second type of reinforcement (Figure 25a) was generated with our algorithm.

To match the setup of Li et al. [2017], we set a minimum width for the beams. In contrast to that work, for similar loads, our optimization results in a lower-weight variable-thickness wall without protruding ribs, due to our use of the exact volume formulation (Section 3).

To compare the results fairly, with no dependence on the beam model used, we post-process both results using TetWild meshing software [Hu et al. 2018] to create tetrahedral meshes of similar
density, and then use Finite Element Analysis to solve for the displacements with the loads and supports used in optimization (we use the PolyFem implementation [Schneider et al. 2019]). We impose zero-displacements on the nodes at floor level, a uniform vertical load on the rest of the structure, a Young’s modulus $E = 210000$ and Poisson’s ratio $\nu = 0.3$. The resulting displacements are shown on Figure 25c, d. Our reinforced structure shows a maximum displacement of 67.31, while the rib structure of Li et al. [2017] shows a higher maximum of 97.16. Thus, under the same boundary conditions, our result achieves 1.44 times higher stiffness compared to a rib-reinforced structure of same weight.

This comparison does not include the I-beam profile optimization of beam cross-sections introduced by Li et al. [2017] as I-beam shapes were not included in the geometry provided by the authors and relevant code is no longer available. However, in the case of dominant tensile/compression loads along the beam, this technique is not likely to yield significant improvement, as these forces depend primarily on the cross-section area, not its shape.

A minor difference with Li et al. [2017] concerns the smoothing of the final result. The approach of Li et al. [2017] is to add an additional smoothing term in the function that optimizes the material distribution. This term smooths the volume along sequences of consecutive aligned edges, classified into curved, circular, and tree-like rib elements. Our smoothing procedure is a part of the post-optimization geometric process along consecutive aligned edges, that is easier to control (see Figure 14.b). As a future improvement, we plan to include a global smoothing term in the energy formulation.

**Physical Experiments.** We have printed several simple structures to validate our optimization experimentally (Figures 26, 27, 28). Due to the highly approximate nature of the physical model used, we did not attempt a quantitative match to the simulated values, but we did closely match the values of the printed models. The comparison in all cases is between an optimized model and a uniform-thickness model of the same weight. Observed displacements in all cases differed by a factor more than 3, suggesting a similar difference in stress.

The optimization of the shelf was done using distributed loads (Figure 30, last row). However, when the shelf bends, the physical load is applied in only two separate regions. Using properly distributed loads would, most likely, only increase the difference in performance, as the structure was optimized for this case.

Finally, Figures 29 and 30 show a set of optimized shells obtained for a variety of shapes using our method; in all cases we have preserved a minimal width/thickness beam to indicate the mesh edges, but the load is carried by a relatively small number of beams; we found that ribs tend to appear, even in tension areas, unless the thickness bound is set to very low values. This is consistent with the observation that with no thickness bound, the load carried by bending forces is maximized, if the goal is to reduce the volume.
Table 1. Statistics for the 3D models in Figure 30 and Figure 1: the bounding box diagonal $d$ (mm), the number of triangles $|F|$ in the input mesh, the maximum thickness $h$ (mm) of the beams, the constant shell thickness $h^s$ (mm), the number of iterations $k$ and step-size $s$ of the optimization solve, and the time $t_{solve}$ it took to complete all iterations.

|  | $d$ (mm) | $|F|$ | $h$ | $h^s$ | $k$ | $s$ | $t_{solve}$ (s) |
|---|---|---|---|---|---|---|---|
| Aquadom | 85.60 | 7742 | 0.5 | 0.005 | 100 | 0.2 | 72.330 |
| Basket | 127.56 | 10943 | 2.0 | 0.06 | 30 | 0.5 | 72.513 |
| Beetle | 169.01 | 7558 | 1.2 | 0.012 | 50 | 0.5 | 35.999 |
| Boat | 92.60 | 9084 | 1.0 | 0.05 | 100 | 0.2 | 33.067 |
| Botanic | 42.59 | 2152 | 0.5 | 0.01 | 100 | 0.5 | 35.586 |
| Bowl | 117.77 | 4779 | 2.0 | 0.0 | 32 | 0.2 | 12.354 |
| Bucket | 150.96 | 7860 | 2.1 | 0.21 | 50 | 0.5 | 45.744 |
| Bunny | 106.00 | 7471 | 2.0 | 0.9 | 100 | 0.25 | 22.291 |
| Duct | 460.58 | 6932 | 1.0 | 0.01 | 100 | 0.2 | 69.593 |
| Leaf | 99.24 | 2980 | 2.0 | 0.9 | 46 | 0.5 | 1.954 |
| Neumunster | 111.92 | 6072 | 2.0 | 1.2 | 63 | 0.2 | 6.959 |
| Pipe | 159.48 | 10409 | 2.0 | 0.0 | 82 | 0.2 | 80.226 |
| Shelf | 88.54 | 2400 | 4.0 | 0.5 | 60 | 0.5 | 1.663 |
| Spoon | 133.11 | 6528 | 4.0 | 0.9 | 29 | 0.5 | 4.129 |
| Stevia | 90.68 | 4840 | 2.0 | 0.9 | 50 | 0.5 | 23.941 |
| Vase | 69.64 | 4798 | 1.0 | 0.45 | 100 | 0.5 | 21.817 |

Table 2. Performance for the 3D models in Figure 30. We show the ratio $c_{eq}/c$ between the compliance $c$ of our results and the compliance $c_{eq}$ of a shell of constant thickness of the same weight. The fixed shell volume $V^s$, the support structure volume $V^b$, the thickness $h_{eq}$ of the equivalent-weight constant thickness shell.

<table>
<thead>
<tr>
<th></th>
<th>$V^s$ (mm$^3$)</th>
<th>$V^b$ (mm$^3$)</th>
<th>$h_{eq}$ (mm)</th>
<th>$c_{eq}/c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aquadom</td>
<td>16.989</td>
<td>90.335</td>
<td>0.032</td>
<td>1.601</td>
</tr>
<tr>
<td>Basket</td>
<td>1207.584</td>
<td>3392.022</td>
<td>0.229</td>
<td>7.983</td>
</tr>
<tr>
<td>Beetle</td>
<td>170.128</td>
<td>1287.227</td>
<td>0.103</td>
<td>5.739</td>
</tr>
<tr>
<td>Boat</td>
<td>189.997</td>
<td>418.539</td>
<td>0.160</td>
<td>7.099</td>
</tr>
<tr>
<td>Botanic</td>
<td>9.134</td>
<td>80.533</td>
<td>0.098</td>
<td>1.266</td>
</tr>
<tr>
<td>Bowl</td>
<td>0.000</td>
<td>4083.710</td>
<td>0.410</td>
<td>4.437</td>
</tr>
<tr>
<td>Bucket</td>
<td>105.195</td>
<td>58.274</td>
<td>0.326</td>
<td>26.924</td>
</tr>
<tr>
<td>Bunny</td>
<td>7270.428</td>
<td>1134.806</td>
<td>1.040</td>
<td>1.229</td>
</tr>
<tr>
<td>Duct</td>
<td>1075.645</td>
<td>9777.817</td>
<td>0.194</td>
<td>2.105</td>
</tr>
<tr>
<td>Leaf</td>
<td>2304.430</td>
<td>659.348</td>
<td>1.158</td>
<td>11.05</td>
</tr>
<tr>
<td>Neumunster</td>
<td>44378.721</td>
<td>3902.169</td>
<td>9.139</td>
<td>1.076</td>
</tr>
<tr>
<td>Pipe</td>
<td>0.000</td>
<td>45833.029</td>
<td>2.551</td>
<td>1.161</td>
</tr>
<tr>
<td>Shelf</td>
<td>1176.000</td>
<td>1386.550</td>
<td>1.090</td>
<td>7.252</td>
</tr>
<tr>
<td>Spoon</td>
<td>1544.960</td>
<td>326.733</td>
<td>0.226</td>
<td>18.621</td>
</tr>
<tr>
<td>Stevia</td>
<td>8329.664</td>
<td>492.212</td>
<td>0.103</td>
<td>7.099</td>
</tr>
<tr>
<td>Vase</td>
<td>90.68</td>
<td>7558</td>
<td>0.5</td>
<td>0.012</td>
</tr>
</tbody>
</table>

9 CONCLUSIONS AND FUTURE WORK

In this paper, we have described a way to approximate and efficiently solve the problem of minimizing the weight of a support structure for a shell. Our work focuses on choosing a simple, yet sufficiently expressive, geometric description of the support structure, and an optimization algorithm capable of optimizing it. The proposed method separates the construction into three stages, with the first stage optimizing the field the beam directions must follow and creating a corresponding quad-dominant mesh, the second stage creating a cell structure with optimized shape parameters, and the third stage creating an actual realization.

This makes our approach particularly flexible and allows us to integrate a variety of additional user inputs and constraints, e.g., by modifying the field to change the truss directions, or by adding beams in the second stage to support a connection to a separate object. It is also very efficient, with optimization converging in a few iterations, and quite consistently.

While we use a highly simplified model for cell mechanics, the overall approach admits replacing this model with a more advanced finite element formulation. In this case, the local step would likely require numerical optimization; however, as long as the model for a cell stays low-parametric, one is likely to be able to solve it efficiently.

Future Work. In the future, we will study the effect of alternating optimizing the cell mesh $P$, which defines the layout of the beams, and optimizing beam parameters. While our evaluations on the role of beam directions (Figure 17) suggest only an incremental benefit, it might help regions with large stress concentrations. Multiple loading scenarios can be optimized for simultaneously, following the approach of [Sokół and Rozvany 2016], i.e., computing stresses for each load case and constraining them to be bounded.

Limitations. Our work has two main limitations: first, the formulation for the field optimization still has restrictive low-volume and fixed-thickness assumptions. Based on our evaluation of field direction sensitivity of the final design, we do not view this as a significant limitation. The second, more significant, limitation is the...
Fig. 30. Examples of structures obtained for a variety of shapes and loads. For each one, images show loads and initial stress distribution, quadrangulation, cell optimization (colored by thickness in logarithmic scale) and final geometry.
highly simplified model we used. If exact results are needed, this model can be used to quickly obtain an initial result, which can then be refined using a more advanced mechanical description of cells and shape optimization.

ACKNOWLEDGMENTS

We thank Chelsea Tymms for help with printing and photography, and the anonymous reviewers for their useful comments. This work was supported in part by the NSF grant OAC-1835712, the NSF award OIA-1937043, and a gift from Adobe Research.

REFERENCES


ACM Trans. Graph., Vol. 1, No. 1, Article 1. Publication date: January 2020.
We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

We drop the cell superscript which verifies convexity of the energy.

The minimum of this expression is

This derivation follows [Strang and Kohn 1983], which we include.


Structural and Multidisciplinary Optimization 52, 6 (2015), 1161–1184.

We express the bending strain in the form

where

A direct evaluation shows that \( u^T H(z) u \) evaluated for \( v = [1, 0, 0] \) is negative for positive \( z_1 \) and \( g_b \). Similarly is true for \( v = [0, 1, 0] \) and \( v = [0, 0, 1] \). We conclude if \( g_b \) is zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.

The constraint \( y_1 + y_2 + y_3 \leq 1 \) defines, if \( g_b \) is zero for some i, any critical point in the interior of the domain is a maximum or a saddle, so there are no minima in the interior. If all \( g_b \) are zero, the volume is a linear function of \( z_1 \), and the optimum is also on the boundary.
For a vector $\mathbf{a} = [a_x, a_y, a_z]$, let $R(\alpha)$ be the infinitesimal rotation matrix about $\alpha$:

$$
\begin{bmatrix}
0 & -a_z & a_y \\
 a_z & 0 & -a_x \\
-ay & a_x & 0
\end{bmatrix}.
$$

Then

$$
\Delta \hat{n}_i = \sum_{l \in N(i)} M^i_l (u_l - u_i)
$$

where $M^i_l = \frac{1}{|n_i|} P_l R(g_l)$ is a $3 \times 3$ matrix.

$$
\epsilon^{h}[ij] = \hat{h} \left( \sum_{l \in N(j)} (d^j_l)^T (u_l - u_j) + \sum_{l \in N(i)} (d^i_l)^T (u_l - u_i) \right)
$$

where $d^i_l = -(M^i_l)^T \hat{e}_{ij} / l_{ij}$, a vector of length 3. From this expression, we can immediately obtain $D_{ij}$. 