Fine-scale Structure Design for 3D Printing

by

Francis Julian Panetta

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May, 2017

Denis Zorin

Abstract

Modern additive fabrication technologies can manufacture shapes whose geometric complexities far exceed what existing computational design tools can analyze or optimize. At the same time, falling costs have placed these fabrication technologies within the average consumer's reach. Especially for inexpert designers, new software tools are needed to take full advantage of 3D printing technology.

This thesis develops such tools and demonstrates the exciting possibilities enabled by fine-tuning objects at the small scales achievable by 3D printing. The thesis applies two high-level ideas to invent these tools: two-scale design and worst-case analysis.

The two-scale design approach addresses the problem that accurately simulating—let alone optimizing—the full-resolution geometry sent to the printer requires orders of magnitude more computational power than currently available. However, we can decompose the design problem into a small-scale problem (designing tileable structures achieving a particular deformation behavior) and a macro-scale problem (deciding where to place these structures in the larger object). This separation is particularly effective, since structures for every useful behavior can be designed once, stored in a database, then reused for many different macroscale problems.

Worst-case analysis refers to determining how likely an object is to fracture by studying the worst possible scenario: the forces most efficiently breaking it. This analysis is needed when the designer has insufficient knowledge or experience to predict what forces an object will undergo, or when the design is intended for use in many different scenarios unknown a priori.

The thesis begins by summarizing the physics and mathematics necessary to rigorously approach these design and analysis problems. Specifically, the second chapter introduces linear elasticity and periodic homogenization.

The third chapter presents a pipeline to design microstructures achieving a wide range of effective isotropic elastic material properties on a single-material 3D printer. It also proposes a macroscale optimization algorithm placing these microstructures to achieve deformation goals under prescribed loads.

The thesis then turns to worst-case analysis, first considering the macroscale problem: given a user's design, the fourth chapter aims to determine the distribution of pressures over the surface creating the highest stress at any point in the shape. Solving this problem exactly is difficult, so we introduce two heuristics: one to focus our efforts on only regions likely to concentrate stresses and another converting the pressure optimization into an efficient linear program.

Finally, the fifth chapter introduces worst-case analysis at the microscopic scale, leveraging the insight that the structure of periodic homogenization enables us to solve the problem exactly and efficiently. Then we use this worst-case analysis to guide a shape optimization, designing structures with prescribed deformation behavior that experience minimal stresses in generic use.

Table of Contents

List of Figures

Chapter 1

Introduction

3D printing technology has placed powerful fabrication tools within the average consumer's reach. It is now affordable to produce customized, one-off designs, whether they be sculptures, jewelry, or special-purpose tools invented to solve specific problems. Browsing the thousands of designs in online 3D printing marketplaces like Shapeways, Kraftwürx, and i.materialize, one sees just how diverse and exciting the applications can be.

The promise of additive fabrication extends far beyond customizing the same types of shapes that engineers and sculptors have designed for centuries, though. New fabrication technologies like projection micro-stereolithography and direct laser writing enable a qualitatively different class of designs, with complex geometry on the scale of micrometers. Moreover, objects of this type, with thousands of intricate features at the microscopic scale, cost no more to produce—in terms of money and fabrication time—than traditional, simple shapes.

It has long been known that the optimal solutions to several important engineering problems, such as finding the stiffest structure not exceeding a weight bound, are designs with microscopic perforations. Using additive fabrication, we can finally hope to realize such structures. Indeed, as this thesis explores, by tuning fine-scale geometry, we can create objects exhibiting interesting and useful deformation behavior impossible to achieve with solid designs.

However, navigating this expanded design space to take full advantage of additive fabrication technologies is a challenging task, calling for the development of new software tools to analyze and optimize designs. Existing design tools running on modern workstations are incapable of directly analyzing and optimizing shapes at the resolutions we can now fabricate: the B9Creator, the inexpensive stereolithography printer which fabricated many of the results in this thesis, has a maximum resolution of over eight billion voxels ($1920 \times 1080 \times 4000$).

1.1 Two-Scale Design

To reduce the design problem's complexity while still leveraging the 3D printer's full resolution, this thesis employs a *two-scale design approach*. We take advantage of the fact that fine-scale structure, when fabricated at a small enough scale relative to the full object, behaves approximately like a homogeneous material with some particular macroscopic properties (density, elasticity, etc.). This approximation allows us to divide the design problem into two stages: first, we design a database of structures that, when tiled periodically, achieve as wide a range of macroscopic material properties as possible. Second, we solve a coarse-resolution design problem, where the variables are now the material properties at each point. After we obtain an optimal coarse design, we convert it into a high-resolution, single material object by replacing each point's chosen material with a structure approximating it.

This thesis focuses primarily on designing objects that can be fabricated on a single-material printer to meet specified deformation goals under prescribed forces. For many applications, these goals are impossible without modulating the printing materials' properties using fine-scale structures: most solid printing materials are far too stiff to deform perceptibly under forces of reasonable magnitude. Even in engineering applications where deformations are intended to be small, they still need to be controlled precisely.

For deformation goals, the relevant material properties are encoded by *elasticity tensors*, which generalize the concept of a spring constant to specify the elastic forces arising when a linear $3D$ material is stretched or sheared. General linear 3D materials can behave counter-intuitively, for instance shearing in response to a uniaxial force, or exhibiting drastically different stiffness when stretched along different directions. This anisotropic behavior can be useful for many applications, but this thesis chooses to focus on designing structures with *isotropic* elastic properties.

Isotropic materials are more intuitive to work with than general anisotropic materials due to their orientation invariance: they respond identically no matter how the material is rotated relative to the stretch. They also yield particularly simple variables for the coarse-resolution design problem: the homogenized material is described by only two parameters, each with intuitive physical significance. Several parametrizations can be used to specify isotropic materials, but we choose the Young's modulus (specifying stiffness), and the Poisson's ratio (specifying how the material expands or contracts orthogonally in response to a uniaxial stretch). Finally, since the isotropic material property space has just two dimensions, we can design a database of structures densely sampling the space; the curse of dimensionality precludes this for, e.g., orthotropic materials (described by nine parameters).

Chapter 3 details a full software pipeline implementing this two-scale approach. It discusses in detail how to design periodic microstructures emulating a broad range of isotropic material properties using shape optimization. One key distinction from previous work in the engineering and mathematics communities is that our structures are constrained to be directly *manufacturable* on off-the-shelf 3D printers like the B9Creator. Another is that we design a large, dense collection of structures uniformly sampling the full range of achievable material properties, as opposed to designing a few structures demonstrating extremal behavior. This dense coverage permits fine-tuned control of properties throughout the full achievable range. The chapter also shows how to automatically place these microstructures in a macroscopic object to best achieve a deformation goal and demonstrates that objects designed with this approach do indeed produce the desired deformations when fabricated.

1.2 Worst-Case Stress Analysis

A key problem emerges for practical 3D printing applications, especially when producing objects with small features: objects are often fragile and may not even survive the printing process. Particularly for consumers without backgrounds in mechanical engineering, it is difficult to design objects that will survive typical force loads (or predict what these loads may be to begin with). Even for seasoned engineers, the designs may be so complex due to the intricate details permitted by 3D printing that analysis is difficult. To provide a designer with guarantees that their shape is sufficiently robust, we turn to worst-case analysis.

Worst-case stress analysis refers to finding the greatest stress that could possibly occur at a point in the object when the shape is subjected to any load from some space of possible forces. If we can determine that, for every point in the object, this stress level falls below the printing material's yield stress, we can reassure the user that their design is sound. On the other hand, if we discover a worst-case load that induces stresses exceeding this threshold, we can warn the user, providing a visualization of the dangerous forces and deformations. Even better, we can automatically adjust the design to relieve these worst-case stresses while constraining the object to still meet its performance goals.

As implied above, worst-case analysis involves a maximization over the space of forces, performed separately for each point in the object. Unfortunately, this optimization turns out to be non-convex, making the analysis for even just a single point quite difficult. This is because even though we wish to optimize a convex quantity (the maximum eigenvalue of stress) over a convex set of forces (bounded pressure quantity on each boundary point), our goal is to maximize rather than minimize. However, Chapter 4 argues that, in worst-case scenarios, one stress tensor eigenvalue typically dominates so that the maximum principal stress can generally be approximated by the stress tensor's trace. This approximation converts the optimization into a linear program, which can be solved reasonably efficiently. To further accelerate the analysis, we predict regions likely to experience high stresses by analyzing the object's vibrational modes and restrict the worst-case analysis to only these regions (as opposed to every element in the simulation mesh). The result is a pipeline allowing novice users to analyze the structural stability of their designs without needing to know ahead of time what forces might break it.

For several reasons, worst-case structural analysis is an even more essential design tool for the two-scale design approach we advocate. First, the microstructures we construct to emulate various macroscopic behaviors tend to concentrate stresses. Particularly for emulating soft, flexible materials, the structures end up quite fragile and tend to break under even small loads. Detecting and correcting these stress concentrations is essential. Second, because a robust, efficient design tool requires pre-computing a database of microstructures for the full range of achievable material properties (rather than attempting to design microstructures on-the-fly when the course-resolution optimization requests them), all microstructure design is performed before the specific use-case is known. The structures must perform well in all possible macroscopic designs in which they might be used, necessitating analysis of the *worst case*. Finally, there is a large degree of redundancy in our structure's coverage of isotropic elastic properties: for each covered elasticity tensor, there is generally an entire space of structures achieving that tensor. Within this space, there is dramatic variation in pointwise stress behavior, meaning that incorporating worst-case stress consideration into the microstructure design tool can produce a collection of significantly more robust structures.

Fortunately, periodic homogenization happens to substantially simplify the worst-case stress analysis problem, to the point where it can be solved exactly at every element in the simulation mesh with negligible computational cost. Further, the simple formulation makes it straight-forward to differentiate the worst-case stress with respect to perturbations of the shape, enabling shape optimization.

Chapter 5 applies this insight to design microstructures producing prescribed isotropic elasticity tensors while experiencing reduced peak stresses in the worst case. In order to ensure every design can be fabricated, we perform this optimization over a parametric design space (where fabrication constraints can be formulated as simple inequality constraints on the design variables) instead of using a traditional SIMP or level-set topology optimization algorithm. To obtain meaningful stress reduction and good coverage of macroscopic properties, the design space must be defined carefully. We start with the truss-like structures introduced in Chapter 3, but enrich the designs with additional parameters controlling the smoothness of the joints. This enriched space leads to a substantial increase in material property coverage over Chapter 3 (extending the lower bound of Poisson's ratios from -0.16 down to -0.7 and the upper bound of relative Young's moduli up from 0.1 to nearly 1) and achieves a typical stress reduction of 5×.

Chapter 5 uses implicit surface modeling to define the microstructure geometry, as such modeling techniques are well suited for defining smooth, organic structures. However, existing techniques for constructing smooth joints in the implicit surface modeling communities violate our fabrication constraints. One of the primary contributions of this work is a new shape blending operator that restricts where material is added during the joint smoothing process to ensure fabrication constraints are not violated.

Chapter 2

Background

This chapter provides a brief introduction to linear elasticity and periodic homogenization used throughout the thesis.

2.1 Linear Elasticity

We use linear elasticity to solve for the static equilibrium of solid objects—both macroscopic and microscopic—under prescribed forces and displacements. We provide a brief, informal derivation of linear elasticity and introduce notation.

The state of a deformed body is encoded by a displacement vector field, $\mathbf{u}$, defined over its original, undeformed configuration, Ω (so that each point $\mathbf{x} \in \Omega$ is moved to point $\mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x})$ by the deformation). As the body deforms, its material at each point stretches and exerts restoring forces. To fully describe the stretch and resulting forces at a point requires two 3×3 matrices: the stress and strain tensors.

2.1.1 Strain Tensor

The quantity giving rise to internal restoring forces is the amount by which the deformation stretches or compresses an arbitrarily-oriented line segment between two infinitesimally spaced material points in Ω. We describe this line segment by the tangent vector $\mathbf{v}$ and note that the deformation stretches it into the new segment $(I + \nabla \mathbf{u})\, \mathbf{v} \overset{\text{def}}{=} F\mathbf{v}$ in the deformed body, where F is the Jacobian of the deformation map $\mathbf{x} \mapsto \mathbf{x} + \mathbf{u}(\mathbf{x})$. We define a strain tensor to measure the change in length of this segment.

There are several different ways to measure strain for large deformations used in practice, each defining a corresponding strain tensor that can be computed from the deformation's Jacobian F (or its inverse) [34]. Since each strain tensor reduces to the same linear elasticity strain tensor for small deformations, we consider only the simplest: the Green-Lagrange strain.

We derive this tensor by computing (half) the change in the tangent vector's *squared* length. Since the squared length after the deformation is $\|F\mathbf{v}\|^2 = \mathbf{v}^T F^T F \mathbf{v}$, this quantity can be written as a quadratic form:

$$\frac{1}{2}\Big(\|F\mathbf{v}\|^2 - \|\mathbf{v}\|^2\Big) = \frac{1}{2}\mathbf{v}^T\Big(F^T F - I\Big)\mathbf{v} = \frac{1}{2}\mathbf{v}^T\Big(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u}\Big)\mathbf{v}.$$

The Green-Lagrange tensor is the matrix corresponding to this quadratic form, $\frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u})$, and thus encodes the change in length for *every* oriented infinitesimal line segment at a point in the undeformed body.

For small deformations, $\|\nabla\mathbf{u}\| \ll 1$, and we can neglect the nonlinear term $\nabla\mathbf{u}^T\nabla\mathbf{u}$, arriving at the linear Cauchy strain tensor:

$$\varepsilon(\mathbf{u}) \overset{\text{def}}{=} \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right).$$

This linear strain tensor is also referred to as the "symmetrized gradient" because it is the symmetric part of $\nabla \mathbf{u}$.

Symmetrization projects out the antisymmetric part $\frac{1}{2}\left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right)$, which is the infinitesimal (linearized) rotation applied by the displacement $\mathbf{u}$. Consequently, the linear strain measure is approximately invariant to small rotations, which is desirable since rotations do not stretch the material. Unfortunately, large rotations are not filtered out properly by the symmetrization, leading to inaccurate behavior for large deformations, where rotations are interpreted as stretching the material. This is the price we pay for using the simple linear elasticity equations.

2.1.2 Stress Tensor

When the material at a point is strained, it applies internal forces to the surrounding material. Consider a plane through the point in question with normal $\hat{\mathbf{n}}$. The material on the positive side of this plane, m^+, will apply *traction* (force per unit area) $\tau(\hat{\mathbf{n}})$ to the material on material on the negative side, m^-. One can show that, for the net forces acting on an infinitesimal tetrahedron ("Cauchy's tetrahedron") to be finite, $\tau(\hat{\mathbf{n}})$ is actually a linear function. Thus τ can be expressed as a matrix: $\tau(\hat{\mathbf{n}}) \stackrel{\text{def}}{=} \sigma \hat{\mathbf{n}}$. This matrix is the stress tensor.

Further, one can show that σ must be symmetric in order for the torque on an infinitesimal cube of material to be finite. This means there are always three principal stresses (the eigenvalues) acting on planes with mutually orthogonal normals (the eigenvectors). The maximum eigenvalue, the "maximum principal stress," will be used to establish failure criteria in Chapters 4 and 5.

2.1.3 Elasticity Tensor

For small deformations, the material's response to strains is approximately linear. This is expressed by Hooke's law:

$$\sigma = C : \varepsilon(\mathbf{u}), \tag{2.1}$$

or in index notation (where repeated indices are summed over the range $0, 1, 2$):

$$\sigma_{ij} = C_{ijkl}[\varepsilon(\mathbf{u})]_{kl}.$$

Operation ":" is called double contraction; it is similar to matrix multiplication, but instead of summing over the single innermost index, it sums over the inner two indices.

Here, C is a rank-four tensor implementing the linear map from strain to stress, generalizing the $1D$ spring constant to 3D solids. This tensor fully encodes the body's (linear) elastic material properties (which can vary from point to point). In Chapters 3 and 5, we design structures to achieve particular elasticity tensors in the sense of periodic homogenization, described in Section 2.3 below.

2.1.4 Static Equilibrium

Here, we derive the (linear elastostatic) equations solving for the deformation putting body Ω in static equilibrium with the applied external loads.

Consider how internal and external forces must balance for a small internal portion of material, $D \subset \Omega$, in static equilibrium. The material experiences a net internal force from the material surrounding it of:

$$f_{\text{internal}} \overset{\text{def}}{=} \int_{\partial D} \sigma \hat{\mathbf{n}} \, \mathrm{d}A(\mathbf{x}) = \int_{D} \nabla \cdot \sigma \, \mathrm{d}\mathbf{x}.$$

The second equality follows from Green's theorem. The region also experiences some total external force, f_{external}, and for force balance:

$$f_{\text{internal}} + f_{\text{external}} = 0$$

If we assume f_{external} is distributed smoothly throughout the object, we can consider a small enough portion of material D that the *force density* (force per unit volume), f, is essentially constant over D: $f_{\text{external}} \approx f|D|$ (where $|D|$ is the region's volume). Also, assuming stresses are sufficiently smooth, $\nabla \cdot \sigma$ is also essentially constant over $|D|$, so force balance looks like:

$$(\nabla \cdot \sigma)|D| + f|D| = 0 \quad \Longrightarrow \quad -\nabla \cdot \sigma = f \ \text{ in } D.$$

But, for D overlapping the boundary, we of course find that the external forces due to boundary tractions (force per unit area) are fundamentally different from f: they *not* smooth, or even finite, when viewed as a volume force density. To see this, notice that we can shrink D's volume to zero while keeping it overlapping the same boundary region $\Gamma \overset{\text{def}}{=} \partial D \cap \partial \Omega$ by making it arbitrarily skinny.

So to formulate the boundary force balance, it is difficult to work in terms of volume force densities in the interior; instead we directly consider tractions (area force densities) on $\partial\Omega$. Recall that, by definition, the material inside Ω pushes back on Γ with traction $-\sigma\hat{\mathbf{n}}$. So the net force acting on the surface is:

$$\int_\Gamma -\sigma\hat{\mathbf{n}} \, \mathrm{d}A(\mathbf{x}) + f_{\text{external}} = 0.$$

Note that f_{external} now has zero contribution from f—i.e. all forces measured in this equation live solely on the surface Γ—because we have shrunk $|D|$ to zero.

Now, assuming f_{external} is spread smoothly over Γ, for small enough Γ we have an approximately constant force area density (traction) $g \overset{\text{def}}{=} \frac{f_{\text{external}}}{|\Gamma|}$. Since σ is also assumed smooth over Γ, the surface force balance can be rewritten as:

$$-\sigma\hat{\mathbf{n}}|\Gamma| + g|\Gamma| \quad \Longrightarrow \quad \sigma\hat{\mathbf{n}} = g \ \text{ on } \Gamma.$$

Considering regions D around all points in the interior and on the boundary of Ω, we argue that for the whole body to be in static equilibrium the following must hold:

$$\begin{aligned} -\nabla \cdot \sigma &= f \quad \text{in } \Omega \\ \sigma\hat{\mathbf{n}} &= g \quad \text{on } \partial\Omega. \end{aligned} \tag{2.2}$$

Again, the units of the first equation are force per unit volume (divergence has units of inverse distance), and the units of the second equation are force per unit area. This equation can be expressed in terms of the unknown displacement field, $\mathbf{u}$, using Hooke's law and the linear strain tensor:

$$\begin{aligned} -\nabla \cdot C : (\nabla\mathbf{u} + (\nabla\mathbf{u})^T) &= f \quad \text{in } \Omega \\ \hat{\mathbf{n}} \cdot (\nabla\mathbf{u} + (\nabla\mathbf{u})^T) &= g \quad \text{on } \partial\Omega. \end{aligned} \tag{2.3}$$

Finally, instead of specifying tractions on the full boundary (Neumann boundary conditions), we can directly specify the displacement on part or all of the boundary with a Dirichlet boundary condition. This condition will enforce the desired displacement by applying whatever traction necessary to the boundary region.

2.2 Right and Left Nullspace of Linear Elasticity

To formulate well-posed simulations, it is useful to know precisely the space of displacement fields inducing no internal forces in the material (the right nullspace), and precisely the space of force fields that cannot arise from elastic forces (the left nullspace).

2.2.1 Infinitesimally Rigid Motions

Because the displacement field only enters into (2.3) via the strain tensor strain tensor (which computes a derivative), obviously all global translations (spatially constant displacements) are in the nullspace. These are spanned by the 3 canonical basis vectors $\mathbf{e}_i$.

However, as mentioned when deriving the linear strain tensor, symmetrization drops exactly the skew-symmetric part of $\nabla \mathbf{u}$. This means that any displacement vector field with a skew symmetric Jacobian at every point is in the nullspace (and, assuming C is positive definite, no other displacement fields are). The space of skew symmetric matrices is 3 dimensional, and it turns out that the integrability condition $\nabla \times \nabla \mathbf{u} = 0$ (applying the curl to each row) requires the skew-symmetric component of $\nabla \mathbf{u}$ to be *globally constant*. This proves the intuitively clear statement that symmetrizing $\nabla \mathbf{u}$ expands the nullspace by exactly 3 dimensions corresponding to global infinitesimal rotations. We exhibit an explicit basis for these additional dimensions by noting:

$$\varepsilon(\mathbf{e}_l \times \mathbf{x}) = \varepsilon(\delta_{lr} x_s \epsilon_{rst} \mathbf{e}_t) = \frac{1}{2}\left(\frac{\partial x_s \epsilon_{lst} \delta_{ti}}{\partial x_j} + \frac{\partial x_s \epsilon_{lst} \delta_{tj}}{\partial x_i}\right) = \frac{1}{2}\left(\epsilon_{lji} + \epsilon_{lij}\right) = 0.$$

Here we used the Levi-Civita symbol:

$$\epsilon_{ijk} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } (i,j,k) \text{ is an even permutation of } (0,1,2), \\ -1 & \text{if } (i,j,k) \text{ is an odd permutation of } (0,1,2), \\ 0 & \text{if any index is repeated,} \end{cases}$$

to express the cross product:

$$\mathbf{a} \times \mathbf{b} = a_i b_j \mathbf{e}_k \epsilon_{ijk},$$

with implied summation over i, j, and k. The antisymmetry property $\epsilon_{lji} + \epsilon_{lij} = 0$ used above follows from the definition of ϵ.

Hence, our explicit basis for the right nullspace of the linear elasticity operator comprises the *vector fields*:

$$\big\{\mathbf{e}_0,\ \mathbf{e}_1,\ \mathbf{e}_2,\ \mathbf{e}_0 \times \mathbf{x},\ \mathbf{e}_1 \times \mathbf{x},\ \mathbf{e}_2 \times \mathbf{x}\big\}. \tag{2.4}$$

2.2.2 Zero Net Force and Torque

Vector fields (2.4) also happen to form a basis for the left nullspace. This is not surprising, since taking an inner product of a force field, $\mathbf{f}$, with the forces arising from a stress field, σ:

$$\int_\Omega -\mathbf{f} \cdot \Big(\nabla \cdot \sigma\Big)\ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \mathbf{f} \cdot \sigma\hat{\mathbf{n}}\ \mathrm{d}A(\mathbf{x}) = \int_\Omega \varepsilon(\mathbf{f}) : \sigma\ \mathrm{d}\mathbf{x},$$

(using integration by parts (2.7)). So, for this inner product to vanish for arbitrary σ, the symmetrized gradient of f must vanish, which was the condition leading to (2.4).

In this context, the first three basis fields show that the net internal force vanishes:

$$\int_\Omega \mathbf{e}_i \cdot (-\nabla \cdot \sigma)\ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \mathbf{e}_i \cdot \sigma\hat{\mathbf{n}}\ \mathrm{d}A(\mathbf{x}) = \mathbf{e}_i \cdot \left(\int_\Omega -\nabla \cdot \sigma\ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \sigma\hat{\mathbf{n}}\ \mathrm{d}A(\mathbf{x})\right).$$

(Vanishing net force could also be proved directly by applying divergence theorem to the right-hand side). The remaining three show that net torque vanishes:

$$-\int_\Omega (\mathbf{e}_i \times \mathbf{x}) \cdot (\nabla \cdot \sigma)\ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} (\mathbf{e}_i \times \mathbf{x}) \cdot \sigma\hat{\mathbf{n}}\ \mathrm{d}A(\mathbf{x}) = \mathbf{e}_i \cdot \left(-\int_\Omega \mathbf{x} \times \nabla \cdot \sigma\ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \mathbf{x} \times \sigma\hat{\mathbf{n}}\ \mathrm{d}A(\mathbf{x})\right).$$

One could also directly show that net torque around the origin vanishes using an integration by parts:

$$\begin{aligned}\int_\omega x_i \frac{\partial \sigma_{jd}}{\partial x_d} \epsilon_{ijk} \, \mathrm{d}\mathbf{x} &= -\int_\omega \frac{\partial x_i}{\partial x_d} \sigma_{jd} \epsilon_{ijk} \, \mathrm{d}\mathbf{x} + \int_{\partial\omega} n_d x_i \sigma_{jd} \epsilon_{ijk} \, \mathrm{d}A(\mathbf{x}) \\ &= -\int_\omega \cancel{\sigma_{ji}\epsilon_{ijk}} \, \mathrm{d}\mathbf{x} + \int_{\partial\omega} n_d x_i \sigma_{jd} \epsilon_{ijk} \, \mathrm{d}A(\mathbf{x}),\end{aligned}$$

where the canceled integrand vanishes because a symmetric tensor (σ) is double-contracted with an antisymmetric tensor (ϵ). Interpreting this identity in index-free notation, we see that net torque vanishes:

$$\int_\Omega \mathbf{x} \times \nabla \cdot \sigma \, \mathrm{d}\mathbf{x} = \int_{\partial\Omega} \mathbf{x} \times \sigma \hat{\mathbf{n}} \, \mathrm{d}A(\mathbf{x}).$$

2.2.3 No Rigid Motion Constraint

We can use the nullspace bases above to formulate no-rigid-motion and no-net-force constraints. By setting equal to zero the inner product of displacement field $\mathbf{u}$ with each basis field from (2.4), we find:

$$\int_\Omega \mathbf{u} \, \mathrm{d}\mathbf{x} = 0, \quad \int_\Omega \mathbf{u} \times \mathbf{x} \, \mathrm{d}\mathbf{x} = 0. \tag{2.5}$$

Likewise, by taking the inner product of the basis with loads f and g, we obtain the no net force/torque constraints:

$$\int_\Omega f \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} g \, \mathrm{d}A(\mathbf{x}) = 0, \quad \int_\Omega f \times \mathbf{x} \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} g \times \mathbf{x} \, \mathrm{d}A(\mathbf{x}) = 0. \tag{2.6}$$

If the no-rigid-motion constraint (2.5) is applied to the simulation, then the Lagrange multipliers for this constraint apply an additional force to counter whatever net force/torque arise from the explicitly applied forces.

For the worst-case stress problem in Chapter 4 (formulated with no-rigid-motion constraints), it will therefore be essential to additionally require forces to satisfy (2.6) to prevent additional “ghost” Lagrange multiplier forces not accounted for in the force budget (and violating the model of user interaction which only permits inward surface forces).

2.2.4 An Integration by Parts Formula

We derive a useful integration by parts formula for linear elasticity that “moves the strain operator off of” an arbitrary vector field ϕ:

$$\begin{aligned}
\int_\Omega \varepsilon(\phi) : \sigma \;\mathrm{d}\mathbf{x} &= \int_\Omega \frac{\partial \phi_i}{\partial x_j} \sigma_{ij} \;\mathrm{d}\mathbf{x} \\
&= \int_\Omega \frac{\partial}{\partial x_j}\Big(\phi_i \sigma_{ij}\Big) - \phi_i \frac{\partial \sigma_{ij}}{\partial x_j} \;\mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \phi_i \sigma_{ij} n_j \;\mathrm{d}A(\mathbf{x}) - \int_\Omega \phi_i \frac{\partial \sigma_{ij}}{\partial x_j} \;\mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \phi \cdot \sigma \hat{\mathbf{n}} \;\mathrm{d}A(\mathbf{x}) - \int_\Omega \phi \cdot \Big(\nabla \cdot \sigma\Big) \;\mathrm{d}\mathbf{x}. \qquad (2.7)
\end{aligned}$$

The first step uses the symmetry of σ to drop the explicit symmetrization of ϕ’s gradient, the second applies the product rule, the third applies the divergence theorem, and the last reinterprets the result in index-free notation.

2.3 Periodic Homogenization for Linear Elasticity

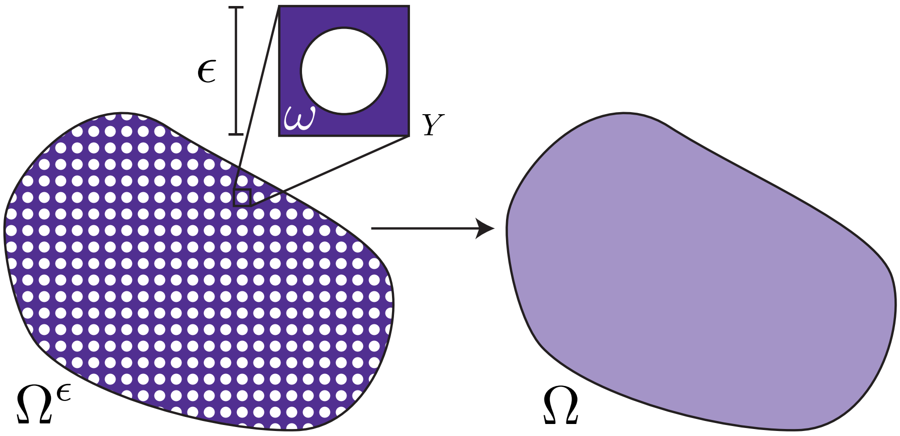


Figure 2.1: Base cell geometry ω is tiled with period ϵ throughout Ω to obtain a porous, microstructured object Ω^ϵ. As $\epsilon \to 0$, Ω^ϵ behaves identically to a solid version of Ω filled with "homogenized" material C^H.

In solid mechanics, we model objects as continua. That is, we assume that an object is of large enough size that we needn't consider its individual molecules and their interactions. We instead capture the effect of all these micro-scale interactions using a constitutive relation, like Hooke's law (2.1), describing how much force is required to stretch an infinitesimal piece of the object.

But what if the object is porous or a fine-scale composite of multiple materials? Can we apply the same assumptions at a larger length scale to still model the object as a solid, homogeneous continuum? It turns out that we can, and homogenization theory gives a rigorously justified method to do it.

This thesis employs periodic homogenization to determine the effective elastic material properties of a periodic microstructure when printed using a single material printer.

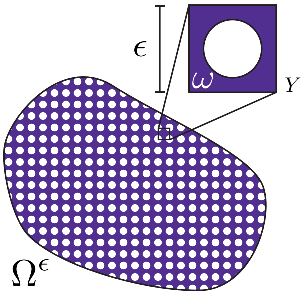
2.3.1 Periodic Homogenization

Our goal is to find the homogenized elasticity tensor C^H, which represents the effective properties of the microstructure when it is fabricated at a small enough scale, ϵ. That is, if we fabricate a solid version of Ω with material C^H, it will deform identically to microstructured object Ω^ϵ for small enough ϵ (Figure 2.1).

One might ask: why not just spatially average the elasticity tensor over the microstructure period cell? But it is clear even from the 1D case why this does not give the correct answer: consider two equal-length springs connected in series with very different spring constants $k_1 \ll k_2$. The spatial averaging approach would claim that the coupled spring's effective spring constant is $\frac{1}{2}(k_1 + k_2)$. However, the combined springs are much less stiff than this; the true effective stiffness is the harmonic mean of k_1 and k_2, much closer to k_1 than to the average. In the limiting case $k_1 \to 0$, it is intuitively obvious that the full system's stiffness should also approach 0.

Framework

The homogenization process is simplified by considering *periodic* microstructures. This means that our microstructured object, Ω^ϵ, is determined by tiling Ω with the geometry ω contained in period cell Y. Parameter ϵ determines the scaled-down size of cell Y, so the smaller ϵ becomes, the more repetitions of ω are fit into Ω, and the closer we get to a homogeneous material (in theory).



We formalize the tiling process as follows. We define an infinite tiling of the period cell Y, with the microscopic variable $\mathbf{y}$ specifying locations in this grid. The material properties of the tiled microstructure at each point $\mathbf{y}$ then define a periodic function $C(\mathbf{y})$, with period in each coordinate direction given by the dimensions of Y. At a given length scale, ϵ, we define the material properties a point $\mathbf{x} \in \Omega^\epsilon$ by evaluating this periodic function C at location $\mathbf{y} = \mathbf{x}/\epsilon$.

For our application, fabricating single-material porous objects rather than composites, the periodic, spatially-varying elasticity tensor is given by:

$$C(\mathbf{y}) = \begin{cases} C^{\text{base}} & \text{if } (\mathbf{y} \mod |Y|) \in \omega \\ 0 & \text{otherwise} \end{cases}, \tag{2.8}$$

where C^{base} is the printing material.

Static Equilibria

We express the static equilibrium conditions (from (2.3)) for both the microstructured object Ω^ϵ and the "homogenized" solid object Ω:

$$\underbrace{\begin{aligned} -\nabla \cdot C\left(\frac{\mathbf{x}}{\epsilon}\right) : \varepsilon(\mathbf{u}^\epsilon) = f \quad \text{in } \Omega \\ \hat{\mathbf{n}} \cdot C\left(\frac{\mathbf{x}}{\epsilon}\right) : \varepsilon(\mathbf{u}^\epsilon) = g \quad \text{on } \partial\Omega \end{aligned}}_{\text{Static equilibrium for } \Omega^\epsilon} \quad \Bigg| \quad \underbrace{\begin{aligned} -\nabla \cdot C^H : \varepsilon(\bar{\mathbf{u}}) = f \quad \text{in } \Omega \\ \hat{\mathbf{n}} \cdot C^H : \varepsilon(\bar{\mathbf{u}}) = g \quad \text{on } \partial\Omega \end{aligned}}_{\text{Static equilibrium for } \Omega} \tag{2.9}$$

Our goal is to determine the spatially-constant elasticity tensor C^H such that $\bar{\mathbf{u}}$ found on the right matches $\mathbf{u}^\epsilon$ (after its high-frequency components are filtered out) in the limit $\epsilon \to 0$ for any smooth loads f and g.

Heuristic Derivation

The microstructured object's equilibrium displacement $\mathbf{u}^\epsilon$ has a high frequency periodic component that, as period $\epsilon \to 0$, averages out to $\bar{\mathbf{u}}$. So $\bar{\mathbf{u}}(\mathbf{x})$ can be thought of as the average displacement over the infinitesimal base cell Y at point $\mathbf{x}$. Likewise, $\varepsilon(\bar{\mathbf{u}})$ is the average strain in the cell. For the object to be in equilibrium, (2.9, right) should represent a balance of the average forces over the cell, meaning $C^H : \varepsilon(\bar{\mathbf{u}})$ should be the average stress tensor. That gives the following intuitive interpretation of C^H: it maps the average strain applied at a point to the average stress resulting within the microstructure geometry.

Thus, applying C^H to $\varepsilon(\bar{\mathbf{u}})$ is equivalent to simulating the microstructure's deformation under that average strain and averaging its stress. We formulate this simulation inside a single base cell Y by assuming that the displacement consists of a linear term (with constant strain $\varepsilon(\bar{\mathbf{u}})$) plus a Y-periodic "microscopic fluctuation" term, $\mathbf{w}$ (with zero average strain by periodicity). This assumption is reasonable because, by the translational symmetry of an infinite tiling, every cell will deform identically (i.e. the strain field will be periodic). Now we simply solve for the particular fluctuation displacement, $\mathbf{w}$, putting the tiled structure in equilibrium:

$$-\nabla \cdot \Big(C(\mathbf{y}) : [\varepsilon(\mathbf{w}(\mathbf{y})) + \varepsilon(\bar{\mathbf{u}})] \Big) = 0 \text{ in } Y, \tag{2.10}$$

where $\mathbf{y}$ is the microscopic variable (the coordinate in Y). Then the average stress is $C^H : \varepsilon(\mathbf{u}) = \frac{1}{|Y|}\int_Y C : [\varepsilon(\mathbf{w}) + \varepsilon(\mathbf{u})] \, \mathrm{d}\mathbf{y}$.

We can extract the components of C^H by applying it to the six canonical symmetric rank 2 basis tensors, $e^{kl} := \frac{1}{2}(\mathbf{e}_k \otimes \mathbf{e}_l + \mathbf{e}_l \otimes \mathbf{e}_k)$. Each application amounts to solving the *cell problem*:

$$-\nabla \cdot \Big(C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \Big) = 0 \text{ in } \omega, \tag{2.11a}$$

$$\hat{\mathbf{n}} \cdot \Big(C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \Big) = 0 \text{ on } \partial\omega \backslash \partial Y, \tag{2.11b}$$

$$\mathbf{w}^{kl} \ Y\text{-periodic}, \tag{2.11c}$$

$$\int_\omega \mathbf{w}^{kl} \, \mathrm{d}\mathbf{y} = 0, \tag{2.11d}$$

where we rephrased the microscopic force balance as a PDE over ω with constant coefficients using (2.8). The last constraint in (2.11) eliminates the rigid translation degrees of freedom that still remain after enforcing Y-periodicity.

Intuitively, solving these six equations corresponds to probing the base cell with six constant strain stretches to determine how the microstructure reacts.

The homogenized elasticity tensor components are finally just the average over Y of the stress components corresponding to e^{kl}:

$$C^H_{ijkl} = \frac{1}{|Y|}\int_\omega C^{\text{base}}_{ijpq}[\varepsilon(\mathbf{w}^{kl}) + e^{kl}]_{pq}\,\mathrm{d}\mathbf{y}. \tag{2.12}$$

It is worth noting that C^H does not depend at all on the macroscopic details (shape Ω, or force terms f and g).

Derivation by Asymptotic Expansion

We now give the standard, more rigorous derivation by a two-scale asymptotic expansion (though this approach is still heuristic since we must assume the existence of such an expansion). The derivation here follows roughly that in [2] with some clarifications and added intuition.

This two-scale approach is motivated by inspecting the deformation of a periodically tiled object, noticing the emergence of behavior at two distinct length scales (Figure 2.2). There is a high-frequency fluctuation—at the same scale as the microstructure—superimposed atop a smooth deformation. The two-scale approach will allow us to determine the smooth "macroscopic" behavior as $\epsilon \to 0$. It begins with us guessing that the equilibrium displacement for a small ϵ can written as an asymptotic expansion:

$$\mathbf{u}^\epsilon(\mathbf{x}) = \sum_{p=0}^{\infty} \epsilon^p \mathbf{u}_p\left(\mathbf{x}, \frac{\mathbf{x}}{\epsilon}\right),$$

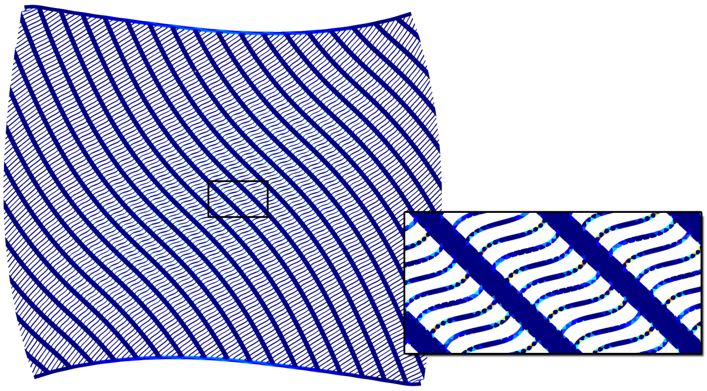


Figure 2.2: Separation of macroscopic and microscopic length scales: notice that the deformation appears smooth (left), but when we zoom in closely, we see a periodic fluctuation in the displacement field. The original, undeformed object was a square, and a uniform compression was applied to its top and bottom edges; if it weren't for the microstructure, the square would contract uniformly in the vertical direction (an isotropic material with Poisson's ratio 0 was used).

with $\mathbf{u}_p(\mathbf{x}, \mathbf{y})$ constrained to be Y-periodic in $\mathbf{y}$. Each function $\mathbf{u}_p$ distinctly separates its spatial dependence at the macroscopic and microscopic length scales. The periodicity constraint formalizes our intuition that the fluctuations are periodic (from Figure 2.2).

We plug this guess into the force balance equation (2.9) and equate coefficients of ϵ^p on the left- and right-hand sides. We use the chain rule to compute:

$$\frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}x_j} = \frac{\partial \mathbf{u}_p}{\partial x_j} + \frac{1}{\epsilon}\frac{\partial \mathbf{u}_p}{\partial y_j}.$$

Letting $\nabla_{\mathbf{x}}$ and $\nabla_{\mathbf{y}}$ be the vectors of partial differentiation operators with respect to the macroscopic and microscopic variables respectively, the force balance equation becomes:

$$-\sum_{p=0}^{\infty} \epsilon^p \left(\nabla_{\mathbf{x}} + \frac{1}{\epsilon}\nabla_{\mathbf{y}}\right) \cdot \left[C(\mathbf{y}) : \left(\varepsilon_{\mathbf{x}}(\mathbf{u}_p) + \frac{1}{\epsilon}\varepsilon_{\mathbf{y}}(\mathbf{u}_p)\right)\right] = f(\mathbf{x}) \quad \text{in } \Omega,$$

where $\varepsilon_{\mathbf{x}}(\mathbf{u}) = \frac{1}{2}\left(\nabla_{\mathbf{x}}\mathbf{u} + (\nabla_{\mathbf{x}}\mathbf{u})^T\right)$ and $\varepsilon_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\left(\nabla_{\mathbf{y}}\mathbf{u} + (\nabla_{\mathbf{y}}\mathbf{u})^T\right)$ are the microscopic and macroscopic strain operators, respectively. Expanding, we find:

$$\begin{aligned}
-\sum_{p=0}^{\infty}\Big(&\epsilon^p \nabla_{\mathbf{x}} \cdot [C(\mathbf{y}) : \varepsilon_{\mathbf{x}}(\mathbf{u}_p)] \\
&+ \epsilon^{p-1}\left(\nabla_{\mathbf{x}} \cdot [C(\mathbf{y}) : \varepsilon_{\mathbf{y}}(\mathbf{u}_p)] + \nabla_{\mathbf{y}} \cdot [C(\mathbf{y}) : \varepsilon_{\mathbf{x}}(\mathbf{u}_p)]\right) \\
&+ \epsilon^{p-2}\nabla_{\mathbf{y}} \cdot [C(\mathbf{y}) : \varepsilon_{\mathbf{y}}(\mathbf{u}_p)]\Big) = f(\mathbf{x}),
\end{aligned}$$

and by equating powers of ϵ, one obtains an infinite system of equations.

As one might suspect, $\mathbf{u}_0$ ends up being the macroscopic displacement, and we derive the periodic homogenization equations by considering only the equations in which it appears. These are just the three lowest order equations:

$$\begin{aligned}
\epsilon^{-2}: &\quad -\nabla_{\mathbf{y}} \cdot [C : \varepsilon_{\mathbf{y}}(\mathbf{u}_0)] = 0 && (2.13)\\
\epsilon^{-1}: &\quad -\nabla_{\mathbf{y}} \cdot [C : \varepsilon_{\mathbf{y}}(\mathbf{u}_1)] - \nabla_{\mathbf{x}} \cdot [C : \varepsilon_{\mathbf{y}}(\mathbf{u}_0)] - \nabla_{\mathbf{y}} \cdot [C : \varepsilon_{\mathbf{x}}(\mathbf{u}_0)] = 0 && (2.14)\\
\epsilon^{0}: &\quad -\nabla_{\mathbf{y}} \cdot [C : \varepsilon_{\mathbf{y}}(\mathbf{u}_2)] - \nabla_{\mathbf{x}} \cdot [C : \varepsilon_{\mathbf{y}}(\mathbf{u}_1)] - \nabla_{\mathbf{y}} \cdot [C : \varepsilon_{\mathbf{x}}(\mathbf{u}_1)] \\
&\qquad - \nabla_{\mathbf{x}} \cdot [C : \varepsilon_{\mathbf{x}}(\mathbf{u}_0)] = f(\mathbf{x}). && (2.15)
\end{aligned}$$

We notice immediately that a function of the form $\mathbf{u}_0(\mathbf{x}, \mathbf{y}) = \mathbf{u}(\mathbf{x})$ (independent of $\mathbf{y}$) satisfies (2.13). The Fredholm alternative proves this solution is unique [2, Lemma 2.3.21], but this is intuitive: (2.13) is a force balance equation for $\mathbf{u}_0$ in each instance of the periodic cell (i.e., at each point $\mathbf{x}$). The periodicity requirement pins down the (infinitesimal) rotational degrees of freedom of $\mathbf{u}_0$, and all that remains is the translational degree of freedom: a function $\mathbf{u}(\mathbf{x})$ independent of $\mathbf{y}$. This $\mathbf{u}(\mathbf{x})$ ends up being the macroscopic displacement, as we will soon see.

Plugging in $\mathbf{u}_0 = \mathbf{u}(\mathbf{x})$ simplifies (2.14) to:

$$-\nabla_{\mathbf{y}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})]) = 0, \qquad (2.16)$$

where we used linearity to collect the strain terms. For fixed $\mathbf{u}$, this equation yields a unique solution for $\mathbf{u}_1(\mathbf{x}, \mathbf{y})$ at each point $\mathbf{x}$ up to a constant. Notice also that it is linear in $\varepsilon_{\mathbf{x}}(\mathbf{u})$, so if we find $\mathbf{u}_1$ for each of 6 sample $\mathbf{u}$'s carefully chosen to have (macroscopic) strains spanning the 6-dimensional space of strain tensors, then we can find $\mathbf{u}_1$ for any $\mathbf{u}$ by superposition. Notice that this equation is identical to (2.10), solving for the microscopic fluctuation placing the structure in static equilibrium under constant strain $\varepsilon(\mathbf{u})$.

Finally, we consider the ϵ^0 equation:

$$-\nabla_{\mathbf{y}} \cdot \Big(C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_2) + \varepsilon_{\mathbf{x}}(\mathbf{u}_1)]\Big) = \nabla_{\mathbf{x}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})]) + f$$

Integrating both sides of this equation over Y and applying the divergence theorem on the left:

$$-\int_{\partial Y} \Big(C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_2) + \varepsilon_{\mathbf{x}}(\mathbf{u}_1)]\Big) \hat{\mathbf{n}} \, \mathrm{d}A(\mathbf{y}) = \int_Y \nabla_{\mathbf{x}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})]) + f \, \mathrm{d}\mathbf{y}$$

Periodicity of $C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_2) + \varepsilon_{\mathbf{x}}(\mathbf{u}_1)]$ (C, $\mathbf{u}_1$, and $\mathbf{u}_2$ are periodic in $\mathbf{y}$) means that the left-hand side is 0. Thus, for this equation to have a solution, we must have:

$$0 = \int_Y \nabla_{\mathbf{x}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})]) + f \, \mathrm{d}\mathbf{y}$$

Exchanging the $\mathbf{x}$ divergence with the $\mathbf{y}$ integration, and pulling out f (constant wrt. $\mathbf{y}$) this equation becomes:

$$-\nabla_{\mathbf{x}} \cdot \left(\int_Y C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})] \, \mathrm{d}\mathbf{y}\right) = |Y| f \tag{2.17}$$

This is beginning to look like simply another force balance equation! Recall that $\mathbf{u}_1$, up to a constant, depends linearly on $\varepsilon_{\mathbf{x}}(\mathbf{u})$ (see (2.16)), meaning that "$\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})$" is a linear function of $\varepsilon_{\mathbf{x}}(\mathbf{u})$. We can express this relationship with a rank 4 tensor G that maps macroscopic strain to microscopic strain at $\mathbf{y}$. This allows us to simplify:

$$C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})] = C(\mathbf{y}) : G(\mathbf{y}) : \varepsilon_{\mathbf{x}}(\mathbf{u}) \tag{2.18}$$

Using this in (2.17) and pulling $\varepsilon_{\mathbf{x}}(\mathbf{u})$ (constant wrt. $\mathbf{y}$) outside the integral, we arrive at:

$$-\nabla_{\mathbf{x}} \cdot \left(\frac{1}{|Y|} \int_Y C(\mathbf{y}) : G(\mathbf{y}) \, \mathrm{d}\mathbf{y} : \varepsilon_{\mathbf{x}}(\mathbf{u}) \right) = f.$$

Setting $C^H = \frac{1}{|Y|} \int_Y C(\mathbf{y}) : G(\mathbf{y}) \, \mathrm{d}\mathbf{y}$, we finally obtain our homogenized force balance equation:

$$-\nabla_{\mathbf{x}} \cdot [C^H : \varepsilon_{\mathbf{x}}(\mathbf{u})] = \mathbf{f} \quad \text{in } \Omega,$$

agreeing with (2.9).

Cell Problems

All that remains is to use (2.16) to determine rank 4 tensor G appearing in C^H. First we introduce the canonical basis for symmetric rank 2 tensors:

$$e^{ij} = \frac{1}{2} \left(\mathbf{e}_i \otimes \mathbf{e}_j + \mathbf{e}_j \otimes \mathbf{e}_i \right).$$

Then we can trivially expand the macroscopic strain at any point in this basis:

$$\varepsilon_{\mathbf{x}}(\mathbf{u}) = e^{kl} [\varepsilon_{\mathbf{x}}(\mathbf{u})]_{kl}$$

We now take advantage of linearity to state that if Y-periodic $\mathbf{w}^{kl}(\mathbf{y})$ solves (2.16) for $\varepsilon_{\mathbf{x}}(\mathbf{u}) = e^{kl}$:

$$-\nabla_{\mathbf{y}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{w}^{kl}) + e^{kl}]) = \mathbf{0}, \tag{2.19}$$

then $\mathbf{u}_1 = \mathbf{w}^{kl}[\varepsilon_{\mathbf{x}}(\mathbf{u})]_{kl}$ is a solution to (2.16) for arbitrary $\mathbf{u}$:

$$-\nabla_{\mathbf{y}} \cdot \Big(C(\mathbf{y}) : \big[\varepsilon_{\mathbf{y}}(\mathbf{u}_1) + \varepsilon_{\mathbf{x}}(\mathbf{u})\big]\Big) = \Big(- \nabla_{\mathbf{y}} \cdot (C(\mathbf{y}) : \big[\varepsilon_{\mathbf{y}}(\mathbf{w}^{kl}) + e^{kl}\big])\Big)[\varepsilon_{\mathbf{x}}(\mathbf{u})]_{kl} = 0.$$

Since this solution $\mathbf{u}_1$ is unique up to a constant wrt. $\mathbf{y}$, we know

$$\varepsilon_{\mathbf{y}}(\mathbf{u}_1) = \varepsilon_{\mathbf{y}}(\mathbf{w}^{kl})[\varepsilon_{\mathbf{x}}(\mathbf{u})]_{kl},$$

which in turn means that macro-to-micro tensor $G_{ijkl}(\mathbf{y}) = [\varepsilon_{\mathbf{y}}(\mathbf{w}^{kl}) + e^{kl}]_{ij}$. Plugging this into the equation for the homogenized elasticity coefficients, we get in index notation:

$$C^H_{ijkl} = \frac{1}{|Y|}\int_Y C_{ijpq}(\mathbf{y})G_{pqkl}(\mathbf{y})\,\mathrm{d}\mathbf{y} = \frac{1}{|Y|}\int_Y C_{ijpq}(\mathbf{y})[\varepsilon_{\mathbf{y}}(\mathbf{w}^{kl}) + e^{kl}]_{pq}(\mathbf{y})\,\mathrm{d}\mathbf{y}. \qquad (2.20)$$

Thus, once we know each $\mathbf{w}^{ij}$, we can compute the homogenized elasticity tensor with a simple integration over the base cell. We find these by solving the 6 cell problems:

$$\begin{aligned} -\nabla_{\mathbf{y}} \cdot (C(\mathbf{y}) : [\varepsilon_{\mathbf{y}}(\mathbf{w}^{ij}) + e^{ij}]) = 0 \quad &\text{in } Y \\ \mathbf{w}^{ij}(\mathbf{y}) \ Y\text{-periodic} \\ \int_\omega \mathbf{w}^{ij}(\mathbf{y})\,\mathrm{d}\mathbf{y} = 0, \end{aligned} \qquad (2.21)$$

one for each canonical basis tensor, e^{ij}. The last constraint pins down the remaining translational degree of freedom; since we only care about strain $\varepsilon_{\mathbf{y}}(\mathbf{w}^{ij})$, we can arbitrarily choose to enforce an average displacement of 0 over the microstructure geometry.

Writing (2.21), (2.20) in terms of ω using the fact elasticity tensor $C(\mathbf{y})$ vanishes outside ω for our solid-void microstructures, we arrive at the formulas we derived intuitively (2.11), (2.12).

2.3.2 Homogenized Tensor's Energy Form

From here on, we drop the $\mathbf{y}$ subscripts, as only microscopic variables remain. The homogenized elasticity tensor can be written in the following "energy-like" form, which turns out to be easier to differentiate:

$$C^H_{ijkl} = \frac{1}{|Y|} \int_\omega (\varepsilon(\mathbf{w}^{ij}) + e^{ij}) : C^{\text{base}} : (\varepsilon(\mathbf{w}^{kl}) + e^{kl}) \, \mathrm{d}\mathbf{y}. \tag{2.22}$$

This means that when we use the homogenized elasticity tensor to compute the elastic energy density stored at point $\mathbf{x}$ by some macroscopic strain $\bar{e}$:

$$\bar{e} : C^H : \bar{e} = \frac{1}{|Y|} \int_\omega \bar{e}_{ij} (\varepsilon(\mathbf{w}^{ij}) + e^{ij}) : C^{\text{base}} : (\varepsilon(\mathbf{w}^{kl}) + e^{kl}) \bar{e}_{kl} \, \mathrm{d}\mathbf{y},$$

what we get is the average energy density stored by the true microscopic deformation.

To prove (2.22), we notice that the weak form of cell problem kl is:

$$\int_\omega \varepsilon(\phi) : C^{\text{base}} : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \, \mathrm{d}\mathbf{y} = 0 \quad \forall \phi, \tag{2.23}$$

(constraining the trial/test functions to be periodic with zero mean). Since $\mathbf{w}^{ij}$ is in the space of (trial and) test functions, the $\varepsilon(\mathbf{w}^{ij})$ term of (2.22) right-hand side vanishes, leaving:

$$\frac{1}{|Y|} \int_\omega (e^{ij} : C^{\text{base}} : (\varepsilon(\mathbf{w}^{kl}) + e^{kl}) \, \mathrm{d}\mathbf{y} = \left[\frac{1}{|Y|} \int_\omega C^{\text{base}} : (\varepsilon(\mathbf{w}^{kl}) + e^{kl}) \, \mathrm{d}\mathbf{y} \right]_{ij} = C^H_{ijkl}.$$

Chapter 3

Printable Microstructure Design for Isotropic Elastic Properties

This chapter, based on publication [75], introduces a method to design 3D printable microstructures emulating isotropic material properties and implements a design pipeline employing them to achieve deformation goals in macroscopic objects. This project was joint work with Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. I contributed the periodic homogenization and shape optimization parts, as well as the finite element library used throughout. I also generalized and implemented the material optimization algorithm (originally proposed by Qingnan Zhou).

Section 3.8 on material optimization has been expanded in this thesis, discussing the problem in a more general setting and proving the convergence of a local-global material optimization closely related to the original algorithm we proposed in [75].

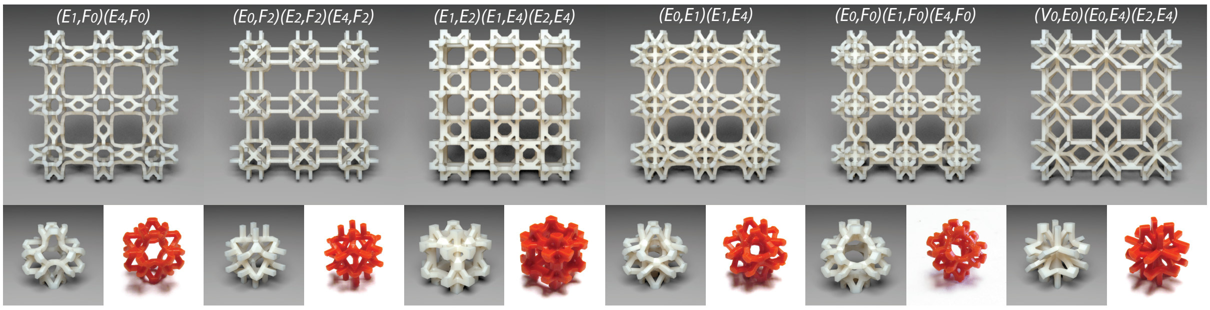

$(E_1,F_0)(E_4,F_0)$
$(E_0,F_2)(E_2,F_2)(E_4,F_2)$
$(E_1,E_2)(E_1,E_4)(E_2,E_4)$
$(E_0,E_1)(E_1,E_4)$
$(E_0,F_0)(E_1,F_0)(E_4,F_0)$
$(V_0,E_0)(E_0,E_4)(E_2,E_4)$

Figure 3.1: Six basic elastic textures are used to obtain a large range of homogenized isotropic material properties. A 3 × 3 × 1 tiling of each pattern is shown, along with rendered (left) and fabricated (right) cell geometry below. The naming convention is explained in Section 3.4.

This work introduces *elastic textures*: a set of parametric, tileable, printable, cubic patterns achieving a broad range of isotropic elastic material properties. The softest pattern is over a thousand times softer than the stiffest, and the Poisson's ratios range from below zero to nearly 0.5. Using a combinatorial search over topologies followed by shape optimization, we explore a wide space of truss-like, symmetric 3D patterns to obtain a small family. This pattern family can be printed without internal support structure on a single-material 3D printer and can be used to fabricate objects with prescribed mechanical behavior. The family can be extended easily to create anisotropic patterns with target orthotropic properties. We demonstrate that our elastic textures are able to achieve a user-supplied varying material property distribution. We also present a material optimization algorithm to choose material properties at each point within an object to best fit a target deformation under a prescribed scenario. We show that, by fabricating these spatially varying materials with elastic textures, the desired behavior is achieved.

3.1 Introduction

Rapid advances in the accessibility of additive fabrication has a significant impact on how manufacturable geometric models are constructed. A key distinctive feature of common additive fabrication technologies is that the cost and time of production is practically uncorrelated with structural complexity: in fact, a complex structure using less material may be both cheaper and faster to produce.

Complex structures, aside from potentially reducing costs, open up many new possibilities, in particular for manufacturing deformable objects. By varying a small-scale structure, one can adjust a variety of material properties, from elasticity to permeability. Importantly, these properties can be varied nearly continuously over the object, something that is not commonly done in traditional processes. As it was observed in prior work, this opens up many new possibilities for object behavior.

Small-scale structures present a set of new design challenges: in all but the simplest cases, these are hard or impossible to design by hand to meet specific goals. At the same time, computational optimization of fine-scale variable structure over a whole object, even of moderate size, can easily result in numerically difficult topology and shape optimization problems with millions of variables.

In this paper, we describe *elastic volumetric textures*, a library of tileable parameterized 3D small-scale structures that can be used to control the elastic material properties of an object. Applying such textures to a hex mesh with target material properties specified per element is similar to using dithering to achieve a continuous variation of brightness or color.

In a sense, almost *all* material properties owe themselves to small-scale structures at the molecular or crystal level, and a large body of work in nanoscience aims to control material properties precisely by structure design. These works must accommodate constraints imposed by the specific properties of the elements and molecules used, the need for self-assembly, and other considerations.

Our focus is on larger-scale structures, which can be manufactured using existing 3D printing technology. With feature sizes at the scale of 10μm-100μm, these are well described by conventional elasticity theory. While this type of structure was also extensively studied, typically this was in the context of a specific problem, such as optimizing strength for a given material volume fraction. Our goal is to maximize the range of effective material properties that can be obtained using a single material by varying the structure.

We consider variable-thickness *truss-like structures*—i.e. structures composed of connected bars—as these cover a considerable range of properties on the one hand, and on the other hand, allow us to work with a relatively small number of parameters. We present a method for building a dictionary of structures that cover a large space of material properties. These structures are *tileable*, which makes it possible to vary material properties across an object, and *printable.*

We demonstrate that elastic volumetric textures allow one to control the deformation behavior of objects, either by painting material properties directly or by a two-stage *shape optimization* procedure, involving solving for variable continuous properties then approximating them using our texture dictionary. We validate our results by measuring samples for different choices of parameters and topologies and by demonstrating the deformation behavior of objects fabricated with spatially varying structures.

3.2 Related Work

Microstructure design and optimization. There is a huge literature on theoretical studies of effective moduli of composites (our periodic structures are an extreme example of a composite combining a material with void). Recent monographs include [29, 68, 100]. Much of the literature focuses on identifying microstructures with *extremal* effective behavior, i.e., with effective elasticity properties at the boundary of the achievable zone for a given class of composites [3, 26, 68]. Many classes of extremal structures were described (see, e.g., [23]), however most of these classes—e.g. sequentially laminated microstructures [9] and microstructures based on inclusions [38, 64]—are either difficult or impossible to manufacture at this time. Interchangeable composites and other structures were found that maximize simultaneously, e.g., the bulk modulus and permeability [40] or electrical conductance [102, 103, 101], but these designs are of limited use for tailoring elastic behavior.

The closest work to ours is [93], which constructs truss microstructures with prescribed elasticity tensors. It starts with a full "ground structure" containing about 2000 candidate members, then optimizes the members' thicknesses *but not offsets* to obtain a microstructure period cell whose homogenized properties (computed using a truss model) match the desired properties. Neither tileability of structures for different parameters nor printability can be guaranteed. We discuss the differences in greater detail at the end of Section 3.4. Further exploration of periodic structures of this type was done more recently in [27], comparing different methods for optimizing these structures.

A number of microstructures were obtained using various types of *topology optimization*, which was originally designed for global structure optimization. In the case of microstructure design, these methods look for a periodic structure minimizing, e.g., compliance for a fixed total volume fraction. The result is normally a single-scale structure, with scale controlled by the resolution of the simulation grid or other types of regularization. Important methods proposed for solving these problems include solid isotropic material with penalization (SIMP) and rational approximation of material properties (RAMP) [14, 15, 73]. [83] demonstrated design of isotropic materials maximizing bulk modulus.

Topology optimization offers more flexibility in the choice of structure, but it requires a relatively expensive optimization for each specific problem. The ability to undergo topological transitions under continuous parameter changes is both a strength, as it allows exploration of a broader space of structures, and a weakness, as it considerably complicates design of parametric families satisfying printability and tileability constraints, which motivates our approach.

Microstructure fabrication. Several groups focusing on additive fabrication have recently obtained encouraging results. In particular, materials previously thought to be unmanufacturable were produced and behave as expected. Notably, the work of Hollister and collaborators [62, 61, 47, 51] in the context of bone scaffold design and fusion cage design demonstrated the use of optimized microstructures. The possibility of manufacturing auxetic (negative Poisson's ratio) materials was demonstrated in [39], and in [89, 22, 7].

The idea of fabricating tileable structures with varying properties also appears in [46] in which the authors discuss "digital materials," as composed of a set of discrete voxels with predefined shapes that can be connected. Similarly, a building-block based approach was also used in the context of bio-printing [69], where the authors use spheroids of living materials with evolving and controllable composition, varying material and biological properties in time.

Compliant mechanisms. The material optimization method that we present solves a similar problem to that of compliant mechanism design. [15] reviews several existing approaches to designing mechanisms that maximize mechanical advantage/output deflection or tune an output displacement to a particular path. These approaches have little control over the resulting structure's macroscopic shape, whereas tuning deformation behavior using our microstructure approach creates a "mechanism" that still looks like the input shape.

Fabrication and computer graphics. A broad variety of fabrication-related work has been done in the computer graphics community. Several techniques have been proposed to design paper craft objects [70], plush objects [71], and objects made of interlocking planar slices [28, 87, 45]. Other techniques use geometric techniques to change surface appearance by synthesizing surface microgeometry [108] or changing the shape to generate custom target caustics [88].

Another close work to ours, [17], introduces an optimization process to find the best combination of stacked layers to satisfy an input deformation, enabling fabrication of objects with complex heterogeneous materials using multi-material 3D printers. Our work can be viewed as complementary, focusing on the design of structures that can be, e.g., used as a part of deformation behavior design; our material optimization method provides an alternative to the method in that paper. In [94], multi-material printing and discrete material optimization is used in a similar way on complex characters to achieve desired deformations with actuation. Our elastic textures can be viewed as a tool for solving this type of problem. Our structures also can be employed in systems like [25] and [107].

Homogenization. A central tool in our work, homogenization was used in graphics for reducing complexity of physical models in [52], finding the constitutive parameters of a low resolution discretization that best approximates the behavior of the original higher complexity material. The periodic homogenization method that we use is based on the one described in [3].

3.3 Overview and Main Results

In this section, we describe our overall approach, visualized in Figure 3.2, and a specific set of patterns that we have obtained.

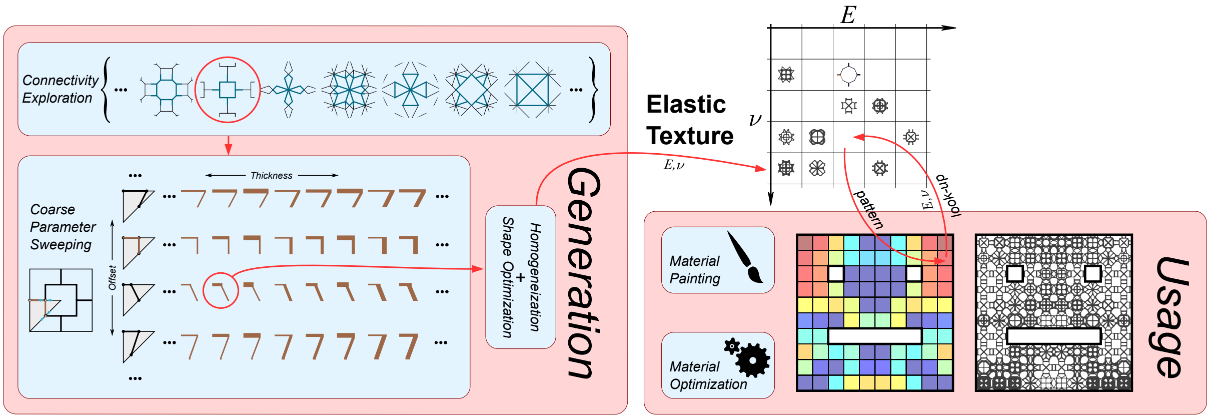


Figure 3.2: Overview of elastic texture generation and use.

Problem. The general problem we solve can be formulated as follows: *for each tensor C from a given range of elasticity tensors, and an isotropic base material with Young's modulus E^b and Poisson's ratio ν^b, find a structure made out of the base material in a unit cubic cell, such that if the cell is infinitely tiled in space, the resulting homogeneous material has elasticity tensor C.*

As discussed in the introduction, we aim to construct a family of patterns that are printable and tileable to enable creation of variable material properties.

Printability is heavily dependent on the choice of technology. We focus on printability criteria related to stereolithography, the most accurate 3D printing method available at this time, but our approach can be easily modified to handle other technologies.

As the printing process proceeds layer-by-layer, we assume that the structure is defined with respect to a fixed coordinate system X, Y, and Z aligned with the printer, with Z being vertical. The (idealized) printability criteria that we use are:

1. There are no enclosed voids.

2. For any point of the structure, the extent covered by the structure in the X, Y, and Z directions from the point are above a printability threshold $d_{\min}$.

3. Every point of the pattern is supported: for every XY slice, all connected compo-

nents of the slice have at least one point connected to lower points in the structure by a segment contained in the structure. While this condition does not prevent long horizontal bars supported at single points, which can be difficult to print, we have found it sufficient in practice for pattern sizes up to 10mm and a $d_{\min}$ of 0.3mm.

We also make our primary goal to generate periodic structures with *isotropic* homogenized properties. Such patterns have the elasticity tensor C defined by two parameters, Young's modulus E and Poisson's ratio ν, and its inverse, compliance tensor S, has the (Voigt notation) form

$$S = \frac{1}{E}\begin{pmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{pmatrix}. \qquad (3.1)$$

Expressing in terms of the shear modulus, $G = E/(2(1+\nu))$, the last three diagonal terms of S are simply $1/G$.

While for many tasks anisotropic materials are either sufficient or preferable, periodic structures with isotropic homogenized properties are easiest to use, as cell orientation is decoupled from material properties. In addition, once an isotropic starting point is obtained, it is easy to obtain a controlled anisotropic behavior.

Searching the space of all possible structures in a cell, even at a finite resolution, is an impossible task. Instead, we choose a space of structures with a limited but sufficiently large set of parameters, that can be optimized to achieve specific material properties.

Truss-like structures. We focus on truss-like structures (*patterns*) as shown in Figure 3.1, consisting of bars of different thicknesses connecting a set of nodes in the cell.

Unlike real truss structures, the connections between bars are not pin joints, and flexural rigidity at the nodes plays a major role.

This particular space of structures is motivated by several considerations. First, the space is known to contain both very stiff and very weak patterns, providing a broad range of behaviors. Second, tileability and printability requirements yield specific geometric conditions, expressed mostly as constraints on the structure's geometry. For example, the requirement of no enclosed voids is automatically satisfied if the frame structure has no self-intersections; the bound on extents can be obtained by bounding the thickness from below; and the support condition is easily formulated as a constraint on node positions. These conditions are detailed in Section 3.4.

Symmetry considerations, as well as restrictions on the number and placement of nodes, yield a space of patterns parametrized by their set of edges connecting some subset of the 15 candidate nodes we define on a tetrahedron (their *topology*), thicknesses of these edges, and offsets of the nodes from their default positions. This space is still very large, and we explore it using both topology and geometry searches, described in Section 3.4. These rely fundamentally on the homogenization and shape optimization procedures described in Sections 2.3 and 3.6.

Resulting family. The search procedure's final result is shown in Figure 3.3; the six pattern topologies themselves are illustrated in Figure 3.1. The complex boundaries of the (E, ν) regions arise from the multiple types of geometric constraints enforcing printability. On the right, Figure 3.3 shows some patterns with topology "(E1,E2)(E1,E4)(E2,E4)."

The family of topologies covers a large range of Young's moduli, with a largest-to-smallest Young's modulus ratio of 1800, and a sizable range of Poisson's ratios, -0.16 to 0.48.

We note that the range of negative Poisson's ratios is somewhat limited, while on the high end, we are able to achieve ratios close to the theoretical maximum. This observation is consistent with [93]: while it is relatively easy to obtain more extreme negative Poisson's ratios for patterns with cubic symmetry but with shear modulus too low for isotropy, the isotropy requirements restrict the range. Printability constraints restrict it further.

Quite remarkably, four out of the six topologies can be transformed into each other by simple operations (single vertex splits, addition of cross-shaped supports connecting some nodes). The other two are also related to each other by a simple transform, but are not related to the first sequence.

We do not claim that the proposed family is in any sense optimal. It is most likely possible to extend the coverage or to cover the same domain with fewer topologies. However, the presented set is already quite useful for controlling material properties, as the examples of Section 3.7 show.

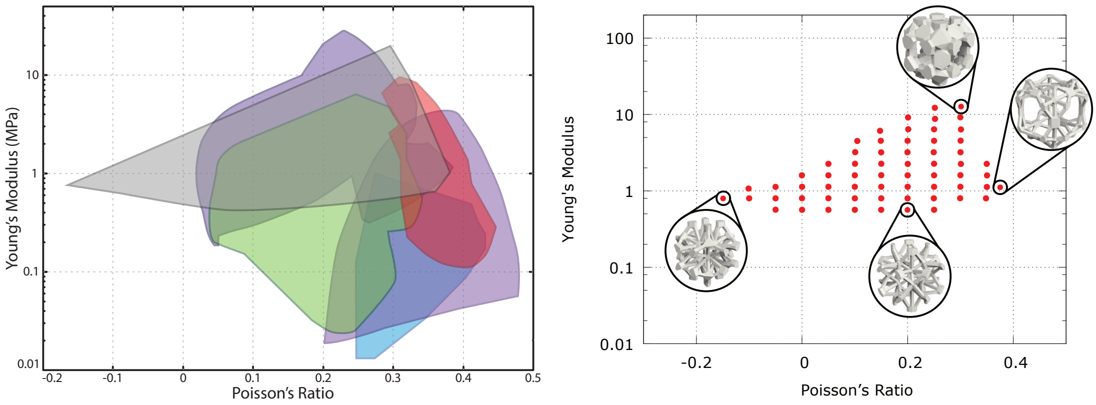


Figure 3.3: Left: region of the (E, ν) space covered by the selected set of patterns. Each topology's coverage is shown in a different color. Right: Samples of the (E, ν) space reached by patterns with topology "(E1,E2)(E1,E4)(E2,E4)."

Accuracy. We fabricated eight patterns with different homogenized Young's moduli using the B9Creator SLA printer and tested their stiffness using the BOSE ElectroForce 3200 measurement system. The machine gradually compressed our samples in the Z direction between two compression plates and measured the displacement resulting from the applied force at each step. We used 6 × 6 × 2 tilings of 5mm cells for this test.

We note that our measured force/displacement slopes are roughly proportional to the homogenized Young's moduli (Figure 3.4a), implying that the measurements are consistent with some (unmeasured) base Young's modulus. The curvature seen could be explained partly by friction in the compression testing setup (Figure 3.4b). Another significant source of error is the inaccuracy of our B9Creator, which tends to thicken thin geometries.

We used a lower-accuracy setup to measure Poisson's ratio but still obtained reasonable agreement with homogenization (Figure 3.5). We compressed the microstructures in the Z direction between two lubricated metal blocks and manually measured the expansion/contraction in the X and Y directions. From these displacement measurements, we computed the X, Y, and Z strains and their ratios.

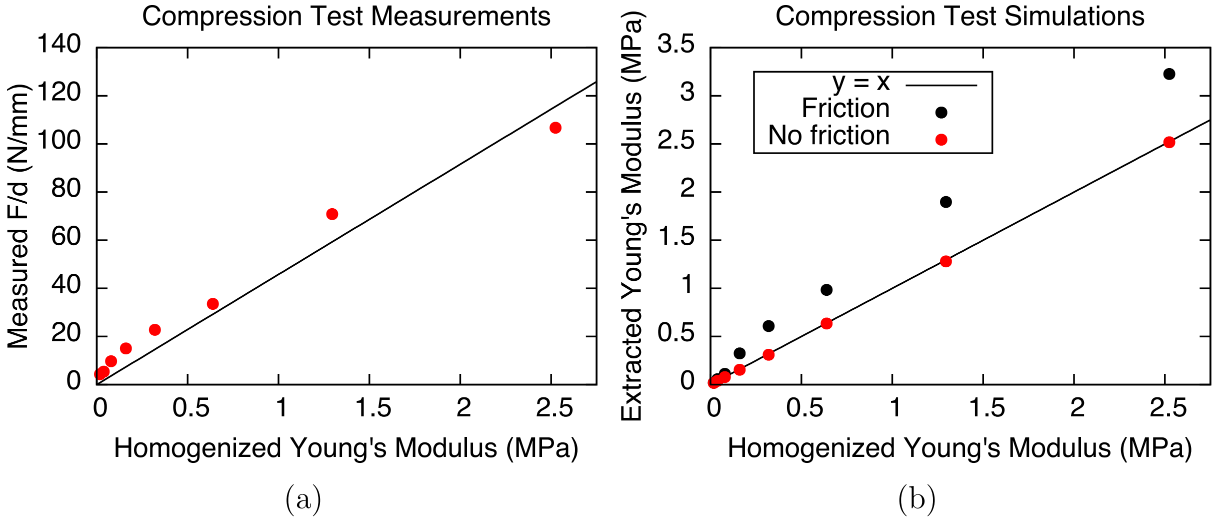


Figure 3.4: Compression test results for eight patterns with varying homogenized Young's moduli (6 × 6 × 2 tiling of 5mm cells). (a) Slopes extracted from the measured force vs. displacement curves along with a best-fit line through the origin. (b) Moduli extracted from simulated compression tests, with and without modeling compression plate friction. Without friction, the simulated test agrees with homogenization perfectly, but friction introduces error.

We also validated the patterns' isotropy by printing a block filled with a tiled pattern that was rotated by 45° around the Z axis and clipped (Figure 3.6). The measured

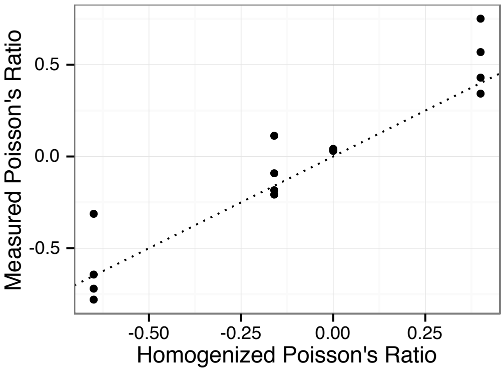


Figure 3.5: Poisson's ratios measured from 3 × 3 × 1 printed tilings of 10mm cells vs. homogenized properties. The $\nu = -0.67$ sample, outside our family's range, violates isotropy and printability constraints (we added support structure manually for this experiment).

effective Young's moduli in the rotated orientations were in good agreement with the unrotated orientation: a compression test in the X, Y, and Z directions extracted effective Young's moduli of 0.635MPa, 0.6293MPa, and 0.628MPa for the example shown.

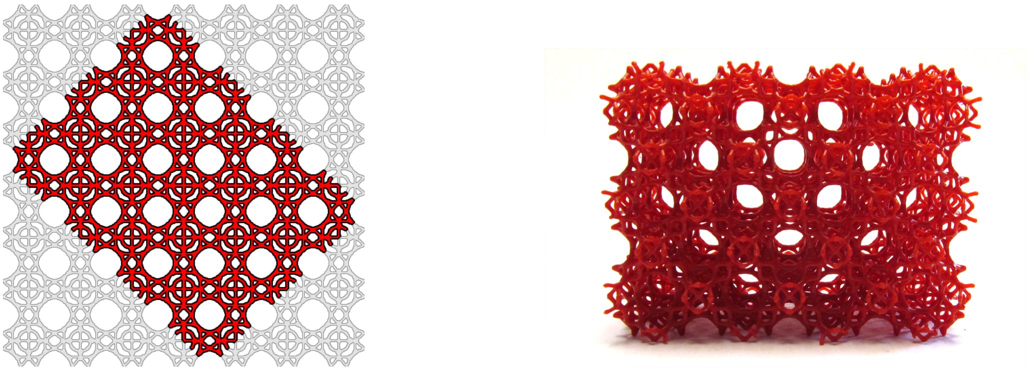


Figure 3.6: We extracted a 45° rotated rectangular block from a regularly tiled 10mm cell microstructure to test Young's modulus in non-axis aligned directions.

Base Material. We used estimated base material properties of $E^b = 200$MPa and $\nu^b = 0.35$ for all results. We estimated E^b using a three point bending test on rectangular bar samples, but the base Young's modulus can also be estimated from the microstructure compression test.

Using a different base Young's material would not qualitatively change our results, apart from making the patterns uniformly softer or stiffer. The homogenized Young's moduli depend linearly on E^b, and the Poisson's ratios are independent of E^b, so scaling the base modulus simply softens/stiffens every pattern by the same factor. In particular, for the displacement-based material optimization of Section 3.7, fabricating the solution with a different base Young's modulus maintains the same target deformation behavior (although the required force will change).

We note that our patterns are not very sensitive to moderate changes in the base Poisson's ratio. Changing from $\nu^b = 0.35$ to $\nu^b = 0.35 \pm 0.05$ results in a median relative change in Young's modulus of 0.683% (max: 1.88%) and a median absolute change in Poisson's ratio of 0.00229 (max: 0.0129) over all patterns. Since most additive fabrication materials fall within this range, we expect similar results for other printers and materials.

3.4 Searching for Patterns

In this section, we describe the class of patterns that we consider and the main steps of the search method.

Ground class of patterns. The topology of patterns is defined by a set of edges connecting nodes in the cube cell. We generate the geometric variations by adding *offsets* to the node positions and by changing edge *thicknesses*.

Motivated by the isotropy requirement, we constrain our search to patterns with cube symmetries, which are guaranteed to have the same Young's moduli and Poisson's ratios in every axis-aligned direction. That is, the compliance tensor has the form (3.1) *except* the last three diagonal entries $1/G$ may not equal $2(1+\nu)/E$. This yields an easy-to-check isotropy measure:

$$A = \frac{2(1+\nu)G}{E}, \tag{3.2}$$

which we use to identify isotropic patterns in our search. The symmetry will also dramatically reduce the space of pattern topologies to a tractable size after a few additional constraints are introduced. Note, however, that cube symmetry is not necessary for isotropy; other isotropic structures exist with, e.g., tet symmetry.

Consider the group of symmetries of a cube O_h, which includes reflections about three symmetry planes orthogonal to the X, Y, Z axes and the six planes orthogonal to the bisector of each pair of axes. By partitioning the cube according to these symmetry planes, we obtain 48 equal tetrahedra as in Figure 3.7a. O_h maps a single one of these tetrahedra to any other, so it is sufficient to define the nodes and edges of the pattern graph—as well as their offsets and thicknesses—on a single tetrahedron.

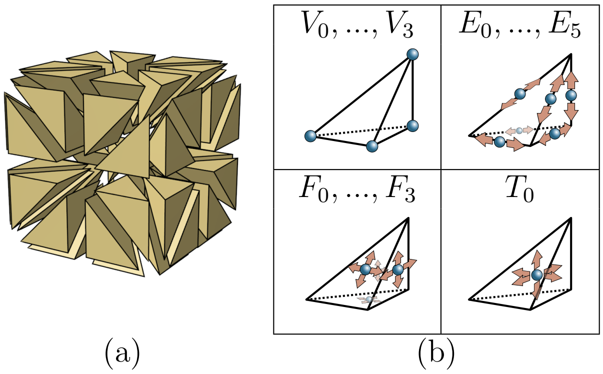


Figure 3.7: (a) The tetrahedral cube decomposition used to generate 3D patterns; (b) The 15 nodes defined on a tetrahedron together with their degrees of freedom.

We generate the different topological configurations by changing the connectivity between 15 nodes on a tetrahedron (see Figure 3.7a): vertex nodes $\{V_0, V_1, V_2, V_3\}$, edge nodes $\{E_0, E_1, E_2, E_3, E_4, E_5\}$, faces nodes $\{F_0, F_1, F_2, F_3\}$, and a single internal node, T_0. Configurations are named by their graphs' edge sets (see labels in Figure 3.1). Each node is constrained to stay on its respective simplex to preserve the topology, so vertex nodes are fixed, edge nodes have a single offset, and so on (Figure 3.7b).

Figure 3.8 shows an example topology colored by its node and edge orbits with respect to symmetry group O_h, and Figure 3.9 demonstrates the effects of the node offset and edge thickness parameters.

The space of possible connectivities, even after accounting for symmetries, is far too

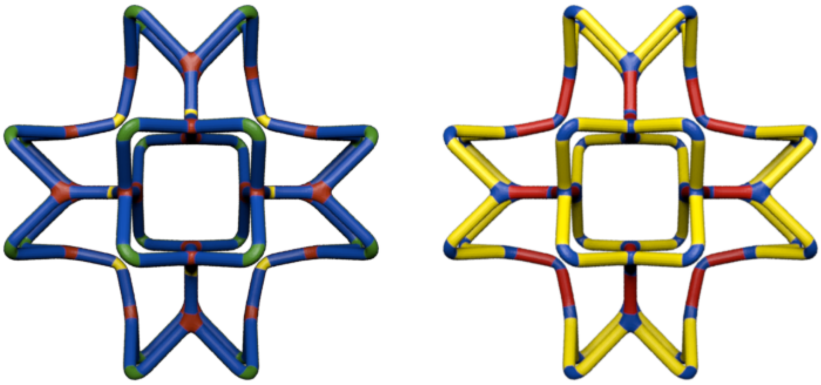


Figure 3.8: Symmetry orbits are colored with yellow, red and green. Left: vertex symmetry orbits. Right: edge symmetry orbits.

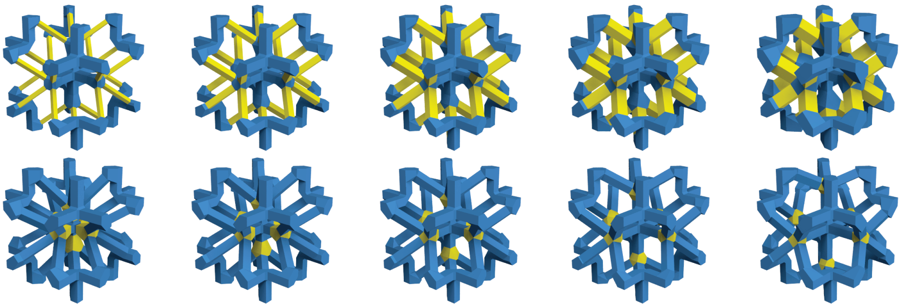


Figure 3.9: The results of varying the thickness (top) and offset (bottom) parameters of a particular pattern topology.

large to explore completely (on the order of 10^{32} configurations). We enforce the following constraints to reduce the space of patterns:

- **Connected**: the tiled pattern is a single connected component.
- **No coinciding edges**: no edge is contained within another. E.g., if graph edge (V_0, V_1) is chosen and E_0 is the midpoint node of the corresponding tet edge, graph edge (V_0, E_0) is forbidden since it overlaps the first for any offset.
- **No dangling edges**: every node has valence greater than 1.
- **Number of edges**: at most 3 graph edges per tetrahedron.
- **Max node valence**: node valences do not exceed 7.

Valences are computed on the graph after periodic tiling of the cube cell. The first two criteria reduce the space to 16221 topologies, and the remaining three to 1205 topologies.

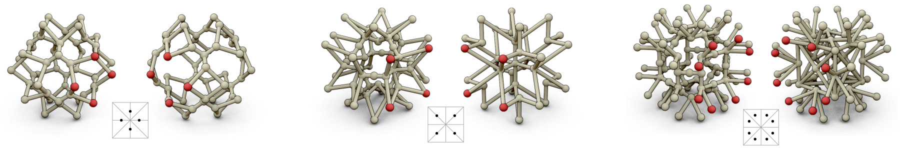


Figure 3.10: Two pattern topologies from each of three different families, shown with the families' *interfaces* (nodes on the cube cell faces).

Printability. For truss-like patterns, printability is affected by two main factors: the pattern graph structure and the edge thicknesses.

The first printability criterion can be defined on the nodes by considering their offset positions. We say that a node n_1 has supporting node n_2, if these are connected by an edge and n_1 is strictly above n_2. We say that n_1 and n_2 are at the same level if they have equal Z coordinates. A pattern is printable only if every connected set of nodes at one level has at least one supporting node.

Printability can be tested by a simple algorithm: we first mark as supported all nodes with a supporting node (considering periodicity). Then we propagate the front of supported nodes to neighbors at the same level. When this breadth first search terminates, the pattern is printable if and only if all nodes are marked as supported. The procedure is illustrated in Figure 3.11. We also note that this constraint can be expressed algebraically as a set of inequality constraints on the offset variables, which can be enforced by an optimization solver.

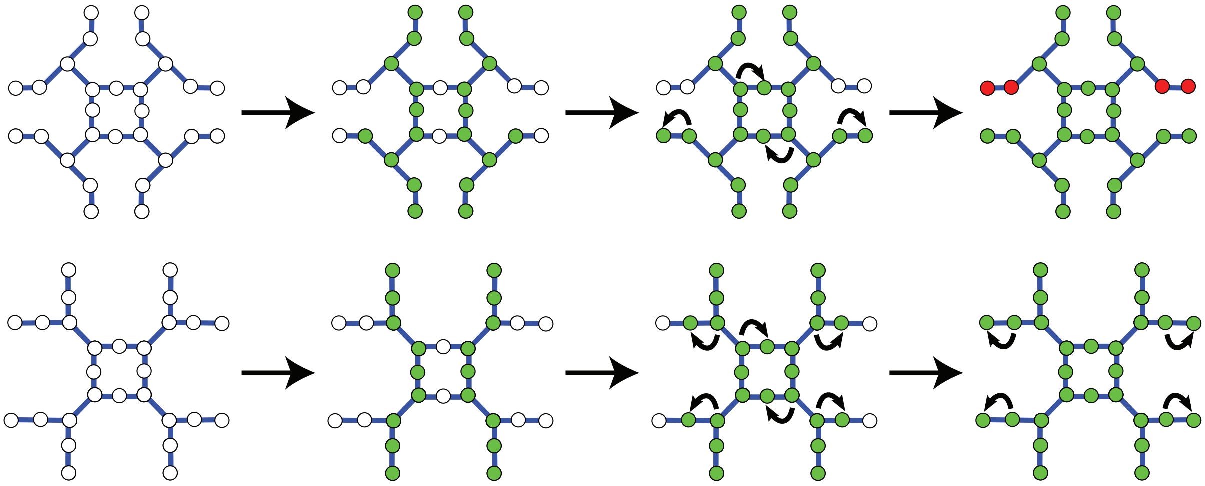


Figure 3.11: 2D examples of the printability detection algorithm. Vertices with supporting nodes are marked (green), then a breadth-first search extends the supported vertex front to horizontal neighbors. The remaining unmarked nodes are unsupported (red). Two cases are shown: unprintable (top) and printable (bottom).

Tileability. The tileability requirement means that all pattern topologies should belong to the same *family*, meaning topologies with the same set of nodes and edges appearing on the faces of the cube cell (Figure 3.10).

Searching the space of topologies. The goal of our search is to identify a family of pattern topologies that covers as much as possible of the (E, ν) space while satisfying the printability and tileability requirements.

The initial space consists of all pattern topologies satisfying the constraints on graph connectivity mentioned previously. We proceed in the following steps:

1. *Coarse geometry sweep.* Geometric variations are generated for each pattern topology by trying thicknesses of 0.3mm and 0.7mm and node offsets corresponding to barycentric coordinates of 0.2, 0.35, 0.5, 0.65, and 0.8 on the associated tetrahedron simplex. The resulting *printable* patterns are meshed, self-intersecting meshes are discarded, and the remaining patterns' effective elasticity parameters are computed using periodic homogenization (Section 2.3).

2. *Isotropy filtering.* We select a subset of the patterns closest to isotropic (we use a heuristic bound of $0.8 < A < 1.2$), which we consider promising candidates as starting points for optimizing pattern parameters to precisely match a range of isotropic elasticity tensors.

3. *Topological family selection.* At this point, we have a rough map of the area in (E, ν) space covered by our set of patterns. We obtain a rough estimate of each topology's coverage by taking the convex hull associated with its nearly isotropic geometric configurations. We manually pick the single family whose pattern topologies cover the largest region of (E, ν) based on these estimates.

4. *Selection of a minimal covering set of topologies.* For the selected family, we run a finer sweep of offsets and thicknesses, again filtering for printability, to compute a more precise estimate of the boundary of the (E, ν) domain that each pattern topology can cover. Among all topologies in the family, we selected 6 such that the union of their coverage areas contains most of the domain covered by the family.

5. *Lookup map construction.* Finally, using the shape optimization machinery of Section 3.6 and the initial nearly-isotropic points for each of the 6 topologies chosen, we optimize each patterns' parameters to reach a grid of isotropic elasticity tensors evenly spaced in $(\log(E), \nu)$.

Our procedure is similar to [93], but with several key differences. First, [93] uses a simplified truss model, whereas our method directly homogenizes and optimizes the printable geometry. Second, by using a full topology (including all possible edges between nodes in a ground structure) and permitting zero edge thickness in optimization, the work avoids the topology enumeration stage. As a side-effect, it cannot accommodate the lower thickness bounds or support criteria needed for printability. While a mixed-integer formulation like [67] could allow enforcement of d_{min} by introducing separate binary variables to disable members, this would involve a difficult mixed-integer nonlinear programming problem in our tensor-fitting setting. Finally, by introducing offset variables, our novel shape optimization approach enables much finer control of the elasticity tensors as the design is not limited to the discrete node positions of a ground structure.

3.5 From Patterns to Material Properties

We use periodic homogenization to find the homogenized elasticity tensor C^H of a microstructure, describing its effective material properties when fabricated at a small enough scale and periodically repeated to fill space.

This process involves probing the intinitely tiled microstructure by stretching it with the six canonical basis strain tensors and measuring its force response. This process is detailed in Section 2.3, and requires solving a periodic elasticity equation on the microstructure cell (2.11).

3.5.1 FEM implementation.

The cell problems (2.11) are solved numerically by a quadratic tetrahedral FEM discretization of ω. The piecewise linear integrand in (2.12) is integrated with exact quadrature.

Given the wire network of the microstructure, defining its topology (Section 3.4), a volume mesh is generated following the STRUT algorithm given in [41]:(i) A polygon is created around both ends of each segment. (ii) For each vertex, the convex hull of the nearby polygons and the vertex is constructed and the polygons themselves are removed from the hull. (iii) For each edge, the convex hull of its two polygons is constructed, and again the two polygons are removed from the hull. A tetrahedral volume mesh is finally created from the resulting closed surface.

The linear elasticity solver must support periodic boundary conditions, which requires the tetrahedral mesh to have an identical tessellation on the opposite periodic cell faces.

3.5.2 Convergence rate.

Remarkably, we have observed that the homogenized coefficients remain accurate when the microstructure varies across cells (with few or no repetitions). In our experiments, the deformation behavior of even very coarse tilings closely matches the homogenized behavior (Figure 3.12), and we expect similar agreement in general for smoothly varying loads.

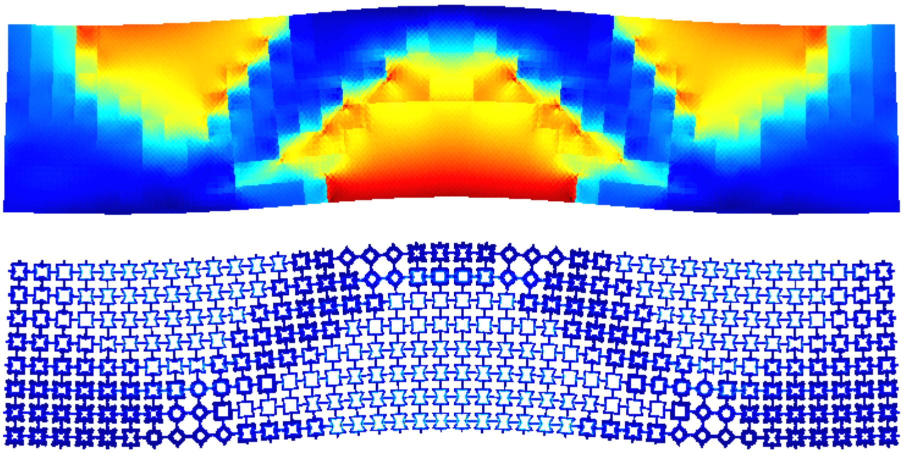


Figure 3.12: Deformation of an object with varying material properties per voxel, and the same object with the material in each voxel replaced with the corresponding pattern. The deformed objects are colored by max stress.

3.6 Optimizing Pattern Parameters

An essential step for creating a map of elastic textures is optimizing a pattern with fixed topology to match particular elasticity parameters. This is achieved using shape optimization with respect to the pattern parameters.

The optimization problem. Our goal is to minimize a functional, $J(\omega)$, measuring the difference between the homogenized elastic properties of the pattern and a target elasticity tensor C^*. We choose an objective that is suitable for designing material distributions with large deformations under moderate forces. The distance of compliance tensors, $S^H - S^*$, as opposed to elasticity tensors, is the better choice, since the strain for a constant stress is directly proportional to S^H, *not* C^H. In fact, minimizing the Frobenius norm of $S^H - S^*$ can be interpreted as a multi-objective least squares optimization to fit the displacements of two cubes—one filled with C^H and the other with C^*—under a set of axis-aligned stretching and shearing loads.

We choose this Frobenius norm as our functional:

$$J(\omega) = \frac{1}{2}\|S^H(\omega) - S^*\|_F^2, \tag{3.3}$$

which we optimize by varying the microstructure shape, ω.

In our case, the microstructure boundary $\partial\omega$ is determined by a small number of parameters $\mathbf{p}$, consisting of wire mesh node offsets and thicknesses. While the number of parameters is small, we still expect multiple solutions for minimizing $J(\omega)$ with respect to these parameters. A simple regularization term (staying close to the initial point of optimization) picks a unique solution. We note that instead some quantity of importance (e.g., weight) can be optimized as it is typically done (cf., [93]), adding another nonlinear term to the functional.

The derivative of the boundary $\partial\omega$ with respect to a parameter p_α is a vector field $\mathbf{v}_{p_\alpha}(\mathbf{y})$ defined at points $\mathbf{y}$ of $\partial\omega$, with $\mathbf{v}_{p_\alpha}(\mathbf{y})$ being the velocity of $\mathbf{y}$ if parameter p_α changes at unit speed.

Using parameters $\mathbf{p}$ as variables, the minimization problem can be written as

$$\underset{\text{admissible } \mathbf{p}}{\operatorname{argmin}} \; J(\mathbf{p}) \text{ where } J(\mathbf{p}) = \frac{1}{2}\|S^H(\mathbf{p}) - S^*\|_F^2. \tag{3.4}$$

The admissibility of parameters is determined by geometric intersection constraints and printability constraints.

The derivative of the objective function with respect to p_α can be obtained from $\mathbf{v}_{p_\alpha}$ using the chain rule:

$$\frac{\partial J}{\partial p_\alpha} = [S^H - S^*] : \frac{\partial S^H}{\partial p_\alpha} = [S^H - S^*] : \mathrm{d}S^H[\mathbf{v}_{p_\alpha}], \tag{3.5}$$

where $\mathrm{d}S^H[\mathbf{v}_{p_\alpha}]$ is the *shape derivative* of S^H applied to $\mathbf{v}_{p_\alpha}$.

Shape derivative of elasticity tensor The derivative of the microstructure's homogenized elasticity tensor in the direction of shape perturbation $\mathbf{v}$ is defined as the Gâteaux derivative,

$$\mathrm{d}C^H[\mathbf{v}] \overset{\text{def}}{=} \lim_{t \to 0} \frac{C^H(\omega(t, \mathbf{v})) - C^H(\omega)}{t}, \tag{3.6}$$

where $\omega(t, \mathbf{v}) \overset{\text{def}}{=} \{\mathbf{x} + t\mathbf{v} : \mathbf{x} \in \omega\}$.

We apply our differentiation to the homogenized tensor's "energy-like" expression, which due to its similarity to the self-adjoint compliance functional, has a surprisingly simple shape derivative.

Consider the perturbation of the shape's boundary, $\delta\omega$, caused by advecting the boundary infinitesimally in direction $\mathbf{v}$. The resulting variation of C^H_{ijkl} for $ij \neq kl$ ($ij = kl$ has an equivalent result by product rule) is:

$$\delta C^H_{ijkl} = \left\langle \frac{\partial C^H_{ijkl}}{\partial \omega}, \delta\omega \right\rangle + \left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{ij}}, \delta \mathbf{w}^{ij} \right\rangle + \left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{kl}}, \delta \mathbf{w}^{kl} \right\rangle .$$

We now evaluate the linear functional $\left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{ij}}, \cdot \right\rangle$ on an arbitrary admissible perturbation of $\mathbf{w}^{ij}$ (periodic and with no rigid translation component), ϕ:

$$\left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{ij}}, \phi \right\rangle \overset{\text{def}}{=} \lim_{h \to 0} \frac{\mathrm{d}}{\mathrm{d}h} \frac{1}{|Y|} \int_\omega (e^{ij} + e(\mathbf{w}^{ij} + h\phi)) : C^{\text{base}} : (e^{kl} + e(\mathbf{w}^{kl}))\, \mathrm{d}\mathbf{y}$$

Differentiating under the integral and using the linearity of strain,

$$\left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{ij}}, \phi \right\rangle = \frac{1}{|Y|} \int_\omega e(\phi) : C^{\text{base}} : (e^{kl} + e(\mathbf{w}^{kl}))\, \mathrm{d}\mathbf{y} = 0,$$

which vanishes by the weak form of the kl^{th} cell problem (2.23). The same argument holds for $\left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{kl}}, \phi \right\rangle$, so we have

$$\left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{ij}}, \delta \mathbf{w}^{ij} \right\rangle = \left\langle \frac{\partial C^H_{ijkl}}{\partial \mathbf{w}^{kl}}, \delta \mathbf{w}^{kl} \right\rangle = 0$$

without solving an adjoint problem. Thus Reynold's transport theorem gives the full shape derivative:

$$\begin{aligned} \mathrm{d}C^H_{ijkl}[\mathbf{v}] \overset{\text{def}}{=} \delta C^H_{ijkl} &= \left\langle \frac{\partial C^H_{ijkl}}{\partial \omega}, \delta\omega \right\rangle \\ &= \frac{1}{|Y|} \int_{\partial\omega} \left[(e^{ij} + e(\mathbf{w}^{ij})) : C^{\text{base}} : (e^{kl} + e(\mathbf{w}^{kl})) \right] \mathbf{v} \cdot \hat{\mathbf{n}}\, \mathrm{d}A(\mathbf{y}). \end{aligned} \tag{3.7}$$

Be aware that, though this shape derivative formula—whose derivation only holds exactly in the continuous case—proved accurate enough for the plain tensor-fitting application in this chapter, later we will derive a more accurate formula that is discretely exact (A.24).

Shape derivative of compliance tensor. The compliance tensor is the symmetric rank 4 inverse of elasticity tensor, i.e. $S_{ijkl}C_{klmn} = \frac{1}{2}(\delta_{im}\delta_{jn} + \delta_{in}\delta_{jm})$. Differentiating and solving for $\mathrm{d}S^H$:

$$\mathrm{d}S^H[\mathbf{v}] = -S^H : \mathrm{d}C^H[\mathbf{v}] : S^H. \tag{3.8}$$

Combining the results from (3.5), (3.7), and (3.8), one can compute $\frac{\partial J}{\partial p_\alpha}$; the shape derivative and an example velocity field $\mathbf{v}_{p_\alpha}$ are shown in Figure 3.13.

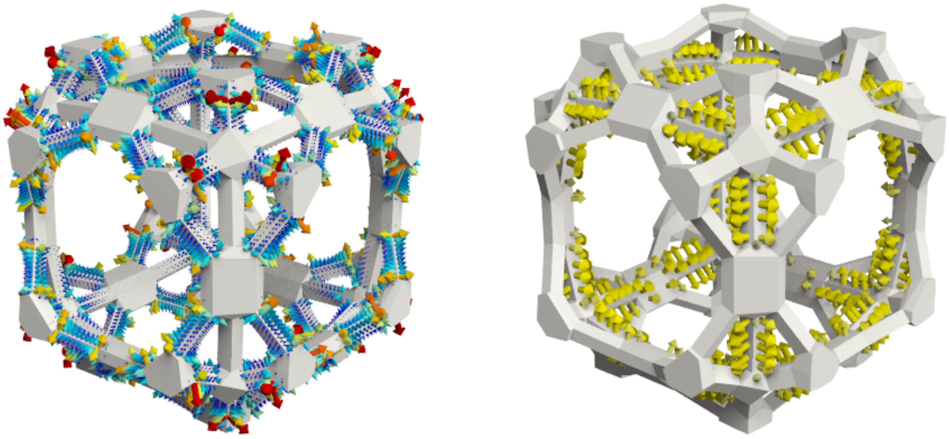


Figure 3.13: Left: a shape derivative, visualized as a steepest ascent normal velocity field for objective (3.3). Right: the shape velocity induced by one of the pattern's thickness parameters.

Numerical computation. The integrand in (3.7) is cubic over each boundary element ($\varepsilon(\mathbf{w}^{ij})$ and $\mathbf{v} \cdot \hat{\mathbf{n}}$ are both linear), and we use quadrature that evaluates the surface integral exactly. To evaluate the gradient for a given shape we need (a) to mesh the shape (we use the TetGen package [92]); (b) solve 6 periodic elasticity problems to obtain $\mathbf{w}^{ij}$, as for homogenization. The cost of a single gradient evaluation (roughly 4.75s on a single core of an Intel Xeon E-2690 v2) is dominated by the cost of periodic meshing and the elasticity solves, which take roughly equal time. We use the Ceres solver [1]'s Levenberg-Marquardt implementation to minimize the objective; the convergence of the solver is quite fast (Figure 3.14).

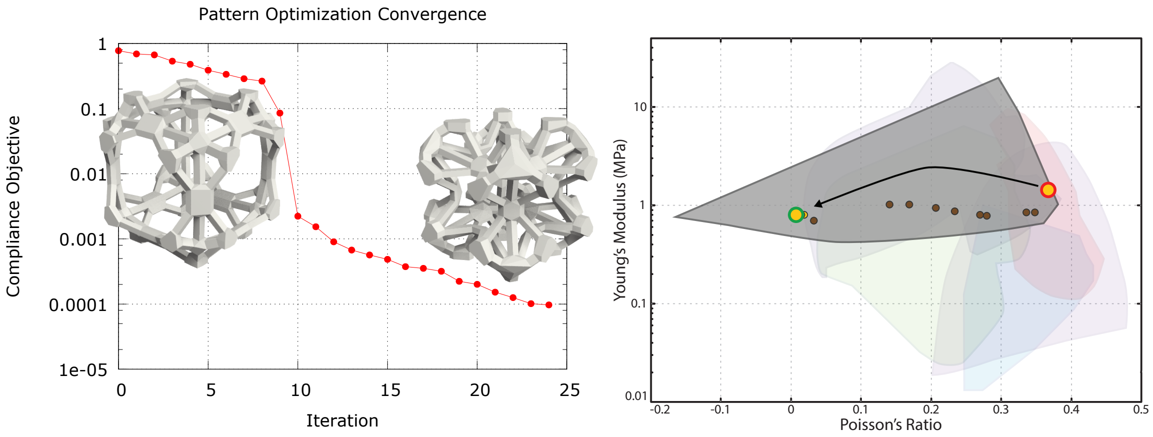


Figure 3.14: Left: convergence of a shape optimization on pattern “(E1,E2)(E1,E4)(E2,E4).” Right: the in (E, ν) space traversed by this optimization. The brown points are intermediate, anisotropic microstructures.

3.7 Applications

While our primary focus is on the design of our pattern family and the exploration of its coverage, we demonstrate the application of our elastic textures in two settings: painted material properties and specified deformation behavior. All printing was done using a B9Creator printer at 50 micron resolution with Cherry resin.

Overall workflow. The result of the preceding sections is a lookup map that, for a given (E, ν), produces an isotropic microstructure with nearby parameters. We assume that we are given a coarse volume mesh filled with identical cube cells.

First, we assign a pair (E_i, ν_i) to each cell i, either directly or via material optimization as described below. For each cell, we retrieve a corresponding pattern (topology id, thicknesses, and offsets) from the lookup table. The microstructures in adjacent cells are stitched together by averaging the offsets of each pair of shared face nodes so that they coincide. This might raise the lower node of the pair above some node it supports, n_s, violating printability, but printability can be restored by lowering the pair to n_s's height.

After this step, the resulting connected wire mesh is inflated with retrieved bar thick-

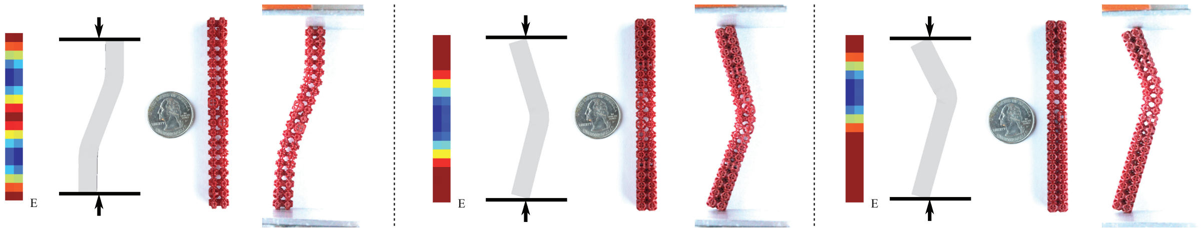


Figure 3.15: Examples of objects with painted material properties. All are fabricated with 5mm cells.

nesses, using the process described in Section 3.5.1. This results in a fine mesh that can be printed.

Material painting. The simplest approach to specifying the material properties (E, ν) is to paint them on a voxel grid. We have created an editor enabling us to paint these layer by layer. The results of fabricating several structures of this type are shown in Figure 3.15.

3.8 Material Optimization

Manually defining material properties to achieve desired behavior may be difficult, and a more systematic approach is to solve for them. For example, consider the following problem: for given applied forces on some part of the object's boundary, we would like to get some target deformation—e.g., compressing a bar along the Z-axis causes it to twist in the X-Y plane—by varying the material properties.

3.8.1 Direct Approach

In this problem's most natural formulation, we wish to find the spatially varying tensor $C_p(\mathbf{x})$ (parametrized by a vector of per-cell isotropic parameters p) solving:

$$\begin{aligned} \min_{C_p} \int_{\Gamma_t} \|\mathbf{u} - \mathbf{u}_t\|^2 \, \mathrm{d}A(\mathbf{x}) & \\ \text{s.t.} - \nabla \cdot [C : \varepsilon(\mathbf{u})] = 0 \quad & \text{in } \Omega \\ \hat{\mathbf{n}} \cdot [C : \varepsilon(\mathbf{u})] = \tau \quad & \text{on } \Gamma_n \\ \mathbf{u} = \mathbf{u}_d \quad & \text{on } \Gamma_d, \end{aligned} \tag{3.9}$$

where τ and $\mathbf{u}_d$ are the applied loads and boundary displacements on the compressed area, and $\mathbf{u}_t$ are the target displacements of the surface Γ_t. In our implementation, these target/applied conditions can be specified on a per-component basis, e.g. to set up the twisting bar example.

For the input to make sense, it should satisfy $\Gamma_n \cup \Gamma_d = \partial\Omega$, $\Gamma_n \cap \Gamma_d = \emptyset$, and $\Gamma_t \subseteq \Gamma_n$ (i.e. the full boundary's loading is specified, the Neumann and Dirichlet regions do not overlap, and the target region does not overlap the Dirichlet region).

The gradient can be computed efficiently using the adjoint method, but we've found this optimization problem difficult to solve robustly. Nevertheless, this approach was used in [19], and [48] applies a similar approach where the traction condition can be made a soft constraint to improve robustness. To further improve robustness and accelerate convergence, [48] searches over a reduced subspace of smooth material distributions.

3.8.2 "Local-Global" Strain-Fitting Approach

Instead of fitting displacements directly, we found an iterative strain-fitting approach to work better. The idea is to run two simulations: one with the loading scenario only and the other with the target condition added as a Dirichlet constraint. These problems are called the "Neumann" and "Dirichlet" problems (with solutions $\mathbf{u}_N$ and $\mathbf{u}_D$) because the former applies the Neumann condition on the target region Γ_t, while the latter enforces the target displacement on Γ_t using a Dirichlet condition:

$$\begin{array}{rlcrl} \multicolumn{2}{c}{\text{“Neumann Problem”}} & & \multicolumn{2}{c}{\text{“Dirichlet Problem”}} \\ -\nabla \cdot [C_{p_i} : \varepsilon(\mathbf{u}_N)] = 0 & \text{in } \Omega & & -\nabla \cdot [C_{p_i} : \varepsilon(\mathbf{u}_D)] = 0 & \text{in } \Omega \\ \hat{\mathbf{n}} \cdot [C_{p_i} : \varepsilon(\mathbf{u}_N)] = \tau & \text{on } \Gamma_n & & \hat{\mathbf{n}} \cdot [C_{p_i} : \varepsilon(\mathbf{u}_D)] = \tau & \text{on } \Gamma_n \setminus \Gamma_t \\ \mathbf{u}_N = \mathbf{u}_d & \text{on } \Gamma_d & & \mathbf{u}_D = \mathbf{u}_d & \text{on } \Gamma_d \\ & & & \mathbf{u}_D = \mathbf{u}_t & \text{on } \Gamma_t \end{array} \tag{3.10}$$

(recall $\Gamma_t \subseteq \Gamma_n$). Unless the current material field $C_{p_i}(\mathbf{x})$ already achieves the target deformation, the strain fields of the two simulations will differ. We can view $\varepsilon(\mathbf{u}_D)$ as an estimate of the target deformation's strain field (since it integrates to the correct target boundary displacement). We can also view $\sigma(\mathbf{u}_N)$ as an estimate of the internal stresses in the target deformation (since it satisfies the loading conditions). This suggests updating $C(\mathbf{x})$ to minimize the Frobenius norm distance between $\varepsilon(\mathbf{u}_D)$ and the strain corresponding to stress estimate $\sigma(\mathbf{u}_N)$:

$$C_{p_{i+1}} = \underset{C}{\operatorname{argmin}} \ \frac{1}{2} \int_{\Omega} \|C^{-1} : \sigma(\mathbf{u}_N) - \varepsilon(\mathbf{u}_D)\|_F^2 \, \mathrm{d}\mathbf{x}. \tag{3.11}$$

This least-squares minimization can be done over any subspace of material fields. For instance, in the results shown in Section 3.9.3, we minimize over the space of piecewise-constant isotropic materials, parametrized by a Young's modulus and a Poisson's ratio per voxel. We add a Laplacian regularization term to encourage smooth variation, but model reduction could be used as in [48]. We can then use $C_{p_{i+1}}$ in the next round of simulations (3.10).

To summarize, our material optimization algorithm consists of repeating the following two steps:

- Solve the "Neuman" and "Dirichlet" problems (3.10) for $\mathbf{u}_N$ and $\mathbf{u}_D$.

- Update the material property field by solving (3.11).

We initialize these iterations with a uniform elastic tensor field $C_{p_0}(\mathbf{x})$.

Without the Laplacian regularization, the strain-fitting step (3.11) can be performed independently in each voxel, and our iterations alternate between global PDE solves and local fitting. This is very similar in spirit to the ARAP deformation and parametrization techniques, and we adopt the local-global name (even though the regularization term makes it really a global-global iteration...).

3.9 Convergence

While it's not immediately clear that this strain-fitting local-global iteration should converge, we found it extremely robust in practice; the typical convergence behavior of the local-global approach is shown in Figure 3.16.

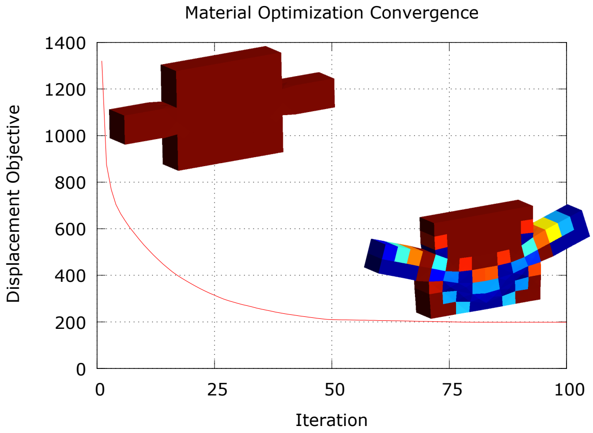


Figure 3.16: Convergence of material optimization.

We can see why this might be the case by considering a similar but more "symmetric" fitting energy. First, notice that instead of fitting strains as in (3.11), we could just as easily have fit stresses:

$$C_{p_{i+1}} = \operatorname*{argmin}_C \ \frac{1}{2}\int_\Omega \|\sigma(\mathbf{u}_N) - C : \varepsilon(\mathbf{u}_D)\|_F^2 \ \mathrm{d}\mathbf{x}.$$

In fact, in force-feedback applications (as opposed to target deformation applications), this version is probably preferable as it penalizes deviation of internal forces directly. A natural compromise between the two is to perform the fitting in a space "halfway between" stress and strain:

$$C_{p_{i+1}} = \operatorname*{argmin}_C \ \frac{1}{2}\int_\Omega \|C^{-\frac{1}{2}} : \sigma(\mathbf{u}_N) - C^{\frac{1}{2}} : \varepsilon(\mathbf{u}_D)\|_F^2 \ \mathrm{d}\mathbf{x}. \tag{3.12}$$

This symmetric fitting energy has a very nice property: we can prove local-global iterations always decrease the fitting energy (3.12). First, we expand the energy:

$$\begin{aligned}
&\frac{1}{2}\int_\Omega \|C^{-\frac{1}{2}} : \sigma(\mathbf{u}_N) - C^{\frac{1}{2}} : \varepsilon(\mathbf{u}_D)\|_F^2 \ \mathrm{d}\mathbf{x} \\
&\quad = \int_\Omega \frac{1}{2}\sigma(\mathbf{u}_N) : C^{-1} : \sigma(\mathbf{u}_N) + \frac{1}{2}\varepsilon(\mathbf{u}_D) : C : \varepsilon(\mathbf{u}_D) - \underbrace{\sigma(\mathbf{u}_N) : \varepsilon(\mathbf{u}_D)}_{I} \ \mathrm{d}\mathbf{x}.
\end{aligned}$$

Integrating the "I" term by parts (using (2.7)):

$$I = -\int_\Omega \cancelto{0}{\Big(\nabla \cdot \sigma(\mathbf{u}_N)\Big)} \cdot \mathbf{u}_D \ \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \Big(\sigma(\mathbf{u}_N)\hat{\mathbf{n}}\Big) \cdot \mathbf{u}_D \ \mathrm{d}A(\mathbf{x}).$$

3.9.1 Simple Case: Full-boundary Target Condition

We first consider the case where $\Gamma_n = \Gamma_t = \partial\Omega$ and $\Gamma_d = \emptyset$. In other words, tractions and target displacements are specified on the entire boundary, and no Dirichlet conditions exist in the loading scenario. Then:

$$I = \int_{\partial\Omega} \Big(\sigma(\mathbf{u}_N)\hat{\mathbf{n}}\Big) \cdot \mathbf{u}_D \ \mathrm{d}A(\mathbf{x}) = \int_{\partial\Omega} \tau \cdot \mathbf{u}_t \ \mathrm{d}A(\mathbf{x}) = \text{const},$$

and the "I" term is just a constant depending on the prescribed forces and target displacements. In this simplest case, it's clear that the fitting energy (3.12) is really just the sum of two elastic energies (up to a constant): the elastic energies of the two simulations! Thus, we can view our "local-global" simulate-then-fit iterations for (3.12) as an alternating minimization,

$$\begin{aligned} C^* &= \underset{C}{\operatorname{argmin}} \min_{\substack{\nabla\cdot\sigma_N=0 \\ \sigma_N\hat{\mathbf{n}}=\tau \text{ on } \partial\Omega}} \min_{\substack{\varepsilon_D=\varepsilon(\mathbf{u}_D) \\ \mathbf{u}_D=u_t \text{ on } \partial\Omega}} \frac{1}{2}\int_\Omega \|C^{-\frac{1}{2}} : \sigma_N - C^{\frac{1}{2}} : \varepsilon_D\|_F^2 \,\mathrm{d}\mathbf{x} \\ &= \underset{C}{\operatorname{argmin}} \min_{\substack{\nabla\cdot\sigma_N=0 \\ \sigma_N\hat{\mathbf{n}}=\tau \text{ on } \partial\Omega}} \min_{\substack{\varepsilon_D=\varepsilon(\mathbf{u}_D) \\ \mathbf{u}_D=u_t \text{ on } \partial\Omega}} \frac{1}{2}\int_\Omega \sigma_N : C^{-1} : \sigma_N + \varepsilon_D : C : \varepsilon_D \,\mathrm{d}\mathbf{x}, \end{aligned} \tag{3.13}$$

proving that each step decreases the energy. In particular, minimizing complementary potential energy over divergence-free σ_N satisfying the traction condition is equivalent to the "Neumann" simulation, and minimizing potential energy over integrable strain fields whose corresponding boundary deformations satisfy the target conditions is equivalent to the "Dirichlet" simulation.

3.9.2 General Case

In general, we can decompose the "I" term into three surface integrals:

$$I = \left(\overset{\text{const}}{\cancel{\int_{\Gamma_t}}} + \int_{\Gamma_n\setminus\Gamma_t} + \int_{\Gamma_d} \right) \Big(\sigma(\mathbf{u}_N)\hat{\mathbf{n}}\Big) \cdot \mathbf{u}_D \,\mathrm{d}A(\mathbf{x}).$$

The first integral depends only on the prescribed tractions and target displacements, as in the simple case. Thus, up to a constant, we can write

$$I = \int_{\Gamma_n\setminus\Gamma_t} \tau \cdot \mathbf{u}_D \,\mathrm{d}A(\mathbf{x}) + \int_{\Gamma_d} \Big(\sigma(\mathbf{u}_N)\hat{\mathbf{n}}\Big) \cdot \mathbf{u}_d \,\mathrm{d}A(\mathbf{x}).$$

The first integral is the work done by the prescribed traction τ in the "Dirichlet problem," and the second is the work done by imposing the Dirichlet loading condition $\mathbf{u}_d$ in the

"Neumann problem." This means we can rewrite the strain fitting energy (3.12) as:

$$\underbrace{\frac{1}{2}\int_{\Omega}\sigma(\mathbf{u}_N) : C^{-1} : \sigma(\mathbf{u}_N)\ \mathrm{d}\mathbf{x} - \int_{\Gamma_d}\Big(\sigma(\mathbf{u}_N)\hat{\mathbf{n}}\Big)\cdot \mathbf{u}_d\ \mathrm{d}A(\mathbf{x})}_{\text{Total Complementary Potential Energy of ``Neumann Simulation''}}$$
$$+\ \underbrace{\frac{1}{2}\int_{\Omega}\varepsilon(\mathbf{u}_D) : C : \varepsilon(\mathbf{u}_D)\ \mathrm{d}\mathbf{x} - \int_{\Gamma_n\setminus\Gamma_t}\tau\cdot \mathbf{u}_D\ \mathrm{d}A(\mathbf{x})}_{\text{Total Potential Energy of ``Dirichlet Simulation''}}.$$

Finally, as in (3.13), we can express the local-global iteration as an alternating minimization of the symmetric fitting energy:

$$\begin{aligned}
C^* &= \underset{C}{\operatorname{argmin}} \min_{\substack{\nabla\cdot\sigma_N=0\\ \hat{\mathbf{n}}\cdot\sigma_N=\tau \text{ on } \Gamma_N}} \min_{\substack{\varepsilon_D=\varepsilon(\mathbf{u}_D)\\ \mathbf{u}_D=\mathbf{u}_d \text{ on } \Gamma_d\\ \mathbf{u}_D=u_t \text{ on } \Gamma_t}} \frac{1}{2}\int_{\Omega}\|C^{-\frac{1}{2}} : \sigma_N - C^{\frac{1}{2}} : \varepsilon_D\|_F^2\ \mathrm{d}\mathbf{x} \\
&= \underset{C}{\operatorname{argmin}} \min_{\substack{\nabla\cdot\sigma_N=0\\ \hat{\mathbf{n}}\cdot\sigma_N=\tau \text{ on } \Gamma_N}} \min_{\substack{\varepsilon_D=\varepsilon(\mathbf{u}_D)\\ \mathbf{u}_D=\mathbf{u}_d \text{ on } \Gamma_d\\ \mathbf{u}_D=u_t \text{ on } \Gamma_t}} \text{``Neumann'' TCPE} + \text{``Dirichlet'' TPE},
\end{aligned} \tag{3.14}$$

where again the inner minimizations are performed by the two global elasticity solves (3.10).

3.9.3 Examples

We have used a number of simple voxelized shapes and created a variety of deformation behaviors shown in Figure 3.18. Finally, we have also generated a set of anisotropic samples, with controlled anisotropy ratio, one of which is shown in Figure 3.17.

3.10 Conclusions

We have presented a family of tileable and printable patterns that can be used to approximate varying isotropic material properties. The family has proved useful on a number of simple shape optimization examples: remarkably, all examples in Section 3.7

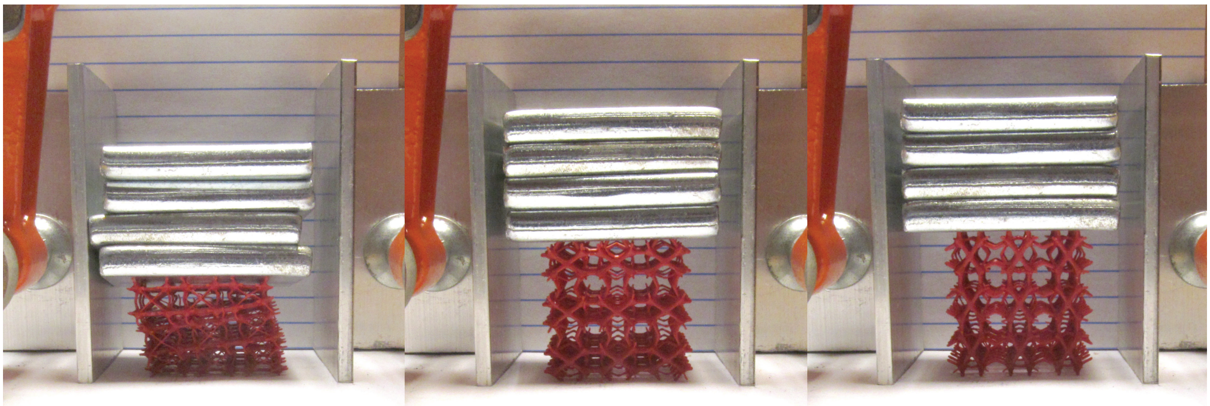


Figure 3.17: Compression of an anisotropic sample along the X, Y, and Z directions.

worked as predicted by simulation without requiring much tuning.

Limitations. There are several limitations of our pattern family. First, some parts of the (E, ν) space are poorly covered. While it is difficult to predict which part of space is theoretically reachable, we conjecture that the space may be significantly broadened. All our simulations and constructions work in the linear regime, not taking into account, e.g., the potential for pattern buckling or other damage. Fortunately, isotropy is correlated with sufficiently high shear modulus, which makes the patterns less prone to buckling. Nevertheless, including this and other nonlinear effects in pattern design is important.

For practical use, it is difficult to restrict the tessellations of objects to equal sized cubes (though one can construct cut cells covered with relatively soft skin). A desirable solution would be to allow patterns to distort to fill arbitrary reasonably well-shaped hex cells.

Acknowledgements

We thank Professor Bob Kohn (NYU Courant Institute) for his many helpful insights. We also thank John Ricci and Yu Zhang (NYU College of Dentistry) and Oran Kennedy and Matin Lendhey (NYU School of Medicine) for their help with material testing.

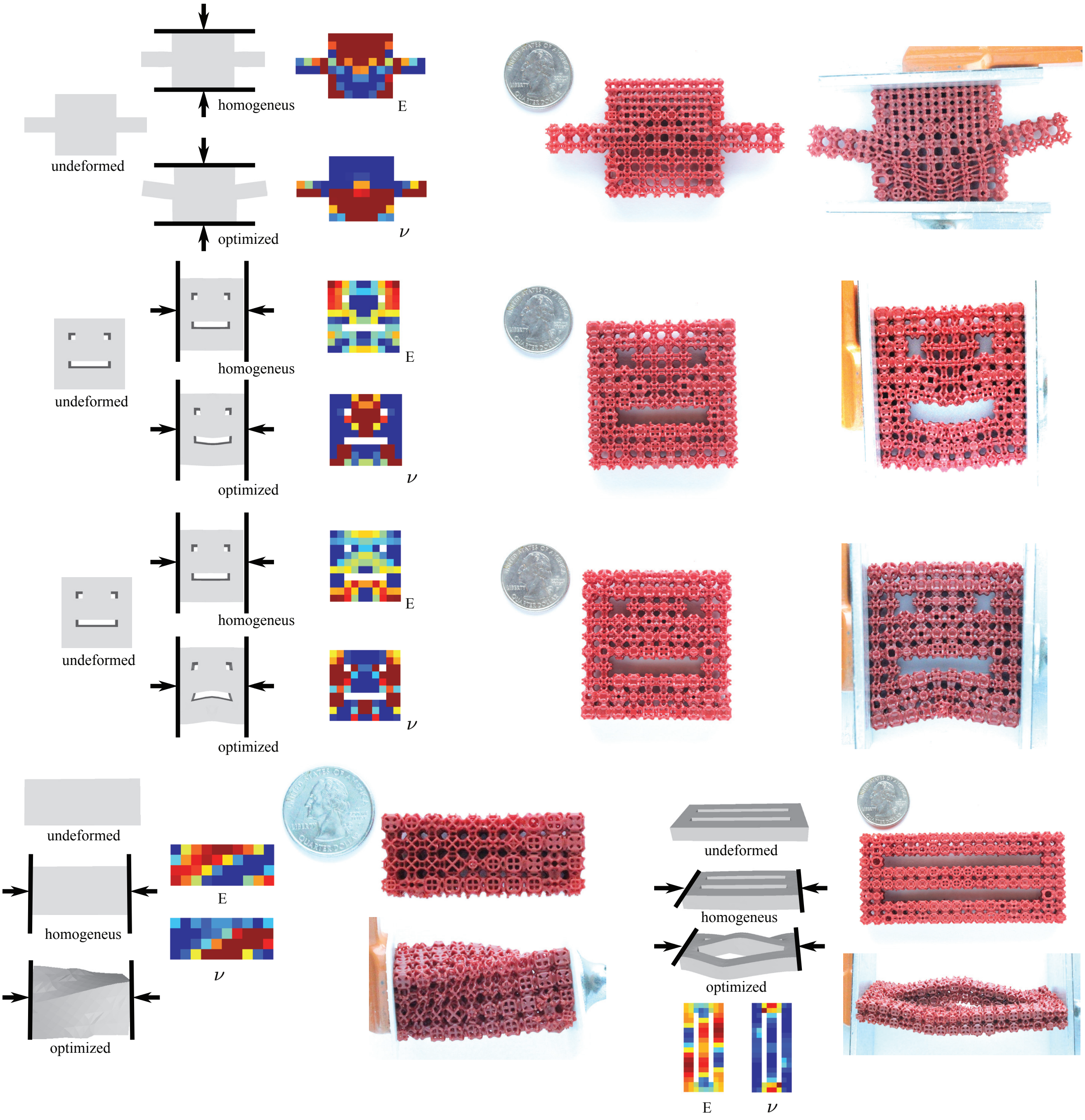


Figure 3.18: Examples of objects with optimized material properties. All are fabricated with 5mm cells.

Considering the "symmetric" version of the strain-fitting energy (3.12) was suggested by Bob Kohn; he proposed the same approach in [53] to determine an object's electrical conductivity tensor distribution from boundary voltage and current measurements.

This work was partially supported by NSF award DMS-1436591.

Chapter 4

Worst-Case Stress Analysis for Macroscopic Objects

The two-scale pipeline presented in Chapter 3 proved to be an effective tool for designing objects to achieve deformation goals when fabricated on a single-material 3D printer. However, we found the resulting objects often fractured in practice, even under mild deformations. For practical applications of 3D printing, and particularly for objects with fine-scale structure, it is essential to keep in mind objects' stress behavior as we design them.

Traditionally, stress analysis requires that one specify the forces applied to his or her design. For many design tasks, the loads a design will undergo in practice are unknown a priori, either because the designer has insufficient experience or—in the case of designing a microstructure database—because the design is inherentely multi-purpose. This chapter tackles the problem of stress analysis under unknown loads at the macroscopic level, preparing us for the minimal-stress microstructure design problem in Chapter 5. It is based on publication [115], which was joint work with Qingnan Zhou and Denis Zorin.

I worked on dramatically speeding up the stress analysis algorithm, comparing various different formulations of the linear stress objective, and developing an algorithm to provide intuitive visual summaries of the worst-case force fields.

I also developed a “mesh-free” worst-case analysis tool (described in Section 4.7), bypassing the need to mesh the shape’s interior. This is important, as volume meshing fails for the ill-behaved (self-intersecting, non-manifold) geometry often present in users’ designs: it is the analysis pipeline’s bottleneck in terms of robustness. This tool works by embedding the object in a regular grid of bilinear or trilinear finite elements and carefully handling the “cut cells” overlapping the object’s boundary. Unfortunately, as discusssed in Section 4.7, this method has limitations: the stresses computed at the cut cells along the boundary tend to be inaccurate. This material appears here for the first time (it was not part of [115]).

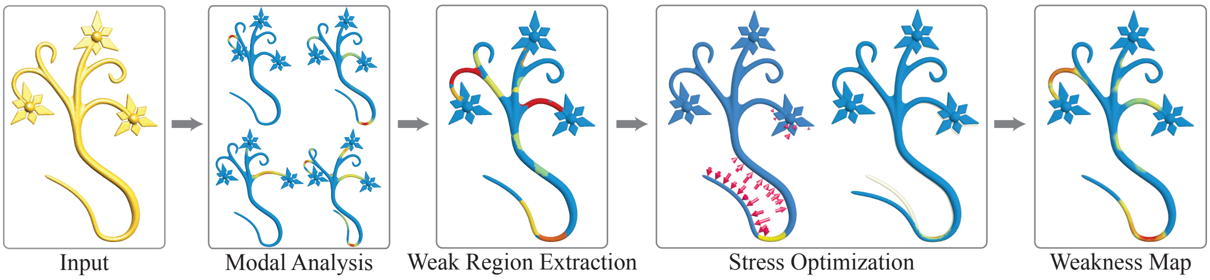


Figure 4.1: Worst-case analysis pipeline.

This chapter presents a method to identify structural problems in objects designed for 3D printing *based on geometry and material properties only*, without specific assumptions on loads and manual load setup. We solve a constrained optimization problem to determine the “worst” load distribution for a shape that will cause high local stress or large deformations. While in its general form this optimization has a prohibitively high computational cost, we demonstrate that an approximate method makes it possible to solve the problem rapidly for a broad range of printed models. We validate our method both computationally and experimentally and demonstrate that it has good predictive power for a number of diverse 3D printed shapes.

4.1 Introduction

We present an algorithm approximating the solution of the following problem: *From the shape of an object and its material properties, determine the easiest (in terms of minimal applied force) ways to break it or severely deform it.*

Our work is largely motivated by applications in 3D printing. The cost of 3D printing has decreased significantly over the past few years, and the industry is undergoing a rapid expansion, making customized manufacturing in an increasingly broad variety of materials available to a broad user base. While many of the users are experienced creators of digital 3D shapes, engineering design expertise is far less common, and widely used 3D modeling tools lack accessible ways to predict the mechanical behavior of a 3D model.

There are a number of reasons why a 3D model might fail, either during or after manufacture:

(1) the dimensions of thin features (walls, cylinder-like features, etc.) are too small for the printing process, resulting in shape fragmentation at the printing stage;

(2) the strength of the shape is not high enough to withstand gravity at one of the stages of the printing process;

(3) the shape is damaged during routine handling during the printing process or shipment;

(4) the shape breaks during routine use.

In most cases, the first problem is addressed by simple geometric rules ([98]), and the second is a straightforward direct simulation problem. Our focus is on the other two problems. On the one hand, many 3D printed objects are manufactured with a specific mechanical role in mind, and full evaluation is possible only if sufficient information on expected loads is available. On the other hand, jewelry, toys, art pieces, various types of clothing, and gadget accessories account for a large fraction of products shipped by 3D printing service providers. These objects are often expected to withstand a variety of poorly defined loads (picking up, accidental bending or dropping, forces during shipping, etc.).

To predict structural soundness of a printed object, we look for *worst-case* loads, within a suitably constrained family, that are most likely to result in damage or undesirable deformations. A direct formulation results in difficult nonlinear and nonconvex optimization problems with PDE constraints. We have developed an approximate method for this search, reducing it to an eigenproblem and a sequence of linear programming problems.

We demonstrate experimentally that our approach predicts the breaking locations and extreme deformations quite well. While primarily designed for 3D printing applications, our method can be applied in any context where loads are unpredictable and structural weaknesses need to be identified.

4.2 Related work

Computational analysis of structural soundness of mechanical parts and buildings is broadly used, but almost always in the context of known sets of loads. While engineers routinely need to evaluate soundness of structures and mechanisms under worst-case scenarios, in most cases, worst-case loads are designed empirically for specific problems (e.g., construction of buildings to withstand loads from flooding or earthquakes). Automatic methods are less common: an important set of methods in the context of modeling under uncertainty is based on the idea of *anti-optimization* (e.g., [32]). Our work is partially inspired by these concepts.

In aerospace engineering, filter-based methods were developed to predict worst-case gusts and turbulence encountered by an airplane. E.g., [113, 35] model the aircraft's response to turbulence as a linear filter's response to Gaussian white noise. From this model, a worst-case noise sample and resulting strain are obtained.

In the context of analysis tailored to 3D printing applications of the type considered in this paper, the closest work to ours is [97]. The paper proposes to evaluate 3D shapes in two main scenarios to discover structure weakness: applying gravity loads and gripping the shape using 2 fingers at locations predicted by a heuristic method. This set of fixed usage scenarios is often insufficient to expose the true structure weakness for many printed shapes, as discussed in greater detail in Section 4.5. The paper also describes methods for automatic improvement of objects. [98] focuses on purely geometric ways to evaluate whether a structure is suitable for 3D printing based on empirical rules formulated by the 3D printing industry ([110], [90]).

Structural stability for simple furniture constructed from rigid planks connected by nails is analyzed at interactive rates in [104]. Their system also suggests corrections when shapes with poor stability are detected.

A number of recent works address various aspects of computational design for 3D printing. [18] provide a pipeline to print objects in a composite material that reproduces desired deformation behavior. To achieve this goal, the authors accurately model the nonlinear stress-strain relationship of their printing materials and how printed models will respond to imposed loads. The space of deformations is a user-supplied input, and structural soundness of the design with respect to other loads is not considered. While some specialized work on CAD for 3D printing exists, (e.g., the system for heterogeneous material design [54]), overwhelmingly, standard tools with little or no analysis support are used.

[65] proposes a framework to decompose 3D shapes into smaller parts that can be assembled without compromising the physical functionality of the shape so that larger

objects can be printed using printers with a small working volume. They use a standard finite element simulation to estimate stress of the input shape under gravity in a user specified upright orientation. Other works aim to print articulated models that maintain poses under gravity but do not require manual assembly. [24] designs and fits a generic, parametrized printable joint template based on a ball and socket joint. Their joint provides enough internal friction and strength to hold poses and survive manipulation, but they tune its parameters experimentally instead of using a physically-based optimization. [10] designs a similar ball and socket joint and a hinge joint. An approximate geometric optimization of stresses is performed by maximizing certain cross-sectional areas of the joint.

3D printing has also been used to reproduce appearance: [44] and [31] optimize the layering of base materials in a 3D multi-material printer to print objects whose subsurface scattering best matches an input BSSRDF.

Our method relies on using eigenmodes of the shape. Modal analysis has proven useful in many contexts. The use of Laplacian eigenmodes of simple shapes for computation predates computers [99] and has a long history in model order reduction for a variety of applications including nonlinear elasticity (e.g., [74]). In graphics literature, [78] first introduced eigenmodes as a basis suitable for simulation applications, and more recently, a number of deformation-related algorithms based on eigenmode bases were proposed, e.g., [43, 11, 12].

At the same time, *experimental* modal analysis (applying periodic forces with different frequencies and measuring displacements at various points) is broadly used to detect structural damage [33].

Finally, [80] presents an overview of several simulations and experiments exploring how printing parameters (build orientation, layer thickness, scan path and speed, temperature, etc.) affect the accuracy and strength of simple shapes. The goal of these works is to evaluate and improve printing technology itself rather than detecting or fixing deficiencies in the input shape.

4.2.1 3D printing processes

To motivate the design of our structural analysis process, we briefly review the most commonly used 3D printing processes and the types of structural problems one can expect. The most relevant aspects of 3D printing processes for structural analysis are the mechanical characteristics of materials produced at different stages and typical loads on the object during and after the production process.

Common *single-stage* 3D printing processes either deposit the liquid material only in needed places (e.g., FDM) or deposit material in powder form layer-by-layer and then fuse or harden it at points inside the object (e.g., stereolithography uses photosensistive polymers, and laser sintering fuses regular polymers by heat).

These processes typically use flexible polymers with large elastic and plastic zones in their stress-strain curves. These polymers rarely break if geometric criteria for printability are satisfied, but they can undergo large plastic deformations.

Printing metal, ceramics, and composite materials often involves multiple stages. For example, the object may be printed layer-by-layer in metal powder with polymer binder. At the next stage, the binder is cured in a furnace, resulting in a *green state* part, and at the last stage, the metal is fused in a furnace and extra metal is added. Green state is brittle and has low strength, so parts in this state are easily damaged. A simpler multistage process is used for relatively brittle composite materials, e.g., gypsum-based multicolor materials: a second curing stage is used to give the material additional strength. Both the green state and the final material are relatively brittle. Whenever binding polymer is mixed layer-by-layer with a different material, the resulting material is likely to be highly anisotropic.

To summarize, both brittle and ductile materials are of importance. The former requires predicting where the material is likely to break, and the latter requires predicting extreme deformations likely to become plastic. Due to the layer-by-layer nature of the printing process, anisotropy is common and needs to be taken into account. Some of the loads even during production stages are hard to predict and quantify.

Goals. These considerations lead to several possible structural analysis goals, unified by a common theme of identifying *worst-case* loads in a family of loads satisfying some constraints (bounds on total force, pressure, direction etc.). The worst-case load is understood to be the one that leads to maximal damage, which can be measured by a norm of stress, maximal displacements, and various functions of these quantities.

4.3 Worst-case structural analysis

Next, we present a formal description of the problem. This formulation is computationally intractable, but it is needed as a foundation for a practical approximate version described in Section 4.4.

Linear elasticity. We use an anisotropic linear material model and the linear elasticity equations to model object behavior for the purposes of determining weak spots and worst-case force distributions. This model is adequate for some materials used in 3D printing, but nonlinear models may be necessary for others, as discussed in greater detail in Section 4.6. We emphasize that a distinction should be made between simulation with given loads used to determine precise stress distributions and computation used to determine approximate worst-case loads: lower accuracy is acceptable for the latter. We briefly review the standard elasticity equations for convenience—refer back to Section 2.1 for the details. The stress-strain relationship is linear, and stress is related linearly to displacement:

$$\sigma = C : \epsilon \quad \epsilon = \frac{1}{2}\left(\nabla u + \nabla u^T\right), \tag{4.1}$$

where ϵ is the strain tensor, σ is the stress tensor, and u is the displacement. C is the rank-4 elasticity tensor, C_{ijlm}, and the notation $C : \epsilon$ denotes application of this tensor to the strain tensor ϵ, $\sum_{l,m} C_{ijlm}\epsilon_{lm}$. We discuss the choice and effects of elasticity tensor C in greater detail in Section 4.6. We assume an orthotropic material, for which the tensor C_{ijlm} has up to 9 independent parameters. In a coordinate system aligned with the material axes, if we represent C as a 6×6 matrix acting on vectors of components of the symmetric strain tensors $[\epsilon_{11}, \epsilon_{11}, \epsilon_{33}, 2\epsilon_{23}, 2\epsilon_{31}, 2\epsilon_{12}]$, its inverse is given by

$$\begin{bmatrix} \frac{1}{Y_1} & -\frac{\nu_{21}}{Y_2} & -\frac{\nu_{31}}{Y_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{Y_1} & \frac{1}{Y_2} & -\frac{\nu_{32}}{Y_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{Y_1} & -\frac{\nu_{23}}{Y_2} & \frac{1}{Y_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{31} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{12} \end{bmatrix}$$

where Y_i are directional Young's moduli, G_{ij} are shear moduli, and ν_{ij} are Poisson ratios for different pairs of directions, satisfying $\nu_{ij}/Y_i = \nu_{ji}/Y_j$.

For dynamic linear problems with volume force density F, the equation of motion is

$$\rho\ddot{u} = \nabla \cdot \sigma + F, \tag{4.2}$$

where ρ is the density, and the dot signifies the time derivative. We are primarily interested in static problems, but as we use modal analysis at an intermediate stage, we retain the term $\rho\ddot{u}$.

Equation 4.2 is subject to boundary conditions: we primarily use a surface force density F_S, which is captured by the condition $\sigma\hat{\mathbf{n}} = F_S$ on the boundary of the object. If the object is attached to a rigid support, Dirichlet conditions $u = 0$ can be imposed on a part of the boundary.

If the equation of motion (4.2) is written directly in terms of displacement u, we get

$$-\nabla \cdot \left(C : \frac{1}{2}(\nabla u + \nabla u^T) \right) := Lu = F - \rho \ddot{u}. \tag{4.3}$$

Rigid motion, torque and translation constraints for static problems. If the object is not fixed at least at 3 non-collinear points, an arbitrary force distribution will result in motion of the whole object. As we are interested in considering unknown forces with no assumptions on attachment, we need to be able to eliminate global motion. We achieve this by imposing zero total force and zero total torque constraints, which can be written as

$$\begin{aligned} &\int_\Omega F dV + \int_{\partial\Omega} F_S dA = 0, \\ &\int_\Omega F \times (x - x_c) dV + \int_{\partial\Omega} F_S \times (x - x_c) dA = 0. \end{aligned} \tag{4.4}$$

(See the derivation of (2.6)).

Displacements enter into this system only in the form Lu, and the operator L has infinitesimal rigid motions in its nullspace (Section 2.2). To have a unique solution in u, we impose a zero rigid motion constraint, similar to total torque and force constraints:

$$\int_\Omega u dV = 0, \quad \int_\Omega u \times (x - x_c) dV = 0. \tag{4.5}$$

(See the derivation of (2.5)).

Surface force model. In cases of interest, the only volume force is gravity. In all but most extreme cases, gravity does not have a major effect, so we concentrate on surface forces. We restrict the forces in three ways.

- *Only inward-pointing normal forces allowed:* $F_S = -p\hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the surface normal, and p is a positive pressure value, thus ignoring friction. This is a reasonable assumption, as for most situations described in Section 4.2.1, friction forces are likely to be significantly lower than normal forces. At the same time, it is hard to model the bounds on ratios between normal and tangential components accurately in the absence of detailed knowledge of loads and surfaces. Without such bounds, any optimization is likely to produce unrealistic tangential results. Similarly, negative surface forces (e.g., electrostatic attraction), are unlikely to play a major role in practice and are excluded.

- *Pointwise pressure is bounded:* $p < p_{max}$. If a pressure may be unbounded, an arbitrarily high stress may be produced at a point on the surface. While highly concentrated forces are possible, these are rare, and we assume that a realistic bound on surface pressure is available.

- *The total force is fixed.* Again, by increasing the total force, arbitrarily high stresses can be obtained.

For example, if our primary target is simulating manual handling situations, one can bound the force by a typical force a human can apply by squeezing, and the maximal pressure is derived from the size of the finger tips.

Problem formulation. It remains to specify the objective function. One commonly used measure of interest is *maximal principal stress*, $\max_\Omega \max_{i=1,2,3} |\sigma_i|$, where σ_i are the eigenvalues of the stress tensor. The complete problem of finding the worst-case force distribution satisfying the constraints of our model and optimizing this objective function, has the form

$$
\begin{aligned}
&\max_{\Omega} \max_{i=1,2,3} |\sigma_i| \to \max; \\
&Lu = 0 \text{ on } \Omega, \quad \hat{\mathbf{n}} \cdot C : (\nabla u + \nabla u^T) = -p\hat{\mathbf{n}} \text{ on } \partial\Omega, \\
&\int_{\partial\Omega} p\hat{\mathbf{n}} dA = 0, \int_{\partial\Omega} p\hat{\mathbf{n}} \times (x - x_c) dA = 0, \\
&\int_{\Omega} u dV = 0, \int_{\Omega} u \times (x - x_c) dV = 0, \\
&0 \le p \le p_{max} \text{ on } \partial\Omega, \int_{\partial\Omega} p dA = F_{tot}.
\end{aligned}
\tag{4.6}
$$

Maximal principle stress is a suitable measure if we are interested in failure of materials, which occurs when the stress in a direction exceeds a bound. For plastic transition, the norm or some other function of the *deviatoric stress*, $\sigma - \frac{1}{3}\mathrm{tr}\sigma I$, may be of interest.

We make an interesting observation when the material is isotropic so that $C = Y\hat{C}$, where Y is the Young's modulus, and $\hat{C}$ is nondimensional, depending only on the Poisson ratio. Then maximal stress does not depend on Y but only on the Poisson ratio.

Solving this problem yields the worst-case principal stress and, importantly, the pressure distribution on the surface resulting in this stress. The maximal stress makes it possible to evaluate the likelihood of damage during the production process, shipping or use. Examining the pressure distribution makes it possible to evaluate how likely such loads would be and determine how the structure of the object can be strengthened.

We observe that all constraints in this problem are linear equality and inequality constraints, i.e., the constraints are *convex.* At the same time, the functional is highly non-linear (in fact, not smooth) and non-convex. Replacing maximal principal stress with another point measure maximized over the surface does not change the nature of the problem.

A brute-force solution can be obtained by solving a sequence of problems in which the objective functional $\max |\sigma_i|^2$ is maximized for every point and then taking the maximum of all results. Because we are interested in *maximizing* the norm, even these simpler per-point problems remain nonconvex and nonlinear.

We conclude that solving the optimization problem in general form is impractical, and due to non-linearities and non-convexity, any optimization is likely to get stuck in local minima.

Extension to displacements. An obvious extension of the algorithm is optimizing for *maximal displacements*. The main change is replacing σ with u in the functional: $\max_\Omega |u| \to \max$. This formulation is more relevant for flexible materials.

4.4 Efficient approximate algorithm

Overview. Figure 4.1 shows the main components of the efficient approximate algorithm for solving (4.6).

There are two problems we need to address to make the solution of (4.6) practical: (1) the need to solve an optimization problem for each point of the object to determine which one results in minimal stress; and (2) the nonlinearity and nonconvexity of each subproblem.

To address the first problem, we use a modal-analysis based heuristic that we found to work remarkably well. The second problem is solved by using $\operatorname{tr}\sigma$ as the *linear* objective functional. The reasons this substitution is possible for a broad range of cases are discussed in detail below.

Modal analysis and weak regions. A crucial ingredient of our method is *modal analysis*, which we use to restrict the part of the object where we need to maximize the stress or another functional.

Computational modal analysis refers to computing eigenvectors (eigenmodes) $\mathbf{u}_i$ and eigenvalues λ_i of L:

$$Lu_i = \lambda_i u_i,\ i = 1, 2 \ldots \tag{4.7}$$

Modal analysis is widely used in engineering and graphics for a variety of purposes. In the context of structural analysis, the most common application is predicting damage or deformations in presence of vibrations.

Our idea is similar in spirit, however there is a significant difference. We do not consider vibrations, i.e., periodically changing loads; rather, we consider static or quasi-static loads. We make the following

Assumption 1: *Examining a small number of eigenmodes allows us to find all regions of an object where the stress may be high under* arbitrary *deformation.* While this observation is difficult to prove mathematically, physical intuition suggests that vibrations of an object at different frequencies will result in high stress in all structurally *weak regions* of the object. Weak regions are those where high maximal stress can be obtained with low energy density relative to other parts of the object.

To validate this assumption, we have performed a brute-force optimization on a number of models (Figure 4.6) and compared with the results obtained using weak regions only. We obtain a remarkably good agreement in all cases.

We search for locations of potentially high stress by computing a number, M_m, of eigenmodes and considering a fraction $1 - \epsilon$ of points with highest stress under these deformations.

We define *weak regions* to be the connected components of this set. Each mode has multiple weak regions, typically associated with local stress maxima. For each mode we select M_r weak regions.

Approximate convex problem. The second important change to the problem is to replace the functional in (4.6) with a functional that can be optimized efficiently and that is minimized by a similar pressure distribution, p, to the original. We focus on the maximal stress, although a similar approach can be used for other functionals. We observe that almost invariably for any deformation and any compressible material with Poisson ratio ν sufficiently different from $1/2$:

For points where a principal stress is maximal, other principal stresses are small relative to the principal stress.

We have performed a validation of this observation by running simulations with a variety of loads and computing the ratio of the maximal principal stress to $|\operatorname{tr}\sigma|$. Over 36 models tested, the average ratio is 0.96 with standard deviation 0.25. Figure 4.2 illustrate that the distributions of trace and maximal principle stress are visually similar.

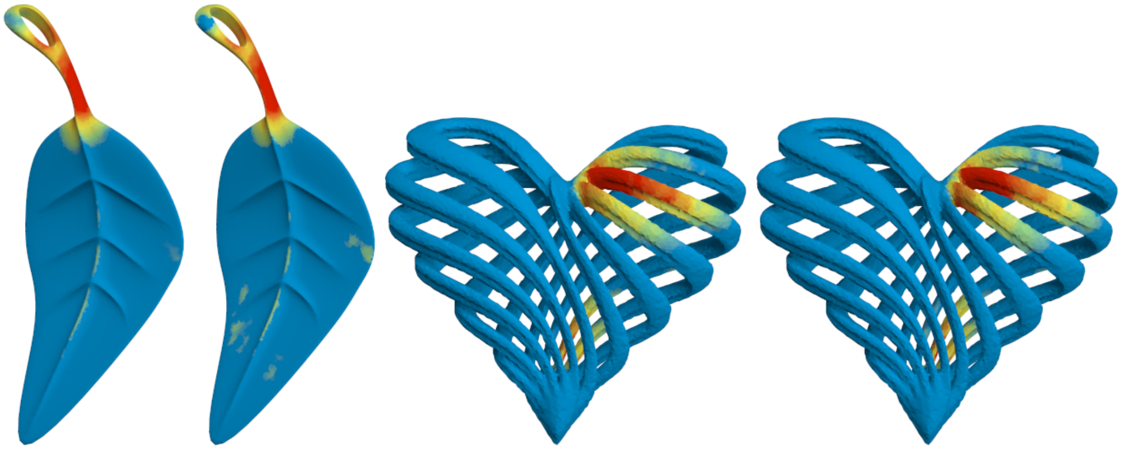


Figure 4.2: The top 10% volume of largest principal stress (left) and largest trace (right) are visually similar

We observe that when this is true, the difference between $|\sigma_{max}| = max_{i=1,2,3}|\sigma_i|$ and $|\sum_{i=1}^{3} \sigma_i|$, i.e., $|\operatorname{tr}\sigma|$ is small, and we can approximate the maximal principal stress with the absolute value of the trace.

As weak regions correspond to the highest stress area, and estimated stress tends to have a significantly lower accuracy vs. displacement, we use a weighted average of the stress over each weak region. The choice of weighting, as long as it falls off towards the boundary of the region, has relatively small effect on the result. We choose the L_2 norm of the stress computed from the eigenmode as the weight w for averaging the stress trace over each weak region. We also predict whether each point will stretch or compress under the worst-case load by computing $\operatorname{tr}\sigma$ under the modal displacement. We choose w's sign to match this quantity.

We finally arrive at the following approximate problem formulation:

For each eigenmode i, $i = 1 \ldots M_m$ and each of its weak regions, D_{ij}, $j = 1 \ldots M_r$, we solve the following linear programming problem:

$$
\begin{aligned}
&\int w \operatorname{tr}\sigma dV \to \max \text{ w.r.t. } u \text{ and } p; \\
&Lu = 0 \text{ on } \Omega, \quad \hat{\mathbf{n}} \cdot C : (\nabla u + \nabla u^T) = -p\hat{\mathbf{n}} \text{ on } \partial\Omega, \\
&\int_{\partial\Omega} p\hat{\mathbf{n}} dA = 0, \int_{\partial\Omega} p\hat{\mathbf{n}} \times (x - x_c) dA = 0, \\
&0 \le p \le p_{max} \text{ on } \partial\Omega, \int_{\partial\Omega} p dA = F_{tot}.
\end{aligned}
\tag{4.8}
$$

Unlike the original problem, this problem has a *unique* solution that can be computed efficiently using a convex solver.

Discretization and additional optimizations. We discretize the problem in the simplest conventional way, using piecewise-linear finite elements. The downside of this approach is that a suitable tetrahedral mesh needs to be generated for each input. For 3D printed models, the task is somewhat simplified: as the cost of printing is dominated by the amount of material used, almost all objects printed in practice are effectively thick shells to the extent this is allowed by the structural requirements. For this reason, tet meshing does not increase the number of vertices used to represent the object as much as one would expect.

Let n be the number of vertices, $n_b \le n$ be the number of boundary vertices, and m be the number of elements. The discretized quantities are: $\mathbf{p}$ the vector of pressures defined at boundary vertices of dimension n_b; and $\mathbf{u}$, the vector of displacements of dimension $3n$.

In the discrete formulation, we optimize the functional

$$\mathbf{w}^T VDB\mathbf{u}. \tag{4.9}$$

In this formula, V is a $6m \times 6m$ matrix, with the volume of element j repeated 6 times on the diagonal for the 6 components of the stress tensor. D is a $6m \times 6m$ block-diagonal matrix. For each element, the corresponding 6×6 block is the rank-4 tensor C in matrix form. B is a $6m \times 3n$ applying the FEM discretization of $\nabla + \nabla^T$. Finally, $\mathbf{w}^T$ is a vector that computes and weights the stress tensor traces, so that $\mathbf{w}^T V\mathbf{x}$ discretizes $\int_\Omega w \operatorname{tr} x dV$.

The discretized static elasticity equation combined with boundary conditions takes the form

$$K\mathbf{u} = -NA\mathbf{p}, \tag{4.10}$$

where K is the standard FEM $3n \times 3n$ stiffness matrix, $K = B^T VDB$. The matrix N is a $3n \times n_b$ matrix of components of surface normals, returning per-vertex components of external forces (0 for internal vertices, pn for the boundary), and matrix A is the $n_b \times n_b$ diagonal vertex area matrix.

The discretized formulation of the total force and torque constraints are:

$$\Sigma NA\mathbf{p} = 0, \, \Sigma TNA\mathbf{p} = 0, \tag{4.11}$$

where Σ is the $3 \times 3n$ matrix, summing n 3D vectors concatenated into a $3n$ vector, and T is $3n \times 3n$ block-diagonal matrix computing the torques of the surface force vectors.

Putting all these together, the discretized optimization problem is:

$$
\begin{aligned}
&\mathbf{w}\cdot(VDB\mathbf{u}) \to \text{max w.r.t. } \mathbf{u} \text{ and } \mathbf{p};\\
&K\mathbf{u} = -NA\mathbf{p},\\
&\Sigma NA\mathbf{p} = 0, \Sigma TNA\mathbf{p} = 0,\\
&\Sigma_v\mathbf{u} = 0, \Sigma_v T_v\mathbf{u} = 0,\\
&0 \le p_i \le p_{max} \text{ for all } i,\\
&\Sigma_s A\mathbf{p} = F_{tot},
\end{aligned}
\tag{4.12}
$$

where Σ_s sums scalars on the surface, Σ_v sums vectors in the volume Ω, and T_v computes torsion for each point. The total dimension of the problem is $n_b + 3n$.

Eliminating displacements. As most of the degrees of freedom in the system are displacements, but the quantities of interest are pressures $\mathbf{p}$, eliminating $\mathbf{u}$ results in significant speedups ($\mathbf{u}$ can be eliminated even for the displacement maximization problem). The elasticity equation $K\mathbf{u} = -NA\mathbf{p}$ is not sufficient for this; it has a nullspace of dimension 6 corresponding to the rigid motion degrees of freedom, so we need to consider the constraints for zero total rigid motion, $R\mathbf{u} = 0$, where $R = \begin{bmatrix} \Sigma_v \\ \Sigma_v T \end{bmatrix}$. Rewriting this system in the standard constrained system form,

$$
\underbrace{\begin{bmatrix} K & R^T \\ R & 0 \end{bmatrix}}_{C^*} \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} NA\mathbf{p} \\ 0 \end{bmatrix}, \tag{4.13}
$$

where λ is the Lagrange multiplier for the constraint $Ru = 0$. It is clear from physical considerations that this system is invertible. Let S be the selection matrix $\begin{bmatrix} I_{3n\times 3n} \\ 0 \end{bmatrix}$. Then, we can express $\mathbf{u}$ as $\mathbf{u} = S^T C^{*-1} SNA\mathbf{p}$. In this form, the objective of (4.12) becomes

$$\mathbf{w}\cdot VDB\mathbf{u} = \underbrace{\mathbf{w}^T VDBS^T C^{*-1} SNA}_{\mathbf{f}^T}\mathbf{p} = \mathbf{f}^T\mathbf{p}.$$

The displacement-free optimization problem is

$$\begin{aligned} &\mathbf{f}^T\mathbf{p} \to \max \ \text{ w.r.t. } \mathbf{p}, \\ &\quad \Sigma NA\mathbf{p} = 0,\ \Sigma TNA\mathbf{p} = 0, \\ &0 \le p_i \le p_{max} \text{ for all } i, \\ &\Sigma_s A\mathbf{p} = F_{tot}. \end{aligned} \tag{4.14}$$

While the final system has only sparse constraint matrices, it may appear that computing $\mathbf{f}^T$ for the objective functional requires inverting C^*; we observe however that $\mathbf{w}^T VDBS^T C^{*-1} SNA = \mathbf{f}^T$ can be rewritten as $\mathbf{f} = (SNA)^T\mathbf{q}$, where $\mathbf{q}$ is the solution of the equation

$$C^{*T}\mathbf{q} = SB^T D^T V^T \mathbf{w}. \tag{4.15}$$

In other words, it is sufficient to be able to solve a linear system with matrix C^*, and the cost of transforming (4.12) to (4.15) is the cost of a single linear solve.

Finally, for modal analysis, we have observed that the results for isotropic models in particular are well-approximated by simpler eigenanalysis of the Laplacian, which yields a considerable speedup (compare the bottom two rows of Figure 4.6).

Algorithm summary and parameters. The main steps of our approach are

1. Compute a tetrahedral mesh Ω for an input triangle mesh.

2. Compute M_m modes using an eigensolver.

3. For each mode, find M_r weak regions with highest total energy.

4. For each weak region, solve (4.14) to obtain worst-case pressure candidate p_i.

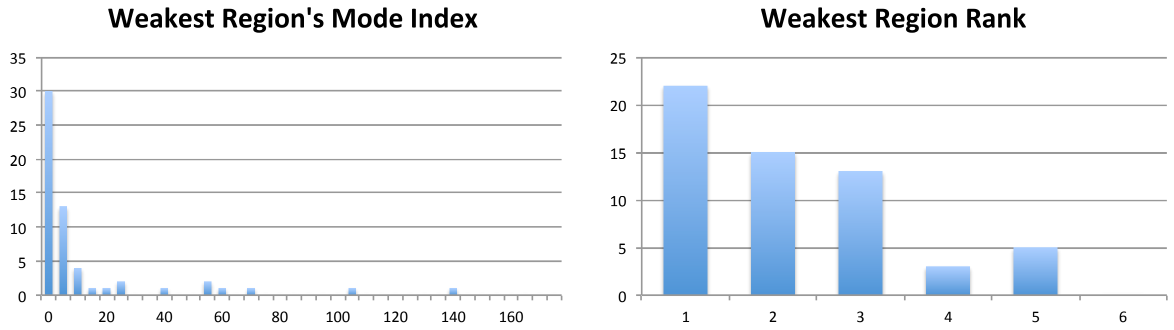


Figure 4.3: Left: histogram of the mode number (horizontal axis) in which the weakest region appears for the first time. Right: histogram of the rank of the weakest region in the weak region list sorted by decreasing energy.

5. Solve $Lu = 0$, with boundary pressures specified by p_i, to obtain displacements u_i, and compute actual maximal principal stress σ_i^{max} for each weak region.

6. Maximal stress is determined as maximum of σ_i^{max}.

Tetrahedral meshes are generated using `tetgen` ([91]).

We use MOSEK ([72]) to solve the linear programming problem, UMFPACK ([30]) for linear solves, and ARPACK ([59]) for computing eigenvectors and eigenvalues.

The parameters of the algorithm include M_m, M_r, the choice of threshold $1 - \epsilon$ for weak regions, and a user-defined maximal pressure p^{max} (the latter can be regarded as a part of the definition of the force model).

To determine reasonable choices of M_m and M_r, we have run modal analysis for a large number of modes (150) and a large number of weak regions per mode for a collection of objects. For each object, we found the weakest region and checked in which mode it first appears. We also computed the weakest region's rank in the list of that mode's weak regions sorted by decreasing energy. The results (Figure 4.3) indicate that considering 15 modes and 5 weak regions per mode is sufficient in over 80% of cases.

We use $\epsilon = 0.025$ in all cases; the dependence of the size of weak regions for one mode on ϵ is shown in Figure 4.4.

Figure 4.5 shows two final results of the algorithm. Red arrows are total forces obtained by summing nearby per-vertex force values (pressures are typically concentrated in small areas). Color maps on the deformed surfaces show weakness maps.

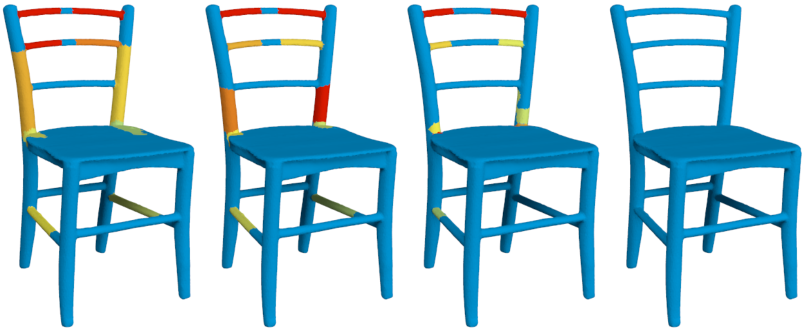


Figure 4.4: Weak regions extracted from three modes with weakness level cutoff, $\epsilon = .10, .05, .03, .01$.

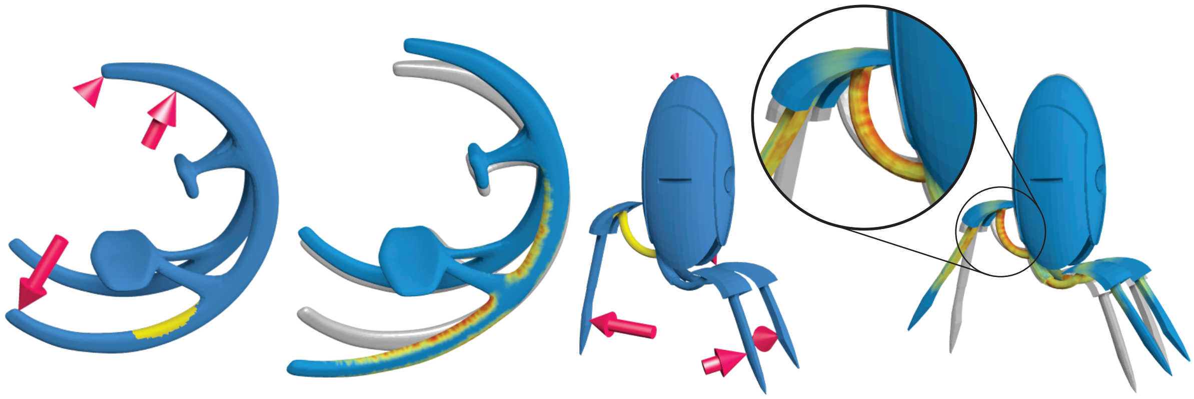


Figure 4.5: Optimal force vectors and weakest regions on the left, resulting deformations and stresses on the right. The gray images in the background show the undeformed state.

4.5 Validation

We validated our algorithm in several computational and experimental ways.

Comparison with direct search for the weakest region. Instead of using the modal analysis stage to identify weak regions and using averaged stress or displacement over weak regions as the target quantity to optimize, we can run the same optimization process directly, treating each tetrahedron as a potential weak region.

We define the *weakness* map as a scalar field on the surface mapping each point to the maximal principal stress at this point obtained by approximate optimization. Using our method yields a partial weakness map on the union of all weakness regions we consider. Figure 4.6 shows a comparison of a complete weakness map, computed using the brute-force approach, with the weakness map obtained by our method. We observe a close agreement between these for all examples in areas where the partial map is defined and never observe high stress values elsewhere.

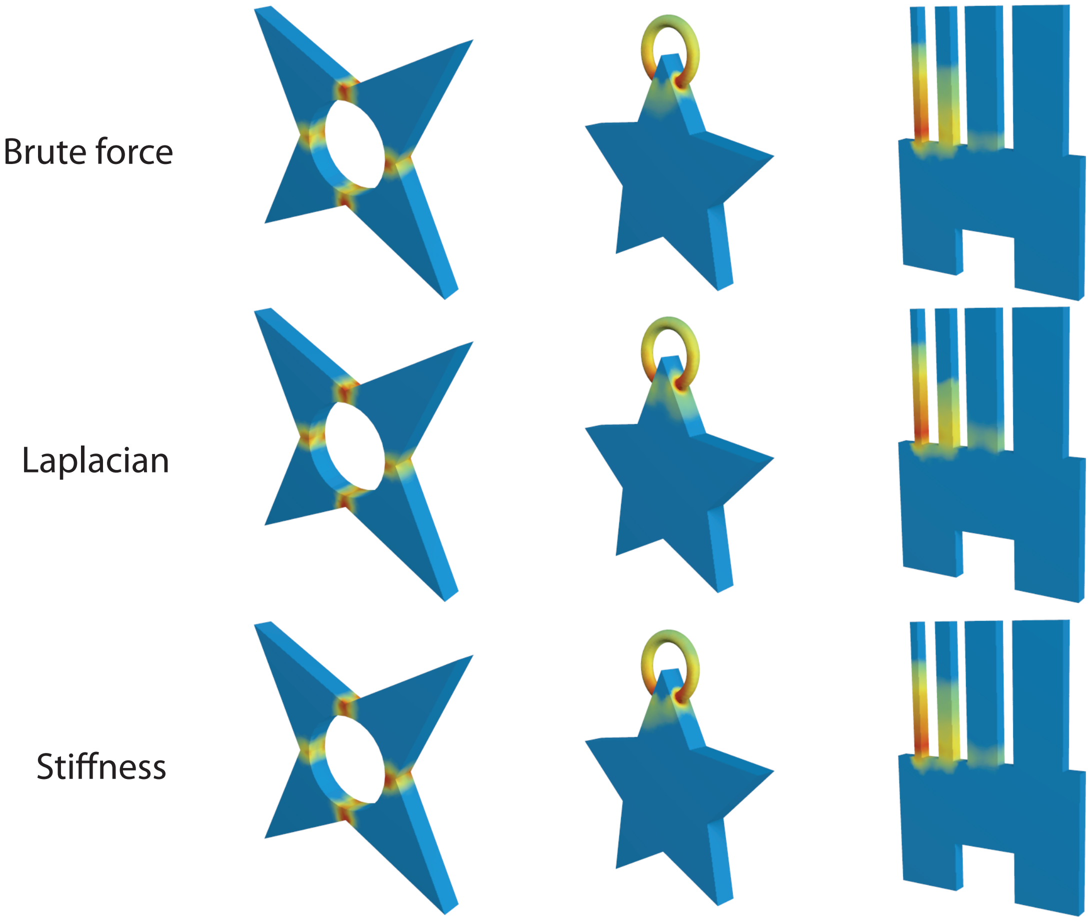
# Tets	Brute Force (s)	Weak Region (s)	Speedup
2723	681.367	1.089	625.939 x
2869	793.362	1.087	729.907 x
2904	894.610	0.641	1396.071 x
5332	2120.361	1.171	1810.199 x
11020	11029.721	2.729	4042.403 x
12853	11334.362	1.694	6692.546 x
14163	27775.900	3.373	8234.925 x
16008	19917.838	1.892	10527.388 x

Table 4.1: Stress analysis timings for brute force optimization vs. weak region optimization. While speedups are already dramatic for extremely small element counts, the higher asymptotic complexity of brute force causes a rapidly increasing speedup for larger models.

Dependence on tetrahedral mesh resolution. To keep the cost of computation low, especially in the context of interactive applications or processing large number of objects at a printing facility, using coarse tetrahedral meshes is desirable. As Figure 4.7 shows, weakness maps for different resolutions are similar, so higher resolution may be used only at the last stage, after the weakest spots are identified.

Drop test. To verify our method for brittle materials, we performed a randomized deformation test by dropping printed models onto horizontal pegs. We dropped the models from 1m high, ensuring a nearly random impact orientation and force application. The test setup is pictured in Figure 4.8. All models were printed with material zp150.

Specific breakages may have two origins: high point forces, which can break even



Analysis Type	Star	Star Pendant	“Test 2”
Brute Force	18.300 MPa	36.767 MPa	73.298 MPa
Laplacian	15.593 MPa	34.151 MPa	71.689 MPa
Stiffness	16.208 MPa	35.939 MPa	70.588 MPa

Figure 4.6: Comparison of the similar optimum stresses found by brute force, Laplacian-based weak region analysis, and stiffness-based weak region analysis. The table reports 99.75 percentile (by volume) element stresses. An isotropic metal material was used for this comparison.

relatively strong spots near the impact point and will vary across drops, and smooth deformations, which are likely to break weak regions consistently. The former does not correspond to typical usage scenarios, which feature distributed bounded forces. Thus, we consider only fractures occurring frequently across drops.

The test results, displayed in Figure 4.9, confirm that the weak regions determined by our method generally agree with the areas with highest occurrence of fracture. Notice in particular the legs of the cow (3^{rd} row, left), the notches of the gear (5^{th} row, left),

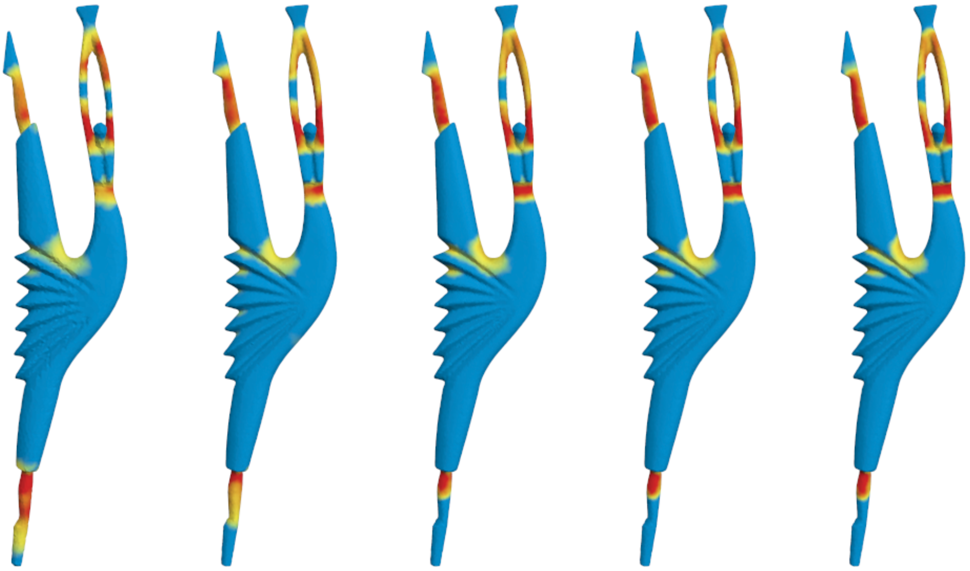


Figure 4.7: For 5 different mesh resolutions (from left to right, the vertex counts are 5K, 13K, 24K, 36K, 50K), the algorithm generates consistent weakness maps.

the arms of the dancer (1^{st} row, right), and the inner piece of the powercog pendant (6^{th} row, left). These are all regions of high weakness map value that break consistently.

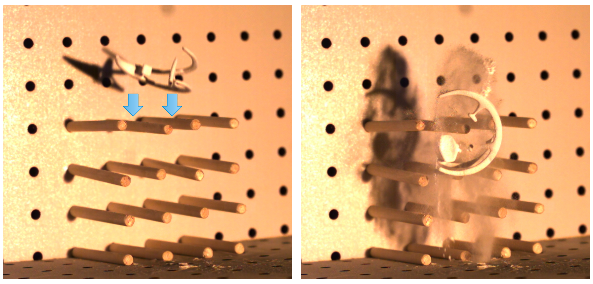


Figure 4.8: We used models printed in green state "sandstone" for the drop tests. The testing models often are covered with a loose layer of powder that shakes off upon impact (see the dust in the right image).

Displacement test. For the objects printed in ductile materials, we performed a different test. We placed the shapes into a cardboard box filled with packaging material and applied pressure to the box's exterior. This pressure permanently deformed the models inside. We took photographs of the deformed models in a registered position and compared them to the 3D model from which they were printed. We observe good agreement with the computed map of maximal displacements, i.e., the map similar to the weakness

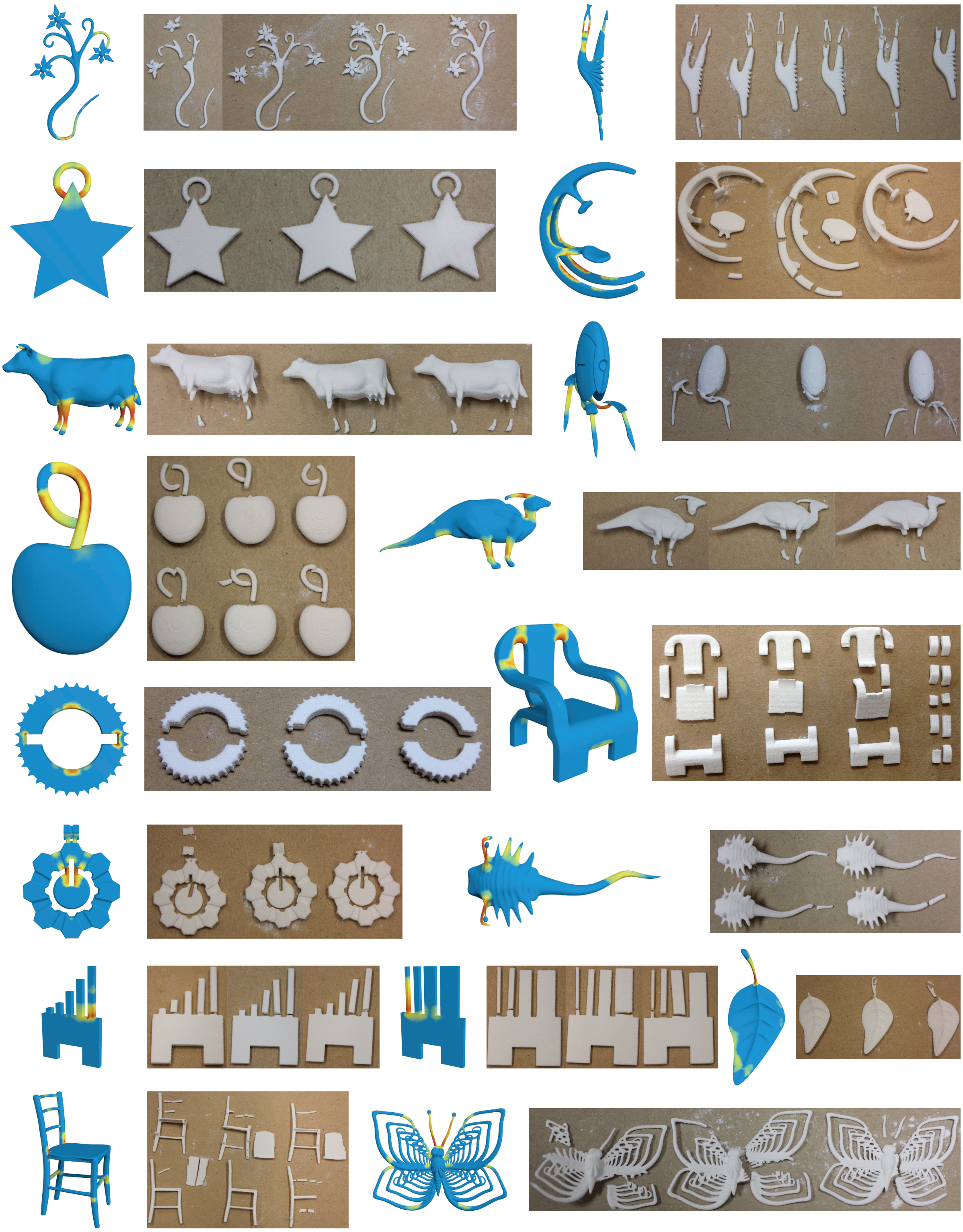


Figure 4.9: Results for a drop test. Model volume is shaded with its weakness map percentile: 90% 99%

map, but for the displacement maximization problem (see Figure 4.10).

Comparison with [97]. We compare to the approach described in [97], as they also aim to predict the loads that a printed model is likely to experience. The authors use a more specific force model: pinch grips. They present an empirical model to predict how the object will be gripped with two fingers. There are many designs for which such a grip does not capture typical use cases or mishaps. Figure 4.11 demonstrates shapes whose worst-case loads cannot be applied or approximated using only a pinch grip.

Figure 4.12 shows three examples for which the authors of [97] have provided us their force application points. Their "cup" example (left) is an excellent candidate for the pinch grip; the highest stress achieved with a fixed total force agrees with ours and even exceeds it. However, the other two objects do not fit their model as well. The "UFO" pinch grip is clearly suboptimal, and the forces applied to the bracelet would have much more leverage if they were moved to the open endpoints. In all three cases, our method generates efficient force vectors.

An interesting observation about the "cup" model is that our method produces a triangle of forces (perhaps at the expense of higher stress) rather than a pair of opposite forces. One possible reason for this is the pressure bound requiring the force to be distributed over a larger area.

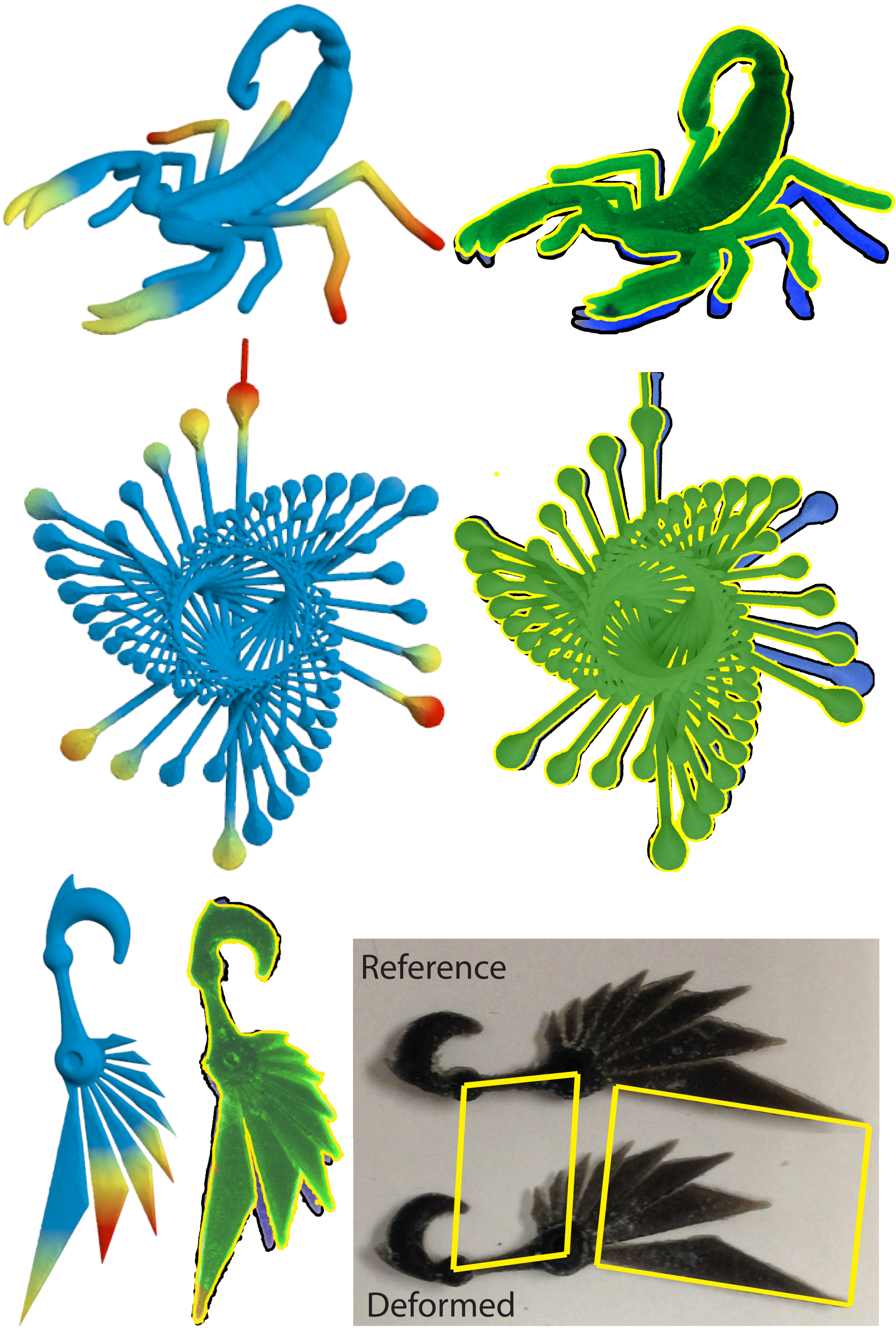


Figure 4.10: Simulation results (left) are compared to the deformed 3D printed model (green) overlaid on an undeformed one (blue). Our algorithm predicts likely regions (red) of large deformation under normal handling. For the blade earring, we confirm that the largest blade deforms little relative to the hook and shaft: after aligning the shafts to be parallel, the largest blades are also roughly parallel (see the yellow parallelograms). The second largest blade is displaced more. Note that the upper right pin of the deformed spinnoloid (middle row, green) was broken during printing.

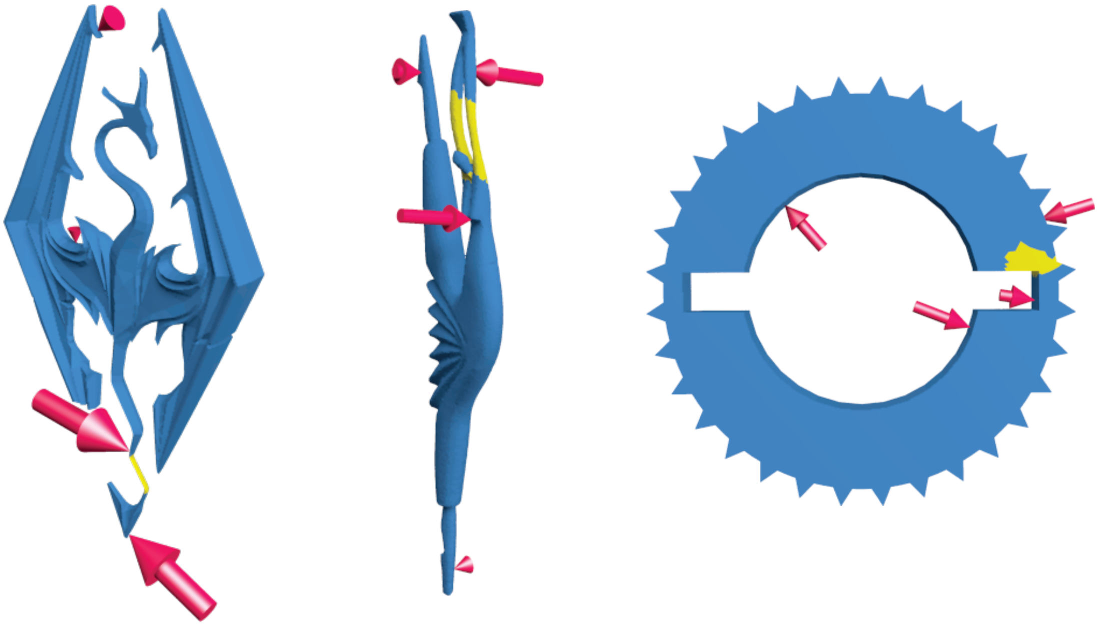


Figure 4.11: Models where pinch grips cannot generate worst-case loads. Our method finds highly intuitive force vectors, regardless. The additional arrows on the top of the Skyrim dragon arise to bring the total force and torque to zero.

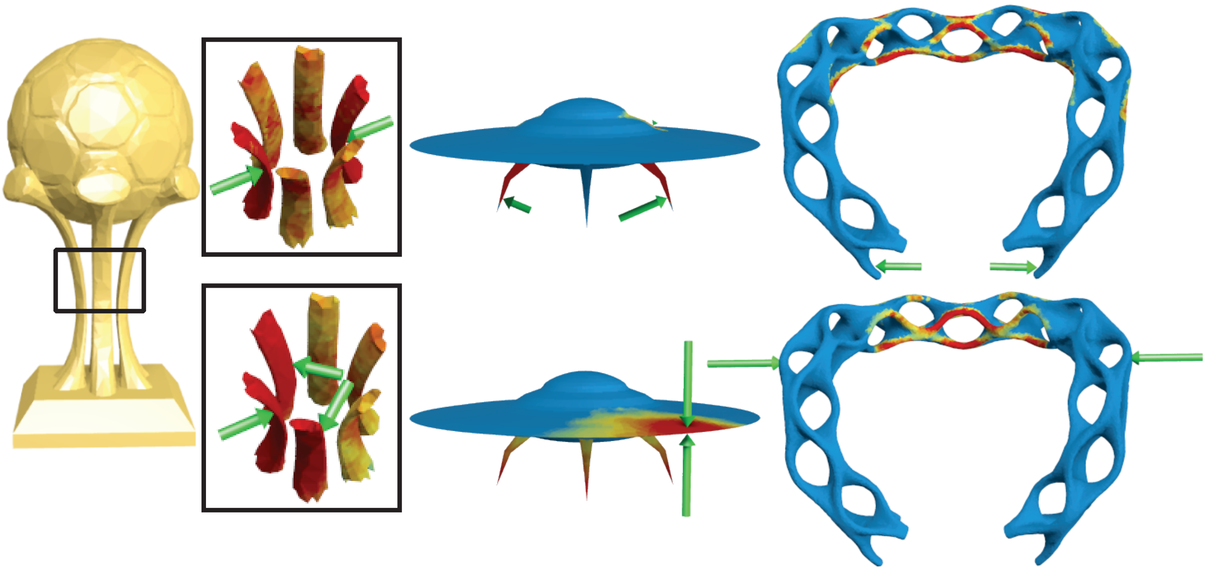


Figure 4.12: Comparison against [97]. Our algorithm's force distribution (top) better identifies structural weakness, especially for the ufo (middle) and bracelet (right).

Timings. Though our pipeline has not yet reached interactive speeds, it is already fast enough to be included in a 3D printing pipeline. For the sizes of models most commonly sent to 3D printing services (see distribution in Figure 4.14: sizes on the order of 100K vertices are most common), our full algorithm takes only a few minutes (right).

Tets	Analysis Time (mins)
2723	0.028
42900	0.308
70356	0.382
155383	2.566
322398	9.601
414894	4.490

Analyzing the algorithm's scaling behavior is complicated by its dependence on structural properties—a separate linear program is run for each weak region that is extracted. To make sense of the timings, they have been separated by stage. Modal analysis and weak region extraction are run only once per model, and Figure 4.13 shows how their execution times depend on element count. The time spent setting up and solving the linear programs ("weakness analysis") is averaged over all weak regions so that it can be plotted against the same x axis. Note that there is one further cost not shown: the single sparse UMFPACK factorization. This timing depends strongly on matrix structure (despite using fill-in reducing permutations), and adds noise to curves when included. Factorization time is included in the timing table.

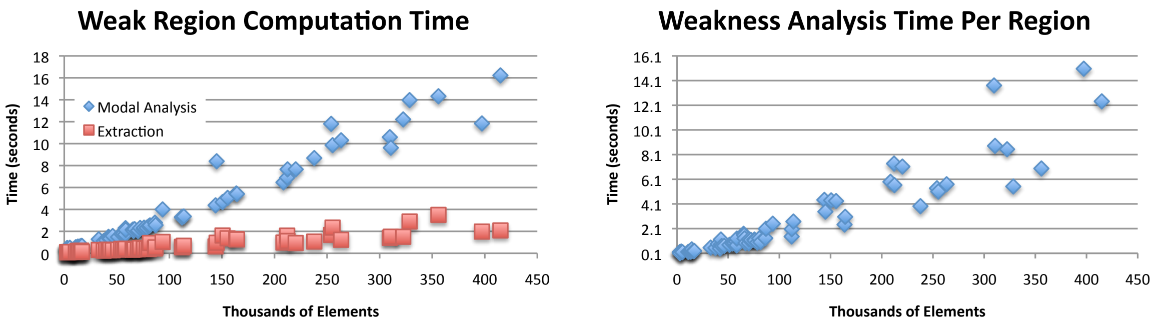


Figure 4.13: The top plots shows how modal analysis and weak region extraction scale with the number of tetrahedra. The dominant cost is the eigensolve. The bottom plots shows the average cost of setting up and running the linear program for each weak region. It excludes the UMFPACK factorization of C^* that only must be run once.

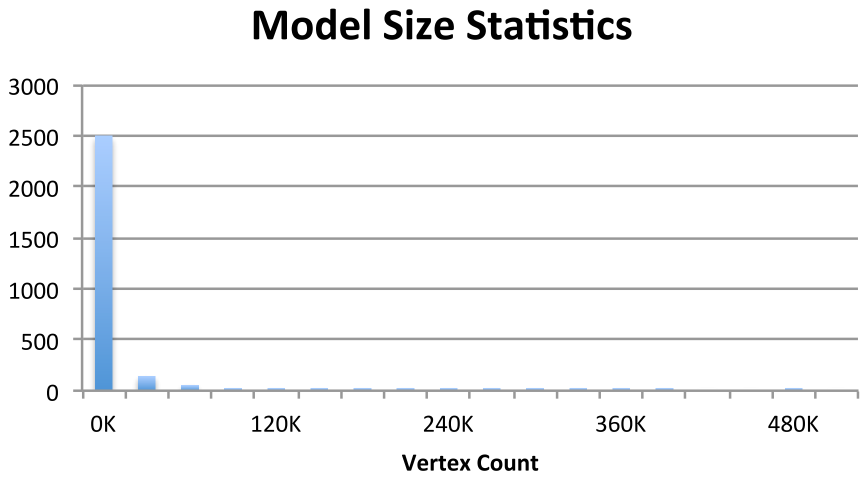


Figure 4.14: Model vertex counts tabulated from 2781 models ordered from Shapeways.

4.6 Material properties

Material parameters defining the elasticity tensor C must be measured for each of the 3D printers' materials. We have observed that the computed maximal stress does not depend on the magnitude of the Young's modulus in the isotropic case. However, in the anisotropic case, it does depend on the ratios of directional elasticity moduli, which can be significant (Figure 4.15). To predict breakage or plastic deformations under loads, the additional material parameters tensile strength and yield strength are needed.

In this section, we present the Youngs modulus ratio measurements for three different 3D printing materials that we used to compute our simulation's elasticity parameters. In addition, we discuss the extent to which various materials match our assumptions on stress-strain linearity and what accuracy one can expect from predictions of the maximal stress to tensile strength ratio. In all cases, we assume a Poisson ratio of 0.3.

We have tested three materials used in 3D printing: nylon (PA 2200 by EOS Electro Optical Systems), "sandstone" (zp150 in the ZPrinter series by 3D Systems), and green state stainless steel (420SS powder bound with proprietary binder used by ExOne). They also represent different classes of materials (brittle vs. ductile, isotropic vs anisotropic).

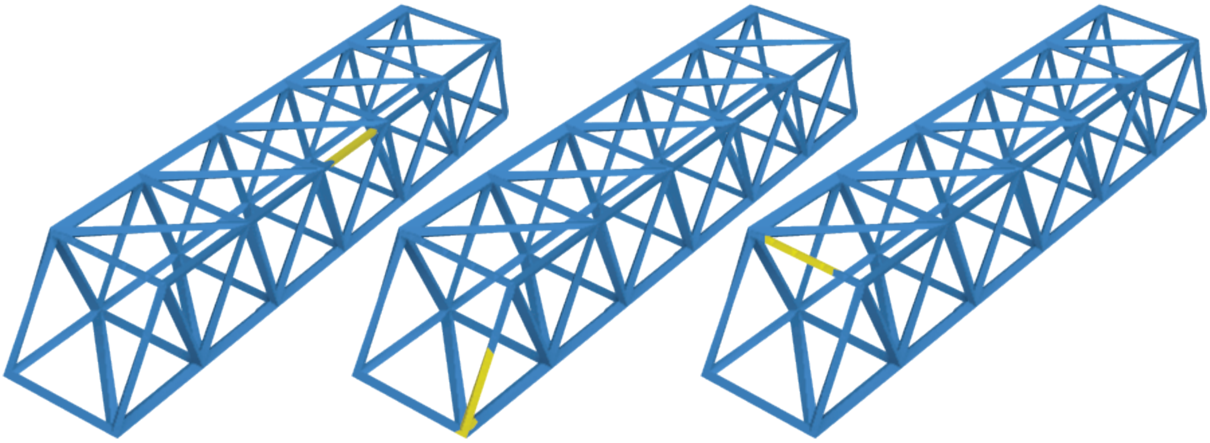


Figure 4.15: Different ratios of directional Young's moduli can lead to different weakest regions. We show the weakest region found for a truss with a Young's modulus that is five times higher in the X (left), Y (middle), and Z (right) directions.

To determine their properties, we conducted three point bending tests consistent with ASTM standard D5032 ([8]) using the Instron 5960 universal testing machine with a 100N load cell and a support span of 40mm. Figure 4.16 illustrates the testing setup. The testing samples are rectangular bars with length 60mm and thickness between 1mm and 5mm. We chose relatively thin test bars because structurally weak models are likely to contain thin features.

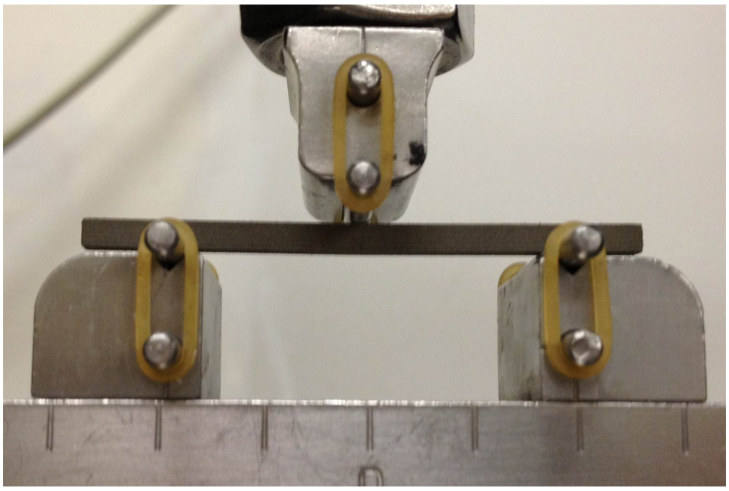


Figure 4.16: Three-point bending test on green state stainless steel.

Of the three materials tested, green state stainless steel best fits the definition of a brittle material. Stress grows linearly with strain for all samples tested until fracture

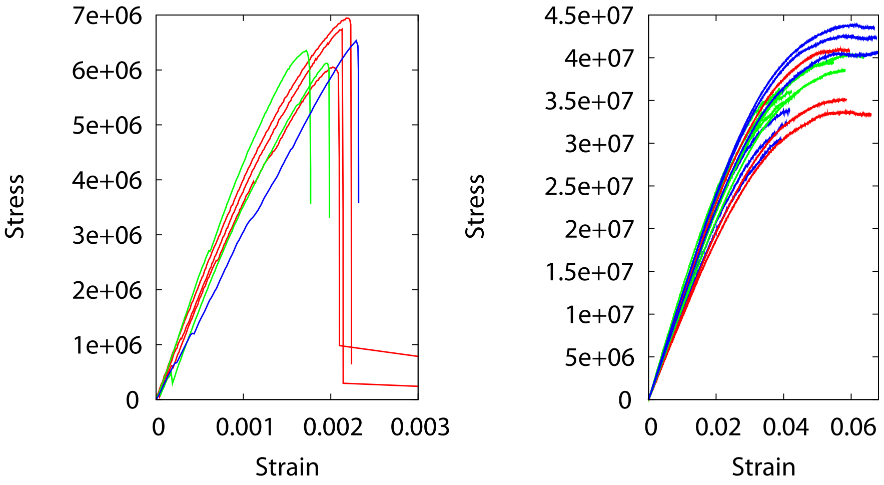


Figure 4.17: Left: Stress vs strain curve measured on samples in green state stainless steel. The colors indicate different sample thickness (1.5mm red, 2mm green, 3mm blue). Right: Stress vs strain plots for nylon testing samples of thickness 1.5mm and 2mm. The samples printed in different orientation are marked with different colors (red: X, green: Y, blue: Z).

(Figure 4.17 left). Bending tests in perpendicular directions show that elastic moduli in these directions are close, with the average Young's modulus 3.59GPa and standard deviation 0.27GPa. Figure 4.18 shows critical stress extracted from measurements, which is mostly consistent across all samples, with the average 6.88MPa and 0.62MPa standard deviation. Overall, this material is consistent with our model for stress optimization.

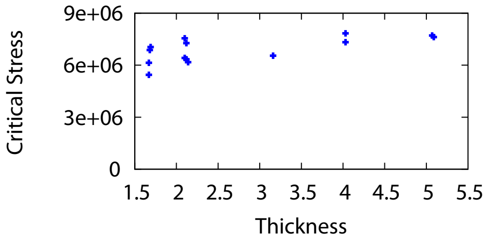


Figure 4.18: The critical stress distribution of green state metal for samples with thickness 1.5mm up to 5mm.

Models printed in nylon are known to withstand a large range of deformations. Figure 4.17 (right) shows the stress vs strain curve for 18 nylon samples. Half of them are 1.5mm thick, and the other half are 2mm. For each thickness group, we printed sets of 3 samples along X,Y and Z directions. From the results, we observed that nylon samples typically have a very large elastic deformation range before entering the plastic stage. We also note a moderate degree of anisotropy (the Young's modulus is 0.80GPa with 0.13GPa deviation for the X samples, 1.02GPa with 0.18GPa deviation for Y, and 0.98GPa with 0.12GPa deviation for Z). See Figure 4.17, right.

The most complex material we tested is the "sandstone" (zp150). Though, like green state metal, it has a relatively low tensile strength, it exhibits a significant plastic region (Figure 4.19) and very high degree of anisotropy: we measured X, Y, and Z Young's moduli of 1.22GPa (standard deviation 0.13GPa), 0.68GPa (standard deviation 0.07GPa), and 0.234GPa (standard deviation 0.02GPa) respectively, noting a greater than 5× difference between the largest and smallest values. Thus, we model zp150 as an orthotropic material with a distinct Young's modulus per printing axis. We obtain our shear moduli using a standard formula from [95]: $G_{xy} = \frac{E_x E_y}{E_x + E_y + 2E_y \nu_{xy}}$. Note that zp150 exhibits a large variance of tensile strength, even for a single direction. This means that only very conservative predictions are possible. Nevertheless, we observe that our weak region detection works well (Figure 4.9).

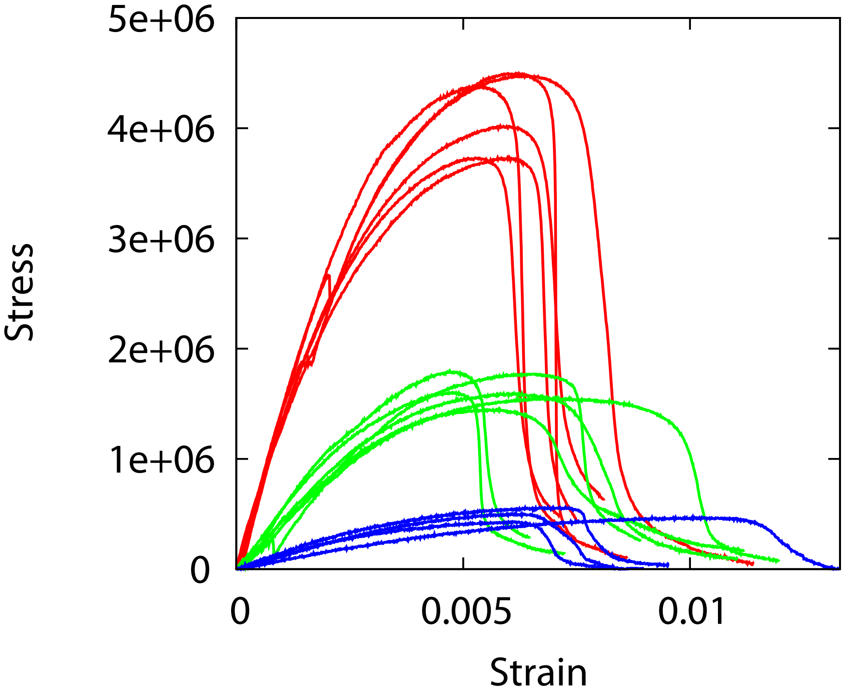


Figure 4.19: Stress vs. strain measurements on rectangular bars printed with green state "sandstone" (zp150) along the printer X (red),Y (green) and Z (blue) direction. The printing direction significantly influences the material properties.

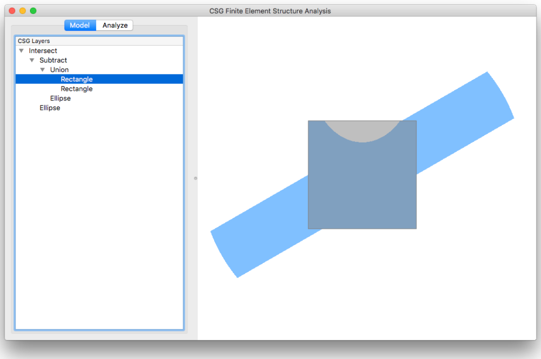
4.7 Mesh-Free Implementation

From the viewpoint of robustness, tetrahedral mesh generation is the bottleneck: volume meshing fails for the ill-behaved (self-intersecting, non-manifold) geometry often present in users' designs. Furthermore, for the purposes of shape optimization—e.g. to relieve the worst-case stresses found by our algorithm—creating a volume mesh to evaluate each iterate slows down the optimization process and makes it brittle.

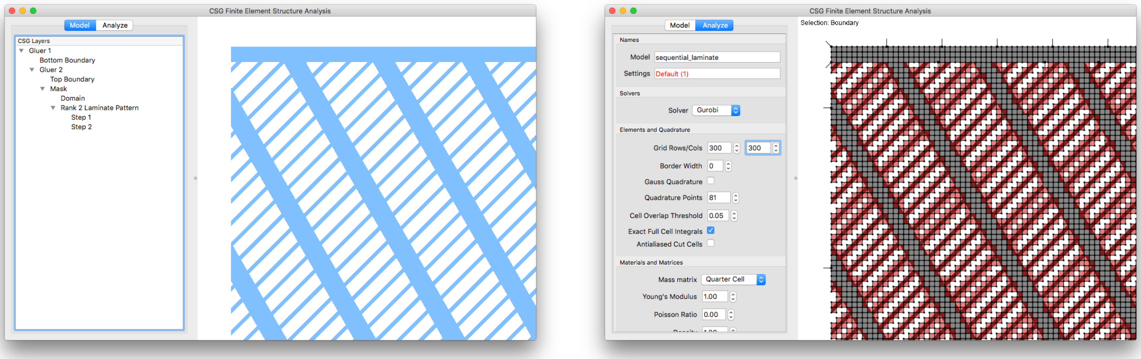
To avoid the volume meshing stage, we implemented the worst-case analysis pipeline in a custom-built mesh-free finite element framework (supporting both 2D and 3D). This pipeline can be applied to any geometry representation supporting an efficient inside/outside query. It is best if the representation also explicitly provides a set of well-spaced boundary points and associated areas/normals (though these can be approximated from the inside/outside test if necessary).

For 2D, we implemented a simple interactive constructive solid geometry (CSG) modeling tool in which worst-case analysis can be applied as the user manipulates the shape.

4.7.1 CSG Modeling



The simple modeling tool allows one to insert various geometry "primitives" (e.g. rectangles and ellipses controlled by size, position, and rotation parameters) and combine them with boolean operations (union, intersection difference). Other special-purpose primitives were implemented, such as (approximate) rank-two sequential laminates:



Names

Model: sequential_laminate

Settings: Default (1)

Solvers

Solver: Gurobi

Elements and Quadrature

Grid Rows/Cols: 300 300

Border Width: 0

Gauss Quadrature: □

Quadrature Points: 81

Cell Overlap Threshold: 0.05

Exact Full Cell Integrals: ☑

Antialiased Cut Cells: □

Materials and Matrices

Mass matrix: Quarter Cell

Young's Modulus: 1.00

Poisson Ratio: 0.00

Figure 4.20: Left: zoomed-in view of the corner of a square tiled with the rank-two sequential laminate primitive. Right: the computation grid used to simulate the structure.

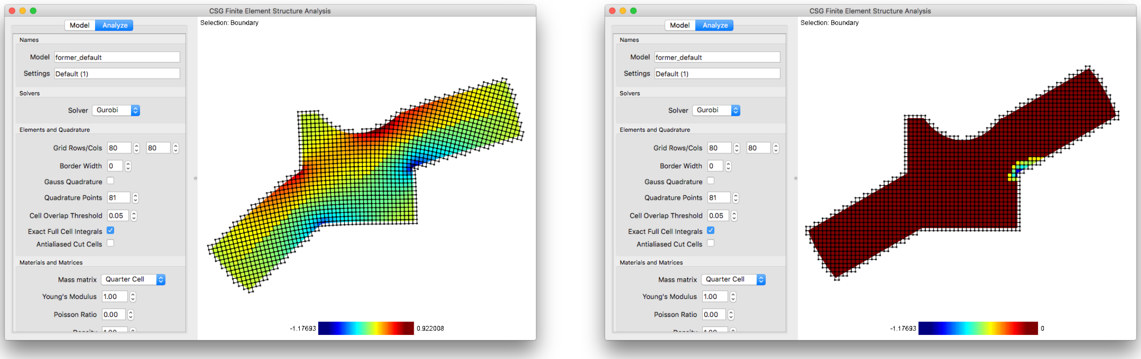
4.7.2 Mesh-Free FEM and Worst-Case Analysis

Shape Functions and Stiffness Matrices

The mesh-free finite element framework works by fitting a regular grid of some chosen resolution to the bounding box of the input geometry. Then a bilinear (2D) or trilinear (3D) element is created for each grid cell.

For grid cells fully overlapping the object (dark gray in Figure 4.20), the standard linear elasticity per-element stiffness matrix for bilinear and trilinear rectangle/rectangular prism elements is used. For grid cells only partially overlapping the object (highlighted in red), the per-element stiffness is computed by integrating the stiffness density only over the portion of the element overlapping the geometry (approximating the integral with a large number of quadrature points). We compute the FEM mass matrix, M, similarly: we integrate the products of pairs of shape functions over only the portion of elements overlapping the geometry.

Cells fully outside the object—or with too few quadrature points falling inside—are removed from the simulation entirely. With the stiffness and mass matrices K and M constructed, we can apply modal analysis by solving the generalized eigenvalue problem $K\mathbf{u} = \lambda M\mathbf{u}$ and extract weak regions as described in Section 4.4. Example results are visualized in Figure 4.21.


CSG Finite Element Structure Analysis

Model | Analyze

Selection: Boundary

Names

Model: former_default

Settings: Default (1)

Solvers

Solver: Gurobi

Elements and Quadrature

Grid Rows/Cols: 80 80

Border Width: 0

Gauss Quadrature □

Quadrature Points: 81

Cell Overlap Threshold: 0.05

Exact Full Cell Integrals ☑

Antialiased Cut Cells □

Materials and Matrices

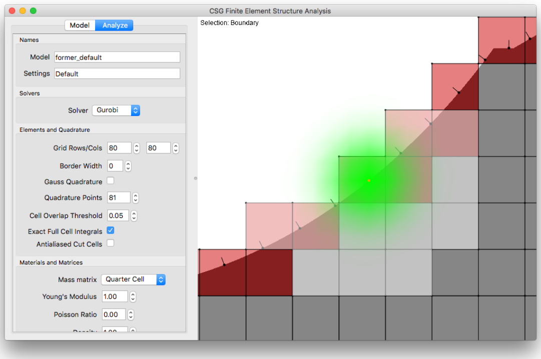
Mass matrix: Quarter Cell

Young's Modulus: 1.00

Poisson Ratio: 0.00

Figure 4.21: Left: the lowest energy vibrational mode computed by eigenanalysis. Right: the weakest region extracted from this mode.

Surface Forces


CSG Finite Element Structure Analysis

Model | Analyze

Selection: Boundary

Names

Model: former_default

Settings: Default

Solvers

Solver: Gurobi

Elements and Quadrature

Grid Rows/Cols: 80 80

Border Width: 0

Gauss Quadrature □

Quadrature Points: 81

Cell Overlap Threshold: 0.05

Exact Full Cell Integrals ☑

Antialiased Cut Cells □

Materials and Matrices

Mass matrix: Quarter Cell

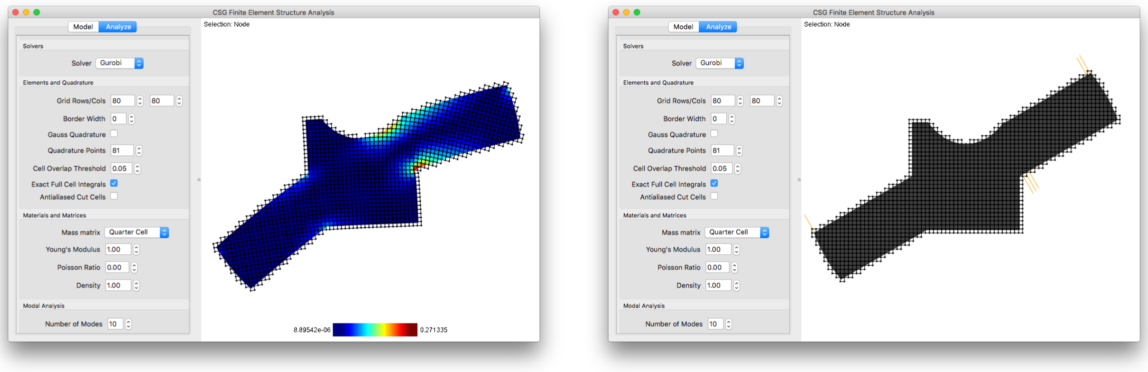
Young's Modulus: 1.00

Poisson Ratio: 0.00

Figure 4.22: Forces are defined on the boundary points and smeared onto the FEM nodes using a radial cubic B-spline (green) with finite support (highlighted).

To run the worst-case stress optimization over pressure fields, we need a discretization of surface forces (more specifically, pressures). We discretize surface forces by a vector assigned to each boundary point. These forces are smoothed into the volume with a radial cubic B-spline (with finite support, overlapping only a few cells for efficiency) so that per-grid-node loads can be computed by integrating the blurred force against the nodal shape functions over the object geometry. We normalize the resulting linear map from boundary point forces to nodal loads to ensure the exact, full force magnitude is distributed to the grid nodes (to counter the fact that only part of the B-spline's support overlaps the object).

With surface forces discretized, we can now construct the linear program (4.14). The linear program assumes roughly the same form as for the tet-mesh case, but there is an additional matrix appearing in the formula for **f** that implements the boundary-point-to-grid-node force smoothing.


CSG Finite Element Structure Analysis
Model
Analyze
Selection: Node
8.89542e-06
0.271335
CSG Finite Element Structure Analysis
Model
Analyze
Selection: Node

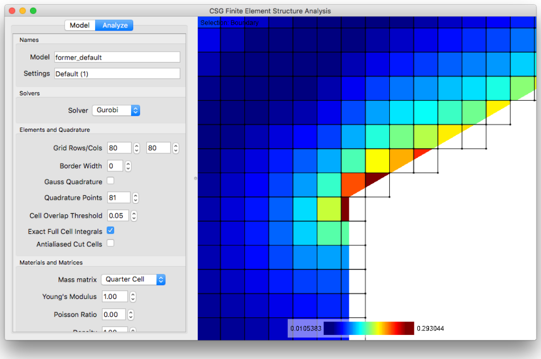
Solvers
Solver: Gurobi
Elements and Quadrature
Grid Rows/Cols: 80 80
Border Width: 0
Gauss Quadrature: □
Quadrature Points: 81
Cell Overlap Threshold: 0.05
Exact Full Cell Integrals: ☑
Antialiased Cut Cells: □
Materials and Matrices
Mass matrix: Quarter Cell
Young's Modulus: 1.00
Poisson Ratio: 0.00
Density: 1.00
Modal Analysis
Number of Modes: 10

Solvers
Solver: Gurobi
Elements and Quadrature
Grid Rows/Cols: 80 80
Border Width: 0
Gauss Quadrature: □
Quadrature Points: 81
Cell Overlap Threshold: 0.05
Exact Full Cell Integrals: ☑
Antialiased Cut Cells: □
Materials and Matrices
Mass matrix: Quarter Cell
Young's Modulus: 1.00
Poisson Ratio: 0.00
Density: 1.00
Modal Analysis
Number of Modes: 10

Figure 4.23: Left: the deformation inducing the worst-case stress in this object. Right: the corresponding forces (found by solving a linear program).

4.7.3 Mesh-Free FEM Limitations

While this mesh-free tool computes deformations accurately, proper handling of the cut cells is challenging and often leads to inaccurate boundary stresses (Figure 4.24). Unfortunately, the boundary is precisely where we need the greatest accuracy: it is where stress concentrates and also where quantities needed for shape optimization primarily live. Also, one must apply a heuristic threshold to decide when a cut cell should be discarded due to insufficient overlap with the object (keeping such low-mass cells leads to poor conditioning in the linear system). A better approach to handling the boundary—



Model: former_default
Settings: Default (1)
Solver: Gurobi
Grid Rows/Cols: 80 80
Border Width: 0
Gauss Quadrature: □
Quadrature Points: 81
Cell Overlap Threshold: 0.05
Exact Full Cell Integrals: ☑
Antialiased Cut Cells: □
Mass matrix: Quarter Cell
Young's Modulus: 1.00
Poisson Ratio: 0.00

Figure 4.24: Worst-case stresses computed on the cut cells at the boundary.

possibly using adaptive refinement and smoothing the geometry’s indicator function—is needed to produce accurate stress results and to permit shape optimization.

4.8 Conclusions

We have presented an efficient approximate method for determining worst-case loads for a geometric object based on its geometry and material properties only. The method is quite reliable (it relies on a linear solver, an eigensolver, and a linear programming solver, which all can provide convergence guarantees), efficient, and approximates well the worst-case stress and displacement distributions.

At the same time, there are numerous limitations. Most importantly, we only consider linear elasticity, so our maximal deformation results may not match reality for large plastic deformations. We note, however, that the robustly obtained approximate solution can serve as a starting point for a nonlinear solver. More generally, 3D printed materials exhibit a broad range of complex behaviors, some of which may exhibit considerable variation even for the same printing process. Using computational models reflecting material complexity and uncertainty is an important future direction.

Acknowledgements

We thank Professor Yu Zhang (NYU Dental School) and Professor Nikhil Gupta (NYU Poly), who allowed us to use their facilities for material testing and Professor Alan T. Zehnder (Cornell) for his helpful discussions regarding these tests. Finally, we thank Shapeways for initiating the project and providing raw materials.

Design credits for tested shapes: Stava et al.: soccer cup (Fig 13); improbablecog: flora (Fig 1,10), powercog (Fig 10), blade earring (Fig 11); Novastar Design: skyrim (Fig 12), heart (Fig 2); unellenu: butterfly (Fig 10), spinnoloids (Fig 11); kspaho: scorpion (Fig 11); TerryDiF: turret (Fig 6,10); Julia Boersma: trilobyte (Fig 10); Aim@Shape: cow (Fig 10), dilo (Fig 10), dancer (Fig 8, 10, 12), wood chair (Fig 5, 10).

NSF awards IIS-0905502, OCI-1047932, and DMS-0602235 partially supported this work.

Chapter 5

Worst-case Stress Relief for Microstructures

We address the problem of robust microstructure design to remedy the deficiencies of Chapter 3 for practical applications. All work presented in this chapter (and the paper on which it is based, [50])—from the problem formulation to the mathematical details, the software implementation, 3D printing, and physical lab testing—is entirely my own. Co-authors Abtin Rahimian and Denis Zorin on [50] contributed to the writing.

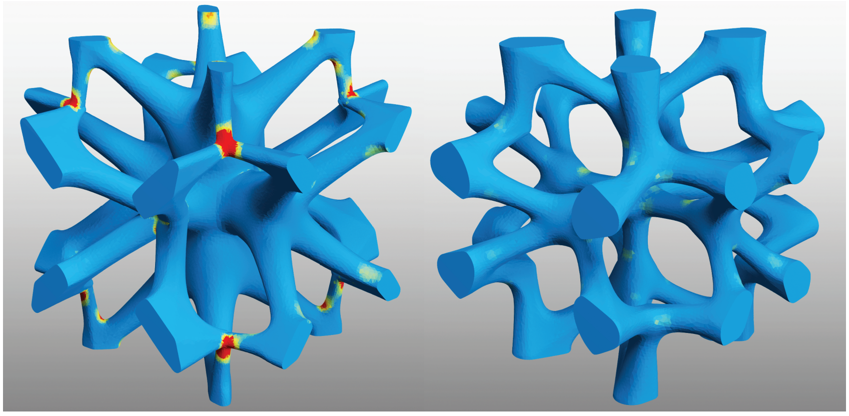


Figure 5.1: We design microstructures emulating a desired material while simultaneously experiencing minimized peak stresses under the worst-case load. Here we show two structures with identical macroscopic properties. Our optimization smoothes out the stress concentrations seen in red on the left, creating a robust, organic microstructure. The meshes are shaded with our novel worst-case micro-stress measure.

Additive fabrication technologies are limited by the types of material they can print: while the technologies are continuously improving, still only a relatively small discrete set of materials can be used in each printed object. At the same time, the low cost of introducing geometric complexity suggests the alternative of controlling the elastic material properties by producing *microstructures*, which can achieve behaviors significantly differing from the solid printing material. While promising results have been obtained in this direction, fragility is a significant problem blocking practical applications, especially for achieving soft material properties: due to stress concentrations at thin joints, deformations and repeated loadings are likely to cause fracture.

We present a set of methods minimizing stress concentrations in microstructures by evolving their shapes. First, we demonstrate that the worst-case stress analysis problem (maximizing a stress measure over all possible unit loads) has an exact solution for periodic microstructures. We develop a new, accurate discretization of the shape derivative for stress objectives and introduce a low-dimensional parametric shape model for microstructures. This model supports robust minimization of maximal stress (approximated by an L_p norm with high p) and an efficient implementation of printability constraints. In addition to significantly reducing stresses (by a typical factor of $5\times$), the new method substantially expands the range of effective material properties covered by the collection of structures.

5.1 Introduction

Most additive fabrication technologies share the distinctive feature that printing cost is primarily determined *by the amount of material used* and is generally independent of the object's *complexity.* These characteristics make it practical to fabricate small-scale structures, which can be used to fine-tune the object's deformation behavior and achieve more effective designs for classical problems such as maximizing a design's strength under particular loads while constraining its weight. More generally, spatially varying effective material properties, including exotic properties such as negative Poisson's ratio, can be achieved on even a single-material printer using fine-scale structures (microstructures).

Geometric complexity is typically associated with high curvature variation and negatively-curved regions. These surface features are well-known to generate high stress concentrations, which may preclude practical applications: even moderate loads may result in cracks or plastic deformations. While the task of distributing stress evenly is essential to many mechanical design problems, stress distribution acquires particular importance for complex geometry, and it is more difficult to address on an ad hoc basis. Optimal designs, minimizing the maximal stress norm subject to a set of constraints, tend to be smooth, "organic" free-form structures, as small surface variations often result in significant changes in local stress.

Shape modeling tools have addressed the problem of creating low-stress designs in a variety of ways, primarily by providing engineers with techniques to add fillets and blends. These approaches are based on the intuition that eliminating sharp concave corners improves stress behavior, and engineers tend to make specific choices of geometry based on prior experience and trial and error.

Shape and topology optimization provide a principled approach to solving this problem and have been used with success to optimize various functionals, most commonly compliance. However, success has been limited so far in minimizing max stress accurately and efficiently. Moreover, the typical setting for stress minimization is to specify a load the structure is required to support. In many cases, especially the case of microstructure design we consider, the loads are not known in advance, so *worst-case analysis* is needed in the optimization loop, i.e. determining at each iteration the loads causing the highest maximal stress.

In this paper, we focus on optimizing *microstructures*: assemblies of small cells, each filled with a pattern that, when periodically tiled, produces a particular averaged (homogenized) elastic behavior. Many of the methods we propose are relevant to other stress optimization problems.

Contributions In this paper, we describe a set of techniques for producing optimized microstructures that cover a broad range of isotropic material properties while at the same time experiencing minimized stress concentrations under the worst-case scenario. We also ensure these structures satisfy printability constraints for additive fabrication. Specifically, our contributions include:

- A novel exact microstructure worst-case stress analysis that expresses worst-case loads as eigenvectors of small tensors, incurring negligible computational cost beyond homogenization.
- An accurate and efficient approach to optimize a nearly nonsmooth large-p L_p norm stress objective approximating maximal stress. Combined with robust, adaptive meshing, this yields a method capable of reliably designing reduced-stress microstructures under the thousands of different target material property constraints from our microstructure database, using only tens of iterations per design.
- A parametric model significantly expanding material property coverage over previous work, while using a single topology and enabling substantial stress reduction (5x for most properties).
- A new blending operator for implicit surface modeling that eliminates bulging outside the convex hull. This is essential to efficiently enforce printability for our parametric model.

We demonstrate experimentally that our pipeline yields structures with significantly reduced stresses and, consequently, greater resilience to repeated loadings and stronger loads.

5.2 Related Work

Stress minimization Several works have focused on minimum stress design using shape and topology optimization. The closest work to ours is [5], which applies topology optimization to design lightweight minimal-stress objects built from sequentially laminated composites. It considers the stress concentrations occurring in the microstructure, but only for a fixed loading scenario (not the worst-case), and only for sequential laminates, microstructures whose properties have nice closed-form expressions but are not manufacturable. Another similar work is [4], which applies the level-set topology optimization method to minimize the p-norm of stress. [109] and [79] propose similar machinery but, to address numerical and convergence problems of p-norm minimization for high p, use different objective functions: the former adds a penalty term that activates for stresses above a specified threshold, and the latter switches to penalizing the deviation of stress from the average stress level. In the engineering literature, several works have considered fillet design with optimal shape, e.g., [96, 105], minimizing the stress concentration factor.

Our work differs from traditional shape and topology optimization approaches in several important ways: (i) we use exact worst-case analysis to determine the loads; (ii) we optimize a parametric model for which formulating fabrication constraints is efficient and exact; and (iii) our formulation for the shape derivative in combination with adaptive meshing, absent in level-set and SIMP-based formulations, yields much better accuracy when evaluating and differentiating stress.

A concurrent work, [60] uses a fully discrete approach to find 2D shapes with optimal L_p-norm of von Mises stress under known loads, treating all vertices as free variables (we use a low-dimensional parametric model instead). This method requires thousands of iterations in 2D compared to the 50 iterations typical in 3D for our method. It also cannot enforce our fabrication and tileability constraints.

In the computer graphics community, [97] presented several heuristic model correction techniques, such as thickening and strut insertion, to improve a structure's resilience to specific loading scenarios. [114] optimize the shell thickness to bound the von Mises stress on all vertices under a given load. This can be viewed as a similar type of optimization to ours, with the shell thickness serving as the shape parameter with a simple, direct relation to the effective material properties. Similarly, [116] bound the von Mises stress under specified force, while penalizing deviation from the input structure, operating on coarse meshes with linear elements and without remeshing.

Worst-case stress analysis In graphics, two papers have presented heuristics to analyze stress under unknown load. [115] approximately determine the most efficient pressure distribution to break or severely deform a structure by solving a large eigenvalue problem and many linear programs. [58] use a PCA-based Monte Carlo sampling to construct a stress probability distribution from a large number of force samples generated by a rigid body simulator. They use the failure probability as a constraint in a costly gradient-based topology optimization to reduce weight, but forbid the object's surface from changing to avoid differentiating the full stress analysis pipeline. In contrast, we derive an *exact*, efficient way to determine worst-case loads for microstructures and compute its exact derivative.

Periodic homogenization Homogenization is a central tool to our work, and our formulation is based on [3]. Homogenization has been used in graphics to reduce complexity of physical models in [52], which finds the constitutive parameters of a low resolution discretization that best approximates the behavior of the original, more complex object.

Microstructure design and optimization There is a large body of literature dedicated to theoretical studies of composites' effective moduli (our periodic structures are a limiting case of composites, combining a single material with free space), which was reviewed in [75]. Recent monographs on this topic include [29, 68, 100, 3, 26], which primarily focus on identifying microstructures with *extremal* behavior (having effective elasticity properties at the boundary of domain achievable by a given class of composites).

Microstructures have been designed using various forms of topology optimization [14, 15, 73], seeking periodic structures minimizing, e.g., compliance for a fixed total volume fraction. The result is normally a single-scale structure, with scale controlled by the design grid resolution or other types of regularization.

[86] and [75] propose methods to design tileable microstructures with printability constraints to fabricate deformable objects with spatially varying elastic properties. Our method follows the general shape optimization approach of [75], but introduces a different functional (making the elasticity tensor fitting a constraint), a different approach for computing shape derivatives, and a new, enriched parametric model.

In contrast to these works, [66] construct aperiodic, printable structures with pointwise control over Young's moduli, but cannot independently control the Poisson's ratio.

Fabrication constraints [81, 36, 57] have made recent progress in incorporating undercut/overhang angle constraints in the topology optimization framework. However, these method enforce the constraints only approximately, requiring parameter tuning, and add nonlinearities to the problem that hinder convergence ([36]). We impose printability constraints natively on our model's parameters similarly to [75]; this is made possible in our smoothness-enriched parametric model by our novel blending algorithm that avoids overhangs if none are present in the pre-blended structure.

Fabrication [62, 61, 47, 51] have demonstrated the fabrication of optimized microstructures in the context of bone scaffold and fusion cage design, and [39, 89, 22, 7] have demonstrated the possibility of manufacturing auxetic materials. The idea of manufacturing objects with spatially varying properties using tileable structures also appears in

[46].

[17] designs and fabricates objects satisfying an input deformation by optimizing for the best combination of stacked layers of their multi-material 3D printer's base materials. [94] applies discrete material optimization to achieve desired deformations of complex characters with actuation, fabricating the results with multi-material printing.

Blending Both parametric and implicit blending have received extensive attention in geometry modeling. A survey of parametric blending methods can be found in [106], and [63] discusses more recent work. A broad range of techniques have been proposed for implicit blending, starting with the foundational works [85] and [20]. The problems of bulging and (lack of) local control—characteristic of the simplest implicit blending constructions—are key topics of research. Bulging is an important problem in our context, as it leads to violation of printability constraints.

Clean union operations and related work can be used to generate smooth surfaces as offsets to the union of input surfaces [76], and can be made local by introducing blending volumes [77, 13, 16]. However, these method cannot easily control radius independently of blending for surfaces built from a 1-dimensional skeleton, and are restricted to binary operators. Convolution surfaces [21] are generally bulge-free; however, the degree of blending at vertices is difficult to control independently at different joints, including in a recent advanced version discussed in [111]. The approach of [84] reduces the bulging problem but may not produce smooth surfaces, and is difficult to localize. More recent work modulating the blending based on the angles between primitive normals [37] produces high-quality results but does not fully control bulging in the sense our application requires (we consider gradient-based methods in more detail in Section 5.6). A gradient-based approach was extended to N-ary operations in [112], but similar considerations apply. In our blending construction, we use the Kreisselmeier-Steinhauser function [56], which is commonly used in optimization and has been applied to implicit surface blending

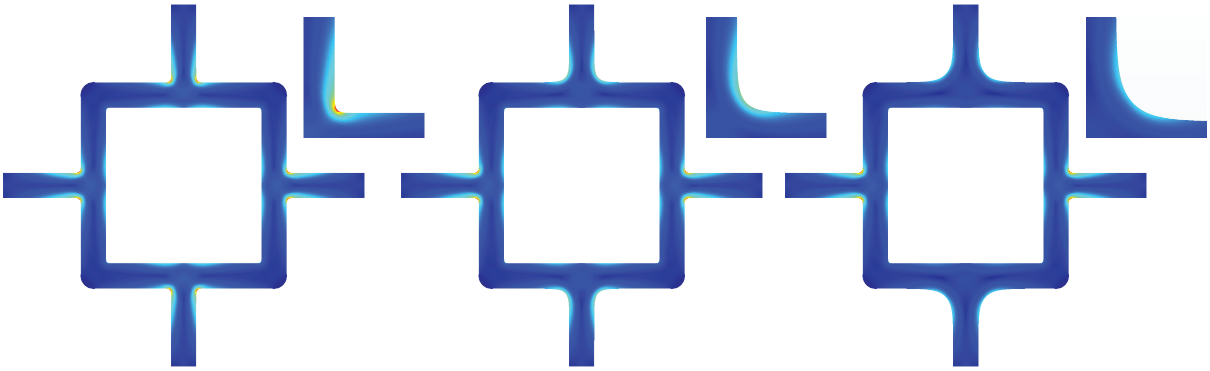


Figure 5.2: The stress a microstructure experiences depends strongly on the structure's corner smoothness.

in production [82].

5.3 Overview

Our aim is to produce printable microstructures achieving a prescribed homogenized elasticity tensor (effective material properties) $C^\star$ with minimal pointwise worst-case stress; the general optimization problem can be written as

$$\underset{\substack{\omega \text{ printable} \\ C^H(\omega)=C^\star}}{\operatorname{argmin}} \; J(s(\omega)), \tag{5.1}$$

where J is an integral stress measure, e.g., the L_p norm of a pointwise stress measure, ω is the microstructure shape, and $s(\omega)$ is the pointwise worst-case stress distribution. The pointwise worst-case stress distribution is a function of the shape only, obtained separately for each point by maximizing a stress norm over all possible unit macroscopic (averaged) loads applied to the structure.

To solve this optimization problem we need (i) a parametric shape description that can be meshed efficiently and differentiated with respect to its parameters; (ii) a way to impose printability and elastic tensor constraints; and (iii) an efficient and accurate method to compute shape derivatives of our stress objective.

Our shape representation is based on a skeleton graph, consisting of vertices connected by edges. We construct a smooth implicit surface from this skeleton by inflating each edge and joining them together smoothly with a custom blending operation (Section 5.6). The final inflated geometry is controlled by radius, position, and smoothing parameters assigned to each vertex. These shape parameters are the variables in the optimization.

To reduce our search space and improve performance, we focus on patterns with reflectional symmetry. These shapes are guaranteed to produce axis-aligned orthotropic material properties and can be designed by meshing and analyzing only the period cell's positive octant (upper-right quadrant in 2D) with appropriate modifications to the homogenization boundary conditions. In the following, we refer to this portion of the period cell as the structure's *symmetry cell*. We enforce reflectional symmetries by assigning independent parameters only to vertices in the positive octant and constraining every vertex initially on a reflection plane to stay on that plane.

We enforce two types of printability constraints: minimal part thickness and a self-supporting constraint. We do this by first *enforcing printability on the pre-blended structure*, that is, the structure where all inflated edges are combined with an exact union. This can be done easily by applying inequality constraints on the position and radius variables during our optimization. In particular, we apply a lower bound constraint to the radius variables and ensure the sphere defined by each vertex is supported:

$$v.z - v.r \geq \min_{u}(u.z - u.r),$$

where z is the vertex's position along the printing axis, r is the radius, and the minimization is over neighboring vertices of v that can "support" v. Then we restrict our blending to avoid bulging that would violate the self-supporting constraint by creating overhanging features (Section 5.6). The result is a printable, smoothly blended structure.

We optimize a discretization of (5.1) using SLSQP [55] within the NLopt package

[49]. At each step of the optimization, we perform the following steps to evaluate the objective function and its gradient:

(i) We use CGAL's [6] 3D Mesh Generation package to create a high-quality tetrahedral mesh of the structure's symmetry cell from our smooth signed distance function.

(ii) We solve the elasticity equations (2.11) within the symmetry cell to find the fluctuation displacements, then compute the effective macro-scale elasticity tensor and its shape derivative, as well as the worst-case stress measure at each point (by solving a small eigenvalue problem at each point). These steps are outlined in Section 5.4.

(iii) We compute the perturbation velocity of the boundary vertices induced by perturbing each shape parameter using automatic differentiation of our signed distance function.

(iv) We solve the adjoint problem for our stress objective and evaluate the shape derivative (Section 5.5): using the adjoint solution, we construct the discrete volume differential form (5.22) that computes the change in stress objective caused by perturbing each mesh vertex. Then from (5.22), we construct an accurate boundary differential form (discrete shape derivative) (5.13) which takes a boundary perturbation velocity, smoothly extends it to the interior and applies the volume differential form. Feeding in the velocities found in (iii), we obtain accurate partial derivatives for each parameter.

5.4 Worst-case stress analysis of periodic structures

In this section, we obtain formulas for the worst-case stress at a point in the microstructure. In general, finding a load distribution (with the total load fixed) that maximizes the stress norm at a point is a difficult nonconvex problem—unlike minimization, maximization of a convex function on a constrained domain is not convex.

However, in our specific setting, the space of loads that we need to consider is low (6-dimensional) and periodic homogenization imposes structure that greatly simplifies the optimization. We consider the idealized setting where the structure's length scale is much smaller than the object to be filled by the structure—a standard assumption in microstructure design—so that at the macro scale (the full object's scale) the structure can be viewed as a homogeneous material experiencing smooth stress and strain distributions varying gradually compared to the size of a single cell. Evaluating these smooth stress and strain distributions at a point in the full object obtains the *average* stress and strain tensor experienced by the microstructure cell at that point, and under the periodic homogenization assumptions either of these tensors completely determines the microstructure's deformation. Thus, the external loads transmitted to the cell are captured by a single averaged "macroscopic stress" tensor, reducing our space of loads to 6 dimensions.

Solving this optimization problem *for every point*, we obtain a distribution of per-point worst-case stress (and corresponding per-point worst-case load), which depends on the microstructure's shape only, not on the full object or choice of loading.

Periodic homogenization Our formulation depends fundamentally on periodic homogenization, which is introduced in Section 2.3. In particular, we use the homogenized elasticity tensor C^H to define the material properties achieved by a microstructure (needed to express the design goal). Computing these effective properties requires probing the microstructure with constant-strain stretches and measuring its elastic response (solving for fluctuation displacements, $\mathbf{w}^{kl}$). This is done by solving (2.11), after which we can compute C^H using (2.12). The compliance tensor, S^H, is defined as the symmetric rank four inverse of C^H.

Macro- to micro-stress We use periodic homogenization to determine the stress in a microstructure as follows. Once we have solved (2.11), we can construct the rank-four tensor mapping the cell's macroscopic strain to the microscopic strain at a point $\mathbf{x} \in \omega$: $G_{ijkl}(\mathbf{x}) \overset{\text{def}}{=} [\varepsilon(\mathbf{w}^{kl})(\mathbf{x}) + e^{kl}]_{ij}$. Notice that we already introduced this tensor in the two-scale asymptotic expansion derivation of periodic homogenization (see (2.18)). Then the microscopic stress at $\mathbf{x}$ under a particular macroscopic loading $\bar{\sigma}$ is

$$\sigma(\mathbf{x}) = C^{\text{base}} : G(\mathbf{x}) : S^H : \bar{\sigma} \overset{\text{def}}{=} F(\mathbf{x}) : \bar{\sigma}. \tag{5.2}$$

Tensor F is the linear map from the macroscopic stress to the microscopic stress at a point. Although the micro-stress can vary from point to point in complicated ways, it actually can be maximized explicitly due to its simple relationship to load $\bar{\sigma}$ at each point.

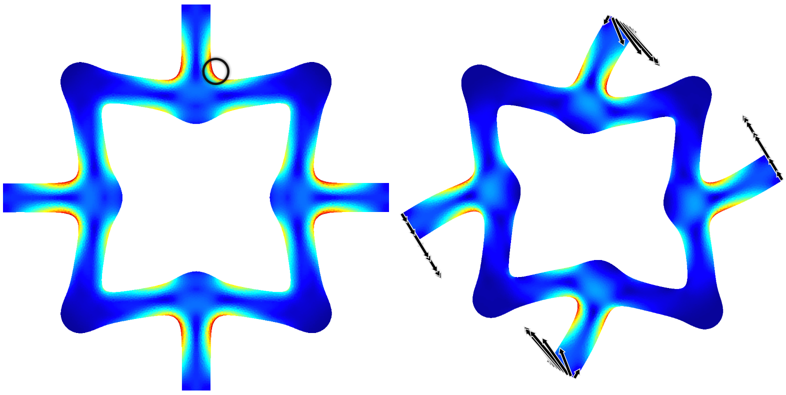


Figure 5.3: Left: worst-case stress field with peak stress value circled. Right: macroscopic stress condition inducing this peak stress.

Various pointwise stress measures can be utilized. The worst-case maximum principal stress, predicting the failure of brittle material, is defined at a point in the microstructure as:

$$s_m(\mathbf{x}) = \max_{\bar{\sigma}:\bar{\sigma}=1} \lambda_{\max}(F(\mathbf{x}) : \bar{\sigma}), \tag{5.3}$$

where we maximize over macroscopic stresses with unit Frobenius norm. The worst-case microscopic stress Frobenius norm is:

$$s_f^2(\mathbf{x}) = \max_{\bar{\sigma}:\bar{\sigma}=1} \bar{\sigma} : \underbrace{F(\mathbf{x})^T : F(\mathbf{x})}_{T^F(\mathbf{x})} : \bar{\sigma}, \tag{5.4}$$

and the maximum von Mises microscopic stress (for predicting ductile failure) is given by

$$s_v^2(\mathbf{x}) = \max_{\bar{\sigma}:\bar{\sigma}=1} \bar{\sigma} : \underbrace{F^T(\mathbf{x}) : V^T : V : F(\mathbf{x})}_{T^V(\mathbf{x})} : \bar{\sigma}, \tag{5.5}$$

where V extracts the von Mises stress: $(V\sigma)_{ij} = \sqrt{\frac{3}{2}}\left(\sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}\right)$.

It turns out that each worst-case stress measure can be evaluated efficiently by solving a tensor eigenvalue problem. The expressions for s_f and s_v are already in the familiar eigenvalue form for symmetric tensors T^F and T^V: they can be computed by flattening these tensors into matrices and finding the maximum eigenvalue. The corresponding eigenvector, when unflattened, is the worst-case load. Worst-case maximum principal stress leads to a different type of eigenvalue problem (see Appendix A.1 for more detail):

$$\mathbf{n}^\star \cdot T^M(\mathbf{x}) : [\mathbf{n}^\star \mathbf{n}^{\star T}] = s_m^2 \mathbf{n}^\star, \tag{5.6}$$

with $T^M(\mathbf{x}) = F(\mathbf{x}) : F(\mathbf{x})^T$. We denote the *per-point* macro-stress tensor attaining the peak stress measure by worst-case load $\bar{\sigma}^\star$.

Surprisingly, all three of these measures are very similar, as the maximum stress tensor eigenvalue tends to dominate all others in the worst-case scenario (see Figure 5.4).

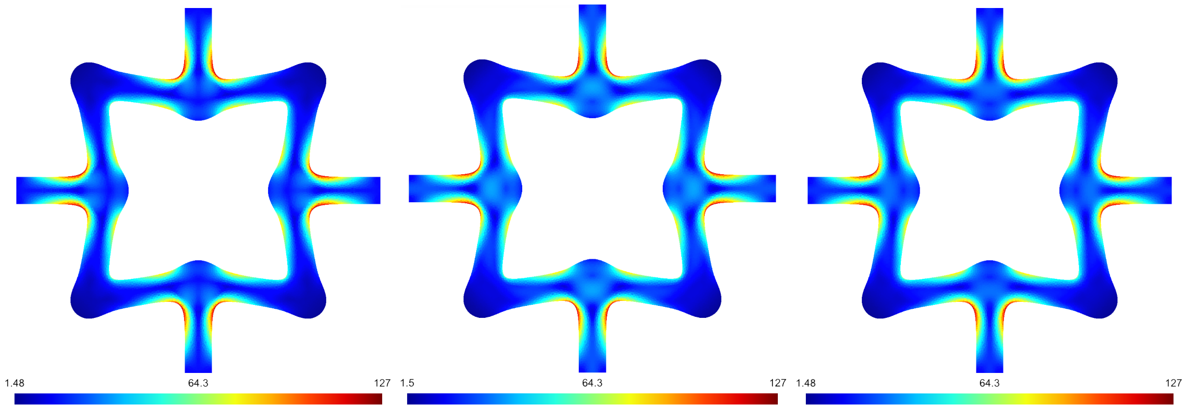


Figure 5.4: The three worst-case microscopic stress measures from left to right: maximum principal stress, Frobenius norm, and von Mises stress. Each is efficient to compute exactly for every element in the mesh.

Optimization problem We wish to find a microstructure that minimizes the L_p norm of the stress measure (either s_m, s_f, or s_v) while achieving a particular homogenized elasticity tensor $C^\star$:

$$\underset{\substack{\omega \text{ printable} \\ C^H(\omega)=C^\star}}{\operatorname{argmin}} J(\omega), \quad J(\omega) = \int_\omega j(s(\mathbf{x}))\, \mathrm{d}\mathbf{x}, \quad j(s) \overset{\text{def}}{=} s^{p/2}. \tag{5.7}$$

Here, s is the *squared* stress measure of interest, e.g. s_v^2. In our experiments, we use p up to 16. For simplicity, the remainder of this paper will consider s_f, but the other cases are nearly identical.

Discretization We discretize the cell problems (2.11) with quadratic tetrahedral FEM, which we found essential for accurate stress and homogenized tensor evaluation. We use straight-edged elements (subparametric FEM) for representing the geometry to simplify meshing and the shape derivative formulas (edge nodes are placed at the edge midpoints).

The integral in the objective function (5.7) is computed with numerical quadrature, and the quadrature points determine where worst-case stress must be evaluated. We use piecewise constant quadrature, meaning *a single worst-case stress quantity is computed for each mesh element* (from the element's averaged fluctuation strains).

5.5 Shape optimization

We found the equality-constrained optimization formulation (5.7) necessary to reliably reduce stress while preserving macroscopic behavior. For instance, naïvely smoothing to reduce stress dramatically changes macroscopic properties and is undone by re-fitting via [75]. Further, though stress-optimal designs generally exhibit smooth features, predicting how smooth is difficult. In fact, [60] demonstrates that theoretically optimal structures can have sharp corners. While we have not encountered this in our experiments, we see significant variation in optimal curvature.

To solve (5.7) with SLSQP, we need partial derivatives of the objective and constraints with respect to each shape parameter. We compute these derivatives in two stages: first, we compute the quantities' *shape derivatives*, i.e., how they change when the domain ω is perturbed by an arbitrary velocity field $\mathbf{v}$. Then, to differentiate with respect to a parameter, we feed into these shape derivatives the velocity field induced by changing the parameter.

5.5.1 Volumetric vs. boundary shape derivatives

The shape derivative of the objective functional J in the direction of a shape perturbation $\mathbf{v}$ is defined as the Gâteaux derivative

$$\mathrm{d}J[\mathbf{v}] \overset{\text{def}}{=} \lim_{t \to 0} \frac{J(\omega(t, \mathbf{v})) - J(\omega)}{t}, \tag{5.8}$$

where $\omega(t, \mathbf{v}) = \{\mathbf{x} + t\mathbf{v}(\mathbf{x}) : \mathbf{x} \in \omega\}$.

Shape derivatives have been used to minimize a wide variety of objective functions, and they are typically expressed as a boundary integral depending only on the boundary's normal velocity, e.g. [4]. However, we found that in our setting (nearly nonsmooth objective functionals resulting from using the L_p norm for a high p), these standard formulas give poor accuracy (Figure 5.5). To address this issue, we have developed a different formulation using volume integrals instead. In this section, we use a simplified model problem to clarify the source of inaccuracy in the standard method and to introduce the main idea of our approach. Our expression for the shape derivative for worst-case stress is described in Section 5.5.2 and derived in Section A.5.

Our model problem is to differentiate $J = \int_\omega j(\nabla u)\,\mathrm{d}\mathbf{x}$ with respect to changes of the domain ω, where u is a scalar field solving

$$\nabla \cdot (\nabla u + g) = 0 \quad \text{in } \omega, \quad \hat{\mathbf{n}} \cdot (\nabla u + g) = 0 \quad \text{on } \partial\omega. \tag{5.9}$$

Here, g is analogous to the macroscopic strain.

The standard expression for J's shape derivative is:

$$\mathrm{d}J[\mathbf{v}] = \int_{\partial\omega} \Big(j(\nabla u) - \nabla p \cdot (\nabla u + g) \Big) \mathbf{v} \cdot \hat{\mathbf{n}} \, \mathrm{d}\mathbf{x},$$

where p is the adjoint scalar field solving

$$\nabla \cdot \Big(\nabla p - j'(\nabla u) \Big) = 0 \quad \text{in } \omega, \quad \hat{\mathbf{n}} \cdot \Big(\nabla p - j'(\nabla u) \Big) = 0 \quad \text{on } \partial\omega.$$

For optimizing relatively smooth functionals like compliance, this method yields good results with standard discretization approaches. To illustrate the problem with it in our setting, we briefly review the main steps of its derivation.

We consider the initial domain ω and slightly perturbed domain ω_t, with the deformation from ω to ω_t given by map $\mathbf{x}_t = \mathbf{x} + t\mathbf{v}$ with Jacobian $I + t\nabla\mathbf{v}$. Using the Reynolds Transport Theorem, the derivative of J can be computed in terms of $\dot{u}$, the *Eulerian derivative* of u_t (the PDE solution at time t) evaluated at $t = 0$:

$$\mathrm{d}J[\mathbf{v}] = \int_{\omega} j'(\nabla u) \cdot \nabla \dot{u} \; \mathrm{d}\mathbf{x} + \int_{\partial\omega} j(\nabla u) \mathbf{v} \cdot \hat{\mathbf{n}} \; \mathrm{d}A(\mathbf{x}),$$

We obtain $\dot{u}$ starting from the weak form of (5.9) in domain ω_t:

$$\int_{\omega_t} \nabla \phi \cdot (\nabla u_t + g) \, \mathrm{d}\mathbf{x} = 0, \quad \forall \phi. \tag{5.10}$$

We apply the Reynolds Transport Theorem for the second time to obtain an equation

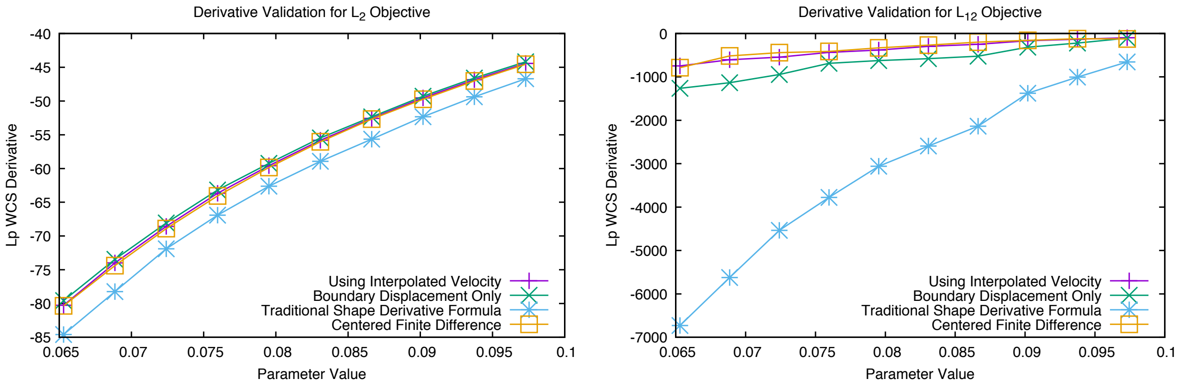


Figure 5.5: Comparisons of different techniques for differentiating our L_p worst-case stress objective with respect to a pattern parameter (in this case, an offset variable whose value is plotted on the horizontal axis). For low p, derivatives are computed reasonably accurately by all methods. However, for L_{12} (which is needed to approximate L_∞ as shown in Figure 5.14) the traditional formula for the shape derivative is wildly inaccurate. Using our discrete differential form with interpolated boundary velocities maintains good accuracy.

for the Eulerian derivative, $\dot{u}$. As detailed in Section A.3, an intermediate step in the derivation of the Reynolds Transport Theorem—when applied to the weak form (5.10)—yields an equation for the *material derivative*, δu:

$$\int_\omega \nabla\phi \cdot \big(\nabla \delta u - \nabla \mathbf{v}(\nabla u + g) - (\nabla \mathbf{v})^T \nabla u + (\nabla u + g)\nabla \cdot \mathbf{v}\big)\, \mathrm{d}\mathbf{x} = 0 \tag{5.11}$$

holding for all ϕ. The derivation then proceeds by substituting the definition $\dot{u} \overset{\text{def}}{=} \delta u - \mathbf{v} \cdot \nabla u$. Applying integration by parts to the $\nabla \cdot \mathbf{v}$ term, we obtain:

$$\begin{aligned}\int_\omega \nabla\phi \cdot \nabla \dot{u}\, \mathrm{d}\mathbf{x} = &\underbrace{\int_\omega \nabla(\nabla\phi \cdot \mathbf{v}) \cdot (\nabla u + g)\, \mathrm{d}\mathbf{x}}_{I_1} \\ &- \int_{\partial\omega} \nabla\phi \cdot (\nabla u + g)(\mathbf{v} \cdot \hat{\mathbf{n}})\, \mathrm{d}\mathbf{x} \quad \forall\phi.\end{aligned} \tag{5.12}$$

Note that I_1 is exactly of the form (5.10), with $\nabla\phi \cdot \mathbf{v}$ instead of ϕ. As ϕ is assumed to be arbitrary (from a sufficiently smooth space of functions), (5.10) implies that this term vanishes, leaving us with a simple equation for $\dot{u}$ (in this case, a Laplace equation with a Neumann boundary condition).

Our key observation is that this term may not vanish identically in the discrete setting, with the error being particularly large for low-order finite element discretizations. We found that for rapidly varying-functionals like our worst-case stress, this leads to large errors in derivatives. To solve this problem, one could choose the finite element basis and test functions for the problem in a way that guarantees that I_1 is exactly zero. Alternatively—and this is the approach we follow—one can skip integration by parts and use (5.11) to obtain the discretization. This approach ends up obtaining our objective's *exact discrete derivative*: it computes the exact change in the objective if the tetrahedral mesh nodes are infinitesimally advected by the velocity field without remeshing.

Extending boundary perturbation into the domain Unlike the traditional shape derivative formulas, this volume form will depend on the perturbations $\mathbf{v}$ on the entire domain ω, rather than only the boundary. This should not be surprising—finite element analysis is sensitive to the interior meshing—but it is problematic: perturbing the microstructure parameters induces a velocity $\mathbf{v}^b$ on ω's boundary *only*, leaving the interior perturbation undefined.

Although we know that, in the limit of refinement, the shape derivative is independent of $\mathbf{v}$ inside the domain (interior perturbations do not impact shape), setting the interior velocity to 0 hurts accuracy in practice (Figure 5.5). This is because perturbing only boundary vertices distorts the incident elements and increases the discretization error as a result. Smoothly extending velocities into the interior (so that elements advect with the boundary, distorting less) improves accuracy.

We perform this extension by solving the Laplace equation with Dirichlet condition $\mathbf{v}^b$ on the microstructure boundary and periodic conditions on the cell boundary. Denoting the velocity at the interior vertices by $\mathbf{v}^i$, we compute $\mathbf{v}^i = -L_{ii}^{-1} L_{ib} \mathbf{v}^b$, where L denotes the linear finite element Laplacian matrix. Using this extension, we can define a

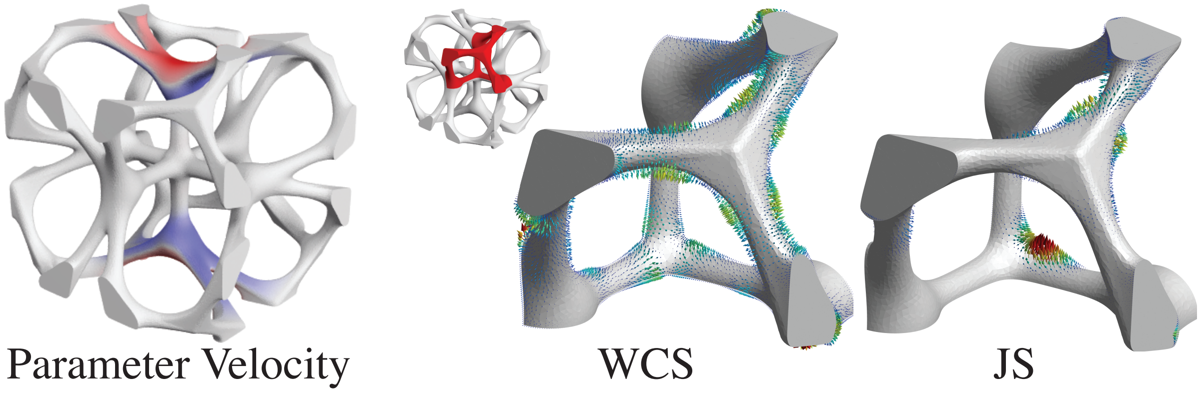


Figure 5.6: Visualization of shape derivative quantities. Left: the normal velocity induced by changing a position variable, as computed by automatic differentiation of the signed distance function. On the right, the steepest descent velocity for our worst-case stress objective and the compliance tensor-fitting objective from [75].

differential form acting on the boundary velocity $\mathbf{v}^b$:

$$\mathrm{d}J^b[\mathbf{v}^b] \overset{\text{def}}{=} \mathrm{d}J[\mathbf{v}] = \mathrm{d}J\left[\begin{pmatrix} -L_{ii}^{-1}L_{ib} \\ I \end{pmatrix}\mathbf{v}^b\right]. \tag{5.13}$$

This differential form can be constructed explicitly by applying the transpose of the interpolation matrix to the explicit representation of the volume differential ((5.22) below). This accelerates gradient computation for patterns with many parameters and recovers an accurate steepest descent boundary velocity, as visualized in Figure 5.6.

5.5.2 Shape derivative in volume form for worst-case stress

We summarize the formulas for stress shape derivatives, in the form we use in our implementation. The derivations of these are somewhat lengthy and appear in Section A.5.

The worst-case stress objective's shape derivative is, in volume form:

$$\mathrm{d}J[\mathbf{v}] = \int_\omega j\nabla\cdot\mathbf{v} + \tau^{kl} : D[\varepsilon(\mathbf{w}^{kl})] + \gamma :: \mathrm{d}C^H[\mathbf{v}]\,\mathrm{d}\mathbf{x}, \tag{5.14}$$

where $D[\bullet]$ denotes the material derivative, and $\mathbf{v}$ is the perturbation velocity field. Tensors τ^{kl} and γ are partial derivatives of $j(s(\mathbf{x})) = j(s(\varepsilon^{kl}, C^H, \mathbf{x}))$ with respect to the fluctuation strains and homogenized elasticity tensor:

$$\begin{aligned}
\delta j &= (j')\frac{\partial s}{\partial \varepsilon^{kl}} : \delta\varepsilon^{kl} + (j')\frac{\partial s}{\partial C^H} :: \delta C^H \overset{\text{def}}{=} \tau^{kl} : \delta\varepsilon^{kl} + \gamma :: \delta C^H, \\
\tau^{kl} &= (2j' C^{\text{base}} : F : \bar{\sigma}^\star)[S^H : \bar{\sigma}^\star]_{kl}, && (5.15) \\
\gamma &= (-2j' F^T : F : \bar{\sigma}^\star) \otimes (S^H : \bar{\sigma}^\star). && (5.16)
\end{aligned}$$

(Recall that the $\bar{\sigma}^\star$ value for a point $\mathbf{x}$ is the unit macroscopic stress maximizing microscopic stress at $\mathbf{x}$). These formulas do not include the derivative of the worst-case load $\bar{\sigma}^\star$ since the derivative of an eigenvalue (pointwise worst-case stress) does not depend on its eigenvector's derivative; see Appendix A.2. In other words, the worst-case load can be considered constant when differentiating.

Because we use straight-edged elements (i.e. the fluctuation displacement fields are piecewise quadratic, but the geometry representation is piecewise linear), the perturbation velocity $\mathbf{v}$ is piecewise linear. Thus $\mathbf{v}$ is represented as a perturbation vector $\delta\mathbf{q}_i$ on each mesh vertex:

$$\mathbf{v} = \sum_i \lambda_i \delta\mathbf{q}_i, \tag{5.17}$$

where λ_i is vertex i's linear shape function.

The first term in the integrand of (5.14) can be computed directly, and the third term is the homogenized elasticity tensor's shape derivative, which can be computed from the fluctuation displacements $\mathbf{w}^{kl}$. The second term includes the unknown material derivative of the fluctuation strains; we re-express this term using the solution to the adjoint cell problems, which are in the same form as (2.11) but with different right hand sides.

Adjoint equation We need to solve an adjoint equation to express the term involving the fluctuation displacements' material derivatives in a computationally tractable form.

The weak form of the kl^{th} adjoint cell problem PDE is:

$$\int_\omega \tau^{kl} : \varepsilon(\phi)\ \mathrm{d}\mathbf{x} = \int_\omega \varepsilon(\mathbf{p}^{kl}) : C^{\text{base}} : \varepsilon(\phi)\ \mathrm{d}\mathbf{x}, \tag{5.18}$$

for all test functions ϕ, where $\mathbf{p}^{kl}$ is the adjoint state vector field. $\mathbf{p}^{kl}$ is discretized using the same piecewise quadratic basis functions as used for fluctuation displacements.

The expression for the discrete shape derivative is:

$$\begin{aligned} \mathrm{d}J[\mathbf{v}] = \int_\omega \Big[\big(j - \varepsilon(\mathbf{p}^{kl}) : \sigma^{kl}\big)\, \nabla \cdot \mathbf{v} + (\nabla \mathbf{p}^{kl} \nabla \mathbf{v}) : \sigma^{kl} \\ + \big(\varepsilon(\mathbf{p}^{kl}) : C^{\text{base}} - \tau^{kl}\big) : (\nabla \mathbf{w}^{kl} \nabla \mathbf{v}) \Big]\ \mathrm{d}\mathbf{x} \\ + \left(\int_\omega \gamma\ \mathrm{d}\mathbf{x} \right) :: \mathrm{d}C^H[\mathbf{v}], \end{aligned} \tag{5.19}$$

where $\sigma^{kl} \overset{\text{def}}{=} C^{\text{base}} : \big[\varepsilon(\mathbf{w}^{kl}) + e^{kl}\big]$ is the microscopic stress corresponding to $\mathbf{w}^{kl}$.

For computation, it is convenient to express $\mathrm{d}J[\mathbf{v}]$ explicitly as a 1-form acting on the per-vertex perturbation vector field $\delta\mathbf{q}$. To do this, we re-express $\mathbf{v}$ in terms of $\delta\mathbf{q}$. Using (5.17),

$$\nabla \cdot \mathbf{v} = \sum_m \delta\mathbf{q}_m \cdot \nabla\lambda_m, \qquad \nabla\mathbf{v} = \sum_m \delta\mathbf{q}_m \otimes \nabla\lambda_m. \tag{5.20}$$

We can write $\nabla\mathbf{p}^{kl}$ in terms of each scalar-valued finite element shape function φ_n and its vector-valued coefficient $\mathbf{p}_n^{kl}$ as:

$$\nabla\mathbf{p}^{kl} = \sum_n \mathbf{p}_n^{kl} \otimes \nabla\varphi_n. \tag{5.21}$$

Substituting these into $\mathrm{d}J[\mathbf{v}]$ and simplifying:

$$\begin{aligned}\mathrm{d}J[\lambda_m \delta\mathbf{q}_m] = \delta\mathbf{q}_m \cdot &\int_\omega a \nabla\lambda_m + \left(\nabla\lambda_m \cdot \mathbf{b}_n\right) \nabla\varphi_n \;\mathrm{d}\mathbf{x} \\ &+ \left(\int_\omega \gamma \;\mathrm{d}\mathbf{x}\right) :: \mathrm{d}C^H[\lambda_m \delta\mathbf{q}_m], \\ a = j - \varepsilon(\mathbf{p}^{kl}) &: \sigma^{kl}, \\ \mathbf{b}_n = \sigma^{kl}\mathbf{p}_n^{kl} &+ \left(\varepsilon(\mathbf{p}^{kl}) : C^{\text{base}} - \tau^{kl}\right) \mathbf{w}_n^{kl},\end{aligned} \tag{5.22}$$

where summation over repeated subscripts is implied.

The exact discrete shape derivative of homogenized tensor C^H is $\mathrm{d}C^H_{ijkl}[\lambda_m \delta\mathbf{q}_m] =$

$$\begin{aligned}\frac{\delta\mathbf{q}_m}{|Y|} \cdot \int_\omega &\left(\sigma^{ij} : C^{\text{base}^{-1}} : \sigma^{kl}\right) \nabla\lambda_m \\ &- \left[\nabla\lambda_m \cdot \left(\sigma^{kl}\mathbf{w}_n^{ij} + \sigma^{ij}\mathbf{w}_n^{kl}\right)\right] \nabla\varphi_n \;\mathrm{d}\mathbf{x},\end{aligned}$$

where again summation over m and n is implied.

To summarize, we compute the shape derivative in three stages: (i) solve the adjoint problem to obtain solutions $\mathbf{p}^{kl}$; (ii) compute the shape derivative of the homogenized elasticity tensor C^H; and (iii) substitute these quantities into (5.22) and evaluate the integrals.

5.6 Convex hull-restricted blending

The parametric microstructure model is an essential part of our method. It is designed to have few parameters (ensuring optimization is stable and fast), while at the same time providing enough degrees of freedom to achieve our stress reduction goal. Our model consists of a skeleton graph with position, radius, and smoothing parameters on each vertex. These parameters describe a sphere for each vertex, and we define our edge geometry *primitives* as the convex hull of the endpoint spheres for each edge. We then combine every edge primitive incident on a vertex to form a smooth joint, with smoothness determined by the vertex's smoothing parameter, and finally blend all the joints together into a single signed distance function.

A variety of techniques, discussed briefly in Section 5.2, have been developed to control smoothness both for parametric and implicit surfaces. Unfortunately, existing techniques do not meet some criteria our method needs to satisfy. Our list of requirements includes: (i) *Smooth geometry:* the surface should be smooth for almost all parameter values. (ii) *Smooth dependence on parameters*: the dependence of surface points on shape parameters should be differentiable, as (efficiently computable) derivatives are needed for shape optimization. (iii) *Locality:* smoothing needs to be controlled locally, with each smoothness parameter affecting a part of the geometry (in our case, a joint). (iv) *No bulging:* the surface does not create unnecessary protrusions, which are a common side effect of smoothed boolean operations; this aspect is particularly important for printability. (v) *Constraint-compatibility:* printability constraints can be efficiently expressed in terms of shape parameters.

As all existing methods we considered appeared difficult to adapt directly to our application, we have developed a new, simple blending technique, which works quite well for our application. We believe that the basic idea may be useful in other contexts, but we have not evaluated its utility for general-purpose implicit modeling.

Our method works on a collection of smooth shapes; for each we need a signed distance function, and for a group of shapes we need to compute signed distances to the convex hull efficiently. We define a *joint* as a maximal set of primitives that have a nonempty intersection. In our setting, this is a set of edge geometry primitives incident on a vertex.

Bulging Intuitively, bulging is a behavior of shape blending methods where material is added to a solid in places unnecessary to create a smoothed shape (e.g., Figure 5.7). It is of special importance in the context of 3D printing, as bulging is likely to violate the printability of certain types of structures. To clarify: we enforce printability constraints on the edge primitives during optimization, so the boolean (nonsmooth) union of edges is guaranteed to be printable. Our blending operator is always additive and thus cannot violate the edge primitives' minimum thickness constraint, but bulging can easily add overhanging material that violates the self-supporting constraint for SLA printers (Figure 5.8).

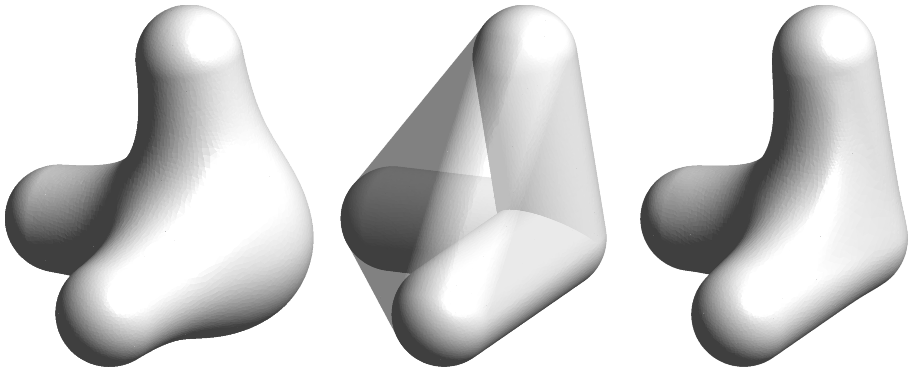


Figure 5.7: Bulging. Left: bulging exhibited by smoothed distance function; Center: blending region defined by convex hull; Right: smooth, bulge-free joint.

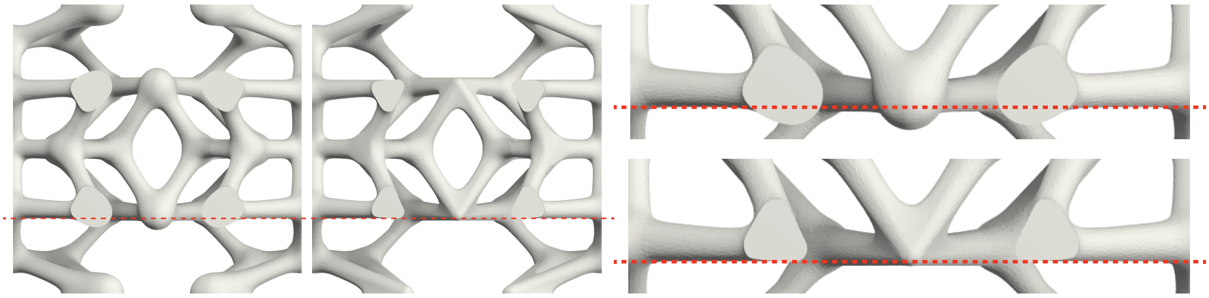


Figure 5.8: Unprintability due to bulging (overhanging features bulging below the red line) is resolved by the convex hull-restricted blending.

While no formal definition of bulging is likely to capture the intuition completely, we use the following definition, based on the intuition for blended sets of convex primitives, which matches our goals well: *a bulge is the difference between the exact union of the constituent primitives and the blended union that is not contained in the convex hull of* *the primitives forming a joint.*

The rationale for this definition is that for smooth primitives, the parts on the surface of the convex hull are smooth, so there is no need for further smoothing.

Summary Our shape's boundary is defined as the zero level-set of a scalar volume function, which is constructed from the distance functions to the convex, tangent-plane continuous primitives. As a basic component of our method, we use the *Kreisselmeier-Steinhauser (KS) function*,

$$KS(y_1, \ldots, y_n; \rho) = -\frac{1}{\rho} \ln \left(\sum_{i=1}^{n} e^{-\rho y_i} \right), \tag{5.23}$$

where variables y_i are the distances to be blended. This function is a smoothed version of the minimum function, with smoothness controlled by parameter ρ. For the purposes of blending, it is more intuitive—and yields better-scaled optimization variables—to apply the change of parameters $s = 1/\rho$. Then, increasing smoothing parameter s increases the amount of blending.

The KS function has two important properties: (i) if the difference between maximal y_i and the rest exceeds approximately $7s$ it vanishes to 10^{-12} accuracy, so it is effectively local; and (ii) it is always smaller than the non-smoothed min function, with the maximal difference proportional to s.

We use a two-stage blending process: first, we perform local blending at joints, using position-dependent smoothing (s drops to zero for points outside the joint's convex hull). This yields larger smooth, non-convex building blocks for the structure: the blended joints, which are connected by shared primitives (the single common edge primitive for a pair of adjacent joints in our setting). Second, we blend all joints together with a spatially-varying smoothing amount that avoids sharp creases while preventing unnecessary bulging on the shared primitives.

Blending algorithm The algorithm computes the level-set function $F(\mathbf{p})$ at a point $\mathbf{p} \in \mathbb{R}^3$ given a collection of convex primitives P_i $(i = 1 \dots N_p)$ with signed distance functions $F_i(\mathbf{p})$. The primitives form joints J_m $(m = 1 \dots N_J)$, each defined by a subset of the primitives. We define R_m to be the minimal distance from the medial axis of any primitive in J_m to the convex hull boundary.

Blending is controlled by smoothing parameters s_m, defined *per joint* (skeleton vertex), and the smoothing amounts s_{ab}, defined per overlapping joint pair. For reasons we explain below, we use only s_m as optimization variables (in addition to the primitives' parameters).

Constructing blended joints. We use the smoothed minimum function KS to construct joint blends, but with a spatially varying smoothing parameter $s(\mathbf{p})$. For points inside the convex hull of J_m, we use the maximal smoothing amount, equal to joint's blending parameter s_m. Outside the convex hull, s quickly decays to zero: we need to leave a small buffer zone outside the convex hull to keep the blended function differentiable and to avoid nearly non-smooth surfaces when an intersection curve of two primitives approaches the boundary. To apply this blending modulation, we define $\eta_J(t)$, where t is the (nondimensionalized) distance from $\mathbf{p}$ to the convex hull boundary, with the following requirements: (i) $\eta_J(t) = 1$ for $t \leq -1$, (well inside the convex hull); and (ii) $\eta_J(t) = 0$ at $t \geq \epsilon$, a user specified distance outside the convex hull.

We define two signed distance functions per joint, F_m^s, representing the blended joint, and F_m^h, representing the (nonsmooth) exact union of the primitives:

$$\begin{aligned} F_m^s(\mathbf{p}) &= KS\left(\{F_i(\mathbf{p}), i \in J_m\}; s_m \eta_J\left(\frac{\mathrm{dist}_{hull(J_m)}(\mathbf{p})}{R_m}\right)\right) \\ F_m^h(\mathbf{p}) &= \min_{i \in J_m} F_i(\mathbf{p}). \end{aligned} \tag{5.24}$$

Combining joints. The exact union of the smoothed joints in general is non-smooth: creases can appear on the shared primitives at the intersection of two joint surfaces.

The obvious solution is to combine all joints using the smoothed minimum function KS; however, applying it in all cases will lead to bulging (Figure 5.9). We apply it for a point $\mathbf{p}$ only to the two joints closest to $\mathbf{p}$ in the sense of smoothed distances F^s_m, and only when we detect that the joints "conflict" in a way that would cause a crease. Specifically, we observe that $|F^s_m(\mathbf{p}) - F^h_m(\mathbf{p})|$, $m = a, b$, measures the degree to which each joint's surface differs from the original primitive near $\mathbf{p}$. A crease forms along the shared primitive only if both of these quantities are nonzero; if one of the joints coincides with the shared primitive near $\mathbf{p}$, an exact union of the joint surfaces simply adds the smooth blend of the other joint (if any) atop the primitive. Thus we apply no blending when either quantity is zero and rapidly but smoothly transition to a slight blend as both quantities become nonzero. We achieve this with a smoothing modulation function, $\eta_F(t)$, of these quantities' squared geometric mean. We design this function to transition from $\eta_F(0) = 0$ quickly to 1 as t increases.

$$\begin{aligned} F(\mathbf{p}) &= KS(F^s_a(\mathbf{p}), F^s_b(\mathbf{p}); \tilde{s}_{ab}(\mathbf{p})) \\ \tilde{s}_{ab}(\mathbf{p}) &= s_{ab}\eta_F\left((F^h_a(\mathbf{p}) - F^s_a(\mathbf{p}))(F^h_b(\mathbf{p}) - F^s_b(\mathbf{p}))\right) \end{aligned} \tag{5.25}$$

Transition functions. Our specific choice of transition functions η_J and η_F is heuristic but yields good results based based on our experience. Other functions could be used as long as the specifications stated above are satisfied. Both functions are constructed using tanh, which can be viewed as constant outside a local interval similarly to the KS function. We prefer this choice to, e.g., splines, which can be made truly constant outside an interval, primarily due to the more compact formulas. The specific choices we found to work well are $\eta_J(t) = 1 - \tanh\left(C \max(1 + t, 0)\right)^r$, with $r = 10$ and $C = 1.025$, which transitions from 1 to 0 roughly in the range -0.5 to 0.1, and $\eta_F(t) = \tanh(1000t)$, which ensures smoothing is applied if even a slight crease appears. We did not attempt to optimize s_{ab}, which we fix to 0.02 for all joint pairs, because transition regions between joints do not typically concentrate stresses in our structures.

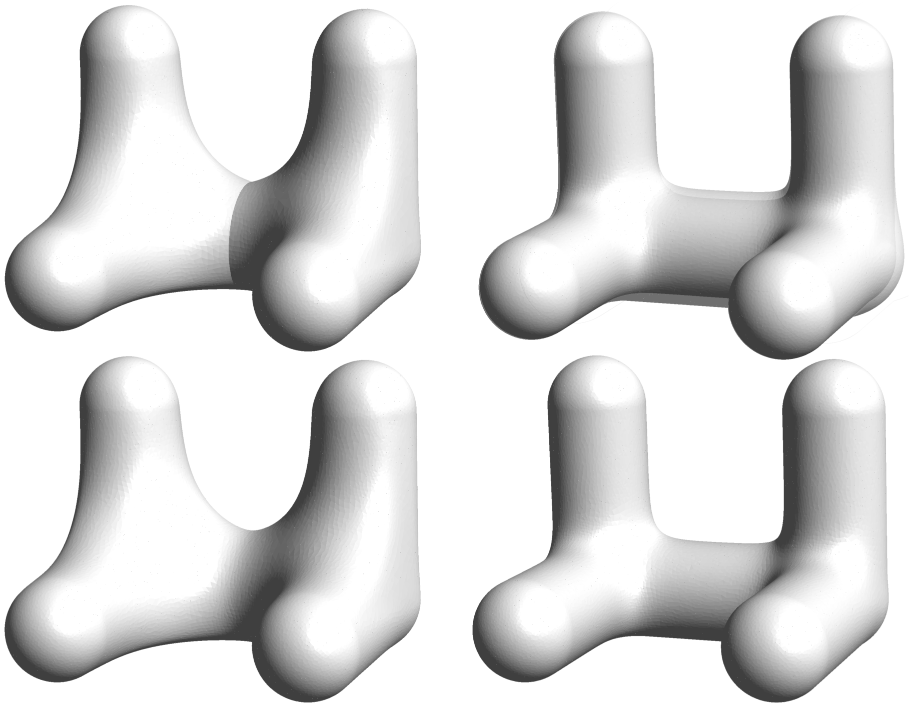


Figure 5.9: Top left: when two joints' blending regions overlap, we must smoothly union them to avoid a sharp crease. Top right: blending using KS with spatially constant parameter s_{ab} dilates the shared edge even when no overlap exists, introducing bulging that violates printability. Blending with smoothing modulation η_F (bottom) solves both problems.

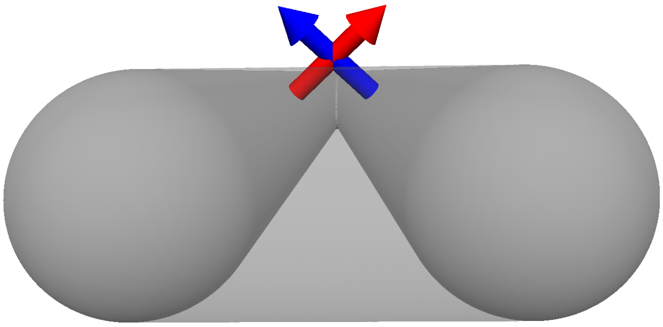


Figure 5.10: Two surfaces' normals intersect forming a right angle at a point outside the convex hull (translucent gray). Blending at this point would lead to potential unprintability.

Gradient-based implicit blending Gradient-based implicit blending ([37]) is a popular method to avoid bulging in implicit modeling. However, it is not guaranteed to restrict blending to the convex hull of the input primitives and therefore cannot preserve the primitives' printability in our setting. One example is demonstrated in Figure 5.10:

evaluating at the arrows' intersection point outside the convex hull, the two primitives' gradients are orthogonal. Since [37] uses only the gradients' inner product to modulate blending, the method is unable to distinguish this situation—where blending would violate printability—from an evaluation point near a 90° intersection of two cylinders for which the maximum amount of blending would be desired.

From parameters to structure We use CGAL's 3D Mesh Generation package, which robustly constructs high-quality adaptive tetrahedral meshes from smooth signed distance functions. For accurate stress evaluation, we found adaptive refinement essential. We achieved this using CGAL's `facet_distance` parameter, which ensures the mesh closely approximates the isosurface, leading to automatic refinement in regions of high curvature where stress concentrations are likely. One could instead drive refinement with a sizing field constructed from the worst-case stress field evaluated on a coarse initial mesh.

Creases in the isosurface lead to low-quality, occasionally non-manifold meshes with many tiny elements created to approximate the sharp features. However, CGAL can still produce high-quality meshes if it is provided all such creases explicitly as polygonal curves. Our tiled structures are smooth everywhere, but to mesh the symmetry cell, we must perform a boolean intersection with the cell's bounding cube. This intersection operation creates sharp curves along the cube faces, which we extract with marching squares and pass to CGAL as feature lines.

The vertices CGAL creates on the symmetry cell boundary do not lie perfectly on the cell's faces and must be snapped. However, points originating from the feature curves we extract *are* placed perfectly on the boundary. We use this fact to snap vertices in a robust way that is not too sensitive to the snapping threshold parameter: we use the feature curves to segment the mesh's boundary vertices into connected components and then decide whether each whole component lies on the boundary.

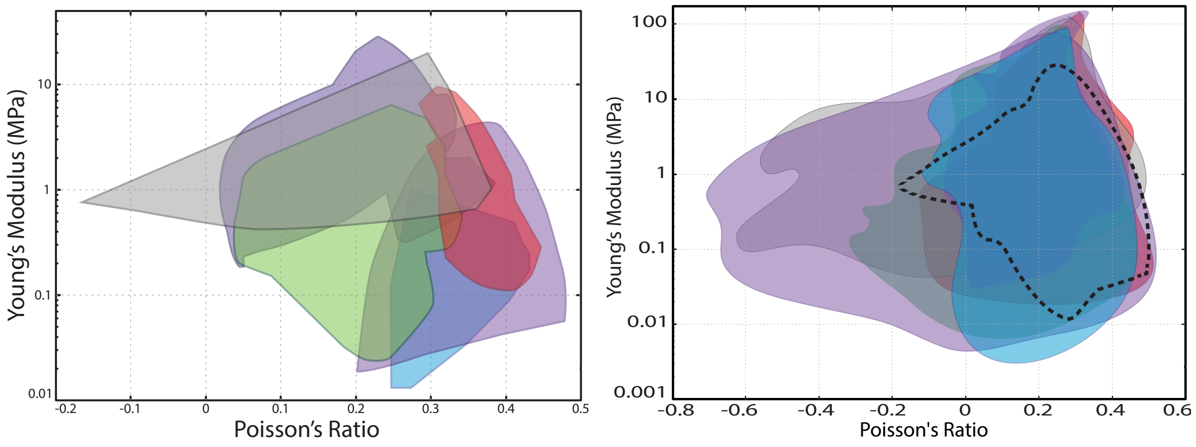


Figure 5.11: Comparison of material properties covered by [75] (left) and our optimization (right). Each colored region shows a single topology's coverage, and the dashed curve at the bottom outlines the region from [75].

Shape velocities When a parameter p (controlling vertex position, radius, or smoothing) changes, each point $\mathbf{x}$ on the microstructure's boundary evolves with some velocity. We need this induced velocity field to evaluate our objective and constraints' partial derivatives. We compute these fields directly from the signed distance function using automatic differentiation (with Eigen's `Auto Diff` module).

Boundary point $\mathbf{x}$'s motion in the normal direction is determined by differentiating the level set equation:

$$\hat{\mathbf{n}} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}p} = -\frac{1}{\|\nabla\phi\|}\frac{\partial\phi}{\partial p}, \tag{5.26}$$

where ϕ is the signed distance function and $\hat{\mathbf{n}} = \frac{\nabla\phi}{\|\nabla\phi\|}$ is the surface normal. The tangential velocity is left undefined, but it is natural to define it as zero.

5.7 Results

In this section, we summarize the results obtained from our method.

Isotropic elasticity dictionary We apply our framework to design structures with each of the six topologies chosen by [75], rather than performing a new topology search.

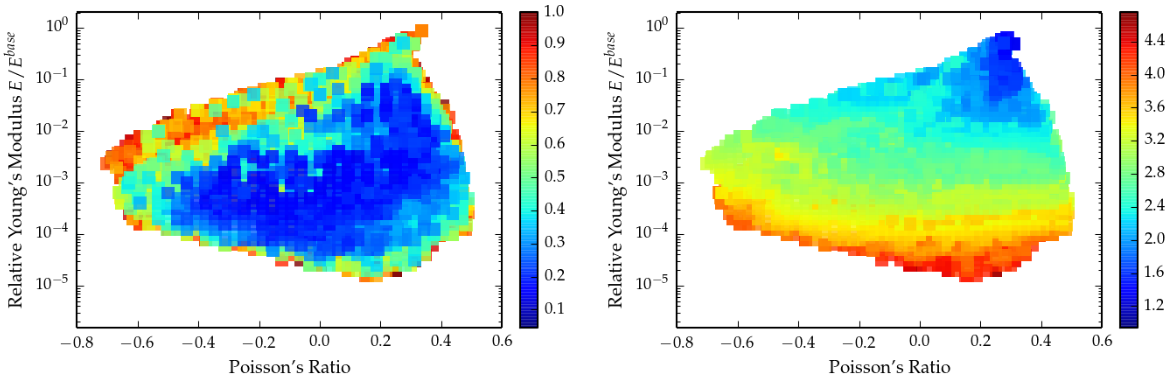


Figure 5.12: Stress reduction (left) and $\log_{10}$ of optimized max stress (right) for each microstructure in our database.

First, for each topology, we explore our new parametric model's coverage of isotropic material space using a breadth-first-style search: we draw an infinite grid in the Young's modulus, Poisson's ratio plane (with grid points spaced logarithmically in Young's modulus) and plot the point corresponding to the default parameter settings. We then design structures for every unattained grid point neighboring known structures by running our optimizer without the stress objective. We repeat until no more progress is made.

Next, starting from each structure found above, we run a worst-case stress minimization while holding fixed the homogenized material properties to design a more robust structure.

As can be seen from Figure 5.11, although we consider the more difficult problem of minimizing stress while simultaneously fitting the elasticity tensor, we achieve far greater coverage of isotropic elastic moduli. This is primarily due to two factors: a different shape parametrization enriched with additional parameters (smoothing factors at the joints) and a greater robustness of the meshing algorithm. Mesher robustness—particularly to topology changes as thicknesses grow and parts merge—helped extend the Young's modulus range considerably (by a factor of 10), and the additional degrees of freedom controlling joint smoothness expanded the Poisson's ratio range much closer to the theoretical limits (−1 to 1/2).

The total range covered is still significantly less than theoretically possible, as given by Hashin-Shtrikman bounds [42], however it is currently unknown if these bounds are realizable by printable structures. We conjecture that a more complete coverage requires finer-scale topology than provided by the type of structures we consider. We note that the narrowing range of achieved Poisson's ratios for higher E is predictable, since a fully solid structure can only produce the printing material's ν. One surprising outcome is that a single topology (number "0646" in [75]'s enumeration) covers essentially the entire space reached by all six topologies (with the caveat that patterns can change topology during the optimization by merging, especially as the thickness parameters increase).

Stress reduction Figure 5.12 shows the magnitude of stress reduction achieved by our algorithm is generally 5× to 6×, but rapidly decays towards the boundary of the covered area. This is expected, as the subspace of shape parameters satisfying the elasticity tensor and printability constraints shrinks as we approach the boundary. We also observe a growth of maximal stress as Young's modulus decreases, since thinner trusses are needed to make the material sufficiently flexible. Because we perform optimization in a reduced parameter space, we do not claim that our structures are optimal. However, the relatively uniform stress distributions over their surfaces suggest that much greater reductions are unlikely.

Note that the initial coverage exploration and worst-case stress minimization are performed over the same design space: the initial designs against which we compare stress levels already have smooth joints, but our worst-case stress objective is needed to exploit the smoothing parameters to significantly reduce stress.

Figure 5.13 plots histograms of per-element stresses. We observe that a large number of elements with high stress disappear after optimization. We also note that the total number of elements has decreased because stress optimization generally eliminates the surface's high curvature regions, which require significant refinement.

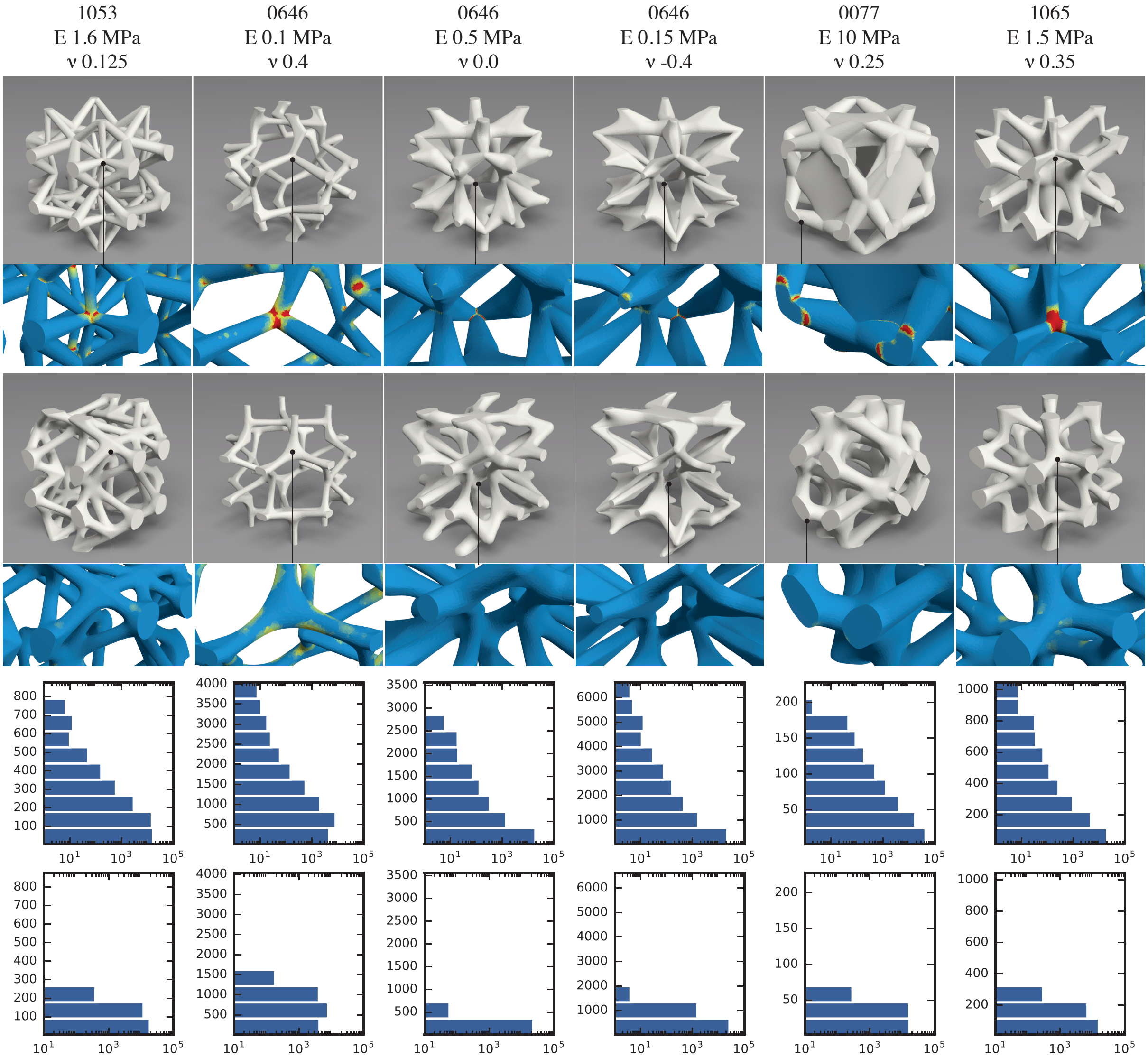


Figure 5.13: Significant stress reduction is achieved for a diverse set of structures. The top row visualizes structures in our pattern subspace found by fitting to $C^\star$ without penalizing stress. The moduli are reported assuming a base printing material, $C^{\text{base}} = (200MPa, 0.35)$. The bottom row displays structures achieving the same $C^\star$ but with dramatically reduced peak stress. Histograms of the element counts at each stress level (reported in MPa) in the unoptimized (top) and optimized (bottom) designs are plotted on a log scale.

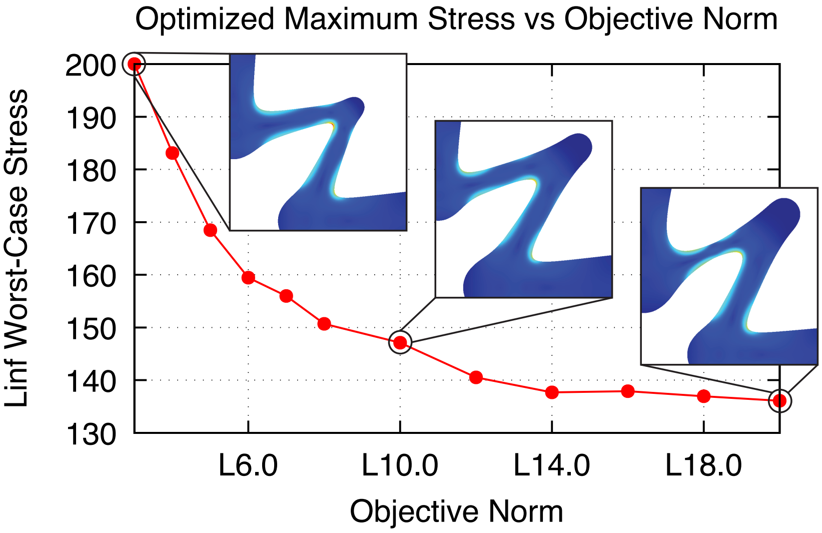


Figure 5.14: The peak worst-case stress level and corresponding structures designed by minimizing the L_p worst-case stress norm.

Choice of p in L_p The dependence of maximal stress achieved by L_p optimization on the chosen objective norm is shown in Figure 5.14. We observe that $p = 12$ is generally adequate for approximating the L_∞ stress norm for optimization purposes. At this value of p, the objective is highly non-smooth, necessitating the accurate computation of shape derivatives described in Section 5.5.

Resolution dependence We have also verified that the optimization's effectiveness at minimizing stress is insensitive to mesh resolution after a reasonable level of refinement is reached (Figure 5.15). We based our meshing settings for the sweep in Figure 5.12 on refinement experiments like this one.

Experimental validation Validating stress reduction experimentally is far more difficult than, e.g., measuring Young's modulus. First, stress concentrations are difficult to measure directly and are affected by small geometry variations caused by printing inaccuracies. Second, though we design our structures for robustness under worst-case loads, these loads are often difficult to apply in a laboratory setting: our testing setup limited

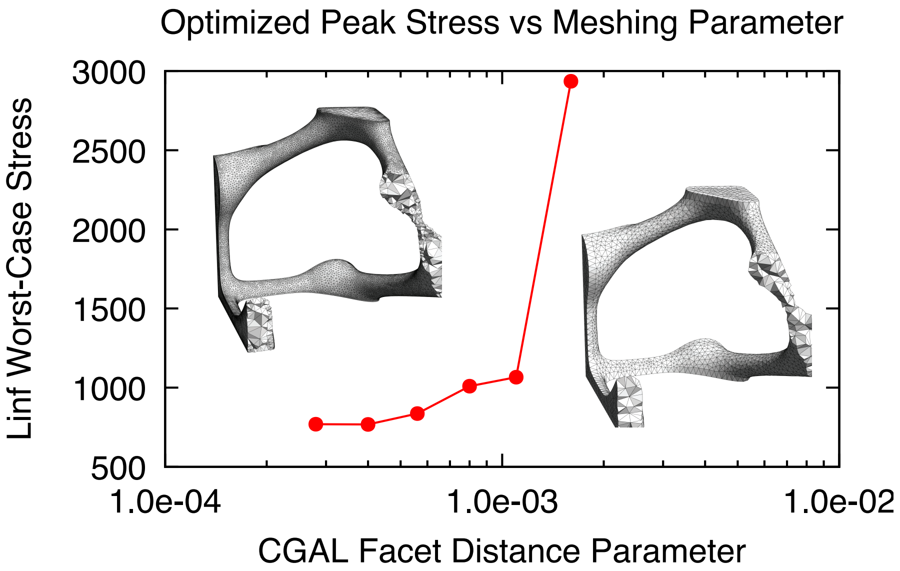


Figure 5.15: Optimizations run at increasing mesh resolution from right to left (controlled by adjusting CGAL's `facet_distance` parameter to adaptively refine the surface mesh based on curvature), with the L_p norm of the stress of the final iterate evaluated on the fine mesh.

us to compression tests, which do not reveal the full benefits of our optimization. Finally, testing small tilings ($2 \times 2 \times 1$) leads to errors, as our compression test does not match the periodic homogenization boundary conditions. These errors often manifest themselves as early fractures in the dangling bars touching the compression plates.

We have, however, consistently observed that our optimized patterns can withstand deformations much larger than the deformations breaking the corresponding unoptimized patterns (Figure 5.16, Figure 5.17). Homogenized moduli are reported based on an assumed isotropic printing material with a Young's modulus of 200MPa and a Poisson's ratio of 0.35. Note that curing time significantly affects the true stiffness; we ensured the unoptimized and optimized structures of each tested pair were post-processed identically.

Figure 5.18 shows the results of loading to a fixed deformation, unloading, and repeating to test for fractures and plastic deformation; we have observed a substantial decrease in strength for unoptimized patterns and no change for the optimized ones.

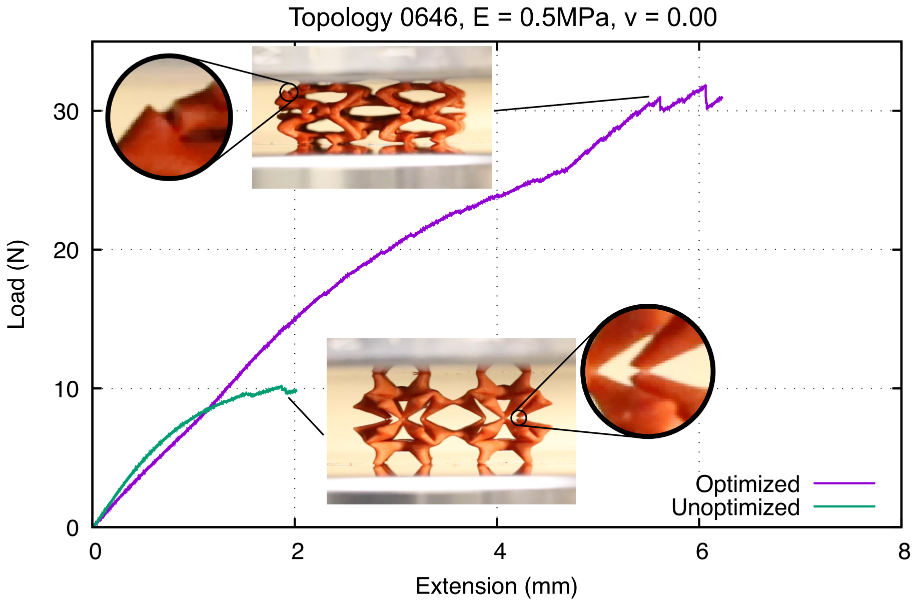


Figure 5.16: We compressed unoptimized and optimized structures designed for the same material tensor until breakage. Despite the unoptimized pattern ending up slightly stiffer due to printing inaccuracies, it fails under much lower deformation.

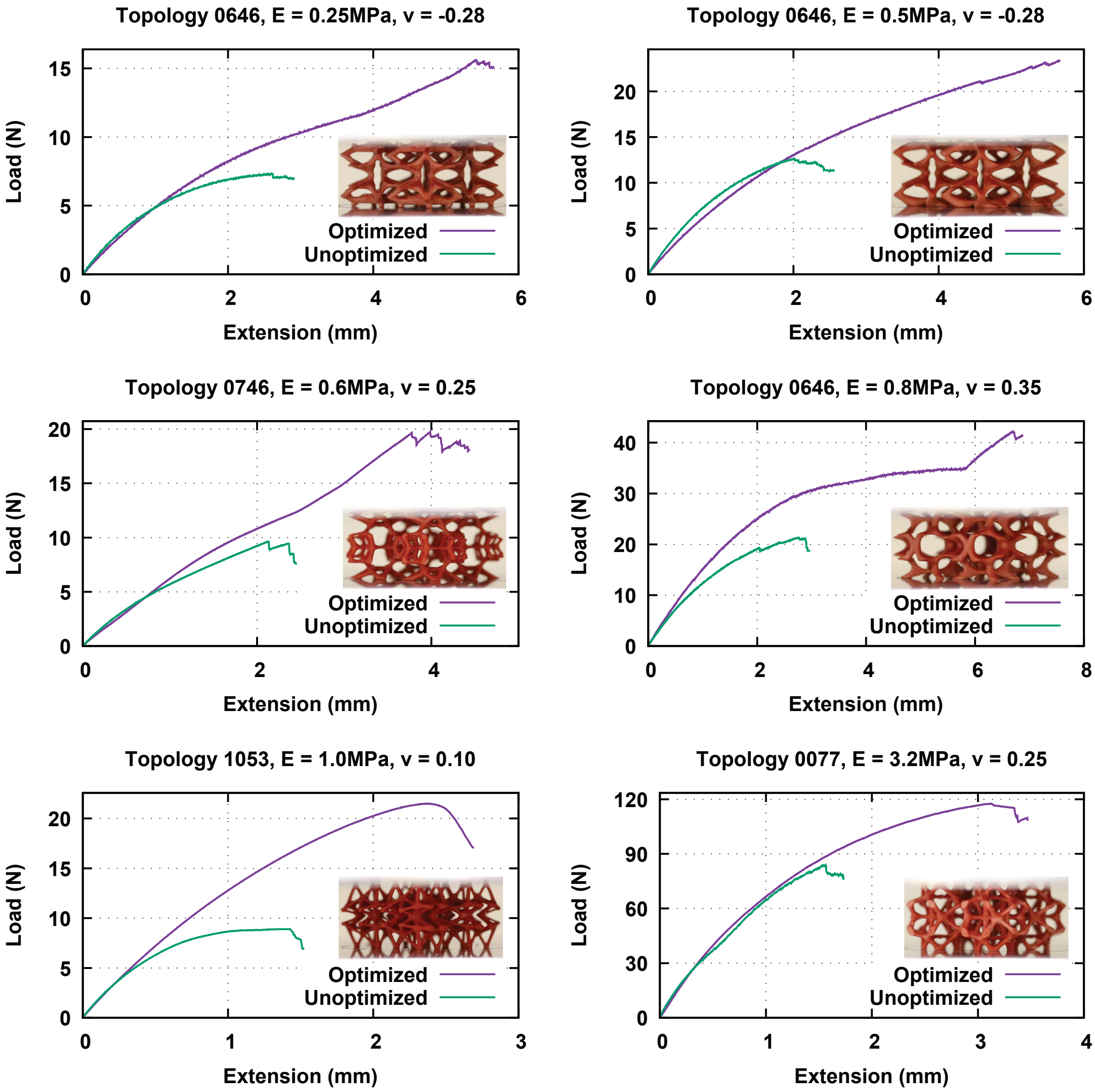


Figure 5.17: Additional compression test results for a variety of topologies and material properties. A photo of the optimized pattern of each pair is inset. The optimized structure in the bottom left example did not actually break: it simply buckled into a lower energy configuration.

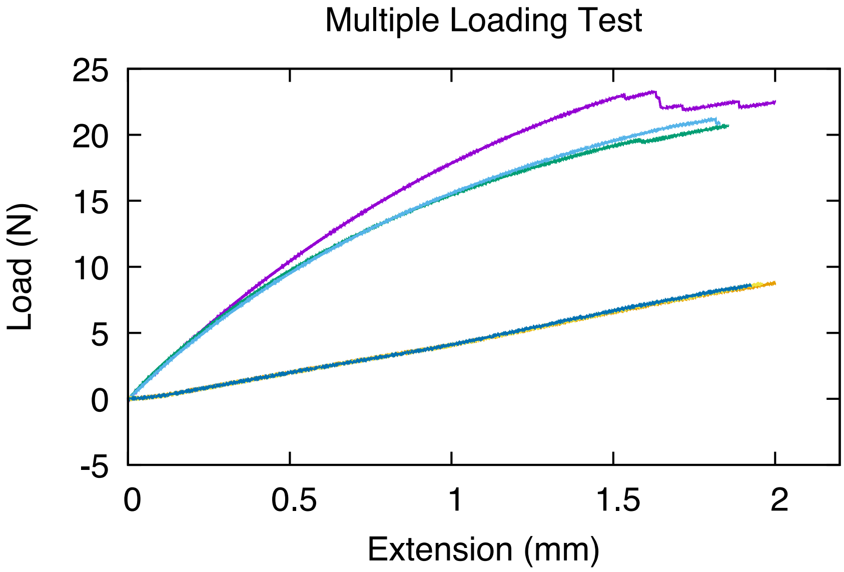


Figure 5.18: We compressed one optimized and unoptimized structure at $0.5mm/s$ to $2mm$. We then unloaded each structure and repeated the experiment twice. The optimized pattern traced out three overlapping lines at the bottom: its moduli were unchanged. The unoptimized structure broke on the first compression (top curve), causing weaker behavior on the two subsequent compressions.

5.8 Conclusions and future work

We have demonstrated that the problem of determining worst-case loads, considered by numerous previous works, admits an exact solution for the specific case of periodic structures. We have developed a robust set of algorithms to minimize worst-case stresses, achieving a 5× reduction for a significant share of patterns and substantially expanding the covered region of (E, ν) space versus previous work.

Our work has several limitations, which are all worth exploring further in future work. First, it is unclear how much further stress can be reduced while maintaining particular elasticity properties—a more in-depth study would be illuminating, but requires L_p stress optimization in the full-dimensional shape space. In our experience, this optimization is unstable, and it is not obvious how to constrain the optimization to printable designs. Second, it may be possible to cover a greater region of the material property space by considering finer topologies for the cells. Finally, one can apply our code to other physics-based design problems involving L_∞ stress optimization.

Chapter 6

Conclusion

This thesis has demonstrated that—by applying the mathematical frameworks of homogenization, shape optimization, and coarse-scale material optimization—we can leverage the high resolution of emerging additive fabrication technologies to achieve improved performance on practical design goals without attempting to solve immense design problems with billions of variables. Guiding our optimization by worst-case stress analysis, we can furthermore ensure the shapes we design are robust even as they incorporate structures at the microscopic scale. By incorporating fabrication constraints directly into the optimal design framework, we produce read-to-print designs and have begun to demonstrate the promise these tools hold for real-world applications. However, much work remains before the two-scale optimal design approach that we advocate is truly ready for practical applications.

For applications in the graphics community, where aesthetics are key, adapting the fine-scale structure to conform to a curved surface or applying an aesthetically appealing skin without violating the design goals are essential future work—at least until printing technology permits our structures to be fabricated at a truly microscopic scale. Already, one could manually specify a flexible skin enclosing the design domain and take the skin's properties into account while optimize the microstructure inside it. However, it's likely that a tool optimizing the skin in concert with the fine-scale structure will be necessary.

For many applications, it is important to analyze and optimize the nonlinear behavior of microstructures. This analysis is considerably more complicated than the linear periodic homogenization we employ since symmetry-breaking buckling can occur at many different length scales. But for large deformation applications, like articulated characters, it will be necessary. Structures can likely be optimized to significantly extend their linear elastic regime, producing more intuitive, predictable metamaterials. Indeed, the optimized structures tested in Figures 5.16 and 5.17 already achieve this to some extent.

A tighter coupling of the design problems at the macro- and micro-scales would likely improve results. Currently our two-scale approach does all microstructure design in a precomputation stage, building a database that can be quickly queried during the macroscale optimization. This approach was chosen because the microstructure design process is computationally expensive. If it becomes cheap enough to run in the inner loop of the coarse-scale design optimization (separately for each material variable in the coarse problem), improved designs could be achieved. For instance, preliminary experiments have shown that, if the macroscopic load is known, our framework can design a structure with noticeably reduced stresses compared to the worst-case-optimal design (of course, at the expense of robustness against other loads).

It would also be interesting to couple the macroscopic worst-case stress analysis tool from Chapter 4 with the microscopic analysis from Chapter 5 to analyze the worst-case stresses of a two-scale design.

Finally, while our work has focused on volumetric microstructures so far, it would be interesting to explore the application of surface (thinly extruded 2D) microstructures to control bending and stretching stiffness. The two-scale pipeline implemented in this thesis already works in 2D, but we have not explored how well the homogenized moduli of the 2D structures can predict the deformation of a finitely-tiled microstructured shell.

Appendix A

Worst-Case Stress Relief Additional Material

A.1 Worst-case maximum stress

The maximum principal stress at a point is given by (5.3):

$$s_m = \max_{\bar{\sigma}:\bar{\sigma}=1} \lambda_{\max}(F : \bar{\sigma}) = \max_{\bar{\sigma}:\bar{\sigma}=1} \max_{\|\mathbf{n}\|=1} \mathbf{n}^T[F : \bar{\sigma}]\mathbf{n}.$$

The Lagrangian for this maximization problem is

$$\mathcal{L}(\bar{\sigma}, \mathbf{n}, \lambda_{\bar{\sigma}}, \lambda_{\mathbf{n}}) = \mathbf{n}^T[F : \bar{\sigma}]\mathbf{n} + \lambda_{\bar{\sigma}} (\bar{\sigma} : \bar{\sigma} - 1) + \lambda_{\mathbf{n}} \left(\|\mathbf{n}\|^2 - 1\right).$$

The stationary condition with respect to $\bar{\sigma}$ is

$$\frac{\partial \mathcal{L}}{\partial \bar{\sigma}} = [\mathbf{n}\mathbf{n}^T] : F + 2\lambda_{\bar{\sigma}}\bar{\sigma} = 0, \tag{A.1}$$

showing that, for arbitrary $\mathbf{n}$, principal stress is maximized by load:

$$\bar{\sigma}^\star = -\frac{[\mathbf{n}\mathbf{n}^T] : F}{2\lambda_{\bar{\sigma}}} = \pm\frac{[\mathbf{n}\mathbf{n}^T] : F}{\sqrt{[\mathbf{n}\mathbf{n}^T] : F : F^T : [\mathbf{n}\mathbf{n}^T]}}$$

(since $\bar{\sigma}^\star$ must have unit Frobenius norm).

This allows us to rewrite the (squared) worst-case principal stress computation as a maximization over $\mathbf{n}$ only:

$$s_m^2 = \max_{\|\mathbf{n}\|^2=1} [\mathbf{n}\mathbf{n}^T] : \underbrace{F : F^T}_{T^M} : [\mathbf{n}\mathbf{n}^T], \tag{A.2}$$

which is a symmetric rank 4 tensor eigenvalue problem. It is straightforward to show that it is maximized by a $\mathbf{n}^\star$ such that

$$\mathbf{n}^\star \cdot T^M : [\mathbf{n}^\star \mathbf{n}^{\star T}] = s_m^2 \mathbf{n}^\star. \tag{A.3}$$

A.2 Eigenvalue Derivatives

Assume matrix A has a non-repeated maximum eigenvalue λ with corresponding unit eigenvector v. Then

$$\dot{\lambda} = \frac{\mathrm{d}}{\mathrm{d}t}(v^T A v) = v^T \dot{A} v + 2\lambda v^T \dot{v} = v^T \dot{A} v,$$

where we used the fact that $v(t)$ is a unit vector ($v^T \dot{v} = 0$). A similar argument holds for rank four tensor T with non-repeated maximum eigenvalue λ and corresponding unit eigenvector n:

$$\dot{\lambda} = \frac{\mathrm{d}}{\mathrm{d}t}\left(T_{ijkl} n_i n_j n_k n_l\right) = \dot{T}_{ijkl} n_i n_j n_k n_l + 4\lambda \dot{n}_i n_i = \dot{T}_{ijkl} n_i n_j n_k n_l.$$

In practice, elements with high stress have one dominant maximum eigenvalue (Figure A.1), so we do not need to worry about a repeated eigenvalue breaking our L_p objective's differentiability

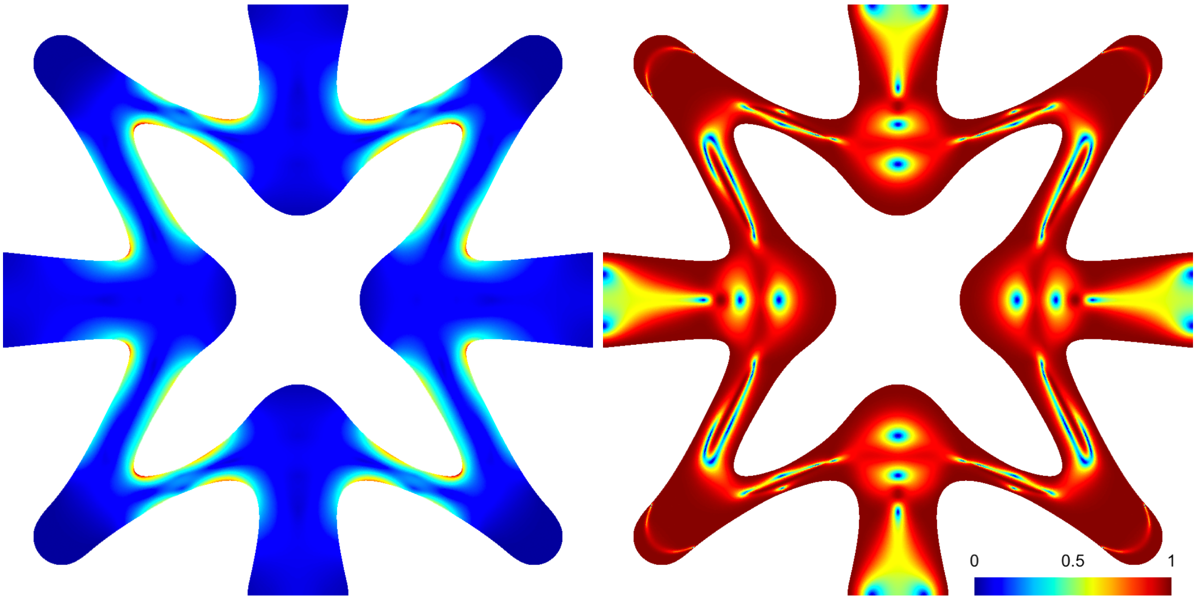


Figure A.1: Regions of high worst-case stress (left) have the greatest difference in eigenvalues (1.0, on right).

A.3 Shape derivative for model problem

We illustrate the inaccuracy of the traditional formula for shape derivatives in a simpler setting:

$$J(\omega) = \int_\omega j(\nabla u)\ \mathrm{d}\mathbf{x} \qquad \text{where } u \text{ solves } \begin{aligned} -\nabla \cdot (\nabla u + g) &= 0 \text{ in } \omega \\ \mathbf{n} \cdot (\nabla u + g) &= 0 \text{ on } \partial\omega. \end{aligned}$$

Later, we will need the weak form of the constraint PDE:

$$\int_\omega \nabla\phi \cdot (\nabla u + g)\ \mathrm{d}\mathbf{x} = 0 \quad \forall \text{ trial functions } \phi. \tag{A.4}$$

Here, g is analogous to the macroscopic strain applied in homogenization.

Computing $\mathrm{d}J[\mathbf{v}]$ in terms of the Eulerian derivative, $\dot{u}$. We seek an expression for $\mathrm{d}J[\mathbf{v}]$, J's initial rate of change as ω is perturbed by velocity field $\mathbf{v}$ into altered shape ω_t. We do this by applying Reynolds Transport Theorem to J, showing a step-by-step derivation of the theorem. Our approach is to express all quantities on the unperturbed reference domain, ω, and then differentiate with respect to "time" t:

$$\mathrm{d}J[\mathbf{v}] \overset{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0} \int_{\omega_t} j(\nabla_t u_t)\ \mathrm{d}\mathbf{x}_t = \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t=0} \int_{\omega} j(F_t^{-T}\nabla \hat{u}_t)\det(F_t)\ \mathrm{d}\mathbf{x},$$

where $\mathbf{x}_t = \mathbf{x} + t\mathbf{v}$ and ∇_t are the spatial variable and gradient for the perturbed domain, ω_t. We have defined state function u_t on ω_t in terms of function $\hat{u}_t$ defined on the reference domain ($u_t(\mathbf{x}_t) = \hat{u}(\mathbf{x}_t - t\mathbf{v})$). Finally, $F_t = I + t\nabla\mathbf{v}$ is the Jacobian of the mapping from ω to ω_t; it is used to re-express the perturbed domain's gradient operator and volume element in terms of the reference domain quantities.

Now that the integration domain is fixed, we can move the time derivative inside to compute:

$$\mathrm{d}J[\mathbf{v}] = \int_{\omega} (j') \cdot \left(-(\nabla\mathbf{v})^T \nabla u + \nabla \left.\frac{\partial \hat{u}_t}{\partial t}\right|_{t=0} \right) + j(\nabla u)\nabla\cdot\mathbf{v}\ \mathrm{d}\mathbf{x}, \tag{A.5}$$

using the identities $\frac{\partial}{\partial t}\big|_{t=0}(I+\nabla\mathbf{v})^{-T} = -(\nabla\mathbf{v})^T$, $\frac{\partial}{\partial t}\big|_{t=0}\det(I+\nabla\mathbf{v}) = \nabla\cdot\mathbf{v}$, and $\hat{u}_0 = u$. Here, $\frac{\partial \hat{u}_t}{\partial t}\big|_{t=0}$ is the *material derivative* at time $t = 0$. We denote it by δu and note its relationship to the Eulerian derivative $\dot{u} \overset{\text{def}}{=} \frac{\partial u_t}{\partial t}\big|_{t=0} = \delta u - \nabla u \cdot \mathbf{v}$. The material derivative's gradient can therefore be written as:

$$\nabla \delta u = \nabla \dot{u} + \nabla\left(\nabla u \cdot \mathbf{v}\right) = \nabla\dot{u} + (\mathbf{v}\cdot\nabla)\nabla u + (\nabla\mathbf{v})^T\nabla u. \tag{A.6}$$

We use this relationship to simplify (A.5). Substituting the rightmost expression for

$\nabla \left. \frac{\partial \hat{u}_t}{\partial t} \right|_{t=0}$:

$$\mathrm{d}J[\mathbf{v}] = \int_\omega (j') \cdot \Big(\nabla \dot{u} + (\mathbf{v} \cdot \nabla) \nabla u \Big) + j(\nabla u) \nabla \cdot \mathbf{v} \; \mathrm{d}\mathbf{x}.$$

Finally, we apply the integration by parts

$$\int_\omega (j) \nabla \cdot \mathbf{v} \; \mathrm{d}\mathbf{x} = -\int_\omega (\mathbf{v} \cdot \nabla) j \; \mathrm{d}\mathbf{x} + \int_{\partial\omega} (j) \mathbf{v} \cdot \mathbf{n} \; \mathrm{d}A(\mathbf{x})$$

to arrive at the simplified formula by cancellation:

$$\boxed{\mathrm{d}J[\mathbf{v}] = \int_\omega j'(\nabla u) \cdot \nabla \dot{u} \; \mathrm{d}\mathbf{x} + \int_{\partial\omega} j(\nabla u) \mathbf{v} \cdot \mathbf{n} \; \mathrm{d}A(\mathbf{x}).} \tag{A.7}$$

Solving for $\dot{u}$. Formula (A.7) requires $\dot{u}$, which we find by differentiating both sides of the constraint's weak form, (A.4). We do this using a second step-by-step application of Reynolds Transport Theorem (first re-expressing the weak form for ω_t on reference domain ω):

$$\left. \frac{\mathrm{d}}{\mathrm{d}t} \right|_{t=0} \int_\omega (F_t^{-T} \nabla \phi) \cdot (F_t^{-T} \nabla \hat{u}_t + g) \det(F_t) \; \mathrm{d}\mathbf{x} = 0 \quad \forall \phi,$$

and manipulating this equation into a PDE for $\dot{u}$. Here, we defined shape functions on the perturbed domain by evaluating ω's shape functions, ϕ, at the material coordinates; in the discrete setting, this definition coincides with the shape functions one would construct on the perturbed finite element mesh (without remeshing). Following the same steps as for differentiating J, we obtain:

$$\int_\omega - \big((\nabla \mathbf{v})^T \nabla \phi \big) \cdot (\nabla u + g) - \nabla \phi \cdot (\nabla \mathbf{v})^T \nabla u + \nabla \phi \cdot \nabla \delta u + \nabla \phi \cdot (\nabla u + g) \nabla \cdot \mathbf{v} \; \mathrm{d}\mathbf{x} = 0 \quad \forall \phi. \tag{A.8}$$

We apply (A.6) to express this as an equation for $\dot{u}$:

$$\int_\omega \nabla \phi \cdot \Big(-\nabla \mathbf{v} (\nabla u + g) - \cancel{(\nabla \mathbf{v})^T \nabla u} + \big[\nabla \dot{u} + (\mathbf{v} \cdot \nabla) \nabla u + \cancel{(\nabla \mathbf{v})^T \nabla u} \big] + (\nabla u + g) \nabla \cdot \mathbf{v} \Big) \; \mathrm{d}\mathbf{x} = 0 \quad \forall \phi.$$

Applying integration by parts to the last integrand, the left-hand side becomes:

$$\int_\omega \nabla\phi\cdot\Big(-\nabla\mathbf{v}(\nabla u+g) + \nabla\dot{u} + (\mathbf{v}\cdot\nabla)\nabla u\Big) - (\mathbf{v}\cdot\nabla)\Big(\nabla\phi\cdot(\nabla u+g)\Big)\ \mathrm{d}\mathbf{x}$$
$$+\int_{\partial\omega}\nabla\phi\cdot(\nabla u+g)(\mathbf{v}\cdot\mathbf{n})\ \mathrm{d}A(\mathbf{x})$$

Simplifying, we arrive at the PDE for $\dot{u}$ in weak form:

$$\underbrace{-\int_\omega \nabla(\nabla\phi\cdot\mathbf{v})\cdot(\nabla u+g)\ \mathrm{d}\mathbf{x}}_{I} + \int_\omega \nabla\phi\cdot\nabla\dot{u}\ \mathrm{d}\mathbf{x} + \int_{\partial\omega}\nabla\phi\cdot(\nabla u+g)(\mathbf{v}\cdot\mathbf{n})\ \mathrm{d}A(\mathbf{x}) = 0 \quad \forall\phi. \tag{A.9}$$

To obtain the traditional boundary integral formula for the shape derivative, we must drop term I. Indeed, provided $(\nabla\phi\cdot\mathbf{v})$ lies is the space of test functions, this term vanishes because u solves (A.4). However, *this is precisely the term that does not vanish* for our Lagrange finite elements. For the moment, we will drop it to show how to arrive at the traditional shape derivative formula:

$$\int_\omega \nabla\phi\cdot\nabla\dot{u}\ \mathrm{d}\mathbf{x} + \int_{\partial\omega}\nabla\phi\cdot(\nabla u+g)(\mathbf{v}\cdot\mathbf{n})\ \mathrm{d}A(\mathbf{x}) = 0 \quad \forall\phi. \tag{A.10}$$

Applying the adjoint method. We apply the adjoint method to express $\mathrm{d}J[\mathbf{v}]$ as an explicit differential form, avoiding the need to solve for $\dot{u}$ for every perturbation $\mathbf{v}$. Suppose we can find a scalar field p in our space of test functions so that:

$$\int_\omega (j')\cdot\nabla\psi\ \mathrm{d}\mathbf{x} = \int_\omega \nabla p\cdot\nabla\psi\ \mathrm{d}\mathbf{x} \quad \forall \text{ trial functions } \psi. \tag{A.11}$$

Then, taking $\psi = \dot{u}$:

$$\int_\omega (j')\cdot\nabla\dot{u}\ \mathrm{d}\mathbf{x} = \int_\omega \nabla p\cdot\nabla\dot{u}\ \mathrm{d}\mathbf{x} = -\int_{\partial\omega}\nabla p\cdot(\nabla u+g)(\mathbf{v}\cdot\mathbf{n})\ \mathrm{d}A(\mathbf{x}),$$

where the last step used (A.10) with p replacing ϕ. Plugging this into (A.7), we arrive at the standard formula:

$$\boxed{\mathrm{d}J[\mathbf{v}] = \int_{\partial\omega} \Big(j(\nabla u) - \nabla p \cdot (\nabla u + g) \Big) (\mathbf{v} \cdot \mathbf{n}) \, \mathrm{d}A(\mathbf{x}).} \tag{A.12}$$

The weak form (A.11) corresponds to the adjoint PDE,

$$-\Delta p = -\nabla \cdot j'(\nabla u) \text{ in } \omega, \qquad \frac{\partial p}{\partial \mathbf{n}} = \mathbf{n} \cdot j'(\nabla u) \text{ on } \partial\omega.$$

A.4 Traditional shape derivative formula for worst-case stress

Now we compute the shape derivative in boundary integral form for our worst-case stress objective. For simplicity, we consider the worst-case Frobenius norm stress measure:

$$J(\omega) = \int_{\omega} j(\bar{\sigma}^{\star}(\mathbf{x}) : T^F(\mathbf{x}) : \bar{\sigma}^{\star}(\mathbf{x})) \, \mathrm{d}\mathbf{x},$$

but the other stress measures' derivations are nearly identical.

To shape-differentiate J, it will be necessary to know how j changes at each $\mathbf{x}$ when the fluctuation strains change. As mentioned in the main text, the worst-case load $\bar{\sigma}^{\star}$ can be considered constant, but $T^F(\mathbf{x}) = F^T : F$ is a function of the fluctuation strains via (5.2) from the main text. Furthermore, though the homogenized elasticity tensor C^H is technically a function of the fluctuation strains, it simplifies our derivation to view C^H as an independent parameter of s and then separately compute its shape derivative, (A.24). Representing these relationships explicitly, we write:

$$j(s(\mathbf{x})) := j(s(\varepsilon^{pq}, C^H, \mathbf{x})),$$

where ε^{pq} expands to six strain field arguments in 3D. Perturbations $\delta\varepsilon^{kl}$ and δC^H of these arguments induce perturbation:

$$\begin{aligned}\delta j &= (j')\frac{\partial s}{\partial \varepsilon^{kl}} : \delta\varepsilon^{kl} + (j')\frac{\partial s}{\partial C^H} :: \delta C^H \\ &= \tau^{kl} : \delta\varepsilon^{kl} + \gamma : \delta C^H,\end{aligned}$$

where we defined

$$\tau^{kl} \overset{\text{def}}{=} (j')\frac{\partial s}{\partial \varepsilon^{kl}}, \quad \gamma \overset{\text{def}}{=} (j')\frac{\partial s}{\partial C^H}.$$

A.4.1 Computing τ^{kl} and γ

First, we compute the rank-two tensor field τ^{kl} expressing the derivative of objective integrand j with respect to fluctuation strain ε^{kl} (holding C^H and thus S^H constant).

$$\begin{aligned}\tau^{kl}_{ij} =& j' \frac{\partial}{\partial \varepsilon^{kl}_{ij}}\left[\bar{\sigma}^\star : F^T : F : \bar{\sigma}^\star\right] = j'\bar{\sigma}^\star : \left[\left(\frac{\partial}{\partial \varepsilon^{kl}_{ij}} F^T\right) : F + F^T : \frac{\partial}{\partial \varepsilon^{kl}_{ij}} F\right] : \bar{\sigma}^\star, \\ =& 2j'\bar{\sigma}^\star : F^T : \left(\frac{\partial}{\partial \varepsilon^{kl}_{ij}} F\right) : \bar{\sigma}^\star,\end{aligned}$$

using the fact that a tensor and its transpose give the same quadratic form. From definition (5.2) in the main text,

$$\frac{\partial}{\partial \varepsilon^{kl}_{ij}} F_{abcd} = C^{\text{base}}_{abef} S^H_{ghcd} \frac{\partial}{\partial \varepsilon^{kl}_{ij}} (\varepsilon^{gh}_{ef} + e^{gh}_{ef}) = C^{\text{base}}_{abef} S^H_{ghcd} \delta_{gk}\delta_{hl}\delta_{ei}\delta_{fj} = C^{\text{base}}_{abij} S^H_{klcd}.$$

After simplification, we have

$$\tau^{kl} = (2j' C^{\text{base}} : F : \bar{\sigma}^\star)[S^H : \bar{\sigma}^\star]_{kl}. \tag{A.13}$$

Next, we compute the rank-four tensor field γ expressing the partial derivative of objective integrand j with respect to the homogenized elasticity tensor C^H.

$$\begin{aligned}
\delta j &= 2j'\bar{\sigma}^\star : F^T : C^{\text{base}} : G : \mathrm{d}S^H : \bar{\sigma}^\star, \\
&= 2j'\bar{\sigma}^\star : F^T : C^{\text{base}} : G : (-S^H : \mathrm{d}C^H : S^H) : \bar{\sigma}^\star, \\
&= (-2j'\bar{\sigma}^\star : F^T : F) : \mathrm{d}C^H : (S^H : \bar{\sigma}^\star), \\
&= \left[(-2j'\bar{\sigma}^\star : F^T : F) \otimes (S^H : \bar{\sigma}^\star)\right] :: \mathrm{d}C^H.
\end{aligned}$$

Thus,

$$\gamma = (-2j'F^T : F : \bar{\sigma}^\star) \otimes (S^H : \bar{\sigma}^\star). \tag{A.14}$$

Now, applying the Reynolds Transport Theorem to our objective, we find:

$$\begin{aligned}
\mathrm{d}J[\mathbf{v}] &= \int_{\partial\omega} (\mathbf{v} \cdot \hat{\mathbf{n}}) j \;\mathrm{d}A(\mathbf{x}) + \int_\omega \tau^{kl} : \varepsilon(\dot{\mathbf{w}}^{kl}[\mathbf{v}]) + \gamma :: \mathrm{d}C^H[\mathbf{v}] \;\mathrm{d}\mathbf{x} \\
&= \int_{\partial\omega} (\mathbf{v} \cdot \hat{\mathbf{n}}) j \;\mathrm{d}A(\mathbf{x}) + \underbrace{\int_\omega \tau^{kl} : \varepsilon(\dot{\mathbf{w}}^{kl}[\mathbf{v}]) \;\mathrm{d}\mathbf{x}}_{I} + \left(\int_\omega \gamma \;\mathrm{d}\mathbf{x}\right) :: \mathrm{d}C^H[\mathbf{v}],
\end{aligned} \tag{A.15}$$

where $\dot{\mathbf{w}}^{kl}[v]$ is an Eulerian derivative with respect to "time" t under the advection velocity $\mathbf{v}$. The first and third terms are straightforward to evaluate (once we derive a formula for $\mathrm{d}C^H[\mathbf{v}]$), but the middle integral "I" involves the problematic term $\dot{\mathbf{w}}^{kl}[v]$, which measures how the fluctuation displacements change when perturbing the shape with velocity field $\mathbf{v}$.

Forward version

The "forward" sensitivity analysis determines, for a particular velocity field $\mathbf{v}$, the change in fluctuation displacements $\dot{\mathbf{w}}^{kl}[\mathbf{v}]$ and substitutes them into (A.15). We can determine an equation for $\dot{\mathbf{w}}^{kl}[\mathbf{v}]$ by differentiating the weak form of the kl^{th} cell problem. To simplify the derivation, we apply our periodic boundary conditions and no-rigid-translation constraints directly to the space of trial and test functions. Then the cell problem's weak form is just

$$\int_{\omega} \varepsilon(\phi) : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \, \mathrm{d}\mathbf{x} = 0 \quad \text{(for all } \phi\text{)}, \tag{A.16}$$

where $\mathbf{w}^{kl}$ and ϕ are periodic vector fields on the unit cell Y. Differentiating both sides of this equation by naïvely applying Reynolds Transport Theorem (assuming shape functions, ϕ, are independent of ω),

$$\int_{\partial\omega} (\mathbf{v} \cdot \hat{\mathbf{n}}) \Big(\varepsilon(\phi) : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \Big) \, \mathrm{d}A(\mathbf{x}) + \int_{\omega} \varepsilon(\phi) : C : \varepsilon(\dot{\mathbf{w}}^{kl}[\mathbf{v}]) \, \mathrm{d}\mathbf{x} = 0 \quad \text{(for all } \phi\text{)}, \tag{A.17}$$

which is the weak from of a cell problem for $\dot{\mathbf{w}}^{kl}[\mathbf{v}]$. Once we solve this equation for each $\dot{\mathbf{w}}^{kl}[\mathbf{v}]$, we can compute (A.15) easily.

Adjoint version

We determine the adjoint equations by noticing the following: suppose we can find an "adjoint solution" $\mathbf{p}^{kl}$ from the same space as ϕ (i.e., a periodic test function for the original PDE) such that

$$\int_{\omega} \tau^{kl} : \varepsilon(\psi) \, \mathrm{d}\mathbf{x} = \int_{\omega} \varepsilon(\mathbf{p}^{kl}) : C : \varepsilon(\psi) \, \mathrm{d}\mathbf{x} \quad \text{(for all } \psi\text{)}, \tag{A.18}$$

where ψ is from the same space as $\dot{\mathbf{w}}^{kl}$ (i.e., a periodic trial function for the original PDE). Then we can use $\mathbf{p}^{kl}$ to compute integral I as follows:

$$\begin{aligned} I = \int_{\omega} \tau^{kl} : \varepsilon(\dot{\mathbf{w}}^{kl}) \, \mathrm{d}\mathbf{x} &= \int_{\omega} \varepsilon(\mathbf{p}^{kl}) : C : \varepsilon(\dot{\mathbf{w}}^{kl}) \, \mathrm{d}\mathbf{x} \\ = -\int_{\partial\omega} (\mathbf{v} \cdot \hat{\mathbf{n}}) &\Big(\varepsilon(\mathbf{p}^{kl}) : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \Big) \, \mathrm{d}A(\mathbf{x}). \end{aligned}$$

The second step follows by substituting $\dot{\mathbf{w}}^{kl}$ for ψ in (A.18), and the third by substituting $\mathbf{p}^{kl}$ for ϕ in (A.17). Using this formula, our full shape derivative can be computed efficiently as:

$$\boxed{\mathrm{d}J[\mathbf{v}] = \int_{\partial\omega} \left(j - \varepsilon(\mathbf{p}^{kl}) : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \right) \mathbf{v} \cdot \hat{\mathbf{n}} \, \mathrm{d}A(\mathbf{x}) + \left(\int_{\omega} \gamma \, \mathrm{d}\mathbf{x} \right) :: dC^H[\mathbf{v}],} \tag{A.19}$$

summing over kl. We recognize (A.18) as the weak form of the adjoint cell problem PDE:

$$\begin{aligned} -\nabla \cdot \sigma(\mathbf{p}^{kl}) &= -\nabla \cdot \tau^{kl} && \text{in } \omega, \\ \sigma(\mathbf{p}^{kl})\hat{\mathbf{n}} &= \tau^{kl}\hat{\mathbf{n}} && \text{on } \partial\omega, \\ \mathbf{p}^{kl} \text{ periodic,} & && \int_{\omega} \mathbf{p}^{kl} \, \mathrm{d}\mathbf{x} = 0. \end{aligned}$$

A.5 Accurate discrete formulation (volume form)

Since we consider straight-edged finite elements, the perturbation velocity $\mathbf{v}$ is a piecewise linear vector field and is represented as a perturbation vector on each mesh vertex:

$$\mathbf{v} = \sum_i \lambda_i \delta\mathbf{q}_i \ ,$$

where λ_i is vertex i's linear shape function (barycentric coordinates) and $\delta\mathbf{q}_i$ is its perturbation. Unfortunately, the most straight-forward approach to computing shape derivatives of plugging this piecewise linear $\mathbf{v}$ into a discretized version of (A.19) leads to wildly inaccurate results for the high L_p norms needed to prevent stress concentrations.

It turns out that the Reynolds Transport Theorem and the Eulerian derivatives used in (A.15) are the source of the error. As in the model problem, where proceeding from (A.8) to (A.10) introduced an error, the analogous step for our objective is problematic. We avoid this step by keeping everything in terms of material derivatives.

A.5.1 Discrete sensitivity of the objective

The analog to (A.5) in our setting is:

$$\begin{aligned}
\mathrm{d}J[\mathbf{v}] &= \int_\omega j\nabla\cdot\mathbf{v} + \tau^{kl} : D[\varepsilon(\mathbf{w}^{kl})] + \gamma :: dC^H[v]\ \mathrm{d}\mathbf{x} \\
&= \int_\omega j\nabla\cdot\mathbf{v} + \underbrace{\tau^{kl} : \varepsilon(D[\mathbf{w}^{kl}])}_{II} - \tau^{kl} : \mathrm{sym}(\nabla\mathbf{w}^{kl}\nabla\mathbf{v})\ \mathrm{d}\mathbf{x} + \left(\int_\omega \gamma\ \mathrm{d}\mathbf{x}\right) :: \mathrm{d}C^H[\mathbf{v}],
\end{aligned} \tag{A.20}$$

where $D[\cdot]$ denotes the material derivative, and we use ∇ applied to a vector field to denote the Jacobian (not its transpose). Note that $D[\cdot]$ and $\varepsilon(\cdot)$ do not commute, but the following identity holds for any linear combination, $\mathbf{w}$, of shape functions, ϕ:

$$D[\varepsilon(\mathbf{w})] = \varepsilon(D[\mathbf{w}]) - \mathrm{sym}(\nabla\mathbf{w}\nabla\mathbf{v}). \tag{A.21}$$

Again, II is the difficult term to compute.

Discrete Forward Sensitivity of $\mathbf{w}^{kl}$

We can determine $D[\mathbf{w}^{kl}]$, the material derivative of the fluctuation displacements, by differentiating the weak form (A.16). First, we define microscopic stress $\sigma^{kl} \overset{\text{def}}{=} C : \left[\varepsilon(\mathbf{w}^{kl}) + e^{kl}\right]$ to simplify notation. Then, differentiating both sides of the weak form:

$$\begin{aligned}
\forall\phi : \quad 0 &= \int_\omega \left(\varepsilon(\phi) : \sigma^{kl}\right)\nabla\cdot\mathbf{v} + D\left[\varepsilon(\phi) : C : \left[\varepsilon(\mathbf{w}^{kl}) + e^{kl}\right]\right]\ \mathrm{d}\mathbf{x} \quad \text{(for all } \phi\text{)}, \\
&= \int_\omega \left(\varepsilon(\phi) : \sigma^{kl}\right)\nabla\cdot\mathbf{v} - \mathrm{sym}(\nabla\phi\nabla\mathbf{v}) : \sigma^{kl} \\
&\qquad + \varepsilon(\phi) : C : \left(\varepsilon(D[\mathbf{w}^{kl}]) - \mathrm{sym}(\nabla\mathbf{w}^{kl}\nabla\mathbf{v})\right)\ \mathrm{d}\mathbf{x},
\end{aligned} \tag{A.22}$$

where we used the fact that $D[\phi] = 0$ because the test functions for straight-edged finite elements are expressed in terms of the mesh's barycentric coordinate functions and thus are tied to material points (i.e. their values advect with the mesh and have zero material derivative). (A.22) is the weak form of a PDE solving for $D[\mathbf{w}^{kl}]$, which can be discretized in the straight-forward way: as a vector holding the material derivative of $\mathbf{w}^{kl}$ at each mesh node. Notice that this equation is the analog of (A.8).

Discrete adjoint sensitivity

To obtain an explicit representation of the differential form accepting the perturbation velocity fields on ω and outputting a change in the objective, we must apply the adjoint method.

The adjoint equations turn out to be identical to (A.18) due to the similarity of integrals I and II; simply substitute $D[\mathbf{w}^{kl}]$ for $\dot{\mathbf{w}}^{kl}$ in the derivation. However, once we have the adjoint solutions $\mathbf{p}^{kl}$, the exact discrete gradient differs from (A.19). Instead, we derive it by computing II as follows: First, substitute $D[\mathbf{w}^{kl}]$ for ψ in (A.18) to determine:

$$II = \int_\omega \tau^{kl} : \varepsilon(D[\mathbf{w}^{kl}]) \, \mathrm{d}\mathbf{x} = \int_\omega \varepsilon(\mathbf{p}^{kl}) : C : \varepsilon(D[\mathbf{w}^{kl}]) \, \mathrm{d}\mathbf{x}$$

Next, substitute $\mathbf{p}^{kl}$ for ϕ in (A.22) to rewrite the first integrand again:

$$II = \int_\omega - \left[\varepsilon(\mathbf{p}^{kl}) : \sigma^{kl}\right] \nabla \cdot \mathbf{v} + \mathrm{sym}(\nabla \mathbf{p}^{kl} \nabla \mathbf{v}) : \sigma^{kl} + \varepsilon(\mathbf{p}^{kl}) : C : \mathrm{sym}(\nabla \mathbf{w}^{kl} \nabla \mathbf{v}) \, \mathrm{d}\mathbf{x}.$$

Finally, the full discrete shape derivative is evaluated as:

$$\boxed{\begin{aligned} \mathrm{d}J_d[\mathbf{v}] = \int_\omega & \left[j - \varepsilon(\mathbf{p}^{kl}) : \sigma^{kl}\right] \nabla \cdot \mathbf{v} + (\nabla \mathbf{p}^{kl} \nabla \mathbf{v}) : \sigma^{kl} + (\varepsilon(\mathbf{p}^{kl}) : C - \tau^{kl}) : (\nabla \mathbf{w}^{kl} \nabla \mathbf{v}) \, \mathrm{d}\mathbf{x} \\ & + \left(\int_\omega \gamma \, \mathrm{d}\mathbf{x}\right) :: dC^H[\mathbf{v}], \end{aligned}} \tag{A.23}$$

which gives the exact discrete shape derivative when the piecewise polynomial FEM fields are substituted for $\mathbf{p}^{kl}, \mathbf{w}^{kl}$, and $\mathbf{v}$. We dropped the symmetrization operator $\operatorname{sym}(\cdot)$ since its output is always double contracted with a symmetric tensor.

Discrete Differential Form

It is convenient to express $\mathrm{d}J_d[\mathbf{v}]$ as an explicit inner product with the per-vertex perturbation vector field $\delta\mathbf{q}$. To do this, we must re-express the terms involving $\mathbf{v}$ in terms of $\delta\mathbf{q}$. The easiest is $\nabla \cdot \mathbf{v} = \sum_m \nabla\lambda_m \cdot \delta\mathbf{q}_m$. The terms like $\tau^{kl} : (\nabla\mathbf{p}^{kl}\nabla\mathbf{v})$ take more work. Recalling that we take ∇ to represent the Jacobian when applied to vectors (rather than its transpose),

$$\nabla\mathbf{v} = \sum_m \delta\mathbf{q}_m \otimes \nabla\lambda_m.$$

We can write $\nabla\mathbf{p}^{kl}$ in terms of each *scalar*-valued finite element shape function φ_n and its *vector*-valued coefficient $\mathbf{p}_n^{kl}$ as:

$$\nabla\mathbf{p}^{kl} = \sum_n \mathbf{p}_n^{kl} \otimes \nabla\varphi_n.$$

Plugging these Jacobian expressions into the double contraction we wish to compute:

$$\begin{aligned}\tau^{kl} : (\nabla\mathbf{p}^{kl}\nabla\mathbf{v}) &= \sum_{n,m} \tau^{kl} : \left[(\mathbf{p}_n^{kl} \otimes \nabla\varphi_n)(\delta\mathbf{q}_m \otimes \nabla\lambda_m)\right] \\ &= \sum_m \delta\mathbf{q}_m \cdot \left(\sum_n \left[\nabla\lambda_m \cdot (\tau^{kl}\mathbf{p}_n^{kl})\right] \nabla\varphi_n\right).\end{aligned}$$

Finally, we make these substitutions in $\mathrm{d}J_d[\mathbf{v}]$ to express the differential form as an inner product with the vertex node perturbations (here summation over vertices, m, and FEM nodes, n, is implied):

$$\boxed{\begin{aligned}&\mathrm{d}J_d[\lambda_m \delta\mathbf{q}_m] = \\ &\qquad\left(\int_\omega \left[j - \varepsilon(\mathbf{p}^{kl}) : \sigma^{kl}\right] \nabla\lambda_m + \left[\nabla\lambda_m \cdot \left(\sigma^{kl}\mathbf{p}^{kl}_n + (\varepsilon(\mathbf{p}^{kl}) : C - \tau^{kl})\mathbf{w}^{kl}_n\right)\right] \nabla\varphi_n \; \mathrm{d}\mathbf{x}\right) \cdot \delta\mathbf{q}_m \\ &\quad + \left(\int_\omega \gamma \; \mathrm{d}\mathbf{x}\right) :: dC^H[\mathbf{v}].\end{aligned}}$$

Homogenized Tensor Discrete Shape Derivative

The evaluation is completed once we substitute the discrete formula for $\mathrm{d}C^H$. We start with the "energy form" of the homogenized tensor (2.22):

$$C^H_{ijkl} = \frac{1}{|Y|}\int_\omega [\varepsilon(\mathbf{w}^{ij}) + e^{ij}] : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \; \mathrm{d}\mathbf{x}.$$

Applying the analog of (A.5) for this expression:

$$\begin{aligned}\mathrm{d}C^H_{ijkl}[\mathbf{v}] = \frac{1}{|Y|}\int_\omega \left(\sigma^{ij} : C^{-1} : \sigma^{kl}\right)\nabla \cdot \mathbf{v} + \left(\varepsilon(D[\mathbf{w}^{ij}]) - \operatorname{sym}(\nabla\mathbf{w}^{ij}\nabla\mathbf{v})\right) : C : [\varepsilon(\mathbf{w}^{kl}) + e^{kl}] \\ + \left(\varepsilon(D[\mathbf{w}^{kl}]) - \operatorname{sym}(\nabla\mathbf{w}^{kl}\nabla\mathbf{v})\right) : C : [\varepsilon(\mathbf{w}^{ij}) + e^{ij}] \; \mathrm{d}\mathbf{x}.\end{aligned}$$

Finally, because $D[\mathbf{w}^{ij}]$ can be written as a linear combination of the shape functions ϕ, the two terms involving it vanish due to (A.16) (so no adjoint problem is required). Applying the same manipulations as in the previous section, we arrive at the explicit differential form:

$$\boxed{\begin{aligned}&\mathrm{d}C^H_{ijkl}[\lambda_m\delta\mathbf{q}_m] = \delta\mathbf{q}_m \cdot \\ &\qquad\left(\frac{1}{|Y|}\int_\omega \left(\sigma^{ij} : C^{-1} : \sigma^{kl}\right) \nabla\lambda_m - \left[\nabla\lambda_m \cdot \left(\sigma^{kl}\mathbf{w}^{ij}_n + \sigma^{ij}\mathbf{w}^{kl}_n\right)\right] \nabla\varphi_n \; \mathrm{d}\mathbf{x}\right),\end{aligned}} \tag{A.24}$$

where again summation over m and n is implied.

Bibliography

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] G. Allaire. *Shape Optimization by the Homogenization Method*. Number v. 146 in Applied Mathematical Sciences. Springer, 2002.

[3] G. Allaire. *Shape optimization by the homogenization method*, volume 146. Springer, 2002.

[4] G. Allaire and F. Jouve. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 32(11):909 – 918, 2008. Shape and Topological Sensitivity Analysis: Theory and Applications.

[5] G. Allaire, F. Jouve, and H. Maillot. Topology optimization for minimum stress design with the homogenization method. *Structural and Multidisciplinary Optimization*, 28(2-3):87–98, 2004.

[6] P. Alliez, C. Jamin, L. Rineau, S. Tayeb, J. Tournois, and M. Yvinec. 3D mesh generation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.9 edition, 2016.

[7] E. Andreassen, B. S. Lazarov, and O. Sigmund. Design of manufacturable 3D extremal elastic microstructure. *Mechanics of Materials*, 69(1):1–10, 2014.

[8] ASTM. D5023-07 standard test method for plastics: Dynamic mechanical properties: In flexure (three-point bending). In *American Society for Testing and Mate-*

rials, 2007.

[9] M. Avellaneda. Optimal bounds and microgeometries for elastic two-phase composites. *SIAM Journal on Applied Mathematics*, 47(6):1216–1228, 1987.

[10] M. Bächer, B. Bickel, D. L. James, and H. Pfister. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph.*, 31(4):47:1–47:9, July 2012.

[11] J. Barbi**v**c and D. L. James. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Trans. Graph.*, 24(3):982–990, July 2005.

[12] J. Barbi**v**c and D. L. James. Subspace self-collision culling. *ACM Trans. Graph.*, 29(4):81:1–81:9, July 2010.

[13] L. Barthe, B. Wyvill, and E. De Groot. Controllable binary CSG operators for ”soft objects”. *International Journal of Shape Modeling*, 10(02):135–154, 2004.

[14] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202, 1989.

[15] M. P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods and applications.* Springer, 2003.

[16] A. Bernhardt, L. Barthe, M.-P. Cani, and B. Wyvill. Implicit blending revisited. In *Computer Graphics Forum*, volume 29, pages 367–375. Wiley Online Library, 2010.

[17] B. Bickel, M. Bächer, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10, July 2010.

[18] B. Bickel, M. Bächer, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10, July 2010.

[19] B. Bickel, M. Bächer, M. A. Otaduy, W. Matusik, H. Pfister, and M. Gross. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph.*, 28(3):89:1–89:9, July 2009.

[20] J. F. Blinn. A generalization of algebraic surface drawing. *ACM transactions on graphics (TOG)*, 1(3):235–256, 1982.

[21] J. Bloomenthal. Bulge elimination in convolution surfaces. In *Computer Graphics Forum*, volume 16, pages 31–41. Wiley Online Library, 1997.

[22] T. Bückmann, N. Stenger, M. Kadic, J. Kaschke, A. Frölich, T. Kennerknecht, C. Eberl, M. Thiel, and M. Wegener. Tailored 3d mechanical metamaterials made by dip-in direct-laser-writing optical lithography. *Advanced Materials*, 24(20):2710–2714, 2012.

[23] J. E. Cadman, S. Zhou, Y. Chen, and Q. Li. On design of multi-functional microstructural materials. *Journal of Materials Science*, 48(1):51–66, 2013.

[24] J. Calì, D. A. Calian, C. Amati, R. Kleinberger, A. Steed, J. Kautz, and T. Weyrich. 3d-printing of non-assembly, articulated models. *ACM Trans. Graph.*, 31(6):130:1–130:8, Nov. 2012.

[25] D. Chen, D. I. Levin, P. Didyk, P. Sitthi-Amorn, and W. Matusik. Spec2fab: a reducer-tuner model for translating specifications to 3d prints. *ACM Transactions on Graphics (TOG)*, 32(4):135, 2013.

[26] A. Cherkaev. *Variational methods for structural optimization*, volume 140. Springer, 2000.

[27] J. Chu, S. Engelbrecht, G. Graf, and D. W. Rosen. A comparison of synthesis methods for cellular structures with application to additive manufacturing. *Rapid Prototyping Journal*, 16(4):275–283, 2010.

[28] P. Cignoni, N. Pietroni, L. Malomo, and R. Scopigno. Field-aligned mesh joinery. *ACM Trans. Graph.*, 33(1):11:1–11:12, Feb. 2014.

[29] D. Cioranescu and P. Donato. *An introduction to homogenization.* Oxford University Press, 1999.

[30] T. Davis. Algorithm 832: Umfpack v4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, 30(2):196–199, 2004.

[31] Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph.*, 29(4):62:1–62:10, July 2010.

[32] I. Elishakof and M. Ohsaki. *Optimization and anti-optimization of structures under uncertainty.* World Scientific, 2010.

[33] D. Ewins. *Modal testing: theory, practice and application*, volume 2. Research studies press Baldock, 2000.

[34] C. Felippa. Nonlinear finite element methods. 2016.

[35] K. Fidkowski, I. Kroo, K. Willcox, and F. Engelson. Stochastic gust analysis techniques for aircraft conceptual design. 2008. AIAA paper 2008-5848.

[36] A. T. Gaynor and J. K. Guest. Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5):1157–1172, 2016.

[37] O. Gourmel, L. Barthe, M.-P. Cani, B. Wyvill, A. Bernhardt, M. Paulin, and H. Grasberger. A gradient-based implicit blend. *ACM Transactions on Graphics (TOG)*, 32(2):12, 2013.

[38] Y. Grabovsky and R. V. Kohn. Microstructures minimizing the energy of a two phase elastic composite in two space dimensions. II: the Vigdergauz microstructure.

Journal of the Mechanics and Physics of Solids, 43(6):949–972, 1995.

[39] G. N. Greaves, A. L. Greer, R. S. Lakes, and T. Rouxel. Poisson's ratio and modern materials. *Nature Materials*, 10(11):823–837, 2011.

[40] J. K. Guest and J. H. Prévost. Optimizing multifunctional materials: Design of microstructures for maximized stiffness and fluid permeability. *International Journal of Solids and Structures*, 43(2223):7028 – 7047, 2006.

[41] G. W. Hart. Sculptural forms from hyperbolic tessellations. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 155–161. IEEE, 2008.

[42] Z. Hashin and S. Shtrikman. A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids*, 11(2):127–140, 1963.

[43] K. Hauser, C. Shen, and J. O'Brien. Interactive deformation using modal analysis with constraints. In *Graphics Interface*, volume 3, pages 16–17, 2003.

[44] M. Ha**v**san, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph.*, 29:61:1–61:10, July 2010.

[45] K. Hildebrand, B. Bickel, and M. Alexa. Crdbrd: Shape fabrication by sliding planar slices. *Comp. Graph. Forum*, 31(2pt3):583–592, May 2012.

[46] J. Hiller and H. Lipson. Design and analysis of digital materials for physical 3d voxel printing. *Rapid Prototyping Journal*, 15(2):137–149, 2009.

[47] S. J. Hollister. Porous scaffold design for tissue engineering. *Nature Materials*, 4(7):518–524, 2005.

[48] Y. C. Hongyi Xu, Yijing Li and J. Barbi**v**c. Interactive material design using model reduction. *ACM Trans. Graphics*, 34(2), 2015.

[49] S. G. Johnson. The nlopt nonlinear-optimization package, 2016.

[50] A. R. Julian Panetta and D. Zorin**v**c. Worst-case stress relief for microstructures. *ACM Trans. Graph.*, 36(122), July 2017.

[51] H. S. Kang. *Hierarchical design and simulation of tissue engineering scaffold mechanical, mass transport, and degradation properties.* PhD thesis, The University of Michigan, 2010.

[52] L. Kharevych, P. Mullen, H. Owhadi, and M. Desbrun. Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. Graph.*, 28(3):51:1–51:8, July 2009.

[53] R. V. Kohn and A. McKenney. Numerical implementation of a variational method for electrical impedance tomography. *Inverse Problems*, 6(3):389, 1990.

[54] X. Kou and S. Tan. A systematic approach for integrated computer-aided design and finite element analysis of functionally-graded-material objects. *Materials & design*, 28(10):2549–2565, 2007.

[55] D. Kraft. Algorithm 733: Tomp–fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):262–281, 1994.

[56] G. Kreisselmeier and R. Steinhauser. Application of vector performance optimization to a robust control loop design for a fighter aircraft. *International Journal of Control*, 37(2):251–284, 1983.

[57] M. Langelaar. An additive manufacturing filter for topology optimization of print-ready designs. *Structural and Multidisciplinary Optimization*, pages 1–13, 2016.

[58] T. Langlois, A. Shamir, D. Dror, W. Matusik, and D. I. Levin. Stochastic structural analysis for context-aware design and fabrication. *ACM Transactions on Graphics*

(TOG), 35(6):226, 2016.

[59] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. SIAM, 1998.

[60] H. Lian, A. N. Christiansen, D. A. Tortorelli, O. Sigmund, and N. Aage. Combined shape and topology optimization for minimization of maximal von mises stress. *Structural and Multidisciplinary Optimization*, 55(5):1541–1557, 2017.

[61] C.-Y. Lin, C.-C. Hsiao, P.-Q. Chen, and S. J. Hollister. Interbody fusion cage design using integrated global layout and local microstructure topology optimization. *Spine*, 29(16):1747–1754, 2004. PMID: 15303018.

[62] C. Y. Lin, N. Kikuchi, and S. J. Hollister. A novel method for biomaterial scaffold internal architecture design to match bone elastic properties with desired porosity. *Journal of Biomechanics*, 37(5):623–636, 2004.

[63] H. Lin, Y. Xiong, and H. Liao. Semi-structured b-spline for blending two b-spline surfaces. *Computers & Mathematics with Applications*, 68(7):706–718, 2014.

[64] L. Liu, R. D. James, and P. H. Leo. Periodic inclusionatrix microstructures with constant field inclusions. *Metallurgical and Materials Transactions A*, 38(4):781–787, 2007.

[65] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik. Chopper: partitioning models into 3d-printable parts. *ACM Trans. Graph.*, 31(6):129:1–129:9, Nov. 2012.

[66] J. Martínez, J. Dumas, and S. Lefebvre. Procedural Voronoi foams for additive manufacturing. *ACM Transactions on Graphics (TOG)*, 35(4):44, 2016.

[67] K. Mela and J. Koski. Distributed loads in truss topology optimization. In *Proceedings of the 10th world congress on structural and multidisciplinary optimization, Orlando*, 2013.

[68] G. W. Milton. *The theory of composites.* Cambridge University Press, 2002.

[69] V. Mironov, R. P. Visconti, V. Kasyanov, G. Forgacs, C. J. Drake, and R. R. Markwald. Organ printing: tissue spheroids as building blocks. *Biomaterials*, 30(12):2164–2174, 2009.

[70] J. Mitani and H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 259–263, New York, NY, USA, 2004. ACM, ACM.

[71] Y. Mori and T. Igarashi. Plushie: An interactive design system for plush toys. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM, ACM.

[72] A. MOSEK. The mosek optimization tools manual. version 6.0., 2010, 2010.

[73] P. Nakasone and E. Silva. Dynamic design of piezoelectric laminated sensors and actuators using topology optimization. *Journal of Intelligent Material Systems and Structures*, 21(16):1627–1652, 2010.

[74] R. Nickell. Nonlinear dynamics by mode superposition. *Computer Methods in Applied Mechanics and Engineering*, 7(1):107–129, 1976.

[75] J. Panetta, Q. Zhou, L. Malomo, N. Pietroni, P. Cignoni, and D. Zorin. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)*, 34(4):135, 2015.

[76] A. Pasko and V. Adzhiev. Function-based shape modeling: mathematical framework and specialized language. In *International Workshop on Automated Deduction in Geometry*, pages 132–160. Springer, 2002.

[77] G. I. Pasko, A. A. Pasko, and T. L. Kunii. Bounded blending for function-based shape modeling. *IEEE Computer Graphics and Applications*, 25(2):36–45, 2005.

[78] A. Pentland and J. Williams. Good vibrations: modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph.*, 23(3):207–214, July 1989.

[79] M. Polajnar, F. Kosel, and R. Drazumeric. Structural optimization using global stress-deviation objective function via the level-set method. *Structural and Multidisciplinary Optimization*, 55(1):91–104, 2017.

[80] M. Pratt, A. L. Marsan, V. Kumar, and D. Dutta. An assessment of data requirements and data transfer formats for layered manufacturing. Technical report, National Institute of Standards and Technology, 1998.

[81] X. Qian. Undercut and overhang angle control in topology optimization: a density gradient based integral approach. *International Journal for Numerical Methods in Engineering*, 2016.

[82] I. Quilez. Smooth minimum. 2013.

[83] A. Radman, X. Huang, and Y. Xie. Topological optimization for the design of microstructures of isotropic cellular materials. *Engineering Optimization*, 45(11):1331–1348, 2013.

[84] A. P. Rockwood. The displacement method for implicit blending surfaces in solid models. *ACM Transactions on Graphics (TOG)*, 8(4):279–297, 1989.

[85] M. Sabin. The use of potential surfaces for numerical geometry. *British Aircraft Corporation, Weybridge, UK, Technical Report No. VTO/MS/153*, 1968.

[86] C. Schumacher, B. Bickel, J. Rys, S. Marschner, C. Daraio, and M. Gross. Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):136, 2015.

[87] Y. Schwartzburg and M. Pauly. Fabrication-aware design with intersecting planar pieces. *Comput. Graph. Forum*, 32(2):317–326, 2013.

[88] Y. Schwartzburg, R. Testuz, A. Tagliasacchi, and M. Pauly. High-contrast computational caustic design. *ACM Trans. Graph.*, 33(4):74:1–74:11, July 2014.

[89] J. Schwerdtfeger, F. Wein, G. Leugering, R. F. Singer, C. Krner, M. Stingl, and F. Schury. Design of auxetic structures via mathematical optimization. *Advanced Materials*, 23(22):2650–2654, 2011.

[90] Shapeways. Things to keep in mind when designing for 3d printing, 2011.

[91] H. Si. Tetgen: A quality tetrahedral mesh generator and a 3d delaunay triangulator. *Weblink: http://tetgen. berlios. de/(accessed on: March 31, 2012)*, 2007.

[92] H. Si. A quality tetrahedral mesh generator and a 3D Delaunay triangulator. *URL http://tetgen.berlios.de*, 2010.

[93] O. Sigmund. Tailoring materials with prescribed elastic properties. *Mechanics of Materials*, 20(4):351–368, 1995.

[94] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. Gross. Computational design of actuated deformable characters. *ACM Transactions on Graphics (TOG)*, 32(4):82, 2013.

[95] SolidWorks. Solidworks simulation reference, 2011. `http://help.solidworks.com/2011/english/SolidWorks/cworks/LegacyHelp/Simulation/Materials/Material_models/Linear_Elastic_Orthotropic_Model.htm`, 2011.

[96] F. Sonmez. Optimal shape design of shoulder fillets for flat and round bars under various loadings. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(8):1741–1754, 2009.

[97] O. Stava, J. Vanek, B. Benes, N. Carr, and R. M**v**ech. Stress relief: improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11, July 2012.

[98] A. Telea and A. Jalba. Voxel-based assessment of printability of 3d shapes. In *Proceedings of the 10th international conference on Mathematical morphology and its applications to image and signal processing*, ISMM'11, pages 393–404, Berlin, Heidelberg, 2011. Springer-Verlag.

[99] S. Timoshenko and S. Woinowsky-Krieger. Theory of plates and shells. *Engineering Societies Monographs, New York: McGraw-Hill*, 1940.

[100] S. Torquato. *Random heterogeneous materials: microstructure and macroscopic properties*, volume 16. Springer, 2002.

[101] S. Torquato and A. Donev. Minimal surfaces and multifunctionality. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2047):1849–1856, 2004.

[102] S. Torquato, S. Hyun, and A. Donev. Multifunctional composites: optimizing microstructures for simultaneous transport of heat and electricity. *Physical review letters*, 89(26):266601, 2002.

[103] S. Torquato, S. Hyun, and A. Donev. Optimal design of manufacturable three-dimensional composites with multifunctional characteristics. *Journal of Applied Physics*, 94(9):5748–5755, 2003.

[104] N. Umetani, T. Igarashi, and N. J. Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.*, 31(4):86:1–86:11, July 2012.

[105] L. Van Miegroet and P. Duysinx. Stress concentration minimization of 2d filets using x-fem and level set description. *Structural and Multidisciplinary Optimization*, 33(4-5):425–438, 2007.

[106] J. Vida, R. R. Martin, and T. Varady. A survey of blending methods that use parametric surfaces. *Computer-Aided Design*, 26(5):341–365, 1994.

[107] K. Vidimvce, S.-P. Wang, J. Ragan-Kelley, and W. Matusik. Openfab: A programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics (TOG)*, 32(4):136, 2013.

[108] T. Weyrich, P. Peers, W. Matusik, and S. Rusinkiewicz. Fabricating microgeometry for custom surface reflectance. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 28(3):32:1–32:6, 2009.

[109] Q. Xia, T. Shi, S. Liu, and M. Y. Wang. A level set solution to the stress-based structural shape and topology optimization. *Computers & Structures*, 90:55–64, 2012.

[110] Z CORPORATION. Architectural design guide - printing 3d architectural models, 2011.

[111] C. Zanni, A. Bernhardt, M. Quiblier, and M.-P. Cani. Scale-invariant integral surfaces. In *Computer Graphics Forum*, volume 32, pages 219–232. Wiley Online Library, 2013.

[112] C. Zanni, M. Gleicher, and M.-P. Cani. N-ary implicit blends with topology control. *Computers & Graphics*, 46:1–13, 2015.

[113] T. Zeiler. Matched filter concept and maximum gust loads. *Journal of aircraft*, 34(1):101–108, 1997.

[114] H. Zhao, W. Xu, K. Zhou, Y. Yang, X. Jin, and H. Wu. Stress-constrained thickness optimization for shell object fabrication. In *Computer Graphics Forum*. Wiley Online Library, 2016.

[115] Q. Zhou, J. Panetta, and D. Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4):137:1–137:12, July 2013.

[116] Y. Zhou, E. Kalogerakis, R. Wang, and I. R. Grosse. Direct shape optimization for strengthening 3d printable objects. In *Computer Graphics Forum*, volume 35,

pages 333–342. Wiley Online Library, 2016.