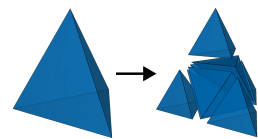


Supplemental Material for Computational Design of Flexible Planar Microstructures

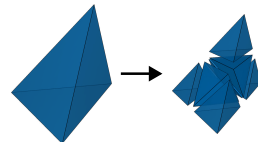
Zhan Zhang, Christopher Brandt, Jean Jouve, Yue Wang, Tian Chen, Mark Pauly, and Julian Panetta

S1 Meshing for C^1 Hermite Tetrahedral Interpolation

Generating tetrahedral meshes that support C^1 trivariate Powell-Sabin interpolation is nontrivial due to strict geometric requirements on the element shapes. Worsley and Piper [1988] prove that C^1 continuity is achieved by placing the scheme’s split points at the circumcenters of each mesh simplex. This construction works provided that those points (*i.e.*, the edge midpoints, face circumcenters, and tetrahedron circumcenter) lie inside their originating simplex. Our need to refine the interpolation mesh requires this condition to further hold for all levels of $1 \rightarrow 8$ tet subdivision (see inset). Satisfying all of these constraints is a challenging meshing problem that, to our knowledge, is not addressed by existing algorithms. For instance, the quality metrics typically maximized by meshing algorithms promote regular tetrahedra like the one at the left of the inset. While a regular tetrahedron is ideal for the coarsest level of interpolation, refining it (right) generates four inner tetrahedra with circumcenters lying on their long edges, not in the interior.



The rhombic tetrahedron shown in the inset below is acceptable (its circumcenters lie within, and Liu and Joe [1996] proved that all 8 tetrahedra generated by subdividing one of these tetrahedra are similar to the original); in fact we use it to construct a feasible initial mesh circumscribing the domain (Section S1.4). However, it is not possible to create a mesh tightly adhering to the domain solely from elements similar to this rhombic tetrahedron.



S1.1 Variational Meshing Approach

We therefore propose a variational meshing algorithm that generates mesh \mathcal{T} by minimizing a shape approximation objective J_{shape} while enforcing that all circumcenters lie inside using a stiff barrier term $J_{\text{barrier}}(\mathcal{T})$:

$$\min_{\mathcal{T}} w_s J_{\text{shape}}(\mathcal{T}) + J_{\text{barrier}}(\mathcal{T}) + J_{\text{StVK}}(\mathcal{T}),$$

where $J_{\text{StVK}}(\mathcal{T})$ is a regularization term measuring the elastic energy needed to deform the initial mesh into the current one using an isotropic St. Venant Kirchhoff (StVK) material model with Young’s modulus 1 and Poisson’s ratio 0. The minimization is performed over the vertex positions of \mathcal{T} using Newton’s method.

This StVK energy is chosen since it permits elements to degenerate to zero volume (unlike a neo-Hookean energy), which we use to implement a simple edge collapse strategy. After each Newton iteration, we check if any edge length has shrunk below a threshold (10^{-2}). If so, we terminate the Newton solver, run edge collapses until all edges are above 10^{-1} in length, and restart the mesh optimization on the collapsed mesh. Two of these rounds of edge collapses were executed for the mesh we used in our experiments (the first collapsing 5 edges, and the second just one edge).

We found that constraining the subdivided tetrahedra’s circumcenters prevented edge collapses, so we actually run our meshing algorithm two phases: (i) optimize with barriers only for the top-level “macro tet” circumcenters, applying the edge collapse strategy described above; then (ii) optimize with barriers for both “macro” and subdivided tet circumcenters. In phase (i) we use $w_s = 10^2$, while in phase (ii) we use $w_s = 10^5$ to promote tight adherence to the input shape. We found it sufficient to include barriers for only one level of subdivision in our optimization (Section S1.6).

S1.2 Shape Approximation

The shapes we need to mesh are very specific: ellipsoids in strain space. In fact, as discussed in Section 4.4 of the main paper, it is advantageous to mesh and perform interpolation in a transformed space—the (approximate) “Mandel” stress space—where the domain is always spherical. Therefore, we use a simplified shape approximation objective in our meshing optimization that fits the boundary to a unit sphere:

$$J_{\text{shape}}(\mathcal{T}) = \frac{1}{4} \int_{\partial\mathcal{T}} (\|\mathbf{x}\|^2 - 1)^2 \, dA,$$

and then after optimization, we scale \mathcal{T} to tightly *circumscribe* the desired sphere (Section S1.5). We use exact Gauss quadrature to evaluate this integral. For general domains, a more traditional distance based on closest point projections could be substituted for J_{shape} (*e.g.*, Mitra et al. [2004]).

S1.3 Circumcenter Barrier Term

We use barrier terms to enforce the constraint that every tetrahedron has an internal circumcenter. It is a straightforward extension to add barriers enforcing internal circumcenters for each element face, but in our experiments these additional constraints were automatically satisfied without the additional barriers (Section S1.6). Note that the edge circumcenters are midpoints, which are always interior.

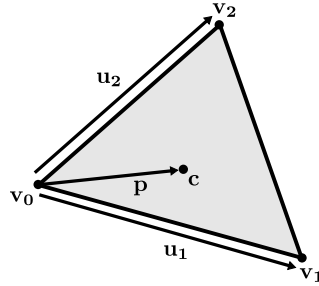
Our constraint on circumcenters is easiest to pose in terms their barycentric coordinates: the circumcenter of a K -simplex e is interior if and only if its coordinates satisfy $\lambda_i^e > 0, 0 \leq i \leq K$. We therefore formulate the barrier term as:

$$J_{\text{barrier}}(\mathcal{T}) = \sum_{e \in \mathcal{T}} \sum_{i=0}^K b(\lambda_i^e), \quad b(\lambda) := \begin{cases} 0 & \text{if } \lambda > \lambda_{\text{activate}}, \\ -\log\left(\frac{\lambda - \lambda_{\min}}{\lambda_{\text{activate}} - \lambda_{\min}}\right)^3 & \text{otherwise.} \end{cases} \quad (\text{S1})$$

This barrier function smoothly ramps up from 0 (when $\lambda = \lambda_{\text{activate}}$) to ∞ as λ decreases towards λ_{\min} .

S1.3.1 Circumcenter Calculation

This section derives a simple generic formula for the barycentric coordinates of circumcenters of K -simplices embedded in dimension $D \geq K$. As an illustrative example, we consider the case of a triangle:



The circumcenter is the point \mathbf{c} with equal distance r to each of the vertices:

$$r^2 = \|\mathbf{p}\|^2 = \|\mathbf{u}_1 - \mathbf{p}\|^2 = \|\mathbf{u}_2 - \mathbf{p}\|^2.$$

Expanding the squared norms obtains equations for \mathbf{p} in terms of its dot product with edge vectors \mathbf{u}_i :

$$\begin{aligned} \|\mathbf{p}\|^2 &= \|\mathbf{u}_1\|^2 - 2\mathbf{u}_1 \cdot \mathbf{p} + \|\mathbf{p}\|^2 = \|\mathbf{u}_2\|^2 - 2\mathbf{u}_2 \cdot \mathbf{p} + \|\mathbf{p}\|^2 \\ \implies [\mathbf{u}_1 \quad \mathbf{u}_2]^\top \mathbf{p} &= \frac{1}{2} \begin{bmatrix} \|\mathbf{u}_1\|^2 \\ \|\mathbf{u}_2\|^2 \end{bmatrix}. \end{aligned}$$

Re-expressing \mathbf{p} as using barycentric coordinates λ_1 and λ_2 as $\lambda_1 \mathbf{u}_1 + \lambda_2 \mathbf{u}_2$:

$$\implies \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{bmatrix}^\top \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \|\mathbf{u}_1\|^2 \\ \|\mathbf{u}_2\|^2 \end{bmatrix}.$$

This process clearly generalizes to arbitrary K and $D > K$: we form the square $K \times K$ matrix

$$A = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_K \end{bmatrix}^\top \underbrace{\begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_K \end{bmatrix}}_U = U^\top U$$

and solve the linear system:

$$A\boldsymbol{\lambda} = \frac{1}{2} \text{diag}(A), \quad \boldsymbol{\lambda} := [\lambda_1 \quad \cdots \quad \lambda_K]^\top \quad (\text{S2})$$

to obtain the circumcenter's barycentric coordinates with respect to vertices $\{\mathbf{v}_1, \dots, \mathbf{v}_K\}$.

S1.3.2 Circumcenter Sensitivity Analysis

Now, we determine formulas for the first and second derivatives of the circumcenter position with respect to the vertex positions. These are obtained by differentiating both sides of the linear system (S2). Instead of differentiating directly with respect to the vertex positions, we first differentiate with respect to \mathbf{u}_i and then apply a trivial chain rule. We obtain the derivatives $\frac{\partial \lambda_k}{\partial \mathbf{u}_i}$ as *row vectors* of the linear system:

$$\begin{aligned} U^\top U \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{u}_i} &= \mathbf{e}_i \otimes \mathbf{u}_i - \left(((\mathbf{e}_i \otimes \mathbf{e}_n)U + U^\top (\mathbf{e}_n \otimes \mathbf{e}_i)) \boldsymbol{\lambda} \right) \otimes \mathbf{e}_n \\ &= \mathbf{e}_i \otimes (\mathbf{u}_i - U\boldsymbol{\lambda}) - U^\top \lambda_i. \end{aligned} \quad (\text{S3})$$

Suppose we now wish to differentiate a generic scalar objective function $\tilde{J}(\boldsymbol{\lambda}) := J(\lambda_0, \dots, \lambda_K)$ of the “independent” barycentric coordinates $\boldsymbol{\lambda} \in \mathbb{R}^K$, where $\lambda_0 = 1 - \text{sum}(\boldsymbol{\lambda})$:

$$\frac{\partial \tilde{J}(\boldsymbol{\lambda}(U))}{\partial \mathbf{u}_i} = \frac{\partial \tilde{J}}{\partial \boldsymbol{\lambda}} \cdot \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{u}_i} = \mathbf{s} \cdot (\mathbf{e}_i \otimes (\mathbf{u}_i - U\boldsymbol{\lambda}) - U^\top \lambda_i) = (s_i (\mathbf{u}_i - U\boldsymbol{\lambda}) - \lambda_i U \mathbf{s})^\top = \left(s_i \mathbf{u}_i - U(s_i \boldsymbol{\lambda} + \lambda_i \mathbf{s}) \right)^\top,$$

where \mathbf{s} solves the adjoint equation $U^\top U \mathbf{s} = \frac{\partial \tilde{J}}{\partial \boldsymbol{\lambda}}$. Note that $\left[\frac{\partial \tilde{J}}{\partial \boldsymbol{\lambda}} \right]_i = \frac{\partial J}{\partial \lambda_i} - \frac{\partial J}{\partial \lambda_0}$. For our meshing problem, we plug in $J(\lambda_0, \dots, \lambda_K) = \sum_{i=0}^K b(\lambda_i)$, and the required derivatives $\frac{\partial J}{\partial \lambda_i}$ can be obtained by directly differentiating the formulas in (S1).

Directional second derivatives $\delta \left(\frac{\partial \tilde{J}(\boldsymbol{\lambda}(U))}{\partial \mathbf{u}_i} \right)^\top = \frac{\partial^2 \tilde{J}(\boldsymbol{\lambda}(U))}{\partial \mathbf{u}_i \partial \mathbf{u}_j} \delta \mathbf{u}_j$ can be obtained by (i) using (S3) to solve for $\delta \boldsymbol{\lambda}$; (ii) solving

$$U^\top U \delta \mathbf{s} = \frac{\partial^2 \tilde{J}}{\partial \boldsymbol{\lambda}^2} \delta \boldsymbol{\lambda} - \delta U^\top U \mathbf{s} - U^\top \delta U \mathbf{s}$$

for the adjoint state perturbation $\delta \mathbf{s}$; and finally (iii) evaluating

$$\delta \left(\frac{\partial \tilde{J}(\boldsymbol{\lambda}(U))}{\partial \mathbf{u}_i} \right)^\top = (\delta s_i) \mathbf{u}_i + s_i \delta \mathbf{u}_i - \delta U (s_i \boldsymbol{\lambda} + \lambda_i \mathbf{s}) - U((\delta s_i) \boldsymbol{\lambda} + s_i \delta \boldsymbol{\lambda} + (\delta \lambda_i) \mathbf{s} + \lambda_i \delta \mathbf{s}).$$

By plugging in each standard basis vector for the edge vector perturbation $\delta \mathbf{u}_j$, columns of the Hessian block $\frac{\partial^2 J(\boldsymbol{\lambda}(U))}{\partial \mathbf{u}_i \partial \mathbf{u}_j}$ can be evaluated one at a time. Again, the required derivatives $\frac{\partial^2 J}{\partial \lambda_i \partial \lambda_j}$ can be obtained by directly differentiating the formulas in (S1).

S1.4 Initialization

As mentioned above, we use the rhombic tetrahedron from Liu and Joe [1996] to construct our initial mesh due its self-similar subdivision property. For a general domain \mathcal{D} we could construct such an initialization by scaling up the rhombic tetrahedron to contain \mathcal{D} and then subdividing it to a desired resolution (right), discarding tets that fall outside. This is essentially the approach that we use for meshing the sphere, except that we begin with the more spherical shape obtained by arranging four rhombic tetrahedra into an octahedron as shown in Figure S1.

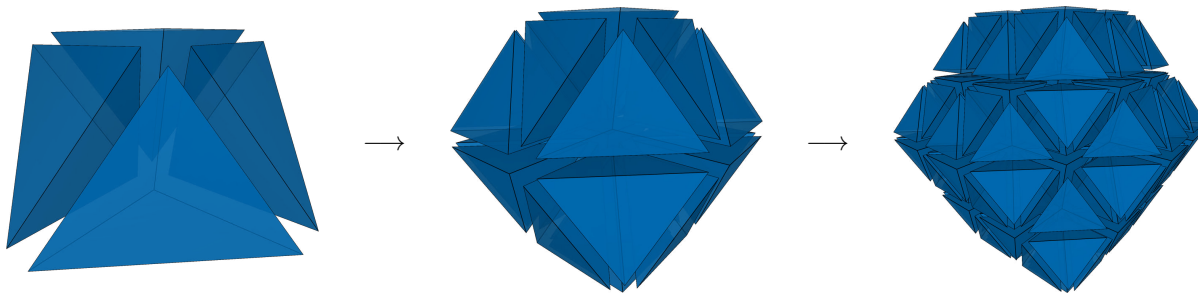


Figure S1 *Initial mesh construction: we begin by arranging four rhombic tetrahedra into a more symmetrical octahedron (left). Then we recursively subdivide these tetrahedra to obtain a mesh of the desired resolution, trimming the subtetrahedra that fall outside the domain.*

S1.5 Make it Circumscribe

To avoid extrapolation when sampling the strain domain, we need our mesh \mathcal{T} to fully contain the Mandel stress sphere of given radius r , which is not guaranteed by our shape approximation term J_{shape} . We address this using a postprocessing step that scales the mesh: we compute the distance d from the origin to the closest point on $\partial\mathcal{T}$ and then scale the mesh by r/d .

S1.6 Results

The result of our algorithm is a mesh that tightly circumscribes the spherical interpolation domain while satisfying the constraint $\lambda_i^e > \lambda_{\min}$ for all simplices e at all levels of subdivision:

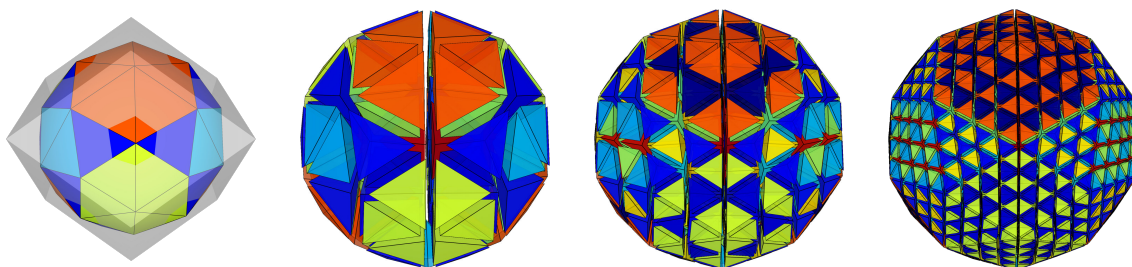


Figure S2 *The optimized mesh generated from the initial mesh at the right of Figure S1, with tetrahedra colored by their circumcenters' minimum barycentric coordinates. On the left, we compare the mesh with the boundary of the initial mesh to show its improved shape approximation. Then to the right we show exploded views revealing interior tetrahedra for the optimized mesh after 0, 1, and 2 levels of refinement. Importantly, no new colors appear after one level of subdivision.*

This optimized mesh encloses roughly 20% less volume than the initial mesh, and its furthest point from the origin roughly 25% closer. It also has 6 fewer vertices (59 vs 65).

We finally confirm that the circumcenters are all interior with the following histograms, and note via an ablation study the necessity of circumcenter barriers:

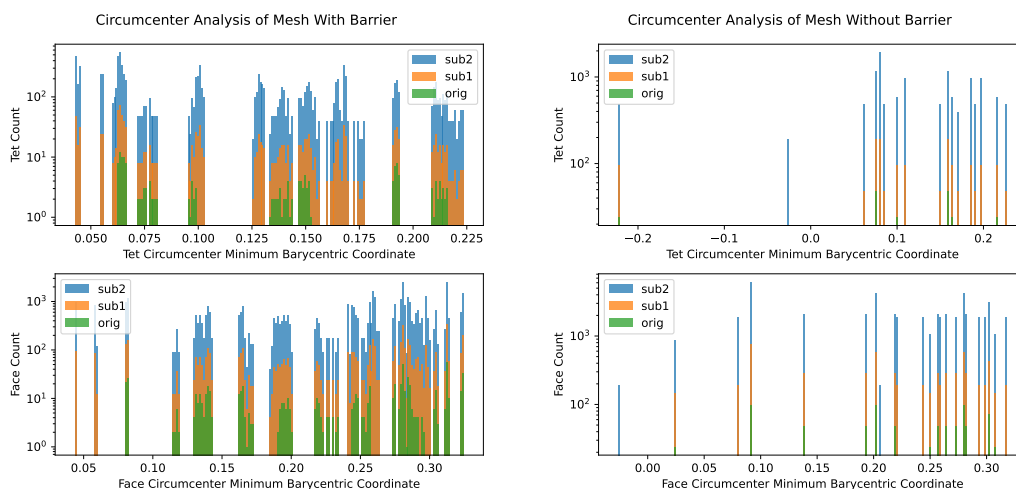


Figure S3 Left: histogram of the minimum barycentric coordinates of our optimized mesh's tet and face circumcenters at various levels of refinement. We note that the distinct values stop changing after one level of subdivision (as further subdivision generates similar tetrahedra for this mesh). Right: the same statistics evaluated for a mesh produced without the circumcenter barriers. We observe several tetrahedra and faces with exterior circumcenters.

S2 Exploiting Reflectional Symmetry (Orthotropic Homogenization)

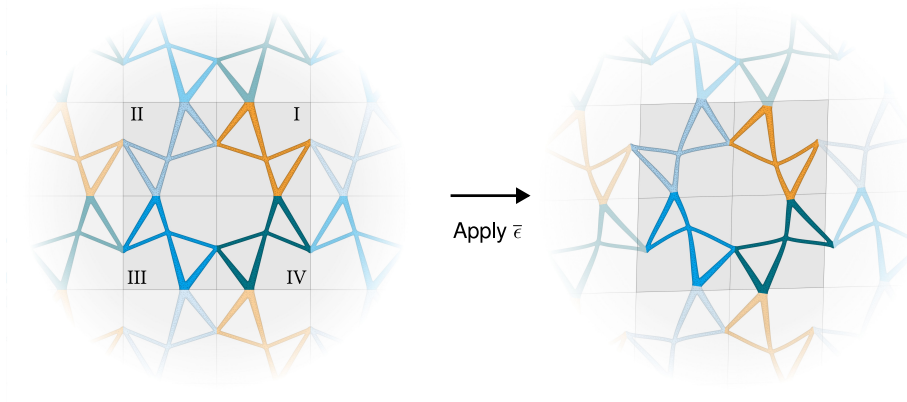


Figure S4 We consider microstructures whose geometry is defined by reflecting a base cell (quadrant I, orange) across the coordinate planes.

In this section, we explain how for a microstructure geometry Ω with reflectional symmetry across the coordinate planes and an axis-aligned orthotropic fabrication material, periodic homogenization can be posed over Ω_I , the portion of Ω occupying a smaller symmetry base cell (the positive quadrant in 2D or octant in 3D). For nonlinear elasticity, we must solve for a *multi-valued field* over Ω_I , and yet we still benefit from a

2 \times reduction in the size of the associated optimization problem. Our reduction of course assumes that the microstructure does not experience a symmetry-breaking buckling mode. Finally, we show in Section S2.3 that for linear elasticity, probing with the canonical basis for strain tensors achieves a decoupling that enables solving for a *single-valued field* (but with different boundary conditions for the shearing probes); that approach has been exploited in past work, *e.g.* Panetta et al. [2015, 2017].

The periodic homogenization problem for an elastic metamaterial is to evaluate:

$$\bar{\psi}(\bar{F}) := \min_{\omega \text{ periodic}} E[\omega, \bar{F}], \quad E[\omega, \bar{F}] := \int_{\Omega} \psi(\nabla\omega + \bar{F}) \, d\mathbf{X} \quad (\text{S4})$$

We begin by considering symmetries of the fabrication material's constitutive law. Any hyperelastic material is invariant to post-rotations:

$$\psi(RF) = \psi(F) \quad \forall R \in SO(n).$$

Isotropic hyperelastic materials additionally satisfy:

$$\psi(FR^{\top}) = \psi(F) \quad \forall R \in SO(n)$$

for all rotations R of the material axes, while *orthotropic* hyperelastic materials with material axes aligned with the coordinate system satisfy:

$$\psi(RFR^{\top}) = \psi(RFR) = \psi(F) \quad (\text{S5})$$

for all *reflections* R across the coordinate planes.

S2.1 Pulling back to Ω_I

Exploiting the reflectional symmetry of the microstructure geometry, we can pull the homogenization problem back to the portion of Ω occupying the symmetry base cell, which we denote as Ω_I . In 2D:

$$E[\omega, \bar{F}] = \int_{\Omega_I} \psi(\nabla\omega_I + \bar{F}) + \psi(R_{II}\nabla\omega_{II}R_{II} + \bar{F}) + \psi(R_{III}\nabla\omega_{III}R_{III} + \bar{F}) + \psi(R_{IV}\nabla\omega_{IV}R_{IV} + \bar{F}) \, d\mathbf{X},$$

where $\omega_I, \omega_{II}, \omega_{III}, \omega_{IV}$ denote the restriction of ω to the respective quadrants *followed by reflection into quadrant I*, and R_{II}, R_{III}, R_{IV} are the associated reflection operators. Instead of minimizing over ω , we could minimize over fields $\omega_I, \omega_{II}, \omega_{III}, \omega_{IV}$ with appropriate conditions bonding the fields together at the quadrant boundaries. However the resulting minimization problem is no smaller than the original, and only limited computational savings have been achieved by representing and operating on Ω_I alone.

The real computational savings come from eliminating half of these subfields, exploiting orthotropic symmetry of the fabrication material:

$$\begin{aligned} \psi(R_{II}\nabla\omega_{II}R_{II} + \bar{F}) &= \psi(\nabla\omega_{II} + R_{II}\bar{F}R_{II}) = \psi(\nabla\omega_{II} + \tilde{F}), \\ \psi(R_{III}\nabla\omega_{III}R_{III} + \bar{F}) &= \psi(\nabla\omega_{III} + R_{III}\bar{F}R_{III}) = \psi(\nabla\omega_{III} + \bar{F}), \\ \psi(R_{IV}\nabla\omega_{IV}R_{IV} + \bar{F}) &= \psi(\nabla\omega_{IV} + R_{IV}\bar{F}R_{IV}) = \psi(\nabla\omega_{IV} + \tilde{F}). \end{aligned}$$

In particular, we note that $R_{III}\bar{F}R_{III} = \bar{F}$ and $R_{IV}\bar{F}R_{IV} = R_{II}\bar{F}R_{II} = \tilde{F}$, where \tilde{F} equals \bar{F} with its two off-diagonal entries negated. An analogous simplification happens in 3D, where a total of 4 distinct macroscopic deformation gradients appear (\bar{F} and its copies with a single off-diagonal pair negated.)

In the absence of a symmetry-breaking buckling mode, we therefore conclude $\omega_{III} \equiv \omega_I$ and $\omega_{IV} \equiv \omega_{II}$, since each pair appears in identical energy terms. This observation halves the size of the optimization problem (S4), allowing us to optimize only over ω_I and ω_{II} (or any other pair of representative subfields):

$$\bar{\psi}(\bar{F}) = \min_{\omega_I, \omega_{II}} E[\omega_I, \omega_{II}, \bar{F}], \quad E[\omega_I, \omega_{II}, \bar{F}] = 2 \int_{\Omega_I} \psi(\nabla\omega_I + \bar{F}) + \psi(\nabla\omega_{II} + \tilde{F}) \, d\mathbf{X}. \quad (\text{S6})$$

S2.2 Boundary Conditions

$R_{III}\omega_I$ $[-\omega_x^I, -\omega_y^I]$	$R_{IV}\omega_{II}$ $[\omega_x^{II}, -\omega_y^{II}]$	$R_{III}\omega_I$ $[-\omega_x^I, -\omega_y^I]$	$\omega_x^I = \omega_x^{II}, \omega_y^I = -\omega_y^{II}$
$R_{II}\omega_{II}$ $[-\omega_x^I, \omega_y^{II}]$	ω_I $[\omega_x^I, \omega_y^I]$	$R_{II}\omega_{II}$ $[-\omega_x^I, \omega_y^{II}]$	$\omega_x^I = -\omega_x^{II}, \omega_y^I = \omega_y^{II}$ $\omega_x^I = \omega_x^{II}, \omega_y^I = -\omega_y^{II}$
$R_{III}\omega_I$ $[-\omega_x^I, -\omega_y^I]$	$R_{IV}\omega_{II}$ $[\omega_x^{II}, -\omega_y^{II}]$	$R_{III}\omega_I$ $[-\omega_x^I, -\omega_y^I]$	$\omega_x^I = -\omega_x^{II}, \omega_y^I = \omega_y^{II}$

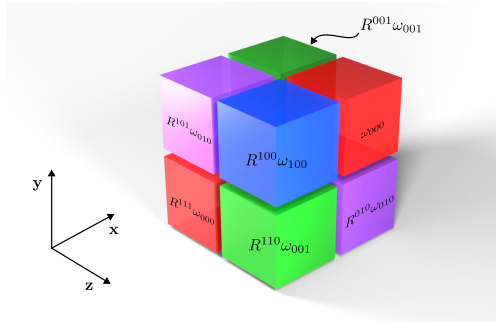
All that remains is to pose the appropriate boundary conditions for the reduced problem (S6). These are determined by enforcing continuity of the full displacement field ω (defined by the subfields over Ω_I) across the base cell boundaries. The conditions for 2D problems are demonstrated in the figure above. We note that each boundary face couples a boundary node displacement from one field with a reflected displacement from another field (where the reflection occurs across the face). These conditions also enforce periodicity of ω as a side-effect. Finally, we note that at the cell corner node, both $\omega_x^I = \omega_x^{II}$ and $\omega_x^I = -\omega_x^{II}$ (similarly for y), forcing $\omega = 0$.

S2.3 Linear Elasticity

Under linear elasticity, the homogenization problem can be solved for any (symmetric) applied macroscopic deformation gradient \bar{F} by forming a linear combination of the microstructure's response to a set of *basis strain probes* $\bar{\varepsilon}^{kl} := \frac{1}{2}(\mathbf{e}_k \otimes \mathbf{e}_l + \mathbf{e}_l \otimes \mathbf{e}_k)$. In other words, for 2D problems we need only solve (S6) for the 3 probes with superscripts $1 \leq k \leq l \leq 2$ (there are 6 in 3D). Furthermore, the diagonal “uniaxial stretch probe” strains $\bar{\varepsilon}^{kk}$ are invariant under axis-aligned reflections, meaning that for these $\omega_I \equiv \omega_{II}$, reducing the unknowns to a single-valued displacement field ω_I . The “shear probe” matrices like $\bar{\varepsilon}^{01}$ *negate* under the corresponding reflection, meaning that $\omega_I \equiv -\omega_{II}$, while they are unchanged by the other reflections; again the cell problem has been reduced to a single unknown vector field.

S2.4 3D Case

In 3D, we have four distinct subfields after accounting for symmetry, as visualized in the following figure:



This figure adopts a systematic numbering convention for the octants and subfields, using a 3-bit subscript that denotes the reflections taking the symmetry base cell, positive octant 000, to the octant in question. For instance, the fluctuation displacement in octant 100 reached by a reflection across the $x = 0$ plane is $R^{100}\omega_{100}$. Note that reflecting across all three coordinate planes effectively multiplies by -1 and preserves

$R^{111}\bar{F}R^{111} = (-1)\bar{F}(-1) = \bar{F}$, consequently identifying the associated subfields. This reflection corresponds to taking the *complement* of the subscript, so it follows that the four distinct subfields are $\omega_{111} \equiv \omega_{000}$, $\omega_{110} \equiv \omega_{001}$, $\omega_{101} \equiv \omega_{010}$, and $\omega_{011} \equiv \omega_{100}$.

We obtain the boundary conditions coupling these subfields by enforcing continuity of the associated ω field across octant boundaries. Different conditions arise from each of the $\binom{4}{2} = 6$ pairings of adjacent colors. From red octant 000, we find:

$$\begin{aligned} [\omega_x^{000}, \omega_y^{000}, \omega_z^{000}] &= [-\omega_x^{100}, \omega_y^{100}, \omega_z^{100}] \quad \text{on “}x\text{ faces” (red-blue)} \\ [\omega_x^{000}, \omega_y^{000}, \omega_z^{000}] &= [\omega_x^{010}, -\omega_y^{010}, \omega_z^{010}] \quad \text{on “}y\text{ faces” (red-purple)} \\ [\omega_x^{000}, \omega_y^{000}, \omega_z^{000}] &= [\omega_x^{001}, \omega_y^{001}, -\omega_z^{001}] \quad \text{on “}z\text{ faces” (red-green),} \end{aligned}$$

from blue octant 100:

$$\begin{aligned} [-\omega_x^{100}, \omega_y^{100}, \omega_z^{100}] &= [-\omega_x^{001}, -\omega_y^{001}, \omega_z^{001}] \quad \text{on “}y\text{ faces” (blue-green)} \\ [-\omega_x^{100}, \omega_y^{100}, \omega_z^{100}] &= [-\omega_x^{010}, \omega_y^{010}, -\omega_z^{010}] \quad \text{on “}z\text{ faces” (blue-purple),} \end{aligned}$$

and from green octant 001:

$$[\omega_x^{001}, \omega_y^{001}, -\omega_z^{001}] = [-\omega_x^{010}, \omega_y^{010}, -\omega_z^{010}] \quad \text{on “}x\text{ faces” (green-purple).}$$

S2.5 Reduced Variable Assignment

We enforce the continuity constraints derived above by elimination, forming a reduced set of vector-valued variables. We describe this process for both the $N = 2$ and $N = 3$ -dimensional cases.

Before presenting our algorithm in the next paragraphs, we first analyze the number of variables that it should end up assigning to each node. After identifying fields with complementary subscripts, each node is associated with 2^{N-1} fluctuation displacement vectors. Constraints are then applied by interfaces between reflected copies of the symmetry base cell. A boundary face of the symmetry base cell reflects into 2^{N-2} such interfaces in the period cell, each applying linearly independent constraints. Nodes in the interior of the orthotropic base cell experience no such constraints and retain this full set of variables (2 vectors in 2D, 4 in 3D). Nodes on a *single* boundary face of the symmetry base cell are left with $2^{N-1} - 2^{N-2} = 2^{N-2}$ variables after elimination (1 in 2D, 2 in 3D). Nodes on *two* boundary faces (a corner in 2D and edge in 3D) lose all their variables in 2D ($\omega = 0$), but actually retain a variable in 3D because the constraints turn out to be not all linearly independent. Nodes on *three* boundary faces (a corner in 3D) lose all variables ($\omega = 0$).

We assign reduced variables using a simple graph traversal algorithm that is executed separately for each finite element node n in Ω_I . For this node, we construct an undirected graph consisting of 2^N vertices, each corresponding to one of the 2^N subfields' values at the node. Constraints are encoded using edges (u, v) connecting graph vertices u and v . Each edge is annotated with a reflection matrix $R_{(u,v)}$ (which can be represented by its diagonal entries $\mathbf{r} \in \{\pm 1\}^N$) and indicates that the subfield values corresponding to u and v are related by reflection $R_{(u,v)}$. If enforcing orthotropic symmetry, an identity-transformation edge is created between each vertex and its complement; omitting these edges results in an optimization problem equivalent to the standard periodic homogenization formulation. Then additional edges are generated to connect vertex pairs (u, v) if their associated values are related by a reflection across a symmetry base cell boundary face *that contains node n* ; this reflection is recorded as $R_{(u,v)}$. Our algorithm assigns a single vector-valued variable to each *consistent* connected component of this graph. By consistent, we mean that the composition of reflections around any cycle is the identity. Inconsistent connected components are assigned no variable, meaning the node's fluctuation displacements in those subfields are all zero.

The algorithm works by iterating over each unvisited graph vertex and initiating a breadth-first search from it. This search labels each vertex v with the accumulated reflection R_v that relates it to the root vertex of the search (whose label is initialized to the identity matrix). Whenever a vertex v is encountered via an edge (u, v) , we compute an accumulated reflection $\tilde{R}_v = R_{(u,v)}R_u$. If v was unvisited, we record $R_v = \tilde{R}_v$. Otherwise, we compare R_v and \tilde{R}_v , flagging the current component as inconsistent if they disagree. When

the search terminates, if the component is consistent, we create one new variable for the root vertex; recorded reflections R_v are later used by the simulation to express the other subfield values in terms of this variable.

S3 Shape Derivatives

We show here how to differentiate the homogenized energy and stress tensors with respect to the rest vertex positions of the finite element mesh. The quantities we wish to differentiate are defined by:

$$\bar{\psi}(\bar{F}) := \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \omega^* + \bar{F}) \, d\mathbf{X}, \quad (\text{S7})$$

$$\bar{\sigma}(\bar{F}) = \frac{1}{|Y|} \int_{\Omega} \psi'(\nabla \omega^*(\mathbf{X}; \bar{F}) + \bar{F}) \, d\mathbf{X}. \quad (\text{S8})$$

We use a standard result from shape optimization to find:

$$\left\langle \frac{\partial \bar{\psi}(\bar{F})}{\partial \Omega}, \mathbf{v} \right\rangle = \int_{\Omega} \psi(\nabla \omega^* + \bar{F})(\nabla \cdot \mathbf{v}) - \psi' : \nabla \omega^* \nabla \mathbf{v} + \psi'(\nabla \omega^* + \bar{F}) : \nabla \left\langle \frac{\partial \omega^*}{\partial \Omega}, \mathbf{v} \right\rangle \, d\mathbf{X},$$

where the first term in the integrand accounts for dilation of the integration domain induced by the advection velocity, the second term accounts for transformation of the differentiation operator ∇ , and the third term accounts for changes in the equilibrium displacements ω^* . Because $\left\langle \frac{\partial \omega^*}{\partial \Omega}, \mathbf{v} \right\rangle$ is an admissible perturbation of ω^* , the final integrand must integrate to zero by the optimality conditions satisfied by ω^* , leaving:

$$\left\langle \frac{\partial \bar{\psi}(\bar{F})}{\partial \Omega}, \mathbf{v} \right\rangle = \int_{\Omega} \psi(\nabla \omega^* + \bar{F})(\nabla \cdot \mathbf{v}) - \psi' : \nabla \omega^* \nabla \mathbf{v} \, d\mathbf{X}.$$

Finally, we note that $\nabla \cdot \mathbf{v} = I : \nabla \mathbf{v}$ and $\psi' : \nabla \omega^* \nabla \mathbf{v} = (\nabla \omega^*)^\top \psi' : \nabla \mathbf{v}$, allowing us to write:

$$\left\langle \frac{\partial \bar{\psi}(\bar{F})}{\partial \Omega}, \mathbf{v} \right\rangle = \int_{\Omega} (\psi(\nabla \omega^* + \bar{F})I - (\nabla \omega^*)^\top \psi') : \nabla \mathbf{v} \, d\mathbf{X} = \int_{\Omega} G_{\bar{\psi}} : \nabla \mathbf{v} \, d\mathbf{X}, \quad (\text{S9})$$

with:

$$G_{\bar{\psi}} := \psi(\nabla \omega^* + \bar{F})I - (\nabla \omega^*)^\top \psi'. \quad (\text{S10})$$

Obtaining an efficient formula for the shape derivative of component $\bar{\sigma}_{ij}$ of $\bar{\psi}'$ is easiest to do by differentiating (S9) with respect to \bar{F} rather than shape-differentiating the macroscopic stress (S8). In fact, we can just differentiate (S10) to obtain:

$$\left\langle \frac{\partial \bar{\sigma}_{ij}(\bar{F})}{\partial \Omega}, \mathbf{v} \right\rangle = \int_{\Omega} G_{\bar{\sigma}_{ij}} : \nabla \mathbf{v} \, d\mathbf{X}, \quad G_{\bar{\sigma}_{ij}} := \frac{\partial G_{\bar{\psi}}}{\partial \bar{F}} : e^{ij},$$

$$G_{\bar{\sigma}_{ij}} = ((\nabla \omega^{ij} + e^{ij}) : \psi')I - (\nabla \omega^{ij})^\top \psi' - (\nabla \omega^*)^\top \psi'' : (\nabla \omega^{ij} + e^{ij}).$$

Exact discrete derivatives with respect to the rest vertex positions of straight-edge subparametric finite elements can be obtained by substituting the velocity field $\mathbf{v} = \delta \mathbf{X}_k \lambda_k$, where λ_k is the piecewise linear shape function¹ for vertex k , and $\delta \mathbf{X}_k$ represents a perturbation of that vertex. For example:

$$\frac{\partial \bar{\psi}}{\partial \mathbf{X}_k} \delta \mathbf{X}_k = \int_{\Omega} G_{\bar{\psi}} : \nabla(\delta \mathbf{X}_k \lambda_k) \, d\mathbf{X} = \int_{\Omega} G_{\bar{\psi}} : (\delta \mathbf{X}_k \otimes \nabla \lambda_k) \, d\mathbf{X} = \delta \mathbf{X}_k \cdot \int_{\Omega} G_{\bar{\psi}} \nabla \lambda_k \, d\mathbf{X},$$

leading us to conclude the gradient formula:

$$\left(\frac{\partial \bar{\psi}}{\partial \mathbf{X}_k} \right)^\top = \int_{\Omega} G_{\bar{\psi}} \nabla \lambda_k \, d\mathbf{X}.$$

¹Note that this usage of λ_k is distinct from the circumcenter coordinates in Section S1 but agrees with the main paper!

S4 Optimization Statistics

We present statistics for two characteristic runs of our nonlinear least squares design optimization in the following figure:

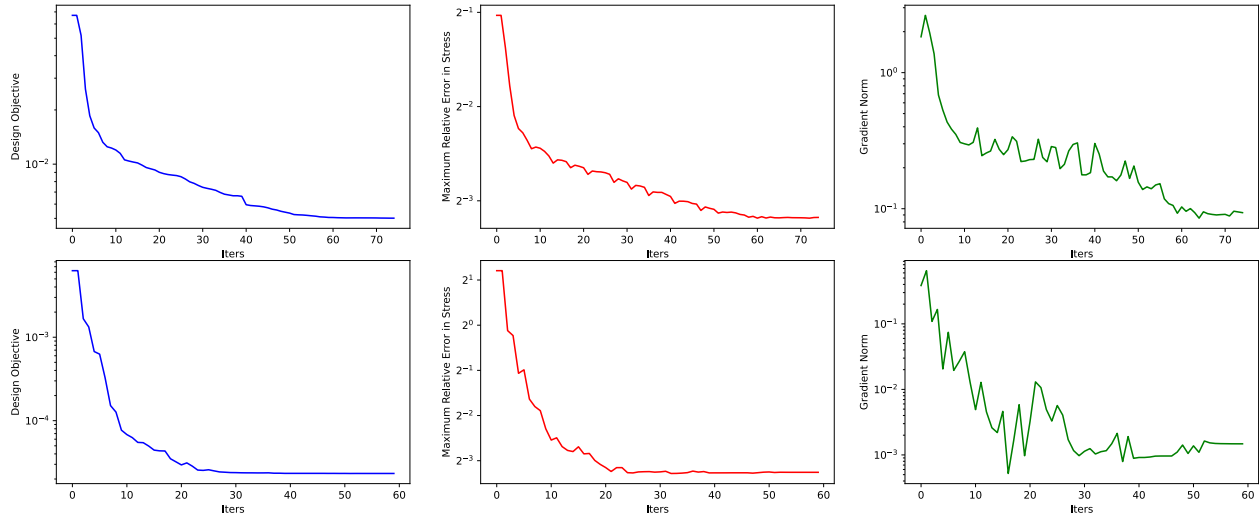


Figure S5 Convergence behavior of the nonlinear least squares solver on two characteristic material design problem instances: the top example achieved a maximum relative stress error improvement of $\sim 4\times$, while the bottom example achieved a $\sim 20\times$ improvement. The left column reports the objective minimized by the design optimization; the middle column reports the maximum relative (Frobenius norm) deviation from the target stress value over the strain domain; and the right column reports the optimality error, the norm of the gradient after removing the entries for variables up against bounds.

S5 Homogenization Statistics

The following expanded version of Figure 13 from the main paper demonstrates how the different variants of our homogenization approach scale with greater requested accuracy (controlled by the tolerance, ϵ , for the residual force norm of the interpolated equilibrium deformation). We note that the number of Newton iterations per sample decreases with increased sampling density as expected due to our initialization with the improving partially built interpolant. The piecewise linear strategy uses slightly fewer iterations per sample because of its much higher sampling density—and because its statistics are dominated by the refinement phase (rather than initial coarse mesh sampling), where better interpolation-based initial guesses are available.

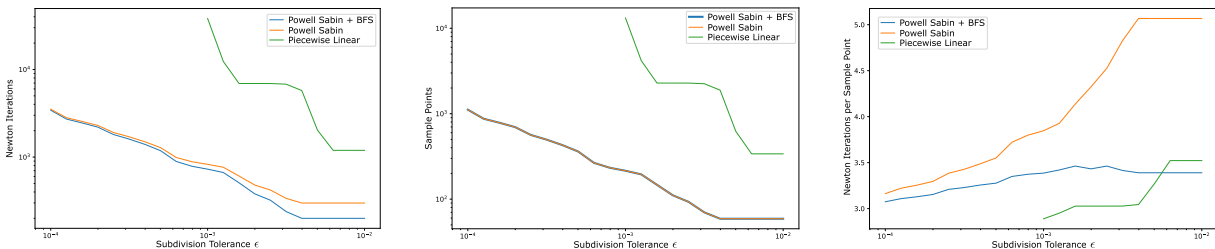


Figure S6 Left: the total number of Newton iterations needed to homogenize a characteristic microstructure geometry. Middle: the number of vertices in the refined interpolation mesh. Right: number of Newton iterations needed per equilibrium solve on average.

S6 Collision Removal Statistics

The following visualizations demonstrate the severity of collisions remaining after incorporating our collision penalty term in the design optimization—and the control we have over them via weight w_c . For these results, we reran our full design pipeline also with $w_c = 10^6$.

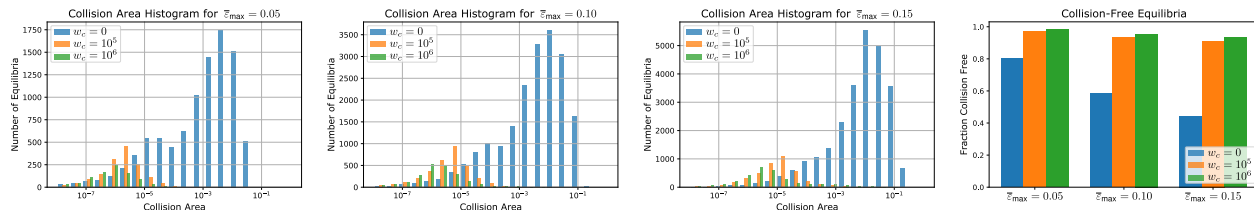


Figure S7 Histograms of the collision area of each equilibrium state computed during the homogenization of the optimal designs reported in Figure S9 for increasing macroscopic strain magnitudes $\bar{\epsilon}_{max}$. For reference, a collision area of 4.0 corresponds to collision polygons completely filling the period cell (occupying $[-1, 1]^2$). We observe a dramatic reduction in number and severity of collisions at $w_c = 10^5$, which is further improved by $w_c = 10^6$. Note that with our penalty, the majority of equilibria are collision free and are omitted from the three histograms at the left; the proportion of equilibria without collisions is reported in the rightmost histogram.

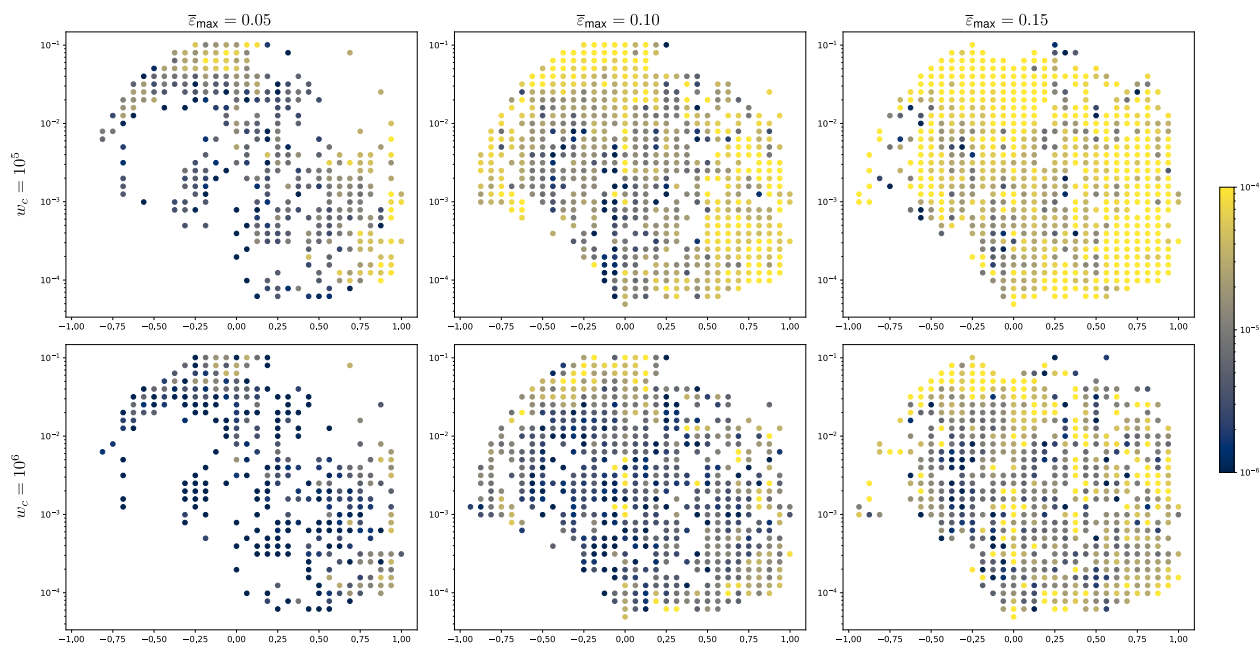


Figure S8 An alternate visualization of the data in Figure S7: the total collision area across all equilibrium samples for the optimal designs produced using penalty weight $w_c = 10^5$ (top) and $w_c = 10^6$ (bottom) at increasing strain magnitude $\bar{\epsilon}_{max}$ (left to right). Designs without collision are omitted from these plots. We note that the small collision areas are further reduced by increasing w_c .

Finally, we confirm with the plots in Figure S9 (an expanded version of the main paper’s Figure 15) that the material property fitting quality is not substantially impacted by increasing the penalty weight.

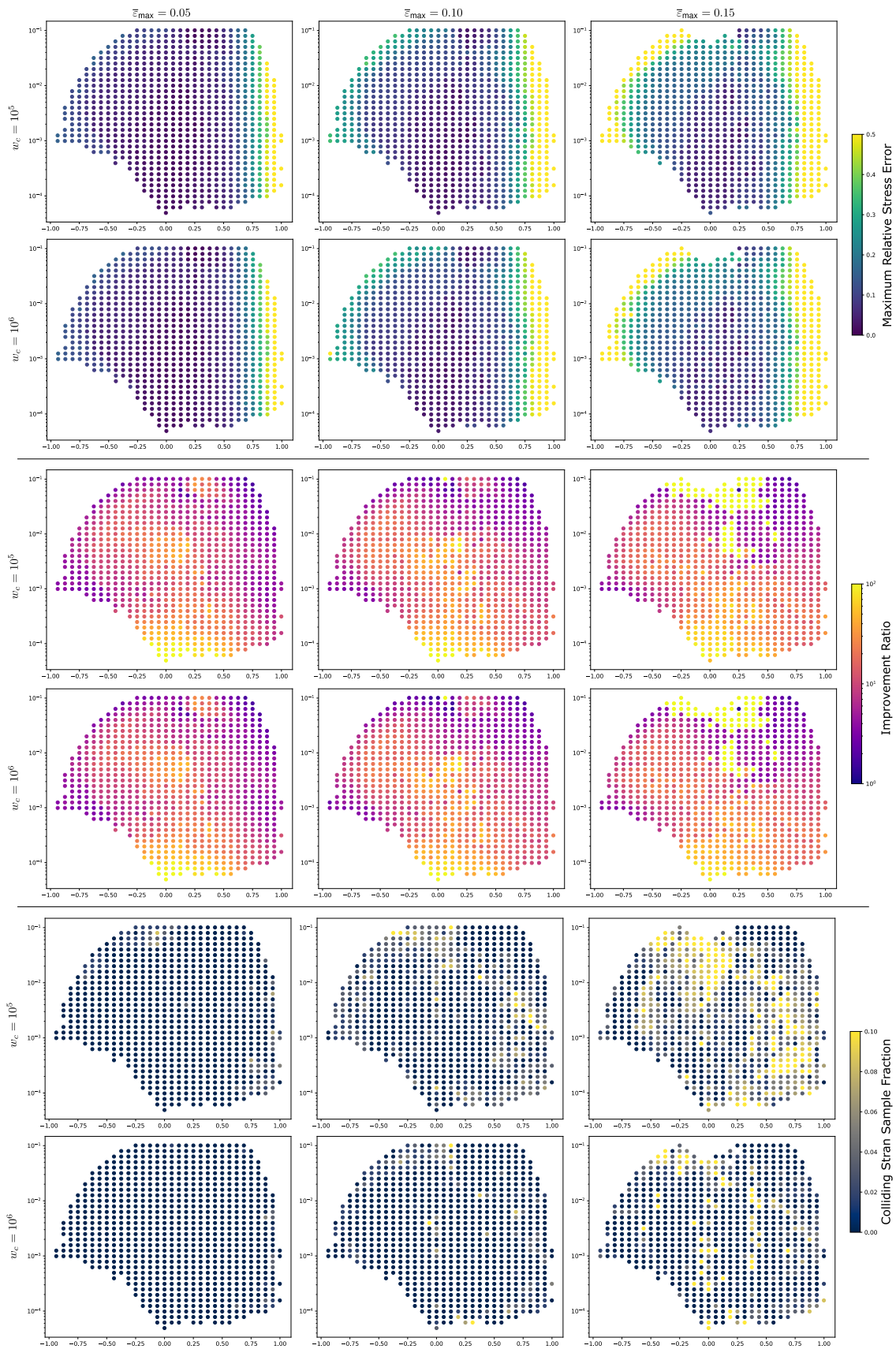


Figure S9 Detailed comparison of optimal designs computed with $w_c = 10^5$ and $w_c = 10^6$. The top pair of rows visualize the maximum relative deviation in stress across the homogenization domain; the middle pair visualizes the improvement in maximum relative stress error over the linearly elastic design; the bottom pair visualizes the fraction of the strain domain samples featuring collision area exceeding 10^{-5} .

References

- Anwei Liu and Barry Joe. 1996. Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. *Math. Comput.* 65, 215 (July 1996), 1183–1200. <https://doi.org/10.1090/S0025-5718-96-00748-X>
- Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. 2004. Registration of Point Cloud Data from a Geometric Optimization Perspective. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/1057432.1057435>
- Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Trans. Graph.* 36, 4, Article 122 (July 2017), 16 pages. <https://doi.org/10.1145/3072959.3073649>
- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages. <https://doi.org/10.1145/2766937>
- A.J. Worsey and B. Piper. 1988. A trivariate Powell-Sabin interpolant. *Computer Aided Geometric Design* 5, 3 (1988), 177–186. [https://doi.org/10.1016/0167-8396\(88\)90001-5](https://doi.org/10.1016/0167-8396(88)90001-5)