

Computational Homogenization and Inverse Design for Mechanical Metamaterial Systems

Boulder Summer School 2026

Julian Panetta

August 27, 2026



Figure 1 Examples of structures whose design we will study in these lectures. From left to right: metamaterials designed to emulate continuum elastic materials ([59, 39, 64]); surface-based inflatable structures that deploy into a programmed shape [41, 46] upon inflation; and two deployable structures based on bistable auxetic metamaterials and Hoberman-sphere type mechanisms [12, 45].

These notes accompany three lectures¹ on the computational design of flexible elastic metamaterials [7] and the larger structures that can be built from them, such as shape-morphing surfaces [16] and compliant mechanisms [23]. We present a two-scale design methodology in which periodic homogenization is used to characterize and tailor the effective mechanical behavior of individual microstructure unit cells, while a higher-level optimization determines how these cells should be spatially arranged to achieve a functional goal at the macroscopic level.

Engineering a material's fine-scale geometry enables mechanical functionality beyond that of a traditional homogeneous material. By carefully designing its microstructure, one can create sports and safety equipment with superior performance, medical prostheses with patient-specific stiffness distributions, and shape-morphing surfaces that deploy from compact configurations into temporary architecture or emergency shelters. Figure 1 previews some of the structures whose design we will study in these lectures.

Our first lecture formulates the forward problem of characterizing the effective mechanical behavior of a given metamaterial design for a few different material systems. The second lecture covers numerical solution of elasticity and homogenization problems using the finite element method and shows how to extract effective material properties such as tangent stiffnesses from these simulations. The final lecture addresses inverse design problems, both at the microstructure level (designing unit cells to achieve desired properties) and at the macrostructure level (*e.g.*, producing a shape-morphing structure), using gradient-based optimization. Our overall goal is to provide a flavor of the many mathematical techniques that combine to form a powerful computational design pipeline for these material systems.

¹The lecture slides are available at <https://julianpanetta.com/teaching/BoulderSchool2026>.

Lecture 1: Forward Homogenization

1 Why use homogenization?

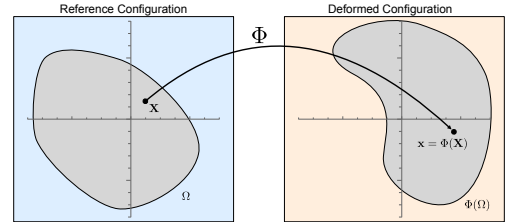
Homogenization theory is not just important in computational mechanics: it is also a powerful tool for computational design. As illustrated in Figure 1, designs incorporating metamaterials contain many repeated, geometrically complex features. Directly optimizing such systems at the resolutions achievable with digital fabrication tools such as CNC machines or 3D printers (potentially at the *teravoxel* scale) is intractable. Homogenization allows us to abstract away these fine-scale details when designing the overall object, yielding much lower-dimensional optimization problems. Even when full-resolution simulation and design iterations are feasible, the resulting large optimization problems are highly nonconvex. Without a good starting point, attempts to solve them are bound to get stuck in poor local optima, and the coarsened version of the design problem can provide an effective initialization.

The basic idea of homogenization is that, as the number of repetitions of a microstructure becomes sufficiently large, its behavior approaches that of a continuum material. Computational homogenization determines the properties of this limiting continuum by analyzing a small unit cell or representative volume element (RVE). Once these properties are known, we can use them to formulate a coarse, low-resolution simulation as an inexpensive surrogate for the inverse design of macroscopic objects. We can also design catalogs of architected materials that can be reused across future engineering applications.

2 Elasticity review

We begin by reviewing nonlinear elasticity theory as introduced in week one. Let $\Omega \subset \mathbb{R}^d$ be a reference configuration of a solid composed of a hyperelastic material with stored-energy density $\psi(F)$, where $F \in \mathbb{R}^{d \times d}$ is the deformation gradient. Points in the reference configuration are denoted by $\mathbf{X} \in \Omega$. Comparing to Professor James' lecture notes, notice that we have made some notational changes (using ψ in place of W and $\Phi(\mathbf{X})$ instead of $\mathbf{y}(\mathbf{x})$). For notational simplicity, we also have assumed that ψ is homogeneous and temperature-independent.

The total potential energy of the system when the body is deformed by a mapping $\Phi : \Omega \rightarrow \mathbb{R}^d$ is



$$E[\Phi] := \int_{\Omega} \psi(\nabla \Phi) \, d\mathbf{X} + \mathcal{L}[\Phi], \quad (1)$$

where $\mathcal{L}$ represents the potential energy of the external loads. A stable equilibrium can then be found by minimizing this potential energy over all admissible deformations, where admissibility is determined by the essential boundary conditions (*e.g.*, fixed displacements on part of the boundary) and potentially other user-defined constraints.

For small displacements $\mathbf{u}$ away from the rest configuration, *i.e.* deformations of the form $\Phi(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X})$, the elasticity term in (1) can be linearized to obtain the linear elasticity strain energy

$$\psi_{\text{lin}}(\mathbf{u}) := \frac{1}{2} \varepsilon_{\text{lin}}(\mathbf{u}) : \mathbf{C} : \varepsilon_{\text{lin}}(\mathbf{u}), \quad \varepsilon_{\text{lin}}(\mathbf{u}) := \text{sym}(\nabla \mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad \sigma_{\text{lin}} = \mathbf{C} : \varepsilon_{\text{lin}},$$

where $\mathbf{C}$ is the fourth-order elasticity tensor mapping infinitesimal strains ε_{lin} to stresses σ_{lin} . For isotropic materials, $\mathbf{C}$ is determined by two constants (*e.g.*, the Young's modulus and Poisson's ratio), but the effective *homogenized elasticity tensor* of a microstructure geometry is generally anisotropic.

3 The cell problem

As stated above, homogenization determines the continuum limit behavior of a periodic microstructure by analyzing only a single unit cell. For nonlinear elasticity, this means determining the *homogenized energy density* $\bar{\psi}(\bar{F})$ that measures the energy stored when the periodic tiling is subjected to a *macroscopic deformation gradient* $\bar{F}$. This analysis boils down to solving a certain equilibrium problem with periodic boundary conditions, which is referred to as the *cell problem*.

3.1 A heuristic cell problem derivation

This section presents a brief heuristic derivation of the cell problem for periodic homogenization of nonlinear elasticity, intended only to give intuition for its form. Rigorous derivations, typically using asymptotic expansions and Γ -convergence, are available in the homogenization literature, [10, 37].

The high-level idea for evaluating $\bar{\psi}(\bar{F})$ is to conceptually distort the infinite periodic lattice by $\bar{F}$ (applying the linear deformation map $\mathbf{X} \mapsto \bar{F}\mathbf{X}$) and then allow it to relax at the microscopic level into its lowest-energy state consistent with this macroscopic distortion. Our derivation makes a fundamental simplifying assumption that the deformed equilibrium state retains the same lattice translational symmetry as the original periodic tiling. An arbitrary such deformation is pictured below:

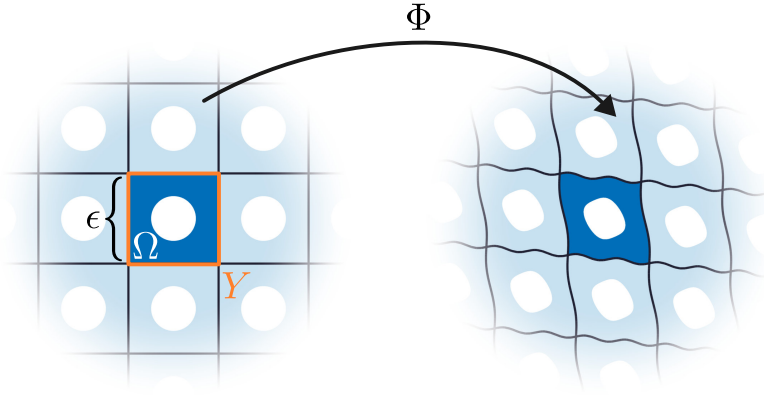


Figure 2 Left: a periodic tiling of the plane by a square “unit cell” Y of side length ϵ filled with a microstructure geometry Ω (dark blue). We use square cells for simplicity, but arbitrary periodic tilings can be treated similarly. Right: image of the tiling under an arbitrary symmetry-preserving deformation Φ .

In other words, just as the original grid is generated by repeating a single unit cell geometry Y at translations along the grid axes $\epsilon\mathbf{c}$ for $\mathbf{c} \in \mathbb{Z}^d$, we assume the full deformed tiling is generated by repeating $\Phi(Y)$ at translations along the distorted axes $\bar{F}\epsilon\mathbf{c}$ (so the distances between adjacent cells grow or shrink according to the macroscopic deformation $\bar{F}$). Formally, our assumption is that:

$$\Phi(\mathbf{X} + \epsilon\mathbf{c}) = \Phi(\mathbf{X}) + \epsilon\bar{F}\mathbf{c} \quad \forall \mathbf{X} \in Y, \mathbf{c} \in \mathbb{Z}^d.$$

We now introduce a convenient change of variables,

$$\Phi(\mathbf{X}) = \bar{F}\mathbf{X} + \omega(\mathbf{X}),$$

decomposing the deformation into a displacement away from the imposed macroscopic deformation $\bar{F}\mathbf{X}$ without losing generality. Restating our assumption in these new variables, we find:

$$\bar{F}(\mathbf{X} + \epsilon\mathbf{c}) + \omega(\mathbf{X} + \epsilon\mathbf{c}) = \bar{F}\mathbf{X} + \omega(\mathbf{X}) + \epsilon\bar{F}\mathbf{c} \quad \forall \mathbf{X} \in Y, \mathbf{c} \in \mathbb{Z}^d,$$

which is precisely a periodicity condition on ω . We conclude that deformation Φ has the requested symmetry if and only if it takes the “linear plus periodic” form $\Phi(\mathbf{X}) = \bar{F}\mathbf{X} + \omega(\mathbf{X})$, where ω is periodic on Y .

The homogenized energy density then is the minimum over such admissible deformations of the tiling’s average energy density, which since every cell deforms the same, is equal to the average energy density in a single cell:

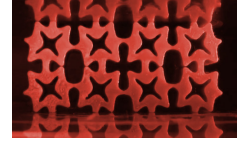
$$\bar{\psi}(\bar{F}) = \min_{\omega \text{ periodic}} \frac{1}{|Y|} \int_{\Omega} \psi(\bar{F} + \nabla \omega) d\mathbf{X}. \quad (2)$$

Note that this minimization problem may admit multiple minimizers in the nonlinear regime, for example due to buckling instabilities. In addition, it possesses a trivial nullspace that can be removed by imposing an additional constraint on ω (see Problem Set Exercise 1).

The first-order optimality condition for the minimization problem (2) is the weak form of the equilibrium equations and can be found simply by taking the first variation of the energy with respect to ω :

$$\left\langle \frac{\partial}{\partial \omega} \int_{\Omega} \psi(\bar{F} + \nabla \omega^*) d\mathbf{X}, \delta \omega \right\rangle = \int_{\Omega} \psi'(\bar{F} + \nabla \omega^*) : \nabla(\delta \omega) d\mathbf{X} = 0 \quad \forall \delta \omega \text{ periodic}. \quad (3)$$

We finally should acknowledge the important caveat that, under finite deformations, symmetry-breaking instabilities can lead to equilibria with a longer-wavelength periodicity, *e.g.* spanning a 2×2 supercell as seen in the inset. Computing the resulting post-buckling equilibrium can require analyzing supercells containing up to an arbitrarily large number of unit cells. The onset of such instabilities can, however, be detected efficiently on a single unit cell using Bloch-wave methods; the underlying mathematical framework is introduced in Xiaoming Mao’s lectures in the context of wave propagation in periodic media. We neglect such instabilities in this lecture.



3.2 Evaluating homogenized stresses

Now equipped with an expression for the homogenized energy density, we can compute the effective stress of the microstructure under the imposed distortion by differentiating with respect to $\bar{F}$.

Letting $\omega^*(\bar{F})$ denote the solution to the cell problem (2) obtained for a given $\bar{F}$, we obtain the homogenized PK1 (first Piola-Kirchhoff) stress under this loading:

$$\begin{aligned} \bar{\psi}'(\bar{F}) &= \frac{\partial}{\partial \bar{F}} \left[\frac{1}{|Y|} \int_{\Omega} \psi(\bar{F} + \nabla \omega^*(\bar{F})) d\mathbf{X} \right] \\ &= \frac{1}{|Y|} \int_{\Omega} \psi'(\bar{F} + \nabla \omega^*(\bar{F})) : \left(\mathbb{I} + \nabla \frac{\partial \omega^*}{\partial \bar{F}} \right) d\mathbf{X}, \end{aligned}$$

where $\mathbb{I}$ is the fourth-order identity tensor arising from $\frac{\partial \bar{F}}{\partial \bar{F}}$. We note the appearance of the equilibrium state derivative $\partial \omega^* / \partial \bar{F}$. While this is possible to compute efficiently (and is needed for tangent stiffness formulas, *i.e.* when differentiating with respect to $\bar{F}$ a second time), it actually does not contribute to the effective stress and can be dropped here. This is a direct consequence of the **envelope theorem**, but it is also clearly true according to (3), since $\partial \omega^* / \partial \bar{F}_{ij}$, whatever it is, must be an admissible perturbation. We conclude:

$$\bar{\psi}'(\bar{F}) = \frac{1}{|Y|} \int_{\Omega} \psi'(\bar{F} + \nabla \omega^*(\bar{F})) d\mathbf{X}. \quad (4)$$

4 Linear elastic homogenization

Although linear elasticity will lose accuracy for the large bending deformations characteristic of flexible structures, it is still an essential tool for computational design given how dramatic the computational savings

are (indeed, the vast majority of inverse design methods for mechanical metamaterials are based on linear elasticity).

In the linear setting, write $\bar{F} = I + \bar{\varepsilon}$, where $\bar{\varepsilon}$ is a symmetric infinitesimal macroscopic strain tensor. Linearizing the weak form (3) about the undeformed configuration and assuming $\psi'(I) = 0$ gives

$$\int_{\Omega} [C : (\bar{\varepsilon} + \varepsilon_{\text{lin}}(\omega^*))] : \varepsilon_{\text{lin}}(\phi) d\mathbf{X} = 0 \quad \forall \phi \text{ periodic}, \quad (5)$$

where $C := \psi''(I)$ is the elasticity tensor of the base material. The appearance of only the symmetric parts of $\nabla \omega^*$ and $\nabla \phi$ follows from the symmetries of C ; see also the second part of Problem Set Exercise 1.

Because this cell problem is linear in both $\bar{\varepsilon}$ and ω^* , the solution under arbitrary $\bar{\varepsilon}$ can be obtained by linear superposition of the solutions corresponding to a basis of symmetric strain tensors. Thus, fully characterizing the microscopic response requires *only three cell problem solves in two dimensions and six in three dimensions*.

The homogenized stress $\bar{\sigma}_{\text{lin}}(\bar{\varepsilon})$ also depends linearly on $\bar{\varepsilon}$, and this relationship can be expressed in terms of a *homogenized elasticity tensor* $\bar{C}$ as $\bar{\sigma}_{\text{lin}} = \bar{C} : \bar{\varepsilon}$ and evaluated using (4):

$$\bar{C} : \bar{\varepsilon} = \frac{1}{|Y|} \int_{\Omega} C : (\bar{\varepsilon} + \varepsilon_{\text{lin}}(\omega^*)) d\mathbf{X}. \quad (6)$$

This tensor $\bar{C}$ is a complete description of the metamaterial's effective material properties at small strains, and metamaterial design problems are often formulated in terms of it.

5 Characterizing flexible metamaterials over finite strain regions

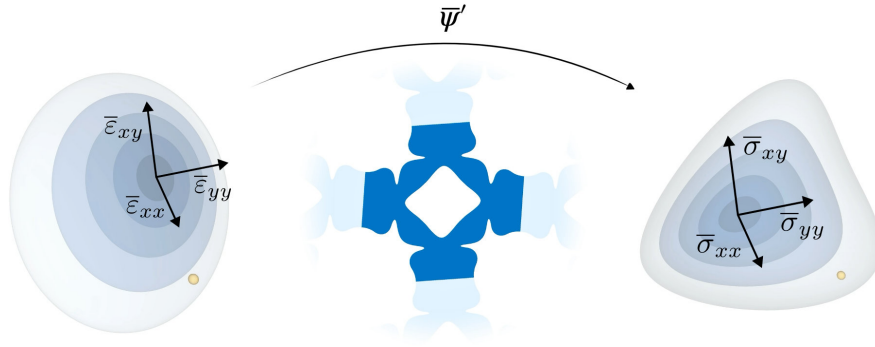


Figure 3 An illustration of the nonlinear homogenization process. Left: a sample in the strain region corresponding to $\bar{F} \in \mathcal{F}$ is selected. Center: the cell problem (3) is solved for the microscopic fluctuations ω^* . Right: Homogenized stress is evaluated using (4), yielding the value indicated by the marker.

For a flexible cell undergoing finite deformations, we cannot employ the efficient formulas (5) and (6) from the linearized setting. Instead, we must solve the nonlinear cell problem (3) for each macroscopic deformation gradient $\bar{F} \in \mathcal{F}$ in some “region of interest” $\mathcal{F}$.

To precisely and completely characterize a metamaterial's effective behavior over a finite region $\mathcal{F}$ (e.g., corresponding to strains up to 10%), we must theoretically solve an infinite number of cell problems: one for each $\bar{F} \in \mathcal{F}$. Most efforts to extend microstructure design to finite strains have avoided this complexity by controlling behavior in only a single loading scenario [35], at a few sampled loads [5], or along one deformation path (e.g., uniaxial stretching) [60, 13, 48].

However, approximate solutions to the characterization problem can be obtained by sampling and interpolation over $\mathcal{F}$ [31, 11, 64]. The approach we used for 2D homogenization in [64] was to construct a tetrahedralization of the 3D strain space $\mathcal{E}$ corresponding to $\mathcal{F}$ and solve the cell problem at each vertex of the tetrahedra. Using both ω^* and $\frac{\partial \omega^*}{\partial \bar{F}}$, we constructed a Hermite interpolation of the cell problem solution ω^* over each tetrahedron. This enables rapidly evaluating an approximate solution at any $\bar{F}$ without solving a new cell problem. The quality of the interpolated solution can be assessed by evaluating the residual of the cell problem (3), and we adaptively refine the tetrahedralization until the residual drops below a chosen tolerance. The nice thing about interpolating the cell problem solution ω^* rather than $\bar{\psi}$ or $\bar{\psi}'$ is that (i) it enables the residual calculation within the adaptive refinement loop; and (ii) when new points are inserted into the tetrahedralization, the interpolated ω^* is a strong initial guess for the new cell problem solve, warm-starting the nonlinear solver.

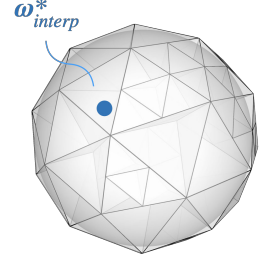


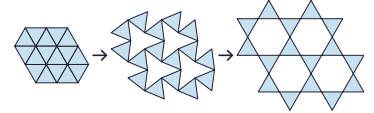
Figure 4 Interpolating ω^* over an adaptive tetrahedralization of strain region $\mathcal{E}$.

Many technical challenges remain open. Buckling instabilities and self-contact can make the response nonsmooth, requiring careful treatment in the interpolation. Our initial work avoided these issues by seeking to design simple as-linear-as-possible metamaterials, where our design objective avoids geometries with these complications. But, with more sophisticated design algorithms, such complicated behaviors could be harnessed to boost shock absorption [18, 17, 62] and tune mechanical resonances [61]. Furthermore, extending to 3D means that the dimensionality of $\mathcal{E}$ increases from 3 to 6, making sampling and interpolation much more challenging.

6 Axial stretching experiments and auxetic metamaterials

Materials tend to be more flexible under certain strains than others, and this should be reflected in the choice of $\mathcal{E}$ for characterization. Zhang et al. [64] discuss a strategy for picking reasonable $\mathcal{E}$ automatically, but sometimes the choice is obvious.

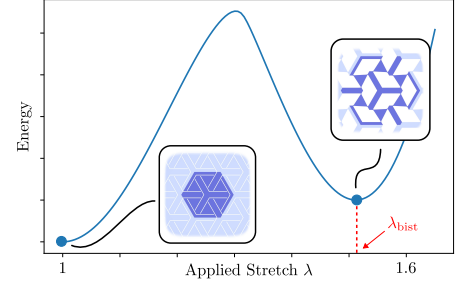
Auxetic metamaterials [44], structures with negative Poisson's ratios ν , *i.e.* that expand transversely when stretched along one axis, have become very popular in computational design. Isotropic metamaterials with $\nu = -1$ are especially convenient for designing shape-morphing surfaces since their behavior can be abstracted using the theory of conformal maps [27]. Such structures (see example in the inset) are unimode materials: they uniformly expand and contract in response to stretching along any axis but strongly resist other deformations. Therefore, the most natural choice of $\mathcal{E}$ is really a one-dimensional manifold parametrized by the stretch magnitude λ . In this case, it is most convenient to formulate the cell problem (2) in terms of this single loading parameter using the constrained optimization:



$$\bar{\psi}(\lambda) = \min_{\substack{\omega \text{ periodic, } \bar{F} \\ \text{s.t. } \bar{F}_{11} = \lambda}} \frac{1}{|\bar{Y}|} \int_{\Omega} \psi(\bar{F} + \nabla \omega) d\mathbf{X}. \quad (7)$$

We note that only the axial stretch component $\bar{F}_{11}$ of $\bar{F}$ is prescribed, while the remaining components are allowed to relax to minimize energy. In other words, the macroscopic deformation gradient $\bar{F}$ is now treated as a simulation variable along with ω .

This framework was applied to the design of bistable auxetic surface structures in [12], creating deployable surfaces that pop between two stable configurations (flat and curved). By solving (7) for a range of λ values, the homogenized energy profile reveals the bistable behavior of the analyzed pattern: the stretch $\lambda_{\text{bistable}}$ at which the second minimum of $\bar{\psi}$ appears indicates the amount of scaling between the open and closed states, and the stiffness $\bar{\psi}''(\lambda_{\text{bistable}})$ indicates how stable the open state is. Both quantities are important for the shape-morphing design problem, as will be discussed in Lecture 3.



7 Cell problem for inflatable membrane structures

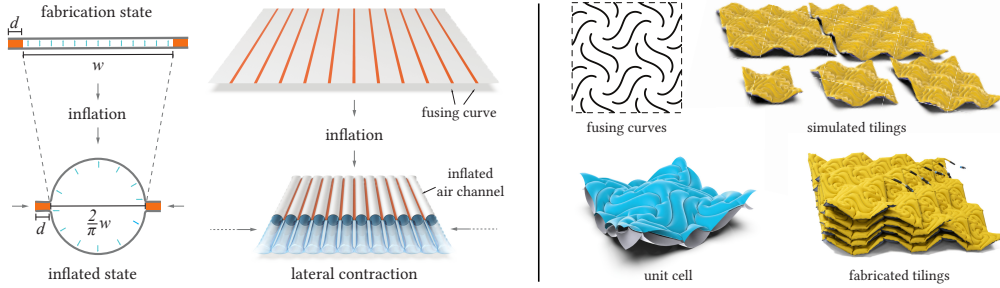


Figure 5 Left (from [41]), the behavior of a simple parallel tube inflatable upon pressurization. Right (from [46]), a more complex pattern and its inflation behavior computed using numerical homogenization.

Surface-based inflatable structures [50, 41] are created by fusing together two membranes along certain curves. This creates pockets that inflate into tubes upon pressurization. The inflation process induces an in-plane contraction that, when varied across the design, can be used to morph the surface into a desired shape. This will be studied in Lecture 3. While the behavior of the parallel tube at the left of Figure 5 can be deduced from basic geometric considerations, more complex patterns, like in the right half of that figure, require numerical homogenization.

The cell problem used to analyze this structure is similar to the two we have seen so far. The differences lie in the physics model, loading mechanism, and the fact that the deformation Φ now maps from the two dimensional design geometry Ω to three dimensional space $\mathbb{R}^3$. Furthermore, Ω can be understood as a non-manifold surface produced by taking sheets and identifying them along the fusing curves.

The elasticity model can be formulated using Koiter's shell theory [26, 56], which measures the energy stored by Φ in terms of how it stretches and bends the thin sheets. However, the sheets we use are so thin that the bending energy effects are negligible. We therefore adopt a membrane-only model, with a tension field theory relaxation [53, 55] for the membrane energy density (*i.e.*, neglecting compressive membrane forces) to account for the macroscopic effects of wrinkling while using coarse computational meshes.

Based on these simplifications, the cell problem for an inflatable structure becomes:

$$\bar{\psi}(p) := \min_{\omega \text{ periodic}, \bar{F}} E_{\text{membrane}}[\Phi] - p \text{Vol}(\Phi(\Omega)), \quad \Phi(\mathbf{X}) := \bar{F}\mathbf{X} + \omega(\mathbf{X}), \quad (8)$$

where $\text{Vol}(\Phi(\Omega))$ is the volume enclosed by the deformed surface $\Phi(\Omega)$, and p is the applied inflation pressure. Additional constraints are imposed to remove the trivial nullspace (see Problem Set exercise 1). Solving (8) yields $\bar{F}^*(p)$ as a byproduct, which informs us of the effective contraction of the tiled inflatable structure. This is the primary quantity needed to design a shape-morphing surface.

We have glossed over macroscopic bending, which is more complicated but important to consider for two reasons. First the inflation can actually induce bending in the deployed structure, which is relevant to shape

design. While this bending could be harnessed, the design algorithm we will present in Lecture 3 assumes that the patterns it uses inflate without bending and achieves shape morphing via metric frustration alone. Therefore we use our homogenization framework to detect when bending occurs and discard those patterns. Second, computing stiffnesses with respect to the macroscopic bending variable provides insights into the stability of the deployed structure. See [46] for how the admissible deformation field $\bar{F}\mathbf{X} + \boldsymbol{\omega}(\mathbf{X})$ is composed with a macroscopic bending operation.

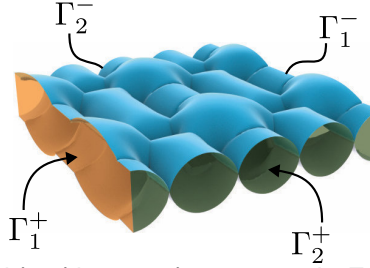


Figure 6 A periodic inflatable patch with arbitrary endcap geometries $\Gamma_i^\pm$ inserted to turn Ω into a closed surface.

One potential point of confusion is how to define the enclosed volume when the unit cell is not a closed surface (*e.g.*, as in Figure 5). It turns out that any way of closing off the surface that respects the periodicity of the cell (*i.e.*, that does not leave gaps or overlaps when tiled) gives the same definition of volume, which can be computed without reference to the endcap geometry. You are asked to prove this in Problem Set exercise 6.

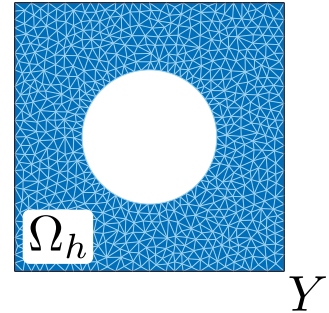
Lecture 2: Numerical Computations

8 Finite elements and numerical optimization

We have introduced several cell problem formulations without referencing how they can be solved efficiently on a computer. There are two quantities that need to be discretized (replaced by finite-dimensional approximations): the microstructure geometry Ω and the fluctuation displacement field $\boldsymbol{\omega}$. The simplest and most flexible approach is to employ the finite element method (FEM) to discretize both. While other possibilities exist, like the especially efficient FFT-based homogenization methods [14], the FEM formulation is straightforward to generalize to other physics systems like inflatables.

In the finite element method, the domain Ω is replaced by a mesh Ω_h of elements at some resolution h . For a 2D problem, an unstructured triangle grid can be used to closely approximate the shape. Interpolation nodes are then defined on each element (the simplest being the mesh vertices), and then approximations to continuous fields like $\boldsymbol{\omega}$ can be defined by interpolation. This interpolation can be written in the form

$$\boldsymbol{\omega}_{\text{fem}}(\mathbf{X}) := \sum_{i=1}^N \boldsymbol{\omega}_i \phi_i(\mathbf{X}),$$



where ϕ_i is the “shape function” (*i.e.*, the interpolation basis function) associated with node i and w_i is the value of $\boldsymbol{\omega}$ assigned to that node. The sum runs over all N interpolation nodes. In this way, FEM discretizes the function space $\boldsymbol{\omega}$ belongs to into a finite-dimensional space spanned by the shape functions ϕ_i .

Collecting the coordinates of all $\boldsymbol{\omega}_i$ into a vector $\mathbf{x} \in \mathbb{R}^{dN}$ and then plugging this interpolation expression into the cell problem objective converts it into an ordinary finite-dimensional optimization problem:

$$\bar{\psi}(\bar{F}) \approx \min_{\mathbf{x}} E_{\text{fem}}(\mathbf{x}, \bar{F}) \quad \text{where} \quad E_{\text{fem}}(\mathbf{x}, \bar{F}) := \frac{1}{|Y|} \int_{\Omega_h} \psi(\bar{F} + \nabla \boldsymbol{\omega}_{\text{fem}}) d\mathbf{X}, \quad (9)$$

$$\nabla \boldsymbol{\omega}_{\text{fem}} = \sum_{i=1}^N \boldsymbol{\omega}_i \otimes \nabla \phi_i.$$

Where did the periodic boundary conditions go? They are enforced by assigning the same degrees of freedom (optimization variables) $\boldsymbol{\omega}_i$ to corresponding nodes on opposite sides of the periodic cell, enabling the use of unconstrained optimization. For instance, all four corners of the 2D period cell (eight corners in 3D) end up sharing the same optimization variable $\boldsymbol{\omega}_i$.

If the cell problem involves optimizing also over a macroscopic state like $\bar{F}$ as in (7) and (8), then these can be included as additional optimization variables in $\mathbf{x}$. That is, we can write $\mathbf{x} = [\mathbf{x}_\mu, \mathbf{x}_M]$ where $\mathbf{x}_\mu$ are the microstructure degrees of freedom encoding $\boldsymbol{\omega}$, and $\mathbf{x}_M$ are the macroscopic degrees of freedom.

This optimization problem is best solved using Newton's method, which starts from an initial guess and then iteratively refines it. To simplify exposition, we will write $E_{\text{fem}}(\mathbf{x}, \bar{F})$ as just $E(\mathbf{x})$ in the following. The basic idea of Newton's method is to approximate the objective by a quadratic Taylor expansion around the current guess $\mathbf{x}_i$:

$$E(\mathbf{x}_i + \mathbf{d}) = E(\mathbf{x}_i) + \frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}_i) \mathbf{d} + \frac{1}{2} \mathbf{d} \cdot \frac{\partial^2 E}{\partial \mathbf{x}^2}(\mathbf{x}_i) \mathbf{d} + O(\|\mathbf{d}\|^3).$$

Then this quadratic approximation is minimized over $\mathbf{d}$ to produce the next guess. It is easy to show that the critical point of this quadratic proxy is the solution to:

$$H \mathbf{d} = -\mathbf{g}, \quad H := \frac{\partial^2 E}{\partial \mathbf{x}^2}, \quad \mathbf{g} := \left(\frac{\partial E}{\partial \mathbf{x}} \right)^\top.$$

This is a large sparse linear system, and so using a sparse solver is crucial for efficiency. To make the optimization robust, a line search is performed along the direction $\mathbf{d}$ to ensure that the energy decreases at each step. In other words, we take the step $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{d}$ where α may be less than 1 to ensure that $E(\mathbf{x}_{i+1}) < E(\mathbf{x}_i)$.

Because E is nonconvex (due to our use of nonlinear elasticity), H may not be positive definite, and the Newton step $\mathbf{d}$ may not be a descent direction (*e.g.*, it may step up toward a saddle point). We can detect these situations efficiently by using Cholesky factorization as our preferred method for solving the linear system. This computes the factorization $H = LL^\top$ where L is lower triangular and then obtains $\mathbf{d}$ by solving two triangular systems (fast forward and backward substitution sweeps). The factorization fails if H is not positive definite, and in this case we can modify H to be positive definite. Our framework for accelerating such finite element computations is described in [33] and available for download [from GitHub](#).

8.1 General form of a metamaterial's tangent stiffness

The discretized cell problem (9) offers a convenient framework for deriving a generic expression for the stiffness of a metamaterial (*e.g.*, to obtain the tangent elasticity tensor $\bar{\psi}''(\bar{F})$ or the stiffness $\partial^2 \bar{\psi} / \partial \lambda^2$ of a uniaxially stretched cell). The approach we use here introduces the same techniques that will be used to compute design sensitivities in Lecture 3.

Let $\mathbf{x}_M$ denote the macroscopic variables we wish to compute stiffness for. Then rewrite $E(\mathbf{x})$ as $E(\mathbf{x}_\mu, \mathbf{x}_M)$ where $\mathbf{x}_\mu$ collects all degrees of freedom allowed to relax when solving the cell problem. This includes $\boldsymbol{\omega}$ and potentially other macroscopic variables, *e.g.* the unconstrained components of $\bar{F}$ when analyzing an auxetic metamaterial. The solution to the cell problem for a fixed setting of $\mathbf{x}_M$ is:

$$\mathbf{x}_\mu^*(\mathbf{x}_M) := \underset{\mathbf{x}_\mu}{\operatorname{argmin}} E(\mathbf{x}_\mu, \mathbf{x}_M), \tag{10}$$

and we seek the stiffness:

$$K_M := \frac{\partial^2}{\partial \mathbf{x}_M^2} E(\mathbf{x}_\mu^*(\mathbf{x}_M), \mathbf{x}_M).$$

Computing a first derivative:

$$\frac{\partial}{\partial \mathbf{x}_M} E(\mathbf{x}_\mu^*(\mathbf{x}_M), \mathbf{x}_M) = \frac{\partial E}{\partial \mathbf{x}_M} + \frac{\partial E}{\partial \mathbf{x}_\mu} \frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M} \overset{0}{\rightarrow},$$

where the second term vanishes by the envelope theorem. Differentiating a second time obtains the stiffness formula:

$$K_M = \frac{\partial^2 E}{\partial \mathbf{x}_M^2} + \frac{\partial^2 E}{\partial \mathbf{x}_M \partial \mathbf{x}_\mu} \frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M}.$$

The derivative of the cell problem solution $\frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M}$ cannot be neglected this time, but we can solve for it assuming that $\mathbf{x}_\mu^*(\mathbf{x}_M)$ is a smooth function of $\mathbf{x}_M$. The cell problem solution $\mathbf{x}_\mu^*(\mathbf{x}_M)$ is defined implicitly by the first-order optimality condition:

$$\frac{\partial E}{\partial \mathbf{x}_\mu}(\mathbf{x}_\mu^*(\mathbf{x}_M), \mathbf{x}_M) = 0.$$

We can then apply the implicit function theorem to obtain the desired derivative. Computing the derivative of both sides of this equation:

$$\frac{\partial^2 E}{\partial \mathbf{x}_\mu^2} \frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M} + \frac{\partial^2 E}{\partial \mathbf{x}_\mu \partial \mathbf{x}_M} = 0 \quad \implies \quad \frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M} = - \left(\frac{\partial^2 E}{\partial \mathbf{x}_\mu^2} \right)^{-1} \frac{\partial^2 E}{\partial \mathbf{x}_\mu \partial \mathbf{x}_M}.$$

Therefore, $\frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M}$ can be obtained by solving linear systems involving the same Hessian $\frac{\partial^2 E}{\partial \mathbf{x}_\mu^2}$ that shows up in Newton's method.

The final computed stiffness can be expressed formally as:

$$K_M = \frac{\partial^2 E}{\partial \mathbf{x}_M^2} - \frac{\partial^2 E}{\partial \mathbf{x}_M \partial \mathbf{x}_\mu} \left(\frac{\partial^2 E}{\partial \mathbf{x}_\mu^2} \right)^{-1} \frac{\partial^2 E}{\partial \mathbf{x}_\mu \partial \mathbf{x}_M}, \quad (11)$$

though an efficient implementation would evaluate it by solving a sequence of linear systems as described above rather than explicitly forming the inverse of the Hessian.

Lecture 3: Computational Design of Metamaterial Systems

Now equipped with simulation models, we can use these models for physics-based computational design (*i.e.*, inverse design). The strategy we will pursue is to embed the simulation within an outer optimization loop that iteratively improves the design with respect to a user-specified objective.

The high-level structure of an inverse design problem is:

$$\min_{\mathbf{x}, \mathbf{p}} J(\mathbf{x}, \mathbf{p}), \quad \text{s.t. } \mathbf{x} = \underset{\tilde{\mathbf{x}}}{\operatorname{argmin}} E(\tilde{\mathbf{x}}, \mathbf{p}), \quad (12)$$

where $\mathbf{p}$ are the *design variables*, $J(\mathbf{x}, \mathbf{p})$ is the design objective measuring performance, and $E(\mathbf{x}, \mathbf{p})$ is the total potential energy of the system (after discretization). Additional constraints expressed in terms of $\mathbf{x}$ and $\mathbf{p}$ may also be imposed. Note that we formulate this optimization problem for the discrete, finite-dimensional simulation; it is instead possible to formulate it before discretization, but our derivations are simpler in the discrete setting.²

²Differentiating the discretized problem has the advantage that the gradients are automatically consistent with the discretization; ensuring consistency when differentiating the continuous problem before discretizing can be more subtle. The “distributed shape derivative” formulas we eventually derive below for a continuous elasticity model do yield the exact discrete gradients when discretized in the most straight-forward way.

We assume that the design variables $\mathbf{p}$ are continuous quantities. Their specific meanings depend on the chosen design representation (Section 10). They could represent the coordinates of mesh nodes for shape optimization, a material occupancy field for density-based topology optimization, parameters of a custom parametric design representation, or homogenized material property fields for a coarsened design optimization.

The “all at once” formulation (12), where $\mathbf{x}$ and $\mathbf{p}$ both appear as optimization variables, has its advantages, but it is a large problem that must be handled by a more complicated constrained optimization algorithm; it doesn’t let us leverage existing fast simulation code; and it does not make it easy to distinguish between unstable and stable equilibria (typically the argmin formulation is replaced with the nonlinear system $\nabla_{\mathbf{x}}E(\mathbf{x}, \mathbf{p}) = 0$, which loses this distinction). A common alternative, and the one pursued in these notes, is the *reduced* formulation, where the simulation state $\mathbf{x}$ is eliminated by solving the equilibrium problem:

$$\min_{\mathbf{p}} J(\mathbf{x}^*(\mathbf{p}), \mathbf{p}), \quad \text{where } \mathbf{x}^*(\mathbf{p}) := \underset{\mathbf{x}}{\operatorname{argmin}} E(\mathbf{x}, \mathbf{p}). \quad (13)$$

We note the similarity to (10), another setting where the simulation was used to eliminate state variables.

9 Sensitivity analysis

The most efficient optimization algorithms use the gradient of the “reduced objective” $\tilde{J}(\mathbf{p}) := J(\mathbf{x}^*(\mathbf{p}), \mathbf{p})$ with respect to the design variables:

$$\frac{\partial}{\partial \mathbf{p}} \tilde{J}(\mathbf{p}) = \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}. \quad (14)$$

The terms $\frac{\partial J}{\partial \mathbf{x}}$ and $\frac{\partial J}{\partial \mathbf{p}}$ are usually quite simple to compute by hand (or, if not, by automatic differentiation), but the term $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$ requires differentiating the simulation. Obtaining this term is referred to as *sensitivity analysis*.

The first steps essentially repeat Section 8.1. We note that $\mathbf{x}^*(\mathbf{p})$ is defined implicitly through:

$$\frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}), \mathbf{p}) = 0,$$

Differentiating with respect to $\mathbf{p}$ gives:

$$\frac{\partial^2 E}{\partial \mathbf{x}^2} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} = 0 \quad \implies \quad \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} = - \left[\frac{\partial^2 E}{\partial \mathbf{x}^2} \right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}. \quad (15)$$

In other words, for each design variable p_j in $\mathbf{p}$, the equilibrium state derivative $\frac{\partial \mathbf{x}^*}{\partial p_j}$ is obtained as the solution of a linear system involving the tangent stiffness matrix $\frac{\partial^2 E}{\partial \mathbf{x}^2}$ and the right-hand side $\frac{\partial^2 E}{\partial \mathbf{x} \partial p_j}$. Once the full matrix $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$ has been computed, it can be substituted into (14) to obtain the reduced gradient. This is called the *forward sensitivity* analysis approach.

The forward approach is intuitive and straightforward to debug (*e.g.*, you can check that $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$ is correct by finite differences), but it scales poorly for inverse design problems, where there can be thousands, millions, or even billions of design variables. Solving that many linear systems is simply not feasible.

Fortunately there is a better way in our setting, where only a few simulation-dependent quantities need to be differentiated (only the objective and any imposed constraints that depend on $\mathbf{x}^*$). Rather than immediately solving for $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$, we insert its unevaluated expression from (15) into (14) and leverage associativity to group the terms differently:

$$\frac{\partial}{\partial \mathbf{p}} \tilde{J}(\mathbf{p}) = - \left[\frac{\partial J}{\partial \mathbf{x}} \frac{\partial^2 E}{\partial \mathbf{x}^2}^{-1} \right] \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}.$$

The expression in brackets is a row vector $\boldsymbol{\lambda}^\top$ that can be computed by solving a *single linear system*:

$$\left(\frac{\partial^2 E}{\partial \mathbf{x}^2}\right)^\top \boldsymbol{\lambda} = \left(\frac{\partial J}{\partial \mathbf{x}}\right)^\top.$$

This equation is called the *adjoint equation* (due to the transposition applied to $\frac{\partial^2 E}{\partial \mathbf{x}^2}$), $\boldsymbol{\lambda}$ is called the *adjoint state*, and this approach is called the *adjoint method* or *adjoint sensitivity analysis*. Since $\left(\frac{\partial^2 E}{\partial \mathbf{x}^2}\right)^\top = \frac{\partial^2 E}{\partial \mathbf{x}^2}$ (Hessians are symmetric), the adjoint system also uses the same tangent stiffness matrix as Newton's method. The final gradient expression is:

$$\boxed{\frac{\partial}{\partial \mathbf{p}} \tilde{J}(\mathbf{p}) = -\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}},$$

which is efficient to evaluate. This process would then be repeated for any additional quantities like constraints that depend on $\mathbf{x}^*$. If you are familiar with automatic differentiation, the adjoint approach is analogous to reverse-mode autodiff (*i.e.* the backprop algorithm), while the forward sensitivity analysis approach is analogous to forward-mode autodiff.

9.1 Evaluating the mixed energy derivative

The sensitivity analysis formulas have surfaced the new derivative term $\frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$ that was not needed for simulation. More precisely, after solving the adjoint equation, we must evaluate its contraction with the adjoint state, row vector $\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$. Although this expression depends on the specific simulation model and meaning of $\mathbf{p}$ (with some examples considered in the next section), obtaining analytical formulas tends not to be fundamentally more difficult than for the Hessian $\frac{\partial^2 E}{\partial \mathbf{x}^2}$.

It usually simplifies hand derivations to exchange the differentiation order so that the energy is first differentiated with respect to $\mathbf{p}$. Also note that only a *directional derivative* needs to be computed along the direction $\boldsymbol{\lambda}$. Both of these observations will be exploited when deriving formulas for shape optimization in Section 10.2.

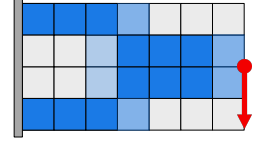
Note that hand calculations for this term can be avoided entirely using automatic differentiation [20]. A naïve global forward-mode implementation would propagate derivatives with respect to all design variables through the full computation of $\frac{\partial E}{\partial \mathbf{x}}$, multiplying its cost by the global parameter dimension (size of $\mathbf{p}$). This is impractical. Instead, autodiff can be applied to the scalar quantity $\boldsymbol{\lambda} \cdot \frac{\partial E}{\partial \mathbf{x}}$, treating $\boldsymbol{\lambda}$ as constant. This quantity is assembled as a sum of element contributions, each depending on only a small number of local design variables. We can therefore perform small, independent forward-mode autodiff computations on each element and assemble their results into the global gradient; the total work scales linearly with the number of elements and design variables, and the computation is easily parallelized.

10 Design representations

We now consider how the generic sensitivity analysis formulation developed above can be specialized to specific design representations. The choice of representation affects which geometries the optimization can explore, how readily manufacturing constraints can be enforced, and how derivatives with respect to the design variables are computed. We investigate two common approaches: density-based topology and shape optimization. Topology optimization involves simple derivative formulas and offers the greatest flexibility in the geometries that can be explored. This comes at the cost of a high-dimensional design space that is difficult to constrain for manufacturability and can be challenging to navigate to find simple, high-quality designs (*i.e.*, *good* local minima). Shape optimization, and more specifically parametric shape optimization of *e.g.* smooth lattice structures, offers a lower-dimensional design space that, while more limited, is easier to explore more thoroughly and to formulate manufacturing constraints over.

10.1 Density-based topology optimization

In density-based topology optimization, the design domain is discretized by a regular finite element grid, and each element e is assigned a density ρ_e . A binary variable $\rho_e \in \{0, 1\}$ would indicate whether the element is left empty or filled with solid, but optimizing over all such assignments is a challenging combinatorial problem. The standard relaxation instead allows $\rho_e \in [0, 1]$, enabling gradient-based optimization.



One possible physical interpretation of an intermediate density is a fine-scale solid-void microstructure having the corresponding volume fraction [2]. Homogenization theory can characterize extremal composites, such as sequential laminates, that achieve optimal stiffness for a prescribed amount of material, which is relevant to the goal of finding the stiffest structure under material usage constraints (*i.e.*, minimum compliance design [52]). This interpretation, however, leads to a considerably more complicated effective constitutive law. When the final design is intended to be binary, the simpler SIMP (Solid Isotropic Material with Penalization) interpolation [6] is commonly used:

$$C(\rho_e) = (\epsilon + (1 - \epsilon)\rho_e^p) C^{\text{base}}. \quad (16)$$

Here, C^{base} is the elasticity tensor of the fabrication material, $\epsilon > 0$ is a small stiffness assigned to empty regions that prevents the global stiffness matrix from becoming singular, and $p > 1$ penalizes intermediate densities. Increasing p makes gray elements inefficient in stiffness-per-unit-volume and thereby encourages a nearly binary result in design problems that involve a trade-off between stiffness and material usage.

The interpolation formula (16), and hence the elastic forces $\frac{\partial E}{\partial \mathbf{x}}$, are straightforward to differentiate with respect to the design variables ρ_e , especially in the case of a linear material model where the interpolation simply scales each element's force contribution.

The raw optimization variables need not be the physical densities themselves. Typically one writes $\rho_e = f_e(\mathbf{p})$, where f is a sequence of spatial filters and projections [9, 21]. These operations avoid non-physical checkerboard-type designs [15], control minimum feature size [51], and can also encode certain manufacturability constraints [29]. After the design objective is differentiated with respect to the physical densities ρ_e , the resulting gradient can be propagated back through the chain of filters to obtain the gradient with respect to the optimization variables $\mathbf{p}$.

10.2 Shape optimization

Shape optimization instead considers the design's topology (*i.e.*, the number of holes and the connectivity of the solid regions) to be fixed and seeks to optimize the rest configuration $\Omega(\mathbf{p})$ by reshaping its boundary, typically treating the interior as a homogeneous solid.

An intuitive and flexible approach is to optimize the positions of the nodes of the finite element computational mesh discretizing Ω (*parameter-free* shape optimization [58]), potentially with incremental remeshing [3] to maintain mesh quality. Alternatively, one can optimize over a *parametric design space*, where a lower-dimensional set of parameters $\mathbf{p}$ controls the shape. The latter is useful for designing smooth lattice microstructures amenable to additive manufacturing: after defining a lattice connectivity, the positions of the nodes, widths of the beams, and the smoothness of the junctions can all be controlled by a small number of parameters. Furthermore, fabrication constraints like minimum thickness and self-supporting angles can be formulated more or less directly on the parametric representation.

The chain rule provides a convenient way to organize sensitivity analysis even in the case of parametric design: first differentiate with respect to each mesh node position (Section 10.2.1) and then consider how perturbations of the parameters $\mathbf{p}$ induce perturbations of the mesh nodes (Section 10.2.2).

10.2.1 Shape derivatives

Whether the design variables $\mathbf{p}$ represent the positions of mesh nodes or some higher-level shape parameters, a perturbation $\delta \mathbf{p}$ can be thought of as changing the rest shape by flowing material points along some velocity

field $\mathbf{v}(\mathbf{X})$. We compute the necessary sensitivity analysis term $\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$ by first investigating how the elastic energy changes under such flows (with the deformed configuration held fixed).

The derivation should be familiar if you've seen a proof of the Reynolds Transport Theorem. We introduce a fictitious time parameter t for the flow, defining the edited shape as $\tilde{\Omega}(t) = \varphi(\Omega(\mathbf{p}))$, where $\varphi(\mathbf{X}) = \mathbf{X} + t\mathbf{v}(\mathbf{X})$ is the flow map. The initial rate of change of elastic energy is:

$$\frac{\partial E}{\partial \mathbf{p}} \delta \mathbf{p} = \left. \frac{\partial}{\partial t} \right|_{t=0} \int_{\tilde{\Omega}(t)} \psi \left(\frac{\partial}{\partial \tilde{\mathbf{X}}} \Phi \left(\varphi^{-1}(\tilde{\mathbf{X}}) \right) \right) d\tilde{\mathbf{X}}$$

Next, we pull the integral back to $\Omega(\mathbf{p})$ to keep the integration domain fixed:

$$\frac{\partial E}{\partial \mathbf{p}} \delta \mathbf{p} = \left. \frac{\partial}{\partial t} \right|_{t=0} \int_{\Omega(\mathbf{p})} \psi \left(\nabla \Phi (\nabla \varphi)^{-1} \right) \det(\nabla \varphi) d\mathbf{X}.$$

We can now move the time derivative inside the integral and, applying the identities

$$\nabla \varphi = I + t \nabla \mathbf{v}, \quad \left. \frac{d}{dt} \right|_{t=0} (\nabla \varphi)^{-1} = -\nabla \mathbf{v}, \quad \left. \frac{d}{dt} \right|_{t=0} \det(\nabla \varphi) = \nabla \cdot \mathbf{v},$$

arrive at:

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{p}} \delta \mathbf{p} &= \int_{\Omega(\mathbf{p})} -\psi'(\nabla \Phi) : (\nabla \Phi \nabla \mathbf{v}) + \psi(\nabla \Phi) (\nabla \cdot \mathbf{v}) d\mathbf{X}. \\ &= \int_{\Omega(\mathbf{p})} G : \nabla \mathbf{v} d\mathbf{X} \quad \text{where} \quad G := -(\nabla \Phi)^\top \psi'(\nabla \Phi) + \psi(\nabla \Phi) I. \end{aligned} \quad (17)$$

Finally, to obtain $\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$, we compute the directional derivative with respect to Φ along $\boldsymbol{\lambda}$. This can be done simply by differentiating G . Identifying the discrete adjoint vector $\boldsymbol{\lambda}$ with its finite-element interpolant:

$$\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} = \int_{\Omega(\mathbf{p})} (\delta G) : \nabla \mathbf{v} d\mathbf{X}, \quad \delta G = -(\nabla \boldsymbol{\lambda})^\top \psi'(\nabla \Phi) - (\nabla \Phi)^\top (\psi''(\nabla \Phi) : \nabla \boldsymbol{\lambda}) + (\psi'(\nabla \Phi) : \nabla \boldsymbol{\lambda}) I, \quad (18)$$

which when inserted into $\delta \tilde{J} = -\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} + \frac{\partial J}{\partial \mathbf{p}} \delta \mathbf{p}$ yields the distributed form of the shape derivative [30].

In contrast to the boundary integral formulas more commonly encountered in the shape optimization literature (which can be obtained by applying an integration by parts to (18)), this volume integral form makes it easy to compute the *exact* derivative of the finite-element elastic energy with respect to rest mesh vertex positions. Perturbing mesh node i induces a flow velocity $\mathbf{v} = \delta \mathbf{x}_i \phi_i$, where ϕ_i is the node's corresponding shape function introduced when defining the element's rest-shape embedding. Then $\nabla \mathbf{v} = \delta \mathbf{x}_i \otimes \nabla \phi_i$, and the derivative with respect to the node position is simply $\int_{\Omega(\mathbf{p})} (\delta G) \nabla \phi_i d\mathbf{X}$ (plus a possible “direct” contribution from $\frac{\partial J}{\partial \mathbf{p}}$). Leveraging the local support of ϕ_i , the derivatives with respect to each mesh node position can be computed efficiently by assembling contributions from the incident elements.

10.2.2 Parametric shape optimization

Implicit modeling [8] is a convenient framework for defining smooth parametric geometries due to its native support for constructive solid geometry operations (union, intersection, difference) and more organic smoothed approximations to them. Computer graphics researcher and artist Inigo Quilez has written several articles introducing implicit modeling techniques and showing the impressive results that can be achieved with them, like <https://iquilezles.org/articles/smin/> [42]. Specifically for metamaterial design, a lattice structure can be defined by unioning together a collection of beams, and sharp corners can be avoided in the resulting geometry by using a smooth approximation to the union operation [39, 64].

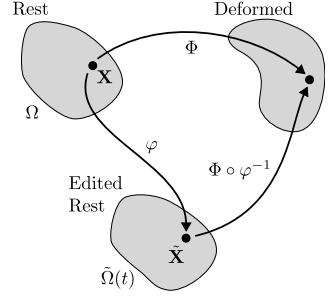


Figure 7 The current rest shape is perturbed by a flow map φ into the edited rest shape $\tilde{\Omega}$ while the deformation Φ is held fixed.

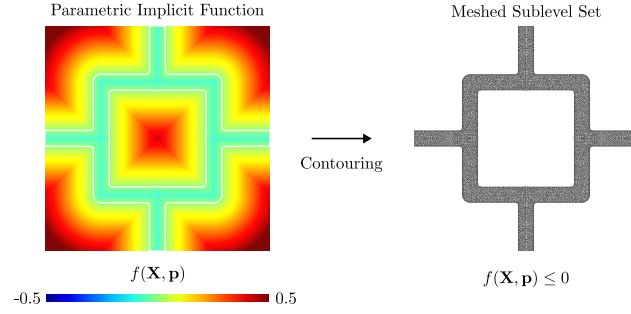


Figure 8 Obtaining a finite element mesh of parametric microstructure geometry $\Omega(\mathbf{p})$ by extracting the sublevel set of a level set function f .

Within this framework, the solid geometry $\Omega(\mathbf{p})$ is defined as the sublevel set of a parametric level set function f :

$$f(\mathbf{X}, \mathbf{p}) \leq 0. \quad (19)$$

A finite element mesh approximating $\Omega(\mathbf{p})$ can be generated directly from f (e.g., from a C++ function written to evaluate f) using meshing routines like those provided in CGAL [4].

Furthermore, the edit velocities $\mathbf{v}$ corresponding to each shape parameter can be obtained by differentiating (19). More precisely, we can understand $f(\mathbf{X}(\mathbf{p}), \mathbf{p}) = 0$ as implicitly defining the position $\mathbf{X}(\mathbf{p})$ of a certain point on the structure boundary. Differentiating this relationship with respect to an arbitrary parameter, we find:

$$\nabla f \cdot \frac{\partial \mathbf{X}}{\partial p_i} = -\frac{\partial f}{\partial p_i}(\mathbf{X}(\mathbf{p}), \mathbf{p}),$$

where ∇f , the derivative of f with respect to its spatial argument, can be rewritten in terms of the unit normal $\mathbf{N}$ as $\nabla f = \|\nabla f\| \mathbf{N}$. We conclude that the *normal velocity* of boundary points (i.e., the component of motion that actually changes the shape) induced by varying p_i is:

$$\mathbf{N} \cdot \frac{\partial \mathbf{X}}{\partial p_i} = -\frac{1}{\|\nabla f\|} \frac{\partial f}{\partial p_i}(\mathbf{X}(\mathbf{p}), \mathbf{p}),$$

the derivatives appearing on the right-hand side of which can be evaluated simply by applying automatic differentiation to the code evaluating f . A corresponding edit velocity $\mathbf{v}$ can be obtained by interpolating these boundary velocities smoothly throughout the interior mesh as discussed in [39] and inserted into (18) to complete the parametric sensitivity analysis.

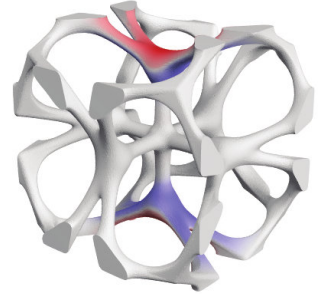


Figure 9 Edit velocity induced by perturbing a joint position parameter.

11 Inverse homogenization

Considering our focus on cell problems in Lecture 1, the most natural first design task for us to consider is *inverse homogenization*, where we seek a microstructure that produces a desired material property.

For linear elasticity, a natural goal is to fit the effective elasticity tensor $\bar{\mathbf{C}}(\mathbf{p})$ to a target tensor $\mathbf{C}_{\text{target}}$:

$$\tilde{J}(\mathbf{p}) = \frac{1}{2} \|\bar{\mathbf{C}}(\mathbf{p}) - \mathbf{C}_{\text{target}}\|_F^2. \quad (20)$$

For nonlinear design, we might want to fit stresses over a strain region $\mathcal{E}$:

$$\tilde{J}(\mathbf{p}) = \int_{\mathcal{E}} \|\bar{\psi}'(\bar{\mathbf{F}}, \mathbf{p}) - \bar{\sigma}_{\text{target}}(\bar{\mathbf{F}})\|_F^2 d\bar{\mathbf{F}}, \quad (21)$$

where the integral can be evaluated by quadrature on the interpolation mesh used for homogenization.

The generic adjoint sensitivity framework can be applied to this task with reasonable efficiency. However, inverse homogenization has additional variational structure: the quantities of interest, such as homogenized stress, are derivatives of the minimized cell energy. Exploiting this structure leads to formulas that are simpler and often more efficient than a direct application of the generic adjoint method. In the linear elasticity setting, the homogenized elasticity tensor $\bar{C}(\mathbf{p})$ can be differentiated efficiently without solving an adjoint equation (see Problem Set Exercise 10; Exercise 9 is related as well).

In the nonlinear setting, we have $\bar{\psi}(\bar{F}, \mathbf{p}) = E(\omega^*(\mathbf{p}, \bar{F}), \bar{F}, \mathbf{p})$. Assuming sufficient smoothness, the order of differentiation can be exchanged:

$$\frac{\partial}{\partial \mathbf{p}} \bar{\psi}'(\bar{F}, \mathbf{p}) = \frac{\partial}{\partial \bar{F}} \frac{\partial}{\partial \mathbf{p}} \bar{\psi}(\bar{F}, \mathbf{p})$$

Differentiating with respect to $\mathbf{p}$ *first* (with $\bar{F}$ fixed) allows the envelope theorem to eliminate $\partial \omega^* / \partial \mathbf{p}$:

$$\frac{\partial}{\partial \mathbf{p}} \bar{\psi}'(\bar{F}, \mathbf{p}) = \frac{\partial}{\partial \bar{F}} \frac{\partial E}{\partial \mathbf{p}}(\omega^*(\bar{F}, \mathbf{p}), \bar{F}, \mathbf{p}) = \frac{\partial^2 E}{\partial \mathbf{p} \partial \omega} \frac{\partial \omega^*}{\partial \bar{F}} + \frac{\partial^2 E}{\partial \mathbf{p} \partial \bar{F}}.$$

The resulting formula is efficient since we already compute $\frac{\partial \omega^*}{\partial \bar{F}}$ for Hermite interpolation.

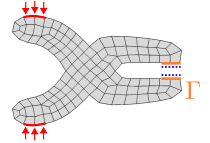
12 Inverse design of deformation behavior

An example task where the general adjoint formulas are useful is to reproduce target deformations. For example, we may wish for a deployable surface structure to approximate some target shape Ω_{target} upon actuation. At a high level, this would use an objective of the form:

$$J(\mathbf{x}, \mathbf{p}) = \text{dist}(\mathbf{x}, \Omega_{\text{target}})^2,$$

where $\text{dist}(\mathbf{x}, \Omega_{\text{target}})$ measures deviation between the deployed shape $\mathbf{x}^*(\mathbf{p})$ and the target shape Ω_{target} . This is often implemented using a one-sided point-to-surface distance: sampled points on the deployed structure are fit in a least-squares sense to their closest-point projections on Ω_{target} .

The distance objective can also be applied only to selected parts of the structure. For example, in designing a compliant gripper, the equilibrium problem includes a load applied to the handle, the design variables $\mathbf{p}$ encode a heterogeneous material-property field (*e.g.*, Young's modulus and Poisson's ratio), and the objective drives the gripper tips toward prescribed target positions.



13 A practical derivative test

Derivative formulas tend to be sufficiently complex that it is easy to make mistakes in their derivation or implementation. It is important to test their correctness thoroughly before attempting optimization. A simple and effective test is to compare against finite differences on random directional derivative calculations (verifying that the finite difference approximation converges to the analytical result at the expected rate). Choose a random perturbation $\delta \mathbf{p}$ and compare

$$\frac{\partial \tilde{J}}{\partial \mathbf{p}} \delta \mathbf{p} \quad \text{with} \quad \frac{\tilde{J}(\mathbf{p} + h\delta \mathbf{p}) - \tilde{J}(\mathbf{p} - h\delta \mathbf{p})}{2h}$$

over a logarithmic range of h . As h shrinks, the error should first decrease quadratically (a slope of 2 on a log-log plot) and then rise from floating-point and solver error. This is remarkably good at catching errors even if they impact only a few components.

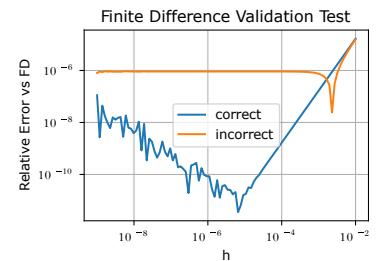


Figure 10 Characteristic test results on incorrect and correct analytical gradients.

14 Optimization algorithms

With design derivatives in hand, there is finally a question of which optimization algorithm to use. Gradient descent converges extremely slowly, and is not recommended for practical inverse design problems. The quasi-Newton L-BFGS method [38] is a good default choice because it requires only objective values and design gradients while converging much faster than gradient descent.

For objectives formulated as nonlinear least-squares problems, as in (20) and (21), and with a reasonably small number of parameters (*e.g.*, for parametric design), Gauss-Newton-based methods like Levenberg-Marquardt are especially effective. These require differentiating each residual rather than the objective, which could require more adjoint solves (although not for the inverse homogenization examples).

Newton's method is not practical for large-scale problems since it requires computing the Hessian $\frac{\partial^2 J}{\partial \mathbf{p}^2}$, which is generally dense and expensive to evaluate. Newton-CG, however, is a practical alternative: it needs only Hessian-vector products $\frac{\partial^2 J}{\partial \mathbf{p}^2} \delta \mathbf{p}$, which can be computed efficiently (*e.g.*, as in [40]), and can converge in substantially fewer iterations than L-BFGS. Each iteration is more costly, however.

Finally for shape optimization problems that directly optimize nodes of a computational mesh, Sobolev gradient descent [36] and Sobolev-BFGS [49, 24] with a well-chosen inner product/preconditioner can be surprisingly effective.

15 Two-scale design of shape-morphing structures

The gradient-based optimization framework described above is a powerful tool for computational design, but it requires a reasonable starting point since design optimizations tend to be highly non-convex. Furthermore, in the case of inverse design of shape-morphing structures like surface-based inflatables, it is usually not even clear how to obtain an initial mesh to begin the process of simulating and optimizing node positions. For such design tasks, we can build on the homogenization techniques introduced in Lecture 1 to develop a coarsened version of the design problem that is efficient to solve and yields high-quality initializations for the full-scale optimization.

We illustrate this approach for two material systems: auxetic metamaterials and inflatable surface structures. For both, the goal will be to design a compact planar structure that deploys into a specified target curved surface $\mathcal{M}^{\text{tgt}} \subset \mathbb{R}^3$ via a simple actuation mechanism (Figure 11).

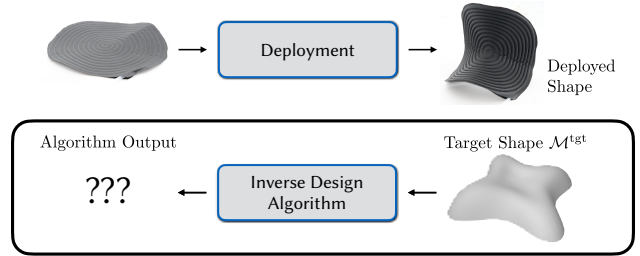


Figure 11 This section presents a two-scale approach for inverse design of deployable surfaces: determining a compact easy-to-fabricate structure that deploys to approximate a target surface $\mathcal{M}^{\text{tgt}}$ via some physical process like inflation or mechanical stretching.

15.1 Auxetic design

Auxetic surface structures, and more specifically 2D patterns with a Poisson's ratio of precisely -1 , have become extremely popular for the design of deployable and shape-morphing surfaces [63]. The reason for this popularity was discussed in Section 6: these are unimode metamaterials that admit uniform scaling but strongly resist other deformation modes. Consider a macroscopic sheet $\mathcal{M}$ fabricated from such metamaterials and, for simplicity, undergoing planar deformation. The unimode behavior constrains the deformation gradient to take the form $\bar{F} = \lambda R$, where $R : \mathcal{M} \rightarrow SO(2)$ represents local rotation and $\lambda : \mathcal{M} \rightarrow [\lambda_{\min}, \lambda_{\max}]$ represents pointwise scaling. In other words, admissible deformations of $\mathcal{M}$ correspond to the space of conformal maps with bounded scale factors (the bounds of which depend on the specific material system, *e.g.* $\lambda_{\min} = 1, \lambda_{\max} = 2$ for the twisted Kagome lattice [57] composed of rotating equilateral triangles).

As observed in [27], bounded-scale conformal maps offer sufficient flexibility to reproduce many detailed curved surfaces from an initially flat sheet. This can be understood geometrically from the prescribed

Gaussian curvature equation:

$$-\Delta_0 u = e^{2u} K, \quad \lambda = e^u, \quad (22)$$

which relates the spatially varying scale factor (or more precisely, the log conformal scale factor u) applied to the initially flat sheet $\mathcal{M}$ to the resulting Gaussian curvature K that the sheet must assume. This relationship tells us that, if we have some mechanism to control the degree of scaling that occurs at each point, we can use it to program the target surface's curvature into the material [28]. As explained in [27], the scale factor bounds do impose limits on what surfaces can be reproduced in this way, *e.g.* varying the scale factor over $[1, 2]$ just barely suffices to reproduce a hemisphere, and surfaces with a greater integrated Gaussian curvature cannot be covered by a single surface patch. However, conformal distortion can be reduced by inserting cone singularities³.

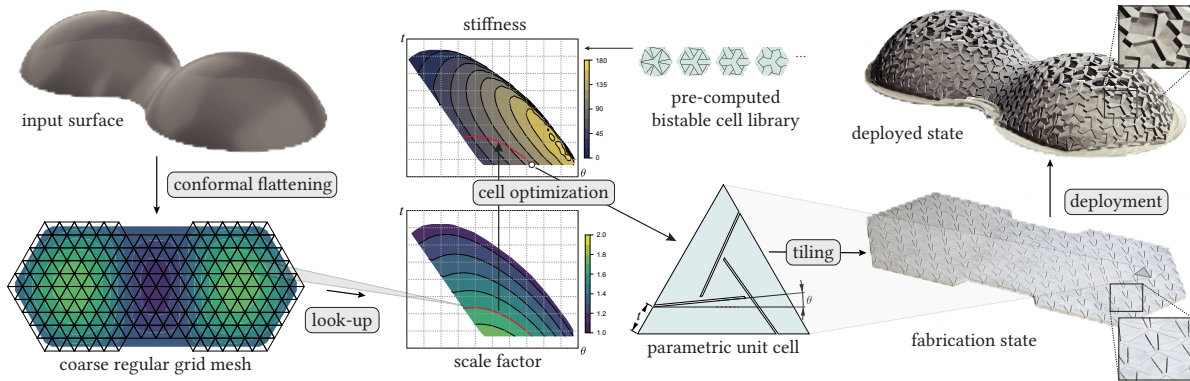


Figure 12 Coarsened inverse design pipeline for bistable auxetic surface structures [12] based on conformal mapping. These structures are spatially graded tilings of the triangular parametric unit cell pictured in the middle.

While equation (22) is a useful theoretical tool for understanding the intrinsic curvature control mechanism, the inverse design process for auxetic metamaterials is even simpler than it might suggest. It starts simply by applying an off-the-shelf conformal mapping software (*e.g.*, [34]) to compute a conformal flattening of $\mathcal{M}^{\text{tgt}}$ into the plane. More recent algorithms like Boundary First Flattening [47] even offer control over the shape of the resulting flattened sheet $\mathcal{M}$ and facilitate interactive cone insertion to reduce scale distortion to within the realizable bounds. Then the conformal scale factors computed from this flattening indicate the amount of scaling needed at each point, which then guides the selection of metamaterial geometry. The second step is typically to approximate the flat domain $\mathcal{M}$ by a regular grid of cells (at a chosen resolution) that match the unit cell shape used for homogenization. The final step uses the average conformal scale factor over each grid cell to select an appropriate unit cell geometry. Joining/blending together the chosen geometries yields a spatially varying design that encodes the curvature of $\mathcal{M}^{\text{tgt}}$ as desired. This design process, illustrated in Figure 12 for bistable auxetics [12], has been used for many different material systems [27, 45, 22].

For bistable auxetics, unit cell geometry selection is based on the results of nonlinear homogenization performed using the uniaxial loading analysis introduced in Section 6. That analysis identified, for a given metamaterial geometry, both the scale factor λ_{bist} at which the bistable unit cell settled into its second “open” equilibrium state and the stiffness around that stable state. The analysis was run for a large number of geometries sampled from a two-dimensional design space parametrized by a thickness parameter t and a cut angle θ (see Figure 12), producing a map from design parameters to the effective scale factor and stiffness properties. When populating a given coarse-grid cell, the conformal scale factor singles out a 1D curve in parameter space; searching along this curve for the highest-stiffness design then indicates parameters for a unit cell that robustly encodes the scaling needed to reproduce the curvature of $\mathcal{M}^{\text{tgt}}$.

³This entails cutting $\mathcal{M}^{\text{tgt}}$ along curves incident to each cone. During flattening, the surface then separates along these cuts to minimize distortion, and after fabrication and deployment the seams can be glued back together.

The bistable auxetic designs produced by this coarse-scale algorithm approximated the target surfaces so closely in deployment simulations that subsequent full-scale gradient-based optimization was deemed unnecessary. However, additional physics-based optimization could still help compensate for external loads.

15.2 Surface-based inflatable design

Surface-based inflatables also rely on spatially varying metric distortion to program the curvature of $\mathcal{M}^{\text{tgt}}$ into the material system. In the original parallel-tubes inflatables [41] pictured at the left of Figure 5, the metric distortion is trivial to determine geometrically from the inset diagram: the unfused pocket region of width w inflates into a tube of circumference $2w$, which has a contracted diameter of $\frac{2}{\pi}w$. This ultimately translates into a transverse length contraction factor of $\alpha = \frac{2w/\pi+d}{w+d}$. At the same time, the lengths along the tube are essentially unchanged by inflation. Therefore, the inflation can be understood geometrically as applying a deformation gradient of the form:

$$\bar{F} = U \begin{bmatrix} 1 & 0 \\ 0 & \alpha \\ 0 & 0 \end{bmatrix} R(\phi)^\top, \quad (23)$$

where $\phi : \mathcal{M} \rightarrow [0, 2\pi)$ indicates the original tube orientation in the plane, $U : \mathcal{M} \rightarrow SO(3)$ describes the orientation of the inflated surface after deployment, and singular value $\alpha : \mathcal{M} \rightarrow [\alpha_{\min}, \alpha_{\max}]$ is the transverse contraction, which can be varied spatially by adjusting $\frac{w}{d}$.

Such mappings feature highly anisotropic distortion and are far from conformal, so off-the-shelf conformal surface parametrization software cannot be used as for auxetics. However, a surface-flattening approach can still be developed to solve the inverse design problem: we seek an “inverse inflation” mapping $f : \mathcal{M}^{\text{tgt}} \rightarrow \mathbb{R}^2$ whose Jacobian at each point looks like the (pseudo-)inverse of an admissible deformation gradient. In other words, we need to enforce at each point of $\mathcal{M}^{\text{tgt}}$:

$$\nabla f = R(\phi) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma & 0 \end{bmatrix} U^\top, \quad \sigma \in [\sigma_{\min}, \sigma_{\max}],$$

where the singular value of ∇f in the transverse direction is the stretch factor $\sigma = 1/\alpha$.

In fact, a surface flattening algorithm targeting precisely this form of metric distortion was already created for inverse design of nematic elastomer sheets [1], another material system that applies length contraction as an actuation mechanism. That algorithm built on the popular local-global alternating optimization framework for distortion minimization [54, 32] developed in the computer graphics and geometry processing communities. The idea is to formulate the surface flattening problem as the following minimization:

$$\min_{f, \sigma, R, U} \int_{\mathcal{M}^{\text{tgt}}} \left\| \nabla f - R \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma & 0 \end{bmatrix} U^\top \right\|^2 dA,$$

subject to bound constraints on $\sigma : \mathcal{M} \rightarrow [\sigma_{\min}, \sigma_{\max}]$. When σ, R, U are held fixed, the problem reduces to solving a discrete Poisson equation (a linear system with a fixed system matrix whose Cholesky decomposition can be computed once at the start of optimization). Conversely, when f is held fixed, the optimal σ , R , and U fields can be extracted from the pointwise singular value decomposition of ∇f (snapping the singular values to the admissible range). Therefore, this minimization problem can be solved efficiently by alternating between global linear solves that update f and local SVD-based updates of the stretch factor and orientation fields.

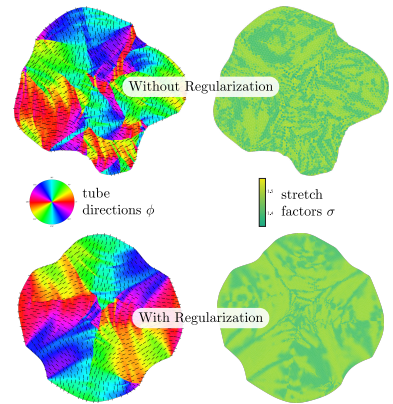
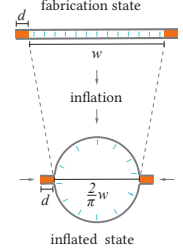


Figure 13 Surface flattenings and associated tube and stretch factor fields computed with and without the regularizations of [41].

Unfortunately, as seen in Figure 13, this algorithm produces tube orientation and stretch factor fields that are too noisy to extract high-quality tube structures from. Panetta et al. [41] therefore instead introduced a regularized version of the flattening problem:

$$\min_f \int_{\mathcal{M}^{\text{tgt}}} (\sigma_2 - 1)^2 + (\sigma_1 - [\sigma_1]_{\sigma_{\min}^{\max}})^2 dA + \mathcal{R}[f], \quad (24)$$

where σ_1 and σ_2 are the singular values of ∇f , $[\sigma_1]_{\sigma_{\min}^{\max}}$ means the projection of the larger singular value σ_1 onto the admissible interval, and $\mathcal{R}$ denotes a regularization functional. The regularization was designed to smooth the tube direction and contraction fields as well as promote beneficial alignment of the tubes with extrinsic curvature directions of $\mathcal{M}^{\text{tgt}}$ (since tubes traversing a ridge tend to unbend during inflation, resisting the encoded curvature); for full details see [41].

After solving (24), the coarse-scale design was “dehomogenized” by generating a stripe pattern [25, 43] at a chosen spatial frequency, yielding a mesh of the inflatable sheet and its fusing curves. Inflating the resulting design generally can reproduce the qualitative features of $\mathcal{M}^{\text{tgt}}$, but a subsequent shape optimization stage (optimizing the mesh vertex positions) significantly improves the shape approximation quality.

While the simple parallel-tube-based inflatables can approximate a large range of surfaces, even with detailed features, they have a significant weakness: in straight regions like Figure 5 (left), the structure has essentially zero resistance to bending around the tube axis, allowing the inflated sheets to freely roll. Depending on tube alignment, this highly anisotropic bending behavior can cause a catastrophic loss of stiffness in the inflated designs to the extent that they even fail to maintain the programmed shape under gravity. Patterns with more isotropic bending stiffnesses are needed to avoid this issue (and to make the alignment of material directions with the extrinsic curvature of $\mathcal{M}^{\text{tgt}}$ less of an issue). For example, breaking the parallel tubes up into a higher-frequency zig-zag pattern gives control over the anisotropy of both the stiffness and length contraction [19]. However, as patterns become more complicated, we lose the ability to predict their properties with geometric arguments alone and rely on the homogenization approach developed in Section 7, example results of which are presented in Figure 14.

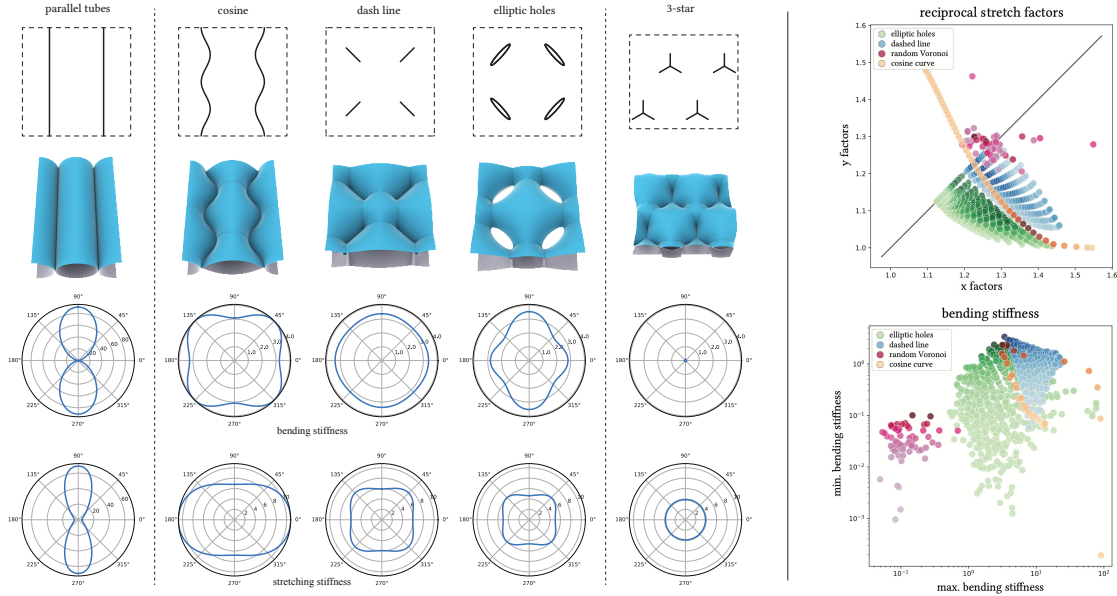


Figure 14 Example homogenization results obtained in [46] by solving the inflatable periodic homogenization cell problem and computing stiffnesses via (11). Contraction factors are computed from the singular values of the equilibrium $\bar{F}$ and plotted at the top right for each tested pattern. The directional stiffness plots reveal vanishing bending stiffness of the parallel tube pattern that is alleviated by even a small-amplitude sinusoidal fusing curve. Surprisingly, the “3-star” pattern, while fully isotropic in contraction and stiffness, has vanishing bending stiffness.

By densely sampling patterns from a chosen parametric design space (*e.g.*, the elliptic holes or cosine curve families), numerical homogenization characterizes the maps from design parameters to contraction factors and stiffnesses. These maps can then be used to formulate a coarsened design optimization problem that is only a slight modification of (24)⁴:

$$\min_{f, \mathbf{p}} \int_{\mathcal{M}^{\text{tgt}}} (\sigma_1 - s_1^{-1}(\mathbf{p}))^2 + (\sigma_2 - s_2^{-1}(\mathbf{p}))^2 \, dA + \mathcal{R}[f, \mathbf{p}]. \quad (25)$$

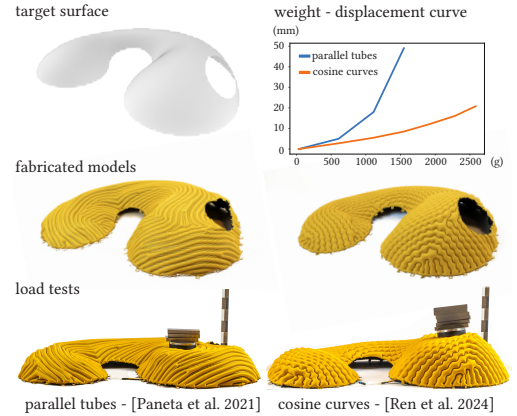
The main change is that the optimization is posed also over a pattern parameter field $\mathbf{p} : \mathcal{M}^{\text{tgt}} \rightarrow \mathbb{R}^{n_p}$ where n_p is the number of shape parameters controlling the unit cell geometry. In the case of the cosine curve family, $n_p = 1$. Each pattern has associated “reciprocal principal stretch factors” $s_1^{-1}(\mathbf{p})$ and $s_2^{-1}(\mathbf{p})$ determined from the equilibrium $\bar{F}$ during homogenization. Note that for parallel tubes, the optimization over $\mathbf{p}$ can be done analytically, reducing these target stretches to $[\sigma_1]_{\sigma_{\min}}^{\sigma_{\max}}$ and 1 as in (24). Constraints are imposed on $\mathbf{p}$ to keep the optimization within the valid, sampled region of the parameter space. Similarly to the parallel-tubes version, the regularization term $\mathcal{R}$ promotes smoothness of the pattern parameter fields and material frame orientation. It also encourages orienting weaker bending axes along the directions of greater normal curvature via a term proportional to:

$$\int_{\mathcal{M}^{\text{tgt}}} \int_0^{2\pi} \kappa_{\mathcal{M}}^{\text{tgt}}(\phi)^2 k_b(\phi; \mathbf{p}) d\phi \, dA,$$

where $\kappa_{\mathcal{M}}^{\text{tgt}}(\phi)$ denotes the normal curvature along a direction making angle ϕ with the first material frame axis (which in turn is chosen to align with the SVD frame of ∇f), and k_b is the pattern’s stiffness to bending onto a cylinder with that principal curvature direction.

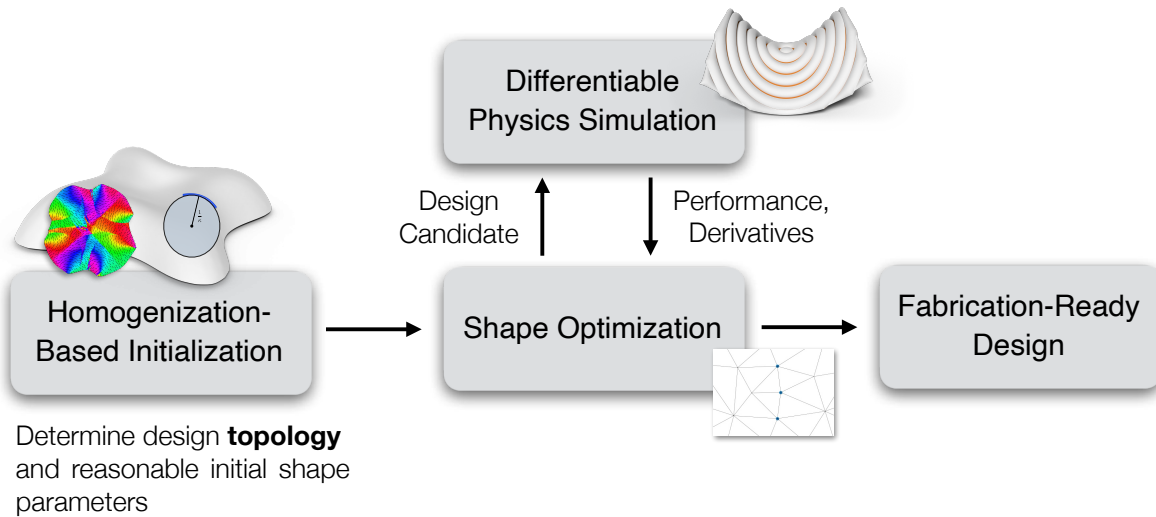
Solving this optimization problem yields the flattened sheet $\mathcal{M} = f(\mathcal{M}^{\text{tgt}})$ along with an associated pattern parameter field. The dehomogenization process is again based on generating material-frame-aligned stripe patterns at a chosen frequency; this time *two orthogonal stripe patterns* are generated to define a 2D texture coordinate system used to map the varying unit cell geometry corresponding to field $\mathbf{p}$ onto $\mathcal{M}$. For the full details, the reader is referred to [46].

The dehomogenized design is again a computational mesh with embedded fusing curves that can be used as a high-quality starting point for vertex-based shape optimization. The final results of this two-scale design process enabled by computational homogenization yield better shape approximation and significantly improved stiffness compared to the original parallel-tube inflatables as shown in the inset figure from [46].



⁴Note that in the case of a pattern undergoing isotropic contraction (like the “3-star” pattern), a conformal mapping approach can be used [22].

15.3 Full design pipeline



We conclude with a high-level summary of the two-scale design approach, which we have shown to be highly effective for optimal design problems involving metamaterials. First, computational homogenization is used to understand the material system and develop a tractable coarsened version of the otherwise high-dimensional, highly nonconvex design problem, generally posed in terms of macro-scale deformations and homogenized property fields. Solving this coarse-scale problem not only provides a good initialization for the full-scale optimization, but can determine basic features of the design, such as its topology, for which there may otherwise be no obvious starting point. The result is then dehomogenized to obtain an initial full-scale design that can be simulated and fine-tuned with shape optimization. As long as the design rules of the fabrication machine are respected by this optimization (*e.g.*, by imposing appropriate constraints on the design parameters), the resulting design can be directly fabricated and physically tested.

References

- [1] Hillel Aharoni, Yu Xia, Xinyue Zhang, Randall D. Kamien, and Shu Yang. Universal inverse design of surfaces with thin nematic elastomer sheets. *Proceedings of the National Academy of Sciences of the United States of America*, 115(28), 2018.
- [2] Grégoire Allaire, Eric Bonnetier, Gilles Francfort, and François Jouve. Shape optimization by the homogenization method. *Numerische Mathematik*, 76(1):27–68, 1997.
- [3] Grégoire Allaire, Charles Dapogny, and Pascal Frey. Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282:22–53, 2014.
- [4] Pierre Alliez, Clément Jamin, Laurent Rineau, Stéphane Tayeb, Jane Tournois, and Mariette Yvinec. 3D Mesh Generation. CGAL User Manual. URL https://doc.cgal.org/latest/Mesh_3/index.html. CGAL 6.2.
- [5] Reza Behrou, Maroun Abi Ghanem, Brianna C. Macnider, Vimarsh Verma, Ryan Alvey, Jinho Hong, Ashley F. Emery, Hyunsun Alicia Kim, and Nicholas Boechler. Topology optimization of nonlinear periodically microstructured materials for tailored homogenized constitutive properties. *Composite Structures*, 266, 6 2021. ISSN 02638223. doi: 10.1016/j.compstruct.2021.113729.

- [6] Martin P Bendsøe and Ole Sigmund. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, 69(9):635–654, 1999.
- [7] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. Flexible mechanical metamaterials, 10 2017. ISSN 20588437.
- [8] Jules Bloomenthal and Chandrajit Bajaj. *Introduction to implicit surfaces*. Morgan Kaufmann, 1997.
- [9] Blaise Bourdin. Filters in topology optimization. *International journal for numerical methods in engineering*, 50(9):2143–2158, 2001.
- [10] Andrea Braides and Anneliese Defranceschi. *Homogenization of Multiple Integrals*. Oxford University Press Oxford, November 1998. ISBN 978-0-19-850246-3 978-1-383-02032-8. doi: 10.1093/oso/9780198502463.001.0001. URL <https://academic.oup.com/book/53965>.
- [11] Antoine Chan-Lock, Jesús Pérez, and Miguel A Otaduy. High-order elasticity interpolants for microstructure simulation. In *Computer Graphics Forum*, volume 41, pages 63–74. Wiley Online Library, 2022.
- [12] Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. Bistable auxetic surface structures. *ACM Transactions on Graphics*, 40(4):1–9, August 2021. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459940. URL <https://dl.acm.org/doi/10.1145/3450626.3459940>.
- [13] Anders Clausen, Fengwen Wang, Jakob S. Jensen, Ole Sigmund, and Jennifer A. Lewis. Topology optimized architectures with programmable poisson’s ratio over large deformations. *Advanced Materials*, 27:5523–5527, 10 2015. ISSN 15214095. doi: 10.1002/adma.201502485.
- [14] T.W.J. De Geus, J. Vondřejc, J. Zeman, R.H.J. Peerlings, and M.G.D. Geers. Finite strain FFT-based non-linear solvers made simple. *Computer Methods in Applied Mechanics and Engineering*, 318:412–430, May 2017. ISSN 00457825. doi: 10.1016/j.cma.2016.12.032. URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782516318709>.
- [15] Alejandro Diaz and Ole Sigmund. Checkerboard patterns in layout optimization. *Structural optimization*, 10(1):40–45, 1995.
- [16] Krzysztof K Dudek, Muamer Kadic, Corentin Coulais, and Katia Bertoldi. Shape-morphing metamaterials. *Nature Reviews Materials*, 10(10):783–798, 2025.
- [17] Eric B. Duoss, Todd H. Weisgraber, Keith Hearon, Cheng Zhu, Ward Small, Thomas R. Metz, John J. Vericella, Holly D. Barth, Joshua D. Kuntz, Robert S. Maxwell, Christopher M. Spadaccini, and Thomas S. Wilson. Three-dimensional printing of elastomeric, cellular architectures with negative stiffness. *Advanced Functional Materials*, 24(31):4905–4913, August 2014. ISSN 1616-301X, 1616-3028. doi: 10.1002/adfm.201400451. URL <https://onlinelibrary.wiley.com/doi/10.1002/adfm.201400451>.
- [18] Tobias Frenzel, Claudio Findeisen, Muamer Kadic, Peter Gumbsch, and Martin Wegener. Tailored buckling microlattices as reusable light-weight shock absorbers. *Advanced Materials*, 28(28):5865–5870, July 2016. ISSN 0935-9648, 1521-4095. doi: 10.1002/adma.201600610. URL <https://onlinelibrary.wiley.com/doi/10.1002/adma.201600610>.
- [19] Tian Gao, Emmanuel Siéfert, Antonio DeSimone, and Benoît Roman. Shape programming by modulating actuation over hierarchical length scales. *Advanced Materials*, 32(47):2004515, 2020. doi: <https://doi.org/10.1002/adma.202004515>.
- [20] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

- [21] James K Guest, Jean H Prévost, and Ted Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International journal for numerical methods in engineering*, 61(2):238–254, 2004.
- [22] Siyuan He, Meng-Jan Wu, Arthur Lebé, and Mélina Skouras. Mataireals: Isotropic inflatable metamaterials for freeform surface design. In *Computer Graphics Forum*, volume 44, page e70190. Wiley Online Library, 2025.
- [23] Larry L Howell. Compliant mechanisms. In *21st Century Kinematics: The 2012 NSF Workshop*, pages 189–216. Springer, 2012.
- [24] Xinzhuo Hu, Roi Poranne, and Julian Panetta. Stress-aware panelization of freeform surfaces. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811387. URL <https://doi.org/10.1145/3811387>.
- [25] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. Stripe patterns on surfaces. *ACM Trans. Graph.*, 34, 2015.
- [26] Warner Tjardus Koiter. On the nonlinear theory of thin elastic shells. *Proc. Koninkl. Ned. Akad. van Wetenschappen, Series B*, 69:1–54, 1966.
- [27] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. Beyond developable: computational design and fabrication with auxetic materials. *ACM Trans. Graph.*, 35(4), July 2016. ISSN 0730-0301. doi: 10.1145/2897824.2925944. URL <https://doi.org/10.1145/2897824.2925944>.
- [28] Mina Konakovic-Lukovic, Julian Panetta, Keenan Crane, and Mark Pauly. Rapid deployment of curved surfaces via programmable auxetics. *Acm Transactions On Graphics*, 37(4):106, 2018. doi: <https://doi.org/10.1145/3197517.3201373>. URL <http://infoscience.epfl.ch/record/261586>.
- [29] Matthijs Langelaar. An additive manufacturing filter for topology optimization of print-ready designs. *Structural and multidisciplinary optimization*, 55(3):871–883, 2017.
- [30] Antoine Laurain and Kevin Sturm. Distributed shape derivative via averaged adjoint method and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(4):1241–1267, 2016.
- [31] B. A. Le, J. Yvonnet, and Q. C. He. Computational homogenization of nonlinear elastic materials using neural networks. *International Journal for Numerical Methods in Engineering*, 104:1061–1084, 12 2015. ISSN 10970207. doi: 10.1002/nme.4953.
- [32] Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. A local/global approach to mesh parameterization. In *Computer graphics forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.
- [33] Haleh Mohammadian, Xinzhuo Hu, Roi Poranne, and Julian Panetta. MeshFEM: A block-accelerated solver for nonlinear finite elements. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811386. URL <https://doi.org/10.1145/3811386>.
- [34] Patrick Mullen, Yiyang Tong, Pierre Alliez, and Mathieu Desbrun. Spectral conformal parameterization. In *Computer Graphics Forum*, volume 27, pages 1487–1494. Wiley Online Library, 2008.
- [35] Praveen Babu Nakshatralla, Daniel A Tortorelli, and KB Nakshatralla. Nonlinear structural design using multiscale topology optimization. part i: Static formulation. *Computer Methods in Applied Mechanics and Engineering*, 261:167–176, 2013.
- [36] J. W. Neuberger. *Sobolev gradients and differential equations*. Number 1670 in Lecture notes in mathematics. Springer-Verlag, Berlin; New York, 1997. ISBN 978-3-540-63537-6.

- [37] Stefan Neukamm and Mathias Schäffner. Quantitative homogenization in nonlinear elasticity for small loads. *Archive for Rational Mechanics and Analysis*, 230(1):343–396, October 2018. ISSN 0003-9527, 1432-0673. doi: 10.1007/s00205-018-1247-z. URL <http://arxiv.org/abs/1703.07947>. arXiv:1703.07947 [math.AP].
- [38] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.
- [39] Julian Panetta, Abtin Rahimian, and Denis Zorin. Worst-case stress relief for microstructures. *ACM Trans. Graph.*, 36(4):122:1–122:16, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073649. URL <http://doi.acm.org/10.1145/3072959.3073649>.
- [40] Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.*, 38(4):83:1–83:15, 2019. ISSN 0730-0301. doi: 10.1145/3306346.3323040. URL <http://doi.acm.org/10.1145/3306346.3323040>.
- [41] Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)*, 40(4): 1–14, 2021.
- [42] Inigo Quilez. Smooth minimum. <https://iquilezles.org/articles/smin/>, 2013. Accessed: 2026-08-25.
- [43] Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, October 2006. ISSN 0730-0301. doi: 10.1145/1183287.1183297. URL <http://doi.acm.org/10.1145/1183287.1183297>.
- [44] Xin Ren, Raj Das, Phuong Tran, Tuan Duc Ngo, and Yi Min Xie. Auxetic metamaterials and structures: a review. *Smart materials and structures*, 27(2):023001, 2018.
- [45] Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. Umbrella meshes: elastic mechanisms for freeform shape deployment. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.
- [46] Yingying Ren, Julian Panetta, Seiichi Suzuki, Uday Kusupati, Florin Isvoranu, and Mark Pauly. Computational homogenization for inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)*, 43(4):1–18, 2024.
- [47] Rohan Sawhney and Keenan Crane. Boundary first flattening. *ACM Transactions on Graphics (ToG)*, 37(1):1–14, 2017.
- [48] Christian Schumacher, Steve Marschner, Markus Cross, and Bernhard Thomaszewski. Mechanical characterization of structured sheet materials. *ACM Trans. Graph.*, 37(4):148:1–148:15, July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201278. URL <http://doi.acm.org/10.1145/3197517.3201278>.
- [49] Tobias Schwedes, David A. Ham, Simon W. Funke, and Matthew D. Piggott. *Mesh Dependence in PDE-Constrained Optimisation*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-59482-8 978-3-319-59483-5. doi: 10.1007/978-3-319-59483-5. URL <http://link.springer.com/10.1007/978-3-319-59483-5>.
- [50] Emmanuel Siéfert, Etienne Reyssat, José Bico, and Benoît Roman. Programming stiff inflatable shells from planar patterned fabrics. *Soft Matter*, 16(34):7898–7903, 2020.
- [51] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4):401–424, 2007.

- [52] Ole Sigmund and Kurt Maute. Topology optimization approaches: A comparative review. *Structural and multidisciplinary optimization*, 48(6):1031–1055, 2013.
- [53] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing inflatable structures. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.
- [54] Olga Sorkine, Marc Alexa, et al. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.
- [55] D. J. Steigmann. Tension-field theory. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 429(1876):141–173, 1990. ISSN 00804630. URL <http://www.jstor.org/stable/51778>.
- [56] David J. Steigmann. Koiter’s Shell Theory from the Perspective of Three-dimensional Nonlinear Elasticity. *Journal of Elasticity*, 111(1):91–107, March 2013. ISSN 0374-3535, 1573-2681. doi: 10.1007/s10659-012-9393-2. URL <http://link.springer.com/10.1007/s10659-012-9393-2>.
- [57] Kai Sun, Anton Souslov, Xiaoming Mao, and Tom C Lubensky. Surface phonons, elastic response, and conformal invariance in twisted kagome lattices. *Proceedings of the National Academy of Sciences*, 109(31):12369–12374, 2012.
- [58] Kenneth E. Swartz, Ketan Mittal, Mathias Schmidt, Jorge-Luis Barrera, Seth Watts, and Daniel A. Tortorelli. Yet another parameter-free shape optimization method. *Structural and Multidisciplinary Optimization*, 66(12):245, December 2023. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-023-03684-9. URL <https://link.springer.com/10.1007/s00158-023-03684-9>.
- [59] Davi Colli Tozoni, Jérémie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, and Denis Zorin. A low-parametric rhombic microstructure family for irregular lattices. *ACM Transactions on Graphics (TOG)*, 39(4):101–1, 2020.
- [60] Fengwen Wang, Ole Sigmund, and Jakob Søndergaard Jensen. Design of materials with prescribed nonlinear properties. *Journal of the Mechanics and Physics of Solids*, 69:156–174, 2014.
- [61] Pai Wang, Filippo Casadei, Sicong Shan, James C. Weaver, and Katia Bertoldi. Harnessing buckling to design tunable locally resonant acoustic metamaterials. *Physical Review Letters*, 113(1):014301, July 2014. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.113.014301. URL <https://link.aps.org/doi/10.1103/PhysRevLett.113.014301>.
- [62] Yifan Wang, Brian Ramirez, Kalind Carpenter, Christina Naify, Douglas C. Hofmann, and Chiara Daraio. Architected lattices with adaptive energy absorption. *Extreme Mechanics Letters*, 33:100557, 2019. ISSN 2352-4316. doi: 10.1016/j.eml.2019.100557. URL <https://www.sciencedirect.com/science/article/pii/S2352431619302251>.
- [63] Yue Wang, Yingying Ren, and Tian Chen. From kirigami to hydrogels: A tutorial on designing conformally transformable surfaces. *Journal of Applied Mechanics*, 90(4):044801, April 2023. ISSN 0021-8936, 1528-9036. doi: 10.1115/1.4056350. URL <https://asmedigitalcollection.asme.org/appliedmechanics/article/90/4/044801/1152136/From-Kirigami-to-Hydrogels-A-Tutorial-on-Designing>.
- [64] Zhan Zhang, Christopher Brandt, Jean Jouve, Yue Wang, Tian Chen, Mark Pauly, and Julian Panetta. Computational design of flexible planar microstructures. *ACM Trans. Graph.*, 42(6), 2023. ISSN 0730-0301. doi: 10.1145/3618396. URL <https://doi.org/10.1145/3618396>.