Computational Homogenization and Inverse Design for Mechanical Metamaterial Systems
Boulder Summer School 2026

Julian Panetta

July 22, 2026

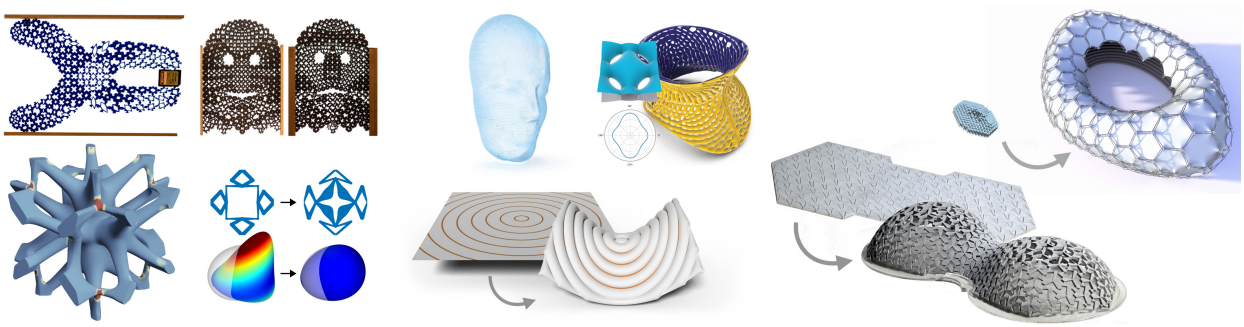


Figure 1 *Examples of structures whose design we will study in these lectures. From left to right: metamaterials designed to emulate continuum elastic materials ([33, 21, 37]); surface-based inflatable structures that deploy into a programmed shape [23, 26] upon inflation; and two deployable structures based on bistable auxetic metamaterials and Hoberman-sphere type mechanisms [5, 25].*

These notes accompany three lectures on the computational design of flexible elastic metamaterials [2] and the larger structures that can be built from them such as shape-morphing surfaces [8] and compliant mechanisms [11]. We present a two-scale design methodology in which computational 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 prescribed macroscopic behavior.

Engineering a material's fine-scale geometry enables mechanical functionality beyond that achievable with 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 and describes how this problem can be solved numerically. The second considers the inverse problem of finding a microstructure that achieves prescribed effective material properties using gradient-based optimization. The final lecture addresses the challenge of using these metamaterials to design a macro-scale object and discusses practical issues arising in fabrication. Our overall goal is to get a flavor of the many mathematical techniques that together produce a powerful computational design pipeline for these material systems.

The division of these notes into lectures is primarily thematic and likely won't align strictly with the time available in each class meeting. In particular, the first lecture is especially lengthy and will likely spill over into day 2.

Lecture 1: Forward Homogenization

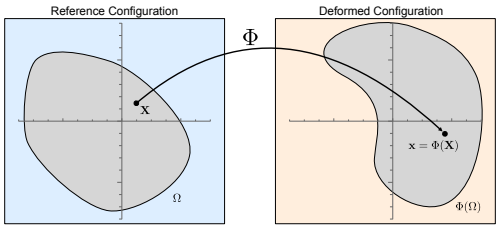
1 Why use homogenization?

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 formulate a low-resolution simulation using the homogenized material 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 made from a hyperelastic material described by stored-energy density $\psi(F)$, where $F \in \mathbb{R}^{d\times d}$ represents 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 $\Phi : \Omega \to \mathbb{R}^d$ is

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

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

$$\psi_{\mathrm{lin}}(\mathbf{u}) := \frac{1}{2}\varepsilon_{\mathrm{lin}}(\mathbf{u}) : C : \varepsilon_{\mathrm{lin}}(\mathbf{u}), \qquad \varepsilon_{\mathrm{lin}}(\mathbf{u}) := \mathrm{sym}\,(\nabla\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top\right), \qquad \sigma_{\mathrm{lin}} = C : \varepsilon_{\mathrm{lin}},$$

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

3 The periodic cell problem

As stated above, homogenization determines the continuum limit behavior of a periodic microstructure by analyzing only a single small unit cell. For nonlinear elasticity, this means determining the *homogenized energy density* $\overline{\psi}(\overline{F})$ that measures the energy stored when the periodic tiling is subjected to a *macroscopic deformation gradient* $\overline{F}$. 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, [3, 19].

The high-level idea for evaluating $\overline{\psi}(\overline{F})$ is to conceptually distort the infinite periodic lattice by $\overline{F}$ (applying the linear deformation map $\mathbf{X} \mapsto \overline{F}\mathbf{X}$) 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:

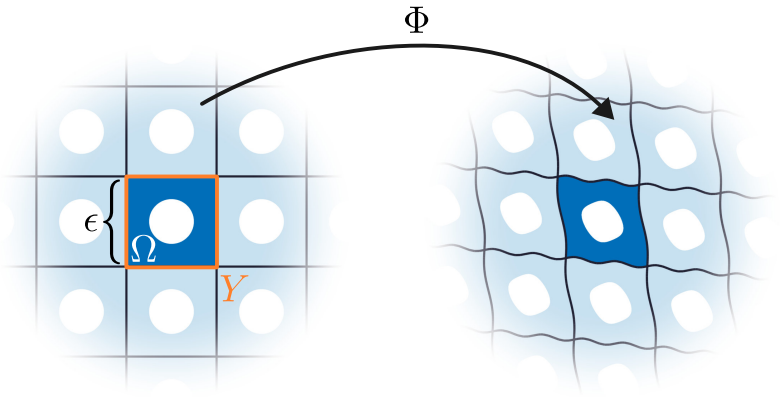


Figure 2 *Left: a periodic tiling of the plane by a square "unit cell" Y of side length ϵ filled with a microstructure geometry Ω (dark blue). We use square cells for simplicity, but arbitrary periodic tilings can be treated similarly. Right: image of the tiling under an arbitrary symmetry-preserving 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 $\overline{F}\epsilon\mathbf{c}$ (so the distances between adjacent cells grow or shrink according to the macroscopic deformation $\overline{F}$). Formally, our assumption is that:

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

We now introduce a convenient change of variables,

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

decomposing the deformation into a displacement away from the imposed macroscopic deformation without losing generality. Restating our assumption in terms of $\boldsymbol{\omega}$, we find:

$$\cancel{\overline{F}\left(\mathbf{X} + \epsilon\mathbf{c}\right)} + \boldsymbol{\omega}\left(\mathbf{X} + \epsilon\mathbf{c}\right) = \cancel{\overline{F}\mathbf{X}} + \boldsymbol{\omega}(\mathbf{X}) + \cancel{\epsilon\overline{F}\mathbf{c}} \quad \forall \mathbf{X} \in Y, \mathbf{c} \in \mathbb{Z}^d,$$

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

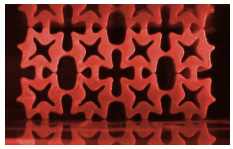
$$\boxed{\overline{\psi}(\overline{F}) = \min_{\boldsymbol{\omega} \text{ periodic}} \frac{1}{|Y|}\int_\Omega \psi(\overline{F} + \nabla\boldsymbol{\omega})\,\mathrm{d}\mathbf{X}.} \tag{2}$$

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 $\boldsymbol{\omega}$ (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 $\boldsymbol{\omega}$:

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

We finally should acknowledge the important caveat that, under finite deformations, symmetry-breaking instabilities can lead to equilibria with a larger 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 $\overline{F}$.

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

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

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

$$\overline{\psi}'(\overline{F}) = \frac{1}{|Y|}\int_\Omega \psi'(\overline{F} + \nabla\boldsymbol{\omega}^*(\overline{F}))\,\mathrm{d}\mathbf{X}. \tag{4}$$

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 $\overline{F} = I + \bar{\varepsilon}$, 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

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

where $C := \psi''(I)$ is the elasticity tensor of the base material. The appearance of only the symmetric parts of $\nabla\boldsymbol{\omega}^*$ and $\nabla\boldsymbol{\phi}$ 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 $\boldsymbol{\omega}^*$, 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 $\overline{\sigma}_{\text{lin}}(\bar{\varepsilon})$ also depends linearly on $\bar{\varepsilon}$, and this relationship can be expressed in terms of a *homogenized elasticity tensor* $\overline{C}$ as $\overline{\sigma}_{\text{lin}} = \overline{C} : \bar{\varepsilon}$ and evaluated using (4):

$$\boxed{\overline{C} : \bar{\varepsilon} = \frac{1}{|Y|} \int_\Omega C : \left(\bar{\varepsilon} + \varepsilon_{\text{lin}}(\boldsymbol{\omega}^*)\right) \mathrm{d}\mathbf{X}.} \tag{6}$$

This tensor $\overline{C}$ 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

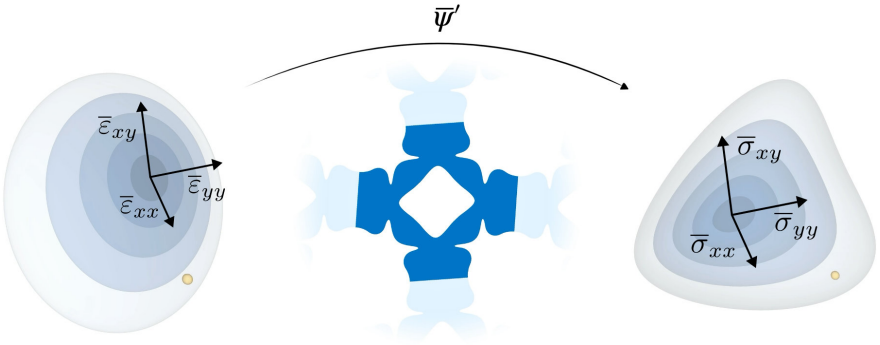


Figure 3 *An illustration of the nonlinear homogenization process. Left: a sample in the strain region corresponding to $\overline{F} \in \mathcal{F}$ is selected. Center: the cell problem* (3) *is solved for the microscopic fluctuations $\boldsymbol{\omega}^*$. Right: Homogenized stress is evaluated 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 $\overline{F} \in \mathcal{F}$ 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 $\overline{F} \in \mathcal{F}$. Most efforts to extend microstructure design to finite strains have avoided this complexity by controlling behavior in only a single loading scenario [17], at a few sampled loads [1], or along one deformation path (*e.g.*, uniaxial stretching) [34, 6, 27].

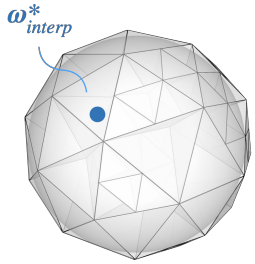


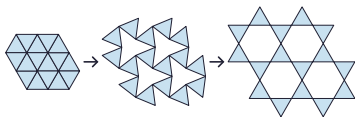
Figure 4 *Interpolating $\boldsymbol{\omega}^*$ over an adaptive tetrahedralization of strain region $\mathcal{E}$.*

However, approximate solutions to the characterization problem can be obtained by sampling and interpolation over $\mathcal{F}$ [15, 4, 37]. The approach we used for 2D homogenization in [37] 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 $\boldsymbol{\omega}^*$ and $\frac{\partial \boldsymbol{\omega}^*}{\partial \overline{F}}$, we constructed a Hermite interpolation *of the cell problem solution* $\boldsymbol{\omega}^*$ over each tetrahedron. This enables rapidly evaluating an approximate solution at any $\overline{F}$ 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 refined the tetrahedralization until the residual was below a chosen tolerance. The nice thing about interpolating the cell problem solution $\boldsymbol{\omega}^*$ rather than $\overline{\psi}$ or $\overline{\psi}'$ 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 $\boldsymbol{\omega}^*$ is a strong initial guess for the new cell problem solve, warm-starting the nonlinear solver.

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 [10, 9, 36] and tune mechanical resonances [35]. 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. [37] discuss a strategy for picking reasonable $\mathcal{E}$ automatically, but sometimes the choice is obvious.

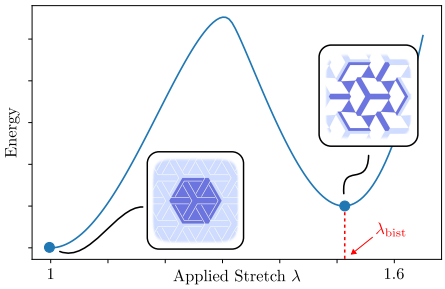


Auxetic metamaterials [24], structures with negative Poisson's ratios ν, *i.e.* that expand transversely when stretched along one axis, have become very popular in computational design. Isotropic metamaterials with $\nu = -1$ are especially convenient for designing shape-morphing surfaces since their behavior can be abstracted using the theory of conformal maps [14]. Such structures (see example in the inset) 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:

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

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

This framework was applied to the design of bistable auxetic surface structures in [5], creating deployable surface structures 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 $\overline{\psi}$ appears indicates the amount of scaling between the open and closed states, and the stiffness $\overline{\psi}''(\lambda_{\text{bistable}})$ 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

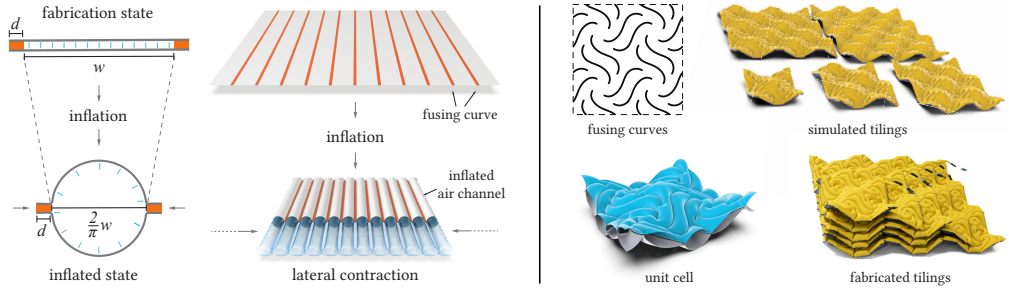


Figure 5 *Left (from [23]), the behavior of a simple parallel tube inflatable upon pressurization. Right (from [26]), a more complex pattern and its inflation behavior computed using numerical homogenization.*

Surface-based inflatable structures [29, 23] 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 two copies of the sheet and identifying them along the fusing curves.

The elasticity model can be formulated using Koiter's shell theory [13, 32], 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 [30, 31] for the membrane energy density (*i.e.*, neglecting compressive membrane forces) to account for wrinkling while using coarse computational meshes.

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

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

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 $\overline{F}^*(p)$ as a byproduct, which informs us of the effective contraction of the tiled inflatable structure. This is the primary quantity we need to know for designing 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 [26] for how the admissible deformation field $\overline{F}\mathbf{X}+\boldsymbol{\omega}(\mathbf{X})$ is composed with a macroscopic bending operation.

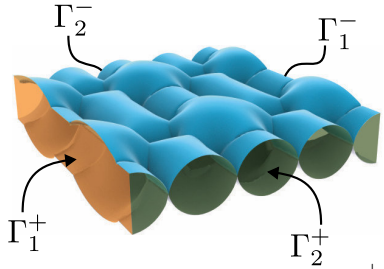


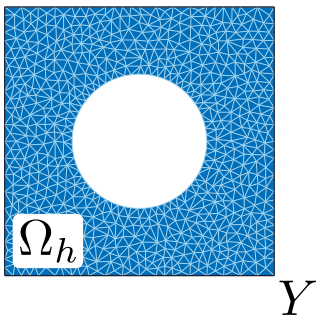
Figure 6 *A periodic inflatable patch with arbitrary endcap geometries $\Gamma_i^{\pm}$ inserted to turn Ω into a closed surface.*

One potential point of confusion is how we can 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.

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 there are other options, like the especially efficient FFT-based homogenization methods [7], 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:

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

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

Where did the periodic boundary conditions go? They are enforced by assigning the same degrees of freedom (optimization variables) $\boldsymbol{\omega}_i$ to nodes on opposite sides of the periodic cell, enabling the use of unconstrained optimization.

If the cell problem involves optimizing also over a macroscopic state like $\overline{F}$ 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 that it iteratively refines. To simplify exposition, we will write $E_{\text{fem}}(\mathbf{x}, \overline{F})$ 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 $\mathbf{x}_i$:

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

Then this quadratic approximation is minimized over $\mathbf{d}$ to produce the next guess. It is easy to show that the critical point 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. We can detect this 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 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 it to be positive definite. Our framework for accelerating such finite element computations is described in [16] 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 $\overline{\psi}''(\overline{F})$ or the stiffness $\partial^2\overline{\psi}/\partial\lambda^2$ of a uniaxially stretched cell). The approach we use here introduces the same techniques used to compute design sensitivities in Lecture 2.

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 $\overline{F}$ when analyzing an auxetic metamaterial. The solution to the cell problem for a fixed setting of $\mathbf{x}_M$ is:

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

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 gives:

$$\frac{\partial}{\partial \mathbf{x}_M} E(\mathbf{x}_\mu(\mathbf{x}_M), \mathbf{x}_M) = \frac{\partial E}{\partial \mathbf{x}_M} + \cancelto{0}{\frac{\partial E}{\partial \mathbf{x}_\mu}\frac{\partial \mathbf{x}_\mu}{\partial \mathbf{x}_M}},$$

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 gives:

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

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:

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

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 2: Computational design and adjoint sensitivity analysis

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 a generic inverse design problem is:

$$\min_{\mathbf{x},\mathbf{p}} J(\mathbf{x}, \mathbf{p}), \qquad \text{s.t. } \mathbf{x} = \operatorname*{argmin}_{\tilde{\mathbf{x}}} E(\tilde{\mathbf{x}}, \mathbf{p}), \tag{11}$$

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). It is also possible to formulate the optimization problem before discretization, but our derivations will be simpler on the discrete problem.[1]

We assume that the design variables $\mathbf{p}$ are continuous. Their specific meanings depend on the chosen design representation. They could represent the coordinates of mesh nodes for shape optimization, a material occupancy field for density-based topology optimization, level set function values for level set-based topology optimization, or parameters of a custom parametric design representation. We will consider the first two cases since the others reduce to them with an appropriate change of variables.

The "all at once" formulation (11), 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

[1] Differentiating 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.

algorithm; it doesn't let us leverage existing fast simulation code; and it does not make it easy to distinguish between unstable and stable equilibrium (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 we will use, is the *reduced* formulation, where the simulation state $\mathbf{x}$ is eliminated by solving the inner equilibrium problem:

$$\min_{\mathbf{p}} J(\mathbf{x}^*(\mathbf{p}), \mathbf{p}), \qquad \text{where } \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p}). \tag{12}$$

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:

$$\frac{\partial}{\partial \mathbf{p}} \tilde{J}(\mathbf{p}) = \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}. \tag{13}$$

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:

$$\frac{\partial^2 E}{\partial \mathbf{x}^2} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} = 0 \quad \Longrightarrow \quad \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}. \tag{14}$$

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 (13) 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 (14) into (13) 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 *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:

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

which is efficient to calculate. 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.

10 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 these depend on the specific simulation model and meaning of $\mathbf{p}$, obtaining analytical expressions for them tends not to be fundamentally more difficult than for the Hessian $\frac{\partial^2 E}{\partial \mathbf{x}^2}$. Time permitting, we will see in class an example for optimizing the nodes of a finite element mesh.

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 $\boldsymbol{\lambda}$.

Hand calculations can be avoided using automatic differentiation. A naïve global forward-mode implementation would propagate derivatives with respect to all design variables through the entire finite element computation, multiplying its cost by the global parameter dimension. 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. For bounded element locality, the total work scales linearly with the number of elements and design variables, and the computation is easily parallelized.

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 $\overline{C}(\mathbf{p})$ to a target tensor C_{target}:

$$\tilde{J}(\mathbf{p}) = \frac{1}{2}\|\overline{C}(\mathbf{p}) - C_{\text{target}}\|_F^2. \tag{15}$$

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

$$\tilde{J}(\mathbf{p}) = \int_{\mathcal{E}} \|\overline{\psi}'(\overline{F}, \mathbf{p}) - \overline{\sigma}_{\text{target}}(\overline{F})\|_F^2 \, \mathrm{d}\overline{F}, \tag{16}$$

where the integral can be evaluated by applying some quadrature rule 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, in some cases, more efficient than a direct application of the generic adjoint method. In the linear elasticity setting, the homogenized elasticity tensor $\overline{C}(\mathbf{p})$ can be differentiated efficiently without solving an adjoint system (see Problem Set Exercise 10; Exercise 9 is related as well).

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

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

Differentiating with respect to $\mathbf{p}$ first allows the envelope theorem to eliminate $\partial \boldsymbol{\omega}^* / \partial \mathbf{p}$. Therefore,

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

The resulting formula is efficient since we already compute $\frac{\partial \boldsymbol{\omega}^*}{\partial \overline{F}}$ for Hermite interpolation. The required mixed derivatives of E can be evaluated using the same element-level differentiation machinery described in the preceding section.

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 like the one in the upper left of Figure 1, 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

The derivative formulas are 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 random normalized perturbations δp 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.

Warning: the equilibrium problem must be solved substantially more accurately than the finite-difference perturbation being tested; otherwise simulation error can dominate before the expected convergence regime appears.

14 Optimization algorithms

Finally, there is 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 [20] 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 (15) and (16), 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 fully 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 [22]), 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 [18] and Sobolev-BFGS [28, 12] with a well-chosen inner product/preconditioner can be surprisingly effective.

Lecture 3 TBA.

References

[1] 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.

[2] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. Flexible mechanical metamaterials, 10 2017. ISSN 20588437.

[3] Andrea Braides and Anneliese Defranceschi. *Homogenization of Multiple Integrals*. Oxford University PressOxford, 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Krzysztof K Dudek, Muamer Kadic, Corentin Coulais, and Katia Bertoldi. Shape-morphing metamaterials. *Nature Reviews Materials*, 10(10):783–798, 2025.

[9] 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.

[10] 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.

[11] Larry L Howell. Compliant mechanisms. In *21st Century Kinematics: The 2012 NSF Workshop*, pages 189–216. Springer.

[12] 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.

[13] Warner Tjardus Koiter. On the nonlinear theory of thin elastic shells. *Proc. Koninkl. Ned. Akad. van Wetenschappen, Series B*, 69:1–54, 1966.

[14] 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.

[15] 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.

[16] 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.

[17] Praveen Babu Nakshatrala, Daniel A Tortorelli, and KB Nakshatrala. Nonlinear structural design using multiscale topology optimization. part i: Static formulation. *Computer Methods in Applied Mechanics and Engineering*, 261:167–176, 2013.

[18] 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.

[19] Stefan Neukamm and Mathias SchÃ¤ffner. 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].

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Emmanuel Siéfert, Etienne Reyssat, José Bico, and Benoît Roman. Programming stiff inflatable shells from planar patterned fabrics. *Soft Matter*, 16(34):7898–7903, 2020.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.