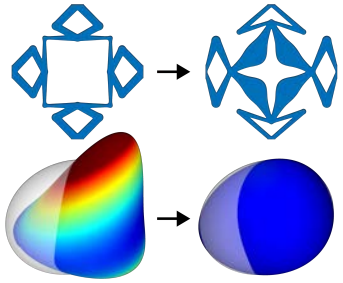
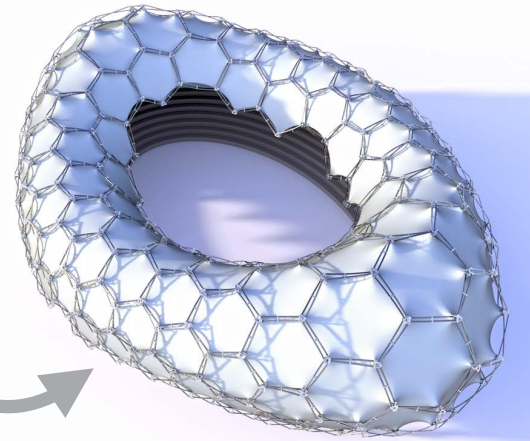
Computational Homogenization and Inverse Design for Mechanical Metamaterial Systems

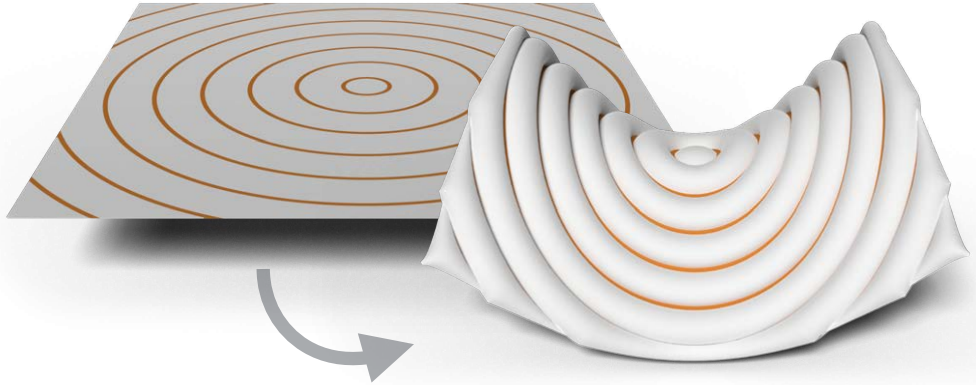




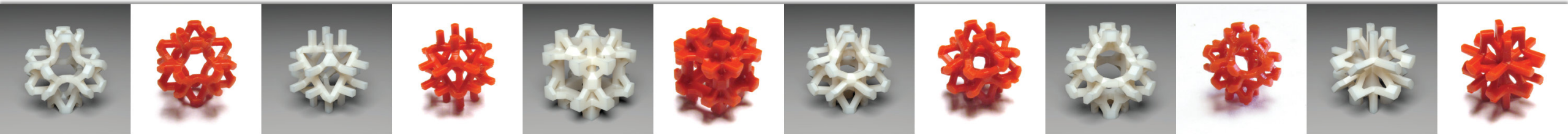
Boulder Summer School, 2026

Julian Panetta
University of California, Davis





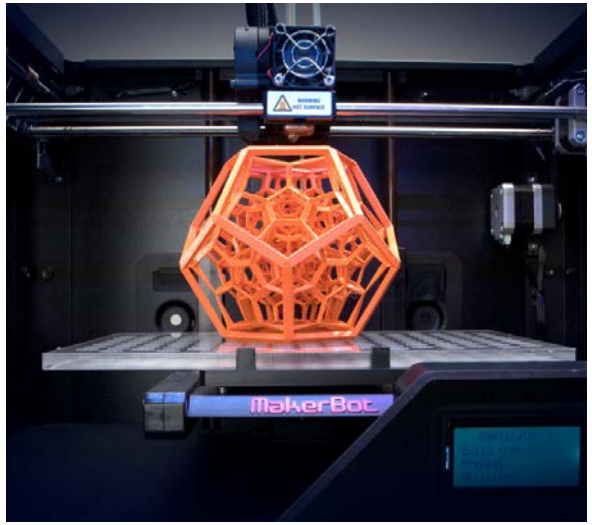




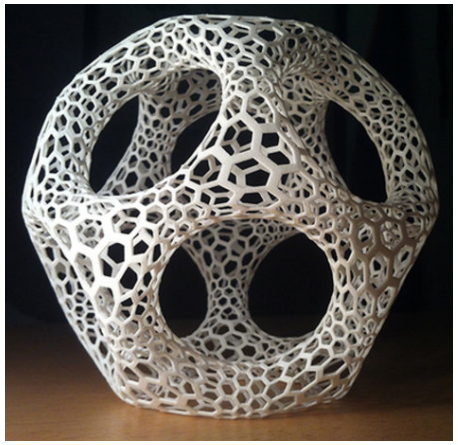
Digital Fabrication Technologies



Formlabs Form 2



MakerBot



Inspiring Structures - Shapeways



Nanoscribe



Pocket NC - CNC Milling



Laser Cutting

Boyer Machinery

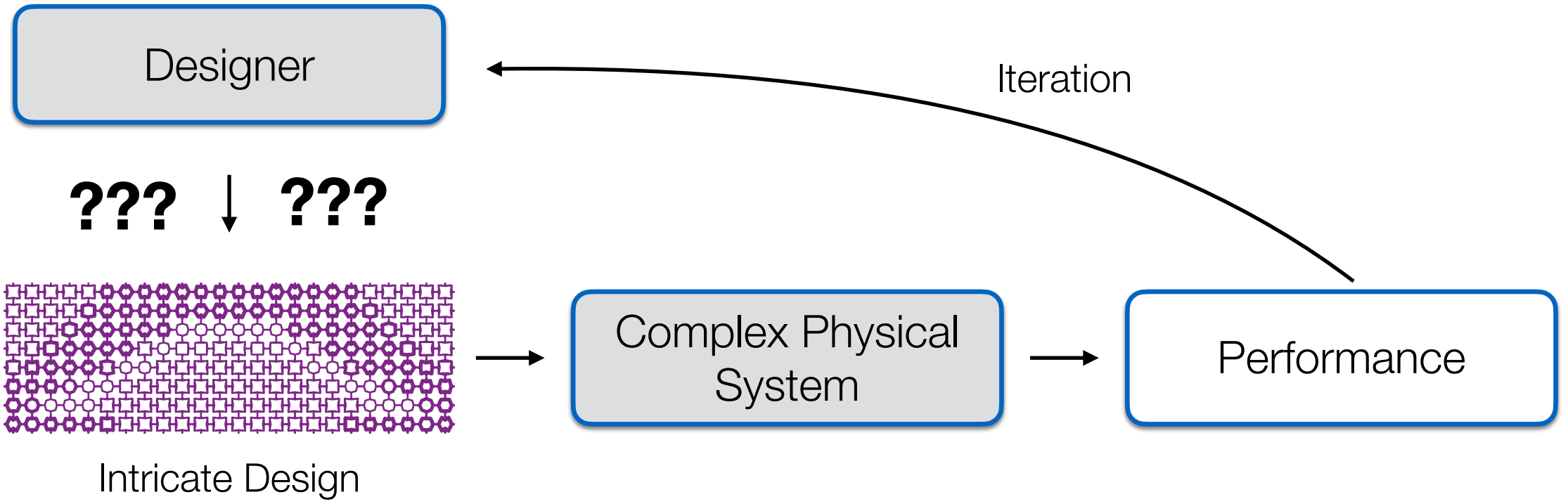


Beyond Developable
Laser Cut Auxetic Structure

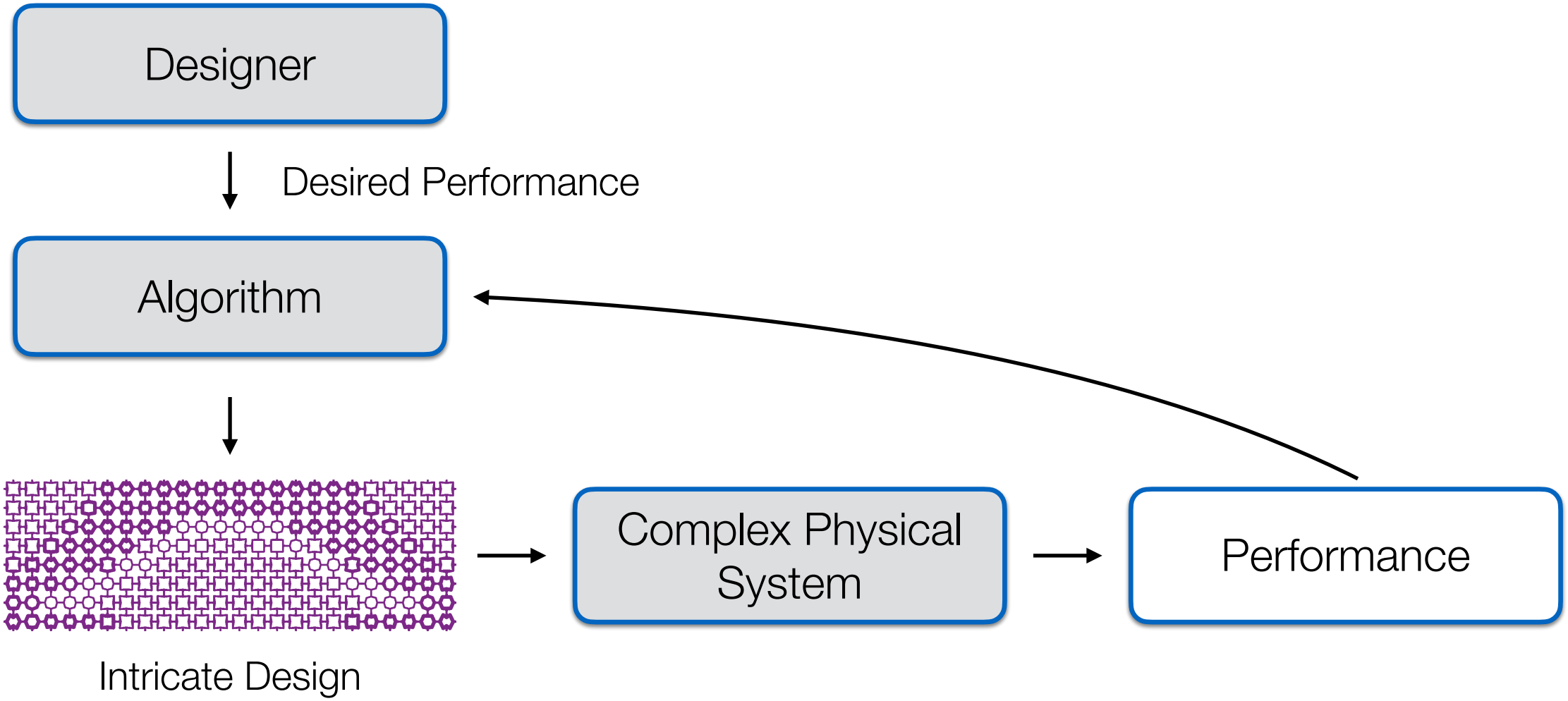


IDG Computerworld
Matthew Mikaelian

Human-Driven Design Approach

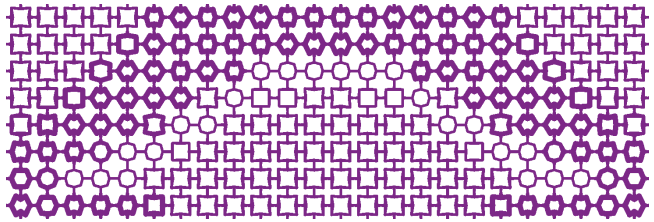


Computational Design Approach



Computational Design Approach

Algorithm Output



Optimal Design

Inverse Design Algorithm

User Input

Desired Performance

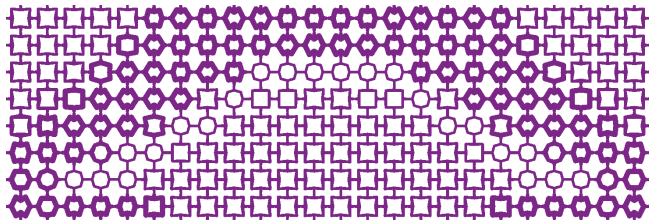
Intricate Design

Complex Physical System

Performance

Computational Design Approach

Algorithm Output



Optimal Design

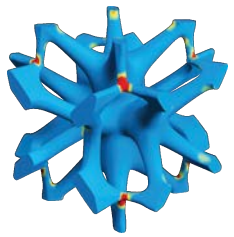
←

Inverse Design Algorithm

←

User Input

Desired Performance

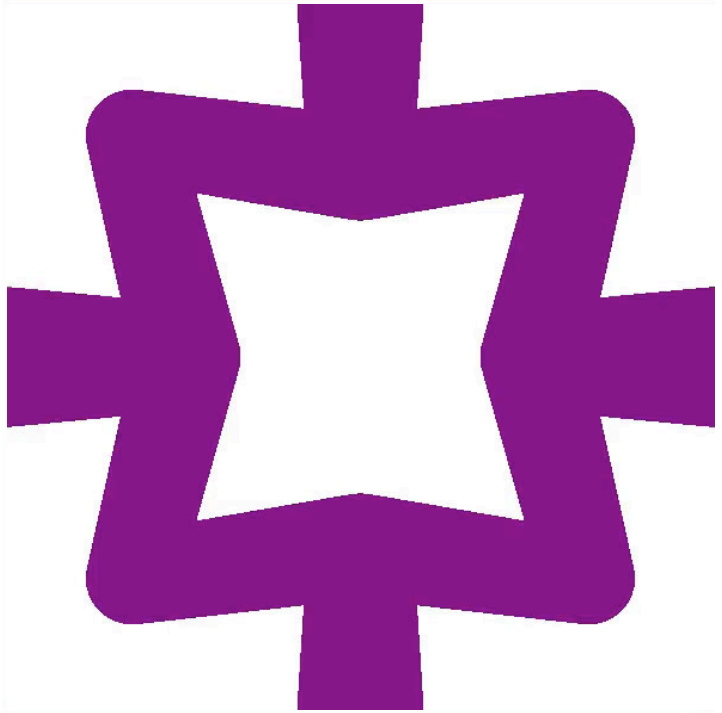


Elastic Metamaterials



Shape-morphing/Deployable Structures

Elastic Metamaterials



Tune material properties with **geometry**

Elastic Metamaterials

Properties impossible with naturally occurring materials

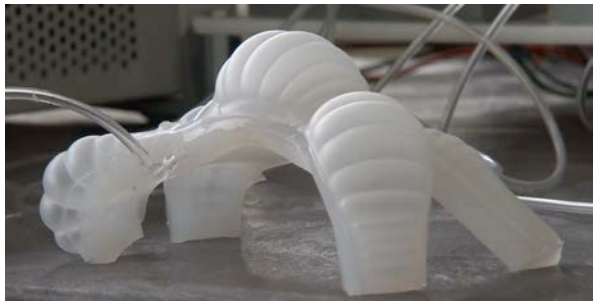
- Better strength-to-weight ratios
- Auxetic behaviors
- Precisely tuned properties

Manufacturable by 3D printers

Many promising applications



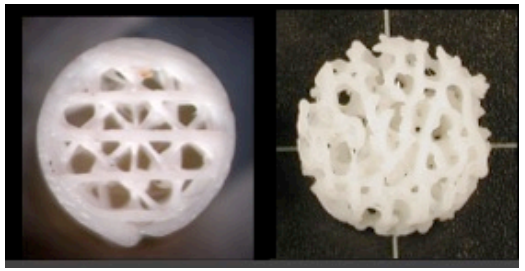
Adidas and Carbon3D



Soft Robots - NASA Langley



Lightweight Structures - [Aage et al 2017]



Bone Scaffolds - [Szivek et al. 2016]

Elastic Metamaterial Design



Microstructure Geometry

→ Homogenization →

Elastic Properties (Stiffness, …)

Elastic Metamaterial Design

Microstructure Geometry

→ Homogenization →

Elastic Properties (Stiffness, …)

Algorithm Output



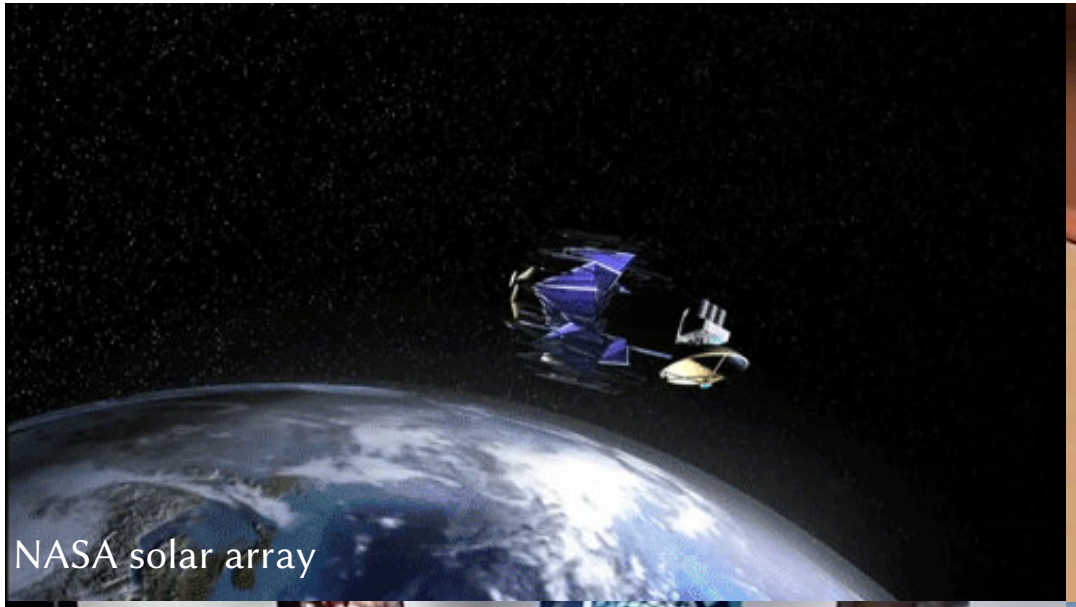
Optimal Geometry

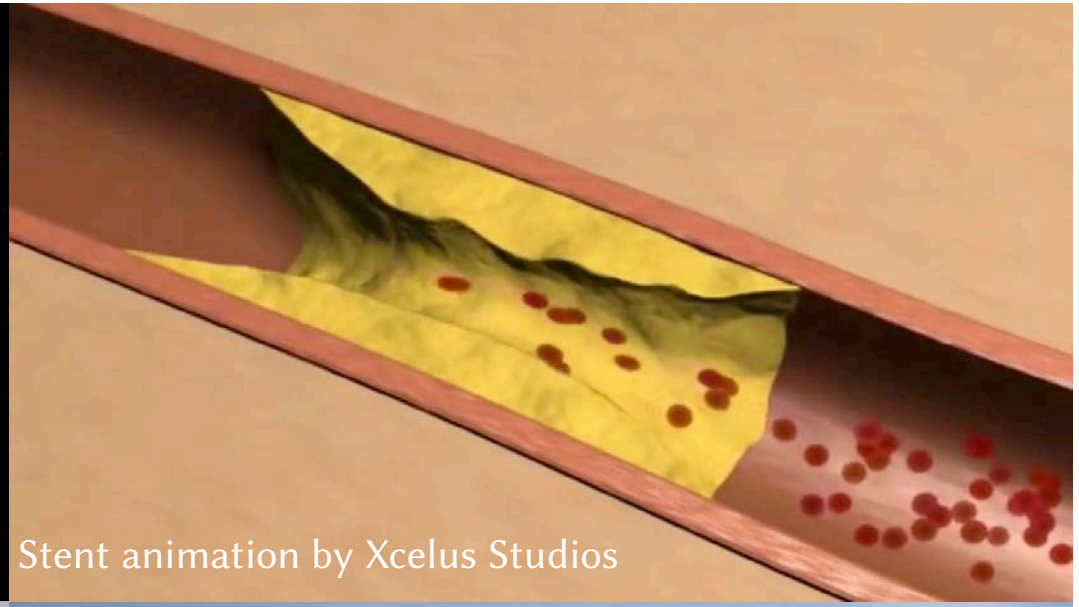
← Inverse Design Algorithm ←

User Input

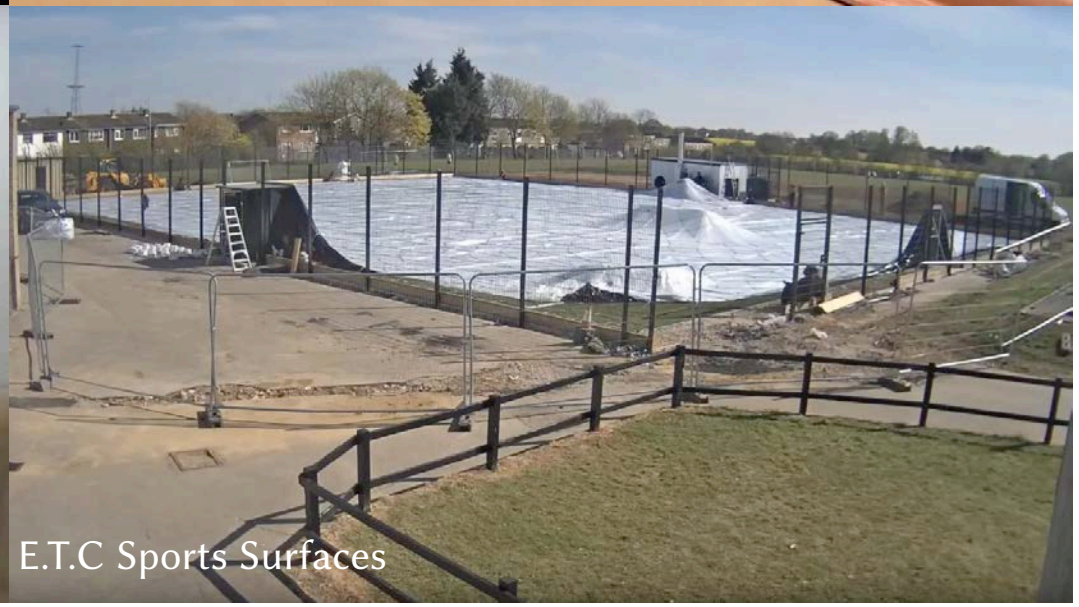
Target Properties

Deployable Structures

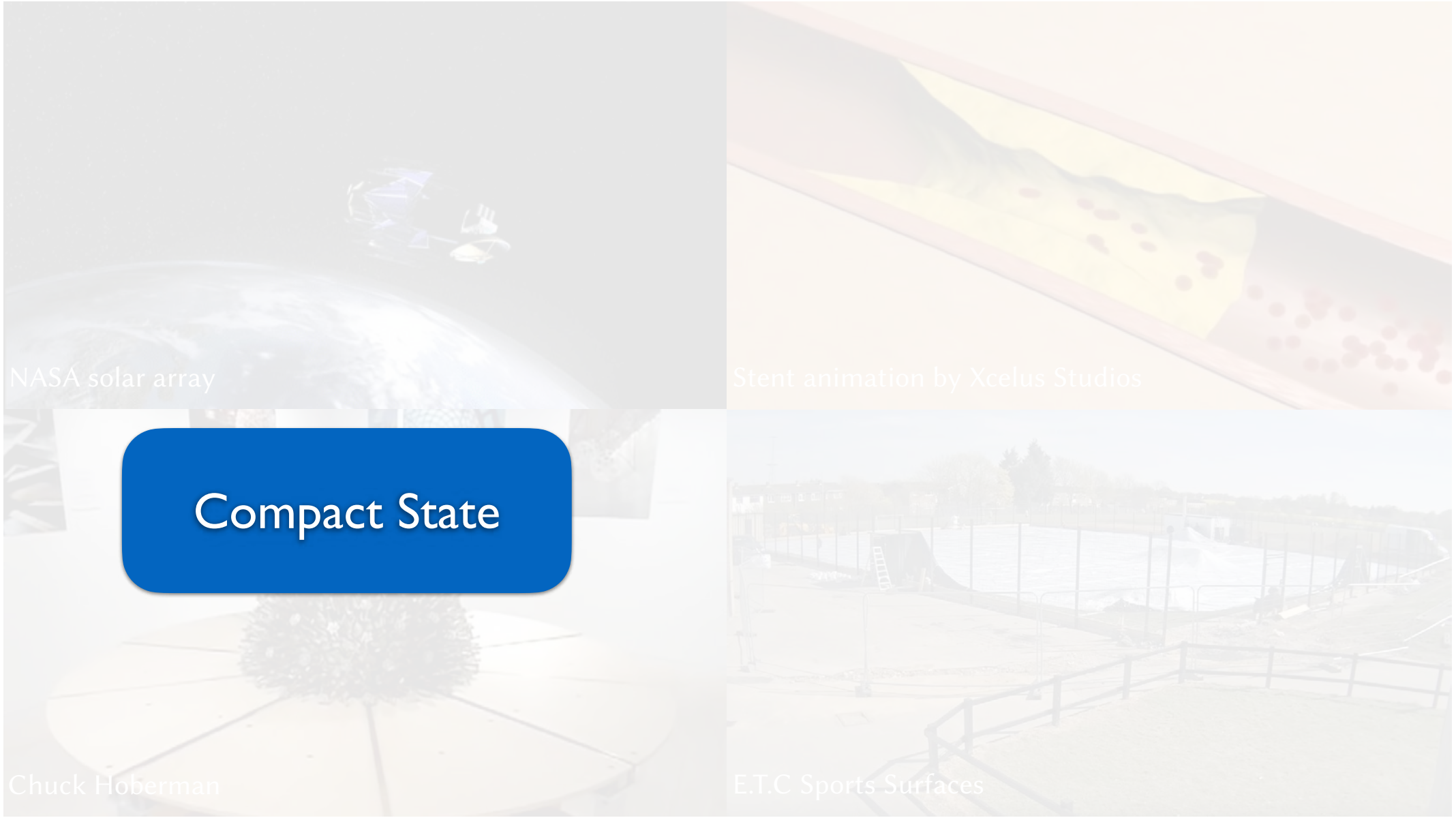

NASA solar array


Stent animation by Xcelus Studios


Chuck Hoberman


E.T.C Sports Surfaces

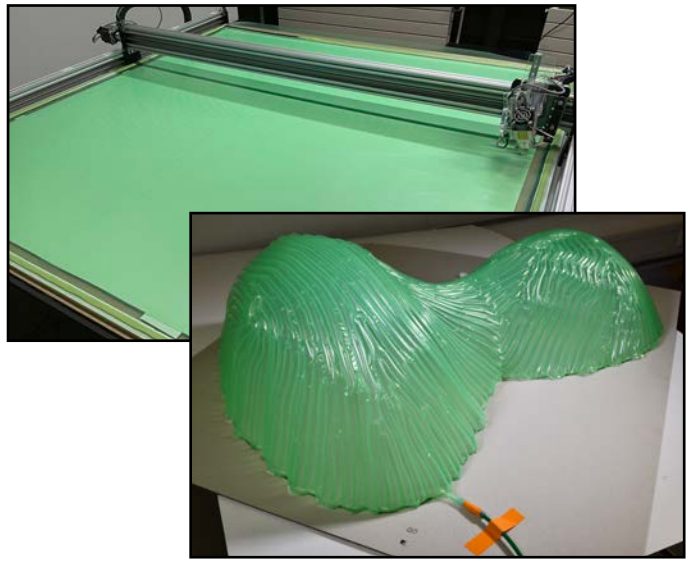
Deployable Structures

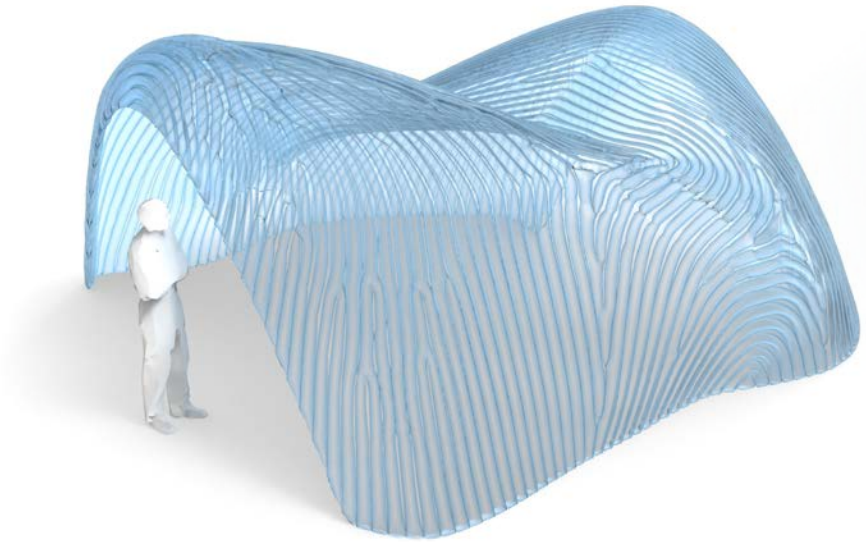


Example: Surface-Based Inflatables



BENEFITS OF SURFACE-BASED INFLATABLES







Programmable Auxetics
[Konaković-Luković et al. 2018]

Ease of Fabrication and Deployment

- Fuse membrane with a modified laser cutter
- Deploy and retract with an air pump

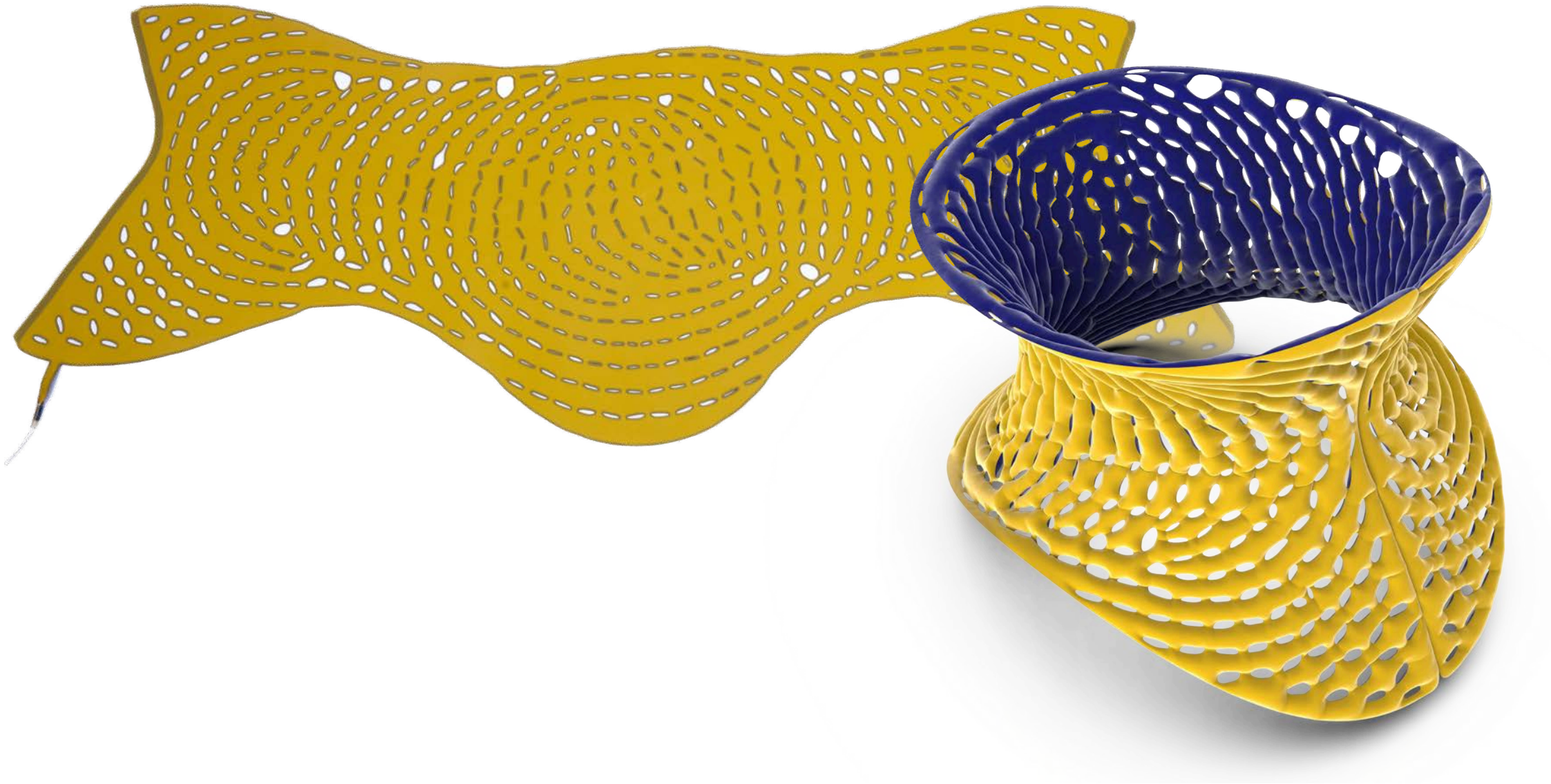
Structural Performance

- Stiff and lightweight
- Unpressurized interior

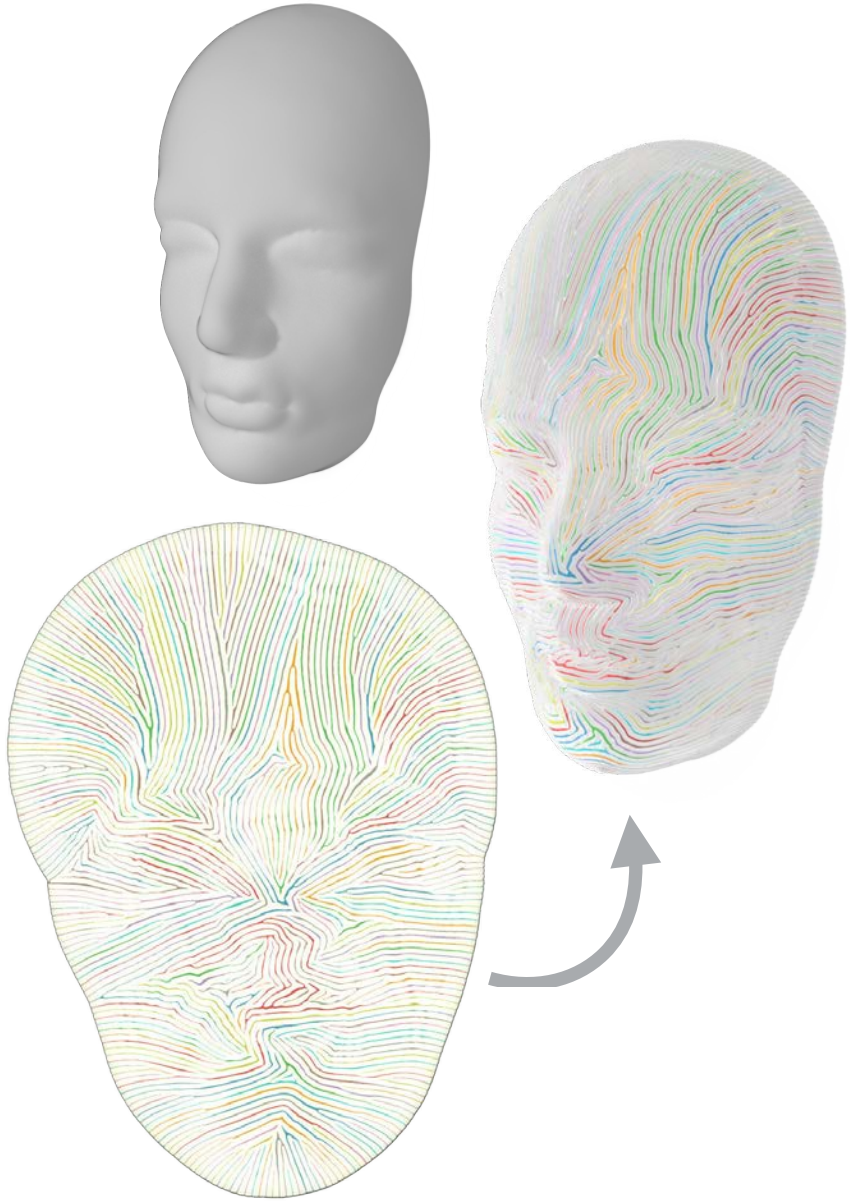
Rich Shape Space

- Negative mean curvature
- Openings

Healthcare Applications



Example: Surface-Based Inflatables



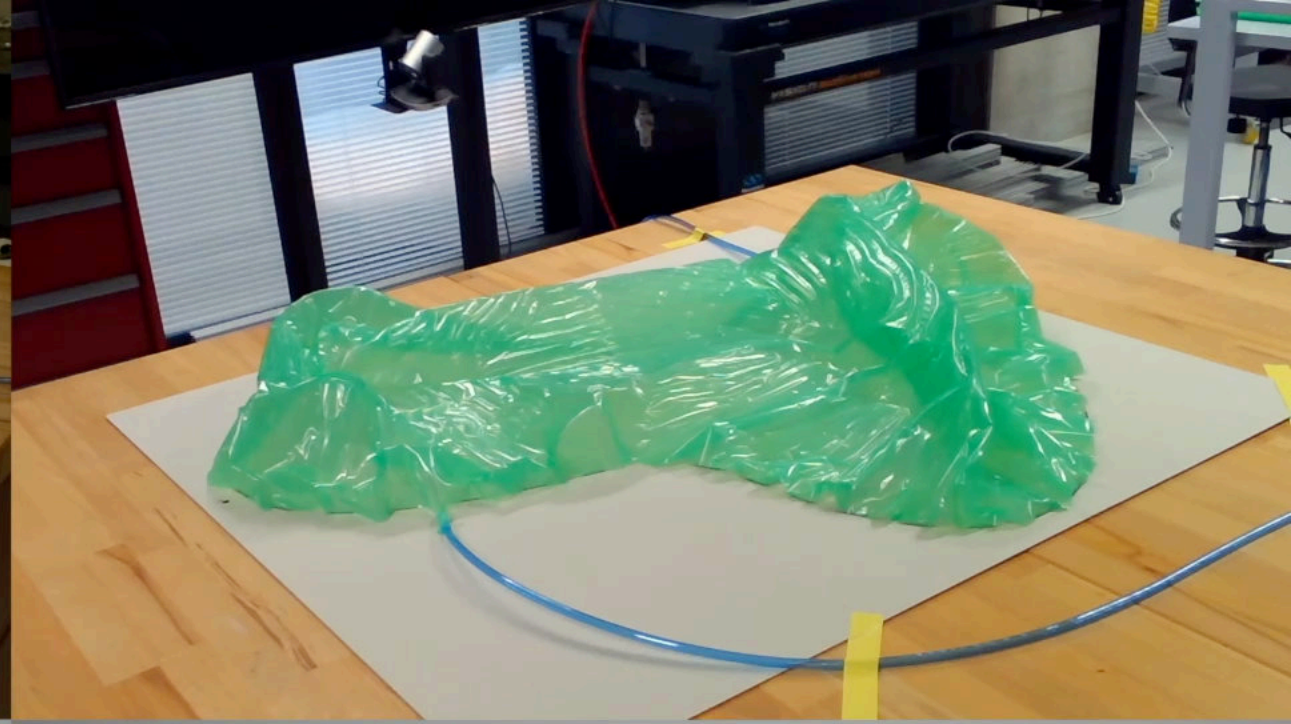


FABRICATION SETUP



FABRICATED EXAMPLES

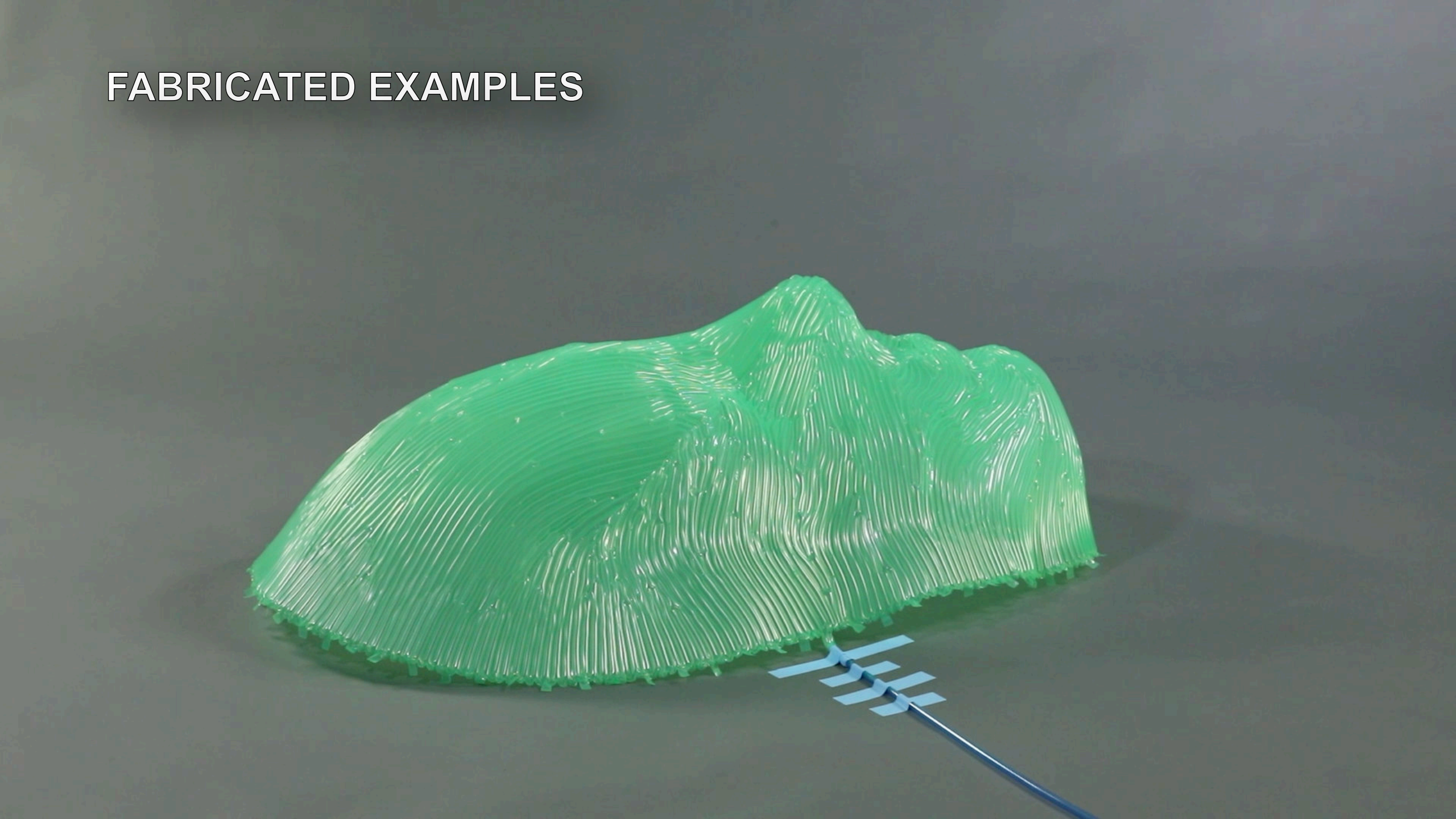




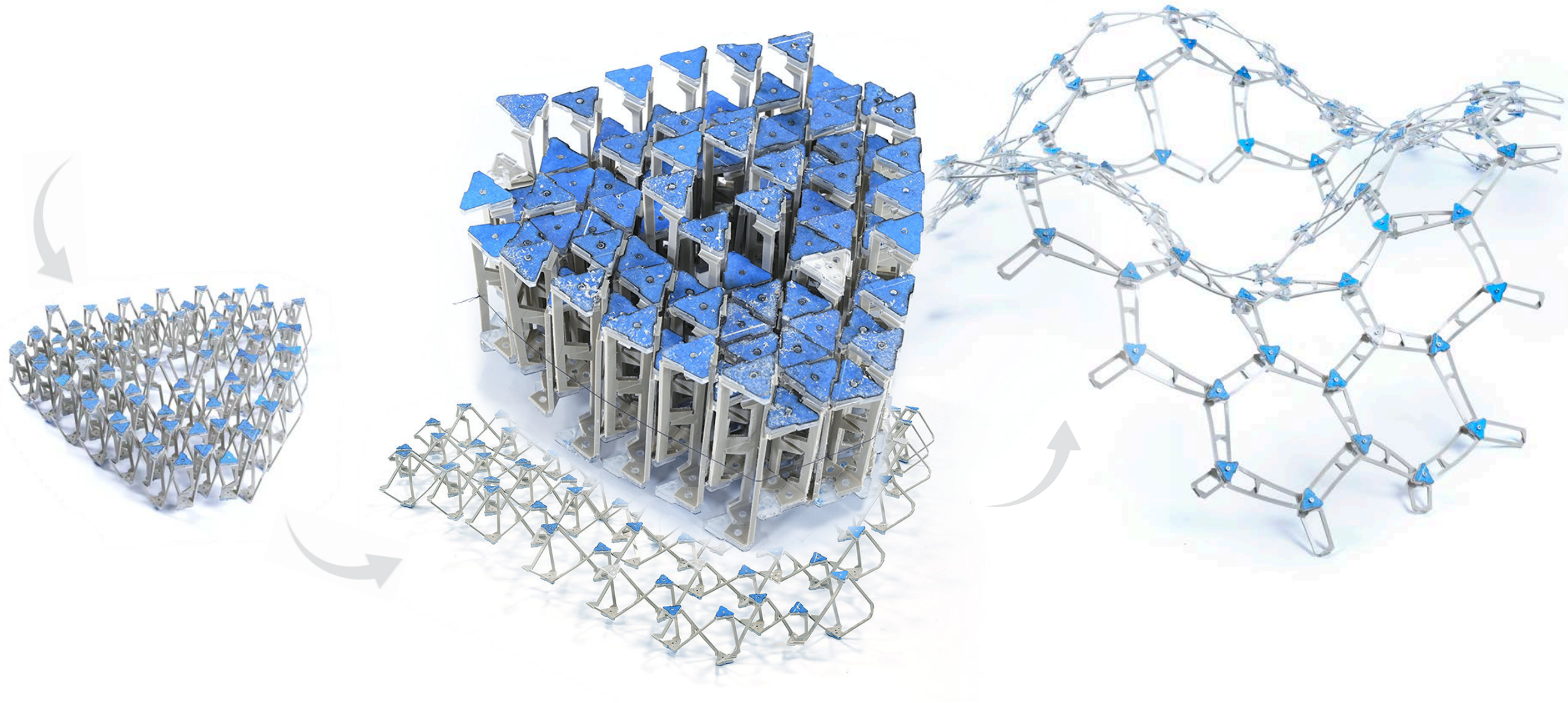




FABRICATED EXAMPLES

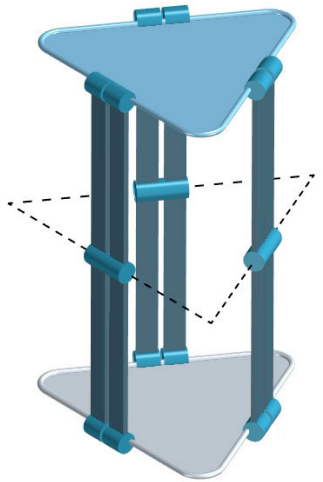


Example: Umbrella Meshes

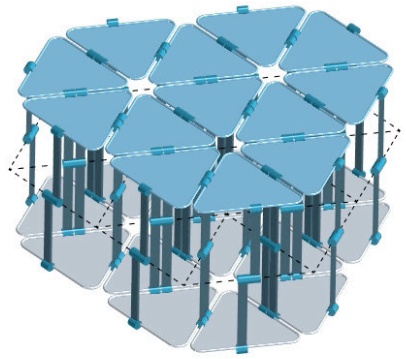


Unit Cell Mechanism

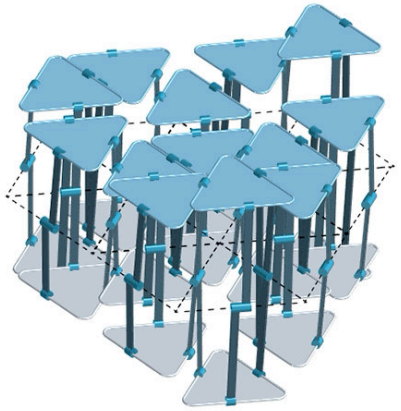




Regular Tessellation with Equal Heights

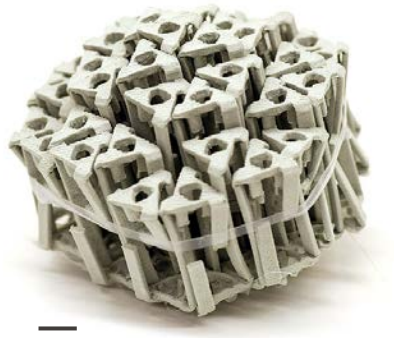


Regular Tessellation with Different Heights



The umbrella mesh elastically deforms into a curved equilibrium

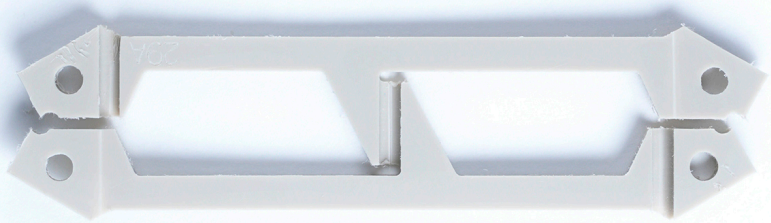
Fabrication: 3D Printing



10mm



Fabrication: CNC Milling





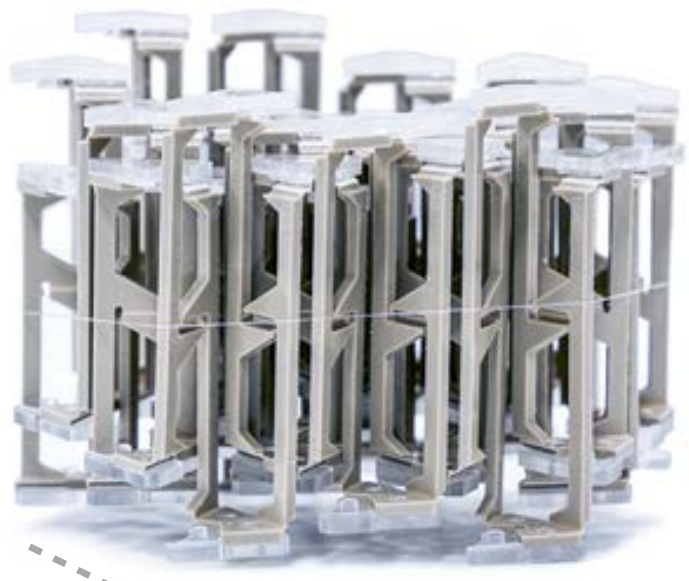




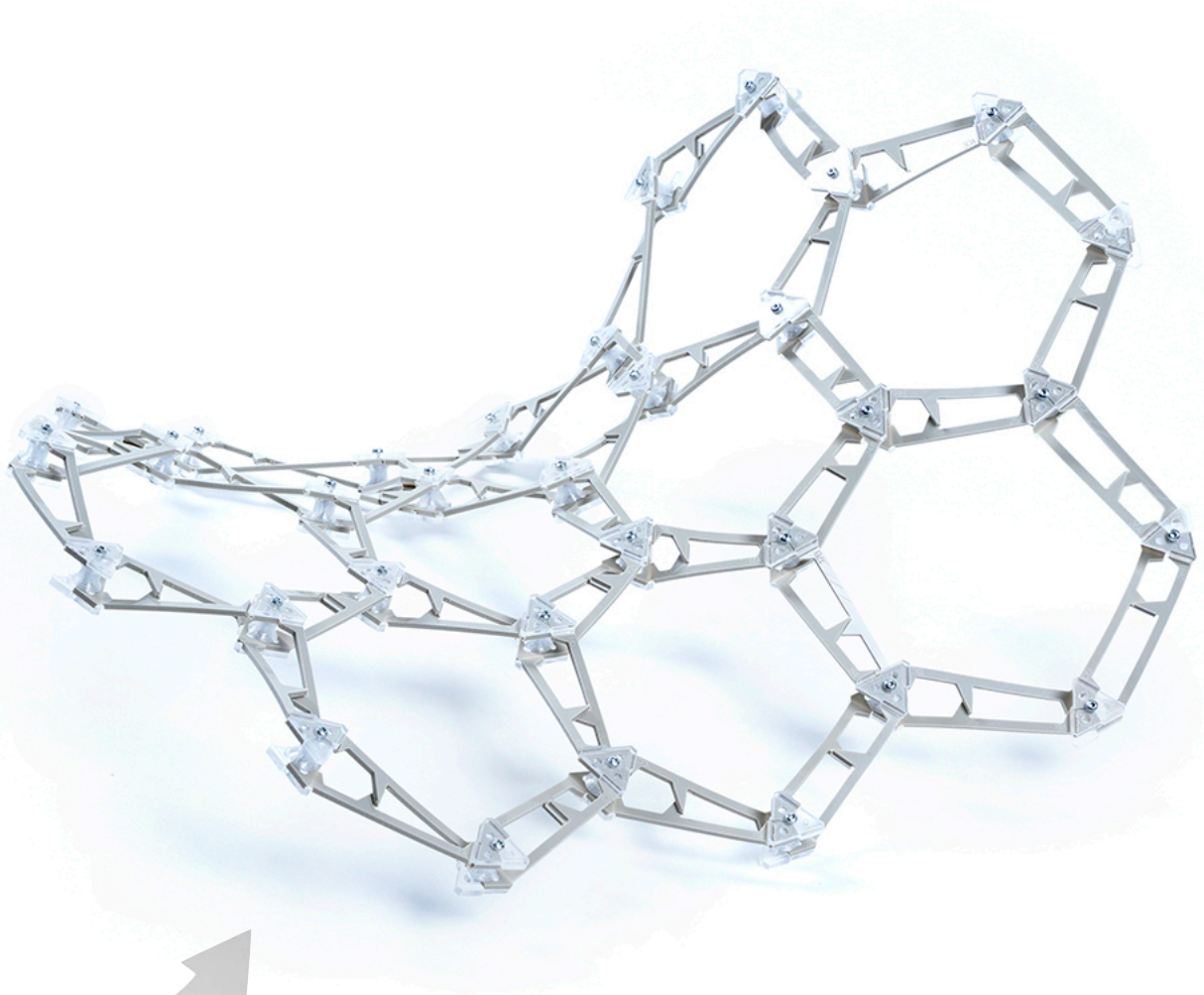




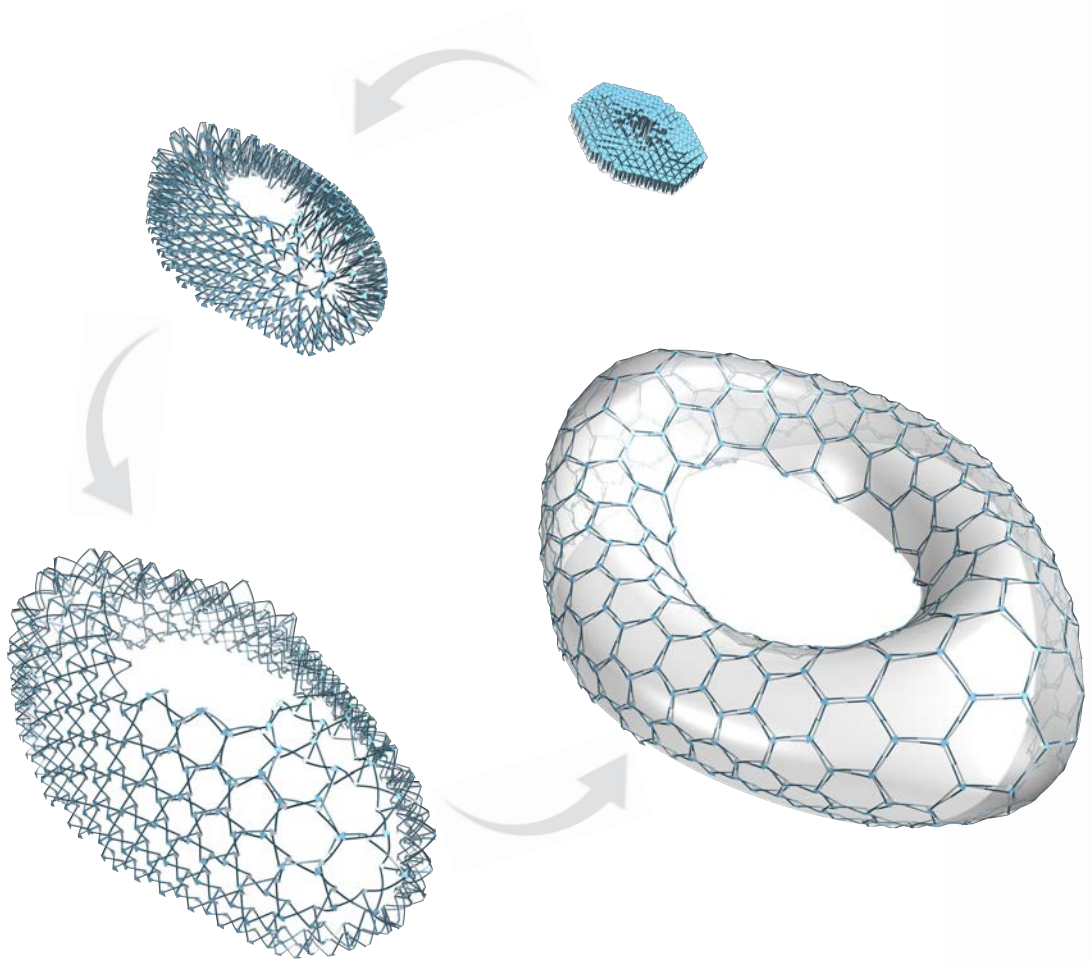
Physical Prototype

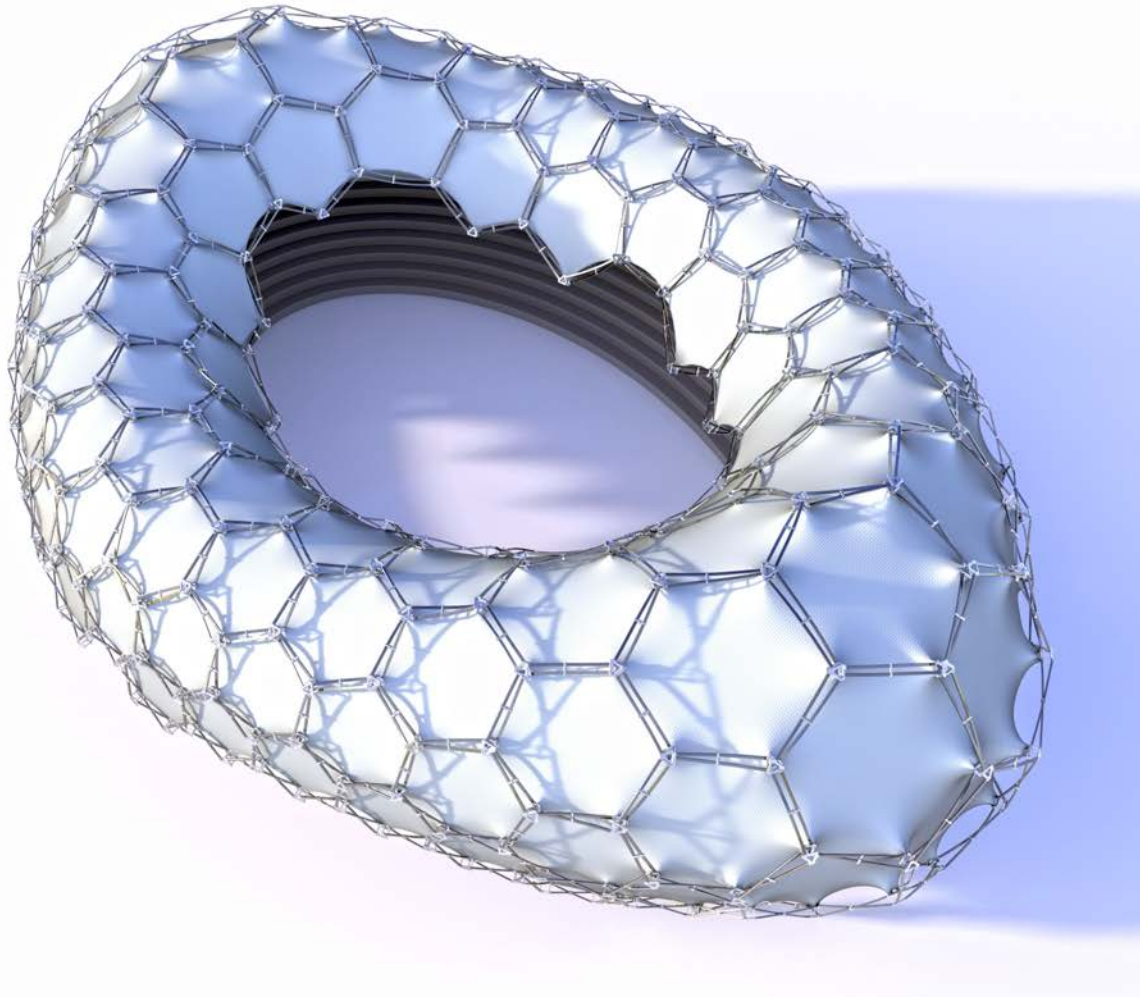




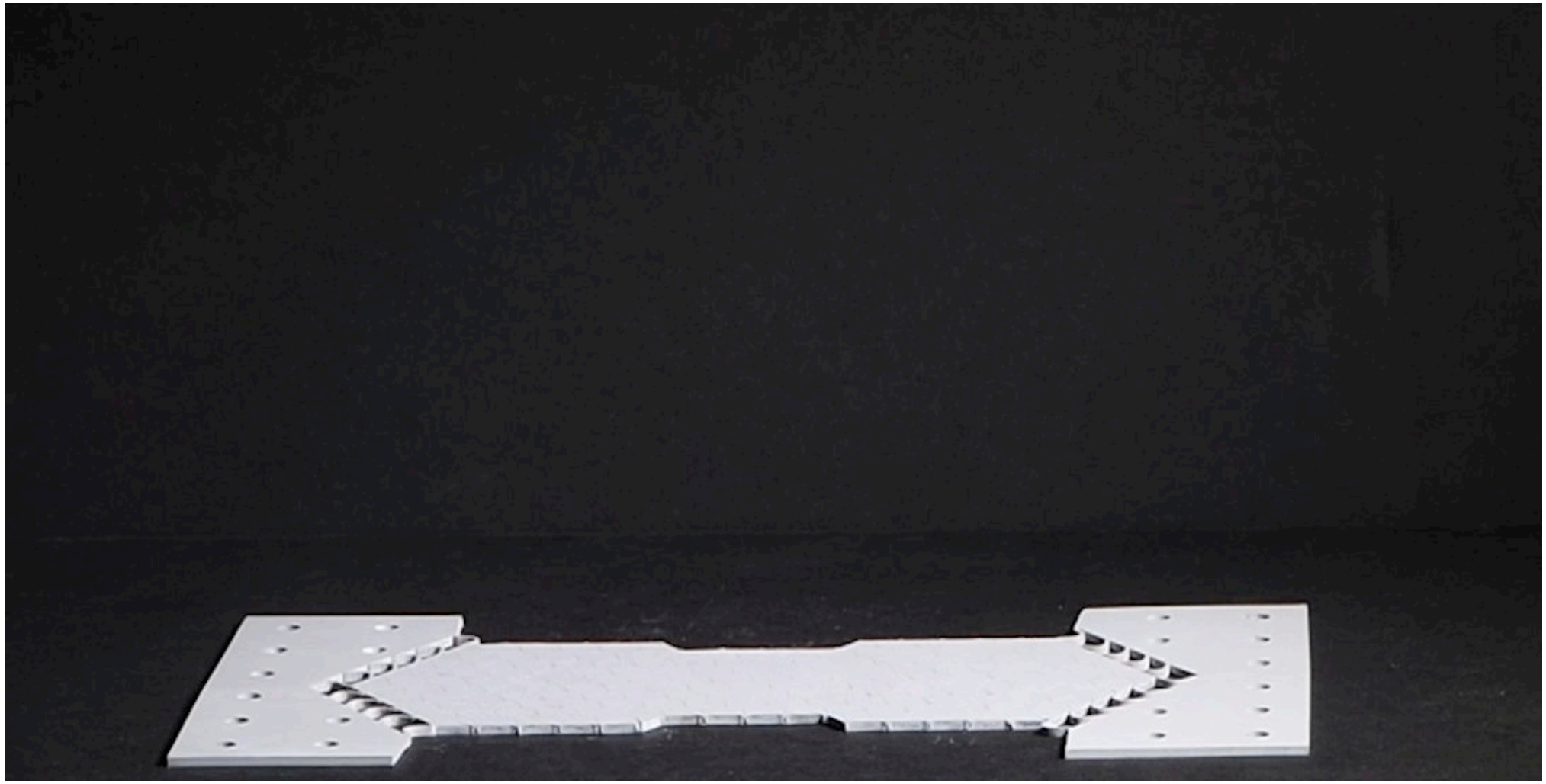


Architectural Example

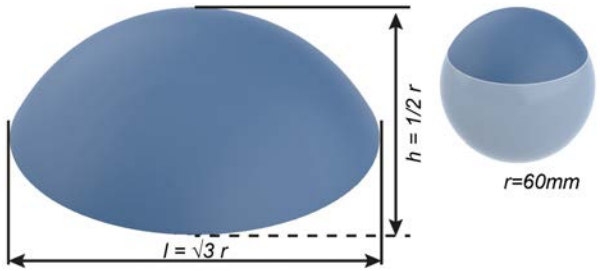




Example: Bistable Auxetic Surface Structures

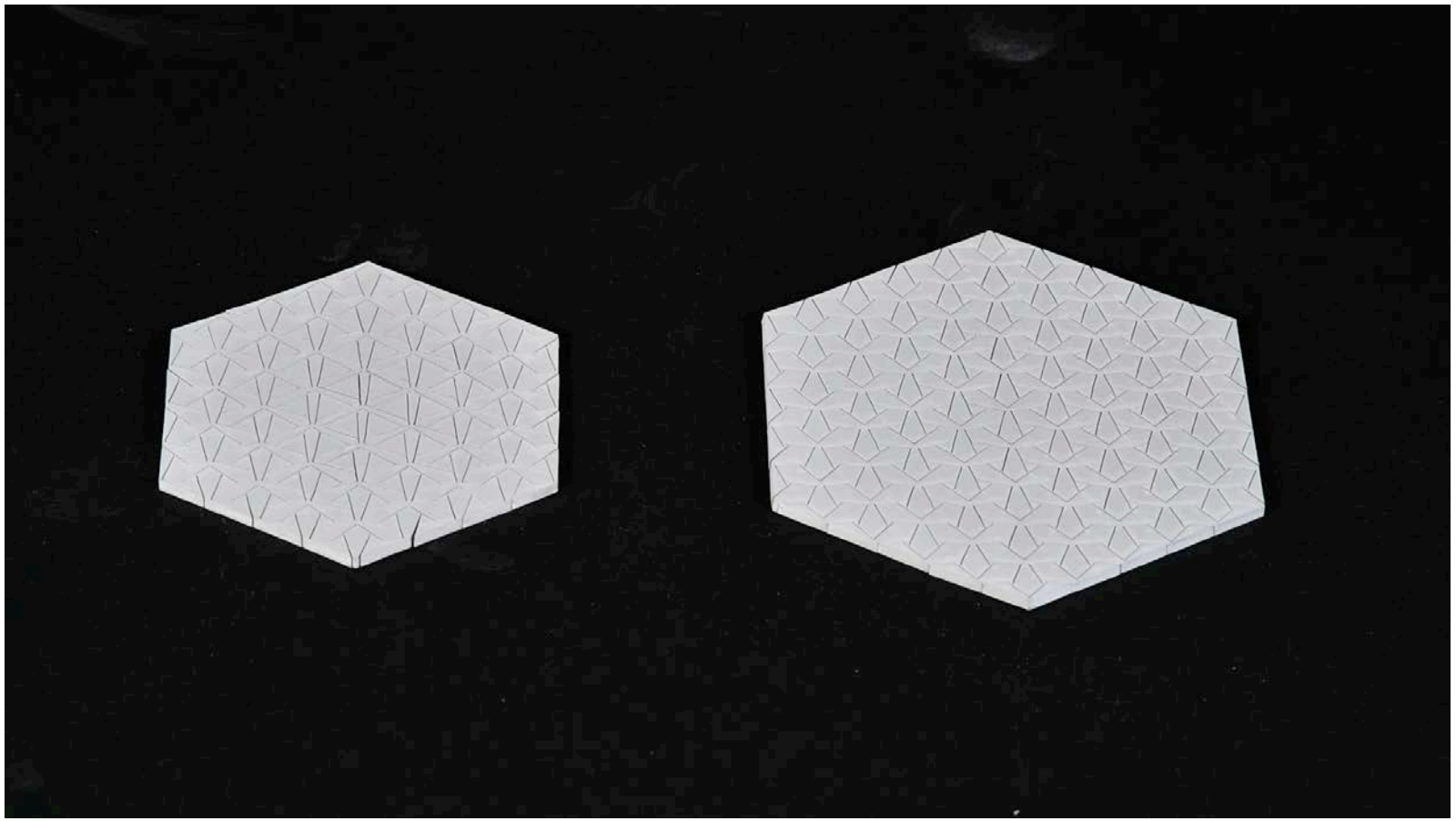


Spherical Cap Examples

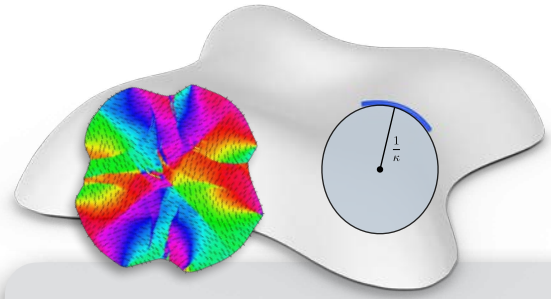








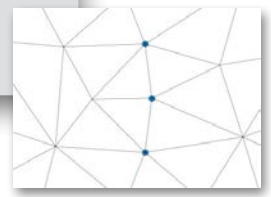
Optimal Design Pipeline For Deployable Structures



Homogenization-Based Initialization

Determine design **topology** and reasonable initial shape parameters

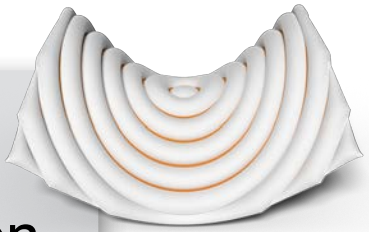
Shape Optimization



Design Candidate

Performance, Derivatives

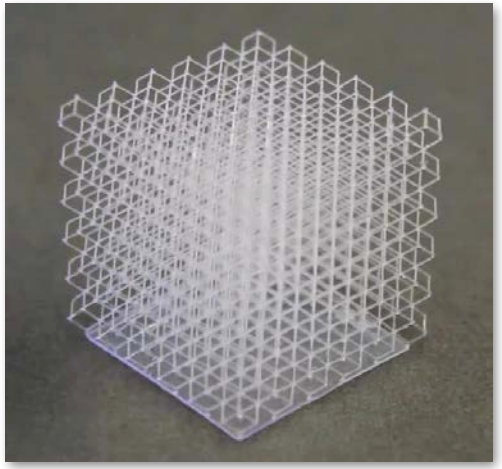
Differentiable Physics Simulation

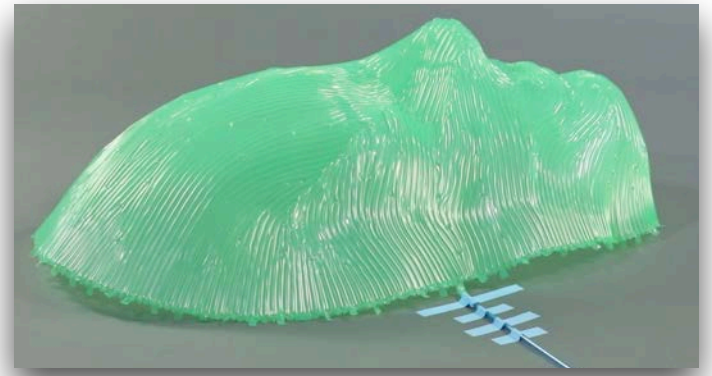


Fabrication-Ready Design

Why Homogenization?

- Physics-based design of the full object at achievable resolutions is not tractable. (Two-scale design)
- Need a good starting point for gradient-based design.
- Design general-purpose, reusable **metamaterial catalogs**





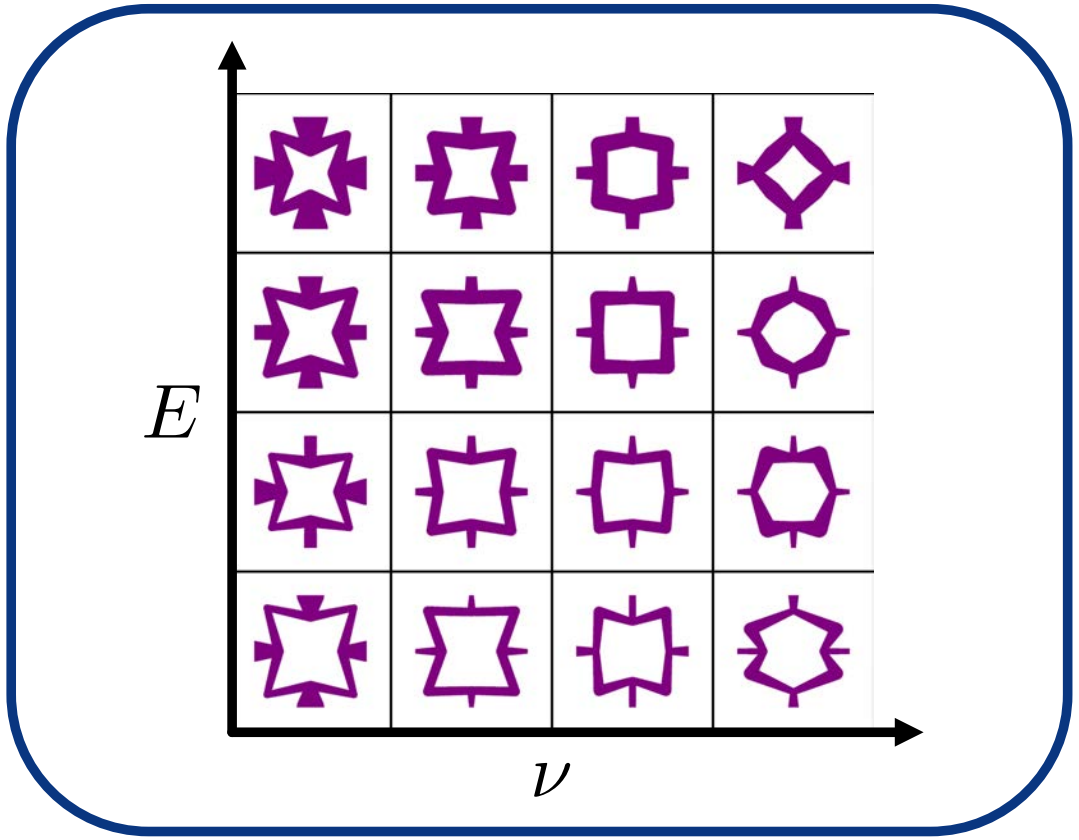


Formlabs



nTop

Two-Scale Design Pipeline

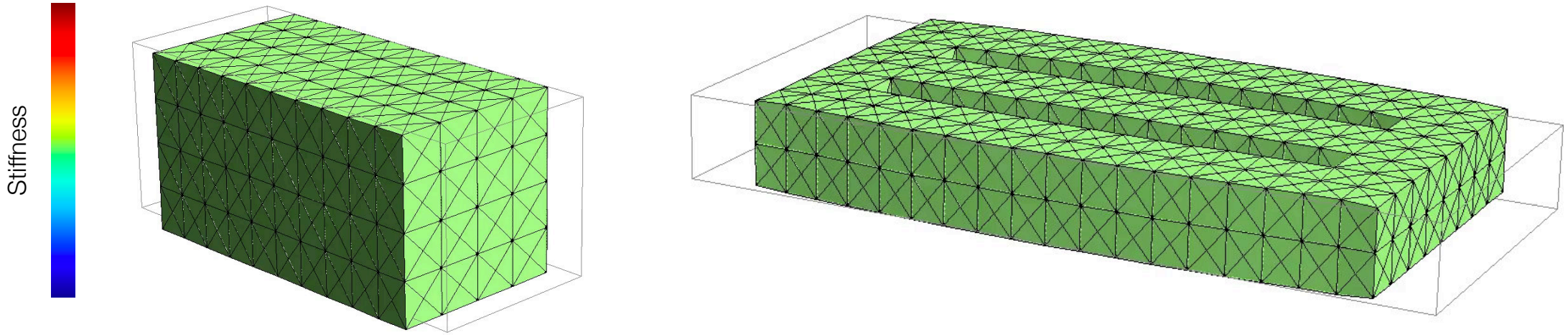


Precomputed Catalog

Database Lookup

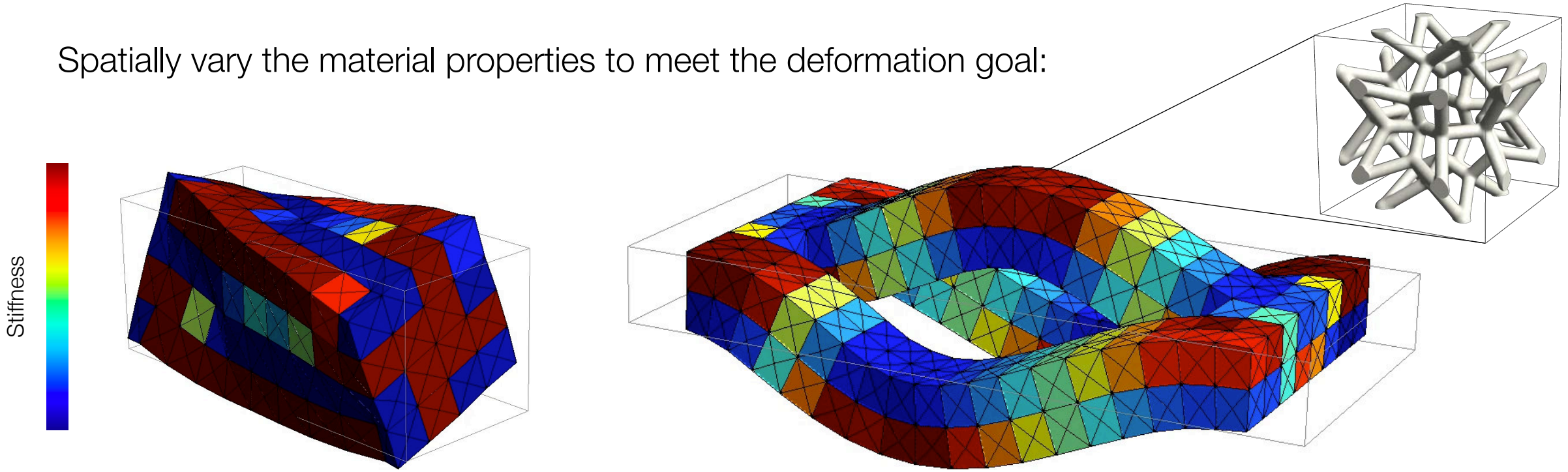
Goal: Customize Deformation Behavior

Spatially vary the material properties to meet the deformation goal:



Goal: Customize Deformation Behavior

Spatially vary the material properties to meet the deformation goal:



Computational Homogenization and Inverse Design for Mechanical Metamaterial Systems

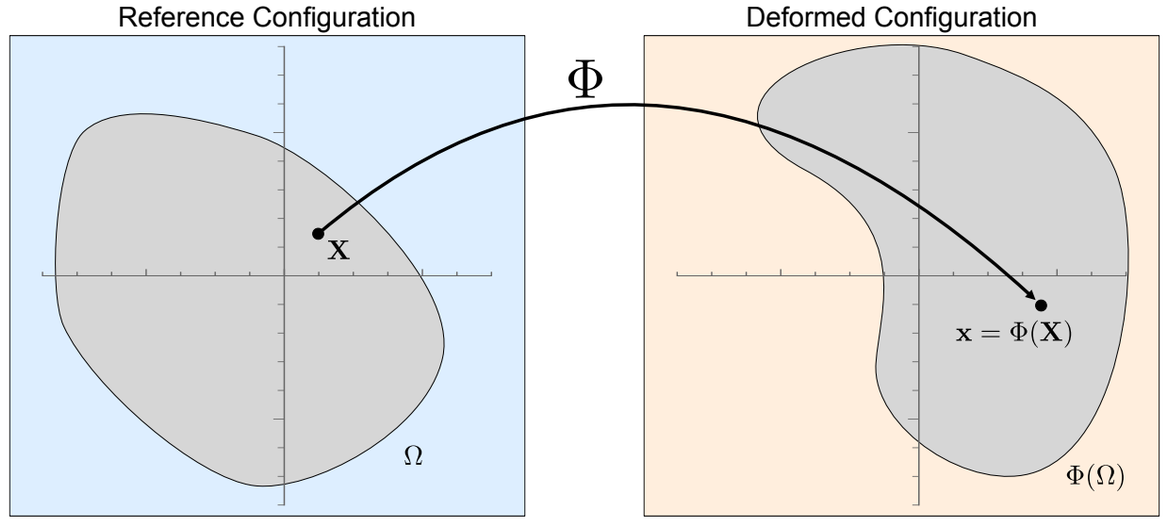
Boulder School 2026

Julian Panetta

University of California, Davis

These slides are best accessed through the HTML version:
https://julianpanetta.com/teaching/BoulderSchool2026/technical_slides/technical-slides-deck.html

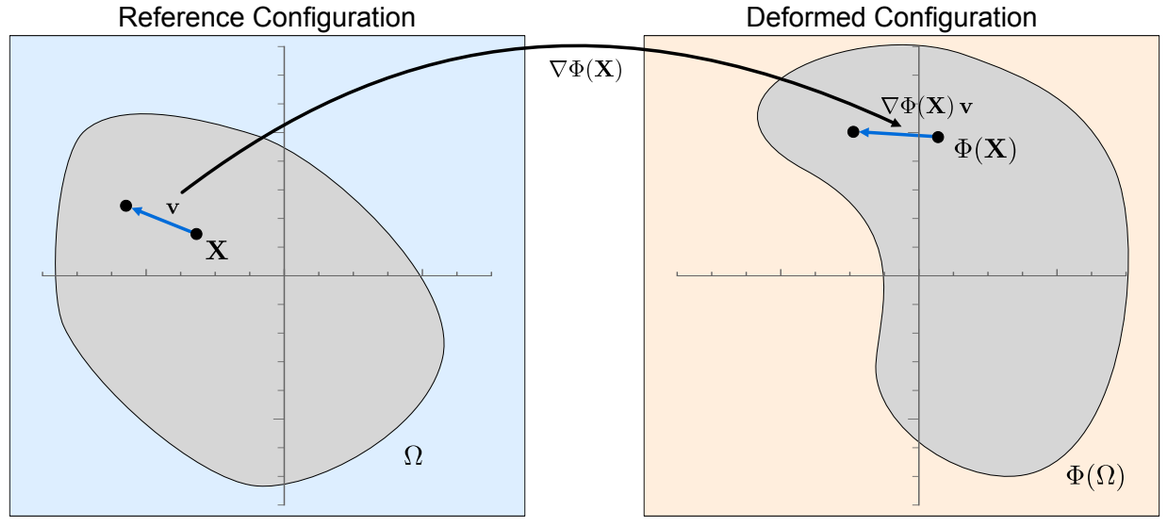
Elasticity Review: Nonlinear Elasticity



- Professor James gave an introduction to nonlinear elasticity in the first week; we'll be using the same model, with just a few notational changes.
- Object is $\Omega \in \mathbb{R}^d$ ($d = 2$ or 3) in the reference configuration.
- Points within Ω are denoted by $\mathbf{X}$.
- Deformation map: $\Phi : \Omega \to \mathbb{R}^d$

We'll see many different elastic objects and generally use the same Ω notation for all.

Elasticity Review: Nonlinear Elasticity



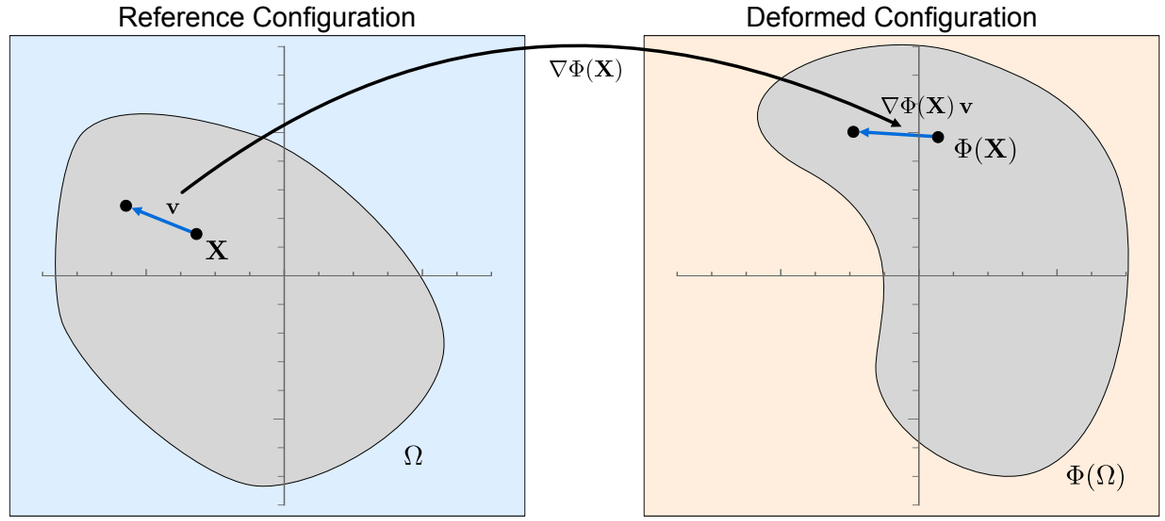
- Local distortion is captured by $F := \nabla\Phi(\mathbf{X})$, the *deformation gradient* (*Jacobian*) of Φ.

$$F_{ij} = \frac{\partial \Phi_i}{\partial X_j} \quad \Longrightarrow \quad F\mathbf{v} = \sum_j \frac{\partial \Phi}{\partial \mathbf{X}_j} v_j$$

- Elastic energy per unit volume can be formulated as an *energy density* function $\Psi(F)$.
  - $\Psi : \mathbb{R}^{d\times d} \to \mathbb{R}$ encodes the material's elastic properties.
  - Our notation will suppress the dependence of Ψ on $\mathbf{X}$ unless necessary.
  - We'll also assume material is temperature-independent.

$$\Psi(QF) = \Psi(F) \quad \forall Q \in SO(d)$$

Elasticity Review: Nonlinear Elasticity



- The full elastic energy in the object is the integral:

$$\boldsymbol{E}_{\text{elastic}}[\boldsymbol{\Phi}] := \int_{\boldsymbol{\Omega}} \boldsymbol{\Psi}(\boldsymbol{\nabla\Phi})\,\mathbf{d}\boldsymbol{\mathbf{X}}.$$

- Simulation is formulated as minimization of a total potential:

$$\min_{\boldsymbol{\Phi}\text{ admissible}} \boldsymbol{E}_{\text{elastic}}[\boldsymbol{\Phi}] + \boldsymbol{\mathcal{L}}[\boldsymbol{\Phi}],$$

where $\boldsymbol{\mathcal{L}}[\boldsymbol{\Phi}]$ is a potential modeling external forces.

Minimization is done over *admissible* deformations satisfying essential boundary conditions.

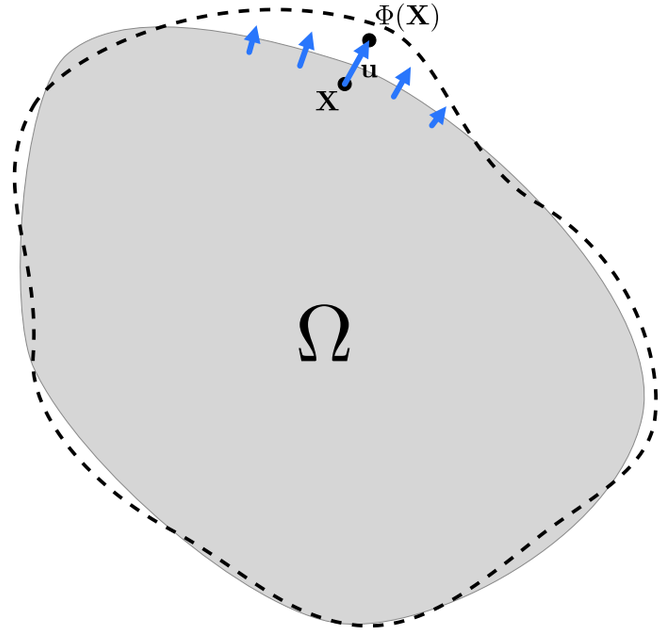
Elasticity Review: Linear Elasticity

- For small displacements $\mathbf{u}$ ($\Phi(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X})$), $E[\Phi]$ can be *linearized* to obtain the linear elasticity strain energy:

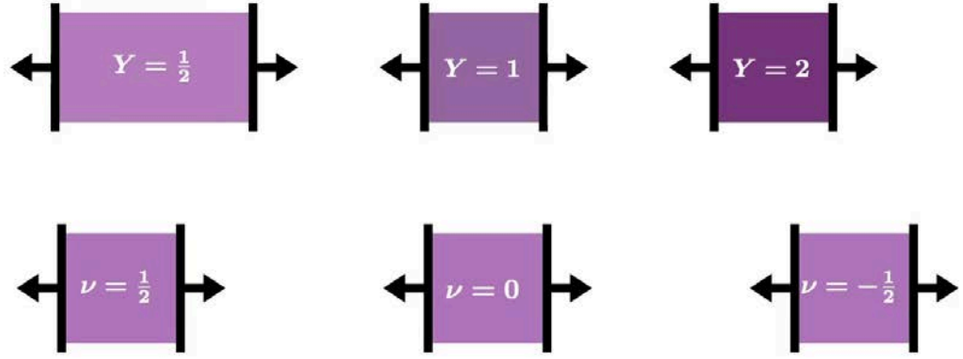
$$\varepsilon_{\text{lin}}(\mathbf{u}) := \text{sym}\,(\nabla \mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top\right)$$

$$\psi_{\text{lin}}(\mathbf{u}) := \frac{1}{2}\varepsilon_{\text{lin}}(\mathbf{u}) : C : \varepsilon_{\text{lin}}(\mathbf{u}),$$

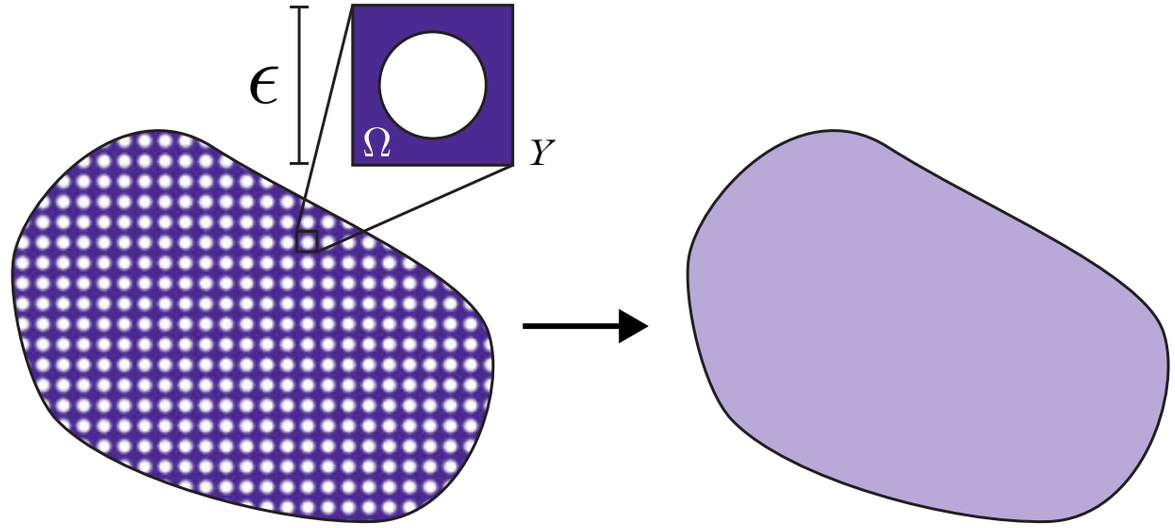
$$\sigma_{\text{lin}} = C : \varepsilon_{\text{lin}},$$



- $C = \psi''(I)$ is the fourth-order *elasticity tensor* encoding the material's linear elastic properties.
- For isotropic materials, C has just two independent parameters *e.g.*, the Young's modulus E and Poisson's ratio ν.



Periodic Homogenization for Nonlinear Elasticity

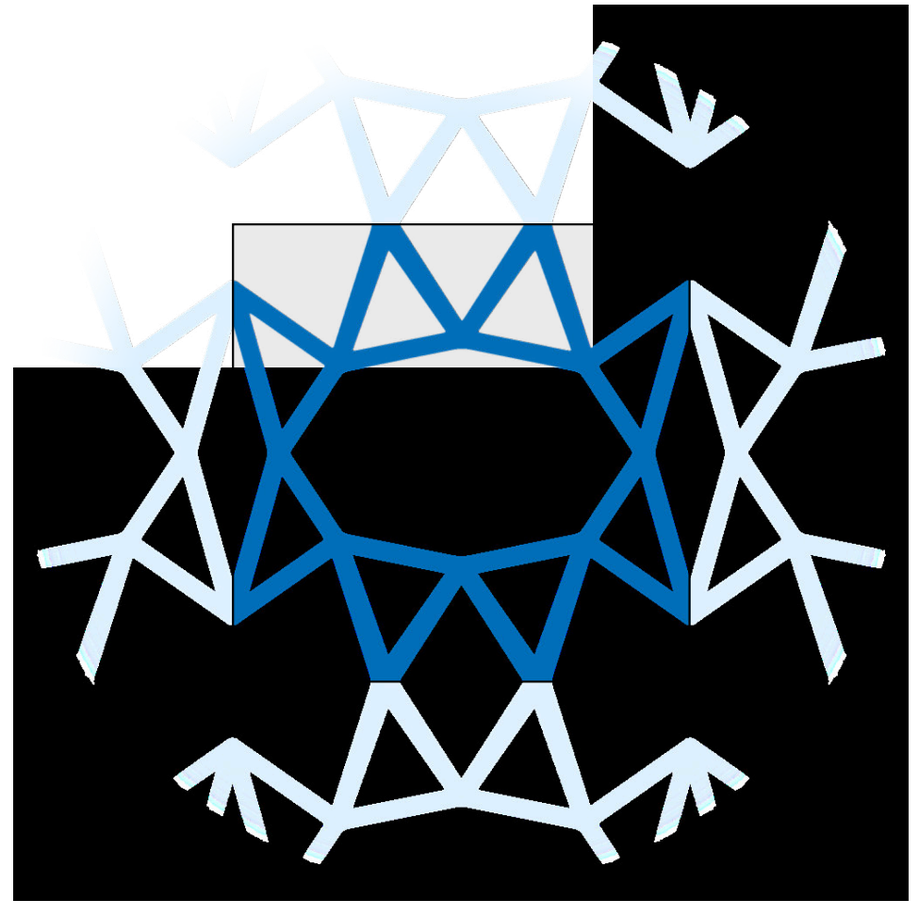


- Basic idea: as the number of repetitions of a microstructure becomes sufficiently large, the collective behavior approaches that of a continuum material.
- Goal: determine properties of that material from microstructure geometry.
- Formally: evaluate the *homogenized energy density* $\overline{\psi}(\overline{F})$.
- This is done by solving a *cell problem* on a single "representative volume element" (RVE) (often one unit cell Y).

$\overline{F}$ is the *macroscopic* or *average* deformation gradient.

Heuristic Derivation of the Cell Problem

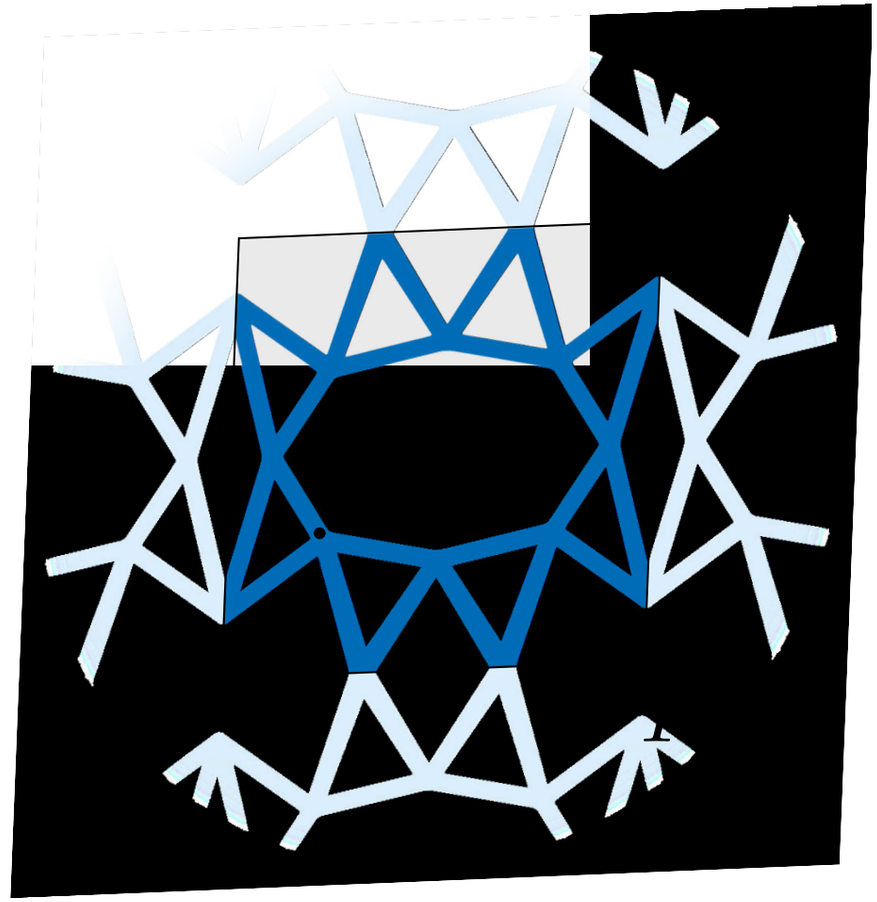
- Warning: not intended as a rigorous derivation; just for intuition!
- High-level strategy:
  - Construct an infinite periodic tiling.



Heuristic Derivation of the Cell Problem

- Warning: not intended as a rigorous derivation; just for intuition!
- High-level strategy:
  - Construct an infinite periodic tiling.
  - Distort it by the macroscopic deformation gradient.

$$\mathbf{x} \to \overline{F}\mathbf{x}$$

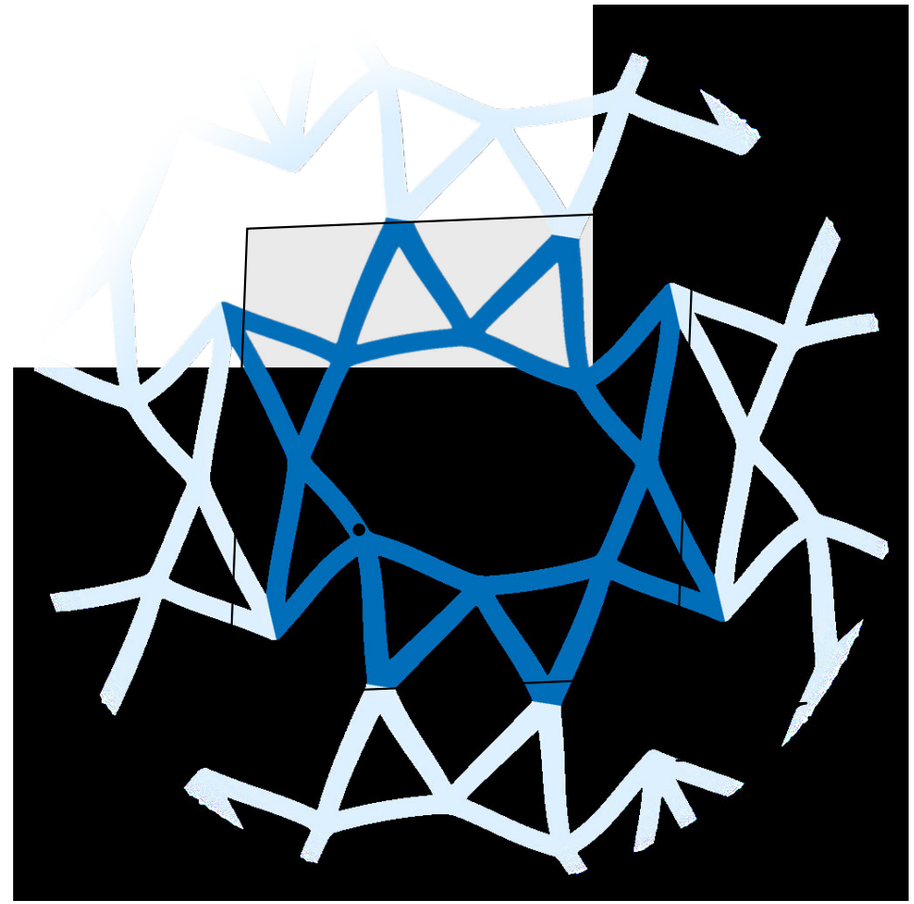


Heuristic Derivation of the Cell Problem

- Warning: not intended as a rigorous derivation; just for intuition!
- High-level strategy:
  - Construct an infinite periodic tiling.
  - Distort it by the macroscopic deformation gradient.

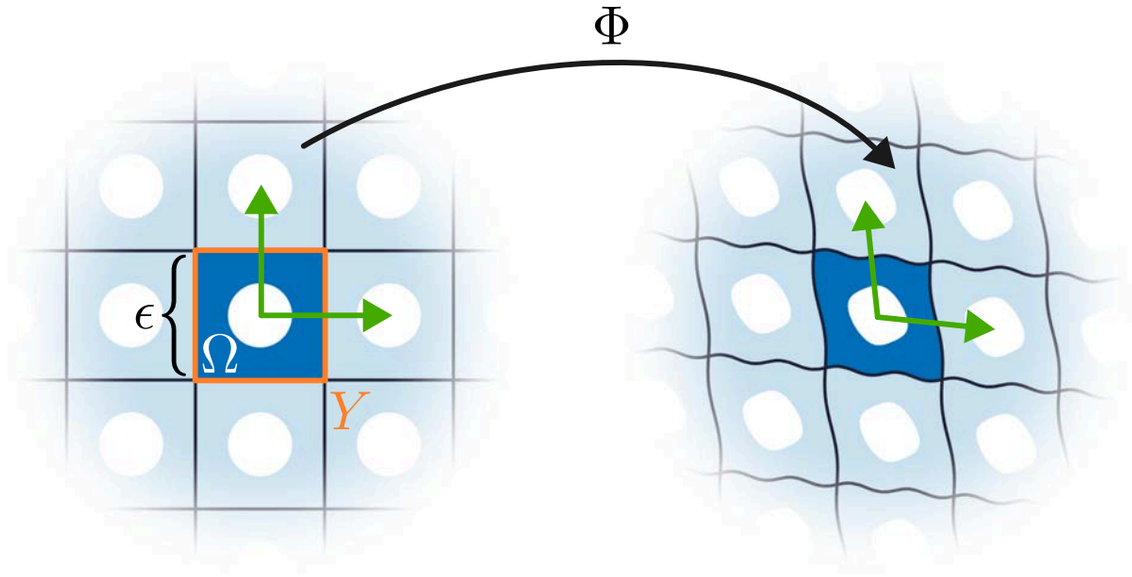
$$\mathbf{X} \to \overline{F}\mathbf{X}$$

  - Determine how the microstructure relaxes to minimize $E[\Phi]$.



Heuristic Derivation of the Cell Problem

Fundamental simplifying assumption:
Φ preserves the lattice translational symmetry.



- Undeformed grid is generated by repeating cell Y at translations $\epsilon\mathbf{c}$ for $\mathbf{c} \in \mathbb{Z}^d$.
- We assume deformed grid repeats $\Phi(Y)$ at translations $\overline{F}(\epsilon\mathbf{c})$.
- Formally: $\Phi\left(\mathbf{X} + \epsilon\mathbf{c}\right) = \Phi(\mathbf{X}) + \epsilon\overline{F}\mathbf{c} \qquad \forall \mathbf{X} \in Y, \mathbf{c} \in \mathbb{Z}^d$

Heuristic Derivation of the Cell Problem

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

- To understand how this assumption restricts admissible deformations, we introduce a convenient change of variables:

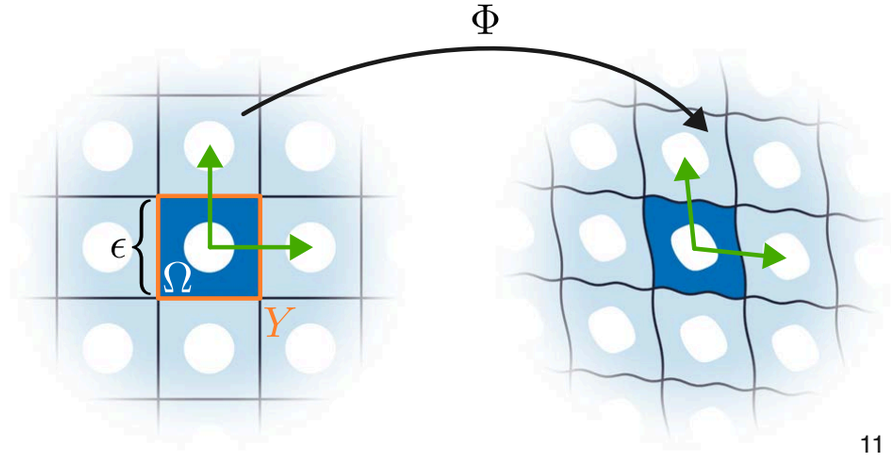
$$\boxed{\boldsymbol{\Phi}(\mathbf{X})=\overline{F}\mathbf{X}+\boldsymbol{\omega}(\mathbf{X})}$$

- This transforms our assumption into:

$$\overline{F}\left(\mathbf{X}+\epsilon\mathbf{c}\right)+\boldsymbol{\omega}\left(\mathbf{X}+\epsilon\mathbf{c}\right)=\left(\overline{F}\mathbf{X}+\boldsymbol{\omega}(\mathbf{X})\right)+\epsilon\overline{F}\mathbf{c}$$

  which simplifies to a periodicity condition on the *fluctuation displacement field* $\boldsymbol{\omega}$:

$$\boldsymbol{\omega}\left(\mathbf{X}+\epsilon\mathbf{c}\right)=\boldsymbol{\omega}(\mathbf{X})$$



- Therefore, $\boldsymbol{\Phi}$ has the requested symmetry
  $\iff$ it is “linear plus periodic.”

The Nonlinear Elasticity Cell Problem

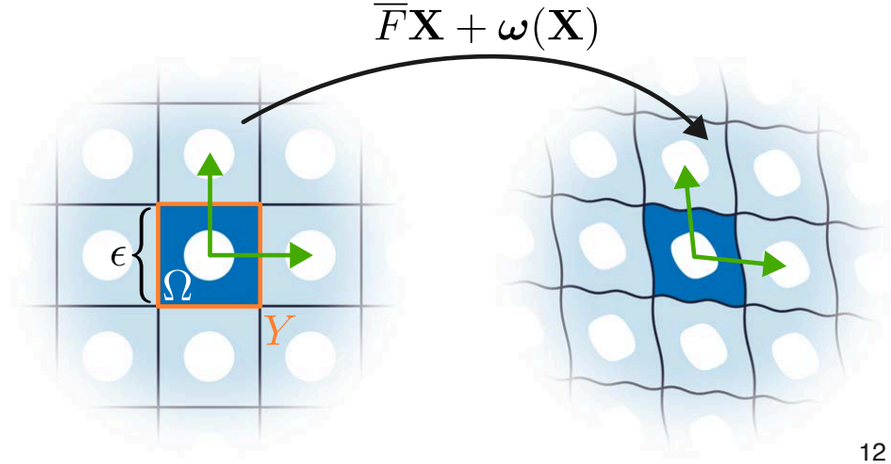
- Homogenized energy density $\overline{\psi}(\overline{F})$ is the minimum over *admissible* Φ of the average energy density in the full tiling.
- Since every cell experiences the same strains by assumption, this is just the average over one cell Y:

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

- Therefore, the periodic homogenization process reduces to solving a “cell problem” on a single unit cell*.

* more on this later

- Uniqueness? See Problem Set Exercise 1.



Weak Form of Cell Problem Equilibrium

- The first-order optimality condition for the cell problem is that, at a minimizer $\boldsymbol{\omega}^*$, all admissible (periodic) perturbations $\delta\boldsymbol{\omega}$ must satisfy:

$$\left\langle \frac{\partial}{\partial \boldsymbol{\omega}} \int_\Omega \psi(\overline{F} + \nabla\boldsymbol{\omega}^*)\,\mathrm{d}\mathbf{X},\, \delta\boldsymbol{\omega} \right\rangle \overset{!}{=} 0$$

- An expression for this first variation can be obtained as:

$$\left.\frac{\mathrm{d}}{\mathrm{d}h}\right|_{h=0} \int_\Omega \psi(\overline{F} + \nabla(\boldsymbol{\omega}^* + h\delta\boldsymbol{\omega}))\,\mathrm{d}\mathbf{X}$$

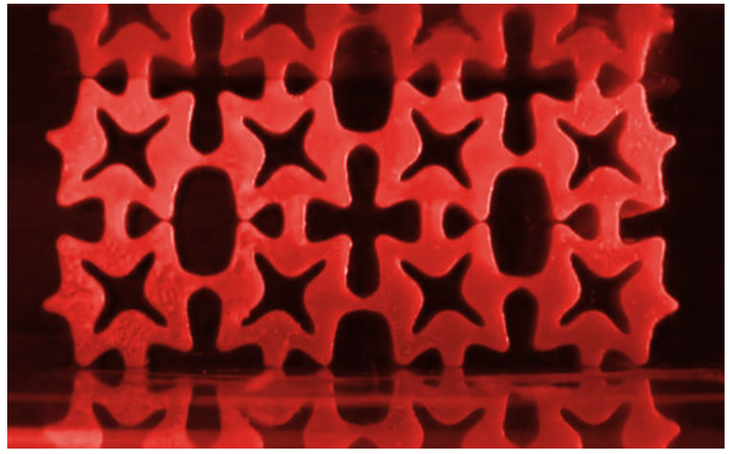
  which, by the chain rule, equals:

$$\int_\Omega \psi'(\overline{F} + \nabla\boldsymbol{\omega}^*) : \nabla(\delta\boldsymbol{\omega})\,\mathrm{d}\mathbf{X}.$$

- This leads to the *weak form of the equilibrium problem*:

$$\boxed{\int_\Omega \psi'(\overline{F} + \nabla\boldsymbol{\omega}^*) : \nabla(\delta\boldsymbol{\omega})\,\mathrm{d}\mathbf{X} \overset{!}{=} 0 \quad \forall \delta\boldsymbol{\omega} \text{ periodic.}}$$

Important Caveat: Symmetry-Breaking Instabilities



- Under finite deformations, instabilities can break the assumed cell-level periodicity.
- The deformation is still periodic, but the wavelength can be larger than the cell size.
- Accurately computing the deformation requires solving the cell problem on a 2×2 or larger supercell.

How large? Cannot be predicted *a priori*!

- Onset of instabilities still can be detected efficiently on a single unit cell using Bloch-wave methods.
- We'll neglect these instabilities in this course, assuming no such instabilities (or that Y has been enlarged appropriately).

Homogenized Stress: The Effective Stress-Strain Relation

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

- The (PK1) stress of a material with energy density $\psi(F)$ is given by $\psi'(F) = \frac{\partial \psi}{\partial F}(F)$.
- Likewise effective stress of the metamaterial is obtained by differentiating $\overline{\psi}(\overline{F})$.
- Introducing

$$\boldsymbol{\omega}^*(\overline{F}) = \operatorname*{argmin}_{\boldsymbol{\omega}\ \text{periodic}} \frac{1}{|Y|} \int_\Omega \psi(\overline{F} + \nabla \boldsymbol{\omega})\, \mathrm{d}\mathbf{X}.$$

  and assuming local uniqueness and smoothness of the minimizer yields an expression we can differentiate:

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

Homogenized Stress: The Effective Stress-Strain Relation

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

- Proceeding via the chain rule:

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

- We note that $\frac{\partial\overline{F}}{\partial\overline{F}}$ is the fourth-order identity tensor.

Recall the weak form of the cell problem equilibrium!

$$\int_\Omega \psi'(\overline{F} + \nabla\boldsymbol{\omega}^*) : \nabla(\delta\boldsymbol{\omega})\,\mathrm{d}\mathbf{X} \overset{!}{=} 0 \quad \forall\delta\boldsymbol{\omega} \text{ periodic}$$

- The $\frac{\partial\boldsymbol{\omega}^*}{\partial\overline{F}}$ term vanishes, leaving just the average of the microscopic stress over the cell:

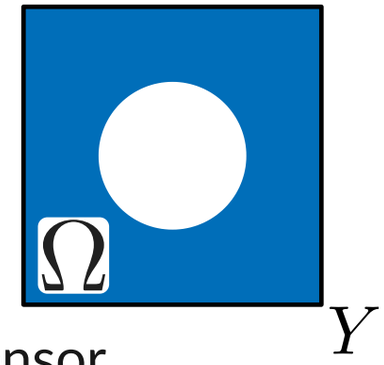
$$\boxed{\overline{\psi}'(\overline{F}) = \frac{1}{|Y|}\int_\Omega \psi'(\overline{F} + \nabla\boldsymbol{\omega}^*(\overline{F}))\,\mathrm{d}\mathbf{X}.}$$

Cell Problem for Linear Elasticity

- While linear elasticity is inaccurate for large bending deformations characteristic of flexible structures, it is still an essential tool for computational design.
- Periodic homogenization for linear elasticity is **far** less expensive than the full nonlinear homogenization problem.
- The vast majority of microstructure design algorithms are based on linear elasticity.



Cell Problem for Linear Elasticity



- The nonlinear weak form of the cell equilibrium problem is:

$$\int_\Omega \psi'(\overline{F} + \nabla \boldsymbol{\omega}^*) : \nabla(\delta\boldsymbol{\omega}) \, \mathrm{d}\mathbf{X} \overset{!}{=} 0 \quad \forall \delta\boldsymbol{\omega} \text{ periodic}$$

- We take $\overline{F} = I + \overline{\varepsilon}$, where $\overline{\varepsilon}$ is a symmetric infinitesimal macroscopic strain tensor.
- Linearizing the weak form about the undeformed configuration ($F = I$) then gives:

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

- This is the weak form of the linear elasticity cell problem, which is a linear PDE for periodic fluctuation displacement $\boldsymbol{\omega}^*$.

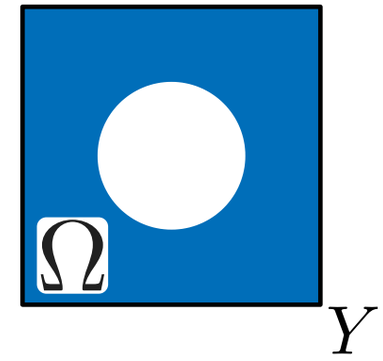
Recall that $C = \psi''(I)$.

- Because the problem is linear also with respect to $\overline{\varepsilon}$, the solution $\boldsymbol{\omega}^*$ depends linearly on $\overline{\varepsilon}$:

$$\boldsymbol{\omega}^* = \sum_{i,j} \overline{\varepsilon}_{ij} \, \boldsymbol{\omega}^{ij}, \quad \text{where } \boldsymbol{\omega}^{ij} \text{ solves cell problem for basis strain } \mathbf{sym}(\mathbf{e}_i \otimes \mathbf{e}_j).$$

- Thus characterizing the microscopic requires *only 3 cell problem solves in 2D and 6 in 3D*!

Periodic Homogenization for Linear Elasticity



- Recall the formula for homogenized stress:

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

- Since $\boldsymbol{\omega}^*$ depends linearly on $\overline{\varepsilon}$, this stress expression does too.
- We write this linear relationship as:

$$\overline{\sigma}_{\text{lin}}(\overline{\varepsilon}) = \overline{C} : \overline{\varepsilon},$$

  where $\overline{C}$ is the *homogenized elasticity tensor*.
- The stress formula can be seen as a definition of $\overline{C}$:

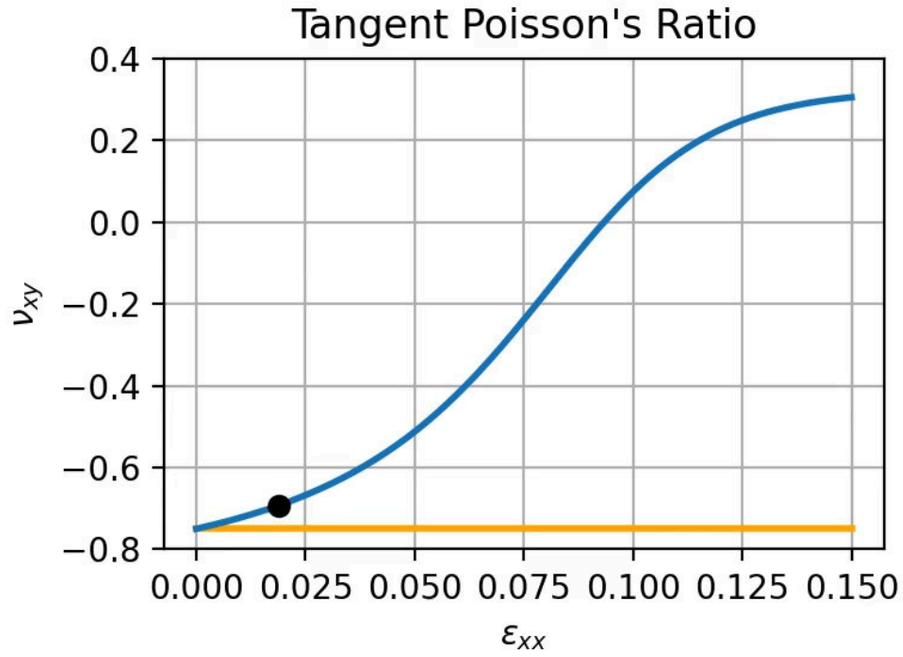
$$\boxed{\overline{C} : \overline{\varepsilon} = \frac{1}{|Y|}\int_\Omega C : \left(\overline{\varepsilon} + \varepsilon_{\text{lin}}(\boldsymbol{\omega}^*)\right)\mathrm{d}\mathbf{X}.}$$

Problem Set Exercise 3 asks you to explain how this definition can be applied to compute the values of $\overline{C}$.

Inadequacy of Linear Elasticity Microstructure Design

- For flexible metamaterials undergoing finite strains (*e.g.*, for shape morphing), the geometric linearization of linear elasticity introduces severe inaccuracies.
- This microstructure was designed for $\nu = -0.8$:



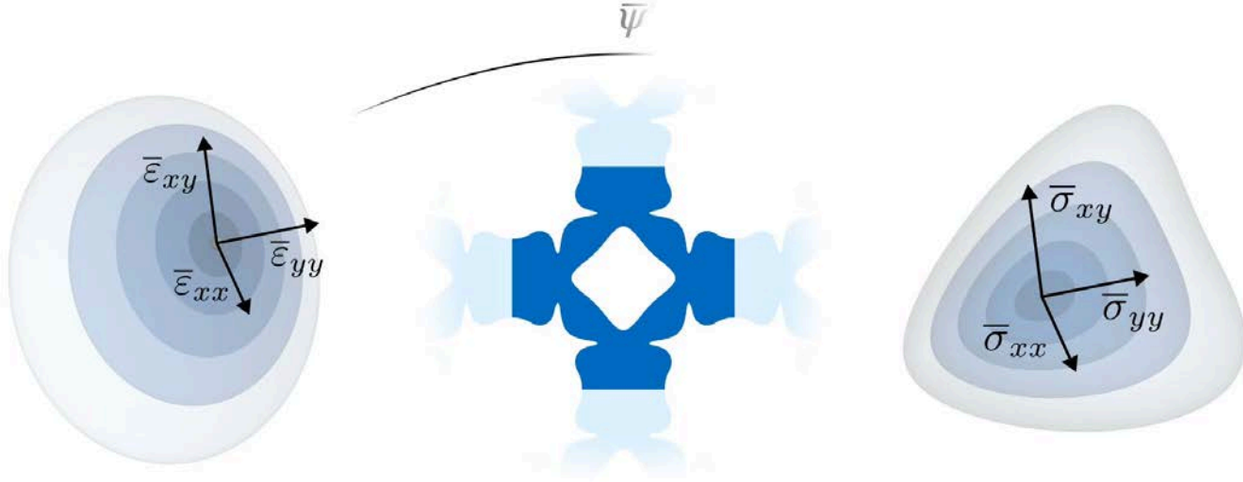


- Solving the full nonlinear cell problem is needed to accurately predict this behavior.

Characterizing Flexible Metamaterials

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

Characterizing a flexible metamaterial requires solving the nonlinear cell problem for for each macroscopic deformation gradient $\overline{F} \in \mathcal{F}$ in some "region of interest" $\mathcal{F}$:



Most efforts to extend microstructure design to finite strains have avoided this complexity by controlling at a few sampled loads or along one deformation path (*e.g.*, uniaxial stretching).

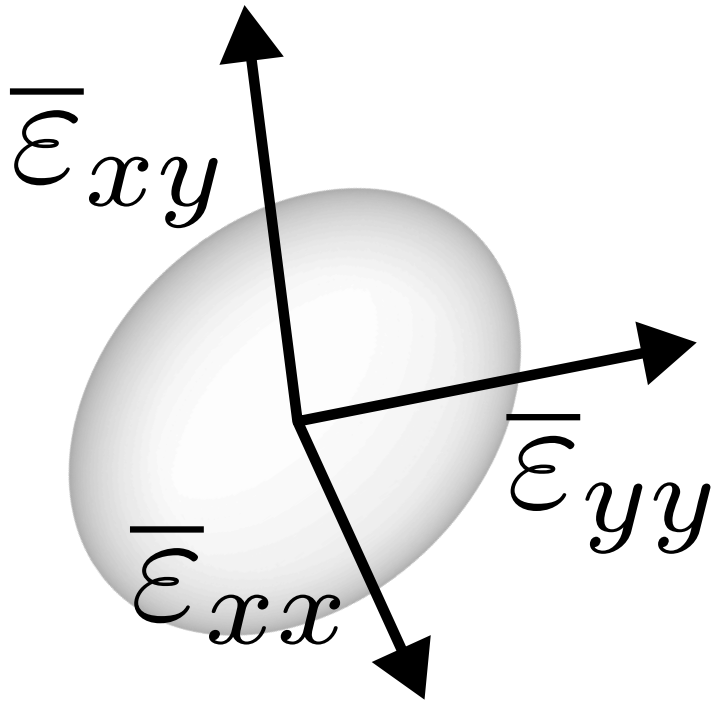
Characterizing Flexible Metamaterials: Interpolation

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

- Exactly solving the characterization problem for all $\overline{F} \in \mathcal{F}$ usually means an *infinite* number of nonlinear cell problem solves.
- However, approximate solutions to the characterization problem can be obtained by sampling and interpolation over $\mathcal{F}$.
- [Other Slide Deck]

HOMOGENIZATION

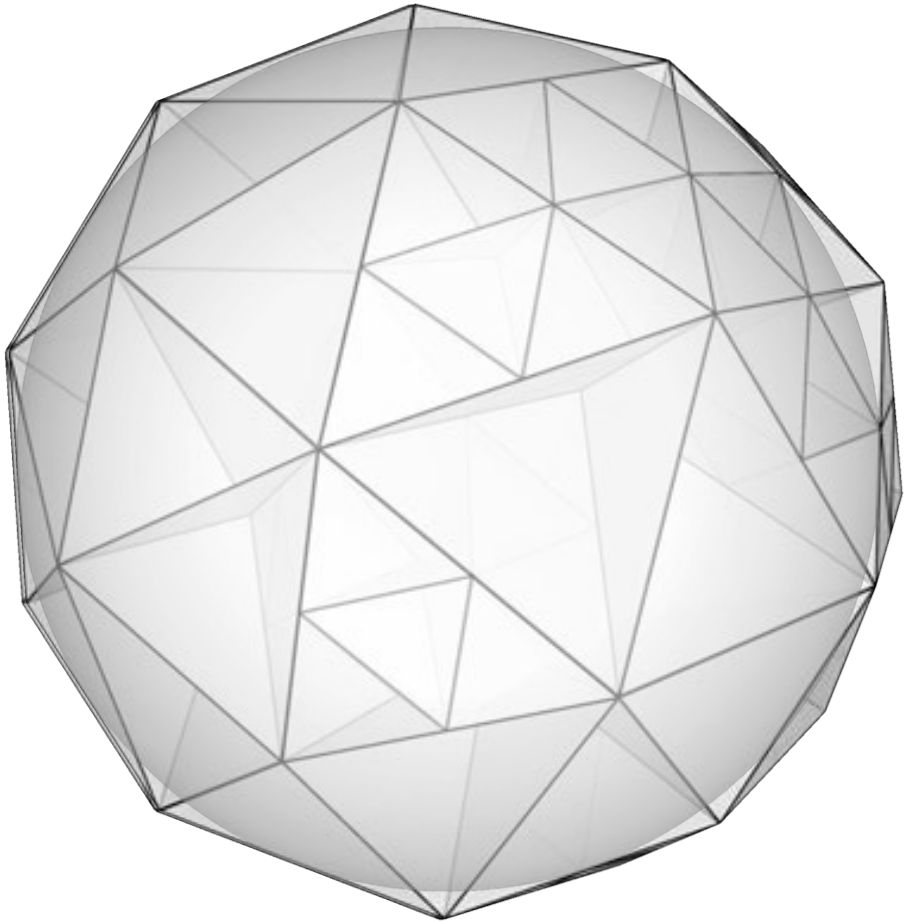
Decomposition Strain Domain



$\mathscr{E}$

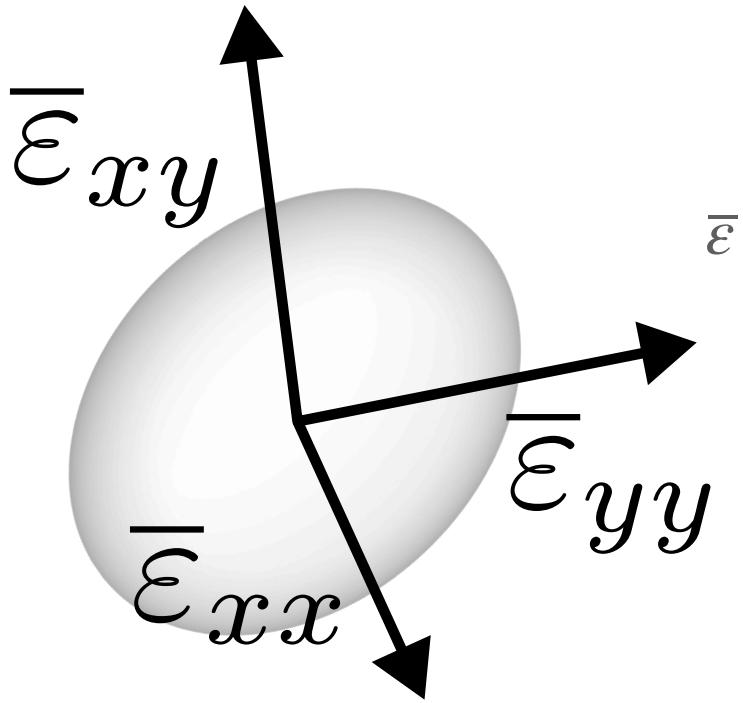
Strain 10%

Adaptive Subdivision



HOMOGENIZATION

Interpolation



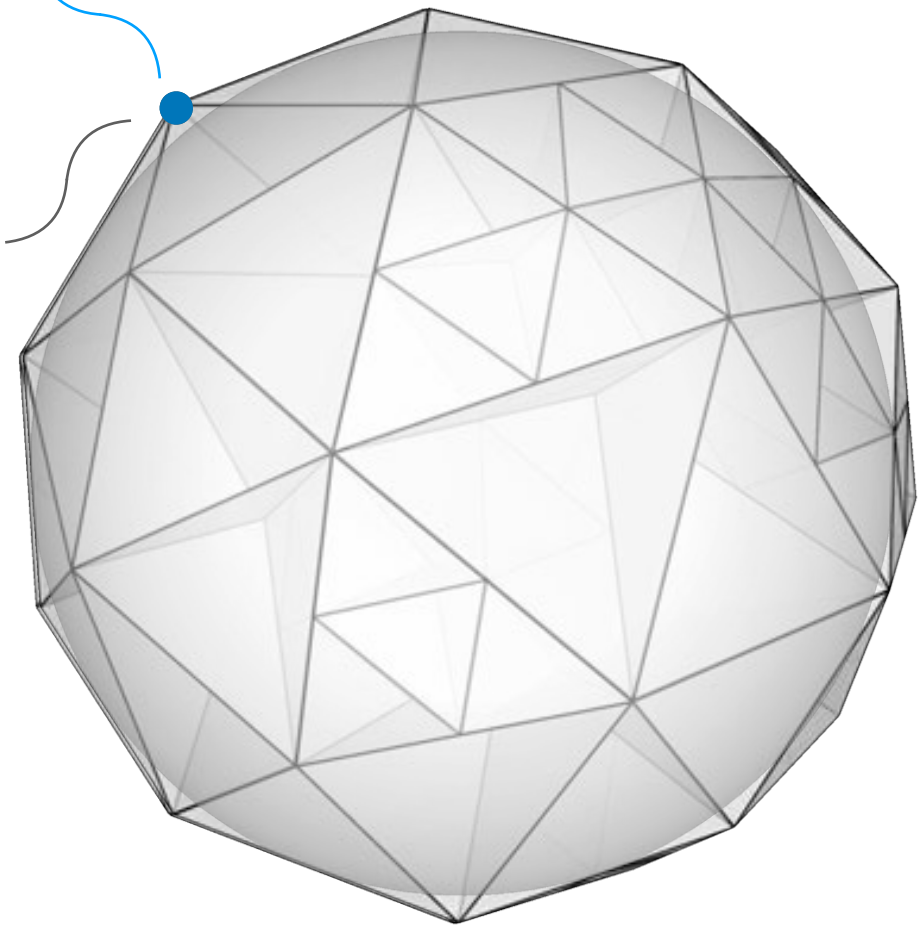
$\mathscr{E}$

Strain 10%

Interpolate displacement field ω^*

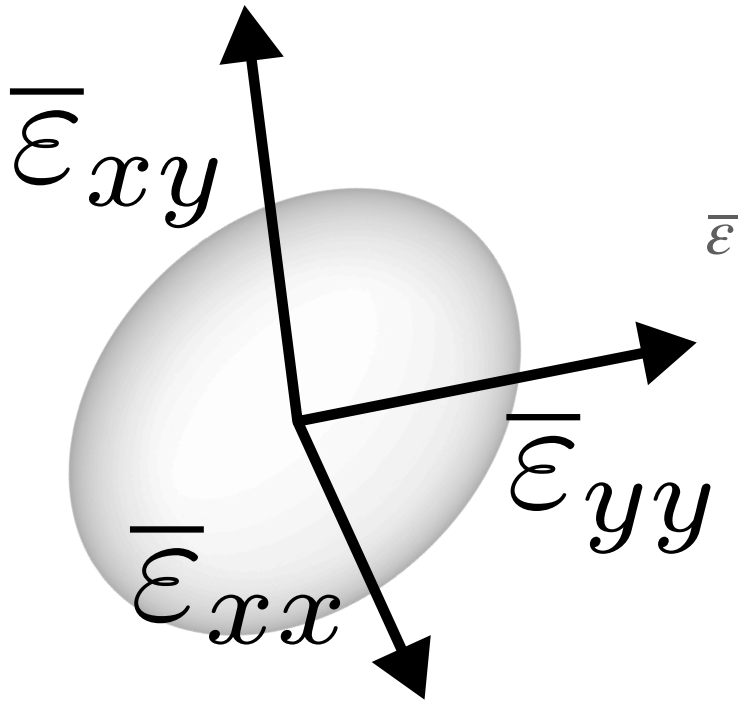
$$\omega^* = \arg \min_{\omega \, periodic} \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \omega + \overline{F}) \, d\mathbf{X}$$

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



HOMOGENIZATION

Interpolation



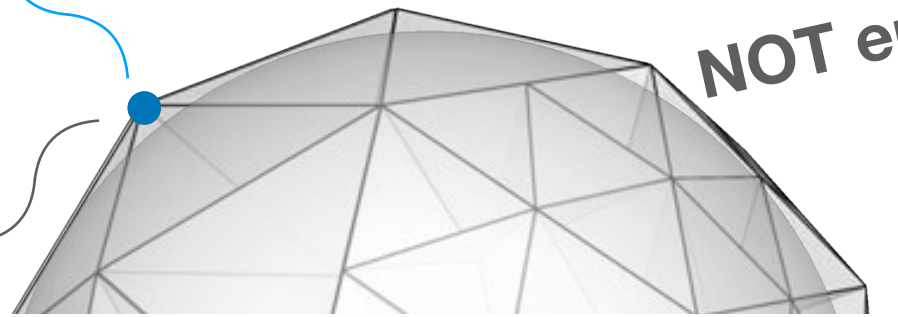
$\mathscr{E}$

Strain 10%

Interpolate displacement field ω^*

$$\omega^* = \arg \min_{\omega \, periodic} \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \omega + \overline{F}) \, d\mathbf{X}$$

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

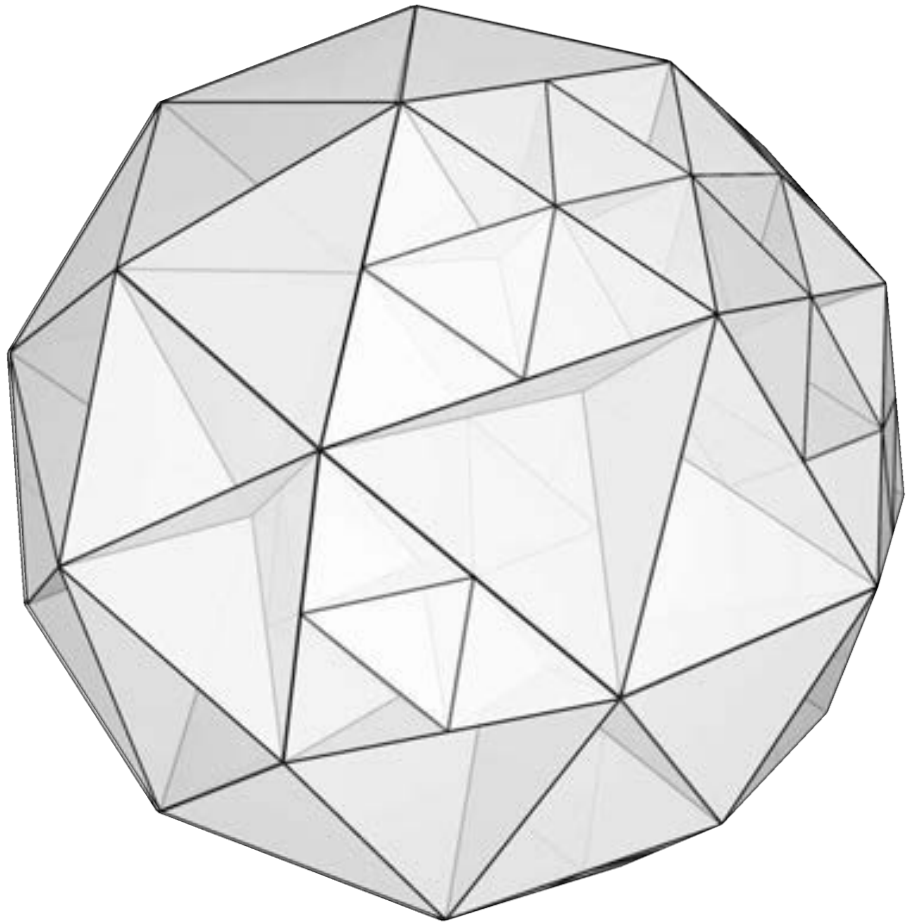


NOT energy $\overline{\Psi}$!

1. Efficient to evaluate the accuracy (no need for ground truth)
2. Accelerates nonlinear solves (provides high-quality initialization)



HOMOGENIZATION
Interpolation



Interpolate displacement field ω^*

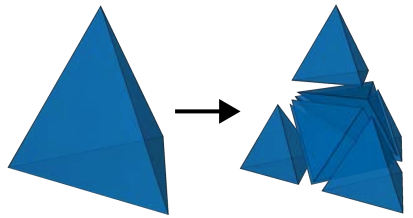
$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}\, periodic} \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \boldsymbol{\omega} + \overline{F})\, d\mathbf{X}$$

Linear Interpolation C^0

ω^*

Evaluate at every center of tet

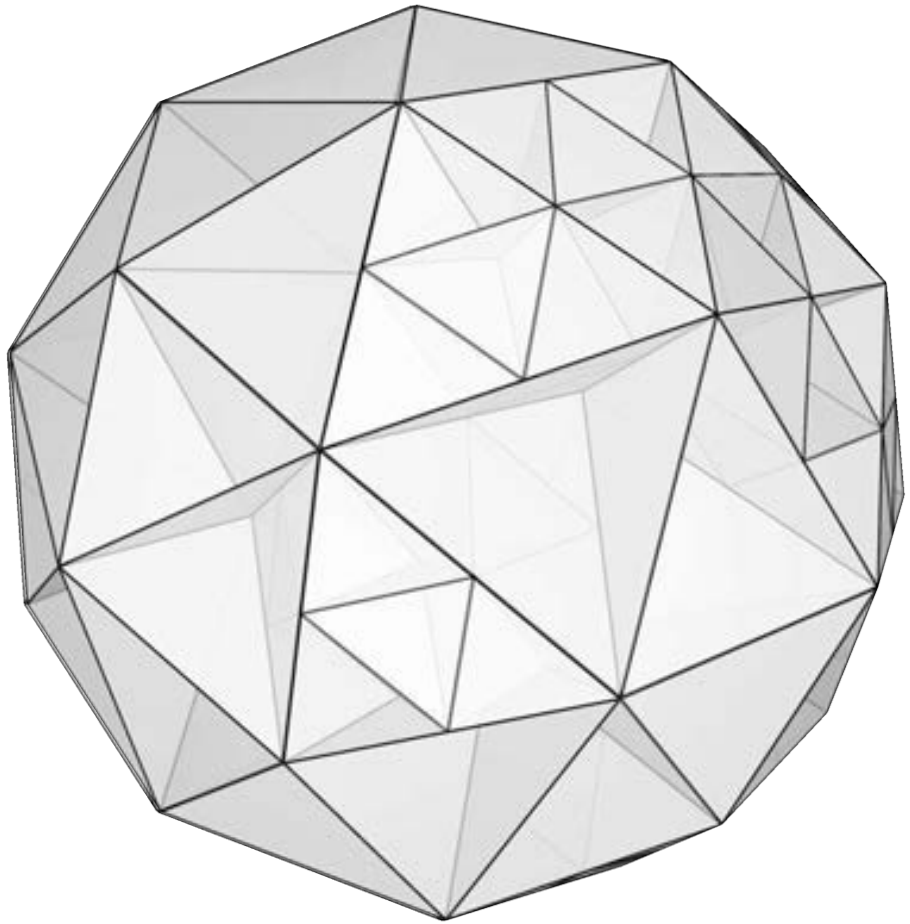
Check for the residual only



UCDAVIS

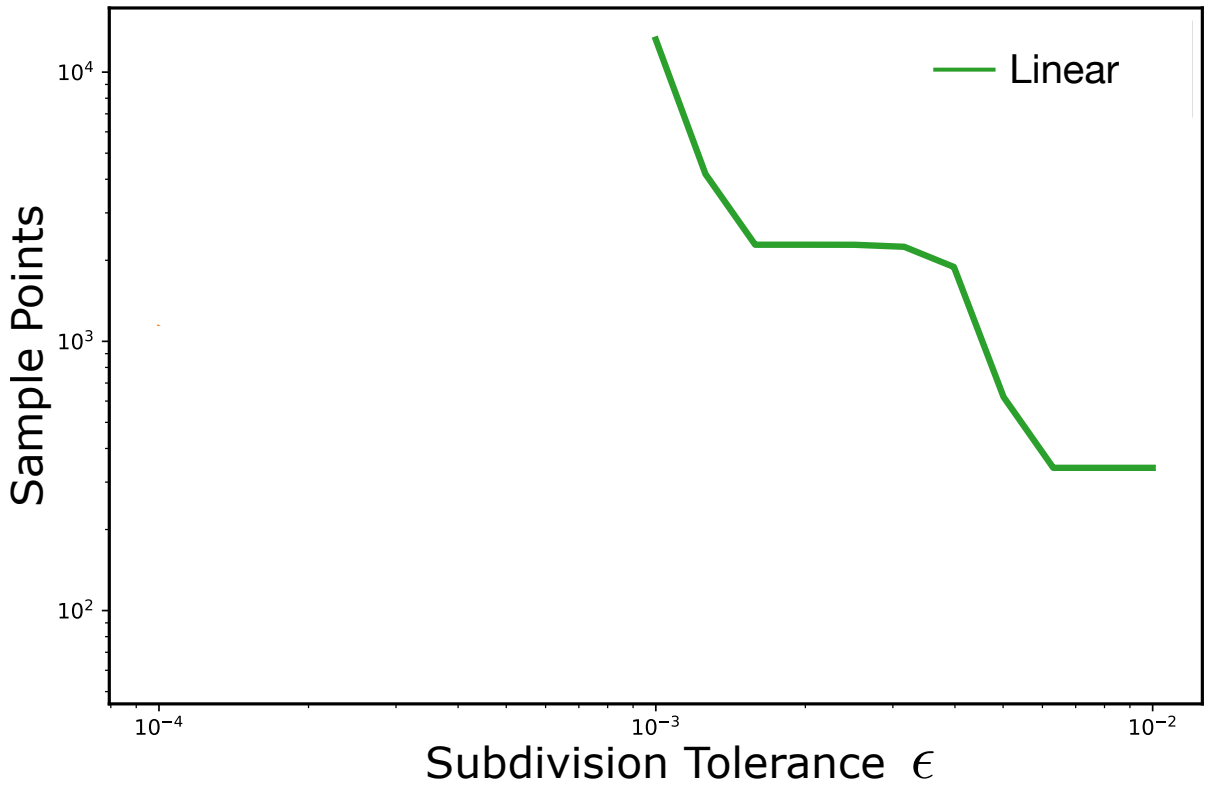
HOMOGENIZATION

Interpolation

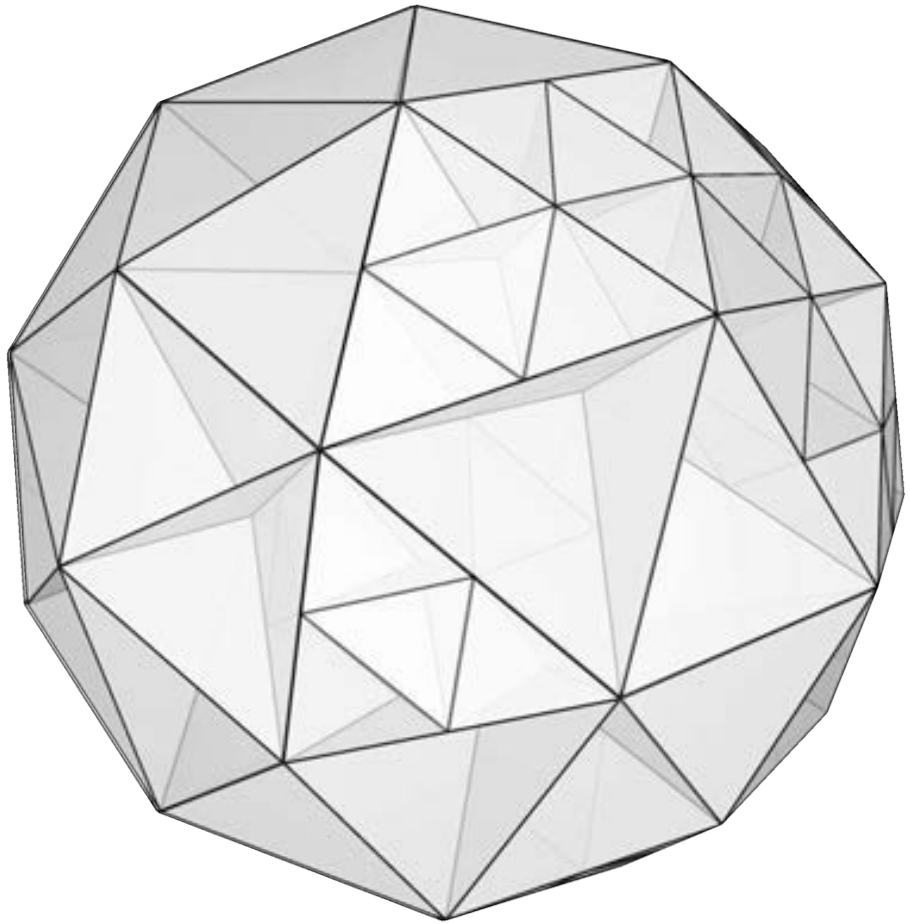


Interpolate displacement field ω^*

$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}\, periodic} \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \boldsymbol{\omega} + \overline{F})\, d\mathbf{X}$$



HOMOGENIZATION
Interpolation



Interpolate displacement field ω^*

$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}\, periodic} \frac{1}{|Y|} \int_{\Omega} \psi(\nabla \boldsymbol{\omega} + \overline{F})\, d\mathbf{X}$$

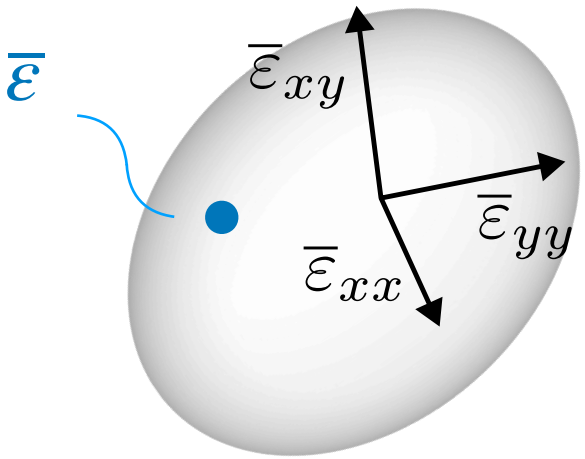
Powell-Sabin Interpolation C^1

$$\left(\boldsymbol{\omega}^*, \quad \frac{\partial \boldsymbol{\omega}^*}{\partial \overline{F}} \right)$$

derivative with respect to
the macroscopic strain

HOMOGENIZATION

Interpolation

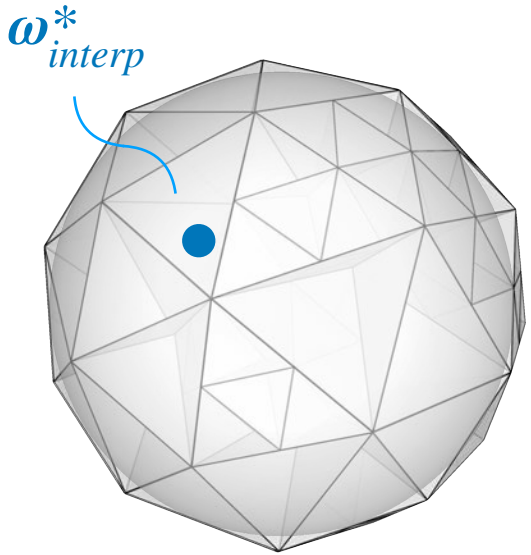


$\mathscr{E}$

Strain 10%

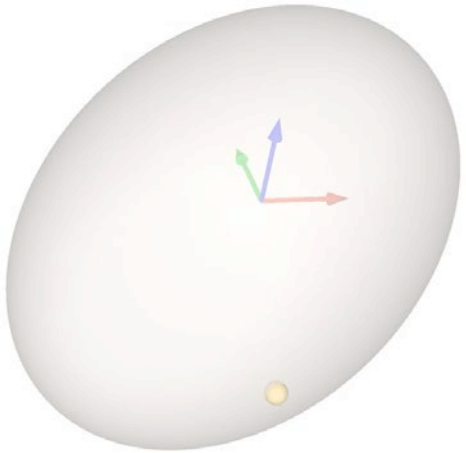


Adaptive Subdivision



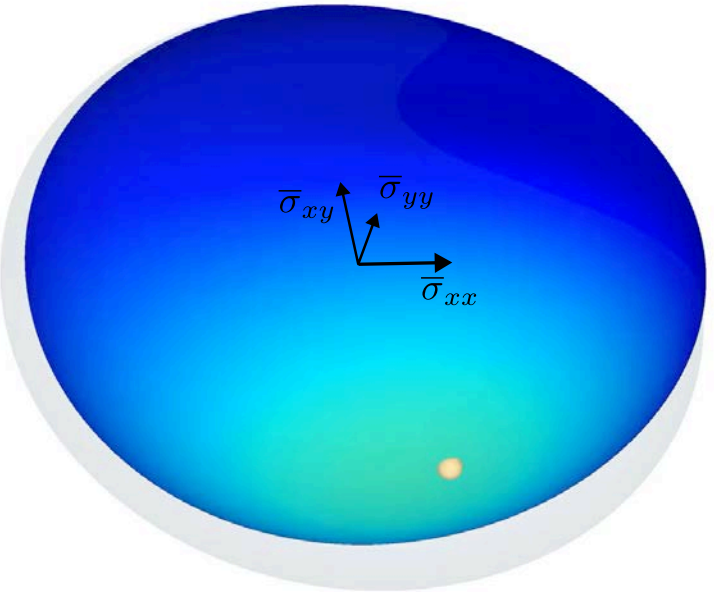
HOMOGENIZATION

Interpolation



$\mathscr{E}$ Strain 10%





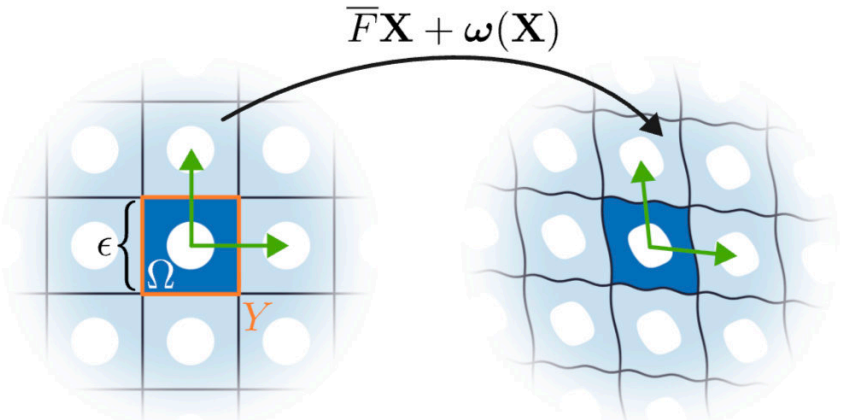
Stress Domain

The Nonlinear Elasticity Cell Problem

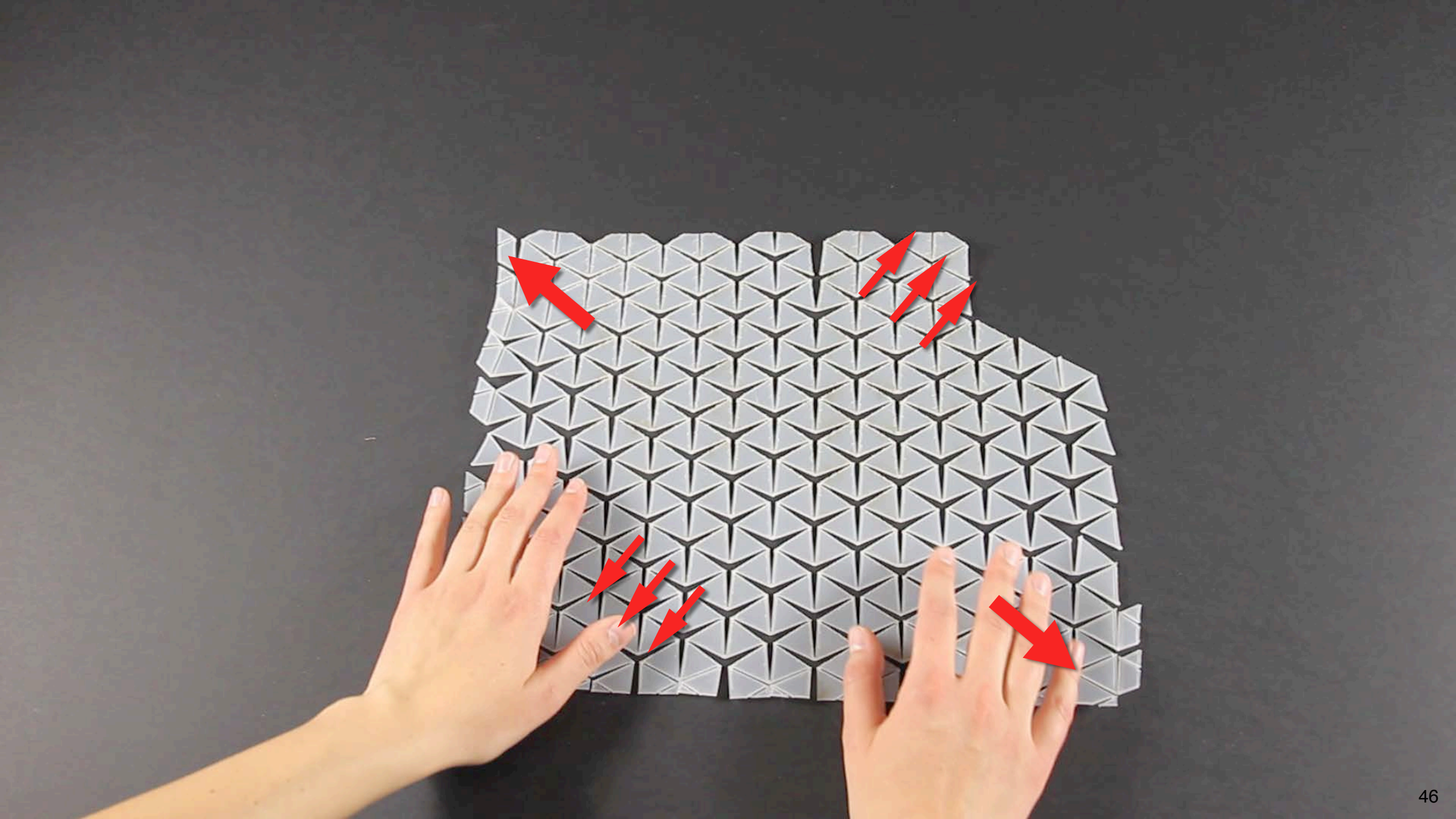
- Homogenized energy density $\overline{\psi}(\overline{F})$ is the minimum over *admissible* Φ of the average energy density in the full tiling.
- Since every cell experiences the same strains by assumption, this is just the average over one cell Y:

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

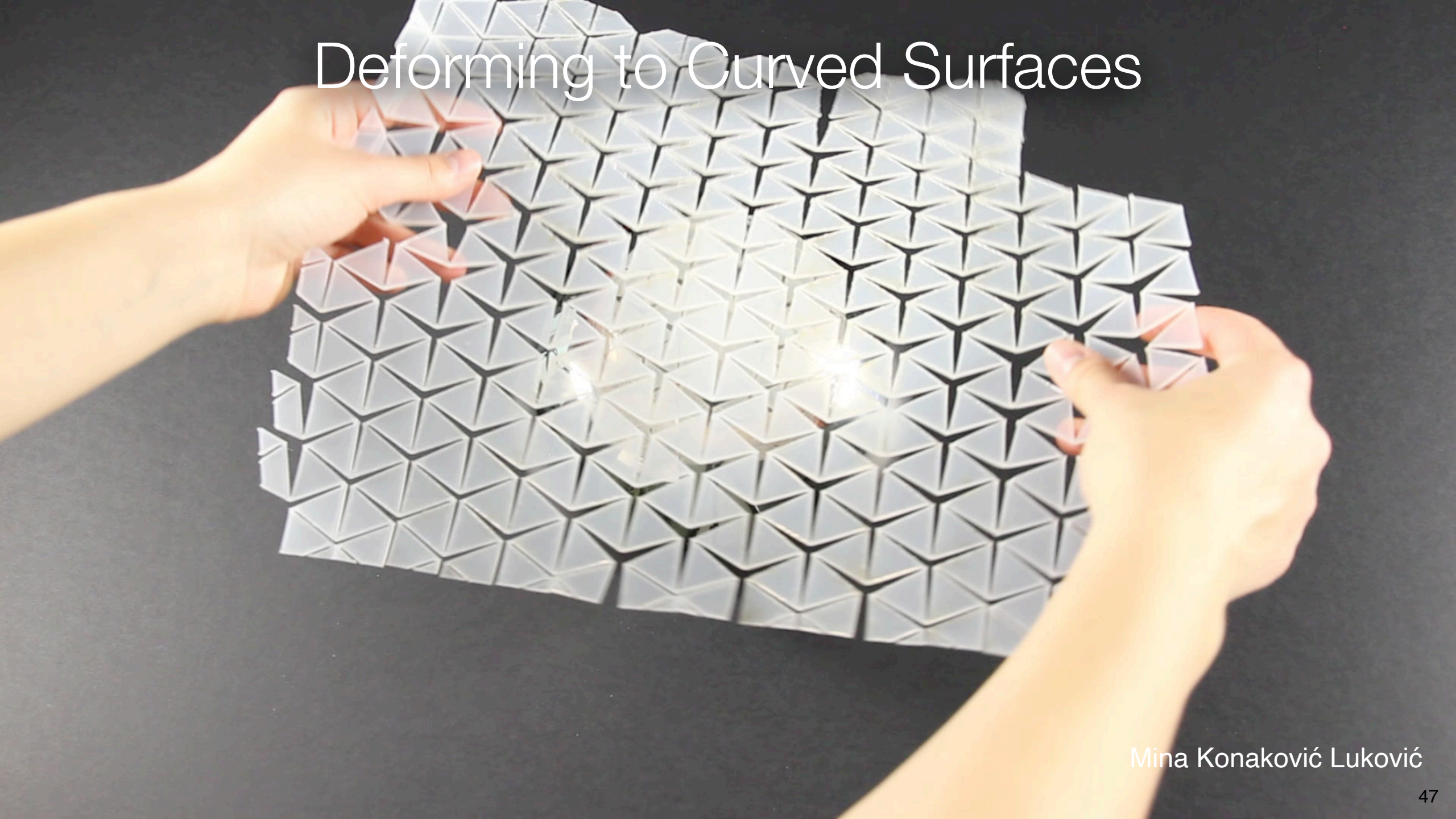
- Therefore, the periodic homogenization process reduces to solving a "cell problem" on a single unit cell*.



Auxetic Metamaterials

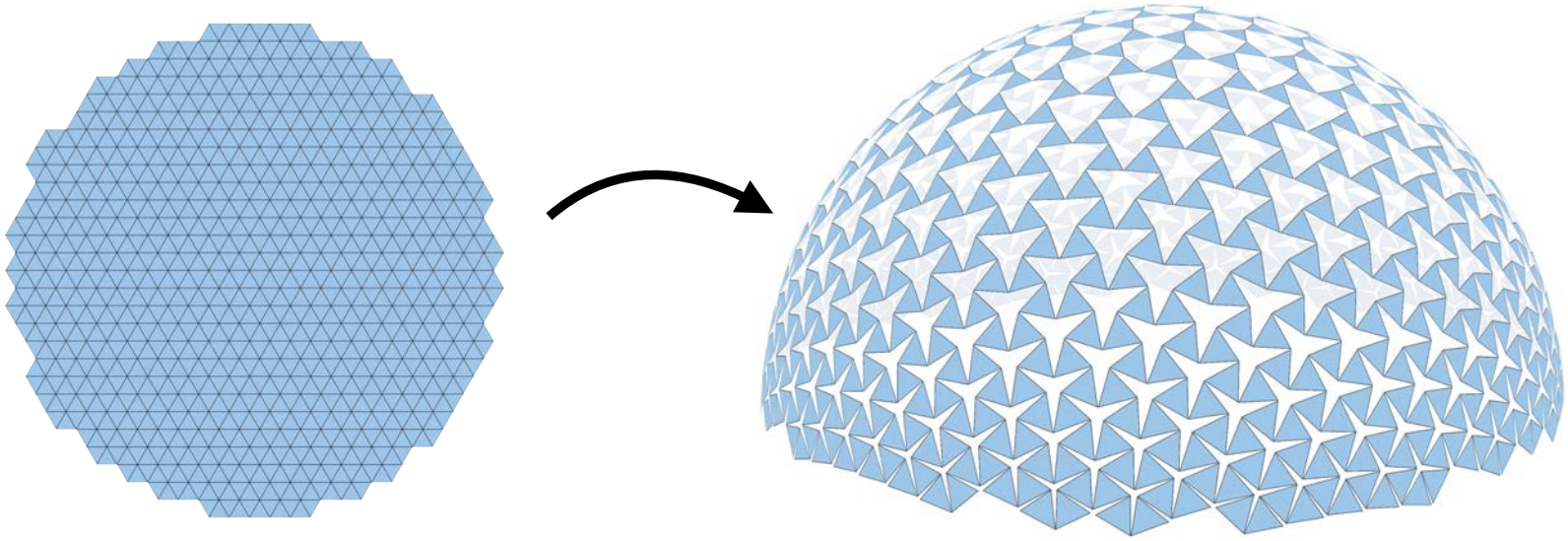


Deforming to Curved Surfaces



Mina Konaković Luković

Deforming to Curved Surfaces

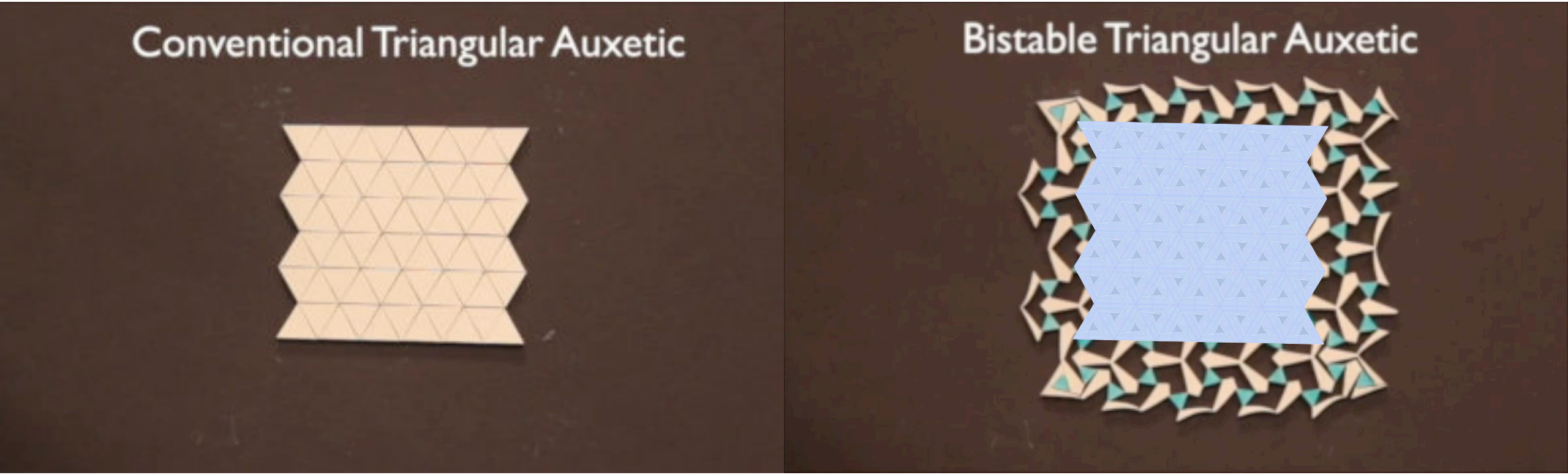


[Konaković Luković et al. 2018]

Nonuniform stretch introduces curvature!

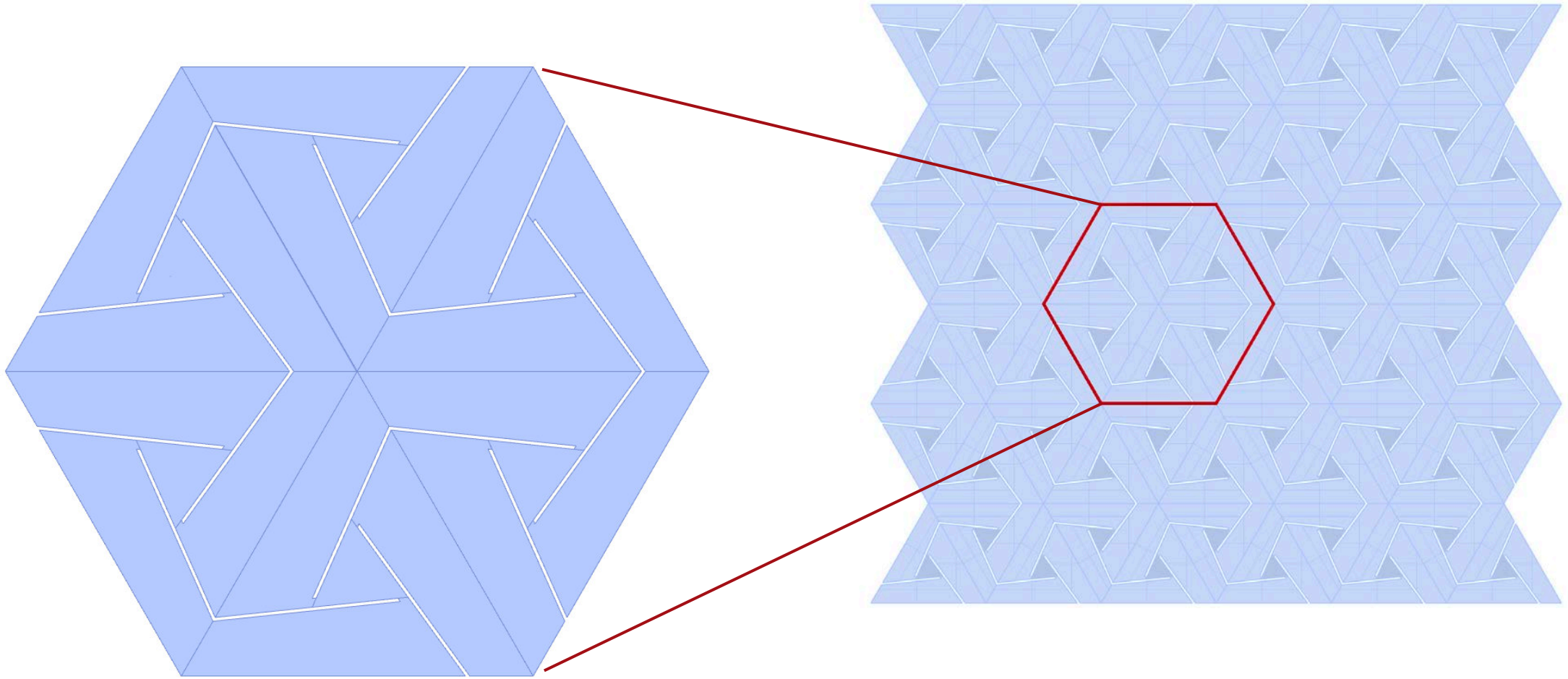
Idea: "program" a target shape into the material by controlling expansion.

Bistable Auxetics

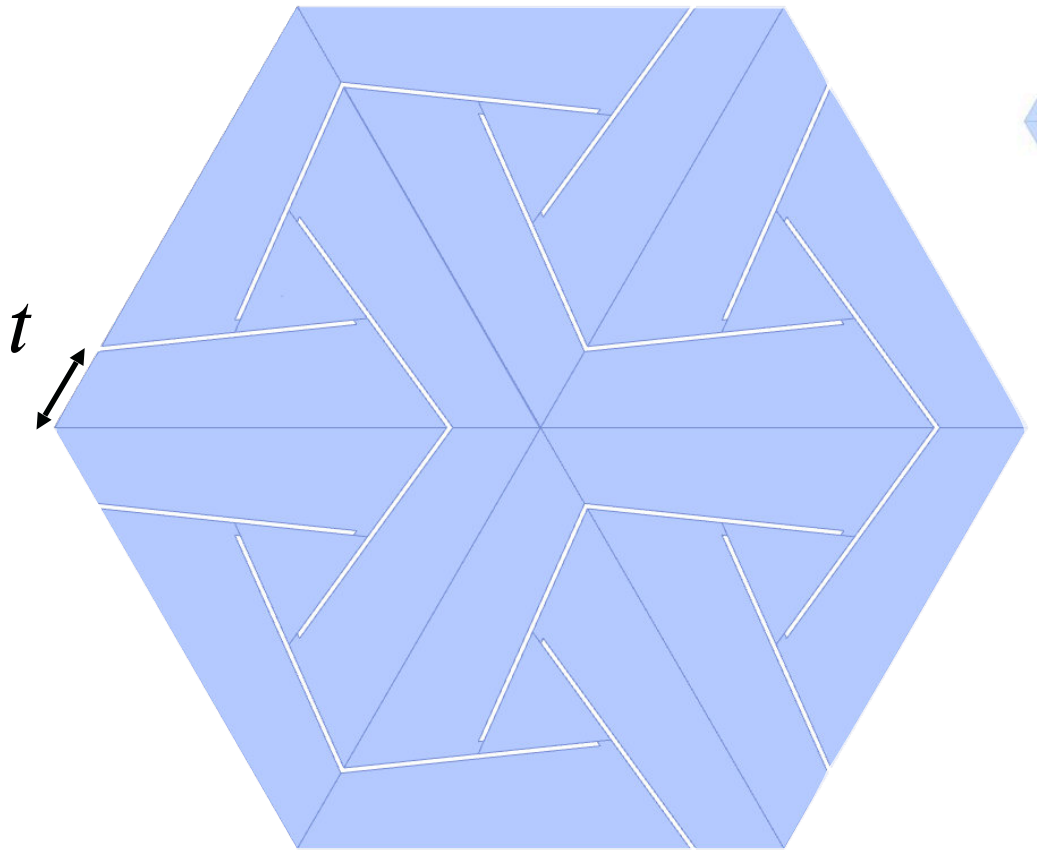


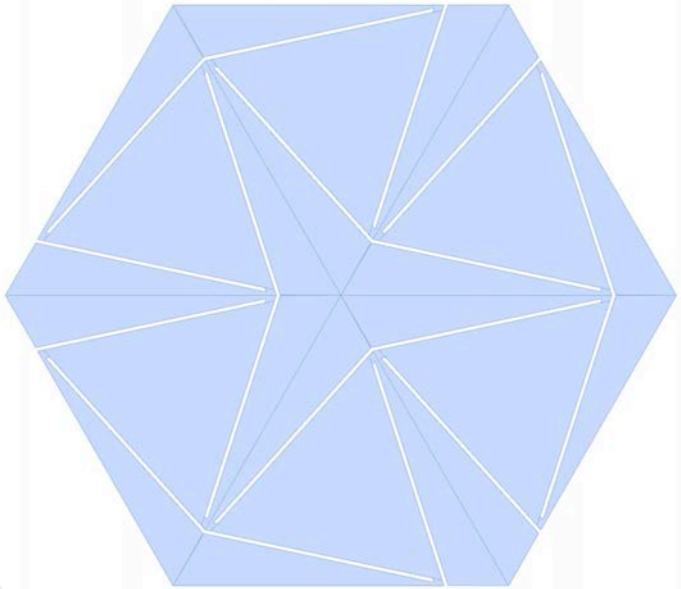
[Rafsanjani and Pasini 2016]

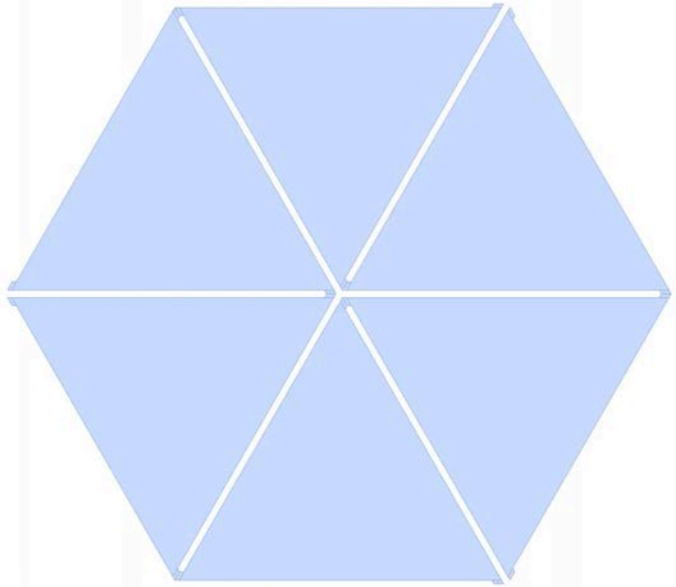
Parametric Unit Cell



Parametric Unit Cell







Parameters influence:

- Stretch factor
- Degree of bistability

Pattern Analysis

- The simple rotating triangle pattern can be understood analytically, but the bistable pattern is more complicated.
- Apply a simplified nonlinear homogenization:
  - Pattern is a unimode material, strongly resisting all deformations but uniform expansion.
  - So we can simply stretch along any axis (allowing transverse relaxation).
- Goal is to determine:
  - stretching at second stable state; and
  - stiffness around that state.

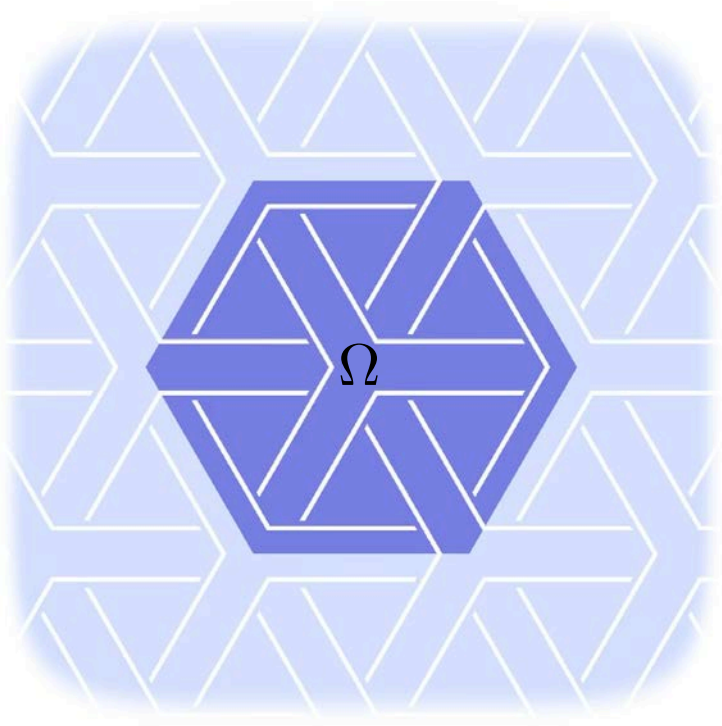


Bistable Auxetics Homogenization

Solve for the minimum energy configuration whose horizontal strain averages to the applied stretch:

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

$(\lambda = 1 + \varepsilon_{11})$



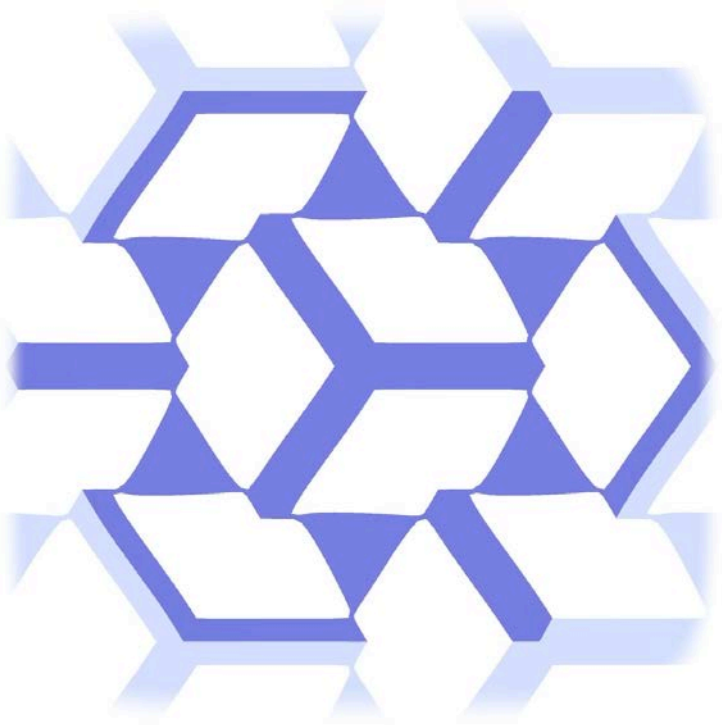
Macroscopic deformation gradient is now an optimization variable!

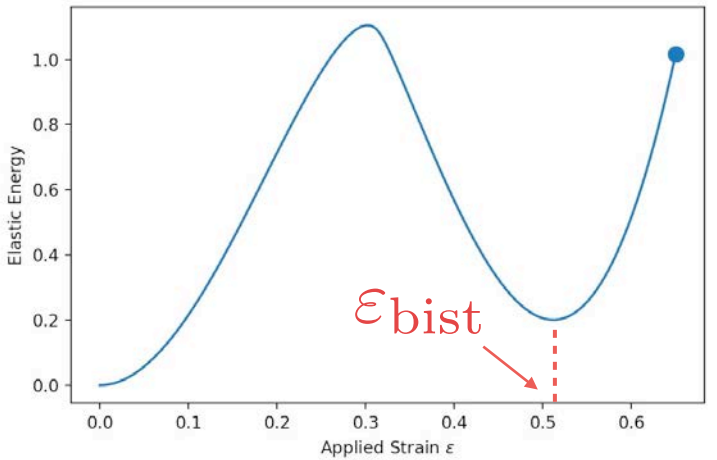
Bistable Auxetics Homogenization

Solve for the minimum energy configuration whose horizontal strain averages to the applied stretch:

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

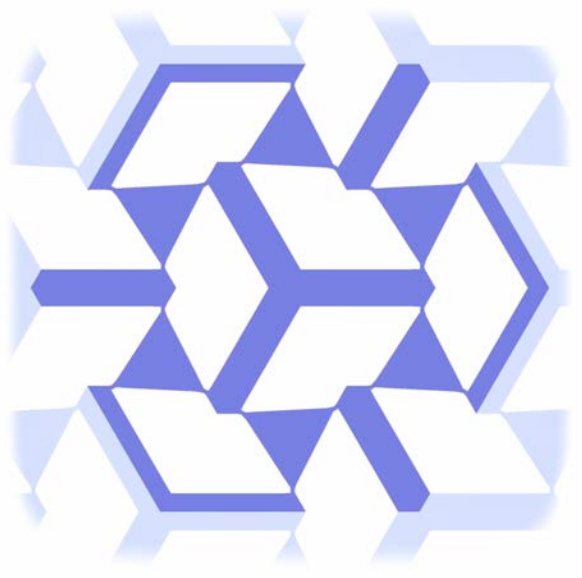
$$(\lambda = 1+\varepsilon_{11})$$

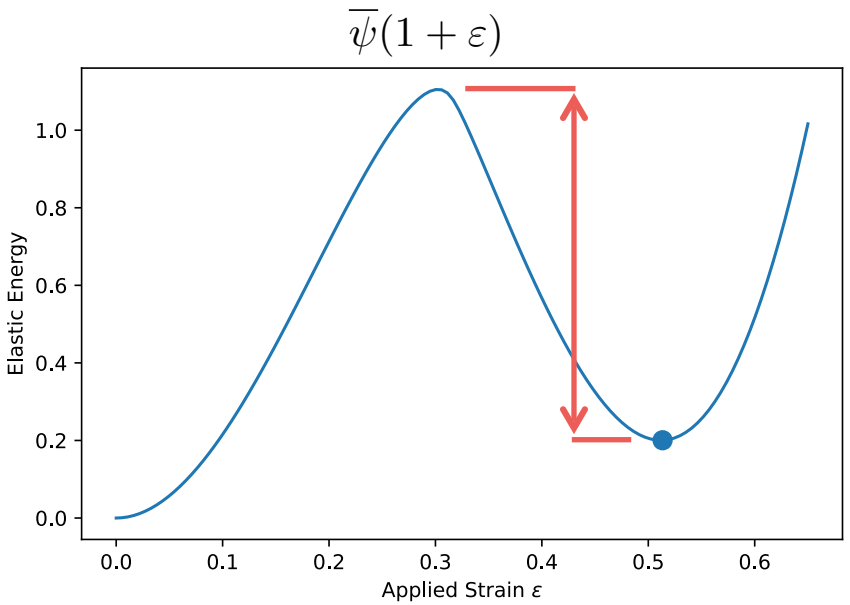




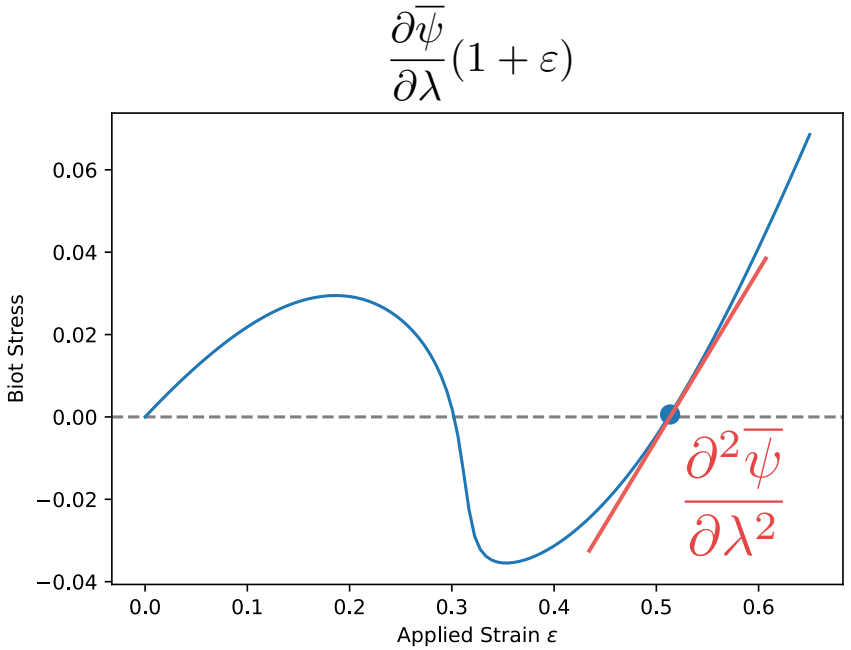
$$\overline{\psi}(\lambda) = \min_{\substack{\overline{F},\boldsymbol{\omega} \\ \boldsymbol{\omega}\ \text{periodic} \\ \overline{F}_{11}=\lambda}} \frac{1}{Y}\int_\Omega \psi(\overline{F}+\nabla\boldsymbol{\omega})\,\mathrm{d}\mathbf{X}$$

Bistability



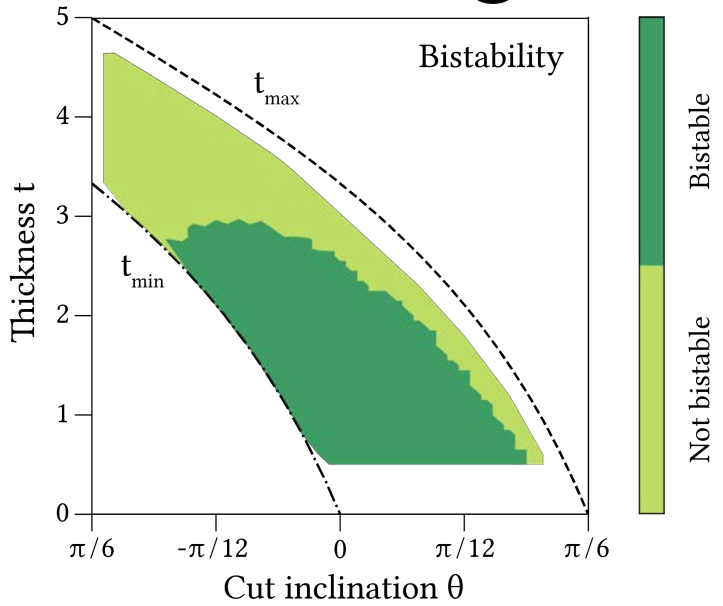


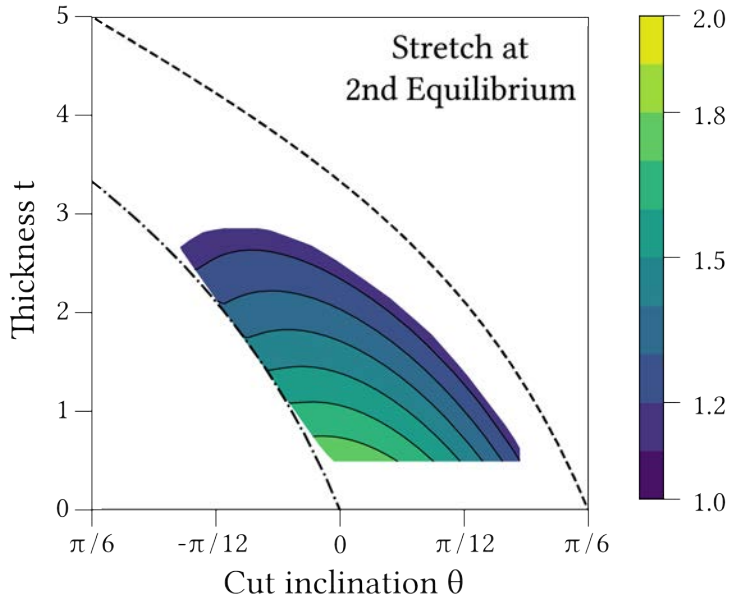
Height of Energy Barrier



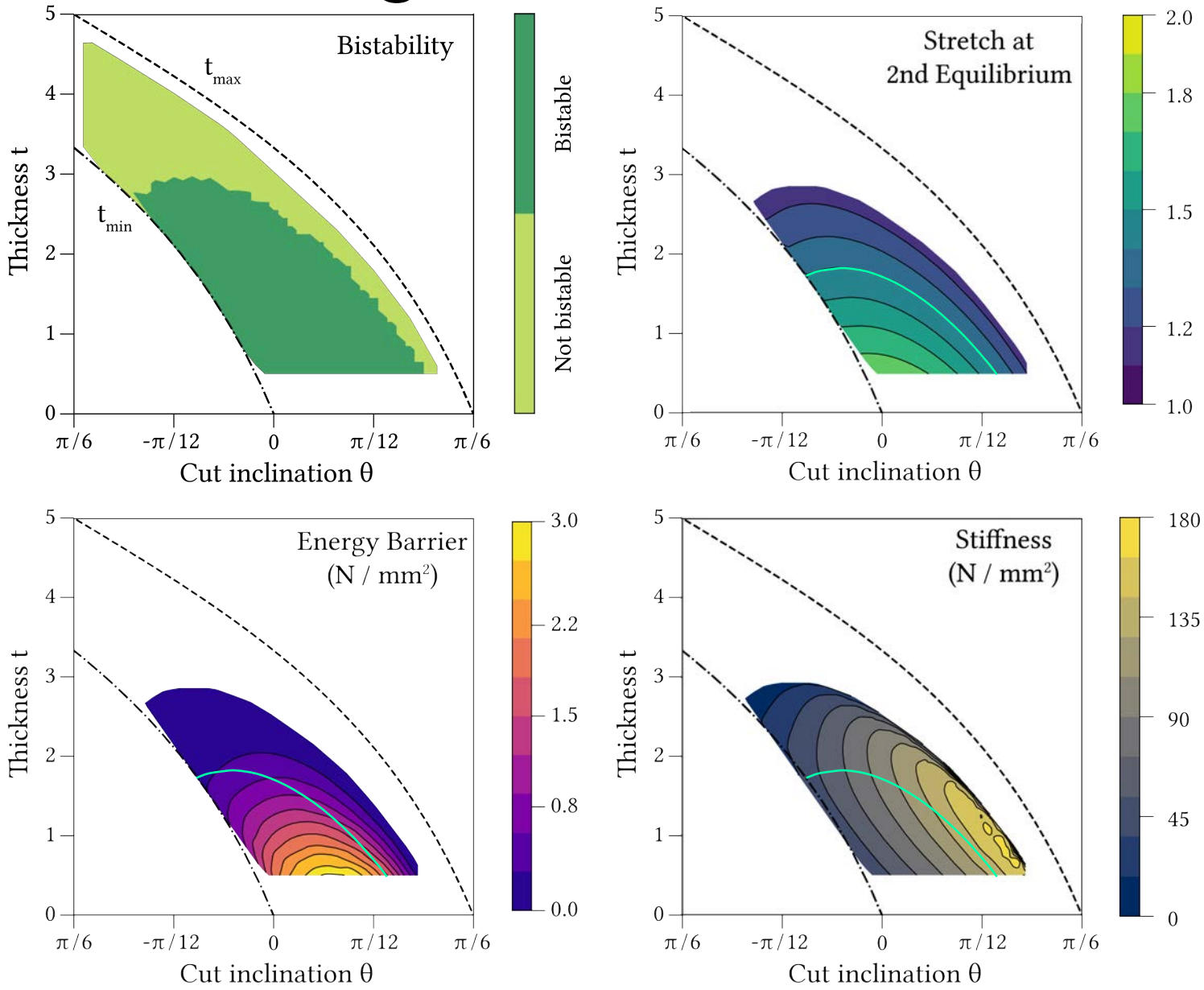
Stiffness

Homogenization Results

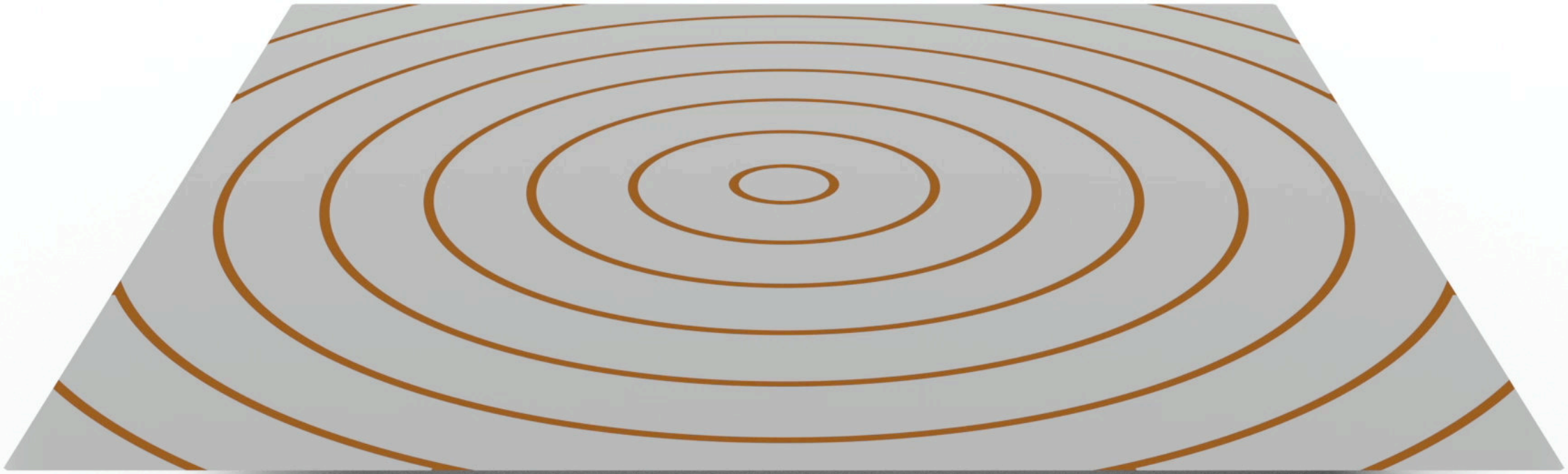




Homogenization Results



SIMULATION

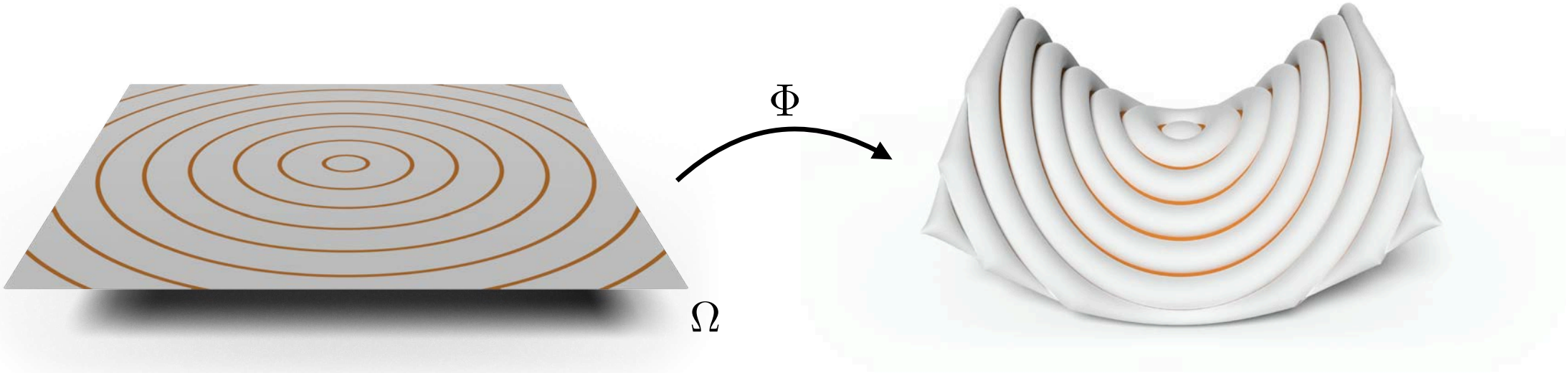


SIMULATION

Minimize total potential energy:

$$\min_{\Phi} E_{\text{Shell}}[\Phi] - p\ \texttt{Vol}(\Phi(\Omega))$$

Work Done by Inflation Pressure



SIMULATION

Minimize total potential energy:

$$\min_{\Phi} E_{\text{Shell}}[\Phi] - p\ \texttt{Vol}(\Phi(\Omega))$$



Wrinkles rather than resisting compression!

Koiter's Shell Theory:

Membrane

Bending

$$E^{\Omega}_{\text{shell}}[\Phi] := \frac{h}{2}\int_{\Omega} \varepsilon_s : C : \varepsilon_s \,\mathrm{d}A + \frac{h^3}{24}\int_{\Omega} \varepsilon_b : C : \varepsilon_b \,\mathrm{d}A,$$

$$\varepsilon_s := \frac{1}{2}\left((\nabla_{\Omega}\Phi)^{\top}\nabla_{\Omega}\Phi - I\right), \quad \varepsilon_b := \mathrm{II} - \mathrm{II}_0$$

TENSION FIELD THEORY MEMBRANES



Minimize total potential energy:

$$\min_{\Phi} E_{\text{Membrane}}[\Phi] - p\ \texttt{Vol}(\Phi(\Omega))$$

- Drop bending energy term.

$$E_{\text{Shell}}[\Phi] \quad \rightarrow \quad E_{\text{Membrane}}[\Phi] = \int_{\Omega} \psi(\nabla_{\Omega}\Phi)\,\mathrm{d}A$$

- Use a relaxed membrane energy density given by tension field theory.

  [Pipkin 1994: Relaxed energy densities for large deformations of membranes]

$$\psi_{\text{rel}}(C) = \min_{A \succeq 0} \psi(C + A) \qquad C = (\nabla_{\Omega}\Phi)^{\top}\nabla_{\Omega}\Phi$$

- This membrane energy is now convex!

TENSION FIELD THEORY MEMBRANES



Minimize total potential energy:

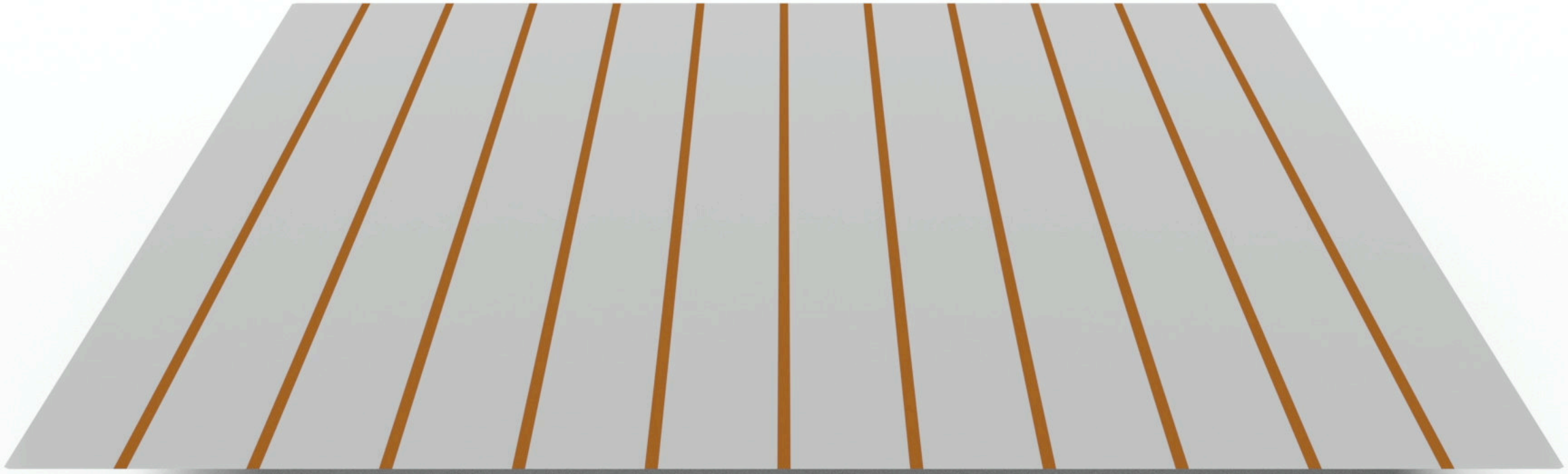
$$\min_{\Phi} E_{\text{Membrane}}[\Phi] - p\ \texttt{Vol}(\Phi(\Omega))$$

Tension field theory relaxation for incompressible neo-Hookean membrane in closed form:

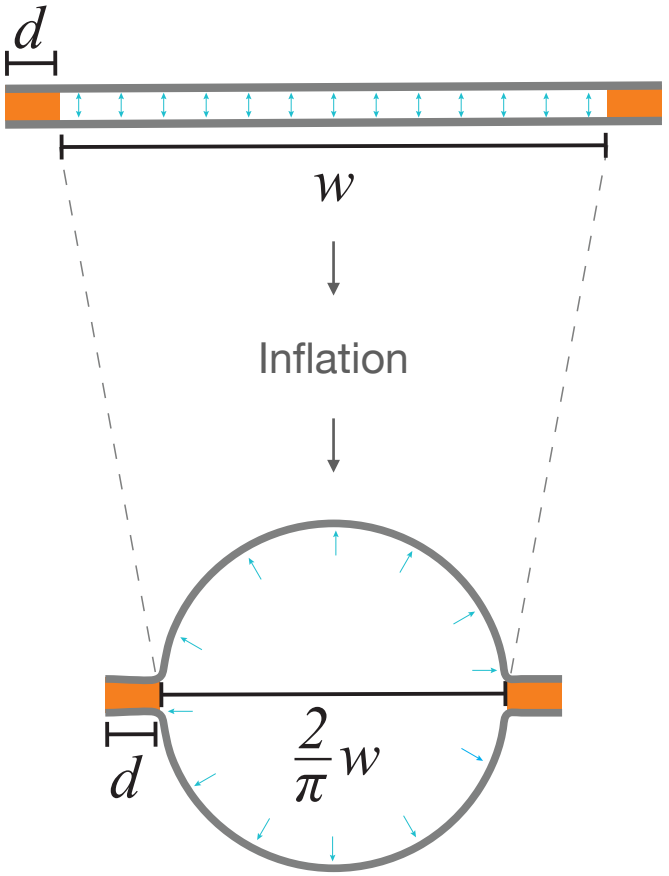
[Simulation Demo]

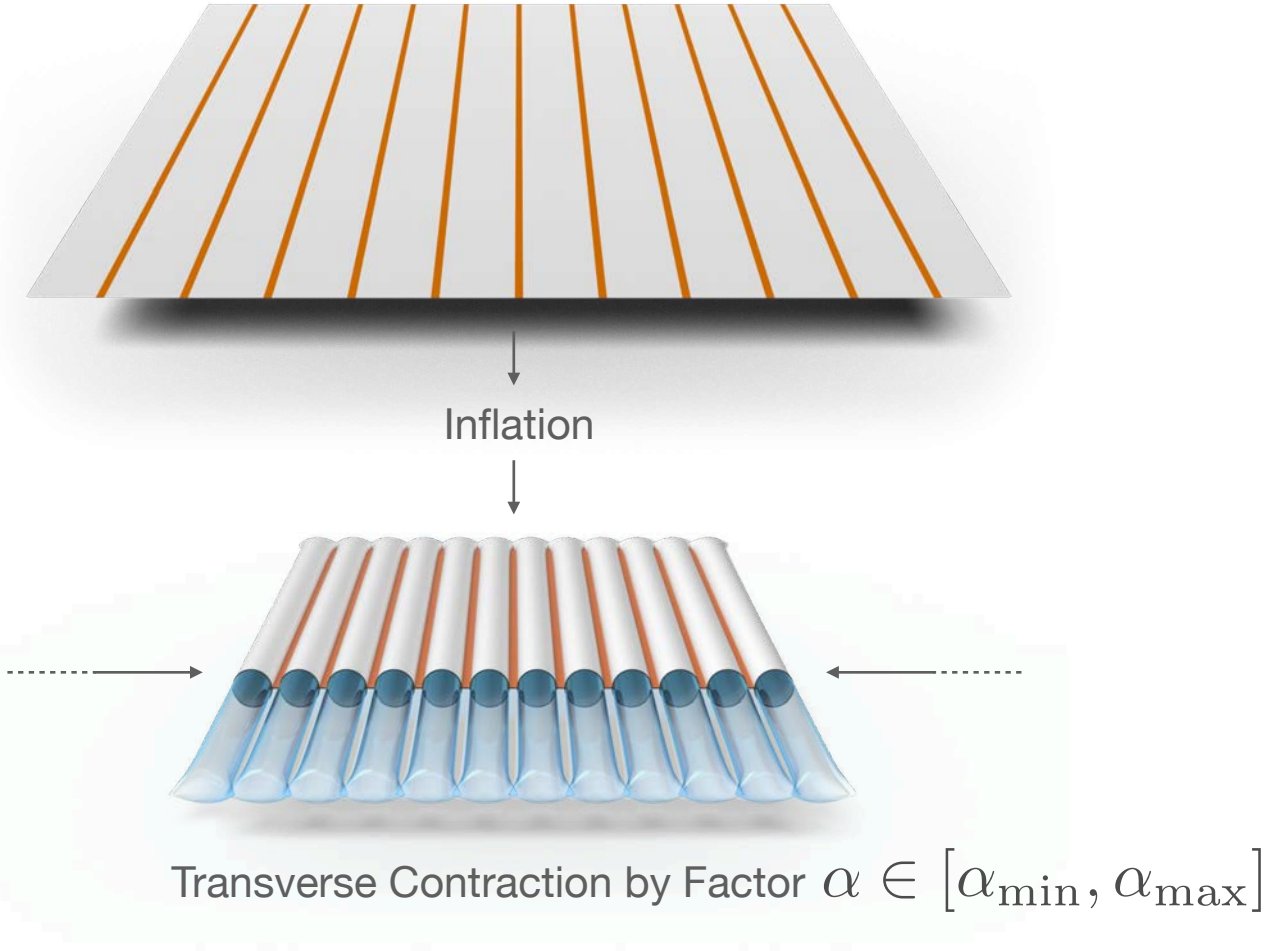
$$\psi(F) = \frac{Yh}{6}\begin{cases} 0 & \sigma_0 \leq 1,\ \sigma_1 \leq 1 \\ \sigma_0^2 + \frac{2}{\sigma_0} - 3 & \sigma_0 \geq 1,\ \sigma_1 \leq \frac{1}{\sqrt{\sigma_0}} \\ F : F + \frac{1}{\det(F^T F)} - 3 & \text{otherwise} \end{cases}$$

GEOMETRIC ABSTRACTION/HOMOGENIZATION



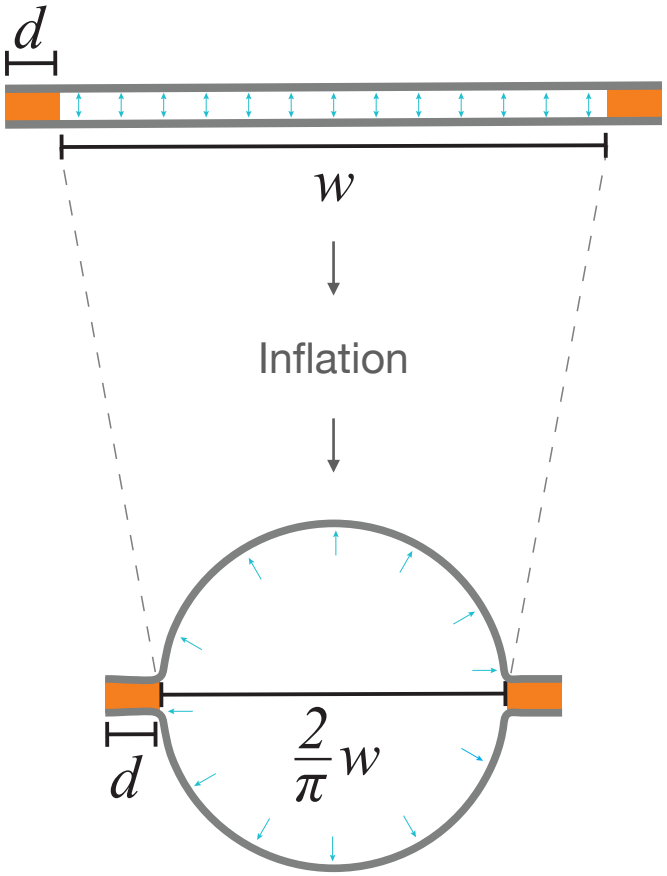
GEOMETRIC ABSTRACTION/HOMOGENIZATION

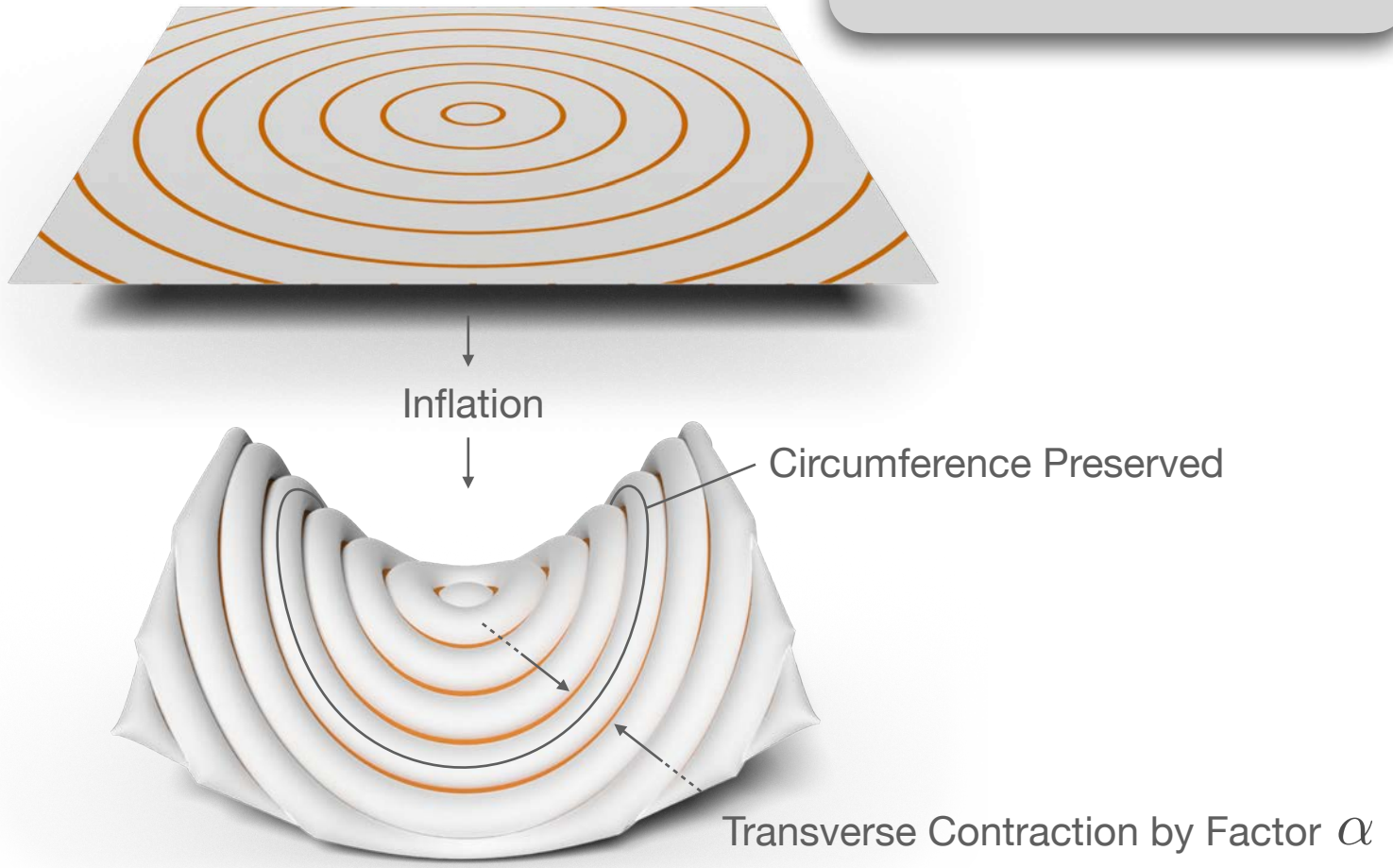




GEOMETRIC ABSTRACTION/HOMOGENIZATION

Homogenized Viewpoint
Inflation assigns a new metric to the flat sheet!



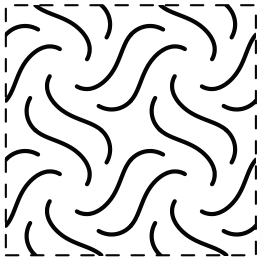


CELL PROBLEM FOR INFLATABLES (WITHOUT BENDING)

$$\overline{\psi}(p) = \frac{1}{|Y|} \min_{\substack{\overline{F}, \\ \boldsymbol{\omega} \text{ periodic}}} E_{\text{mem}}[\Phi] - p\text{Vol}(\Phi(\Omega))$$

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

Quantity of interest for shape morphing: $\overline{F}^*(p)$



fusing curves



simulated tilings

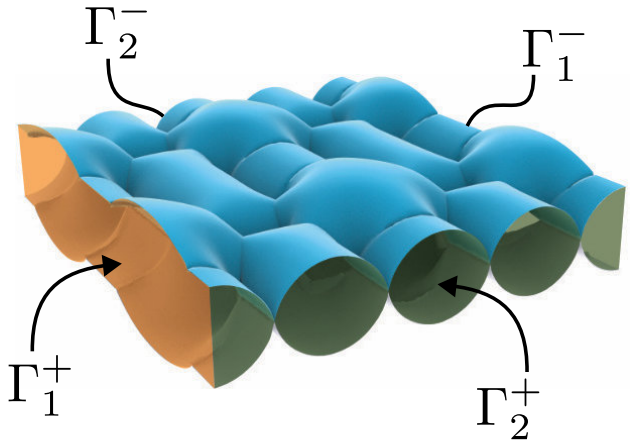


unit cell

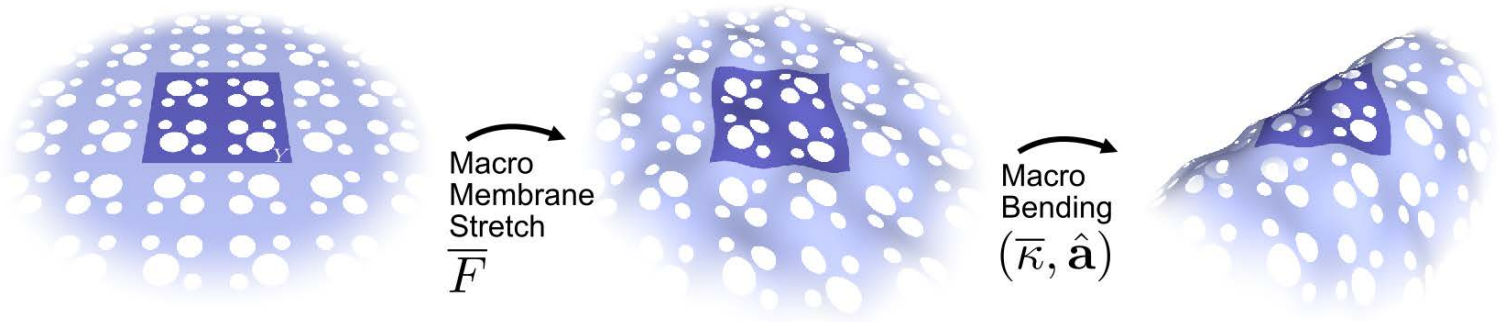


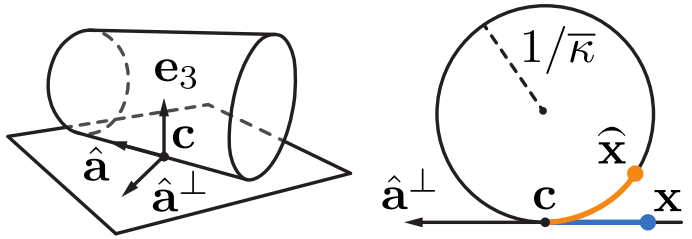
fabricated tilings

What is the enclosed volume?



CELL PROBLEM FOR INFLATABLES (WITH BENDING)



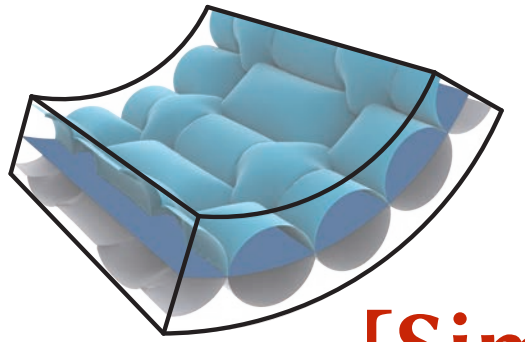


Wrap the tiling onto a cylinder after in-plane stretch:



$$\overline{\psi}(p) = \frac{1}{|Y|} \min_{\substack{\overline{F},\overline{\kappa},\hat{\mathbf{a}} \\ \boldsymbol{\omega} \text{ periodic}}} E_{\text{mem}}[\Phi] - p\text{Vol}(\Phi(\Omega))$$

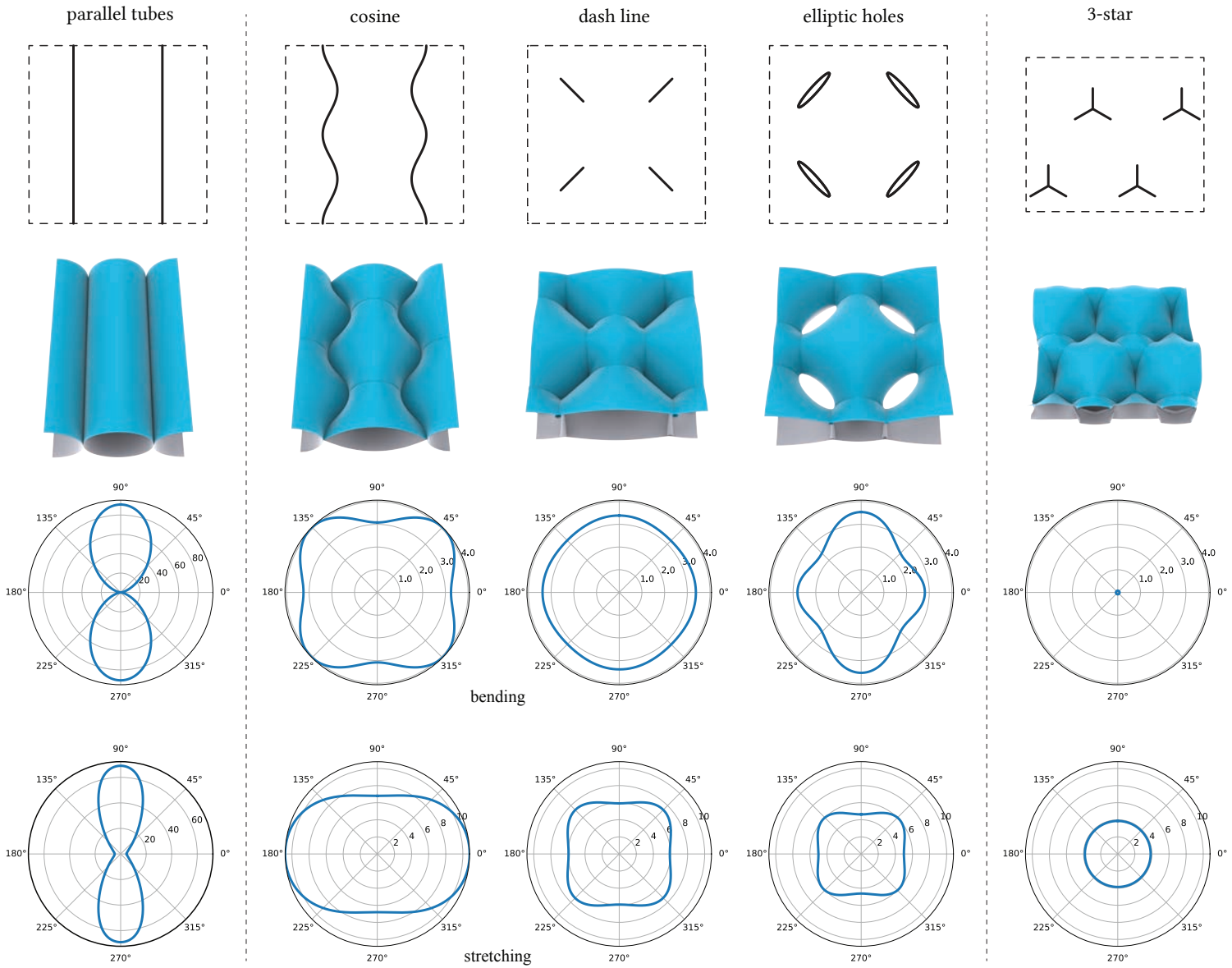
$$\Phi(\mathbf{X}) = \text{bend}_{\overline{\kappa},\hat{\mathbf{a}}}(\overline{F}\mathbf{X} + \boldsymbol{\omega}(\mathbf{X}))$$



Double curvature would break tiling

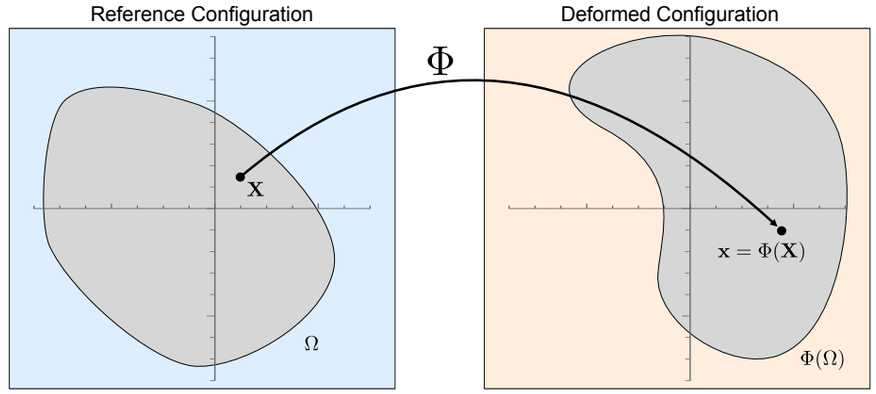
[Simulation Demo]

HOMOGENIZATION RESULTS



Solving the Cell Problems Computationally

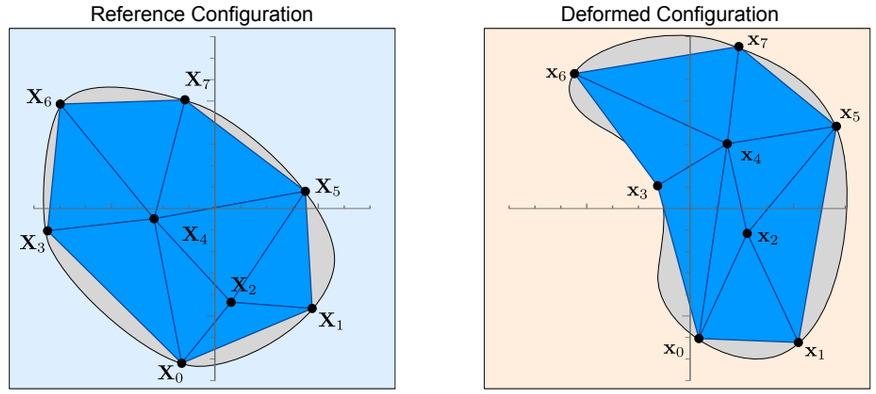
Discretizing the Hyperelastic Model

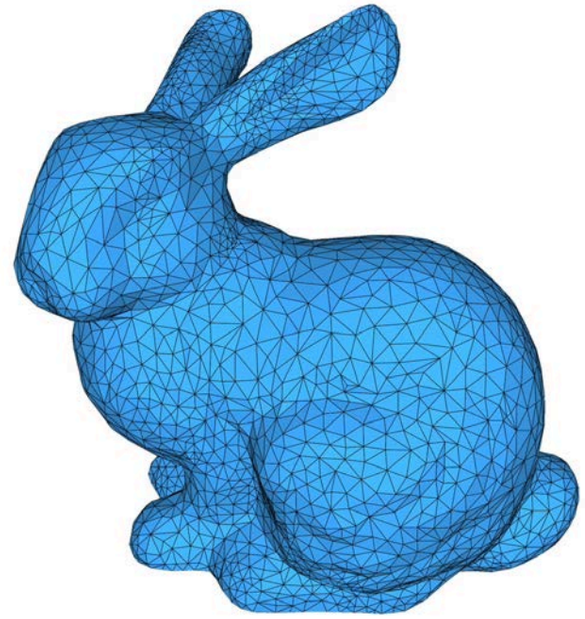


$$E_{\text{elastic}}[\Phi] := \int_{\Omega} \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X}$$

- To minimize this energy on a computer we need to *discretize* (construct finite-dimensional approximations of):
  - The shape Ω.
  - The (infinite dimensional) map Φ.
- In simple cases, integrals can be computed *exactly* with a finite sum over just a few "quadrature" (sample) points; in general cases, this is still sufficiently accurate.

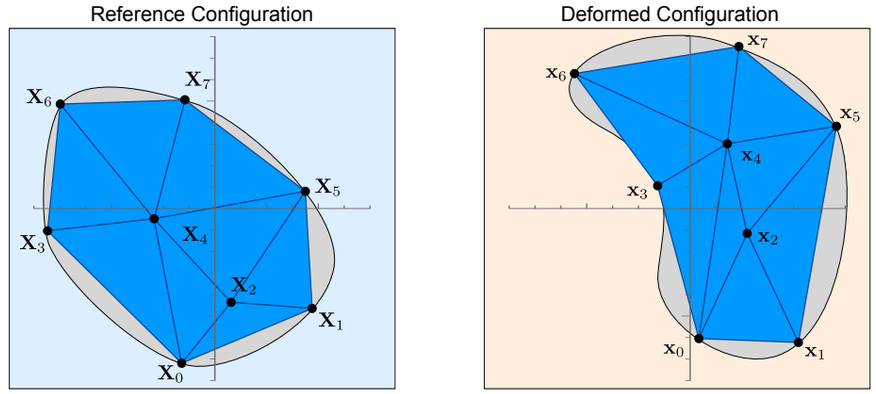
Brief Overview of the Finite Element Method





- Approximate rest-shape *geometry* Ω using a mesh Ω_h.
  - Simplest examples: triangle mesh in 2D, tetrahedral mesh in 3D.
  - Can use *curved* triangles/tets for better accuracy (meshing gets trickier)
  - Also hexahedral meshes, B-splines, etc.
- Lagrange elements: approximate $\Phi(\mathbf{X})$ as an interpolation of the deformed positions $\mathbf{x}_i$ of a finite set of *nodes*.
  - Simplest example: linearly interpolate the corner vertex positions of a triangle/tet mesh.
  - More accurate: quadratic interpolation with additional edge midpoint nodes.

FEM Deformation Map



- The finite-dimensional deformation map will take the form:

$$\mathbf{\Phi}_h(\mathbf{X}) = \sum_i \mathbf{x}_i \phi_i(\mathbf{X}),$$

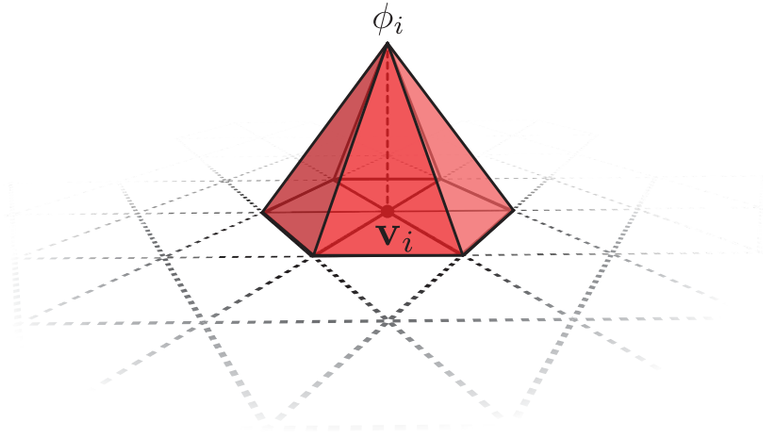
  where $\mathbf{x}_i$ is a deformed mesh node position, and ϕ_i is a scalar-valued *basis function* satisfying:

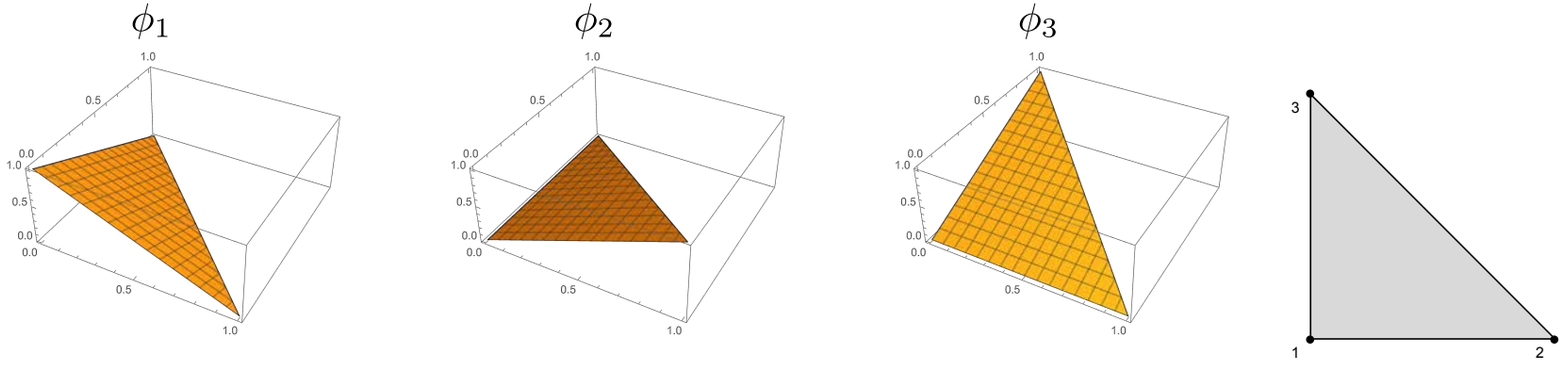
$$\phi_i(\mathbf{X}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \quad \Longrightarrow \quad \mathbf{\Phi}(\mathbf{X}_j) = \sum_i \mathbf{x}_i \phi_i(\mathbf{X}_j) = \mathbf{x}_j$$

- In other words, we are using the Lagrange polynomial basis, hence the term “Lagrange element.”

Piecewise Linear Basis

- The simplest basis functions are *piecewise linear*.
  - Also known as barycentric coordinate functions
  - Within a triangle: the unique linear function taking value $\mathbf{1}$ at one vertex and $\mathbf{0}$ at the others.
  - Simple to work with, but does not strike an optimal balance between accuracy and computational cost.





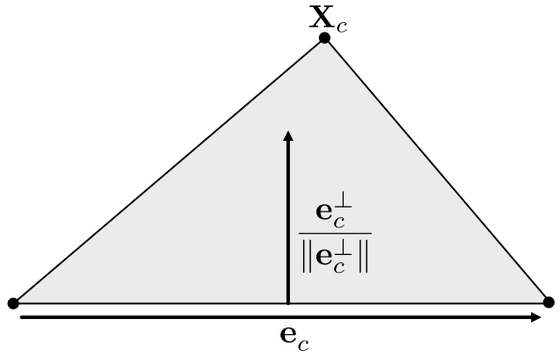
- Especially convenient since the gradients are *constant* over each element!

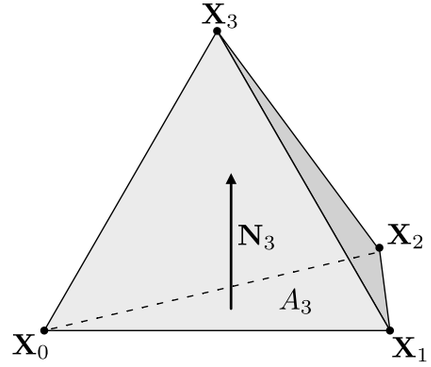
$$\int_\Omega \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X} = \sum_e \left(\int_{\Omega_e} \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X} \right) = \boxed{\sum_e \Psi\left(\nabla\Phi|_e\right)\mathtt{Vol}(e)}$$

This formula is ***exact*** regardless of Ψ!

Gradients of Linear Basis Functions

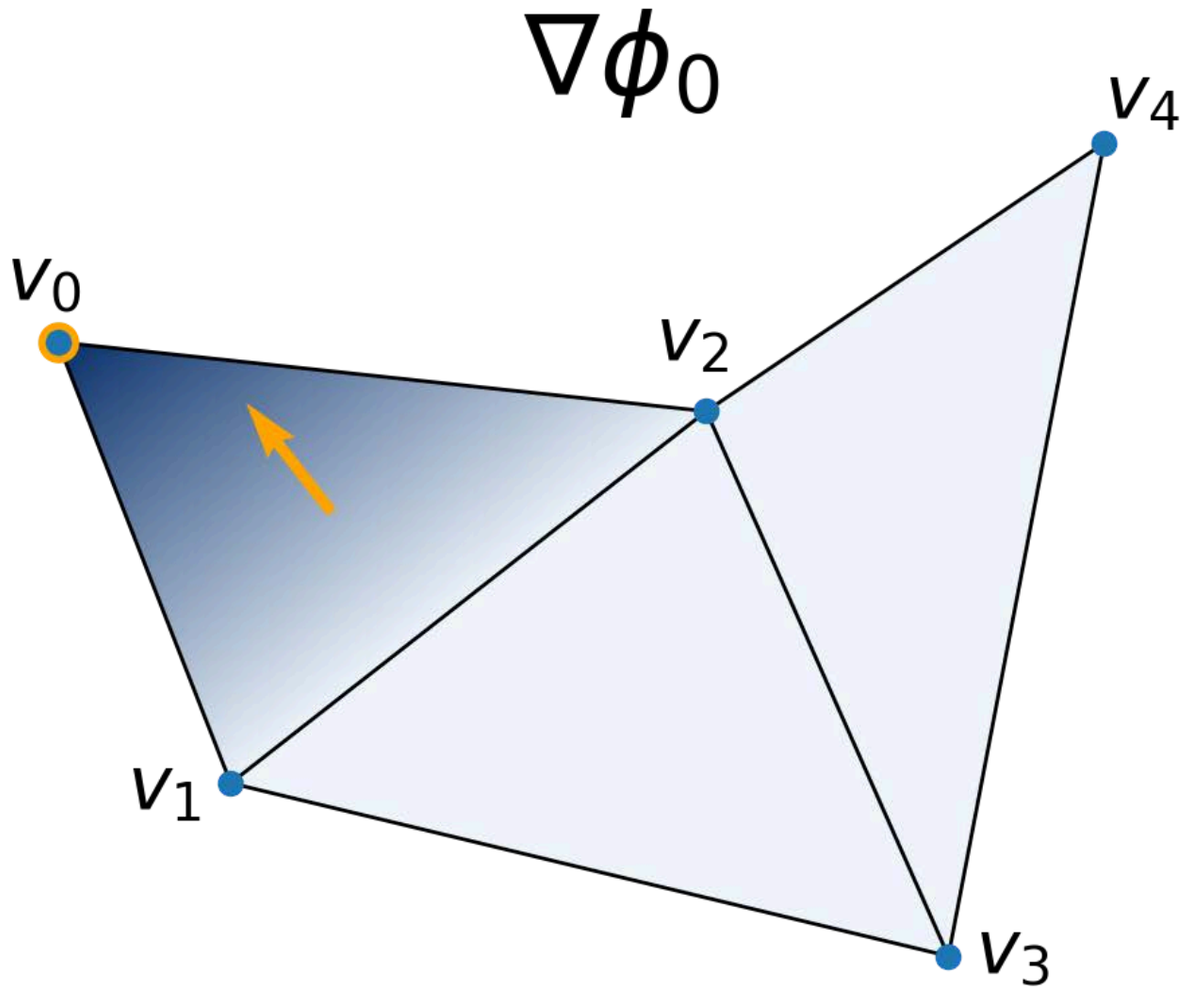
- Within a particular tri or tet element, the gradient of the linear basis function associated with a corner can be found with simple geometric arguments
  - Function is constant along lines (planes) parallel to opposite edge (face)…
    and increases from 0 to 1 throughout the height of the element:



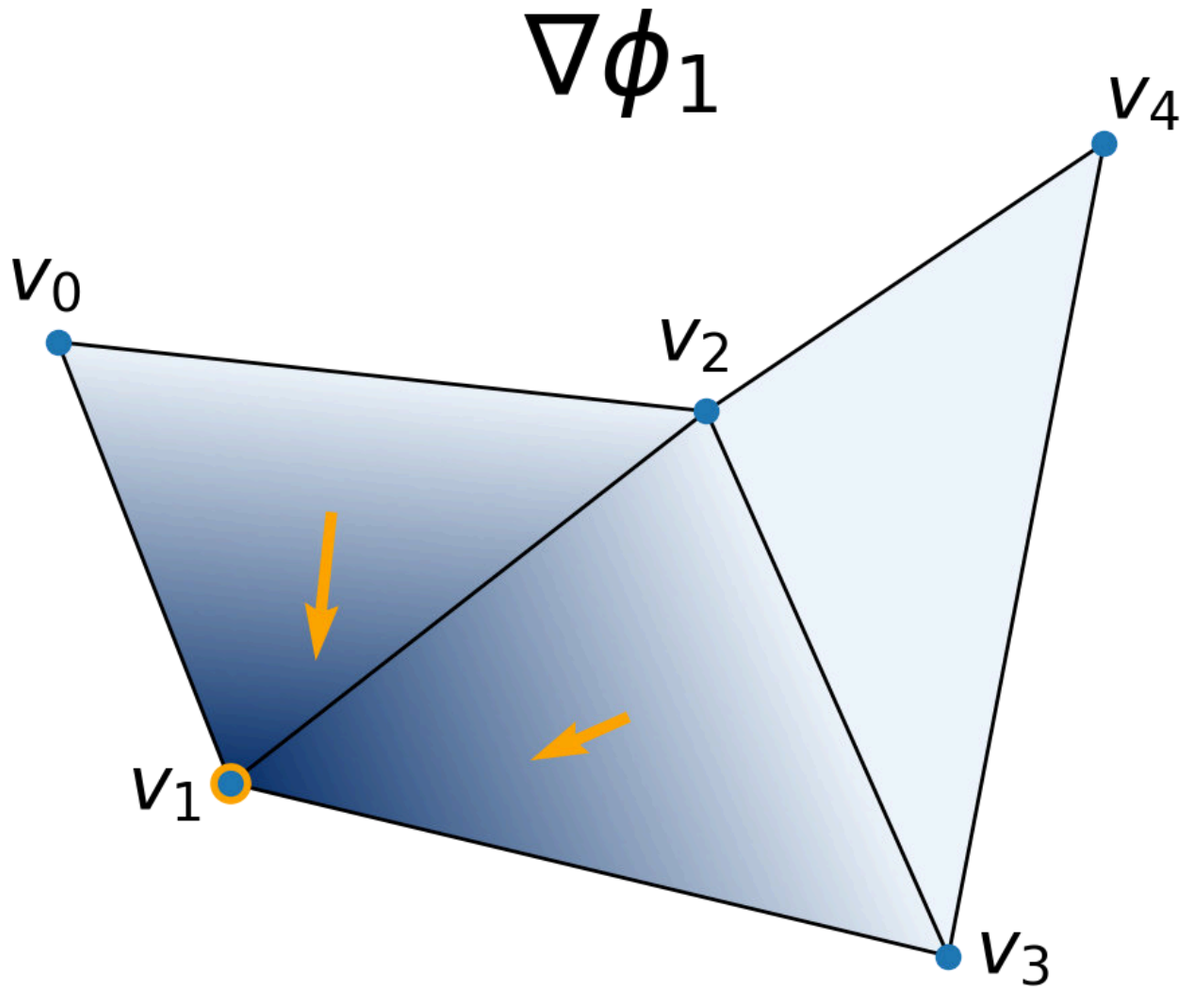


- $\nabla\phi_c^{\text{tri}} = \frac{1}{h_c}\frac{\mathbf{e}_c^\perp}{\|\mathbf{e}_c^\perp\|} = \frac{\mathbf{e}_c^\perp}{2A} \quad \left(A = \frac{1}{2}\|\mathbf{e}_c\|h_c\right)$
- $\nabla\phi_3^{\text{tet}} = \frac{\mathbf{N}_3}{h_3} = \frac{\mathbf{N}_3 A_3}{3V}, \quad \mathbf{N}_3 A_3 = \frac{1}{2}(\mathbf{X}_1 - \mathbf{X}_0) \times (\mathbf{X}_2 - \mathbf{X}_0).$

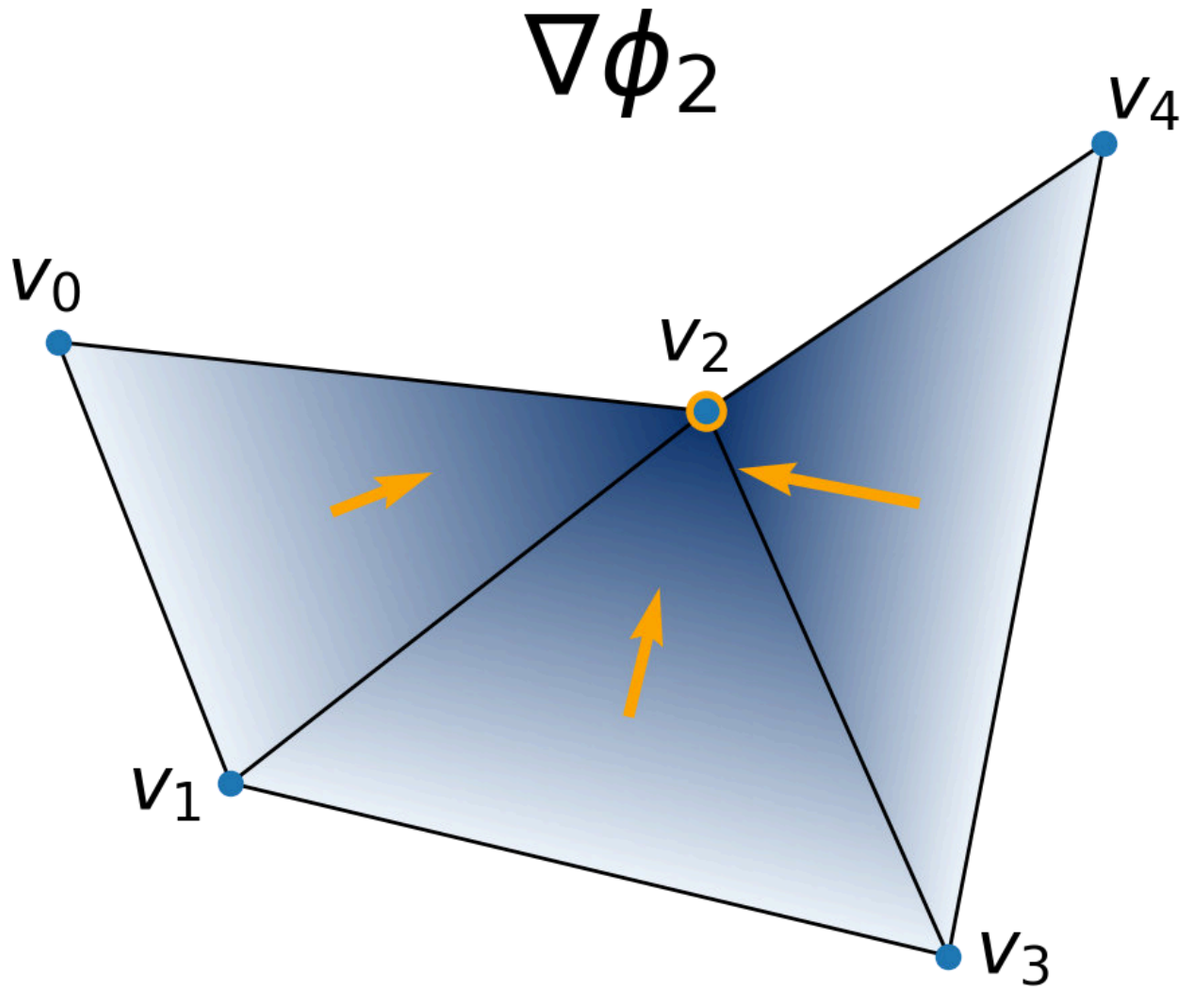
Piecewise Linear Basis Gradients: Example



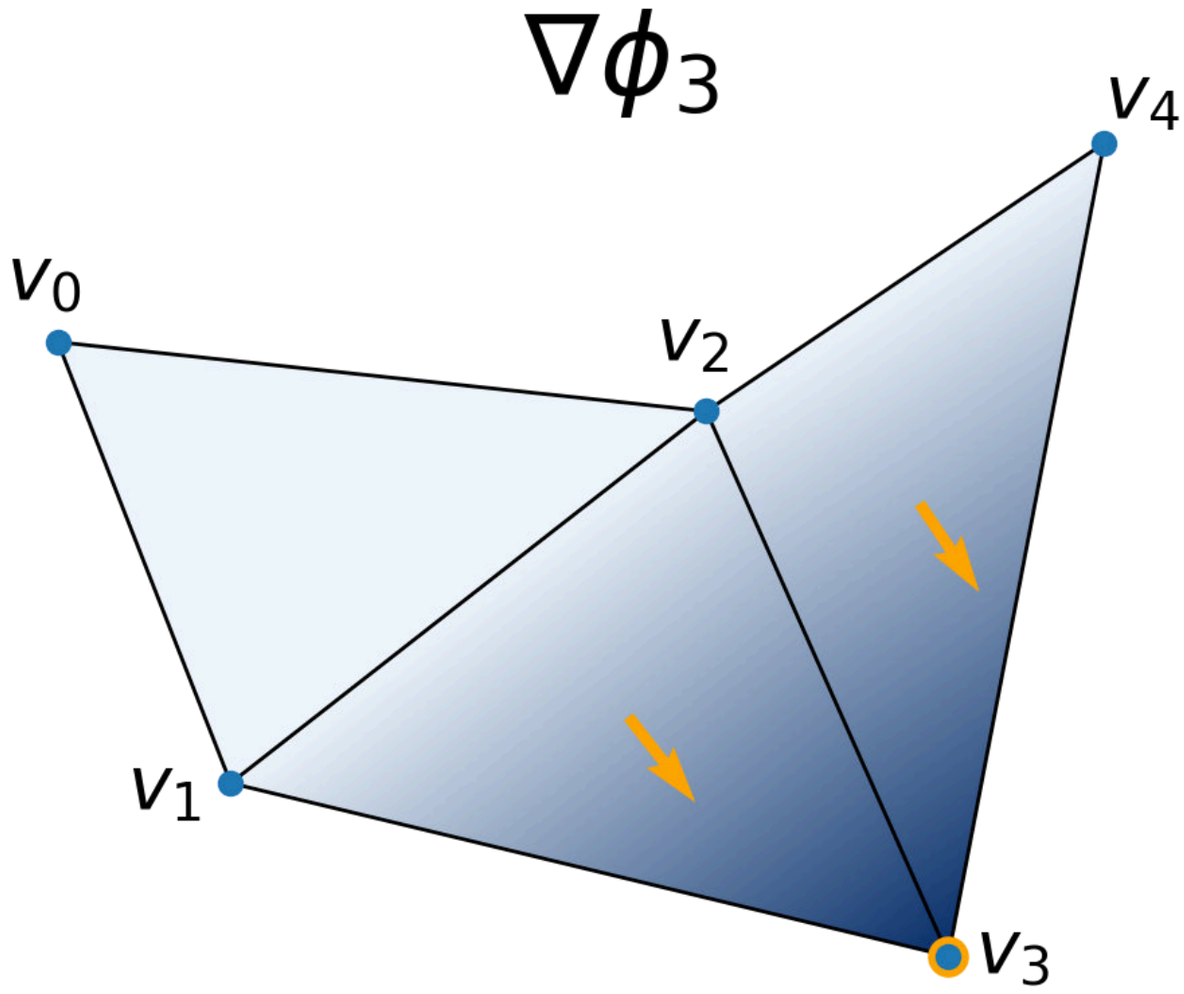
Piecewise Linear Basis Gradients: Example



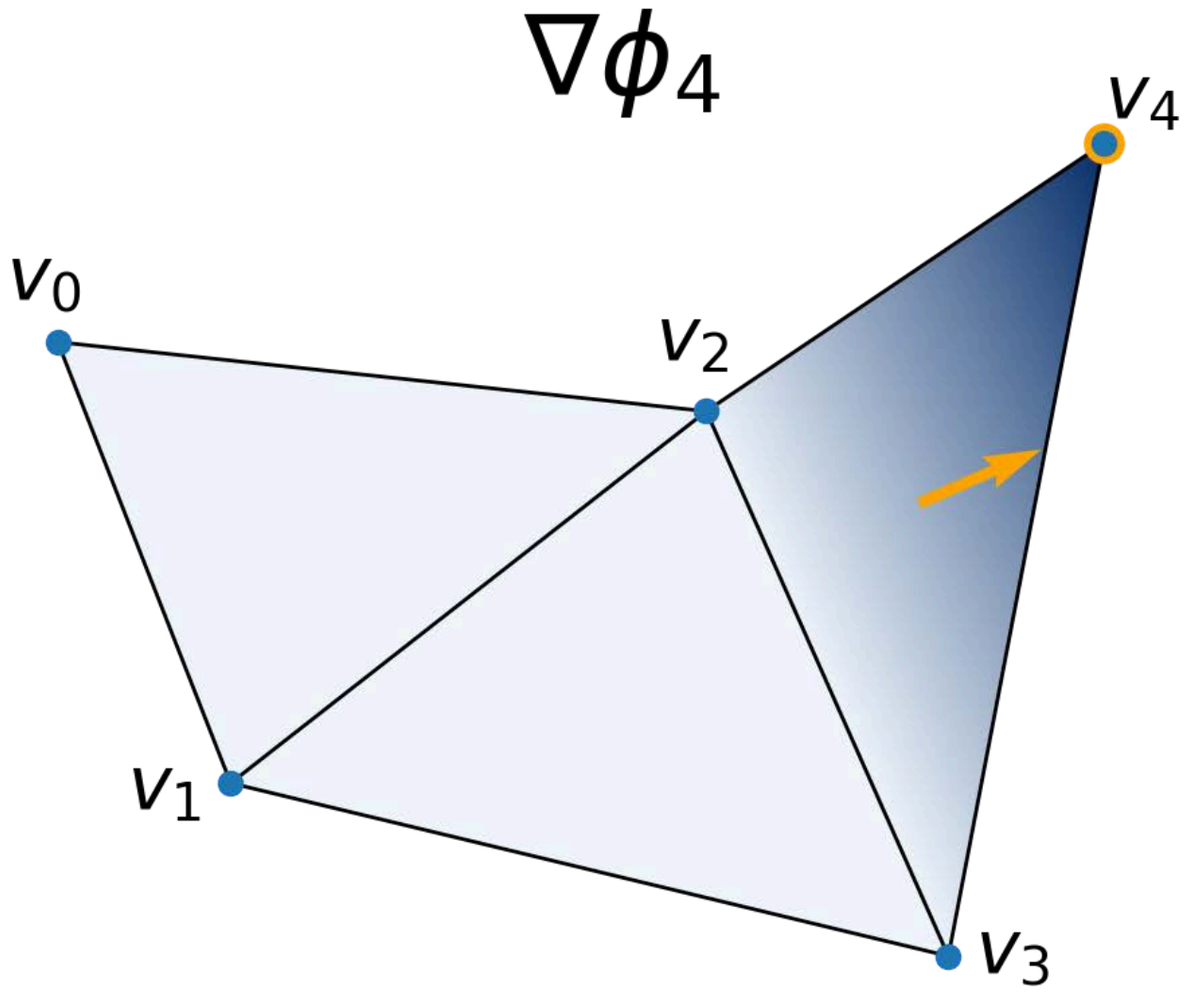
Piecewise Linear Basis Gradients: Example



Piecewise Linear Basis Gradients: Example



Piecewise Linear Basis Gradients: Example



Discrete Energy Minimization

Recall the general simulation problem formulation:

$$\min_{\Phi \text{ admissible}} E_{\text{elastic}}[\Phi] + \mathcal{L}[\Phi],$$

- We have now established how $E_{\text{elastic}}[\Phi]$ can be discretized into $E_h^{\text{elastic}}(\mathbf{x})$.
- The loads, if any, likewise can be discretized into a function $\mathcal{L}_h(\mathbf{x})$.
- The full problem then becomes a finite-dimensional optimization problem over $\mathbf{x}$:

$$\min_{\mathbf{x} \text{ admissible}} E(\mathbf{x})$$

- We need fast simulation methods for computational design, and Newton-type solvers are the preferred choice.

Newton's Method

- Newton's method for optimization iteratively improves an approximation $\mathbf{x}^i$ to a minimizer $\mathbf{x}^*$, starting from some initial guess $\mathbf{x}^0$.
- It works by constructing a local quadratic approximation to $E(\mathbf{x})$ around $\mathbf{x}^i$:

$$E(\mathbf{x}^i + \mathbf{d}) \approx E(\mathbf{x}^i) + \nabla E(\mathbf{x}^i) \cdot \mathbf{d} + \frac{1}{2}\mathbf{d} \cdot H(\mathbf{x}^i)\mathbf{d} \quad \left(H(\mathbf{x}^i) := \frac{\partial^2 E}{\partial \mathbf{x}^2}(\mathbf{x}^i) \right)$$

- Finding the critical point of this approximation is easy:

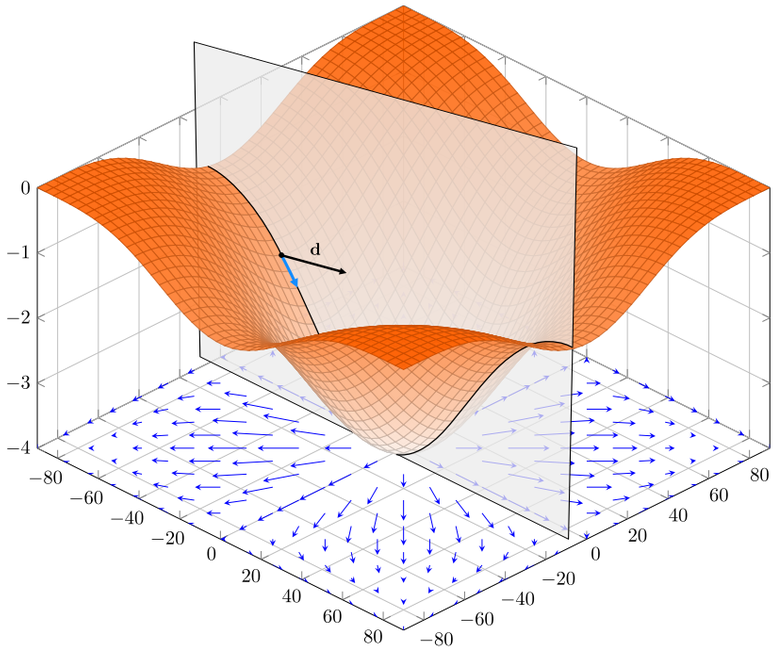
$$H(\mathbf{x}^i)\mathbf{d}^* = -\nabla E(\mathbf{x}^i)$$

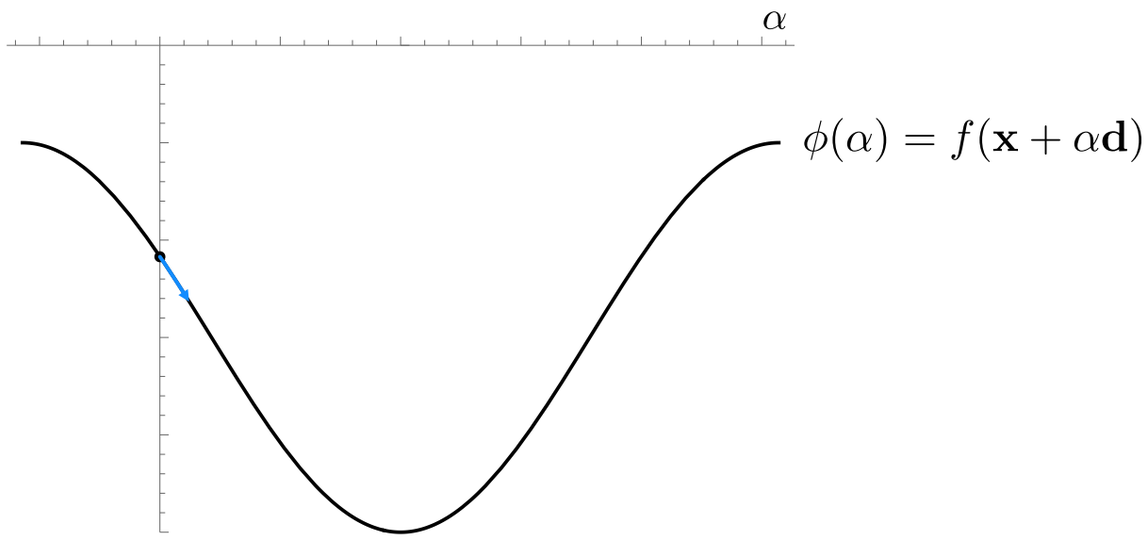
- This is a linear system that is solved for the *Newton step* $\mathbf{d}$, which is used to produce

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha\mathbf{d}$$

How far to step?

- The Taylor expansion suggests the full step $\mathbf{d}$ (= 1) is optimal, but this is not true if higher-order terms are significant.
- Finding the best α is a 1D optimization problem (regardless of n):



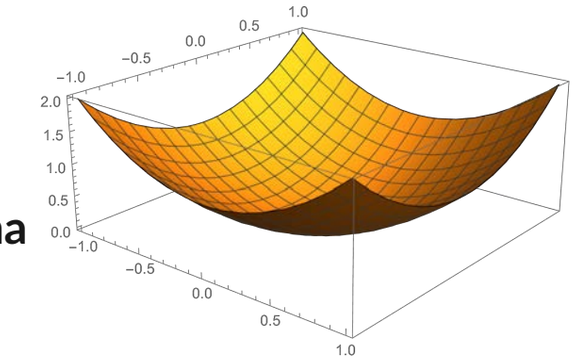


- A line search is used to quickly obtain a step length that guarantees sufficient progress to eventually convergence.

Solving the Newton Equations

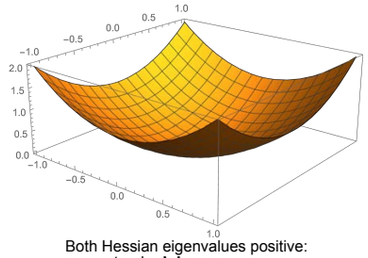
$$H\mathbf{d} = -\nabla f$$

- This will be the *time and memory bottleneck* of the optimization.
- Useful properties of the system:
  - Symmetric ($H^\top = H$)
  - Positive definite **in a neighborhood around local minima**
  - Sparse (for large finite element problems)

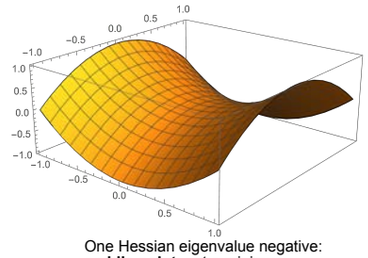


- We can use efficient linear system solvers
  - Sparse Cholesky factorization $H = LL^\top$
  - Conjugate gradient method ("Newton CG")
    - Iterative method with extremely low memory requirements.
    - Building H is not even necessary: CG just needs products $H\mathbf{v}$ (i.e., directional derivatives)
    - Requires a preconditioner to be practical for ill-conditioned H (like ours…)

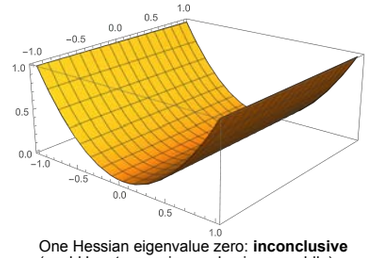
Caveat: Hessian Indefiniteness due to Nonconvexity



Both Hessian eigenvalues positive: guaranteed **minimum**.



One Hessian eigenvalue negative: **saddle point**, not a minimum.



One Hessian eigenvalue zero: **inconclusive** (could be strong min, weak min, or saddle).

- To guarantee $-H^{-1}\nabla f$ is a descent direction, we must detect and modify indefinite H

$$H \text{ indefinite} \quad \rightarrow \quad \tilde{H} \text{ positive definite}$$

- Brute-force strategy:
  - Compute full *eigendecomposition*: $H = Q\Lambda Q^\top$.
  - Modify eigenvalues to $\tilde{\Lambda}$ to be positive.
    - For example: $\tilde{\lambda}_i = \mathbf{max}(\lambda_i, \epsilon),$ with $0 < \epsilon \ll 1.$ (This actually minimizes $\left\|\tilde{H} - H\right\|_F^2$ subject to $\lambda_{\min}(\tilde{H}) \geq \epsilon$!)
    - or: $\tilde{\lambda}_i = |\lambda_i|$ (Flip $\mathbf{d}$'s components along neg. curvature directions.)
  - Construct the modified Hessian: $\tilde{H} = Q\tilde{\Lambda}Q^\top$.
- Efficient version: do it only when necessary (failed Cholesky factorization) and apply it at the level of individual element Hessians

Discrete FEM Energy and Gradient

- With piecewise linear FEM, discrete elastic energy is:

$$E_{\text{elastic}}(\mathbf{x}) = \sum_e \Psi\left(\sum_{i \in e} \mathbf{x}_i \otimes \nabla \phi_i|_e\right) \texttt{Vol}(e).$$

  - Can be evaluated efficiently by looping over triangles/tets e and accessing corner positions.

- What about the gradient (forces)? The c^{th} component of (negative) elastic force on k^{th} vertex is:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}_{k,c}} E_{\text{elastic}}(\mathbf{x}) &= \sum_e \Psi'\left(\sum_{i \in e} \mathbf{x}_i \otimes \nabla \phi_i|_e\right) : \sum_{i \in e}\left(\frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{k,c}} \otimes \nabla \phi_i|_e\right) \texttt{Vol}(e) \\ &= \boxed{\sum_e \Psi' : (\mathbf{e}_c \otimes \nabla \phi_k|_e) \texttt{Vol}(e).}\end{aligned}$$

- Collecting all of these components, the gradient with respect to the k^{th} vertex is:

$$\frac{\partial E}{\partial \mathbf{x}_k} = \boxed{\sum_e \Psi' \nabla \phi_k|_e \texttt{Vol}(e).}$$

Hessian and Per-element Assembly

$$E_{\text{elastic}}(\mathbf{x}) = \sum_e \Psi\left(\sum_{i \in e} \mathbf{x}_i \otimes \nabla \phi_i|_e\right) \texttt{Vol}(e)$$

$$\frac{\partial E_{\text{elastic}}}{\partial \mathbf{x}_{k,c}} E_{\text{elastic}}(\mathbf{x}) = \sum_e \Psi' : (\mathbf{e}_c \otimes \nabla \phi_k|_e) \texttt{Vol}(e)$$

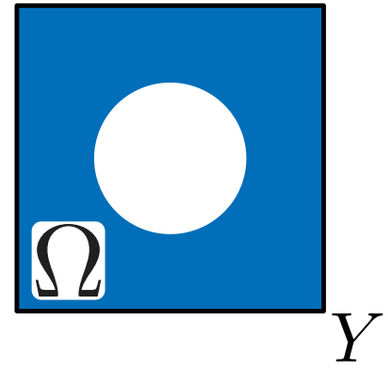
- Differentiating again with respect to d^{th} component of vertex l obtains a Hessian entry:

$$\boxed{\frac{\partial^2}{\partial \mathbf{x}_{k,c} \partial \mathbf{x}_{l,d}} E_{\text{elastic}}(\mathbf{x}) = \sum_e (\mathbf{e}_c \otimes \nabla \phi_k|_e) : \Psi'' : (\mathbf{e}_d \otimes \nabla \phi_l|_e) \texttt{Vol}(e)}$$

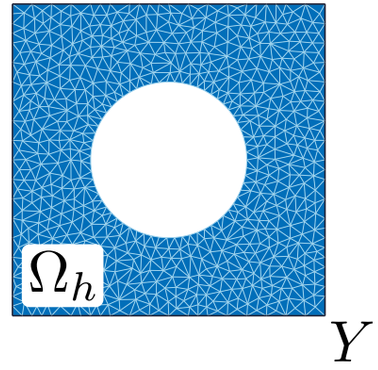
- This can be evaluated efficiently by looping over elements and assembling the contributions to the global Hessian matrix.

Finite Element Cell Problem

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



- Solving the cell problem is now a straightforward application of the FEM machinery we have developed:
  - Discretize the periodic cell Y with a mesh of triangles/tets.
  - Approximate $\boldsymbol{\omega}$ by interpolating fluctuation displacements defined at the nodes.
  - Solve the discrete minimization problem using Newton's method.
- How to handle periodic boundary conditions:
  - Generate a periodic mesh.
  - Node identification: assign same degrees of freedom to opposite sides of the periodic cell.



Tangent Stiffness Formulas

- In addition to solving the cell problem, we can use it to extract *tangent stiffnesses*:
  - Tangent elasticity tensor $\overline{\psi}''(\overline{F})$
  - Bistable auxetic stiffness $\frac{\partial^2}{\partial \lambda^2}\overline{\psi}$
  - Inflatable patch bending stiffness $\frac{\partial^2}{\partial \kappa^2}\overline{\psi}$.
- It is simplest to derive these quantities in the discrete formulation and decompose the variables into relaxed and imposed sets:

$$E(\mathbf{x}_\mu, \mathbf{x}_M),$$

  where $\mathbf{x}_\mu$ represents, e.g., $\boldsymbol{\omega}$ and $\mathbf{x}_M$ could be $\overline{F}$.
- The solution to the cell problem for a fixed setting of $\mathbf{x}_M$ is:

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

  and we seek the stiffness $K_M := \frac{\partial^2}{\partial {\mathbf{x}_M}^2} E(\mathbf{x}_\mu^*(\mathbf{x}_M), \mathbf{x}_M)$

Tangent Stiffness Formulas

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

- Differentiating once:

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

  where the first term vanishes by optimality of $\mathbf{x}_\mu^*$ (envelope theorem).

- Differentiating again:

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

- The equilibrium derivative can no longer be neglected. It is obtained using the implicit function theorem, noting the optimality conditions that implicitly define $\mathbf{x}_\mu^*(\mathbf{x}_M)$:

$$\frac{\partial E}{\partial \mathbf{x}_\mu}(\mathbf{x}_\mu^*(\mathbf{x}_M), \mathbf{x}_M) = 0 \quad \Longrightarrow \quad \frac{\partial^2 E}{\partial {\mathbf{x}_\mu}^2} \frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M} + \frac{\partial^2 E}{\partial \mathbf{x}_\mu \partial \mathbf{x}_M} = 0.$$

Tangent Stiffness Formulas

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

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

$$\frac{\partial^2 E}{\partial {\mathbf{x}_\mu}^2}\frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M} + \frac{\partial^2 E}{\partial \mathbf{x}_\mu \partial \mathbf{x}_M} = 0$$

- This is a linear system for $\frac{\partial \mathbf{x}_\mu^*}{\partial \mathbf{x}_M}$, involving the same sparse Hessian matrix $\frac{\partial^2 E}{\partial {\mathbf{x}_\mu}^2}$ as in the Newton solve for $\mathbf{x}_\mu^*$.
- Formally:

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

(but don't actually invert $\frac{\partial^2 E}{\partial {\mathbf{x}_\mu}^2}$; solve linear systems instead using, *e.g.*, Cholesky).

Optimal Design

Introduction to Sensitivity Analysis

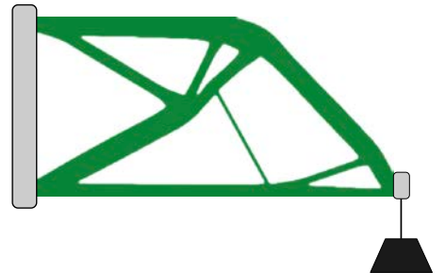
High-level structure of a inverse design problem:

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

$\mathbf{p}$: design parameters,
$\mathbf{x}$: state variables

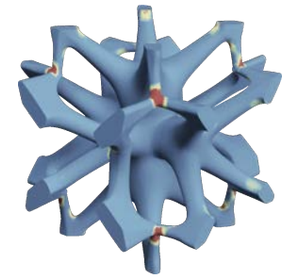
- Examples:
  - Compliance minimization (maximize stiffness):

$$J(\mathbf{x},\mathbf{p}) := \int_{\Omega(\mathbf{p})} \mathbf{f}^{\text{ext}} \cdot \mathbf{u}\,\mathrm{d}\mathbf{X}$$



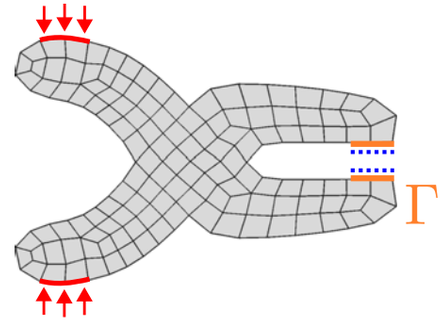
  - L_p stress minimization:

$$J(\mathbf{x},\mathbf{p}) := \int_{\Omega(\mathbf{p})} j^p(\sigma)\,\mathrm{d}\mathbf{X}, \qquad j(\sigma) \in \left\{\sqrt{\sigma:\sigma},\ \sigma_{\text{vm}},\ \lambda_{\max}(\sigma),\ \dots\right\}$$



  - Deformation goal:

$$J(\mathbf{x},\mathbf{p}) := \int_{\Gamma} \left\| \Phi(\mathbf{X}) - \mathbf{x}^{\text{tgt}} \right\|^2 \mathrm{d}\mathbf{X}$$


Γ

Introduction to Sensitivity Analysis

High-level structure of a inverse design problem:

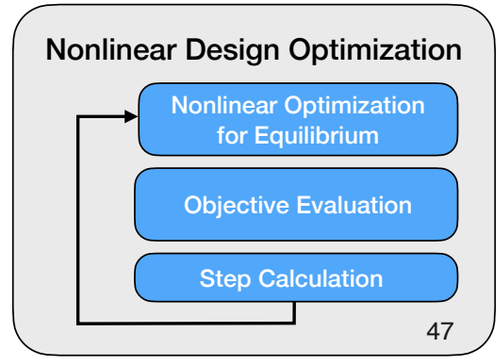
$$\min_{\mathbf{x},\,\mathbf{p}} J(\mathbf{x},\mathbf{p}), \quad \text{s.t.} \quad \mathbf{x} = \operatorname*{argmin}_{\tilde{\mathbf{x}}} E(\tilde{\mathbf{x}},\mathbf{p})$$

$\mathbf{p}$ are the design parameters,
$\mathbf{x}$ the state variables.

- This "all at once" formulation has advantages, but it:
  - is a large constrained optimization problem;
  - cannot leverage existing fast simulation code; and
  - fails to distinguish between stable and unstable equilibria (if the constraint is implemented as the nonlinear system $\frac{\partial E}{\partial \mathbf{x}} = 0$).
- Instead, the simulation can be used to eliminate the state variable $\mathbf{x}$, posing a reduced form of the problem:

$$\min_{\mathbf{p}} \tilde{J}(\mathbf{p}) \quad \text{where} \quad \tilde{J}(\mathbf{p}) := J(\mathbf{x}^*(\mathbf{p}),\mathbf{p})$$

$$\mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\tilde{\mathbf{x}}} E(\tilde{\mathbf{x}},\mathbf{p}).$$



Introduction to Sensitivity Analysis

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

- The design derivative is:

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

- As in tangent stiffness computation, equilibrium derivative $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$ is obtained via the implicit function theorem:

$$\frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}), \mathbf{p}) = 0 \quad \xRightarrow{\frac{\partial}{\partial \mathbf{p}}} \quad \frac{\partial^2 E}{\partial \mathbf{x}^2} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} = 0.$$

- Thus, formally:

$$\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}.$$

Introduction to Sensitivity Analysis

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

$$\frac{\partial \tilde{J}}{\partial \mathbf{p}} = \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}, \qquad \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$$

- This form of the equations suggests the "forward sensitivity" approach:
  first solve for $\frac{\partial \mathbf{x}^*}{\partial \mathbf{p}}$, then substitute into $\frac{\partial \tilde{J}}{\partial \mathbf{p}}$.
- Forward sensitivity involves solving one linear system per design parameter in $\mathbf{p}$
  of which can be thousands, millions, or even billions!
- With just a few simulation-dependent quantities of interest (*e.g.*, only J),
  the *adjoint method* is vastly more efficient.

Adjoint Sensitivity Analysis

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

$$\frac{\partial \tilde{J}}{\partial \mathbf{p}} = \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}, \qquad \frac{\partial \mathbf{x}^*}{\partial \mathbf{p}} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$$

With just a few simulation-dependent quantities of interest (*e.g.*, only J), the *adjoint method* is vastly more efficient:

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial \mathbf{p}} &= -\frac{\partial J}{\partial \mathbf{x}} \left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}} \\ &= -\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}, \end{aligned}$$

where $\boldsymbol{\lambda}^\top = \dfrac{\partial J}{\partial \mathbf{x}} \left[\dfrac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1}$ $\iff$ $\boxed{\left[\dfrac{\partial^2 E}{\partial \mathbf{x}^2}\right]^\top \boldsymbol{\lambda} = \left(\dfrac{\partial J}{\partial \mathbf{x}}\right)^\top.}$

This is the *adjoint equation* solved for the *adjoint state* $\boldsymbol{\lambda}$.

Adjoint Sensitivity Analysis

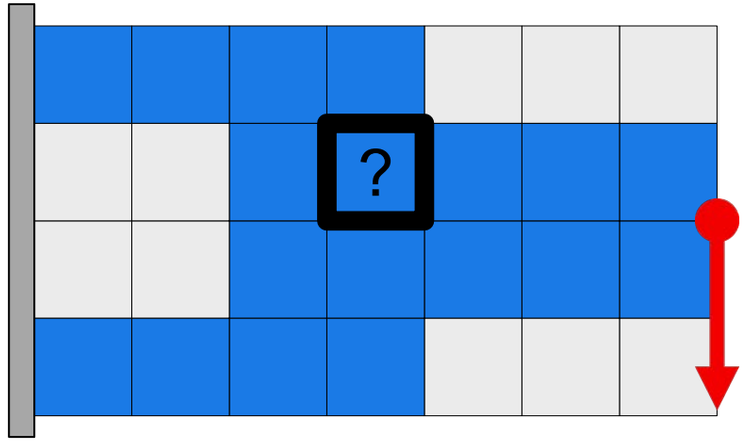
$$\min_{\mathbf{p}} \tilde{J}(\mathbf{p}), \quad \text{where} \quad \tilde{J}(\mathbf{p}) := J(\mathbf{x}^*(\mathbf{p}), \mathbf{p}), \quad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\tilde{\mathbf{x}}} E(\tilde{\mathbf{x}}, \mathbf{p})$$

$$\frac{\partial \tilde{J}}{\partial \mathbf{p}} = \frac{\partial J}{\partial \mathbf{p}} - \boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} \qquad \left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^\top \boldsymbol{\lambda} = \left(\frac{\partial J}{\partial \mathbf{x}}\right)^\top$$

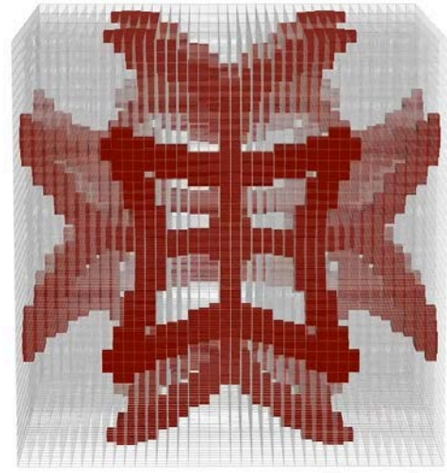
- Forward vs. adjoint sensitivity analysis is analogous to forward- vs. reverse-mode automatic differentiation (*i.e.*, backprop).
- What derivatives must be calculated?
  - $\frac{\partial J}{\partial \mathbf{x}}$ and $\frac{\partial J}{\partial \mathbf{p}}$: generally straightforward to compute by hand or via automatic differentiation.
  - $\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$: depends on simulation model and design parameterization.
    Also can be computed analytically or via automatic differentiation.
    Autodiff must not be done with “brute-force” approach (leverage sparsity of FEM structure).

Design Representations

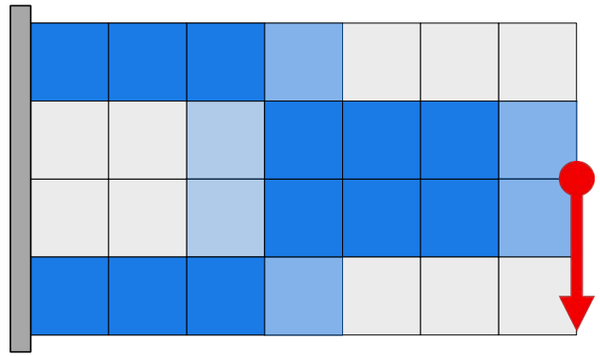
Density-based Topology Optimization



- Superimpose regular grid on design domain.
- Decide whether to fill each grid cell with material.
  - Intractable combinatorial optimization when posed over $\{0, 1\}$ binary variables.
  - Smooth relaxation: assign a *density* variable $\rho_e \in [0, 1]$ for each element e (attenuating stiffness); apply gradient-based optimization.
  - Incorporate *penalties* to push ρ_e to 0 or 1 at the end.

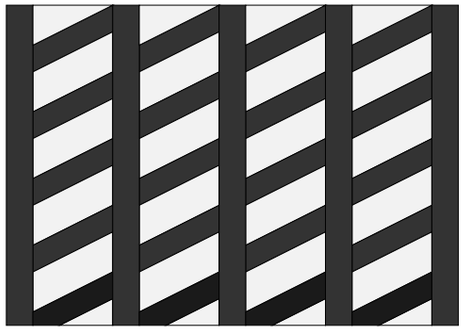


Density-based Topology Optimization



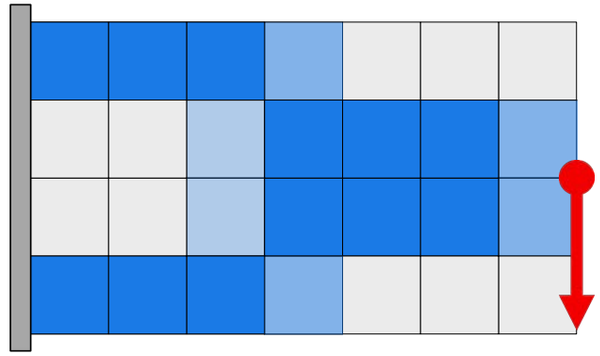
What do intermediate densities $\rho_e \in (0,1)$ mean?

- Homogenization method:
  - Interpret material with density $\rho_e \in (0,1)$ as a maximally stiff microstructure with that volume fraction:



  - Closed-form expressions for the optimal moduli obtained via *sequential laminates*.
  - Still a rather complicated relationship between ρ_e and C.

Density-based Topology Optimization



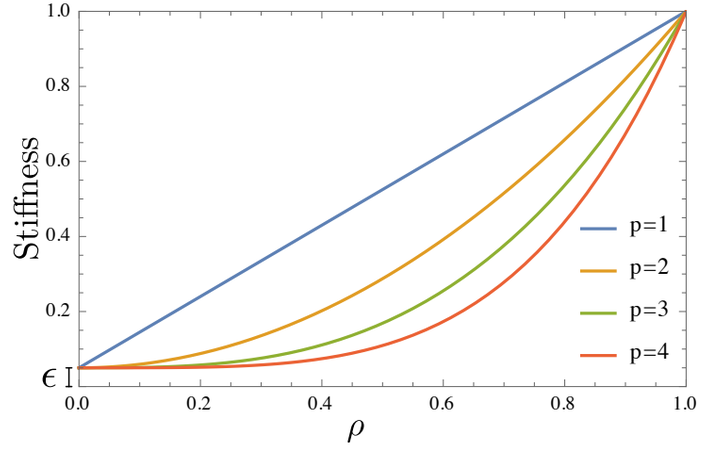
What do intermediate densities $\rho_e \in (0, 1)$ mean?

- SIMP (Solid Isotropic Material with Penalisation) heuristic
  - Final design is usually pushed to binary, so intermediate stiffnesses need not be physically meaningful.
  - Given elasticity tensor C^{base} of the fabrication material, use the interpolation rule:

$$C(\rho) := (\epsilon + (1 - \epsilon)\rho^p)C^{\text{base}},$$

ϵ is a small constant ($\sim 10^{-6}$) to prevent singular K.

  - Increasing power p makes intermediate densities "inefficient" for the compliance objective, encouraging a binary design.



- Design variables $\mathbf{p}$ determine density field via chain of *filters* (length scale enforcement via smoothing, *etc.*) $\rho_e = f_e(\mathbf{p})$.

[Demo]

Topology Optimization for Metamaterial Design

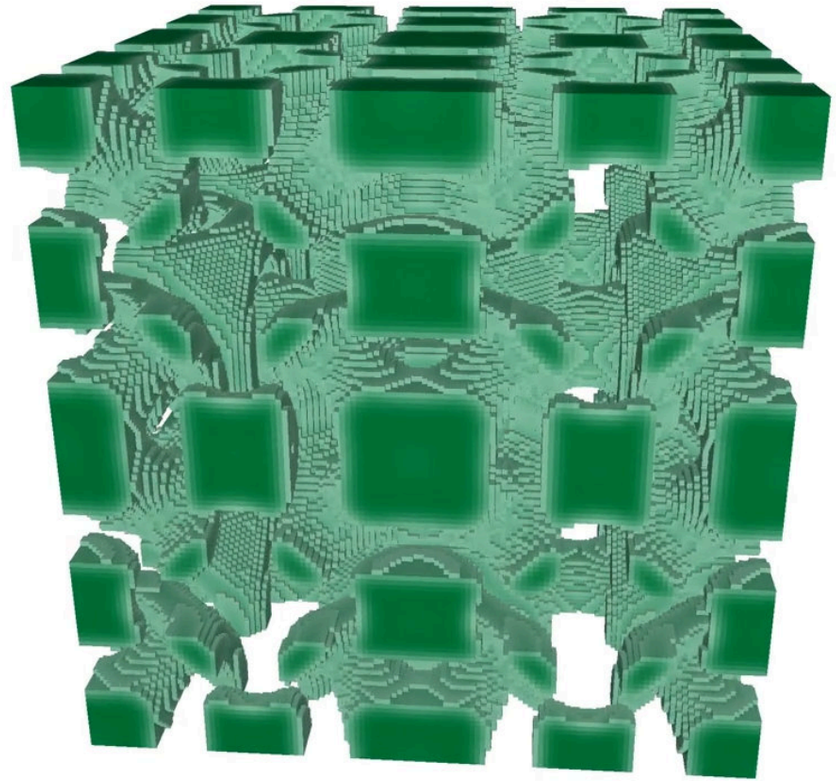
Linearly elastic design:

$$\min_{\mathbf{p}} \|C^H(\mathbf{p}) - C^{\text{target}}\|_F^2 + R(\mathbf{p})$$

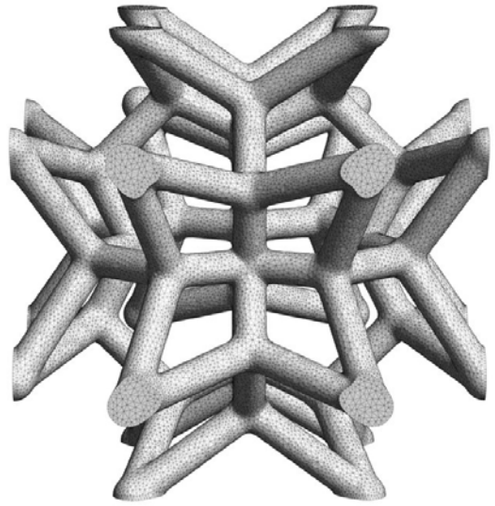
$$C^{\text{target}}: \ (E^{\text{target}}, \nu^{\text{target}}) = (E^{\text{base}}/100, 0.25)$$

Can discover *any* design! Challenges:

- Finding *good* local optima/ avoiding excess complexity
- Pushing toward a binary design
- Evaluating accurate stresses
- Ensuring manufacturability. *E.g.*:
  - controlling minimum feature size,
  - ensuring connectivity,
  - satisfying overhang constraints.



Parametric Lattice Shape Optimization

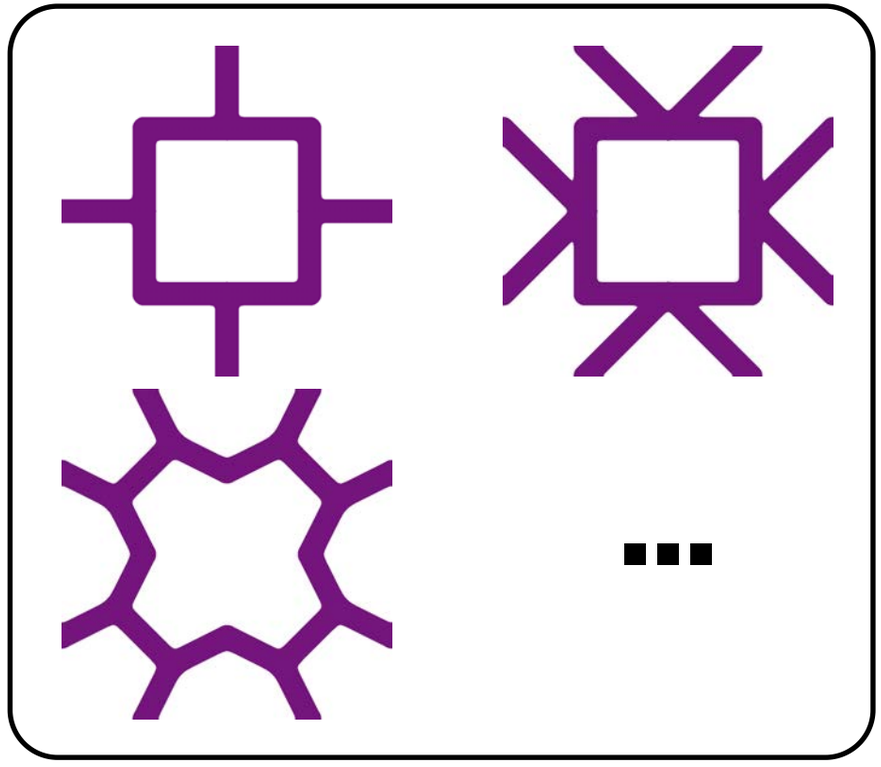


Our preferred approach has instead been to optimize smooth lattice structures:

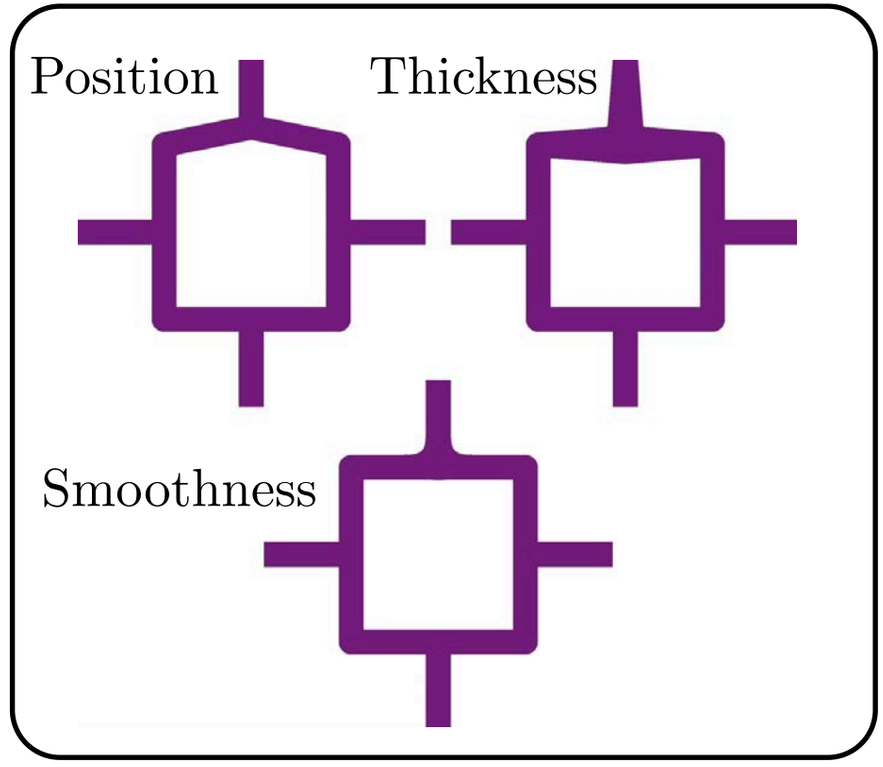
- Simplified manufacturability constraint enforcement
- Stress reduction by smoothing the joints

Parametric Lattice Shape Optimization

Design strategy:

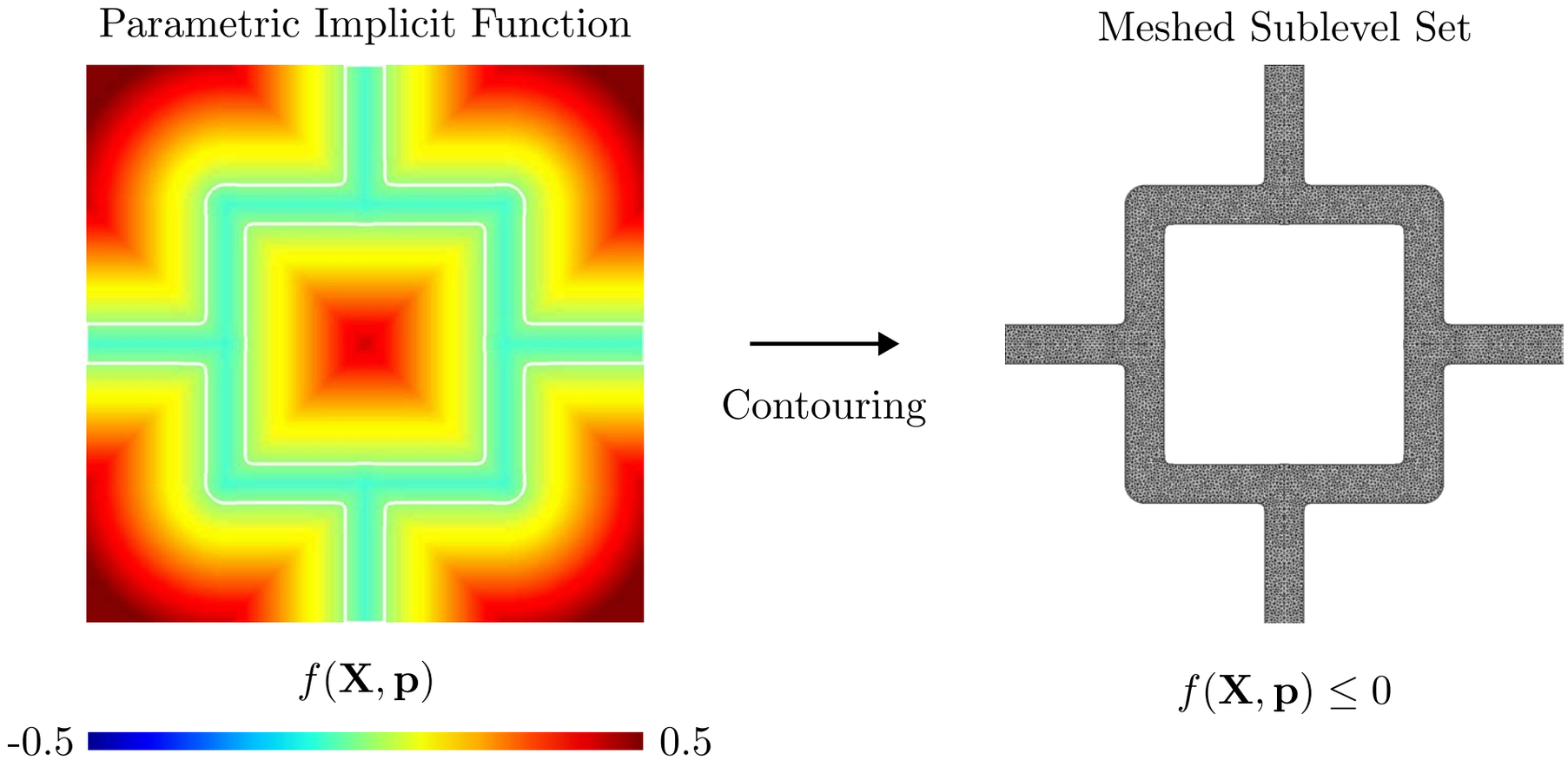


Topology Enumeration



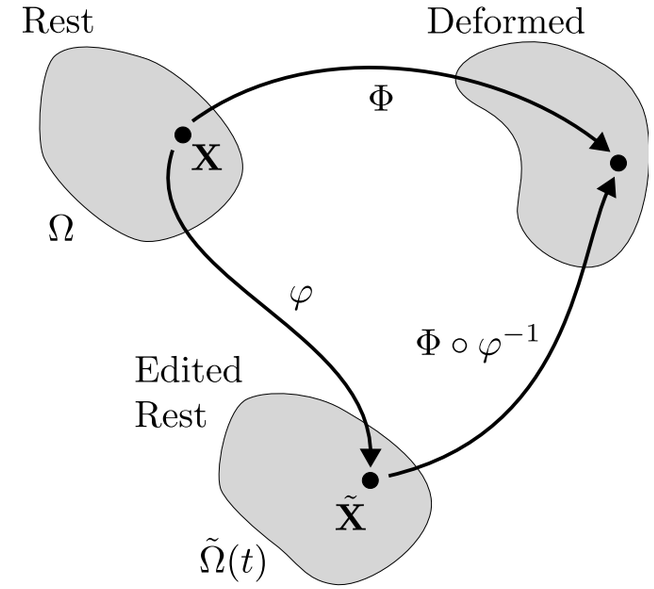
Parametric Shape Optimization

Implicit Modeling of Parametric Lattices



- Represent geometry by an *implicit function* $f(\mathbf{X}, \mathbf{p})$ depending also on shape parameters.
- Mesh the sublevel set $f(\mathbf{X}, \mathbf{p}) \leq 0$.

Shape Optimization: Shape Derivative Formulas



- The adjoint sensitivity formula needs:

$$\boldsymbol{\lambda} \cdot \frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}} = (\boldsymbol{\lambda} \cdot \nabla_{\mathbf{x}}) \frac{\partial E}{\partial \mathbf{p}}$$

- We now determine a formula for $\frac{\partial E}{\partial \mathbf{p}} \delta \mathbf{p}$ for arbitrary $\delta \mathbf{p}$.
- Applying $\delta \mathbf{p}$ flows material points along a velocity field $\mathbf{v}(\mathbf{X})$.
- After "time" t, this flow reaches $\tilde{\Omega}(t) = \varphi(\Omega(\mathbf{p}))$ via flow map $\varphi(\mathbf{X}) = \mathbf{X} + t\mathbf{v}(\mathbf{X})$.
- The initial rate of change of $E(\mathbf{x}, \cdot)$ (holding Φ fixed) is:

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

- Pulling this expression back to $\Omega(\mathbf{p})$ to keep the integration domain fixed:

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

Shape Optimization: Shape Derivative Formulas

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

- Since $\nabla\varphi = I + t\nabla\mathbf{v}$:

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

- So:

$$\frac{\partial E}{\partial \mathbf{p}}\delta\mathbf{p} = \int_{\Omega(\mathbf{p})} -\psi'\left(\nabla\Phi\right) : \left(\nabla\Phi\nabla\mathbf{v}\right) + \psi(\nabla\Phi)(\nabla\cdot\mathbf{v})\,\mathrm{d}\mathbf{X}$$

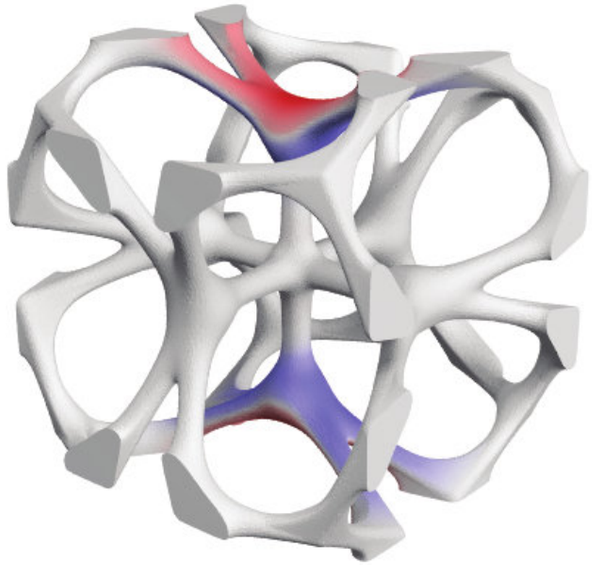
- It is convenient to isolate the $\nabla\mathbf{v}$ term:

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

Shape Optimization: Shape Derivative Formulas

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

- This is the “distributed” form of the shape derivative.
- To obtain exact gradient with respect to mesh node position $\mathbf{x}_i$, substitute $\mathbf{v} = (\delta\mathbf{x}_i)\phi_i$. Since ϕ_i has local support, only incident elements contribute.
- Alternatively, substitute edit velocities induced by changing a lattice parameter:



Computed by differentiating the implicit function

$$f(\mathbf{X}(\mathbf{p}), \mathbf{p}) = 0$$

defining points on the structure boundary.

Shape Optimization: Shape Derivative Formulas

$$\frac{\partial E}{\partial \mathbf{p}}\delta\mathbf{p} = \int_{\Omega(\mathbf{p})} G : \nabla\mathbf{v}\,\mathrm{d}\mathbf{X} \quad \text{where} \quad G := -(\nabla\boldsymbol{\Phi})^{\top}\psi'(\nabla\boldsymbol{\Phi}) + \psi(\nabla\boldsymbol{\Phi})I.$$

- Now what about the full term $\boldsymbol{\lambda}\cdot\frac{\partial^2 E}{\partial\mathbf{x}\partial\mathbf{p}}$?
- Just compute a directional derivative of G with respect to the state field $\boldsymbol{\Phi}$, perturbing $\boldsymbol{\Phi}$ by λ (the field corresponding to adjoint state $\boldsymbol{\lambda}$):

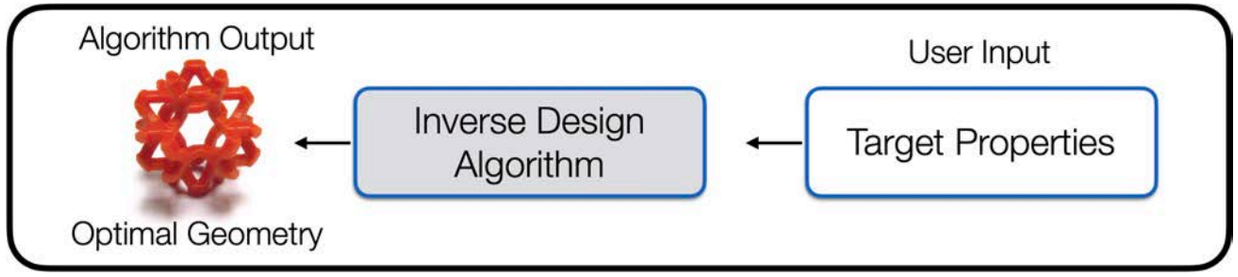
$$\boldsymbol{\lambda}\cdot\frac{\partial E}{\partial\mathbf{p}}\delta\mathbf{p} = \int_{\Omega(\mathbf{p})} \delta G : \nabla\mathbf{v}\,\mathrm{d}\mathbf{X}$$

$$\delta G = -(\nabla\lambda)^{\top}\psi'(\nabla\boldsymbol{\Phi}) - (\nabla\boldsymbol{\Phi})^{\top}\psi''(\nabla\boldsymbol{\Phi}) : (\nabla\lambda) + \psi'(\nabla\boldsymbol{\Phi}) : (\nabla\lambda)I$$

- This completes the derivations needed to evaluate the design derivative:

$$\frac{\partial\tilde{J}}{\partial\mathbf{p}} = \frac{\partial J}{\partial\mathbf{p}} - \boldsymbol{\lambda}\cdot\frac{\partial^2 E}{\partial\mathbf{x}\partial\mathbf{p}}.$$

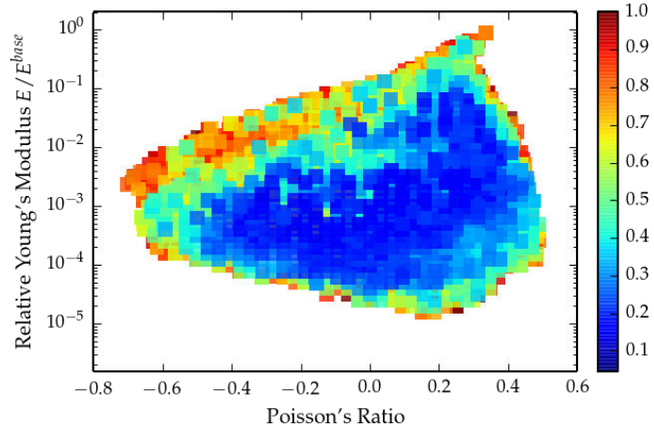
Inverse Homogenization: Linear Elasticity



- Basic linear elasticity metamaterial design:

$$\min_{\mathbf{p}\text{ admissible}} \|C^H(\mathbf{p}) - C^{\text{target}}\|_F^2$$

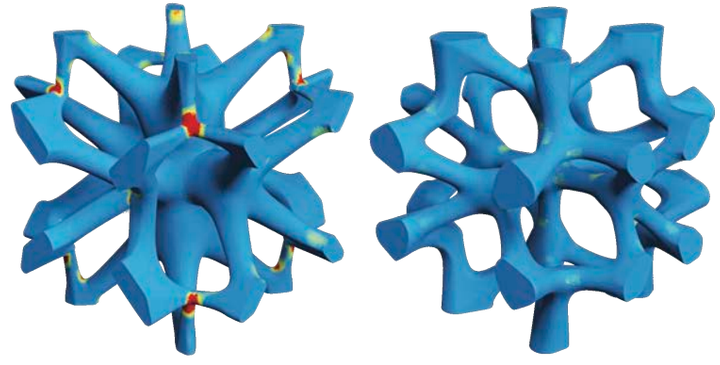
C^H actually can be differentiated without an adjoint solve! See Problem Set Exercise 10.



Property Coverage and Stress Reduction Factor

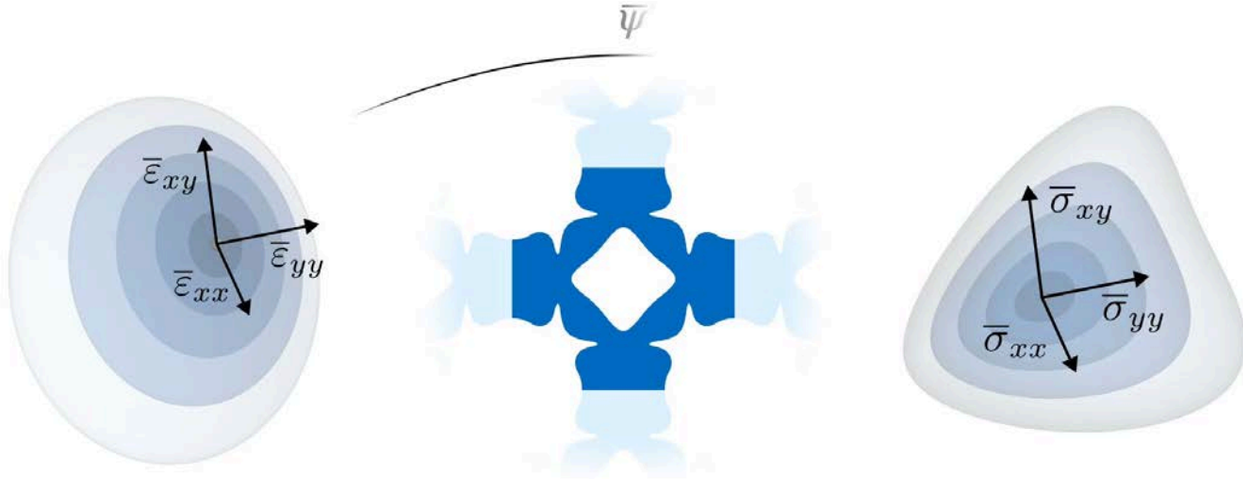
- Avoid stress concentrations:

$$\min_{\substack{\mathbf{p}\text{ admissible} \\ C^H(\mathbf{p})=C^{\text{target}}}} J_{\text{worst\_case\_stress}}(\mathbf{p})$$



Inverse Homogenization: Nonlinear Elasticity

Recall forward homogenization:

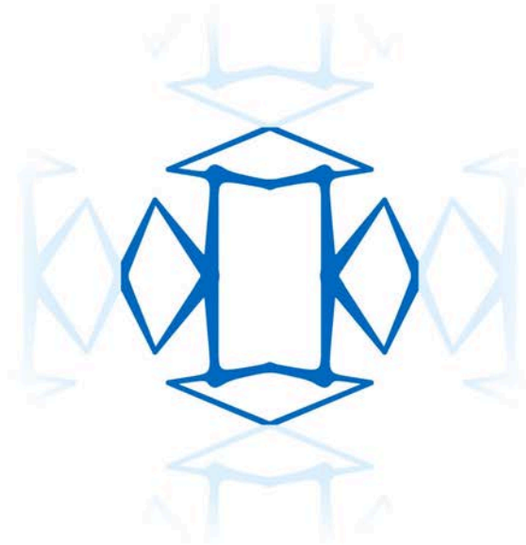


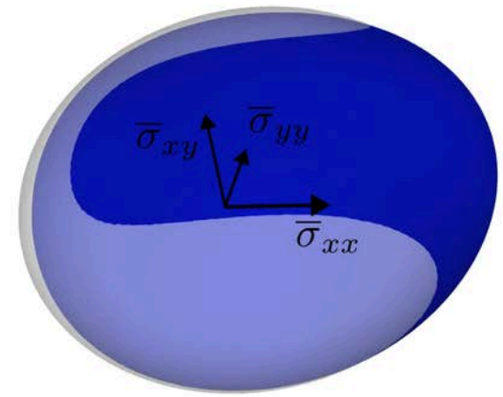
- A finite-strain material fitting formulation:

$$\min_{\Omega} \int_{\mathcal{F}} \left(\overline{\psi} - \overline{\psi}_{tgt}\right)_{rel}^2 + \left\|\overline{\psi}' - \overline{\psi}'_{tgt}\right\|_{rel}^2 d\overline{F}$$

- Design as linear-as-possible materials:

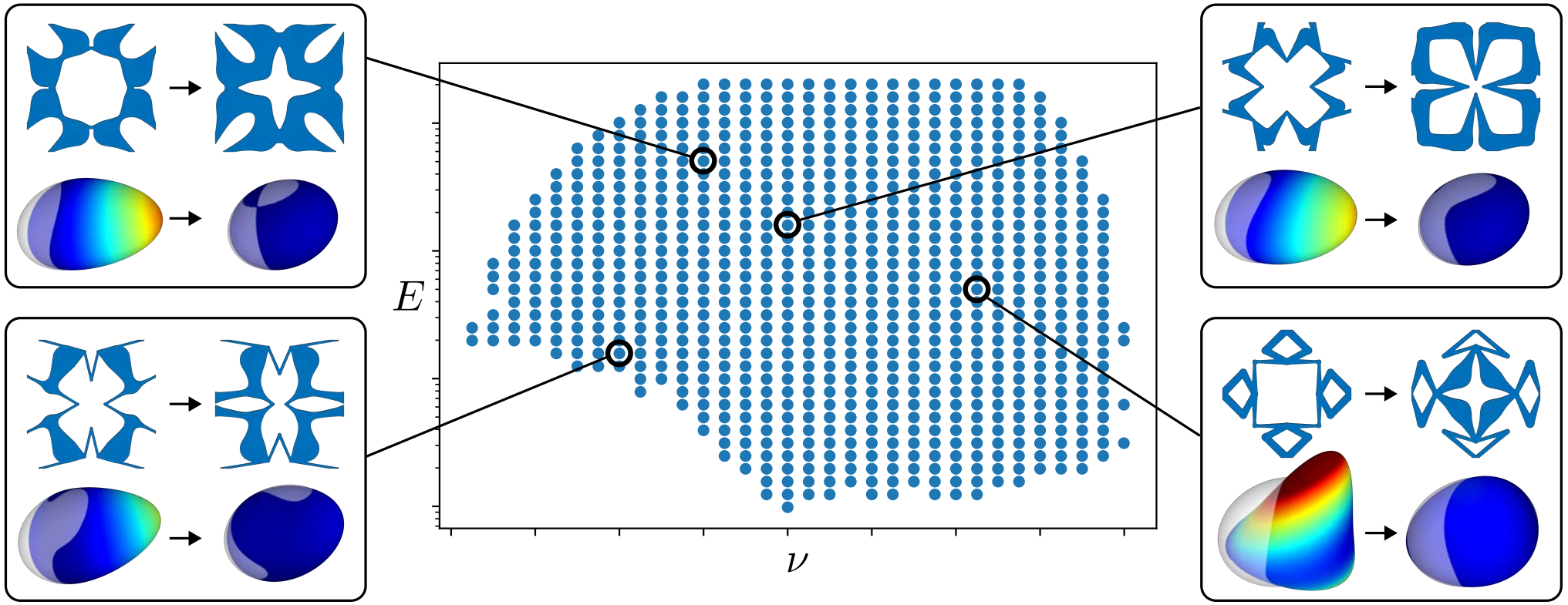
$$\overline{\psi}_{tgt}(\overline{F}) = \frac{1}{2}(\overline{F} - I) : C^{\text{target}} : (\overline{F} - I)$$





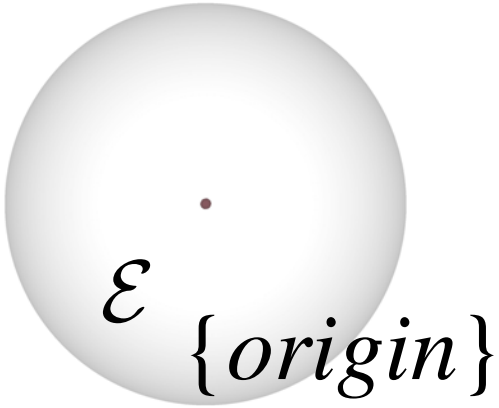
LARGE-SCALE VALIDATION

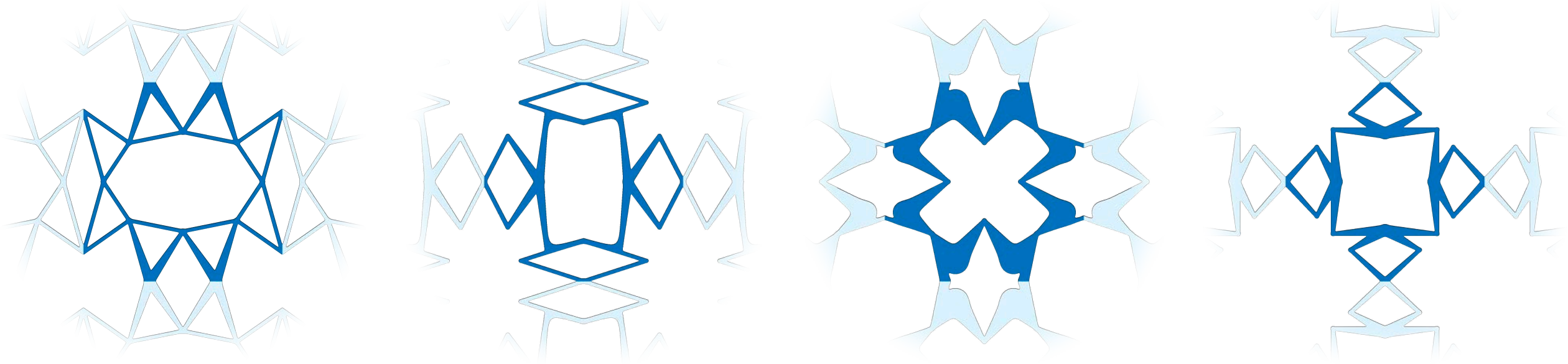
Isotropic Linear Material Design



LARGE-SCALE VALIDATION

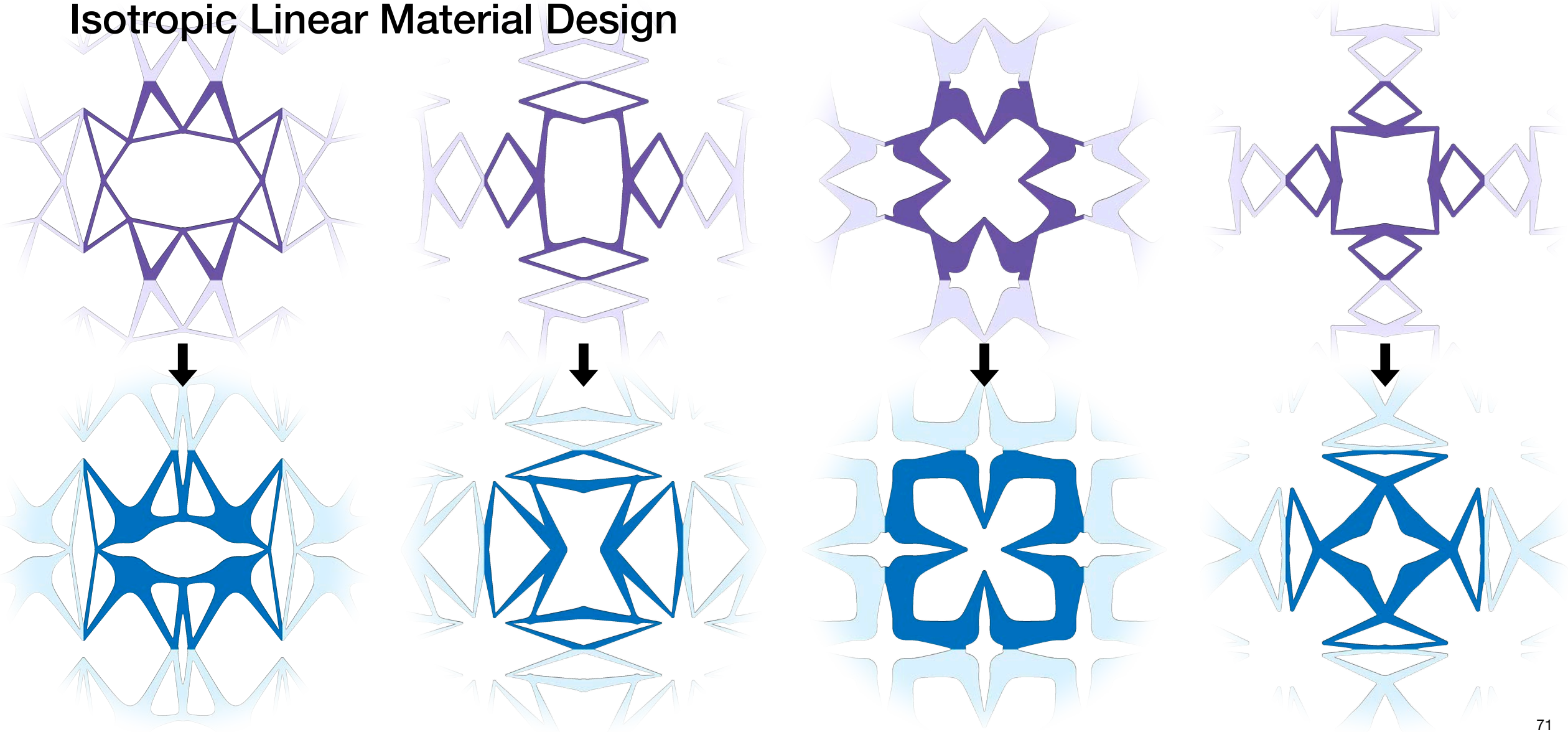
Isotropic Linear Material Design



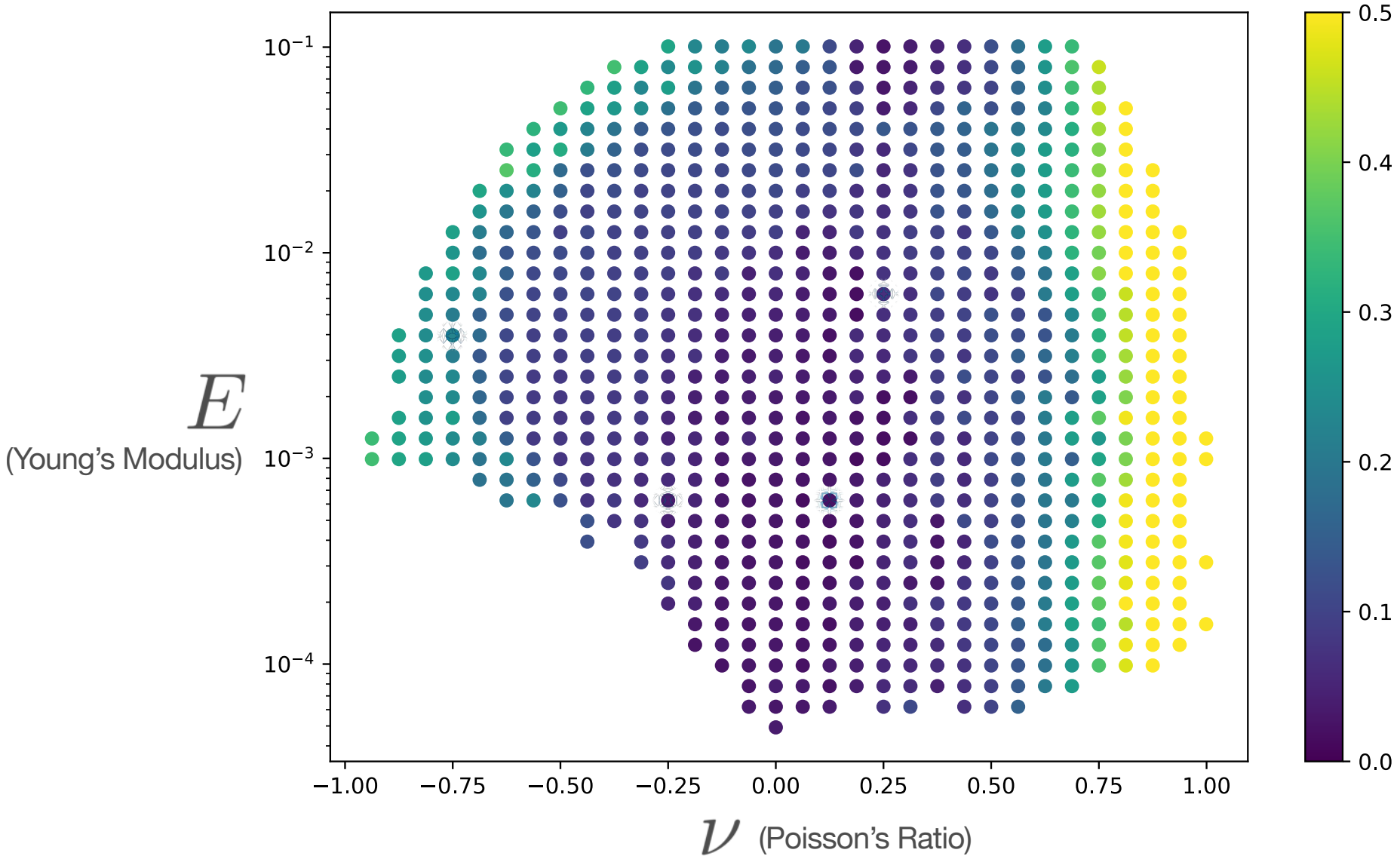


LARGE-SCALE VALIDATION

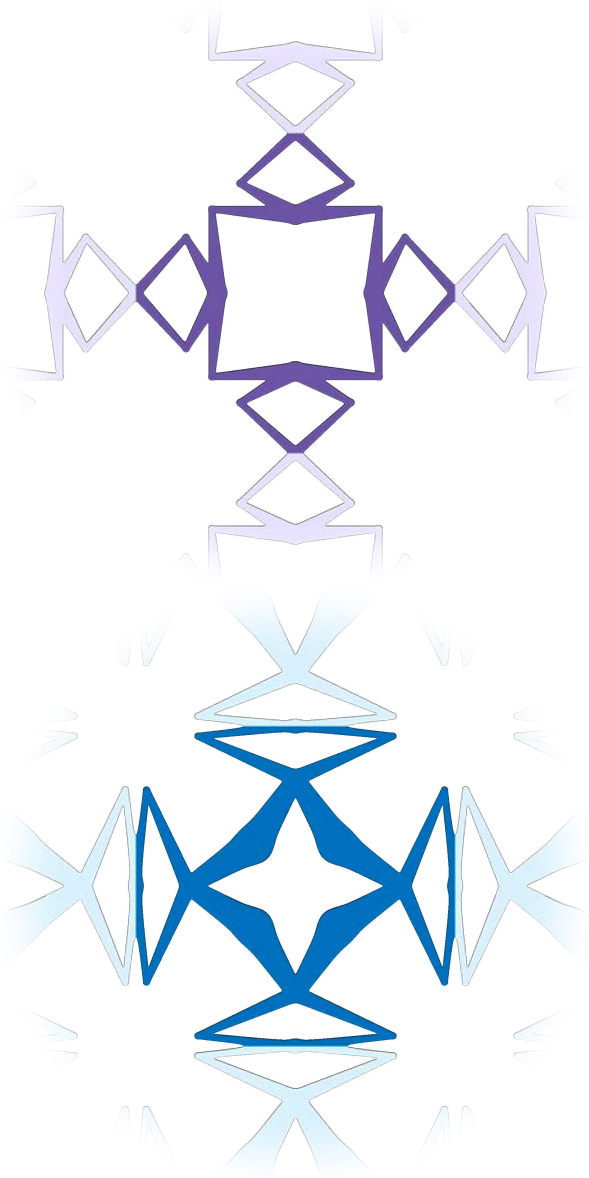
Isotropic Linear Material Design

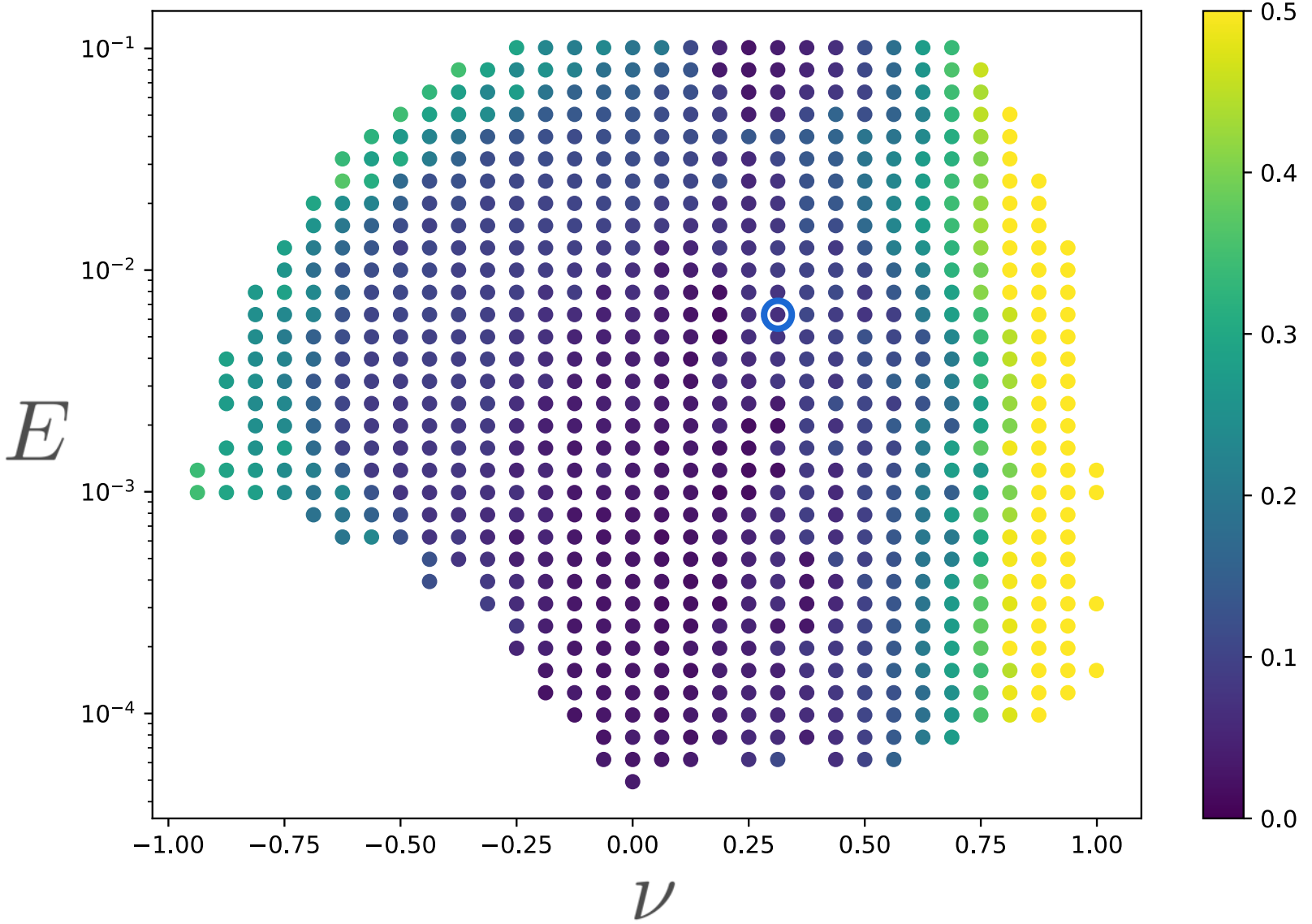


LARGE-SCALE VALIDATION



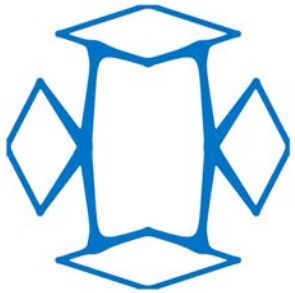
RESULTS

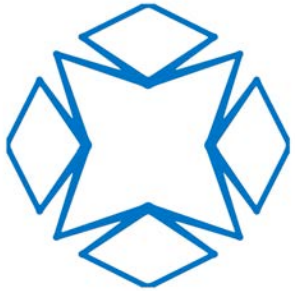


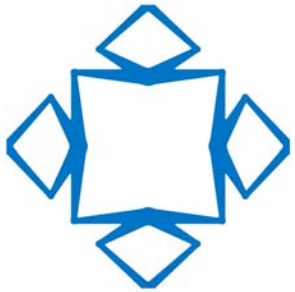

10^{-1}
10^{-2}
10^{-3}
10^{-4}
E
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
ν
0.5
0.4
0.3
0.2
0.1
0.0

RESULTS

Different Poisson's Ratio







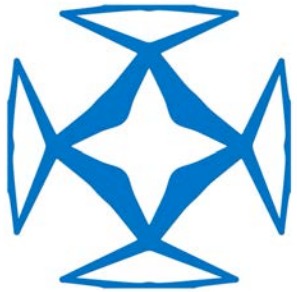
-0.25

0.00

0.25





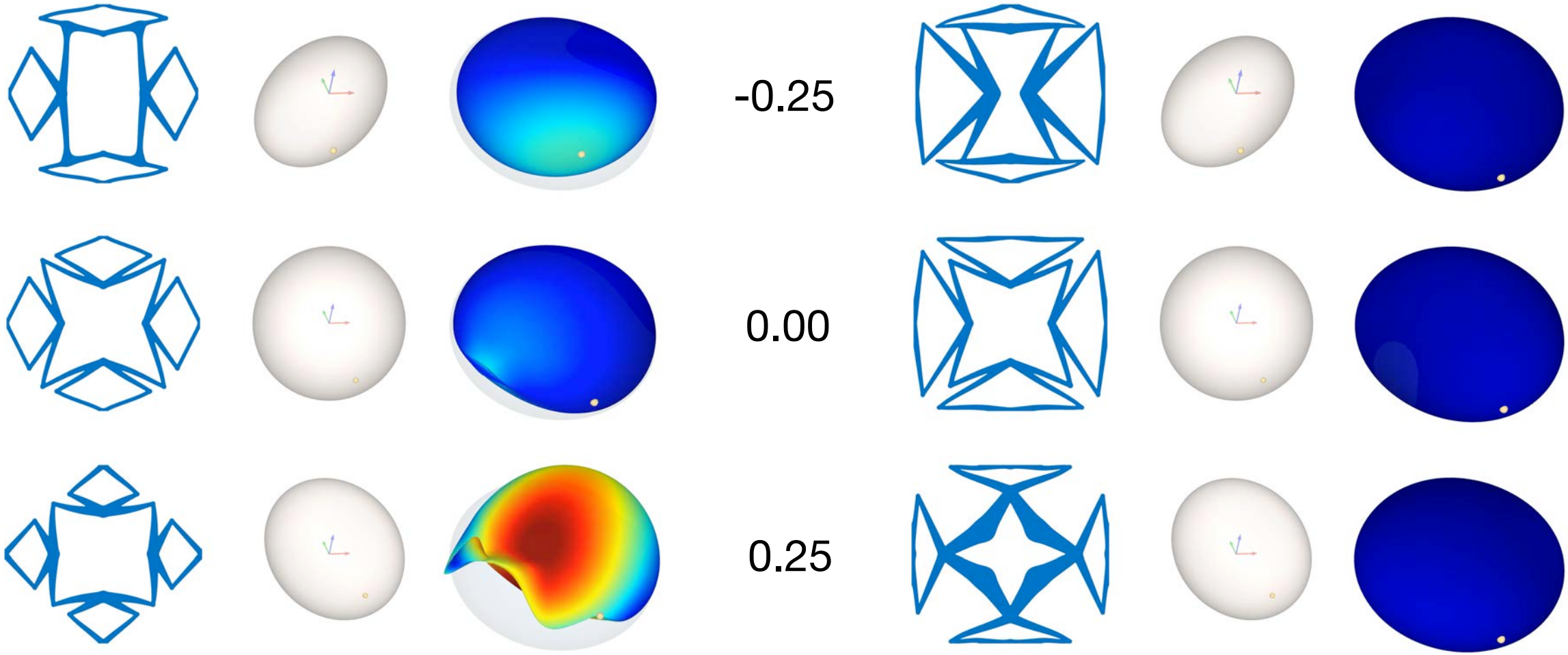


Old

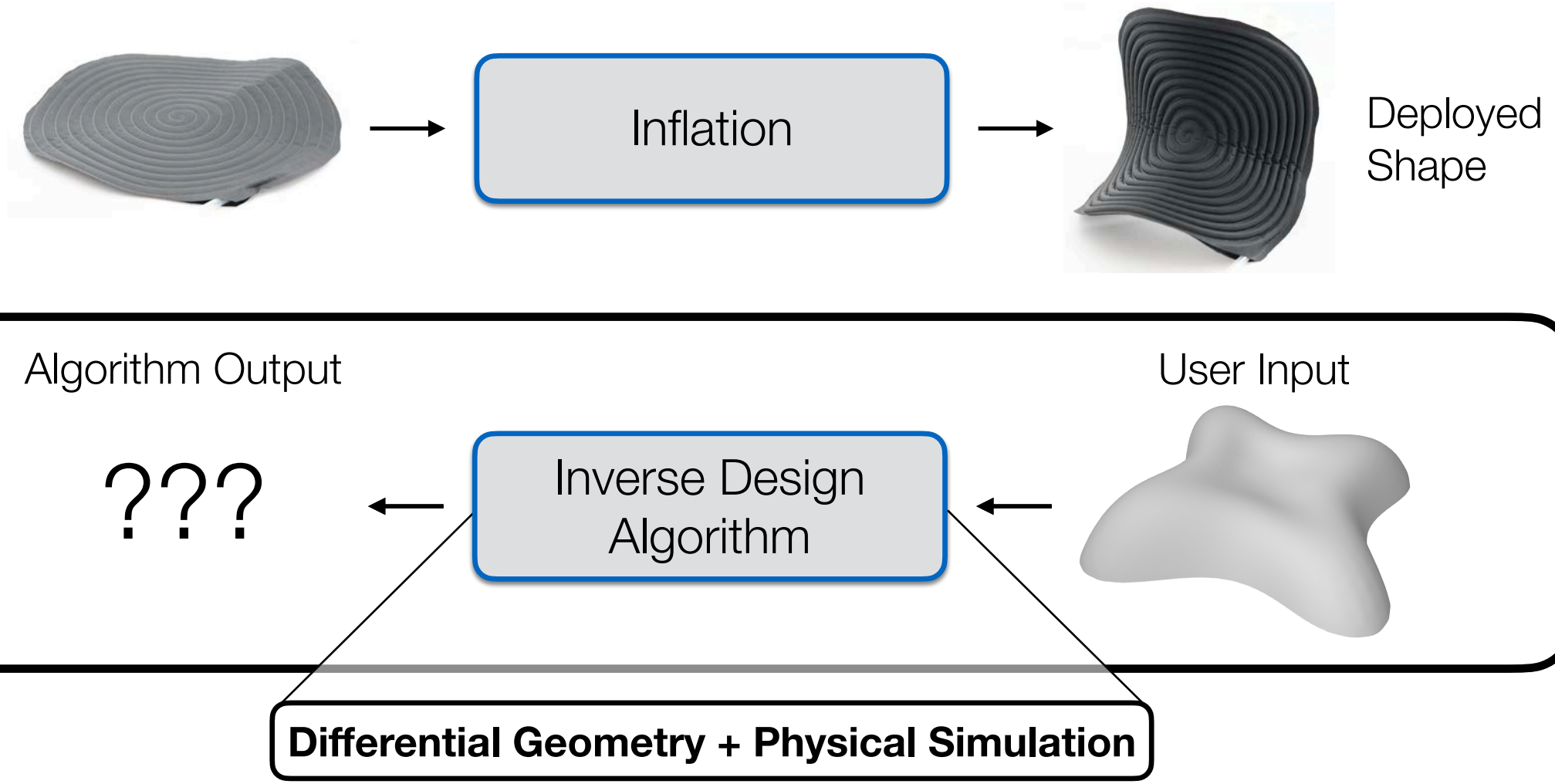
Opt

RESULTS

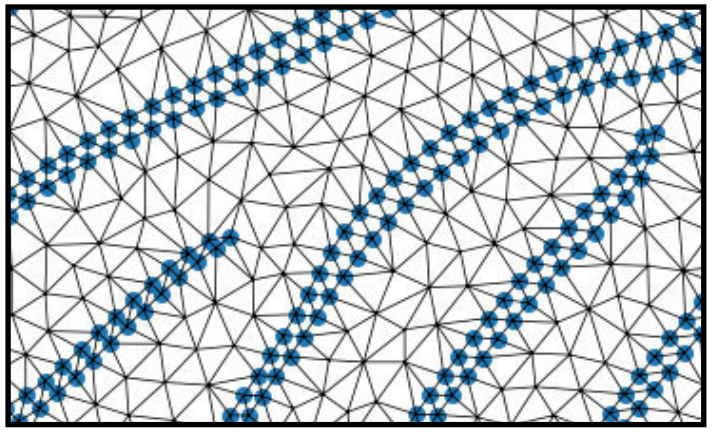
Different Poisson's Ratio



INFLATABLE DESIGN



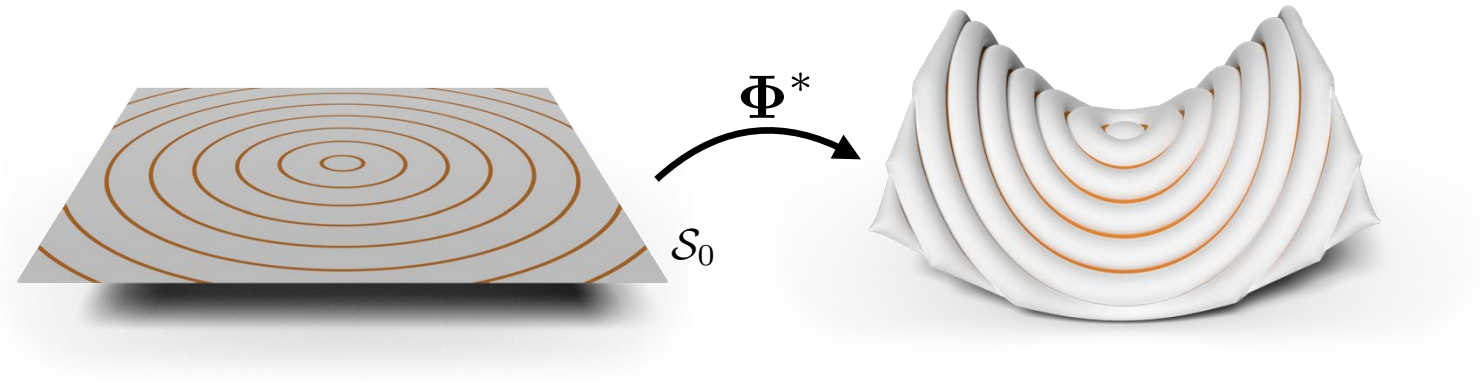
PHYSICS-GUIDED OPTIMIZATION



Optimize the wall vertex positions X_{wall} :

$$\min_{X_{\text{wall}}} \ \text{TargetDist}\Big(\mathbf{\Phi}^*\big(\mathcal{S}_0(X_{\text{wall}})\big)\Big)$$

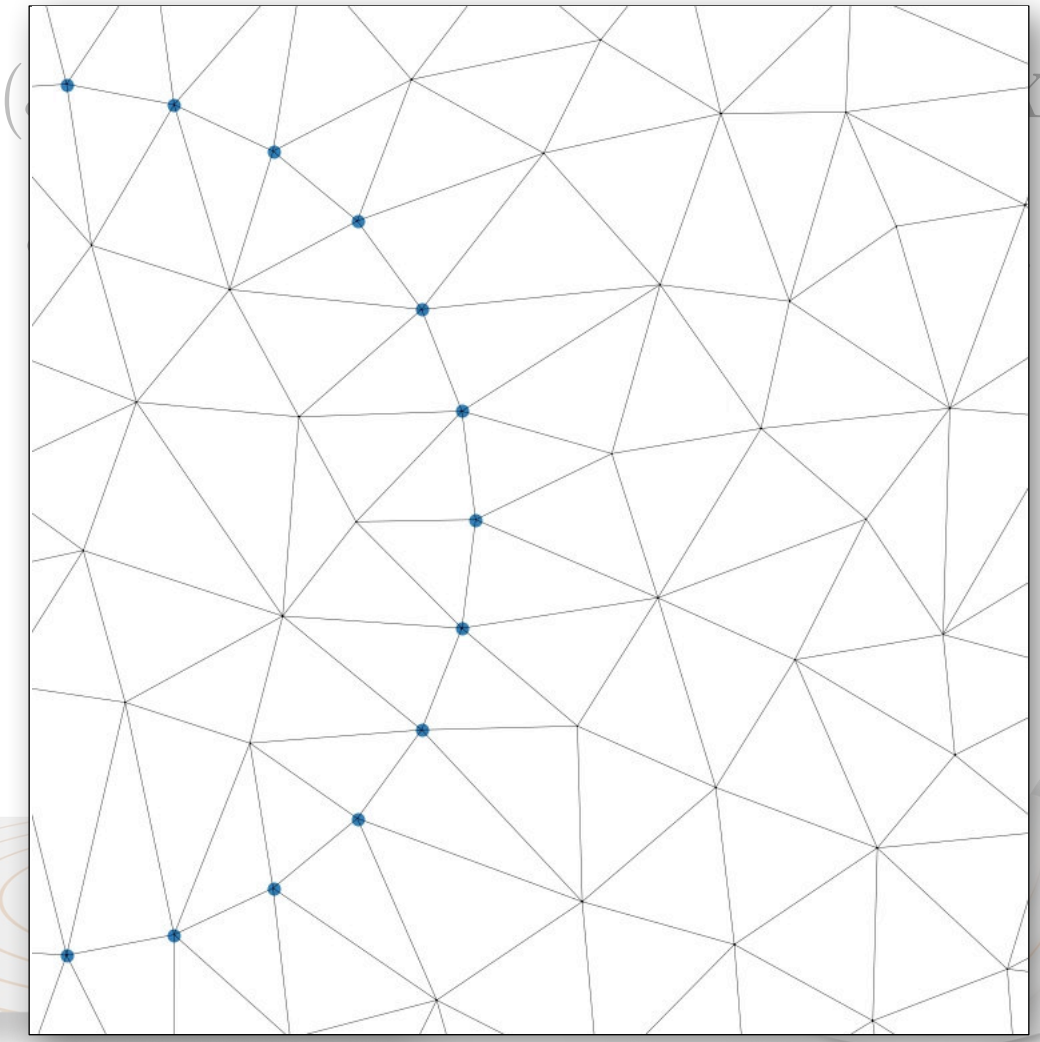
$$\text{Where} \quad \mathbf{\Phi}^* := \operatorname*{argmin}_{\mathbf{\Phi}} \int_{\mathcal{S}_0} \psi(\nabla \mathbf{\Phi}) \, \mathrm{d}A - p \ \texttt{Vol}(\mathcal{S})$$



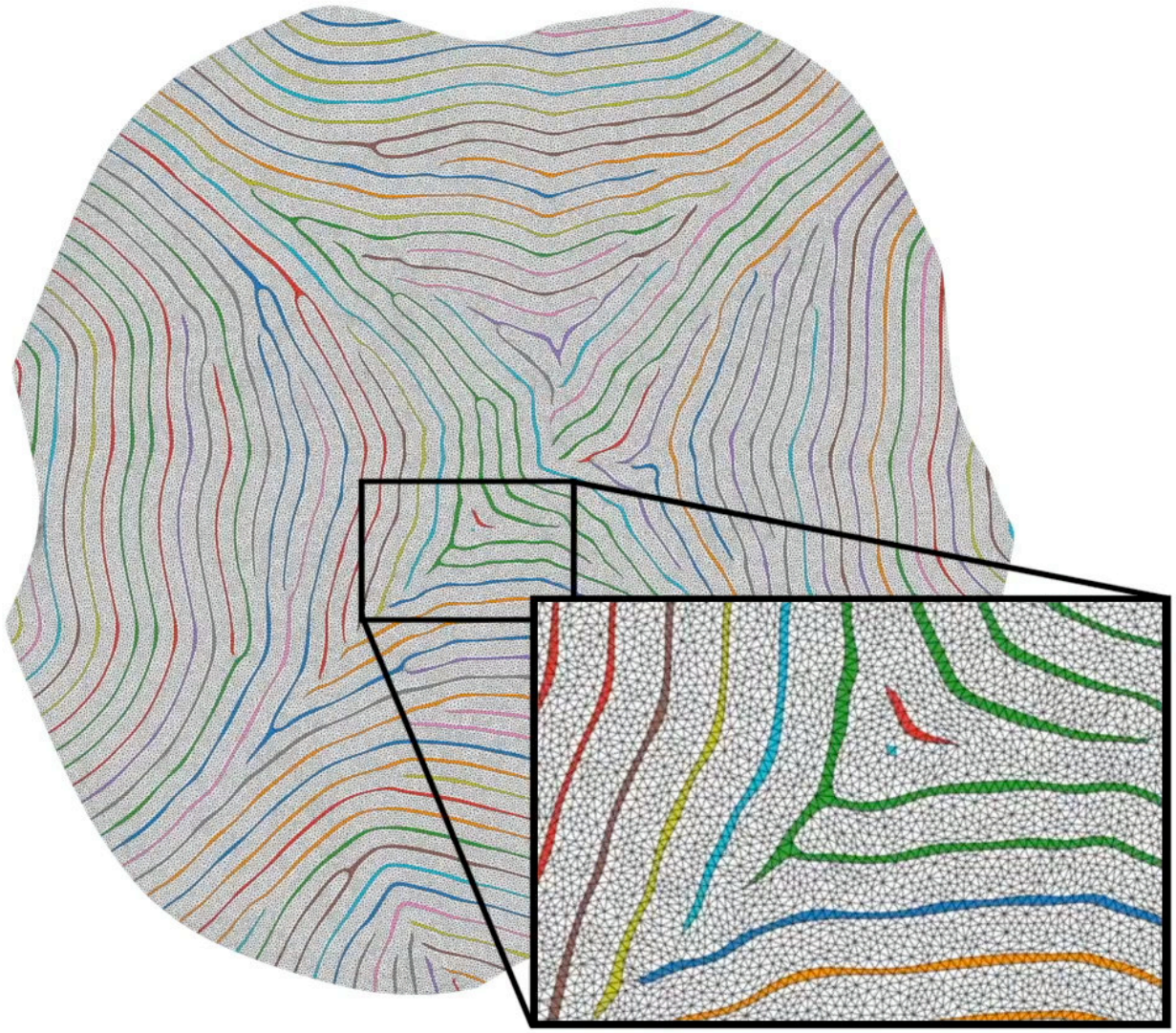
PHYSICS-GUIDED OPTIMIZATION

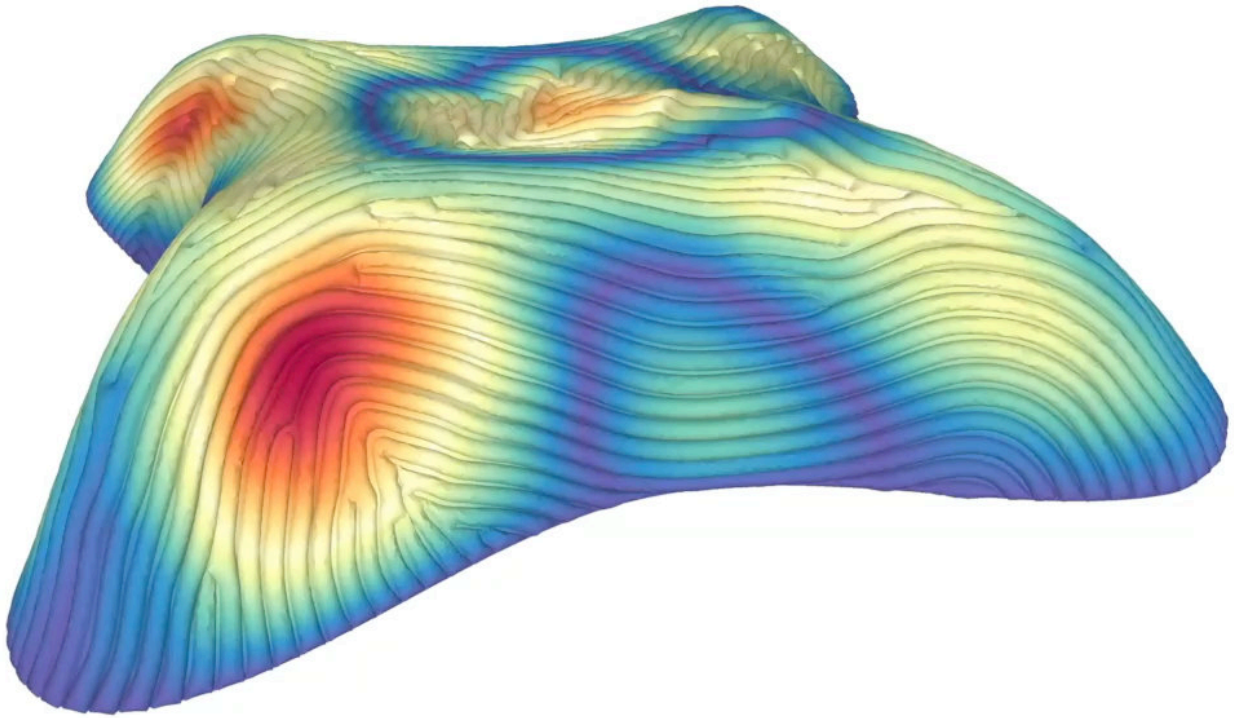
$$\min_{X_{\text{wall}}} \text{TargetDist}\Big(\Phi^*\big(\quad X_{\text{wall}}\big)\Big) + \text{Smooth}(X_{\text{wall}})$$

Where $\quad (\mathcal{S})$

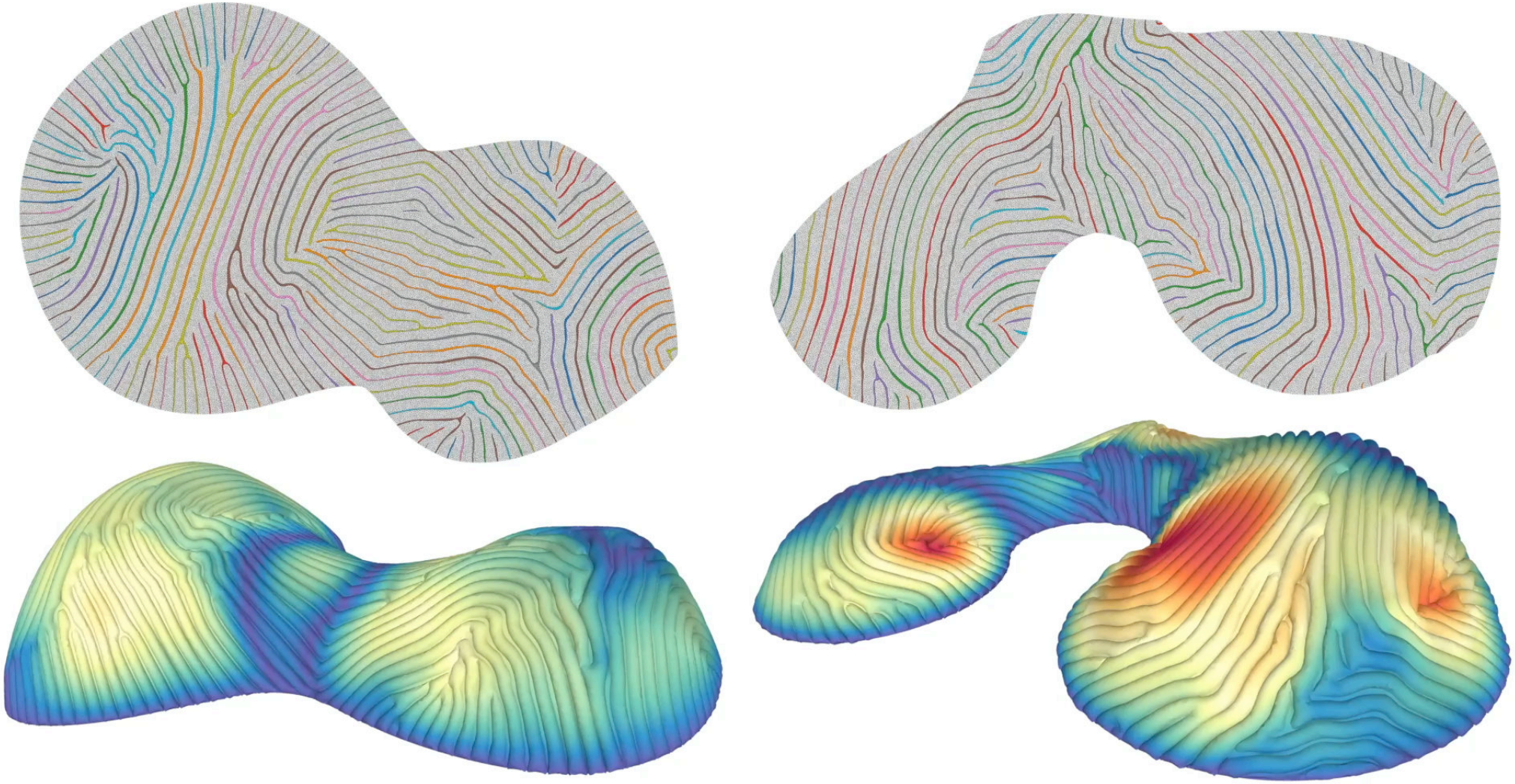


OPTIMIZATION EXAMPLES





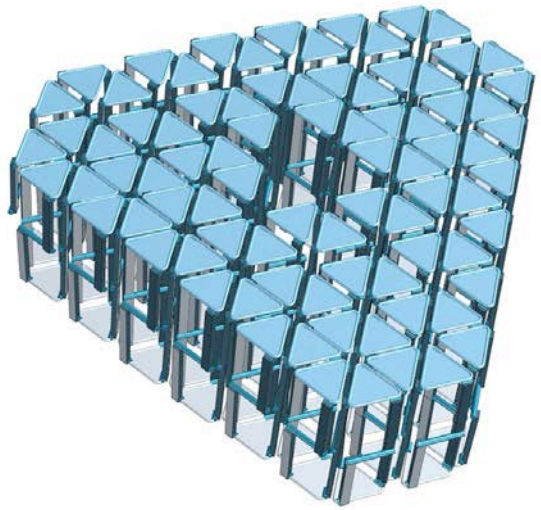
OPTIMIZATION EXAMPLES

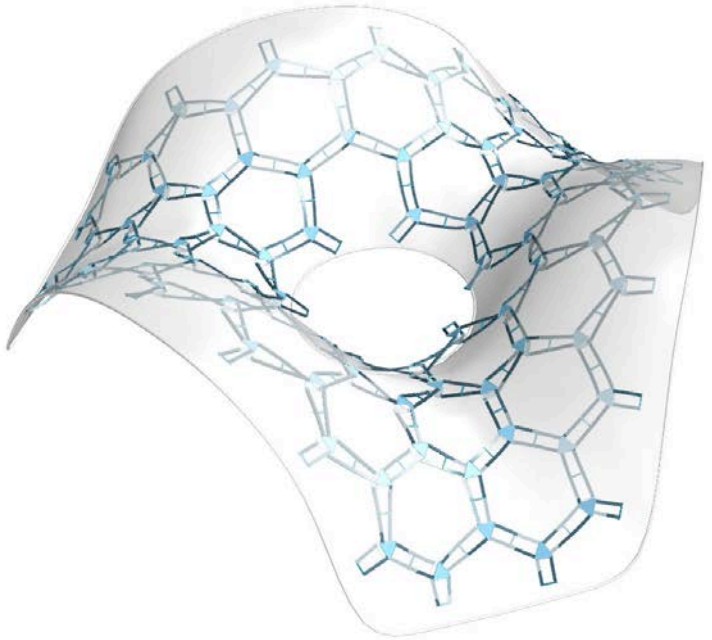


Example: Umbrella Mesh Optimization

$$\min_{\mathbf{x}} \left\| \quad - \quad \right\|^2 + \quad + \left\| \quad - \quad \right\|^2$$

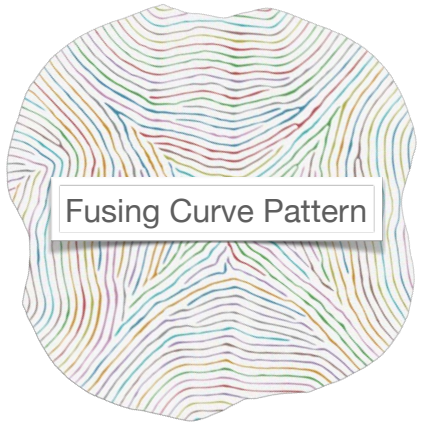
Target surface approximation | Low strain energy | Low separation forces





Motivation: Initializing Inflatables

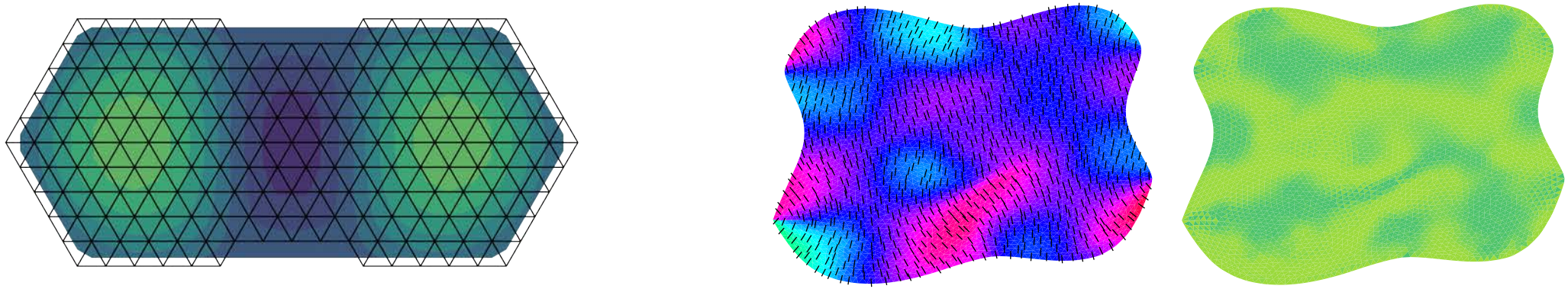
$$\min \left\| \text{Equilibrium Shape} - \text{Target Surface} \right\|^2$$



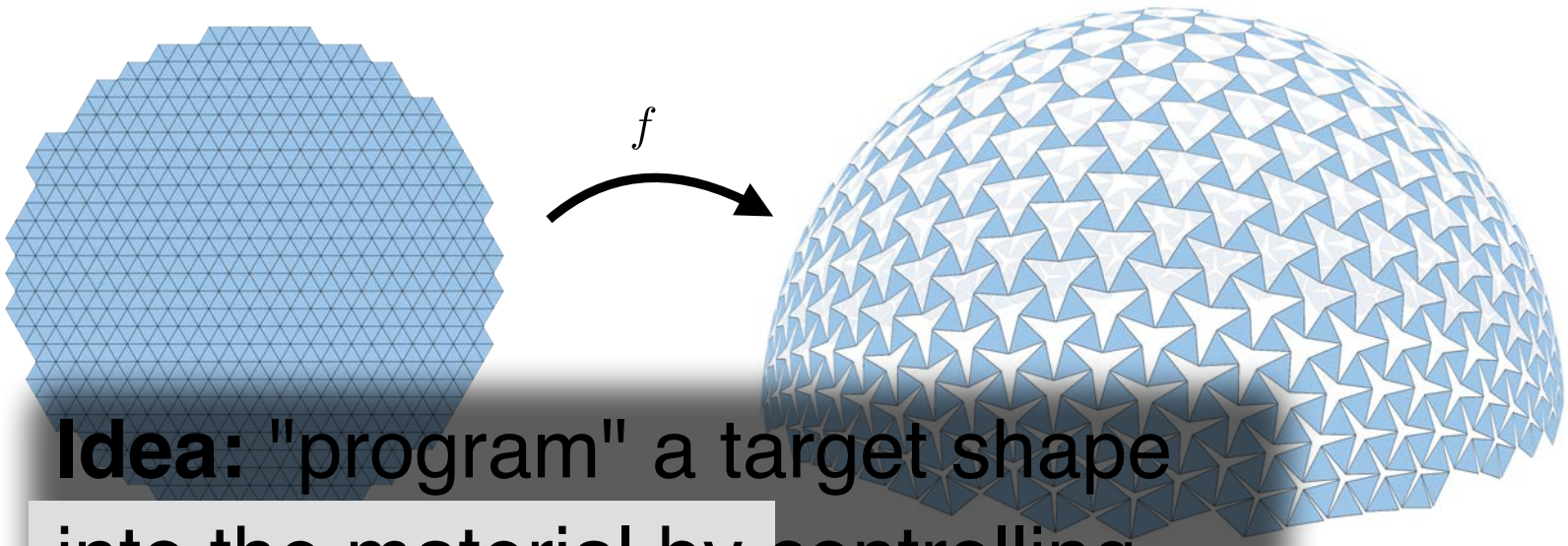
How to obtain fusing curve network topology?

... a reasonable initial mesh for optimization?

Coarsening / Two-Scale Design



Deforming to Curved Surfaces



Idea: "program" a target shape into the material by controlling .

Geometric abs

[Konaković Luković et al. 2018]

- Material undergoes a **uniform scale** and rotation at each point:

$$\nabla f = \lambda R \qquad \lambda : \Omega \to [1, 2]$$

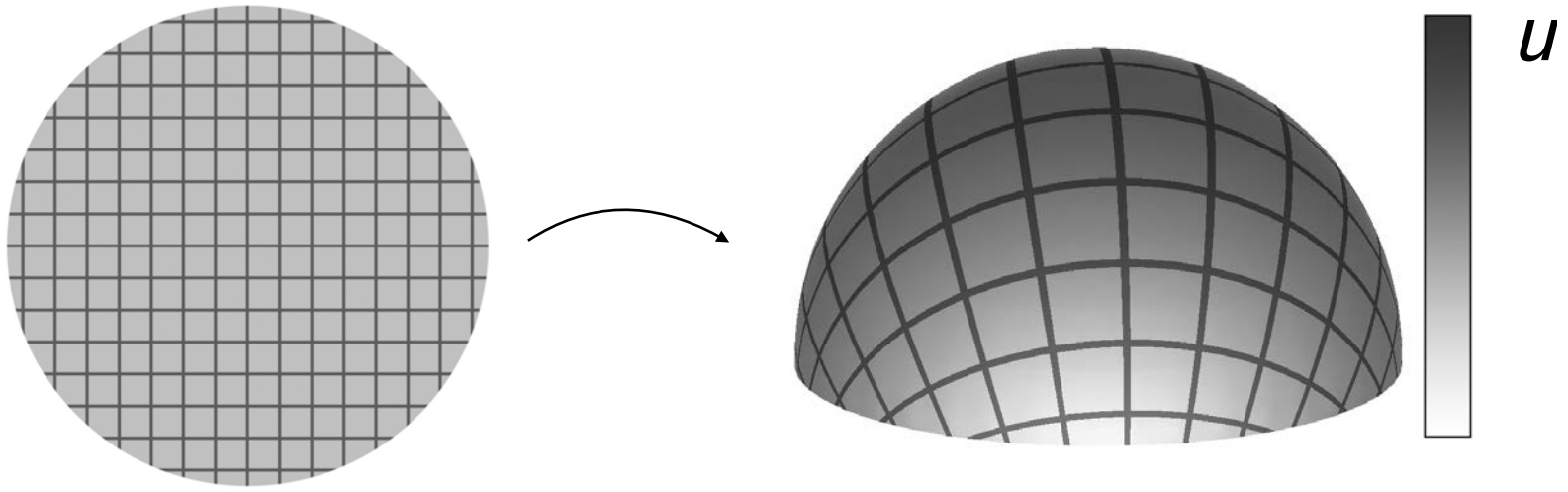
- In other words, f approximates a **conformal map**
- Scale factor field λ determines the metric—and thus the deployed shape (up to isometric deformations).

Deforming to Curved Surfaces

Prescribed Gaussian curvature equation:

$$-\Delta_0 u = e^{2u} K \qquad = e^{u}$$

Conformal Scale Factor



Off-the-Shelf Conformal Mapping Software

Eurographics Symposium on Geometry Processing 2008
Pierre Alliez and Szymon Rusinkiewicz
(Guest Editors)

Volume 27 (2008), Number 5

Spectral Conformal Parameterization

Patrick Mullen[1] Yiying Tong[2] Pierre Alliez[3] Mathieu Desbrun[1]
[1]Caltech [2]Michigan State University [3]INRIA Sophia-Antipolis

Abstract

We present a spectral approach to automatically and efficiently obtain discrete free-boundary conformal parameterizations of triangle mesh patches, without the common artifacts due to positional constraints on vertices and without undue bias introduced by sampling irregularity. High-quality parameterizations are computed through a constrained minimization of a discrete weighted conformal energy by finding the largest eigenvalue/eigenvector of a generalized eigenvalue problem involving sparse, symmetric matrices. We demonstrate that this novel and robust approach improves on previous linear techniques both quantitatively and qualitatively.

1 Introduction

Due to its central importance in geometry processing, the subject of mesh parameterization has been researched for a number of years. "Parameterizing" a triangle mesh traditionally means computing a correspondence between a discrete, triangulated surface patch (possibly with holes) and a homeomorphic planar mesh through a piecewise linear map. Finding this piecewise linear mapping amounts to assigning each mesh node a pair of coordinates (u,v) referring to its position in the planar region. Such (ideally one-to-one) mappings provide a flat parametric space, allowing complex mesh processing operations such as surface fitting and remeshing to be performed directly on a flat domain rather than on the curved, original surface patch. Planar coordinates are also particularly useful to dramatically enhance the visual richness of a 3D surface through texture mapping, both for overly simplified character meshes in game engines and for incredibly detailed surfaces in computer-generated feature films. Consequently, fast methods generating less distortion than current tools are still in high demand.

1.1 Previous Work

Since we are trying to flatten an intrinsically non-flat triangulated surface from 3D down to 2D, there is in general no perfect way to perform this flattening without introducing some form of distortion. Due to the ubiquitous need for local and global parameterizations in geometry processing applications, a number of distortion measures have been defined, and various numerical approaches have been proposed to efficiently provide (u,v) parameter values to meshes minimizing one of these measures. Although thorough surveys are available [FH05, SPR06, HLS07], we go over the methods most related to our contribution to motivate our approach.

Existing methods generally fall into one of three categories: linear, non-linear, and hybrid methods. Non-linear methods, based on discrete or differential-geometric non-linear distortion measures [HG00, SdS01, SLMB05, ZKK02, SGSH02, ZSGS04, KSS06, JKG07, SSP08], offer strong theoretical guarantees on the resulting maps, at the cost of a generally higher computational effort. Recently, hybrid techniques linearizing these non-linear measures have been proposed to emulate non-linear methods at the cost of only a few linear solves [ZLS07, BCGB08], potentially saving execution time—although the trade-off between the effect of the linearization and the computational saving may be very application-dependent.

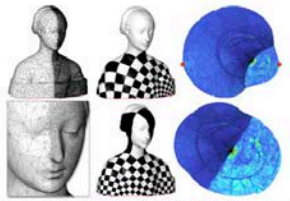


Figure 1: ***Sforza****. On this mesh (50K vertices) with varying sampling rates (left), previous linear methods (top, LSCM/DCP) fail to capture the symmetry of the mesh in the parameterization (solved in 4s). In contrast, our spectral approach (bottom) automatically computes a low-distortion conformal map (solved in 5.2s).*

Linear methods, instead, require solving a *single* linear system. These methods date back to the use of the connectivity-based graph Laplacian in Tutte's embedding theorem [Tut63] and its generalizations [GGT06]. A significant improvement, involving a linear system not only dependent on the connectivity but also on the geometry of the original mesh (through the "cotangent weights"), was introduced

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.
Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Boundary First Flattening

ROHAN SAWHNEY and KEENAN CRANE, Carnegie Mellon University

A *conformal flattening* maps a curved surface to the plane without distorting angles—such maps have become a fundamental building block for problems in geometry processing, numerical simulation, and computational design. Yet existing methods provide little direct control over the shape of the flattened domain, or else demand expensive nonlinear optimization. *Boundary first flattening (BFF)* is a linear method for conformal parameterization which is faster than traditional linear methods, yet provides control and quality comparable to sophisticated nonlinear schemes. The key insight is that the boundary data for many conformal mapping problems can be efficiently constructed via the *Cherrier formula* together with a pair of *Poincaré-Steklov operators*; once the boundary is known, the map can be easily extended over the rest of the domain. Since computation demands only a single factorization of the real Laplace matrix, the amortized cost is about 50x less than any previously published technique for boundary-controlled conformal flattening. As a result, BFF opens the door to real-time editing or fast optimization of high-resolution maps, with direct control over boundary length or angle. We show how this method can be used to construct maps with sharp corners, cone singularities, minimal area distortion, and uniformization over the unit disk; we also demonstrate for the first time how a surface can be conformally flattened directly onto any given target shape.

CCS Concepts: • **Computing methodologies** → **Mesh models**; • **Mathematics of computing** → **Partial differential equations**;

Additional Key Words and Phrases: discrete differential geometry, digital geometry processing, surface parameterization, conformal geometry

ACM Reference format:
Rohan Sawhney and Keenan Crane. 20XX. Boundary First Flattening. *ACM Trans. Graph.* 0, 0, Article 0 (20XX), 14 pages.
DOI: 0000001.0000001_2

1 INTRODUCTION

In recent years conformal flattening has evolved far beyond its humble origins as a tool for texture mapping [Desbrun et al. 2002; Lévy et al. 2002], providing new perspectives on a broad range of applications including surface remeshing [Alliez et al. 2003; Zhong et al. 2014], comparative data analysis [Hurdal and Stephenson 2009; Lipman and Daubechies 2011], computational biology [Koehl and Hass 2015], physical simulation [Segall et al. 2016], sensor networks [Sarkar et al. 2009], and computational design [Konakovic et al. 2016]. Why such great interest in maps that preserve *angles*? One answer is computational: conformal mapping typically amounts to solving easy linear equations, providing fast, scalable algorithms—or cheap initialization for more difficult nonlinear tasks [Chao et al. 2010; Liu et al. 2008]. Another is that angle preservation is directly linked to real mechanical or constitutive properties of physical systems [Kim et al. 2012]. Continued advancement of basic tools for conformal mapping therefore has high utility across a variety of domains. At present, however, linear algorithms for conformal flattening do not exploit the full space of possibilities.

© 20XX ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, http://dx.doi.org/0000001.0000001_2.

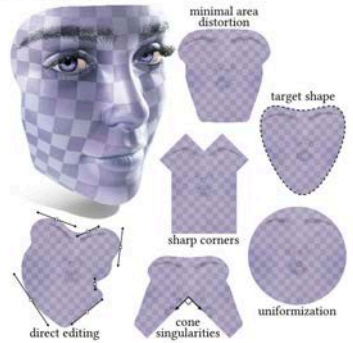


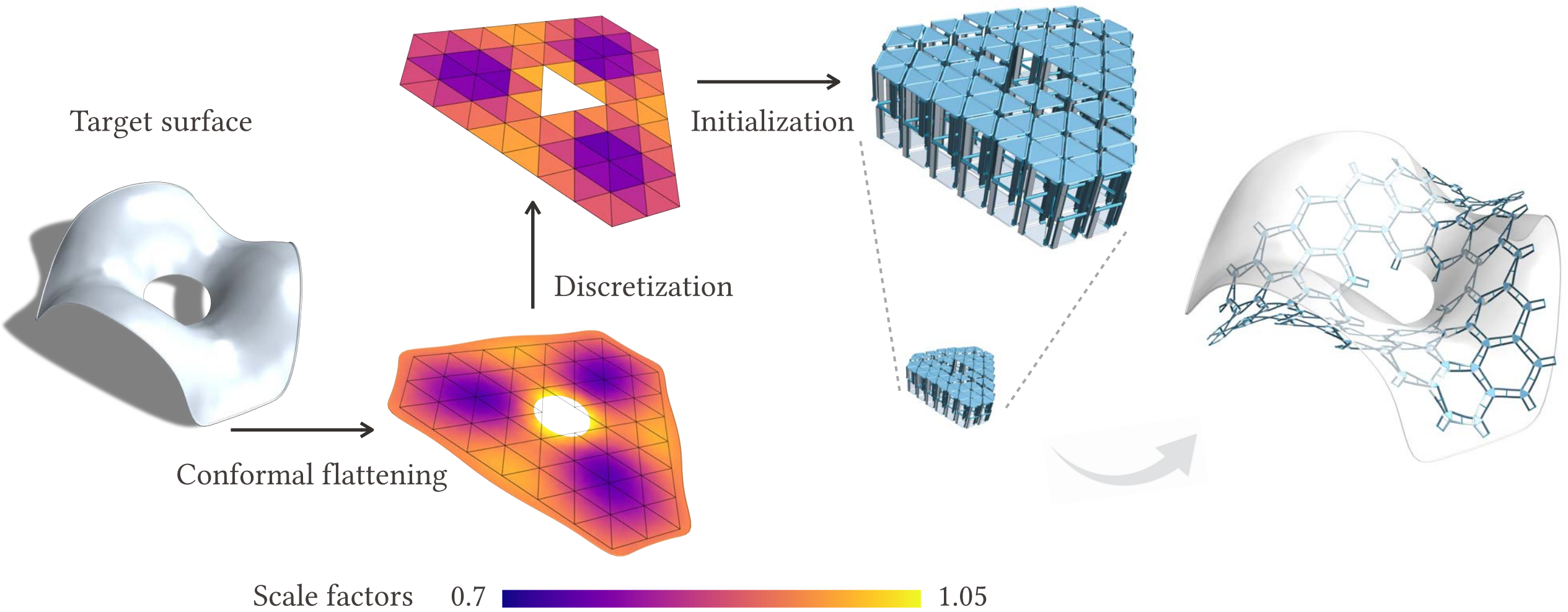
Fig. 1. *Boundary first flattening (BFF)* provides sophisticated control over the shape of an angle-preserving or *conformal* map to the plane. Even on very large meshes, this shape can be edited in real time.

In the smooth setting, conformal maps are quite flexible: a map from a disk-like surface to the flat plane can achieve any target shape as long as it is free to "slide" along the boundary. In stark contrast, existing linear conformal flattening algorithms provide *no* explicit control over the target shape: the user obtains a single, automatic solution and must "take it or leave it." Nonlinear methods provide additional control over shape, but at significantly greater cost. *Boundary first flattening (BFF)* is the first conformal flattening method to provide full control over the target shape via a single sparse matrix factorization, including:

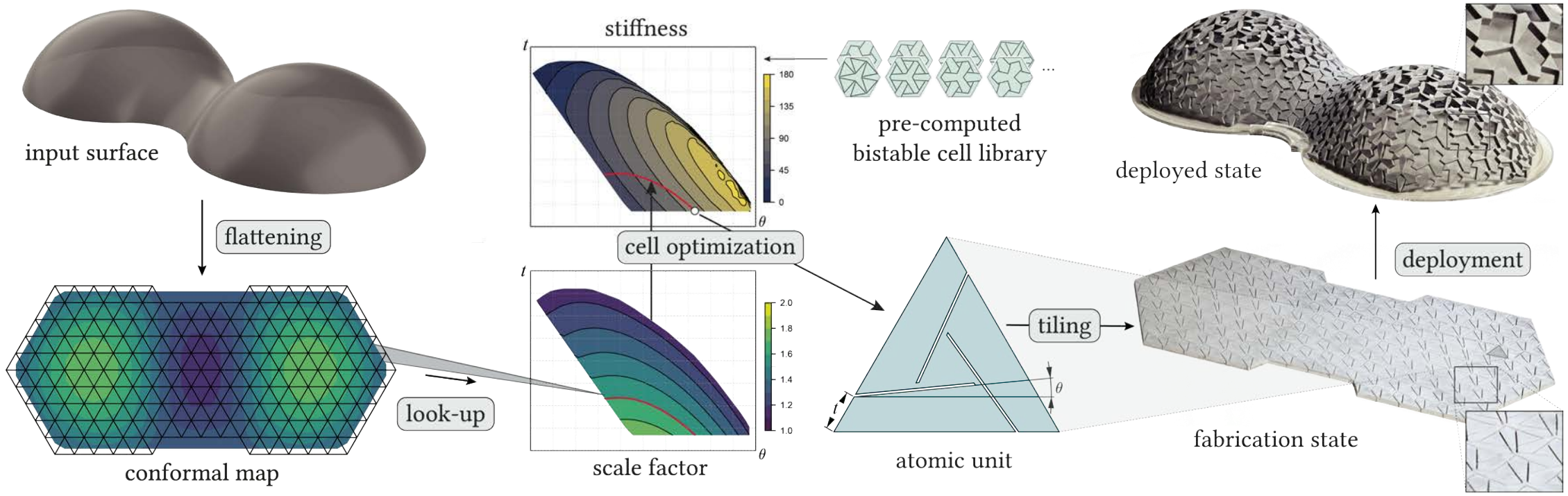
- automatic flattening with optimal area distortion,
- direct manipulation of boundary length or angle,
- exact preservation of sharp corners,
- seamless cone parameterization,
- uniformization over the unit disk, and
- mapping to a given target shape.

(See examples in Figure 1.) The target shape can be efficiently updated via backsubstitution, providing a new paradigm for conformal parameterization: rather than waiting for a single, predetermined flattening, one can interactively edit or optimize the map—even on meshes with hundreds of thousands of elements. More broadly, BFF is a drop-in replacement for widely-used schemes like least-squares conformal maps (LSCM) while providing sophisticated control over features like shape and area distortion.

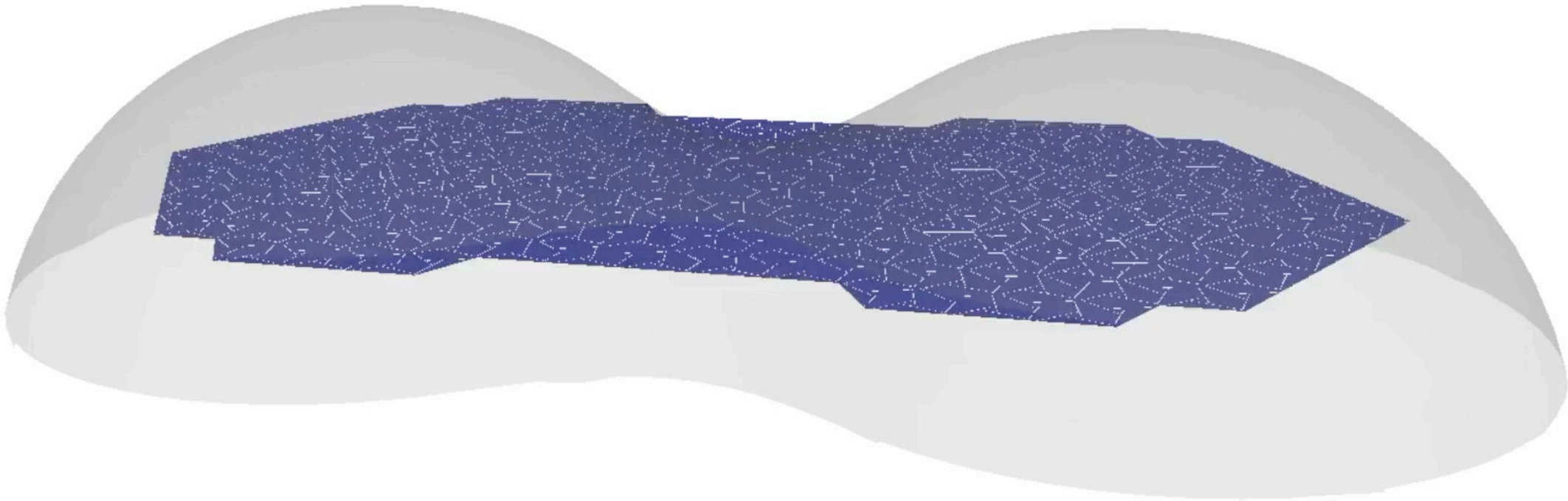
Coarsened Design via Conformal Maps



Coarsened Design of Bistable Auxetics

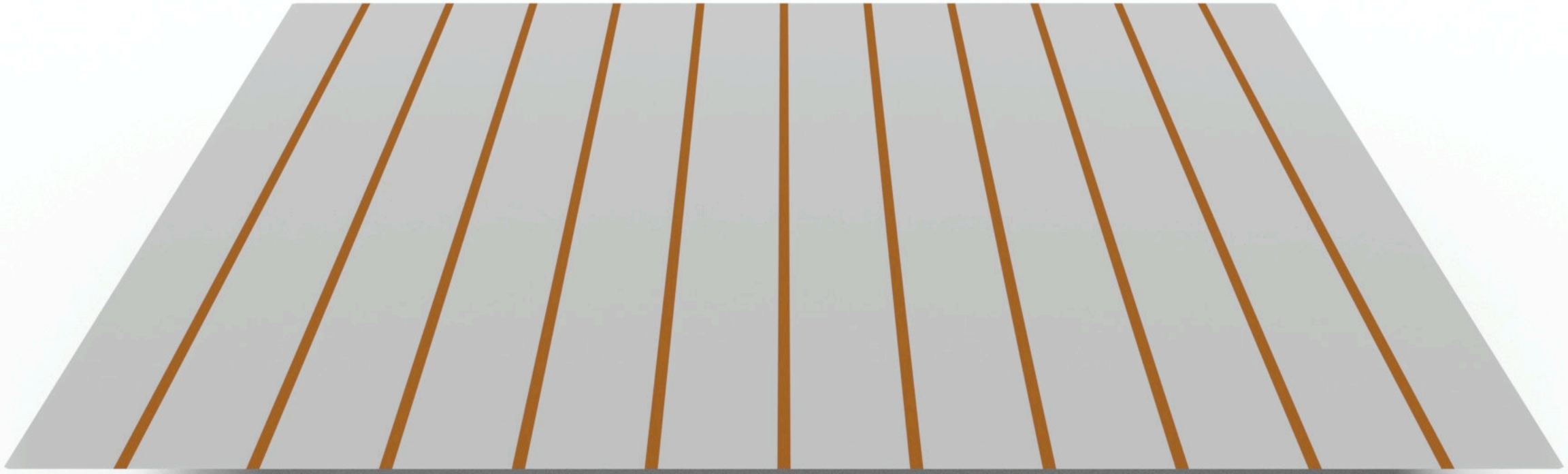


Full Structure Simulation

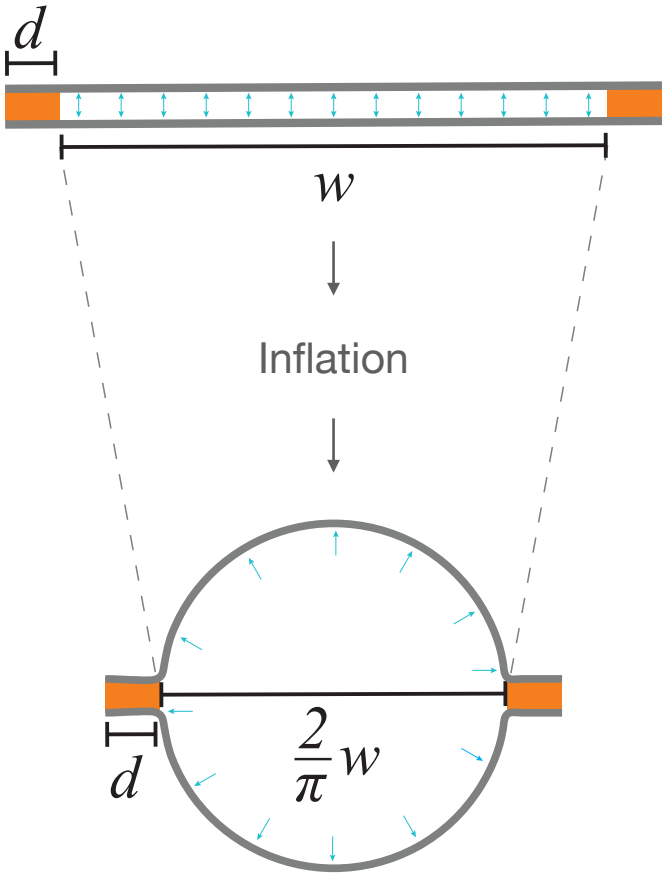


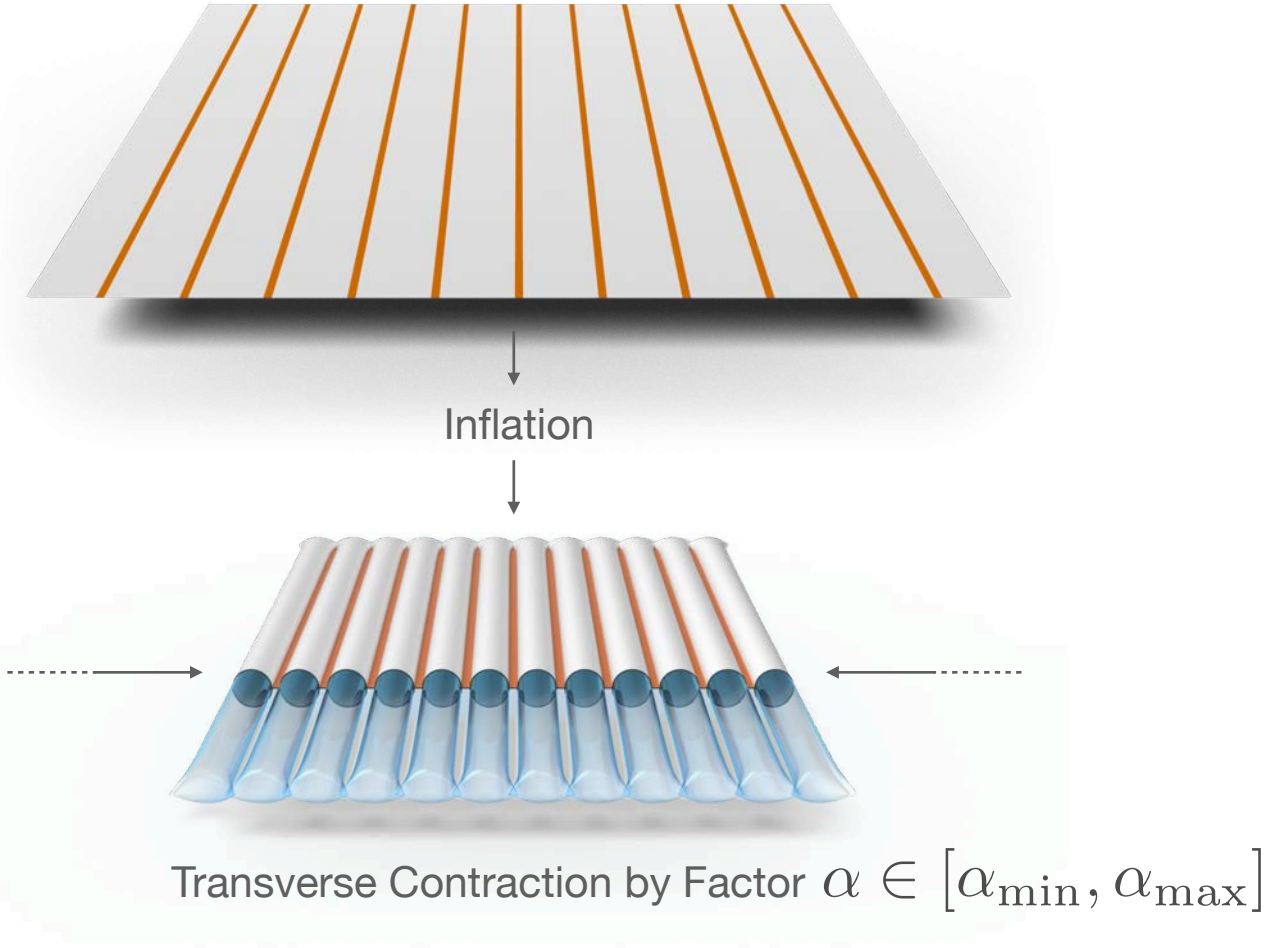
Deployed structure fits target surface well even without shape optimization!

GEOMETRIC ABSTRACTION/HOMOGENIZATION



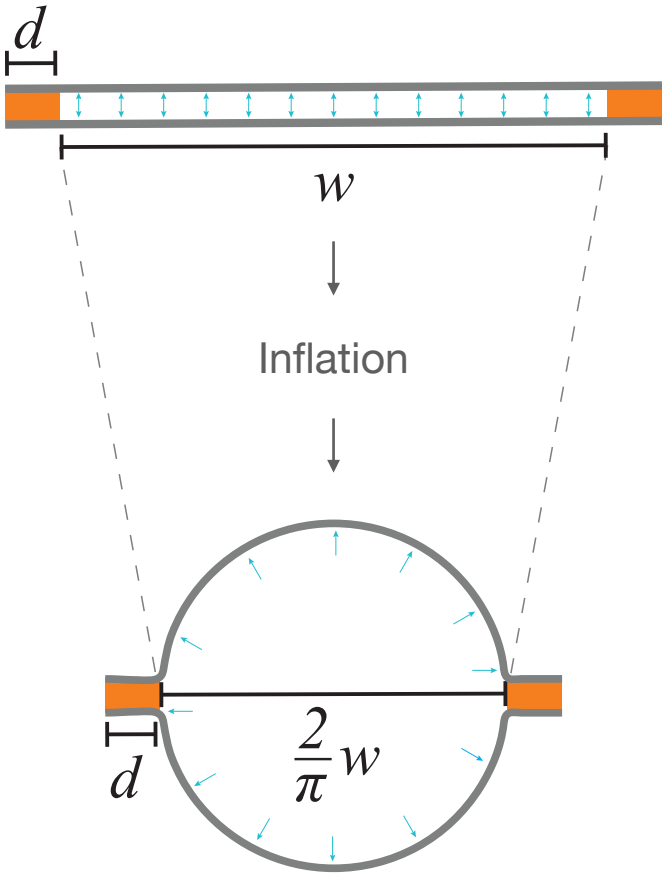
GEOMETRIC ABSTRACTION/HOMOGENIZATION

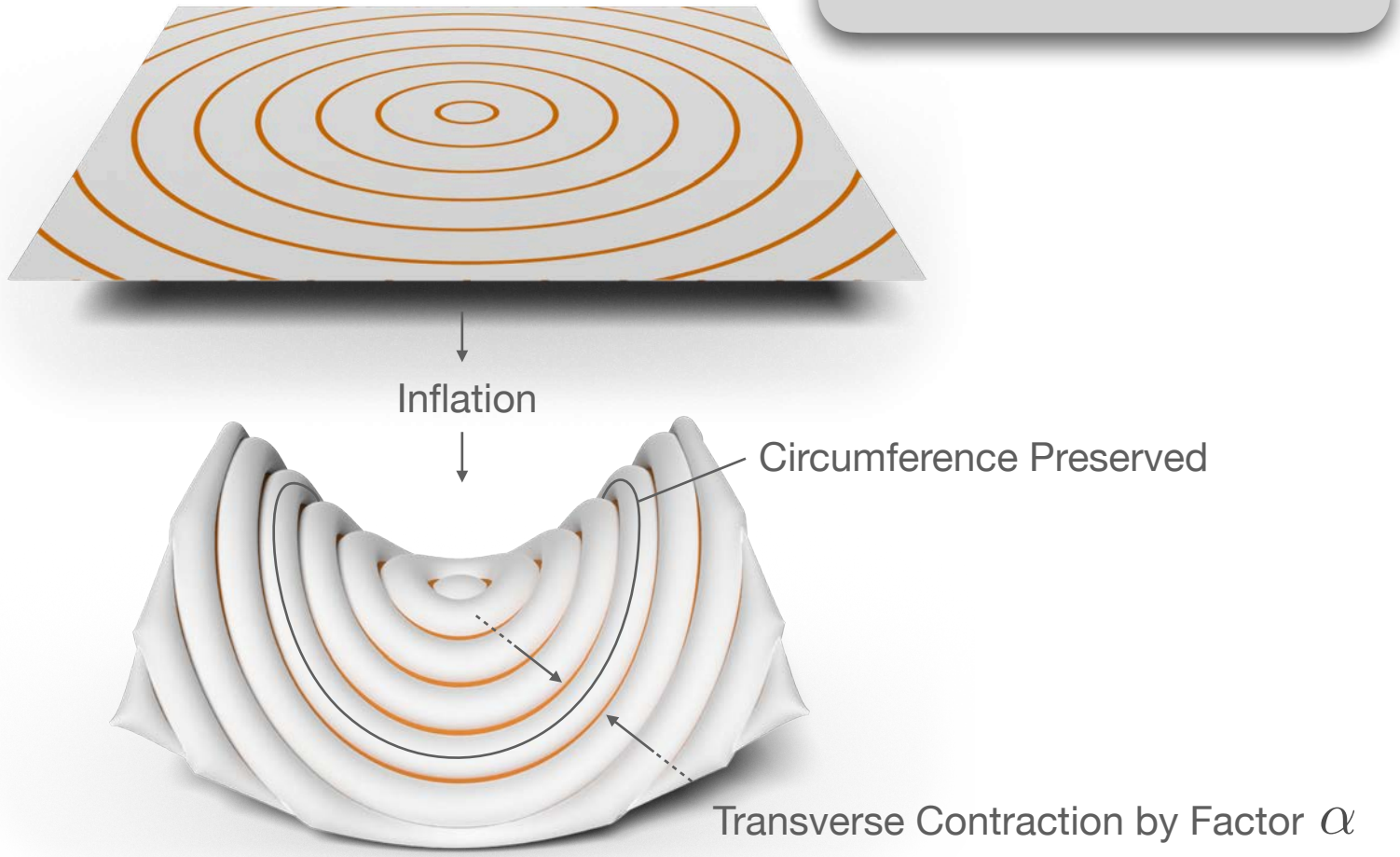




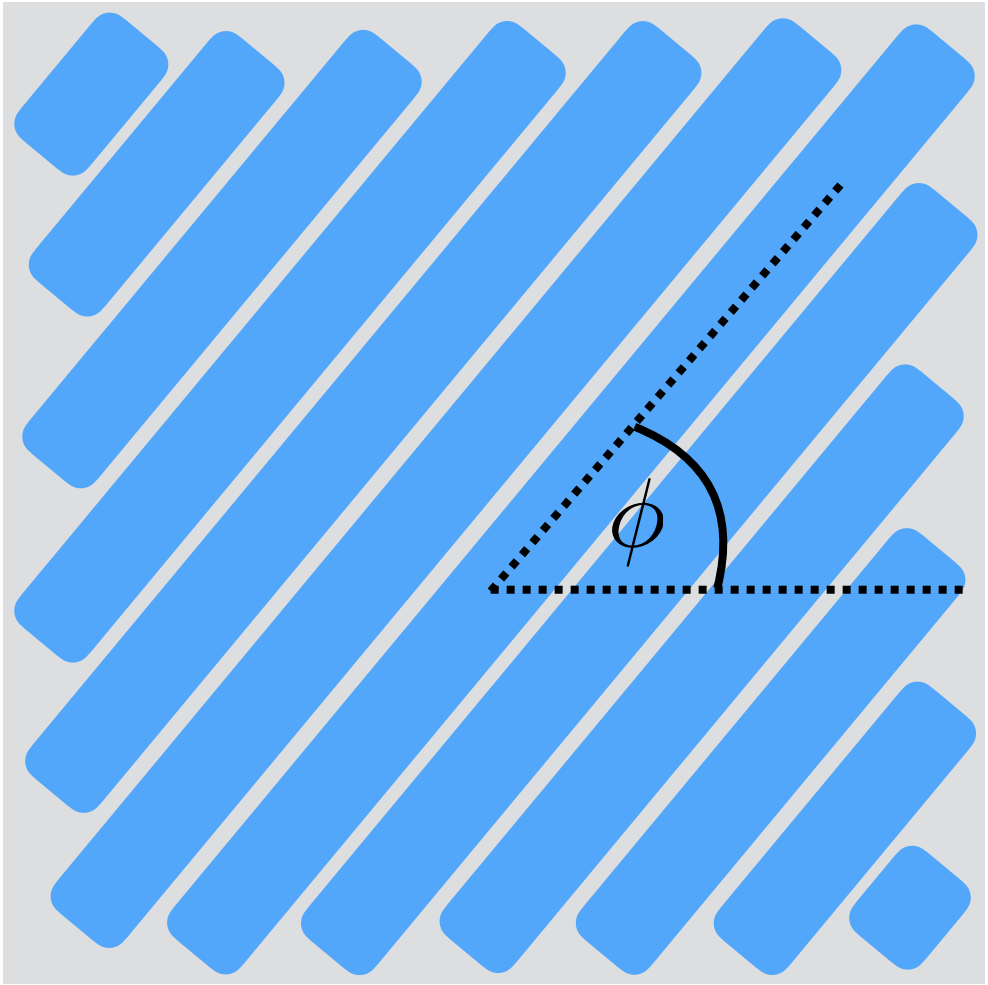
GEOMETRIC ABSTRACTION/HOMOGENIZATION

Homogenized Viewpoint
Inflation assigns a new metric to the flat sheet!





SURFACE CONTRACTION ANALYSIS



Map from flat sheet to inflated sheet midsurface:

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

Jacobian takes the form:

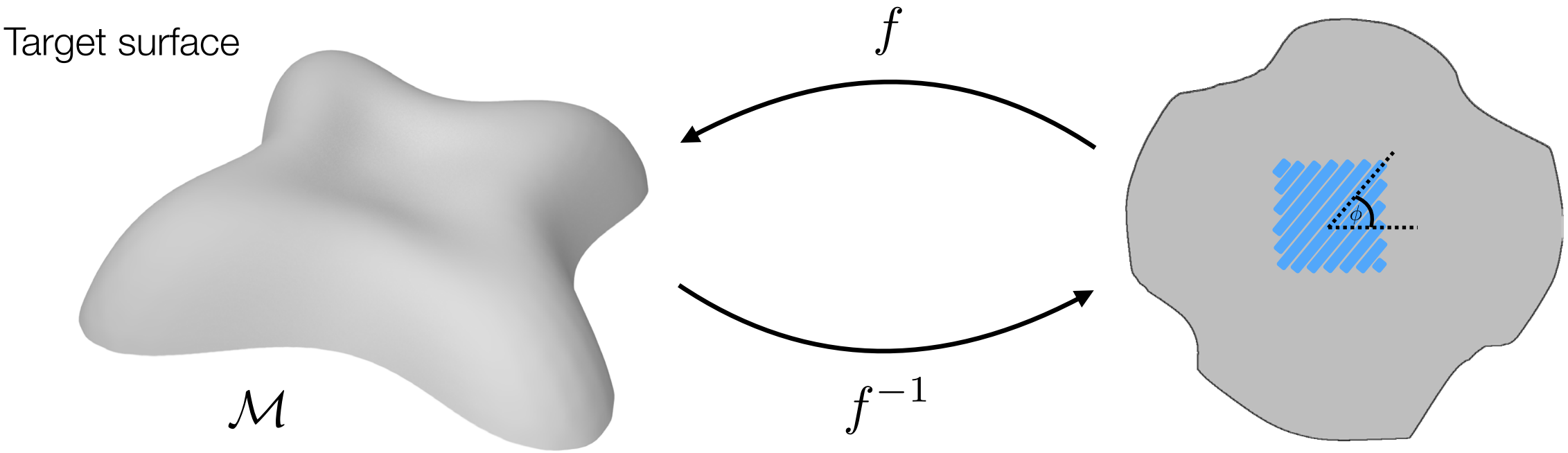
Admissible singular values

$$\nabla f = U \begin{bmatrix} 1 & 0 \\ 0 & \alpha \\ 0 & 0 \end{bmatrix} R(\phi)^T$$

$$U \in SO(3), \quad R \in SO(2)$$

$$\alpha \in [\alpha_{\min}, \alpha_{\max}]$$

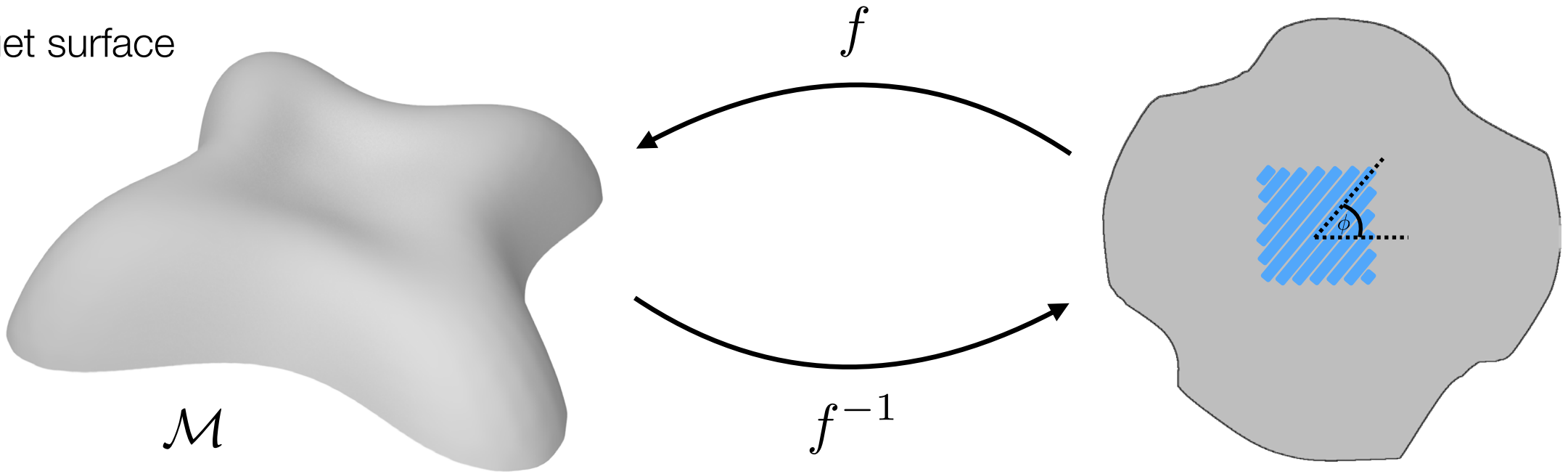
INVERSE DESIGN: PARAMETRIZATION APPROACH



Not an off-the-shelf parametrization algorithm!

INVERSE DESIGN: PARAMETRIZATION APPROACH

Target surface



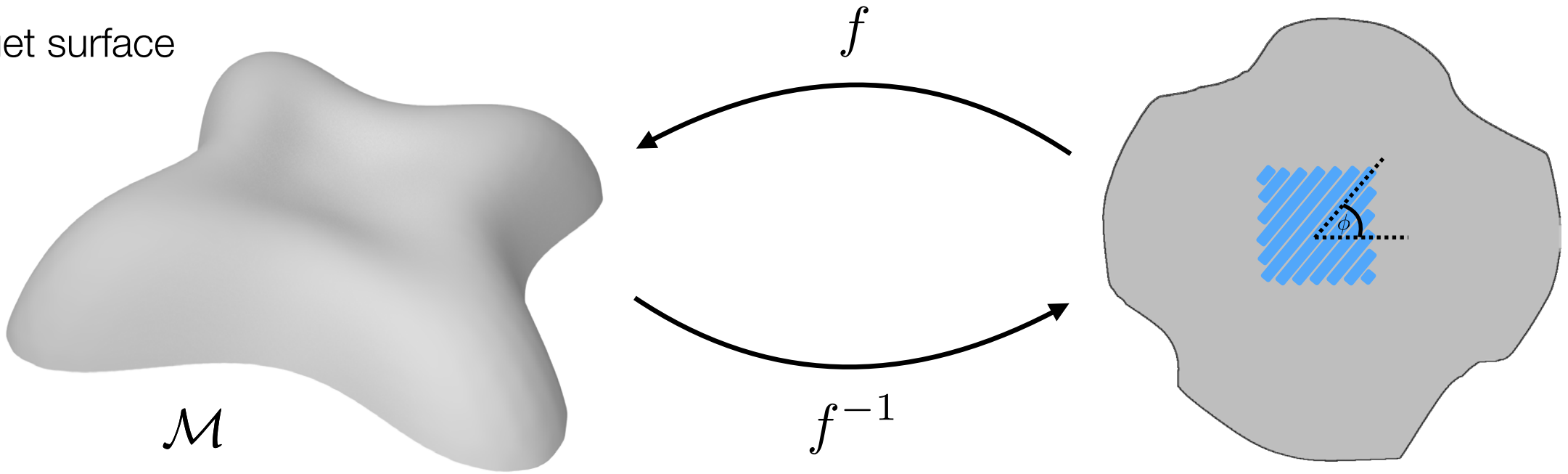
Goal: find **flattening** $f^{-1} : \mathcal{M} \to \mathbb{R}^2$

Such that:

$$\nabla f^{-1} = R(\phi) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\alpha} & 0 \end{bmatrix} U^T \qquad \begin{aligned} &\phi : \mathcal{M} \to \mathbb{R}, \quad U : \mathcal{M} \to SO(3), \\ &\alpha : \mathcal{M} \to [\alpha_{\min}, \alpha_{\max}] \end{aligned}$$

INVERSE DESIGN: PARAMETRIZATION APPROACH

Target surface



Minimize **singular-value-fitting** parametrization energy:

(Singular value σ of **flattening** is $1/\alpha$)

$$f^{-1} = \operatorname{argmin} \int_{\mathcal{M}} (\sigma_1 - 1)^2 + \left(\sigma_0 - [\sigma_0]_{\sigma_{\min}}^{\sigma_{\max}}\right)^2 \, \mathrm{d}A$$

INVERSE DESIGN: PARAMETRIZATION APPROACH

Singular Value Fitting

Local-Global Approach


Universal inverse design of surfaces with thin nematic elastomer sheets

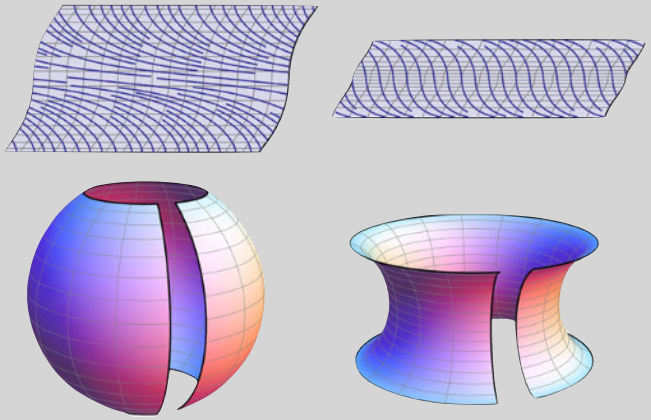
[Aharoni et al. 2018]

ARAP-style alternating minimization of:

$$\min_{f,U,V,\sigma} \int_{\mathcal{M}} \left\| \nabla f - U \begin{pmatrix} \sigma & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} V^T \right\|_F^2 \mathrm{d}A$$

applied to **Nematic Elastomer Design**.

Analytical Formula in Simple Cases



[Aharoni et al. 2014]

Problems:

- Noisy direction fields
- Ignores extrinsic curvature, which is relevant to the physical system

REGULARIZATIONS FOR PARAMETRIZATION

$$f^{-1} = \operatorname{argmin} \int_{\mathcal{M}} (\sigma_1 - 1)^2 + \left(\sigma_0 - [\sigma_0]_{\sigma_{\min}}^{\sigma_{\max}}\right)^2 \, \mathrm{d}A$$

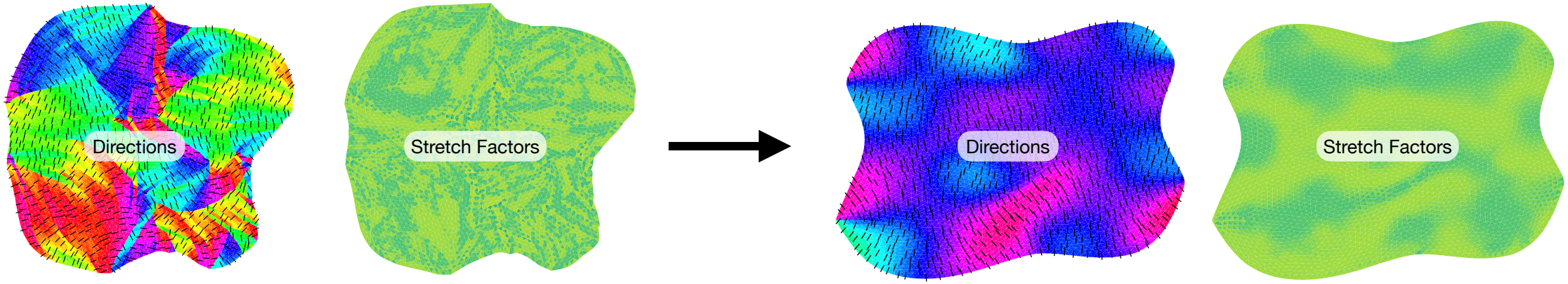
Singular Value Fitting

$$+ w_\phi \int_{\mathcal{M}} \left\| \nabla \begin{bmatrix} \cos(2\phi) \\ \sin(2\phi) \end{bmatrix} \right\|_F^2 \, \mathrm{d}A$$

Direction Regularization

$$+ w_\sigma \int_{\mathcal{M}} \|\nabla \sigma_0\|^2 \, \mathrm{d}A$$

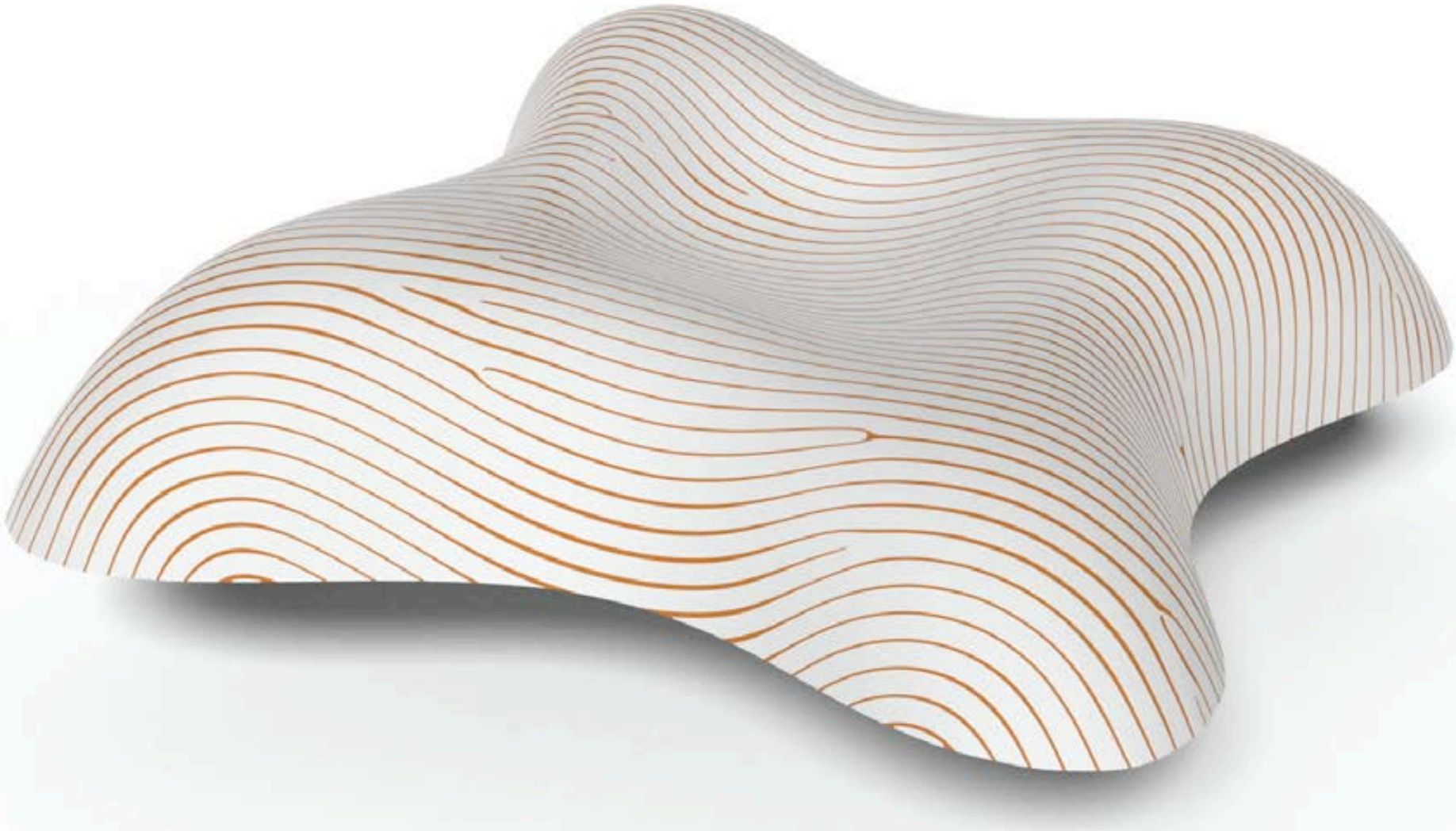
Stretch Regularization



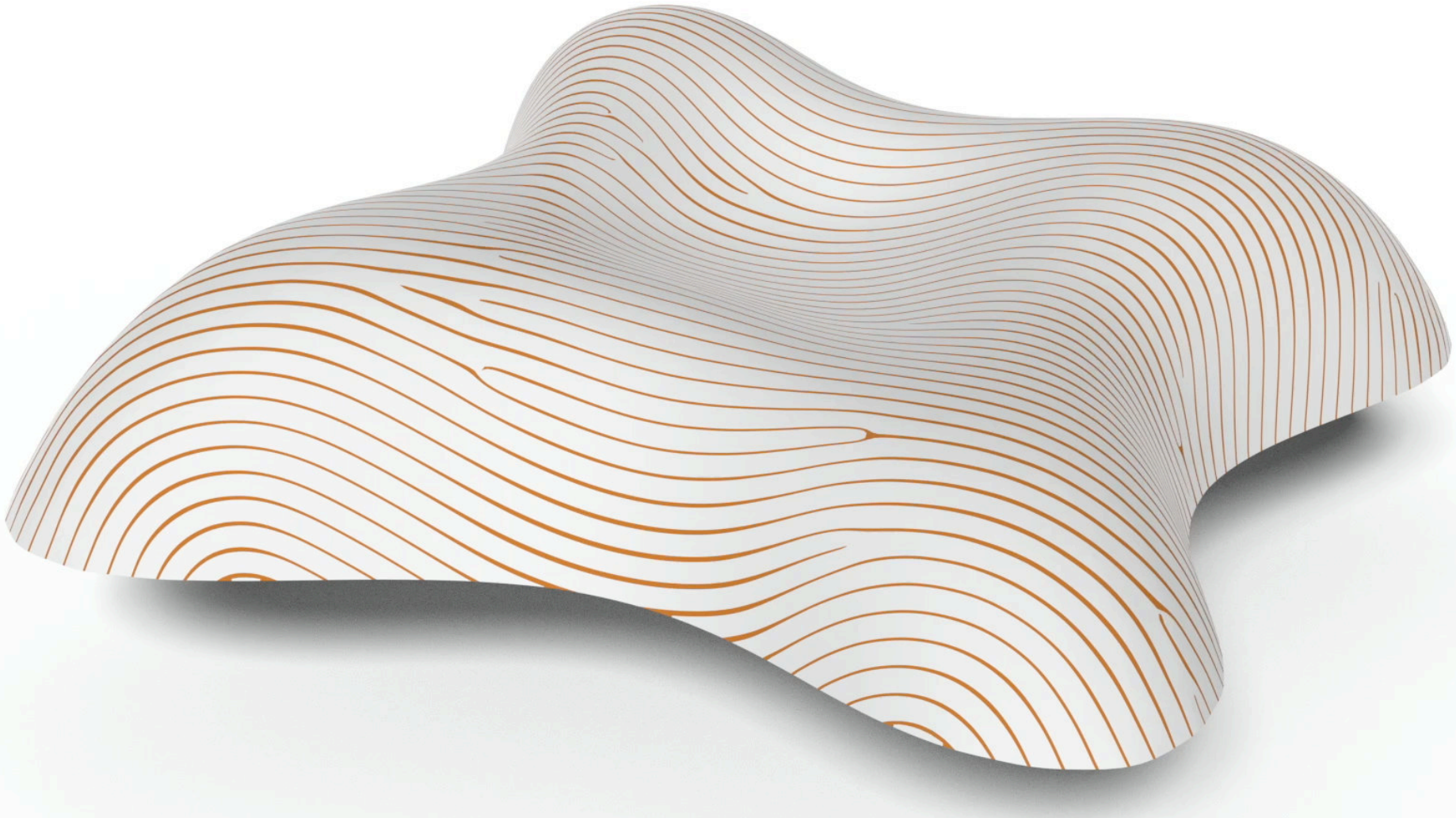
Singular-Value Fitting Only

Regularized

BENDING EFFECTS: INSUFFICIENCY OF METRIC-BASED DESIGN



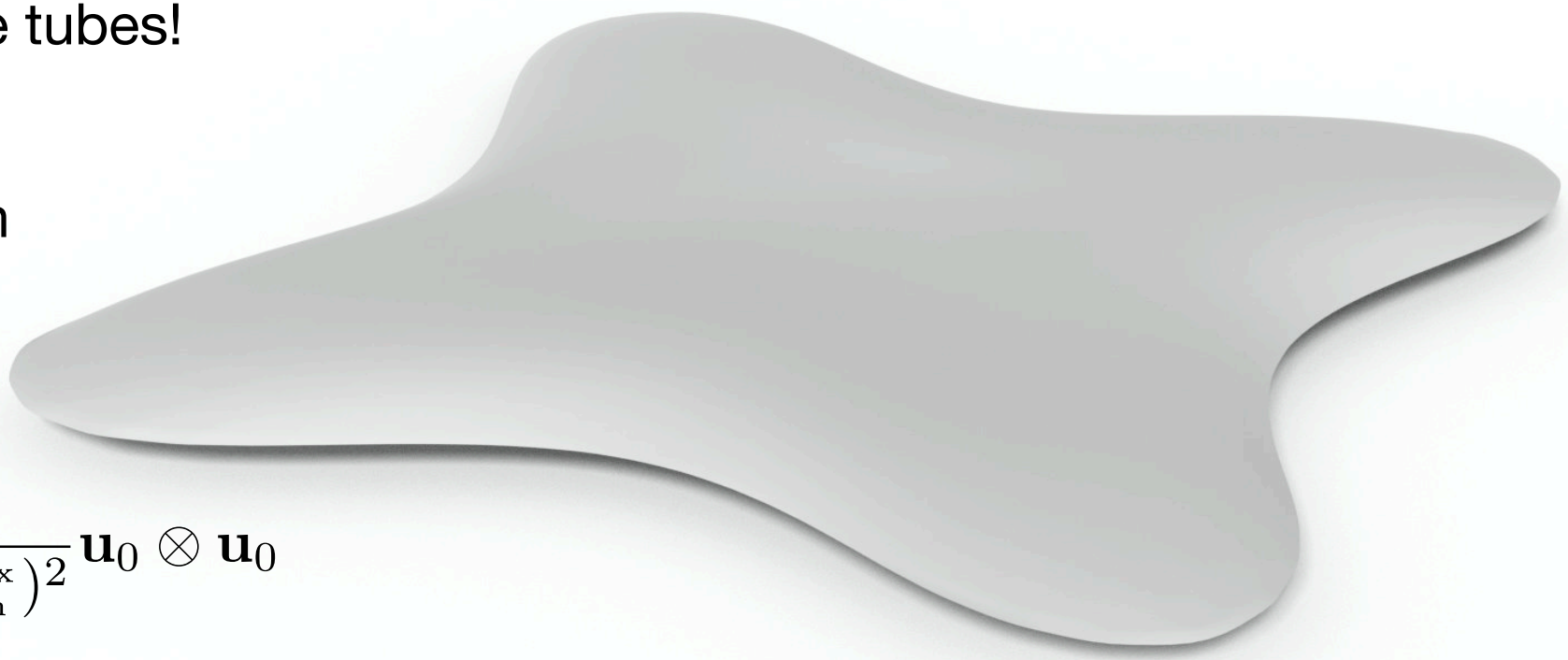
BENDING EFFECTS: INSUFFICIENCY OF METRIC-BASED DESIGN



BENDING EFFECTS: INSUFFICIENCY OF METRIC-BASED DESIGN

The target shape **is** robustly encoded in the metric used to generate the tubes!

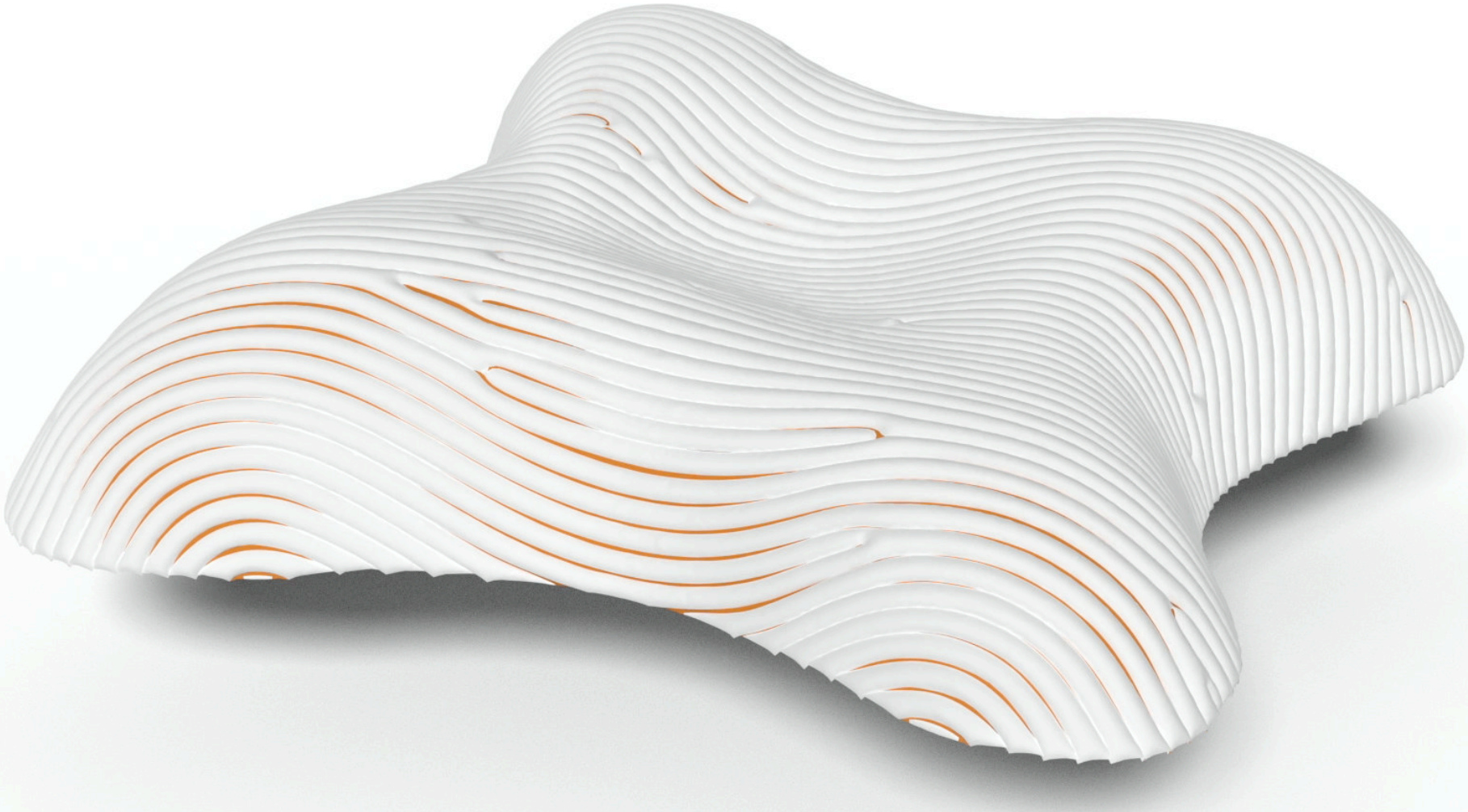
Confirmed by solving for an immersion of the sheet with this metric:



$$g := \mathbf{u}_1 \otimes \mathbf{u}_1 + \frac{1}{([\sigma_0]_{\sigma_{\min}}^{\sigma_{\max}})^2} \mathbf{u}_0 \otimes \mathbf{u}_0$$

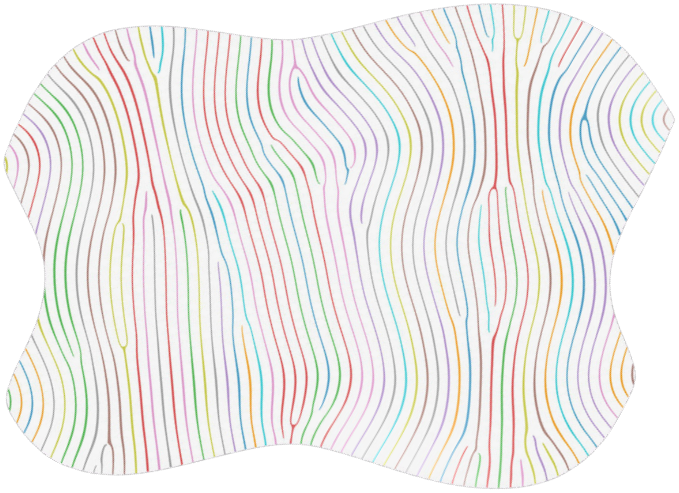
$$\min_f \frac{1}{2} \int_{\mathcal{M}} \left\| \nabla f^T \nabla f - g \right\|_F^2 \, \mathrm{d}A + E_{\text{bending}}(f)$$

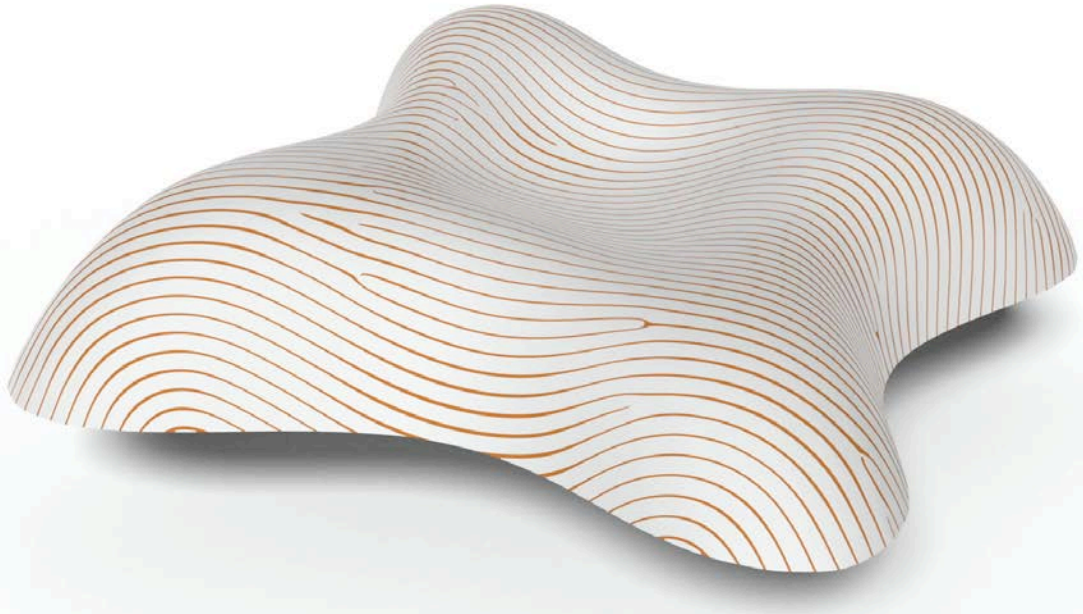
BENDING EFFECTS: INSUFFICIENCY OF METRIC-BASED DESIGN



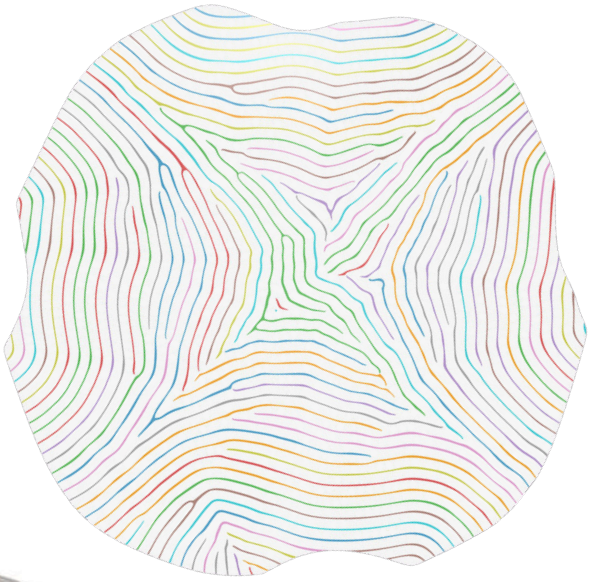
EFFECT OF BENDING REGULARIZATION

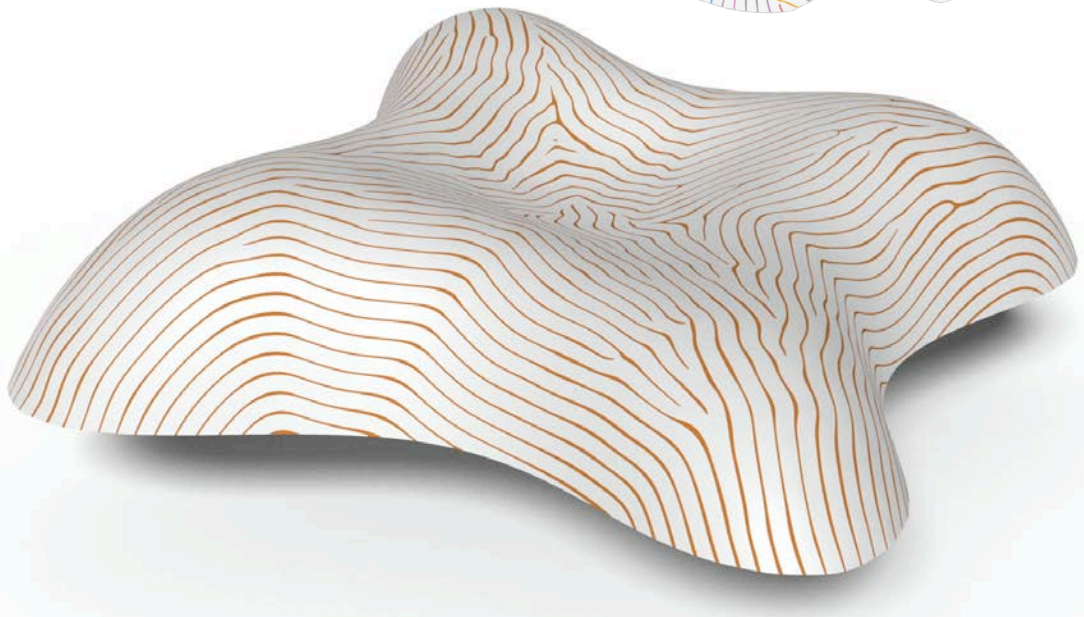
Without Bending Regularization





With Bending Regularization





PARALLEL TUBES PARAMETRIZATION ENERGY

$$f^{-1} = \operatorname{argmin} \int_{\mathcal{M}} (\sigma_1 - 1)^2 + \left(\sigma_0 - [\sigma_0]_{\sigma_{\min}}^{\sigma_{\max}}\right)^2 \, \mathrm{d}A$$

Singular Value Fitting

$$+ w_\phi \int_{\mathcal{M}} \left\| \nabla \begin{bmatrix} \cos(2\phi) \\ \sin(2\phi) \end{bmatrix} \right\|_F^2 \, \mathrm{d}A$$

Direction Regularization

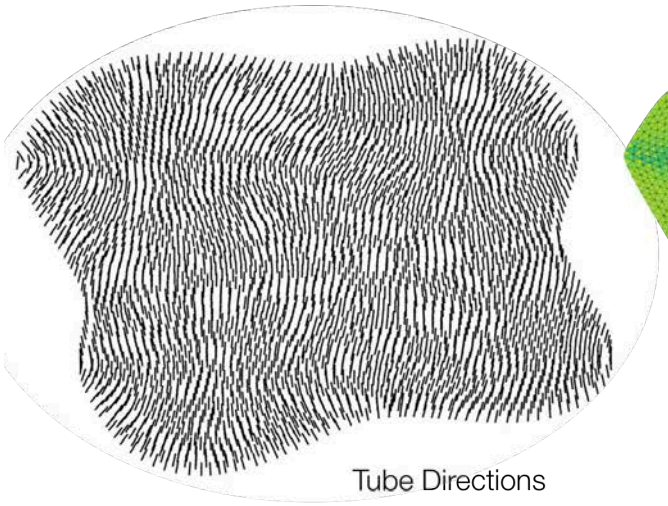
$$+ w_\sigma \int_{\mathcal{M}} \|\nabla \sigma_0\|^2 \, \mathrm{d}A$$

Stretch Regularization

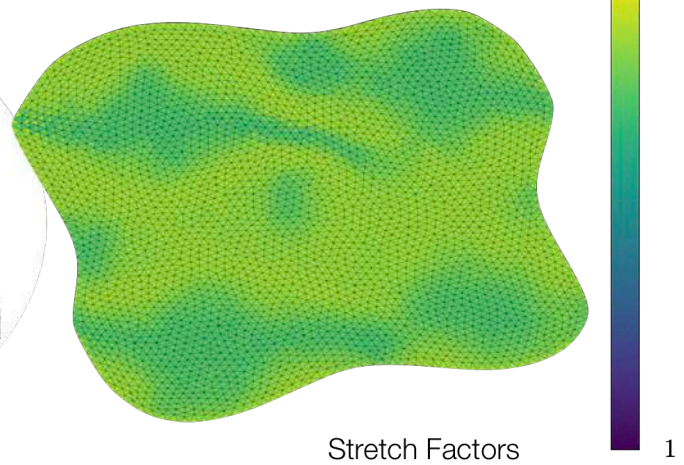
$$+ w_\kappa \int_{\mathcal{M}} (\mathbf{v}_1^T S \mathbf{v}_1)^2 \, \mathrm{d}A$$

Bending Regularization

GENERATING AIR CHANNELS



Tube Directions



Stretch Factors



Stripe Patterns on Surfaces

Felix Knöppel, TU Berlin · Keenan Crane, Columbia University · Ulrich Pinkall, TU Berlin · Peter Schröder, Caltech

Abstract

Stripe patterns are ubiquitous in nature, describing macroscopic phenomena such as stripes on plants and animals, down to material impurities on the atomic scale. We propose a method for synthesizing stripe patterns on triangulated surfaces, where singularities are automatically inserted in order to achieve user-specified orientation and line spacing. Patterns are characterized as global minimizers of a convex quadratic energy which is well-defined in the smooth setting. Computation amounts to finding the principal eigenvector of a symmetric positive-definite matrix with the same sparsity as the standard graph Laplacian. The resulting patterns are globally continuous, and can be applied to a variety of tasks in design and texture synthesis.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

Keywords: discrete differential geometry, digital geometry processing, texture synthesis, direction fields, singularities

1 Introduction

A diverse collection of natural phenomena exhibit the same characteristic pattern: uneven stripes of uniform width that bifurcate at isolated points called *edge dislocations* [Kalpakjian and Schmid 2005, pp. 64–66]. Why does this bifurcation occur? Surfaces of revolution (Fig. 3) provide an instructive example: for any given stripe width there is a maximum integer number of stripes that can fit around the circumference at each height h. As the radius r grows or shrinks this number can suddenly jump—stripes must therefore branch or coalesce in order to maintain a continuous transition between neighboring regions.

Figure 1: *In general, patterns without singularities exhibit uneven spacing, and patterns with even spacing must have singularities.*

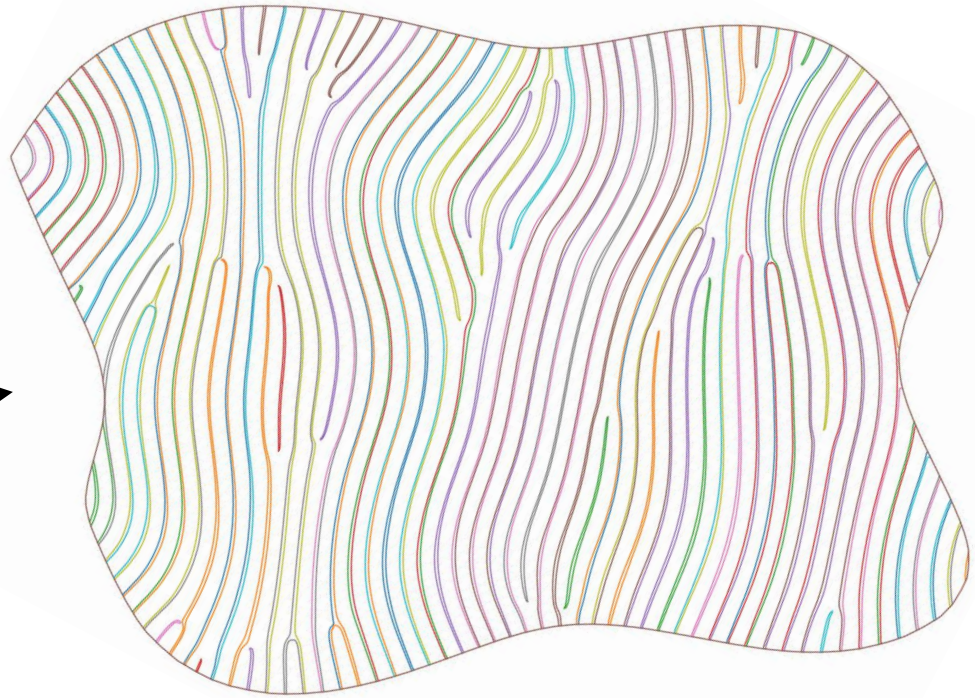
Figure 2: *Natural phenomena like cacti (left) exhibit characteristic branching patterns in an effort to maintain evenly-spaced features. Right: Our algorithm allows one to quickly and automatically synthesize similar patterns that seamlessly cover arbitrary surfaces.*

Our approach to stripe pattern synthesis is closely related to existing methods for field-aligned parameterization (Sec. 1.1). The main point of departure is that existing algorithms try to *prevent* singularities that were not specified as part of the input, whereas we intentionally allow new singularities, which are essential to modelling natural phenomena. In particular, we use a formulation akin to Ray *et al.* [2006], but omit both the curl-correction step and the unit-norm constraint. As a result, we can formulate our problem via a simple convex quadratic energy; the practical payoff is an algorithm about an order of magnitude faster than those based on nonconvex constraints or mixed integer programming (Sec. 5.1). The final algorithm is simple to implement (App. C) requiring no global cutting or integer jumps, is robust to errors in the input (Fig. 16), and requires little work beyond factoring a single sparse matrix. The output is an angle α at each triangle corner which can be used to drive a periodic texture or shader. We also introduce a novel interpolation scheme (Sec. 4.2) that ensures the pattern is globally continuous (C^0) away from a finite collection of isolated singular points.

Figure 3: *To maintain constant stripe width, each "level" of this surface of revolution must exhibit a different number of stripes. Judiciously placed branch points are therefore essential to keeping the pattern continuous. Inset: each pair of stripes from the second level perfectly branches into three.*

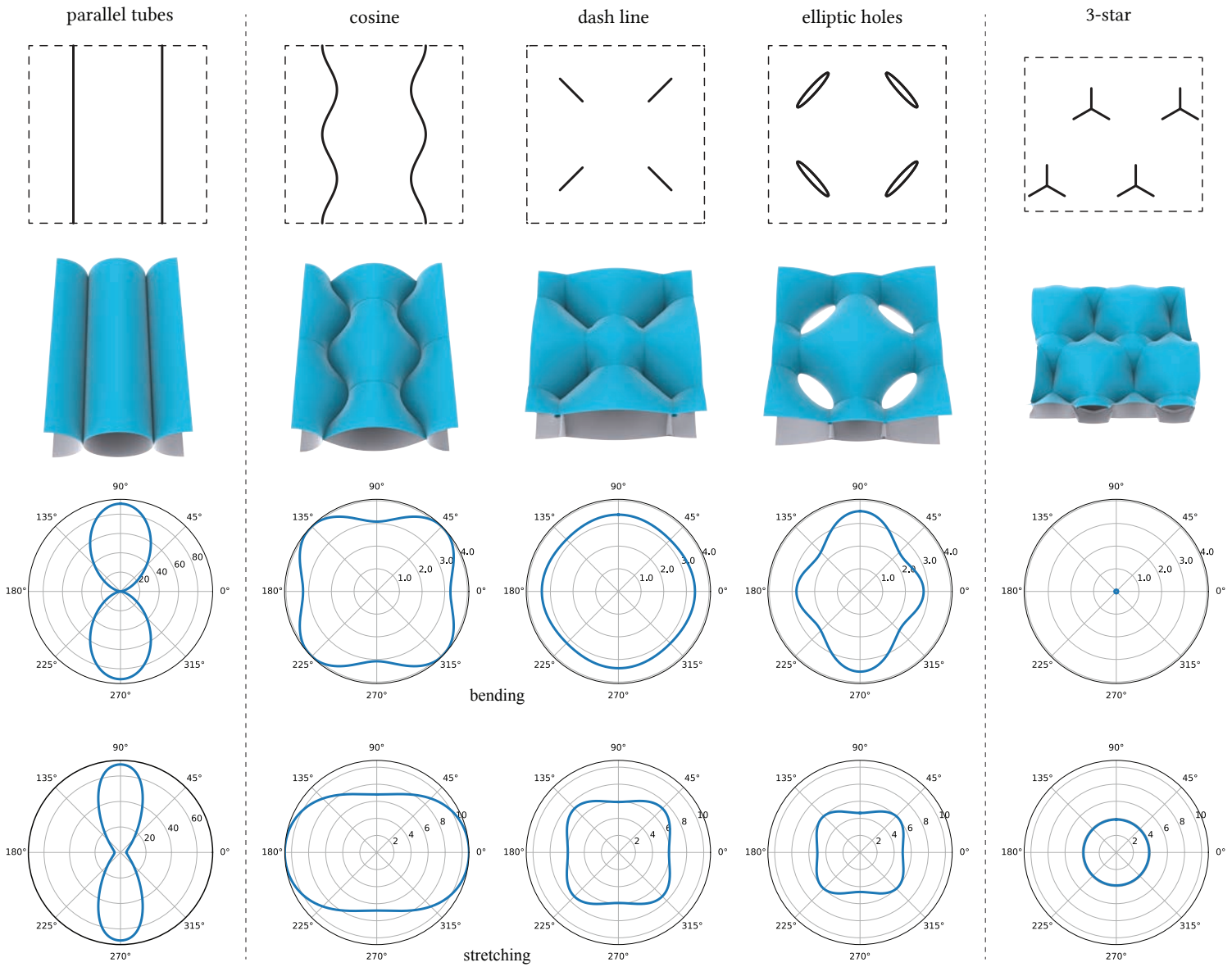
Stripe Patterns on Surfaces
[Knöppel et al. 2015]

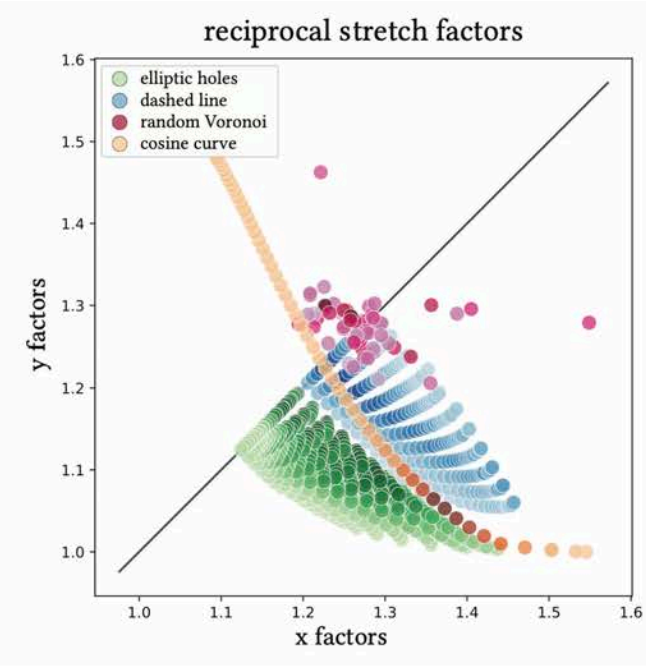


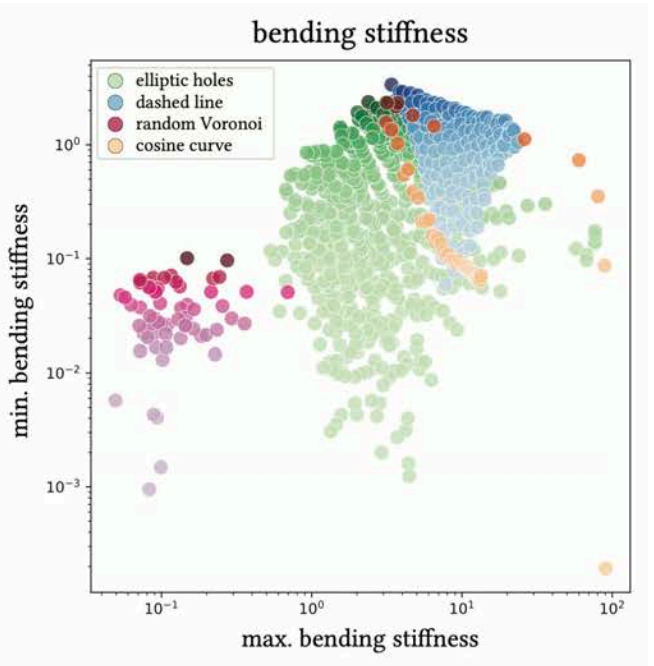


Initial Design Guess

HOMOGENIZATION RESULTS







GENERAL INFLATABLES PARAMETRIZATION ENERGY

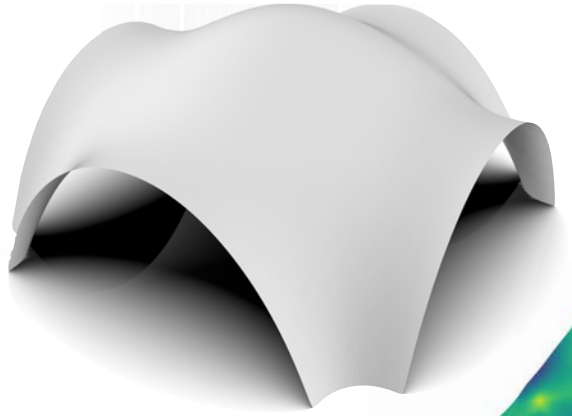
$$\min_{f,\mathbf{p}} E_{\text{flat}}(f,\mathbf{p}) \quad \text{s.t.} \quad \mathbf{p}_{\min} \le \mathbf{p} \le \mathbf{p}_{\max}$$

$$E_{\text{flat}}(f,\mathbf{p}) := E_{\text{fit}}(f,\mathbf{p}) + w_b E_{\text{bend}}(f,\mathbf{p}) + E_{\text{reg}}(f,\mathbf{p})$$

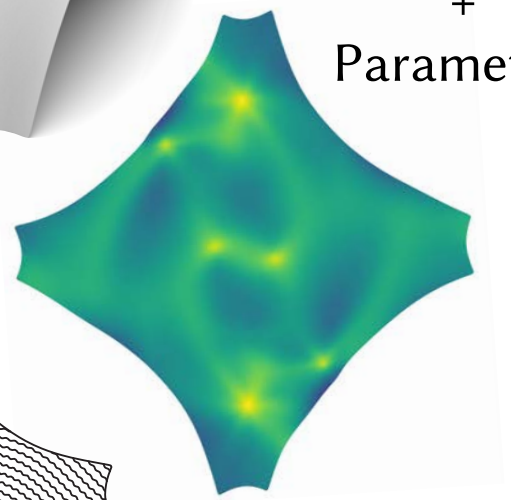
$$E_{\text{fit}}(f,\mathbf{p}) := \frac{1}{2|\mathcal{M}|}\int_{\mathcal{M}} \left(\sigma_1 - 1/s_1(\mathbf{p})\right)^2 + \left(\sigma_2 - 1/s_2(\mathbf{p})\right)^2 \, \mathrm{d}A$$

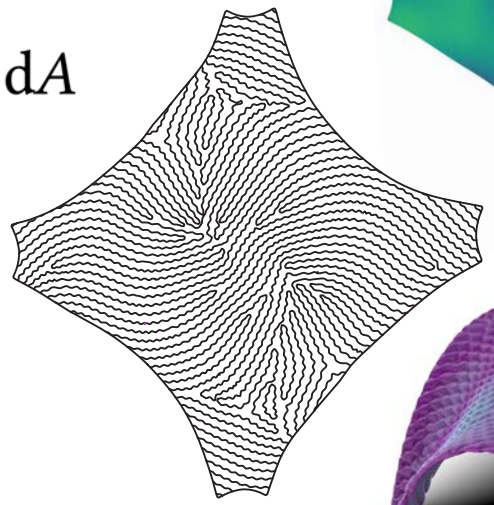
$$E_{\text{bend}}(f,\mathbf{p}) := \frac{1}{|\mathcal{M}|}\int_{\mathcal{M}}\int_0^{2\pi} \kappa_{\mathcal{M}}(\alpha)^2 k_b(\alpha;\mathbf{p}) \mathrm{d}\alpha \, \mathrm{d}A$$

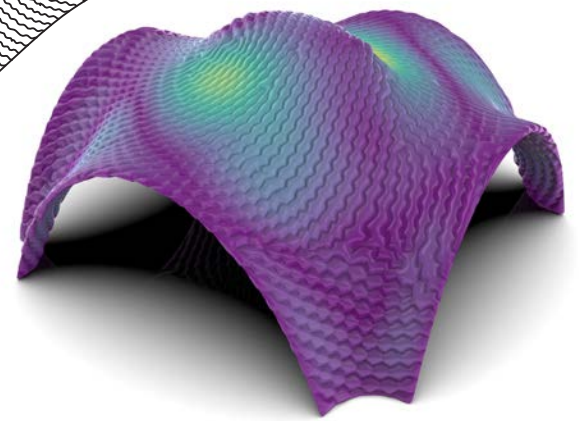
$$E_{\text{reg}}(f,\mathbf{p}) := \frac{w_\phi}{4}\int_{\mathcal{M}} \left\| \nabla \begin{bmatrix} \cos(2\phi) \\ \sin(2\phi) \end{bmatrix} \right\|_F^2 \mathrm{d}A + \frac{w_s}{2}\int_{\mathcal{M}} \|\nabla \mathbf{p}\|_F^2 \, \mathrm{d}A$$



Flattening
+
Parameters



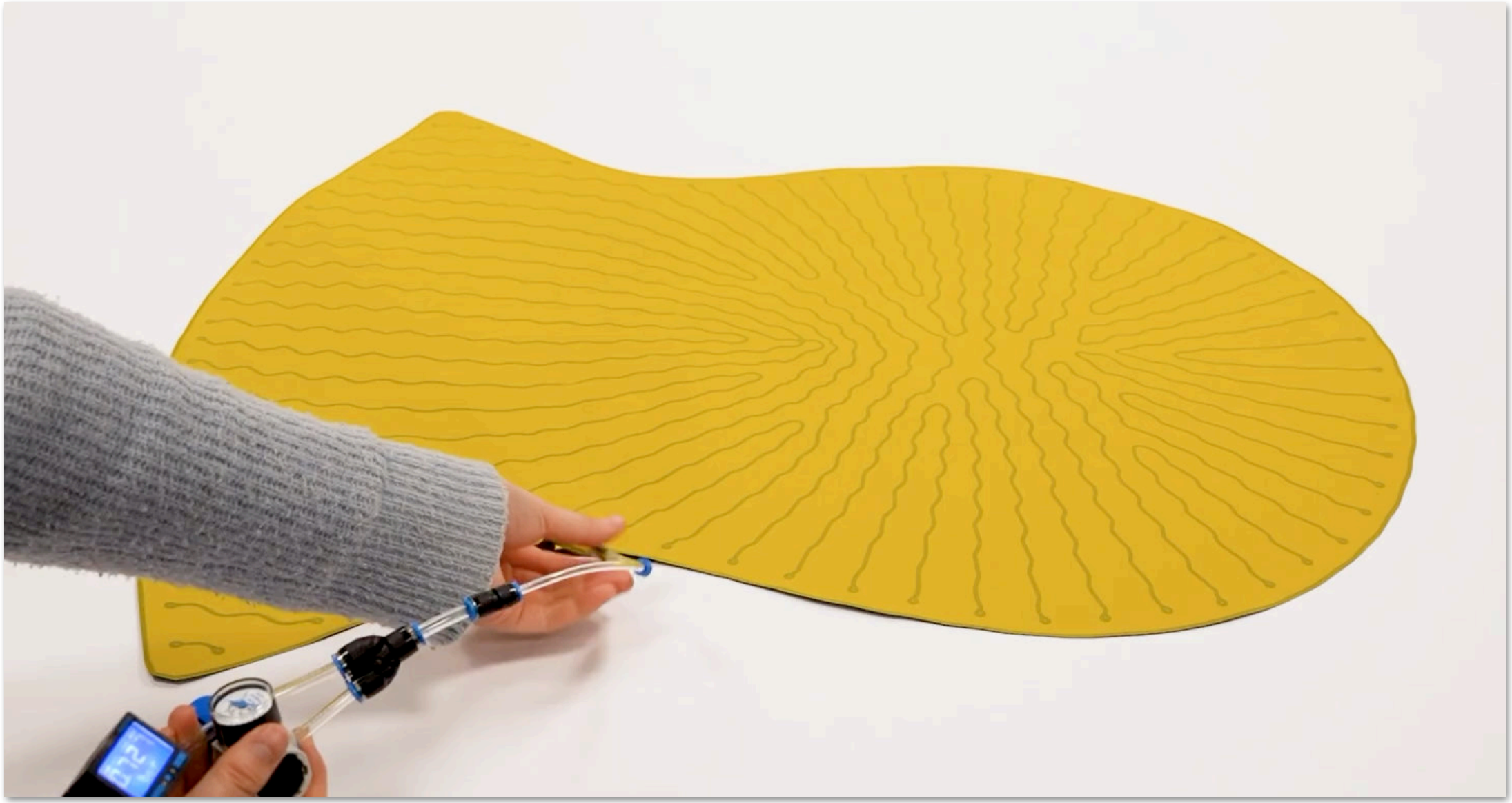




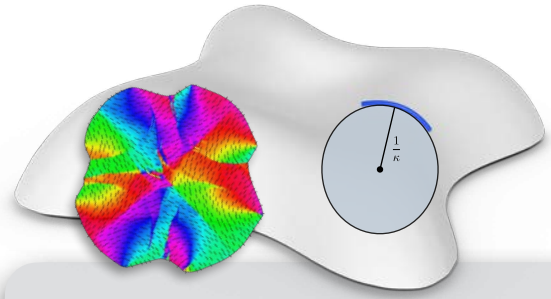
Model zoo - simulation



PHYSICAL INFLATION



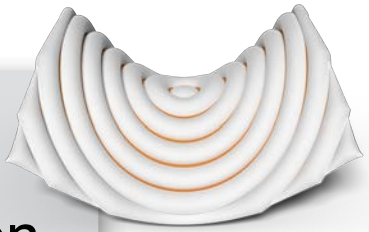
Optimal Design Pipeline For Deployable Structures



Homogenization-Based Initialization

Determine design **topology** and reasonable initial shape parameters

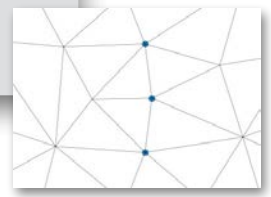
Differentiable Physics Simulation



Design Candidate

Performance, Derivatives

Shape Optimization



Fabrication-Ready Design