CS457 Geometric Computing

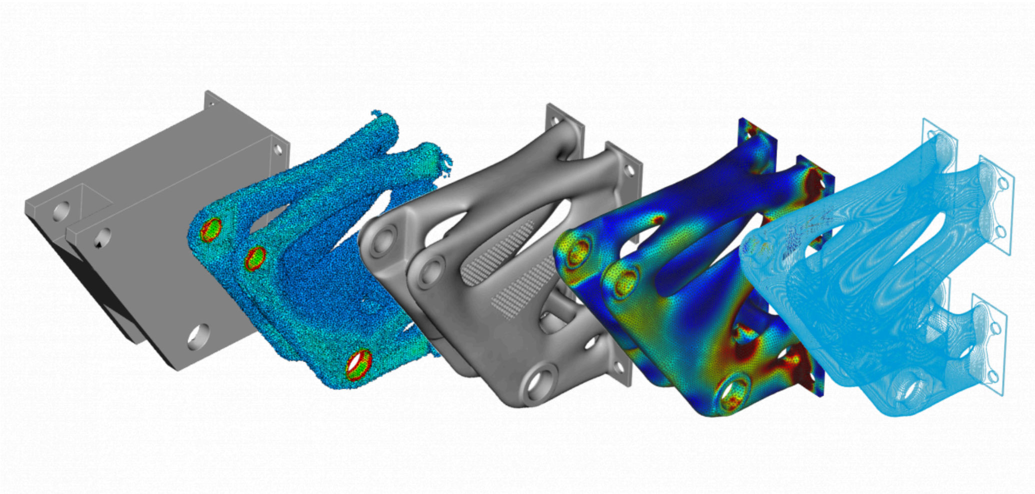
8 - Inverse Design

Mark Pauly

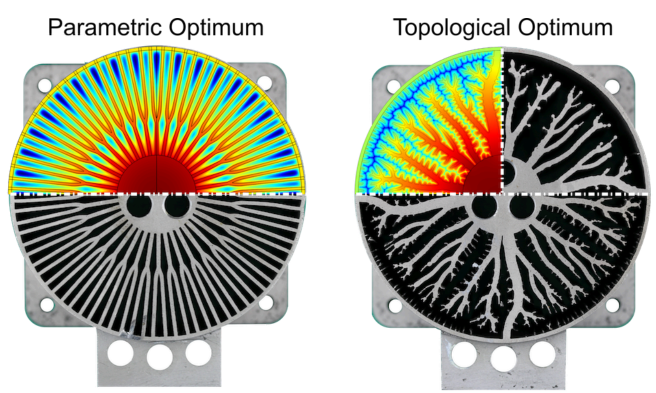
Geometric Computing Laboratory - EPFL

Inverse Design Examples

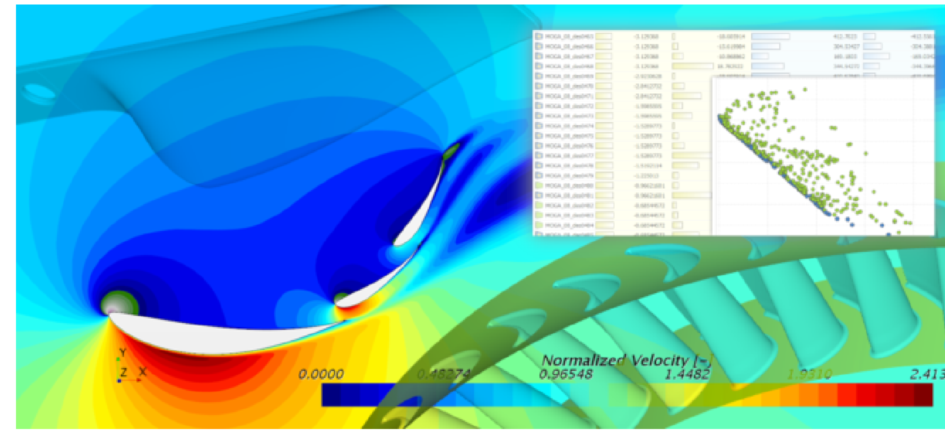
- Mechanical Parts: Maximize strength at minimal weight



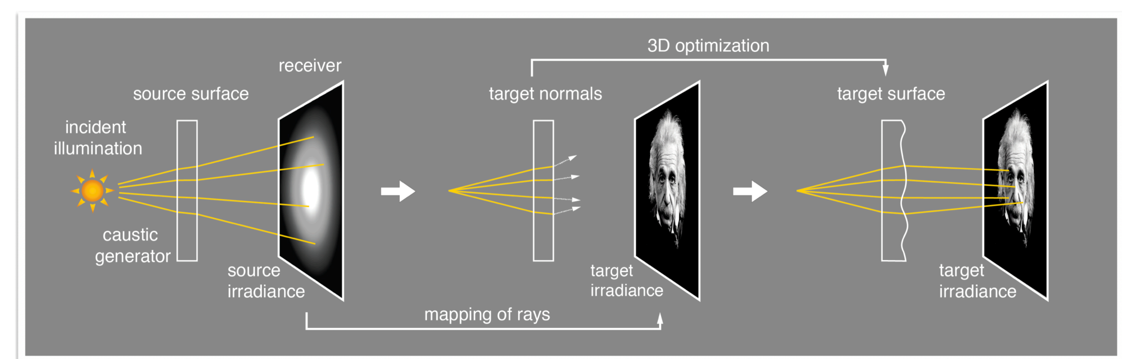
- Heat Sinks: Maximize heat flow at minimal size and weight



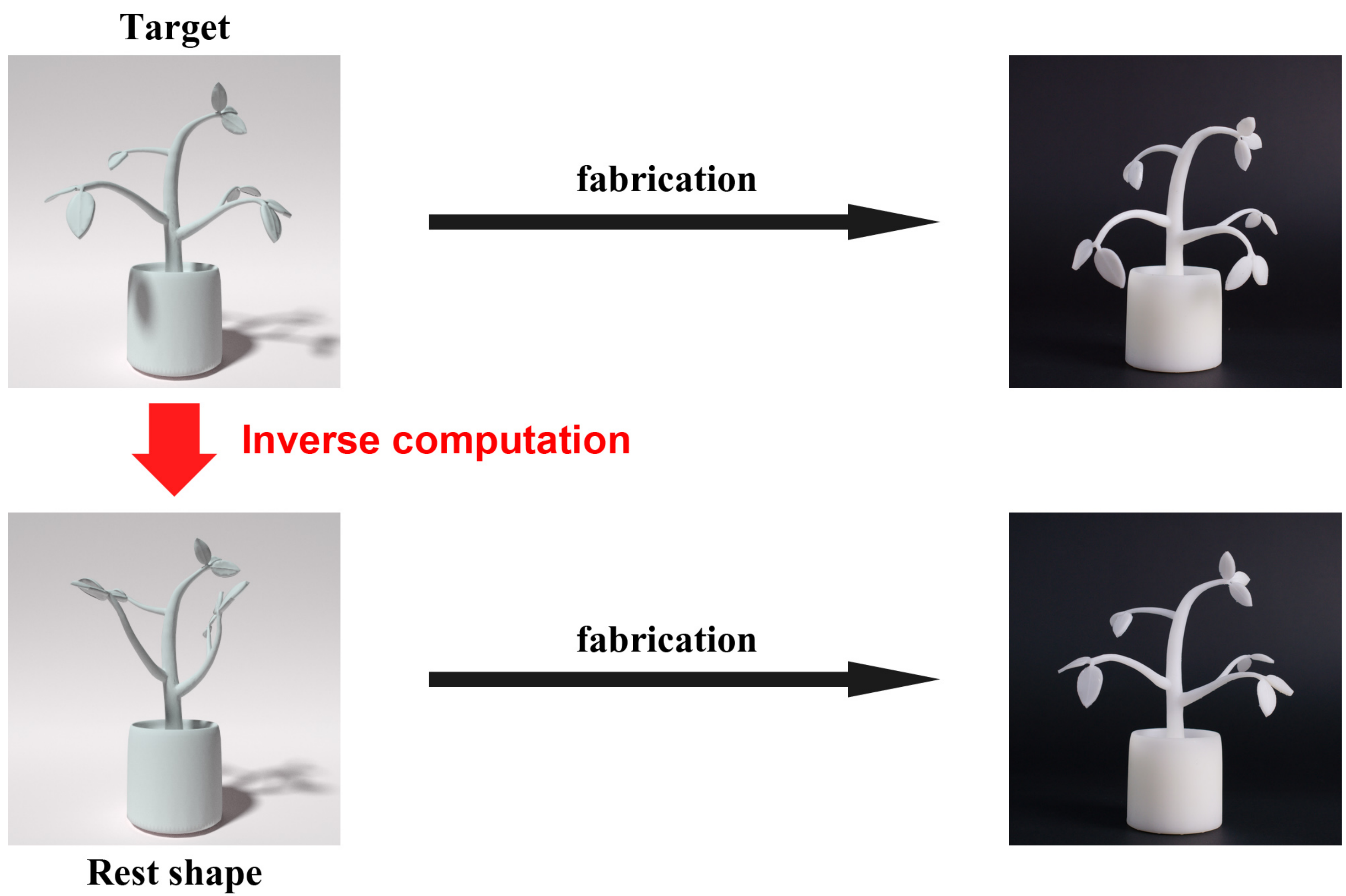
- Fluid Dynamics: Minimize drag (under various constraints)



- Freeform Lens Design: Optimize optical paths



Challenge II: Inverse Elastic Shape Optimization



An Asymptotic Numerical Method for Inverse Elastic Shape Design

Challenge II: Inverse Elastic Shape Optimization

- How can we simulate elastic objects?
  - How can we *model elastic deformation*?
  - How can we *discretize volumetric solids*?
  - How can we *find equilibrium states*?
- How can we optimize elastic objects?
  - How can we *modify the rest shape*?
  - How can we *find the best modification*?
    - such that the object deforms into a given target geometry under given external forces.

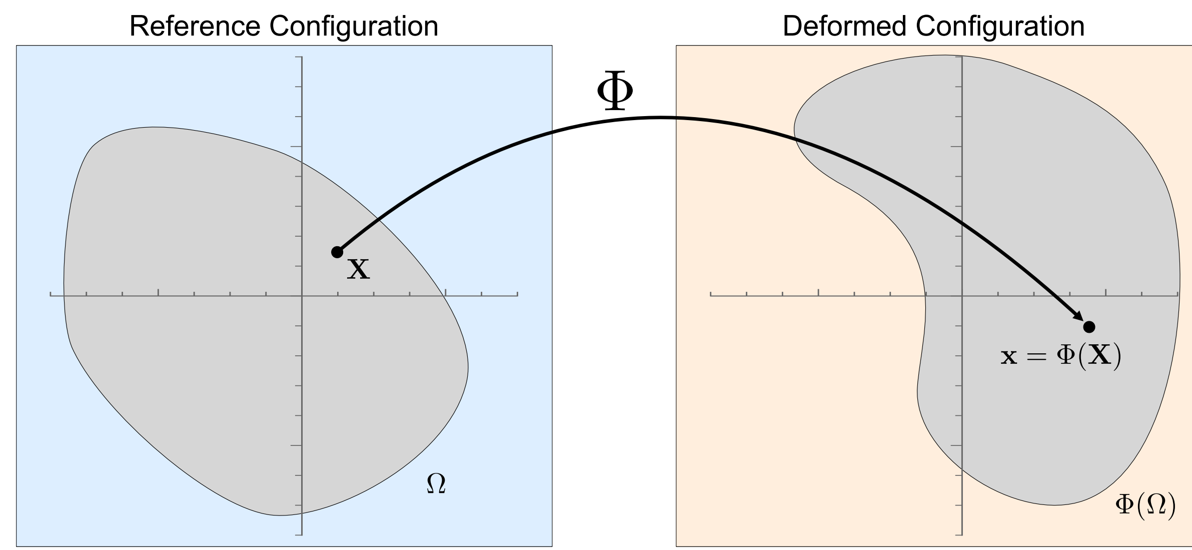
- We will look at:
  - Continuum mechanics
  - Finite element methods for tetrahedral meshes
  - Newton-style methods for energy minimization
- We will look at:
  - Shape preservation
  - Sensitivity analysis and the adjoint method for inverse shape optimization

Inverse Design Objective

- Elasticity simulation:

$$\Phi^* = \operatorname*{argmin}_{\Phi} E[\Phi]$$

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



- Our goal: Find rest shape Ω such that deformed equilibrium shape matches desired target under specified applied forces.

Inverse Design Objective

- Discretize with FEM ⇒ finite vector of variables $\mathbf{x}$ stacking node positions $\mathbf{x}_i$:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, E(\mathbf{x}) \qquad E(\mathbf{x}) := E\left[\sum_i \mathbf{x}_i \phi_i\right]$$

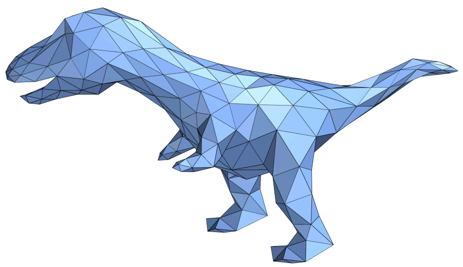
- For a discrete mesh, our goal is to find the vertex positions $\mathbf{X}_i$ of the *rest state* such that the mesh deforms into the target under the applied forces.
- We are given a desired target state and external forces acing on the shape.
  - We will use gravity and additional nodal forces, i.e. point loads.
- We will prescribe target positions for each deformed *boundary* vertex $\mathbf{x}_b$ of our mesh.
  - Some vertices will be pinned, i.e. their target position is the same as the input configuration.

Inverse Design Objective

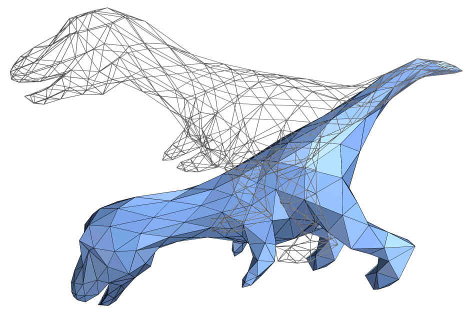
- Can be captured in a design objective function

$$J(\mathbf{x}, \mathbf{x}_b^t) := \frac{1}{2} \|\mathbf{x}_b - \mathbf{x}_b^t\|^2$$

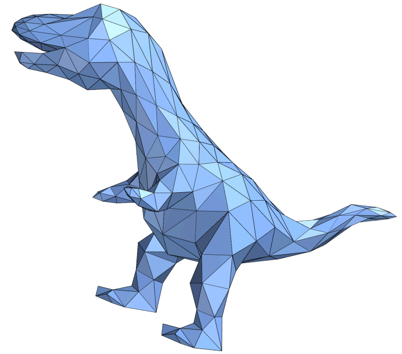
where $\mathbf{x}_b^t$ denotes the target positions of the corresponding boundary vertices.



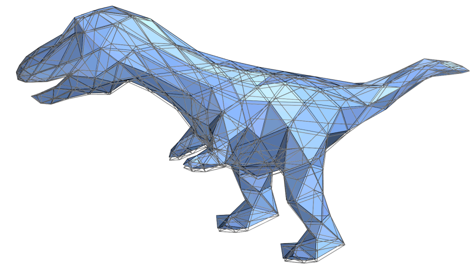
input model, rest state



deformed state under gravity

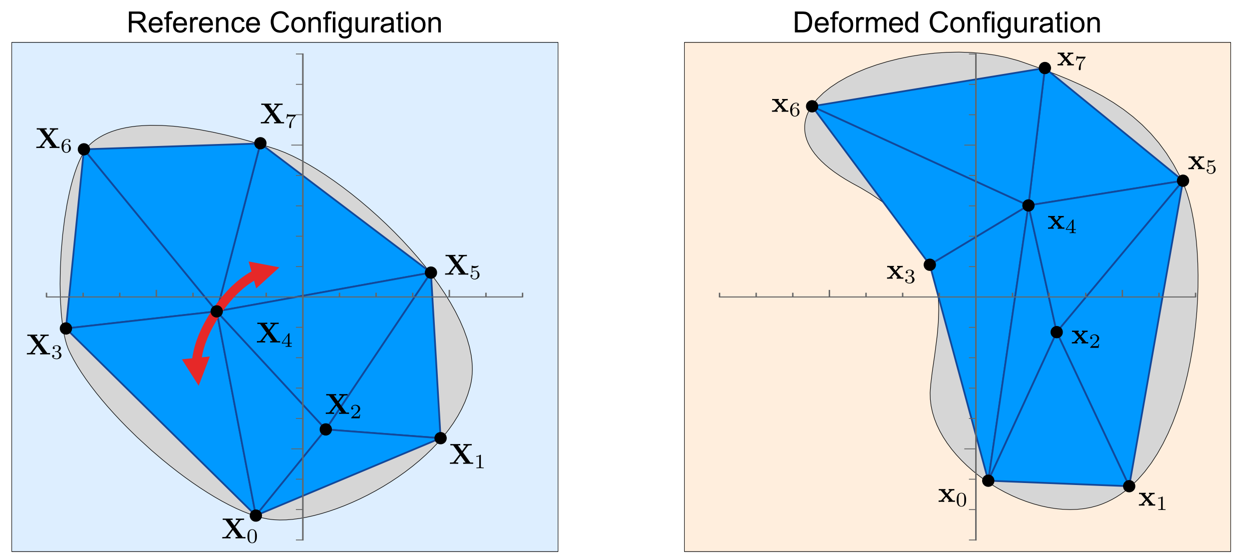


optimized model, rest state



deformed state under gravity

Optimal Design for Nonlinear Elasticity



- Suppose object is determined by *design parameters* $\mathbf{p}$.
  - In our case, $\mathbf{p}$ will be the positions $\mathbf{X}_i$ of the *boundary* vertices in the rest state. Other parameters such as stiffness or other material model parameters could also be used.
  - FEM load vector $\mathbf{f}$ can depend on $\mathbf{p}$ (e.g., if loads include self weight).
  - Discretized total potential energy function now depends on $\mathbf{p}$: $E(\mathbf{x}, \mathbf{p})$.

Optimal Design for Nonlinear Elasticity

- We want to optimize a generic performance metric $J(\mathbf{x}, \mathbf{p})$ that depends on the equilibrium state:

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

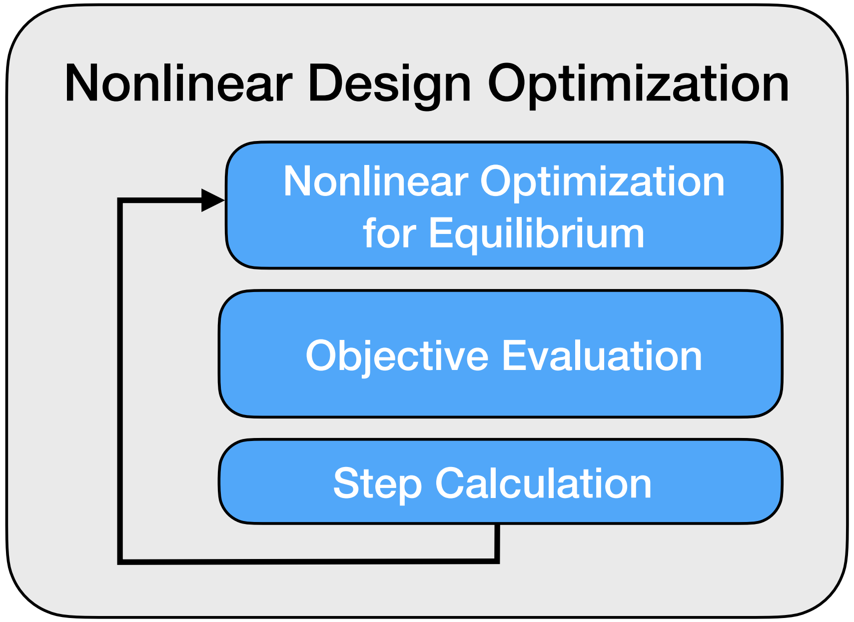
- Reduced approach: eliminate $\mathbf{x}^*$ using equilibrium constraint ⇒ *unconstrained optimization*:

$$\min_{\mathbf{p}} \underbrace{J(\mathbf{x}^*(\mathbf{p}), \mathbf{p})}_{\hat{J}(\mathbf{p})} \qquad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p})$$

Optimal Design for Nonlinear Elasticity

$$\min_{\mathbf{p}} \underbrace{J(\mathbf{x}^*(\mathbf{p}), \mathbf{p})}_{\hat{J}(\mathbf{p})} \qquad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p})$$

- Evaluating the objective requires first solving the nonlinear equilibrium problem.
- Must re-solve every time $\mathbf{p}$ updates
  $\Rightarrow$ nested optimization.



Interlude: Derivatives and Chain Rule

1 of 4

Example A.2 Taylor series expansion for two variables

Consider the following function of two variables:

$$f(x_1, x_2) = (1 - x_1)^2 + (1 - x_2)^2 + \frac{1}{2}\left(2x_2 -\right.$$

Performing a Taylor series expansion about $x = [0, -2]$, w

$$f(x + \alpha p) = 18 + \alpha \begin{bmatrix} -2 & -14 \end{bmatrix} p + \frac{1}{2}\alpha^2 p^\intercal \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Optimal Design for Nonlinear Elasticity

$$\min_{\mathbf{p}} \underbrace{J(\mathbf{x}^*(\mathbf{p}), \mathbf{p})}_{\hat{J}(\mathbf{p})} \qquad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p})$$

- Evaluating the objective requires first solving the nonlinear equilibrium problem.
- Determining a descent direction for $\hat{J}$ requires *differentiating through the nonlinear equilibrium solve*. This can be done using the chain rule:

$$\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = \frac{\partial J}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial J}{\partial p_i}$$

- Direct dependence on design parameters $\frac{\partial J}{\partial p_i}$ is generally easy to derive and evaluate. However, $\frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i}$ is more complicated.

Sensitivity Analysis

$$\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = \frac{\partial J}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial J}{\partial p_i}, \qquad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p})$$

- What nonlinear equation implicitly defines equilibrium function $\mathbf{x}^*(\mathbf{p})$?
- Optimality conditions:

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

- Obtain an equation for $\frac{\partial \mathbf{x}^*}{\partial p_i}$ by differentiating w.r.t. p_i using chain rule:

$$\frac{\mathrm{d}}{\mathrm{d}p_i} \frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}), \mathbf{p}) = \frac{\partial^2 E}{\partial \mathbf{x}^2} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i} = 0$$

$$\implies \quad \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i}.$$

References: Tutorial, Short Overview Article, Wikipedia

Sensitivity Analysis

$$\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = \frac{\partial J}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial J}{\partial p_i}, \qquad \mathbf{x}^*(\mathbf{p}) := \operatorname*{argmin}_{\mathbf{x}} E(\mathbf{x}, \mathbf{p})$$

- Obtain an equation for $\frac{\partial \mathbf{x}^*}{\partial p_i}$ by differentiating w.r.t. p_i using chain rule:

$$\frac{\mathrm{d}}{\mathrm{d}p_i} \frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}^*(\mathbf{p}), \mathbf{p}) = \frac{\partial^2 E}{\partial \mathbf{x}^2} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i} = 0$$

$$\implies \quad \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} = -\left[\frac{\partial^2 E}{\partial \mathbf{x}^2}\right]^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i}.$$

- Note $\frac{\partial^2 E}{\partial \mathbf{x}^2} = H$ is same Hessian matrix used to solve for the Newton step $H\mathbf{d} = -\frac{\partial E}{\partial \mathbf{x}}$.
  - Guaranteed to be positive definite at stable equilibrium ⇒ unique solution $\frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i}$ exists.
  - Backsolving for each p_i will be expensive, though. How can we avoid this?

Sensitivity Analysis

$$\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = \frac{\partial J}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} + \frac{\partial J}{\partial p_i}, \qquad \frac{\mathrm{d}\mathbf{x}^*}{\mathrm{d}p_i} = -H^{-1} \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i}$$

- $\frac{\partial J}{\partial \mathbf{x}}$ is a row vector, and thus so is $\frac{\partial J}{\partial \mathbf{x}} H^{-1}$.
- Define “adjoint state” $\mathbf{y}$ to be the vector such that

$$\mathbf{y}^\top = \frac{\partial J}{\partial \mathbf{x}} H^{-1} \quad \Longrightarrow \quad \mathbf{y} = H^{-1\top} \left(\frac{\partial J}{\partial \mathbf{x}} \right)^\top$$

- In other words, $\mathbf{y}$ is the solution to the *adjoint equation*:

$$H^\top \mathbf{y} = \left(\frac{\partial J}{\partial \mathbf{x}} \right)^\top$$

Sensitivity Analysis

- Adjoint Equation

$$H^\top \mathbf{y} = \left(\frac{\partial J}{\partial \mathbf{x}}\right)^\top$$

  - For elasticity (and many other PDEs), $H = \frac{\partial^2 E}{\partial \mathbf{x}^2}$ is symmetric and thus the system matrix of the adjoint equation is the same as the system matrix of the original state equation (equilibrium problem).

- After we solve this *single* equation, we can compute each gradient component with just a dot product rather than a backsolve:

$$\boxed{\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = -\mathbf{y}^\top \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i} + \frac{\partial J}{\partial p_i}.}$$

Wikipedia: Adjoint State Method

Inverse Design Optimization

- Adjoint method allows us to efficiently calculate gradients for minimizing a function depending on the result of a simulation.

$$\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = -\mathbf{y}^\top \frac{\partial^2 E}{\partial \mathbf{x} \partial p_i} + \frac{\partial J}{\partial p_i}$$

- Our target fitting objective $J(\mathbf{x}, \mathbf{x}_b^t) := \frac{1}{2}\|\mathbf{x}_b - \mathbf{x}_b^t\|^2$ means that $\frac{\partial J}{\partial p_i} = 0$.
- Remaining Questions:
  - How do we compute $\frac{\partial^2 E}{\partial \mathbf{x} \partial p_i}$?
  - How do we initialize the design optimization?

Automatic Differentiation

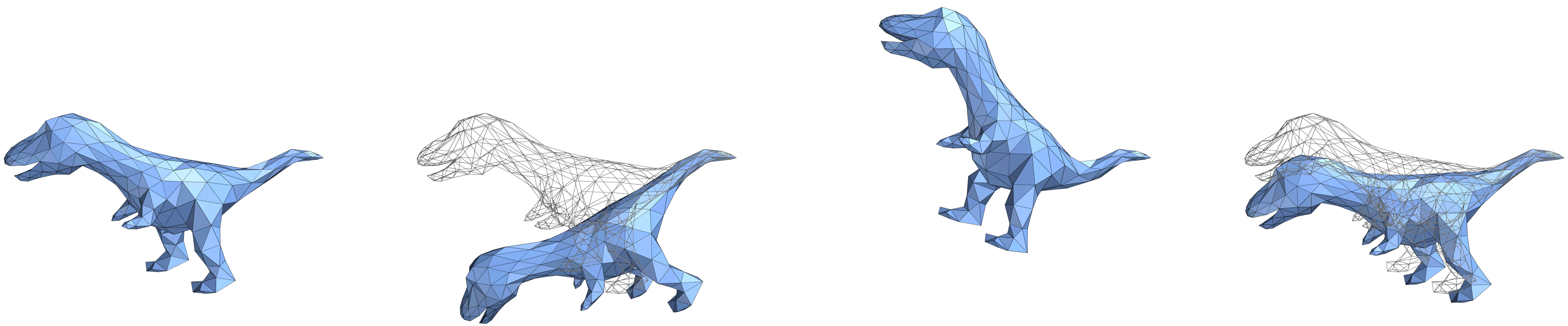
- How do we compute $\frac{\partial^2 E}{\partial \mathbf{x} \partial \mathbf{p}}$?
- Depends on p_i. In the homework we computed derivatives of a function evaluating the dot product of $\frac{\partial E}{\partial \mathbf{x}}$ with "constant" $\hat{\mathbf{y}}$.

$$\left(\frac{\mathrm{d}\hat{J}}{\mathrm{d}\mathbf{p}}\right)^T = -\frac{\partial^2 E}{\partial \mathbf{p} \partial \mathbf{x}} \hat{\mathbf{y}} = \frac{\partial}{\partial \mathbf{p}} \left(-\frac{\partial E}{\partial \mathbf{x}} \cdot \hat{\mathbf{y}}\right) = \frac{\partial}{\partial \mathbf{p}} (\mathbf{f} \cdot \hat{\mathbf{y}})$$

- Can be efficiently evaluated using automatic differentiation.
  - For example using PyTorch or Jax.

Initialization of Design Optimization

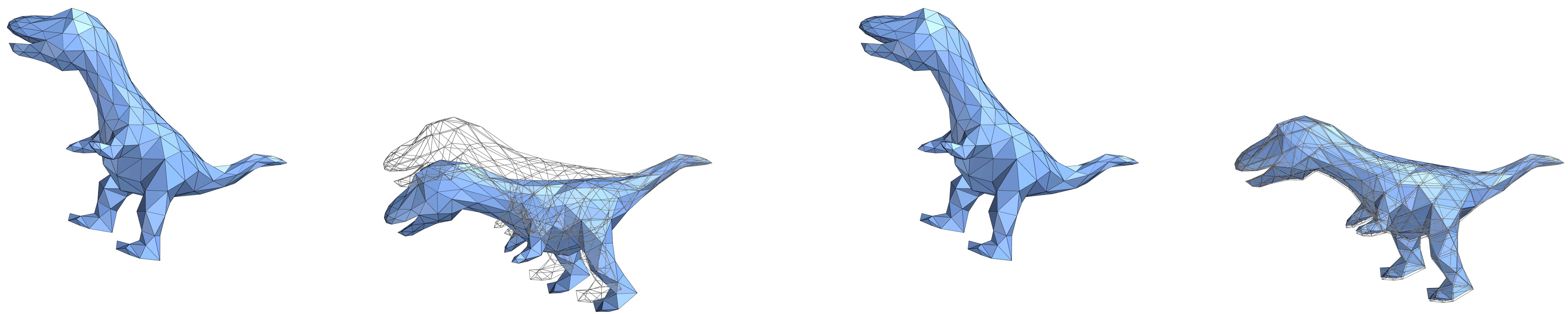
- Starting the inverse optimization of the rest state from initial design is far from optimum.
- Can we find a better initial state?



input model, rest state | *deformed state under gravity* | *input model, inverse gravity* | *deformed state under gravity*

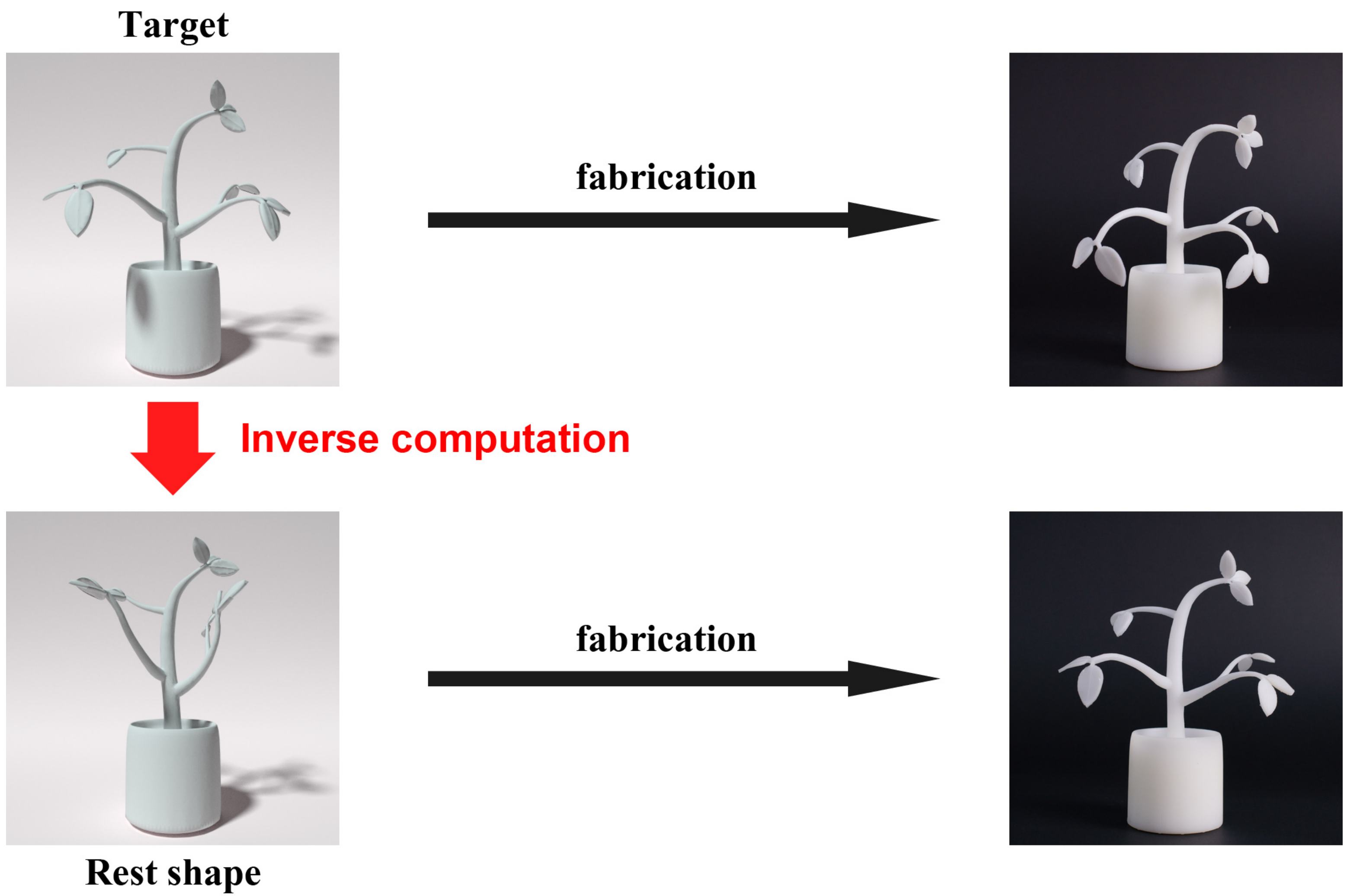
Initialization of Design Optimization

- Use equilibrium state under inverse gravity as starting configuration of rest state optimization!



input model, inverse gravity | *deformed state under gravity* | *optimized model, rest state* | *deformed state under gravity*

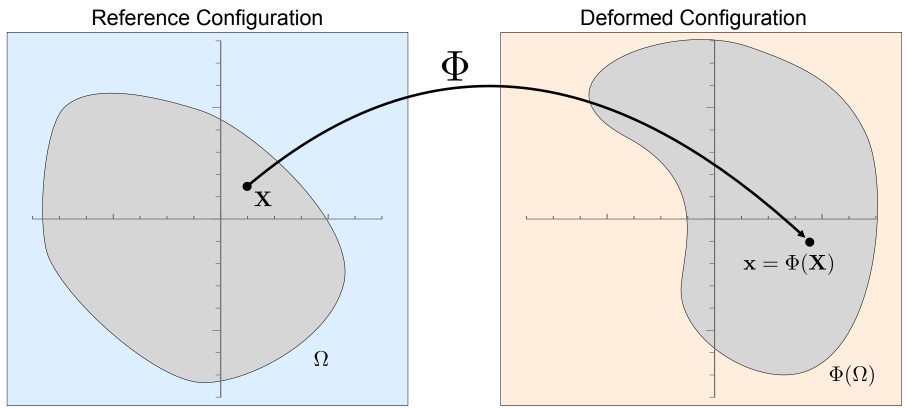
Challenge II: Inverse Elastic Shape Optimization

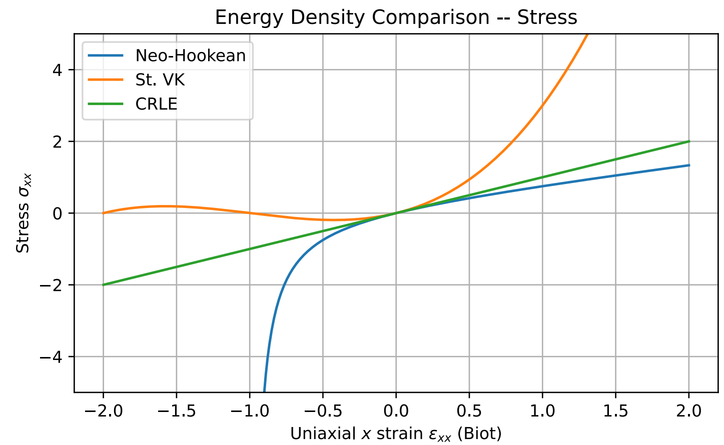


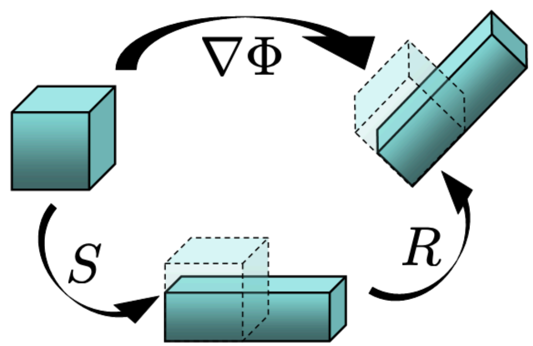
An Asymptotic Numerical Method for Inverse Elastic Shape Design

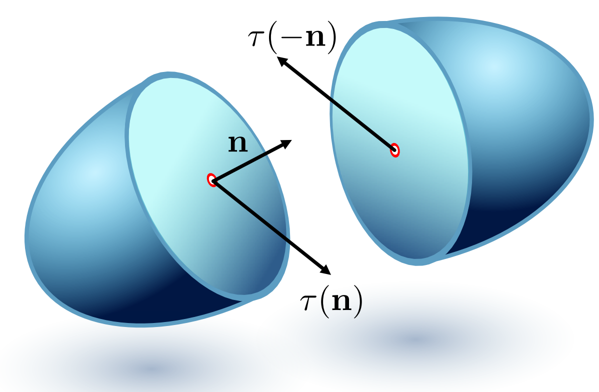
Challenge II: Inverse Elastic Shape Optimization

- How can we simulate elastic objects?
  - How can we *model elastic deformation*?
- We looked at:
  - Continuum mechanics





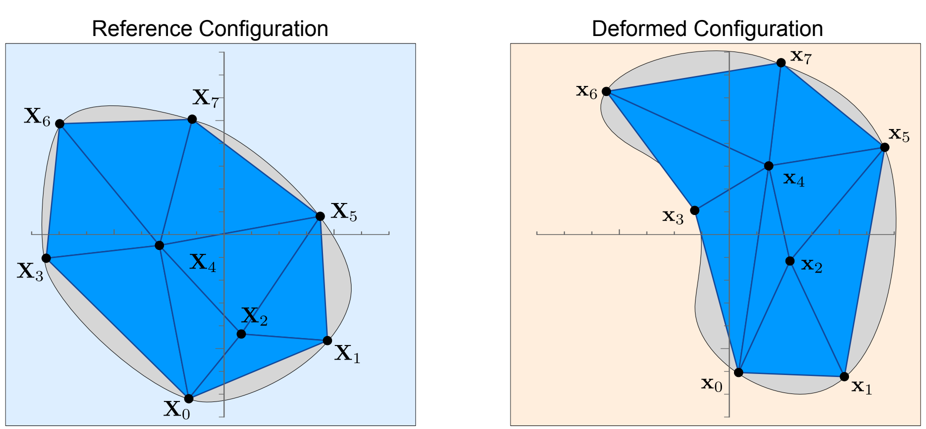


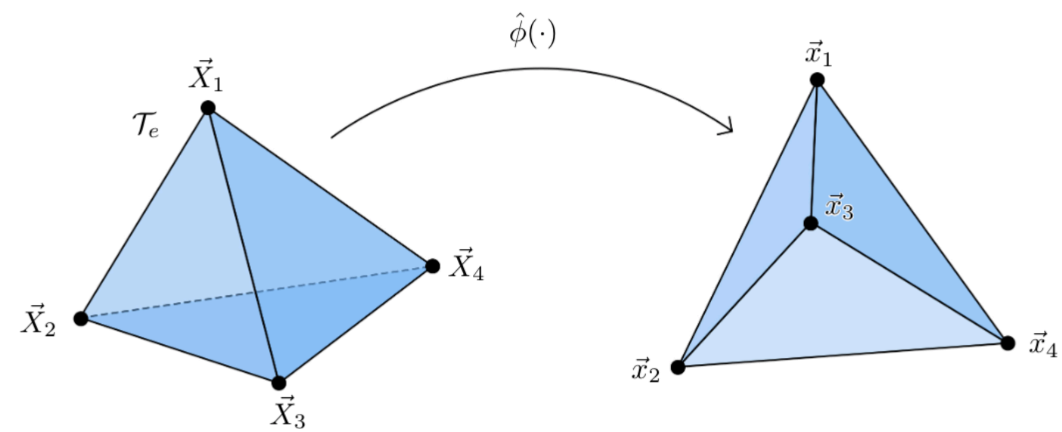


Challenge II: Inverse Elastic Shape Optimization

- How can we simulate elastic objects?
  - How can we *model elastic deformation*?
  - How can we *discretize volumetric solids*?
- We looked at:
  - Continuum mechanics
  - Finite element methods for tetrahedral meshes

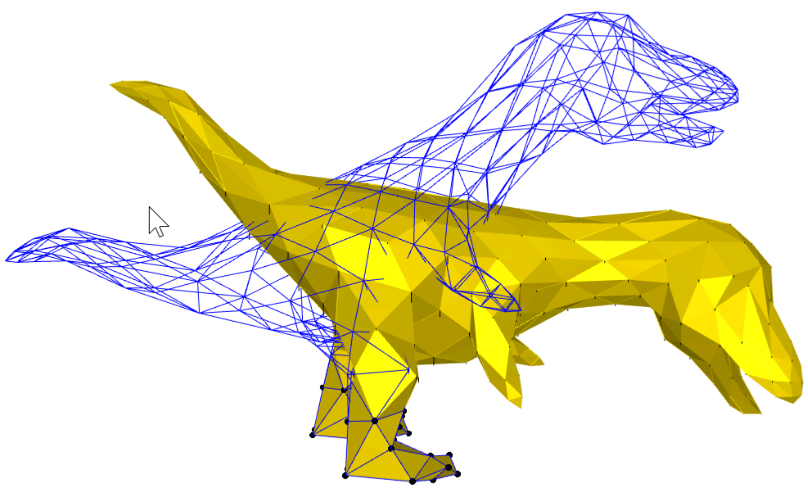
$$E = \int_\Omega \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X} = \sum_e \Psi\left(\nabla\Phi\big|_e\right)\mathtt{Vol}(e)$$

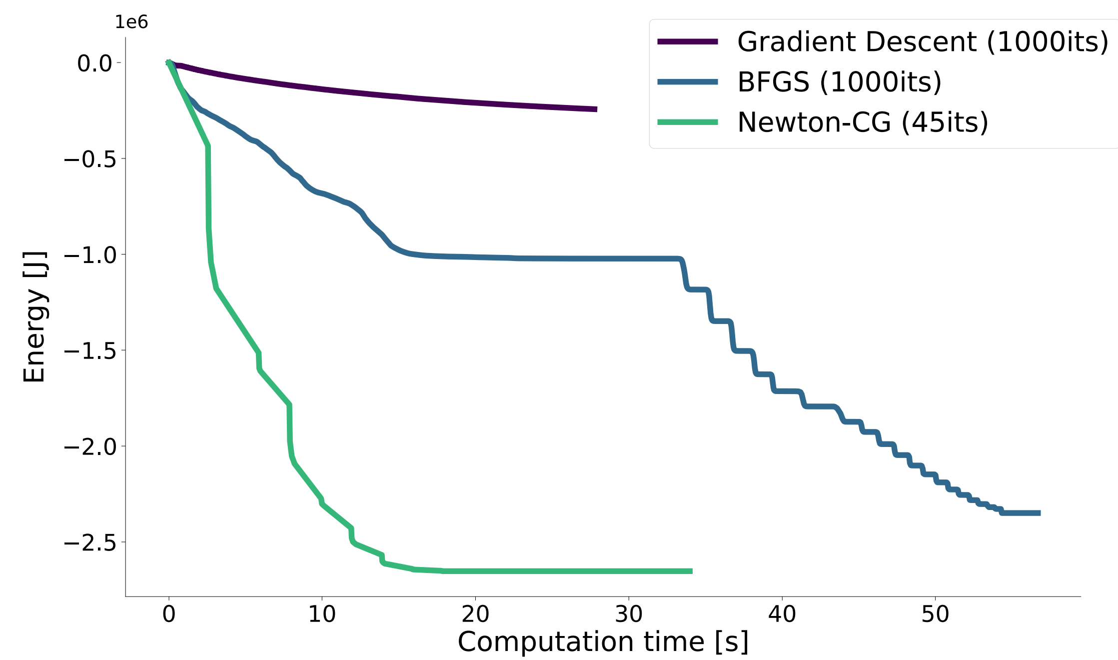




Challenge II: Inverse Elastic Shape Optimization

- How can we simulate elastic objects?
  - How can we *model elastic deformation*?
  - How can we *discretize volumetric solids*?
  - How can we *find equilibrium states*?
- We looked at:
  - Continuum mechanics
  - Finite element methods for tetrahedral meshes
  - Newton-style methods for energy minimization





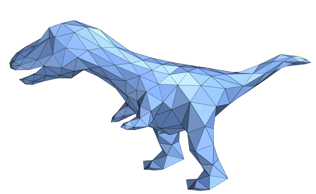
Challenge II: Inverse Elastic Shape Optimization

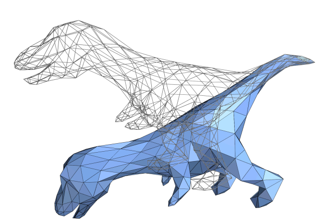
- How can we simulate elastic objects?
  - How can we *model elastic deformation*?
  - How can we *discretize volumetric solids*?
  - How can we *find equilibrium states*?
- How can we optimize elastic objects?
  - How can we *find the best modification*?

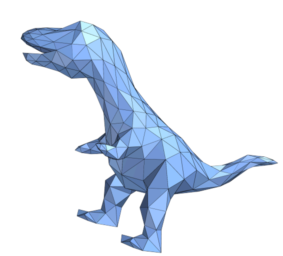
- We looked at:
  - Continuum mechanics
  - Finite element methods for tetrahedral meshes
  - Newton-style methods for energy minimization
- We looked at:
  - Sensitivity analysis and the adjoint method

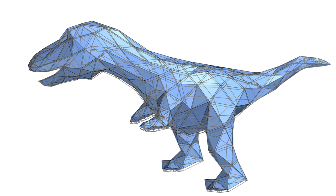
$$H^\top \mathbf{y} = \left(\frac{\partial J}{\partial \mathbf{x}}\right)^\top$$

$$\boxed{\frac{\mathrm{d}\hat{J}}{\mathrm{d}p_i} = -\mathbf{y}^\top \frac{\partial^2 E}{\partial \mathbf{x}\partial p_i} + \frac{\partial J}{\partial p_i}.}$$

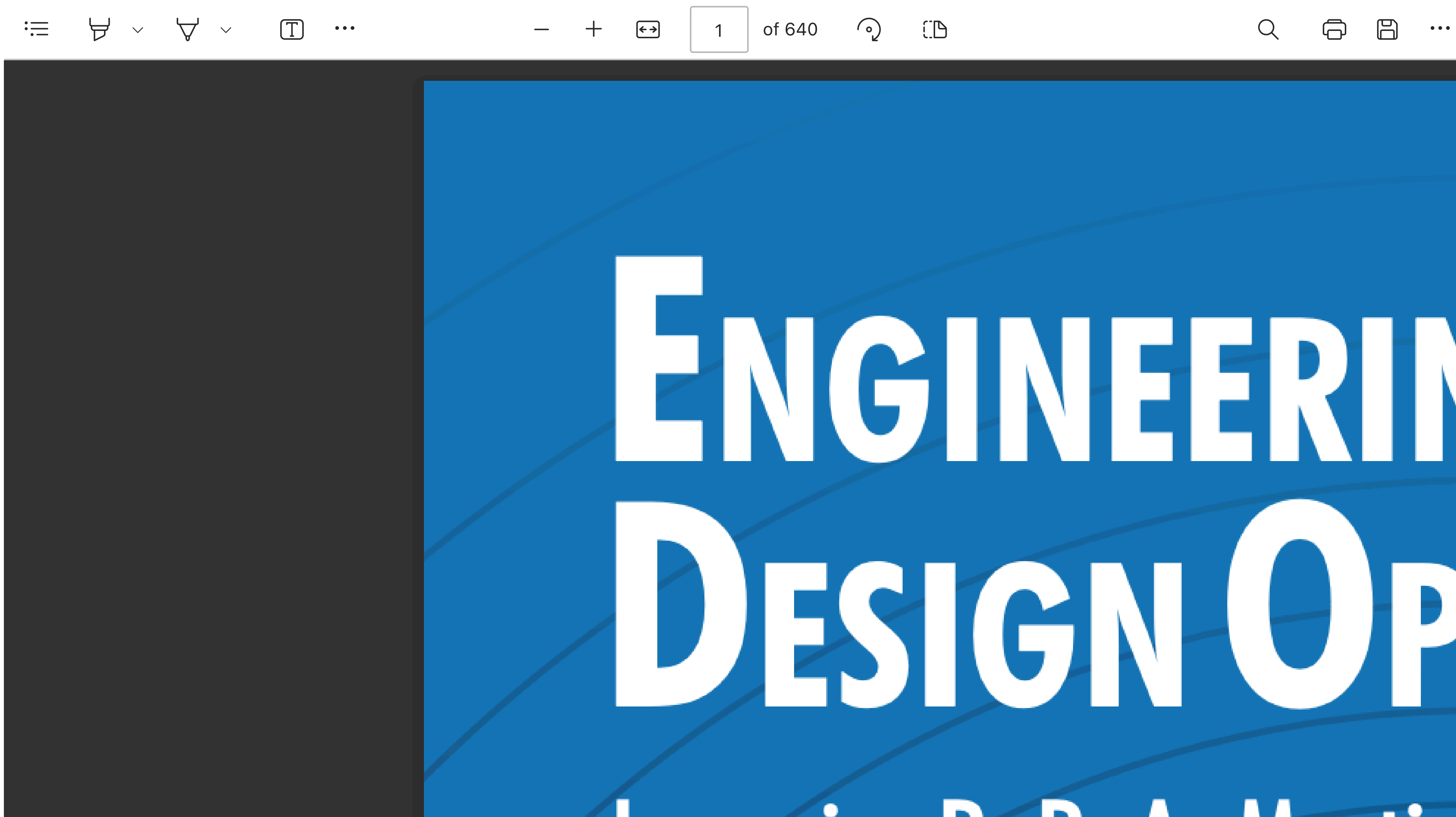

input model, rest state


deformed state under gravity


optimized model, rest state


deformed state under gravity

Reading



Engineering Design Optimization. Chapter 6.7

Outlook: Challenge III: Asymptotic Gridshell Example

Eike Schling

Geometry.Design.Structure

Asymptotic Envelope Substructure

By eikeschling / December 7, 2020 / Academia, Architecture, Research / Leave a comment
Type: BuildingEnvelope
Location: Hong Kong, Taipei
Year: 2020
Status: Completion Substructure
Office: The University of Hong Kong
In collaboration with: National Taiwan University of Science and Technology, Shen Guan Shih
Industry Partner: Gomore Building Envelope Technology, Sam Hsu

Project Team: Eike Schling, Jacky Chu, Muye Ma, Wesley She, Fai Lam Chung, Nuozi Chen
Lee Chun Ki, Yao Dongni, Choi Chung Hei, Chung Bing Tsun, Ma Chun Hon, Ng Sherene Poh Li, So Cheuk Lam, Wang Xiangning, Zhu Xiang, Yang Mei, Chan Ching Yee

Together with the elective course 'Structural Research' at HKU we constructed a steel prototype for a doubly curved curtain wall module this

Source: Eike Schling

Outlook: Challenge III: Asymptotic Gridshell Example

Eike Schling

Geometry.Design.Structure

Canopy for the Hotel Intergroup in Ingolstadt

By eikeschling / December 3, 2019 / Architecture, Research / Leave a comment
Type: Canopy
Location: Ingolstadt, Germany
Year: 2019
Status: completed
Planning Team: Eike Schling, Jonas Schikore
Partner: Brandl Metallbau, Eitensheim (https://www.brandl-eitensheim.de/metallbau/)

The Asymptotic Canopy draws attention to the main entrance to the Intergroup Business and Design Hotel, Ingolstadt. The stainless steel structure is composed of four symmetrical pods of strained lamella gridshells spanning between rigid steel frames. The structure creates a central ornamental portal and two side wings within the semi courtyard in front of the main entrance. The steel structure is covered with a membrane roof, which is tensioned by an elastic steel arch.

The gridshell was designed digitally, following the asymptotic curves on the negatively curved design surface. This allowed for the simple

Source: Eike Schling

Monday!

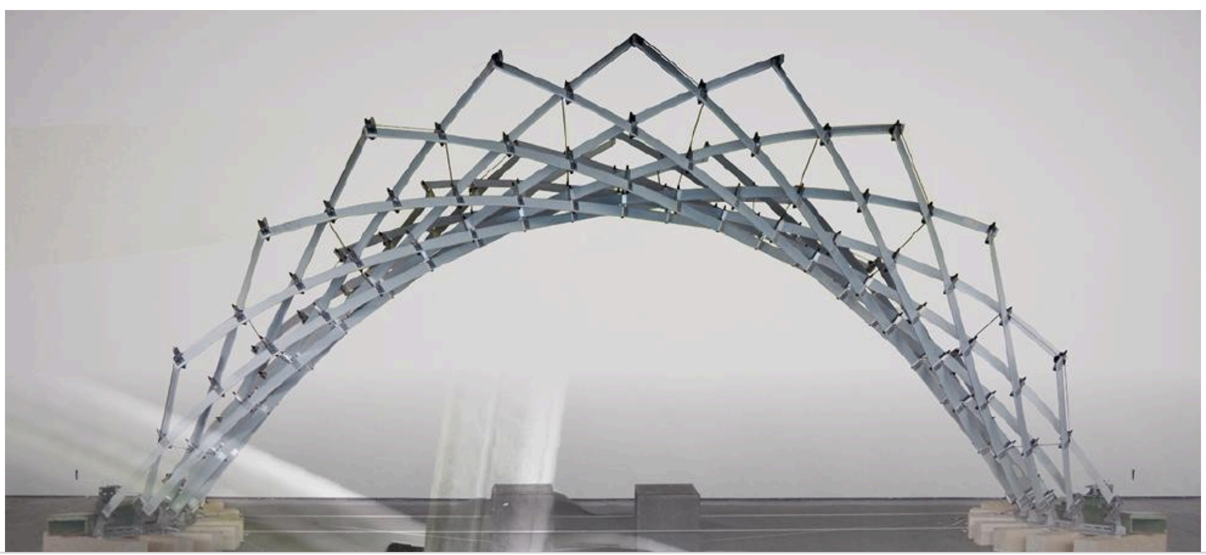
EPFL

MEMENTO

FR EN Menu

/ Memento

Kinetic Grid Structures



Event details

Date	11.11.2024
Hour	12:00 › 13:00
Speaker	Eike Schling
Location	GC B1 10 Online
Category	Conferences - Seminars
Event Language	English

https://memento.epfl.ch/event/kinetic-grid-structures/