CS457 Geometric Computing

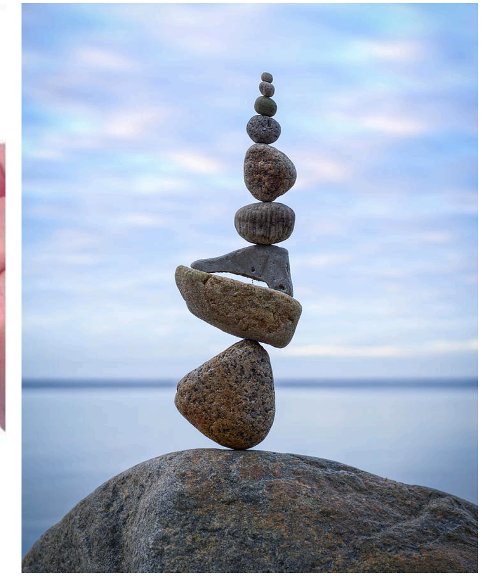
5A - Elastic Shape Optimization

Mark Pauly

Geometric Computing Laboratory - EPFL

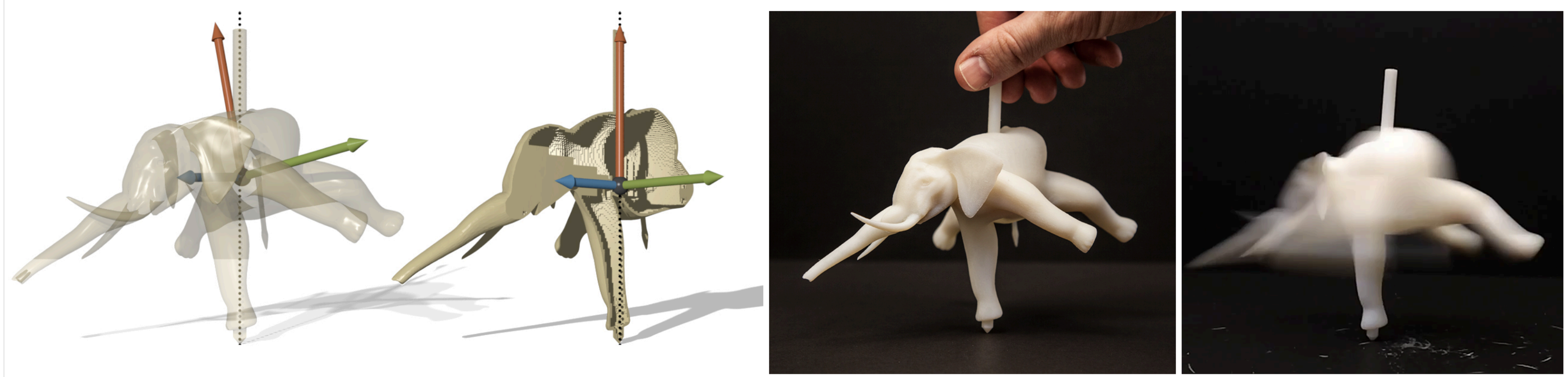
Challenge I: Make it stand

- Given some (digital) geometric object, how can we determine if it stands?
- If it does not stand, how can we optimize its shape, so that it does?



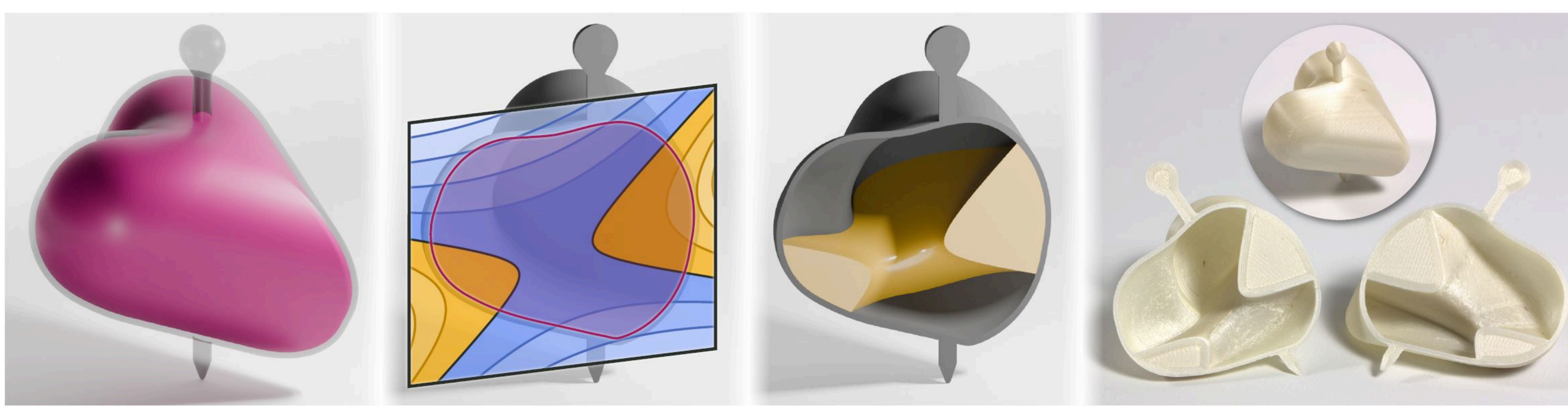
Make It Stand: Balancing Shapes for 3D Fabrication, ACM SIGGRAPH 2013

Make it spin



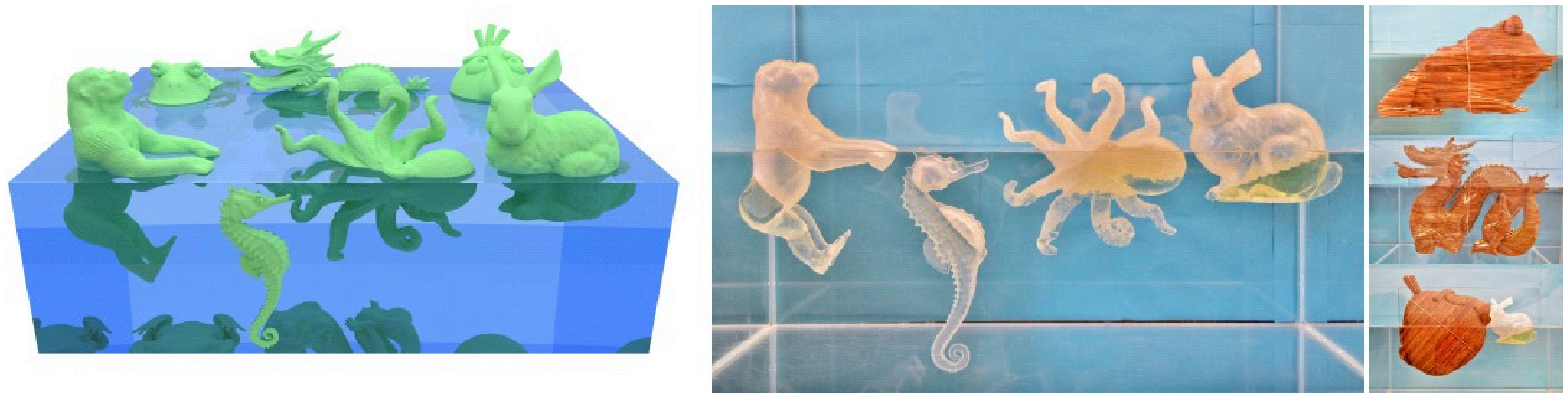
(a) unstable input (b) hollowed, optimized model (c) our spinning top design (d) elephant in motion

Spin-It: Optimizing Moment of Inertia for Spinnable Objects, ACM SIGGRAPH 2014



Spin-It Faster: Quadrics Solve All Topology Optimization Problems That Depend Only On Mass Moments, ACM SIGGRAPH 2024

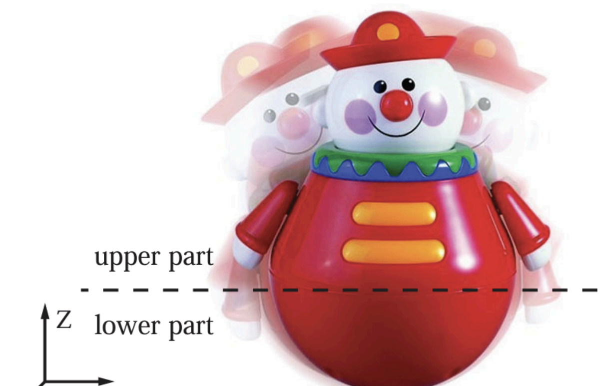
Make it swim



Buoyancy Optimization for Computational Fabrication, Eurographics 2016

Make it swing

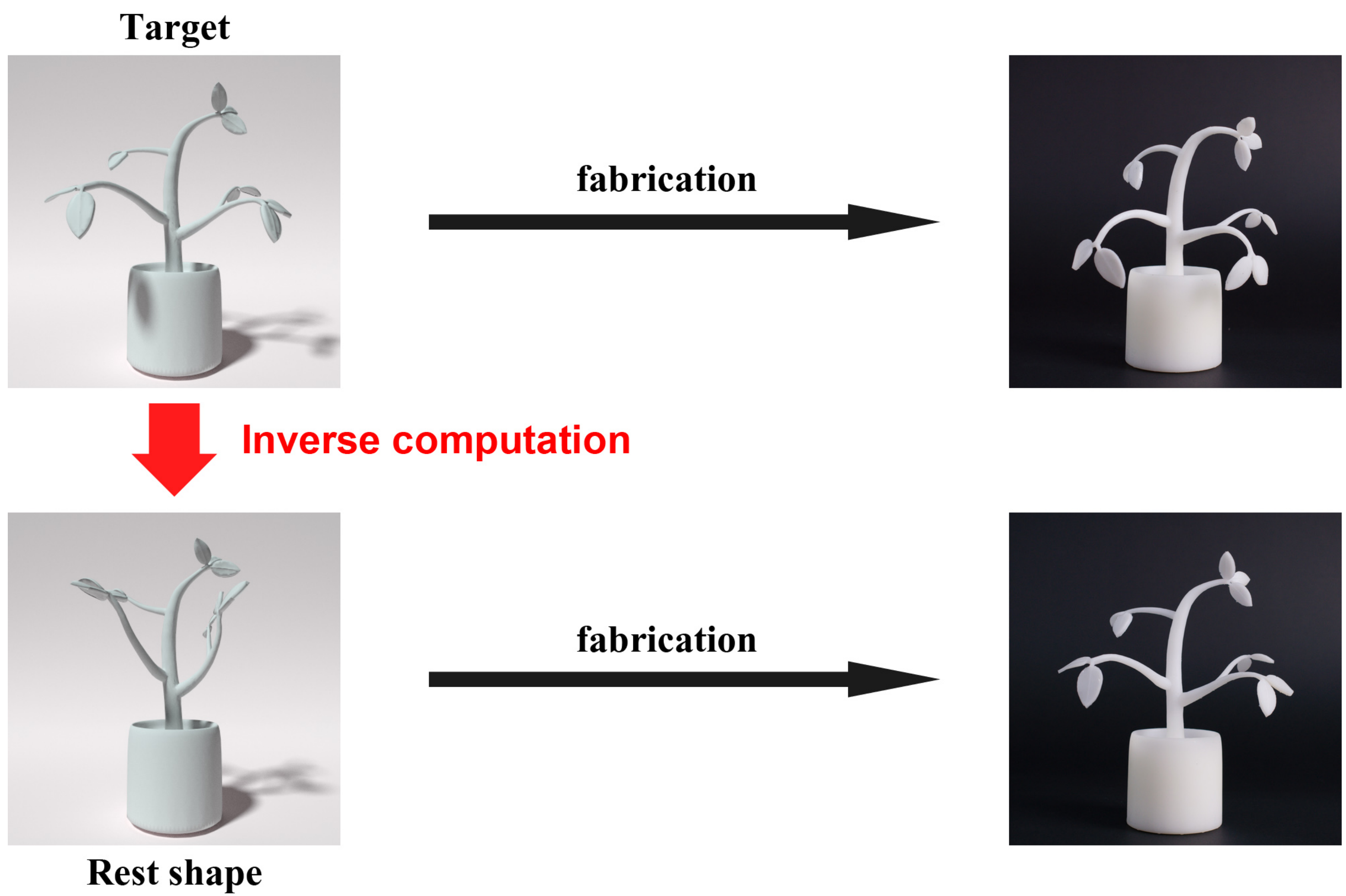
Figure 1: A goblet shape roly-poly toy example designed by our method. The snapshots show that the toy is able to regain balance when pushed over.



(a)

Make it swing: Fabricating personalized roly-poly toys, Computer Aided Geometric Design 2016

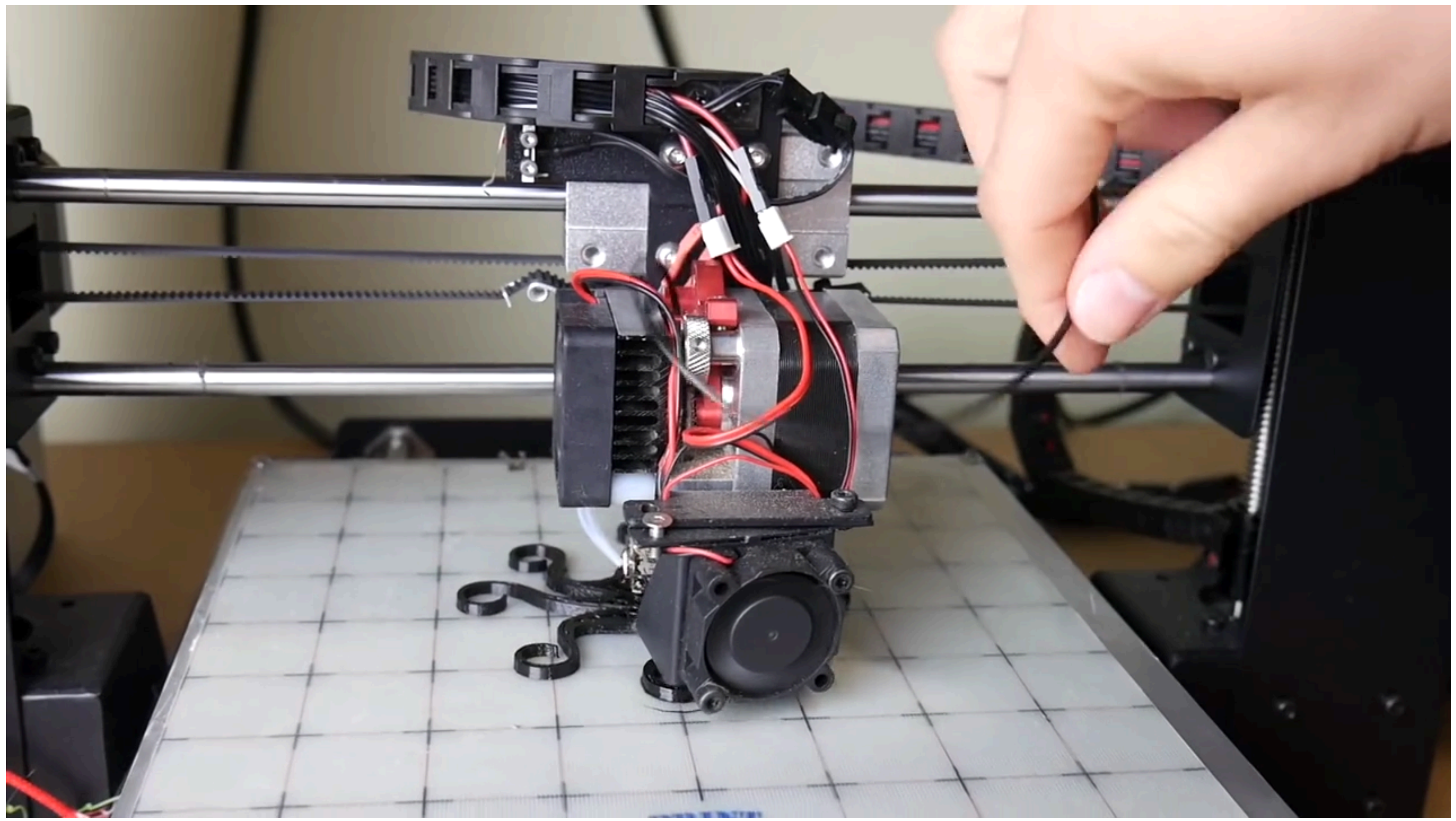
Challenge II: Inverse Elastic Shape Optimization



An Asymptotic Numerical Method for Inverse Elastic Shape Design

Motivation

- **3D printing** or casting can create very flexible objects!



Maker's Muse - Youtube

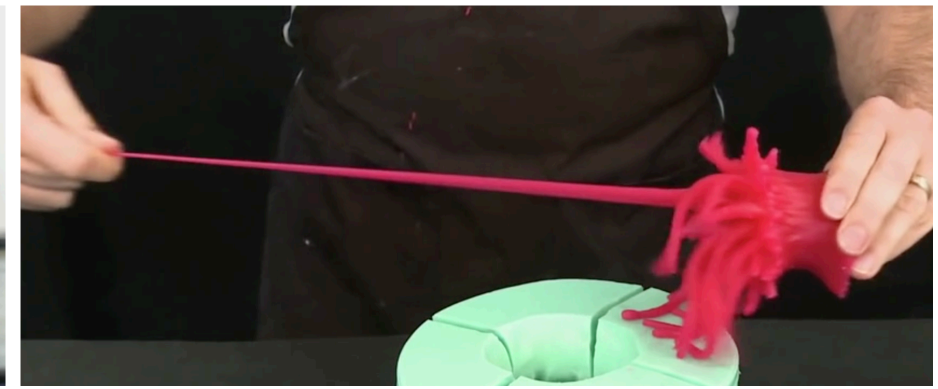
Motivation

- 3D printing or **casting** can create very flexible objects!

Smooth-On - Youtube

Motivation

- 3D printing or casting can create very flexible objects!

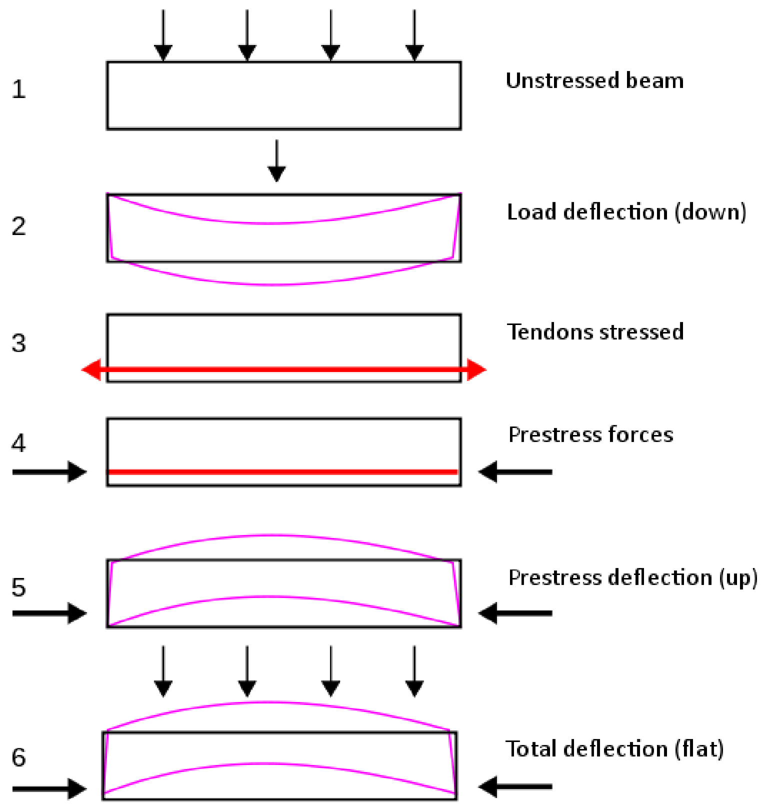
- Such objects (but really, any objects) deform under external forces (e.g gravity).
- How can we design objects that take this deformation into account?
  - How can we modify designs such that under given external forces it deforms into the desired target state?

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

Aside: Prestressed Concrete



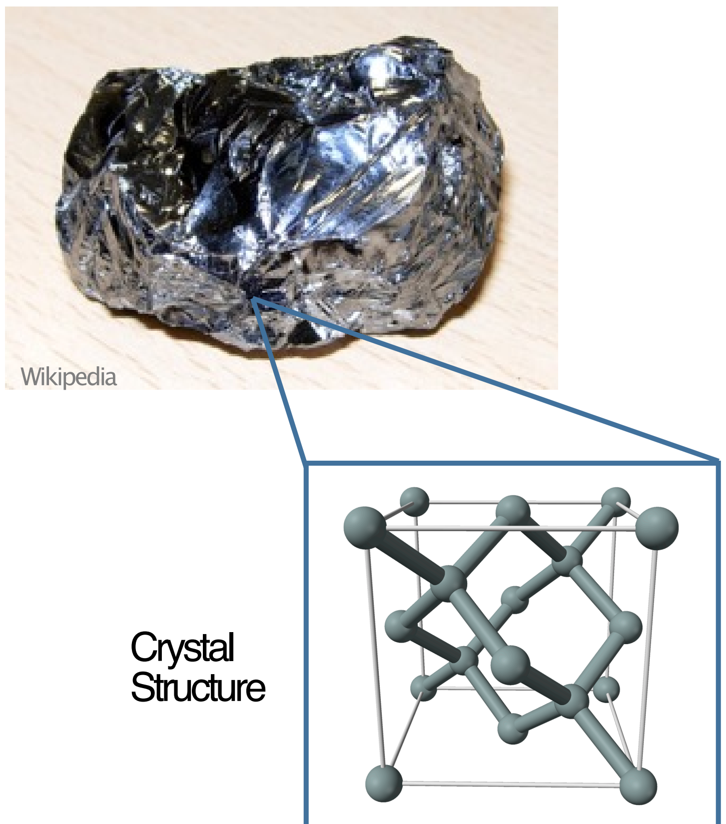
Wikipedia

Source

Origins of Elasticity

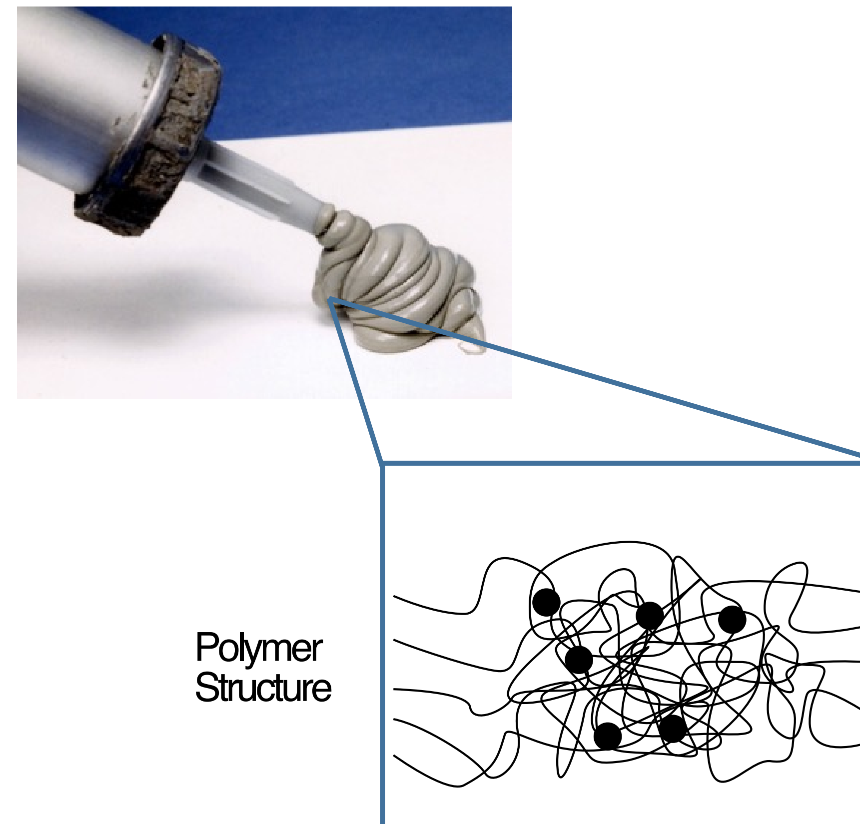
- All solids are elastic to some degree (depending on internal structure).

Silicon Si



Crystal Structure

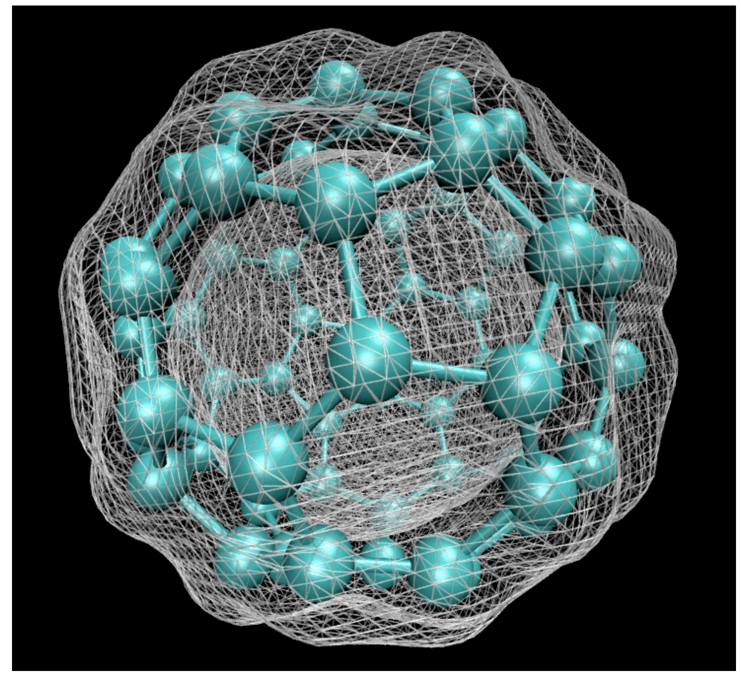
Silicone $[Si(CH_3)_2O]_n$



Polymer Structure

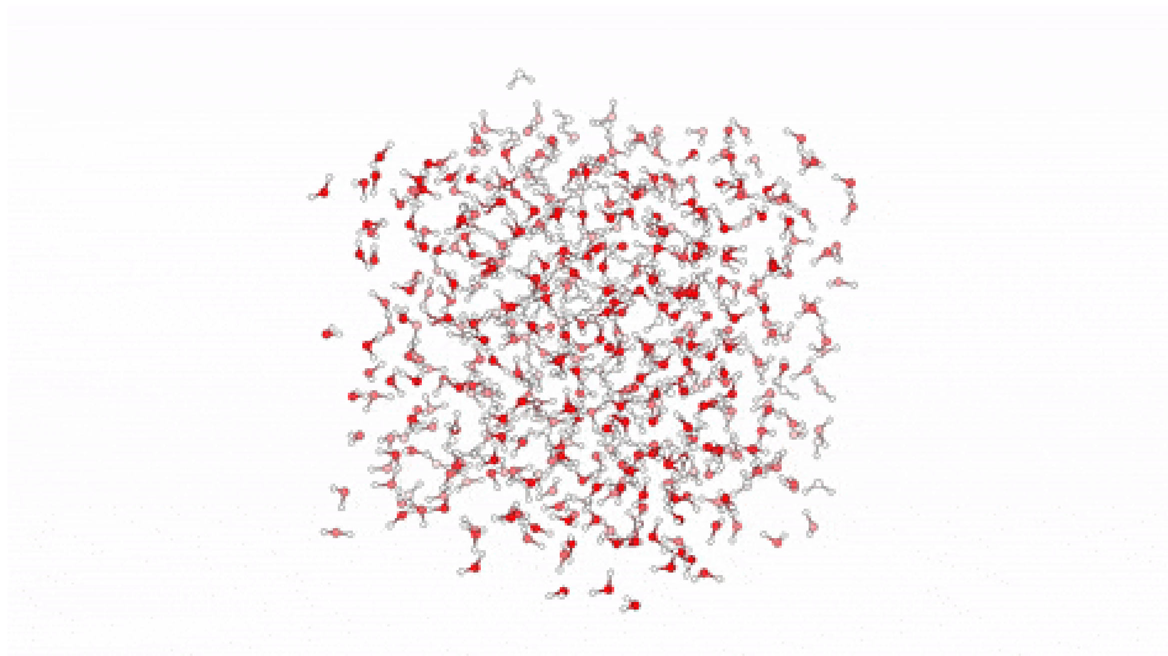
What Level of Simulation?

- Å-scale: Quantum Simulations (first principles: Density-Functional Theory, …)



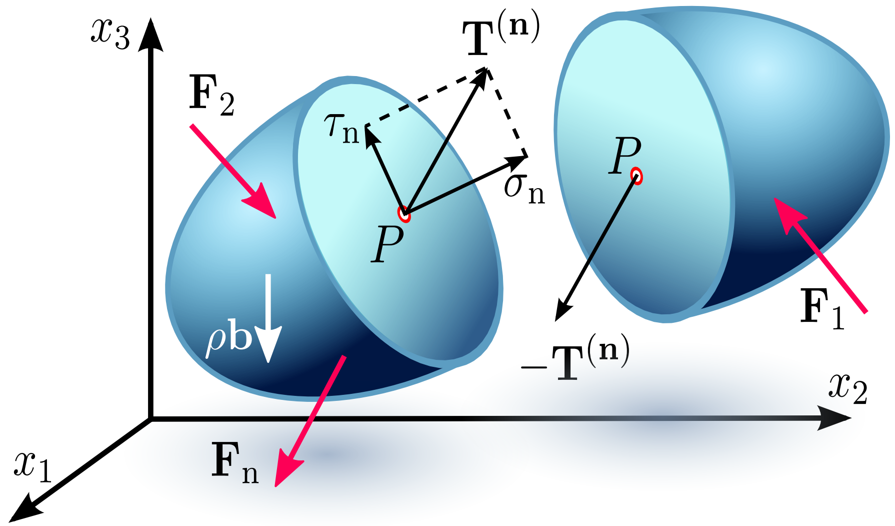
What Level of Simulation?

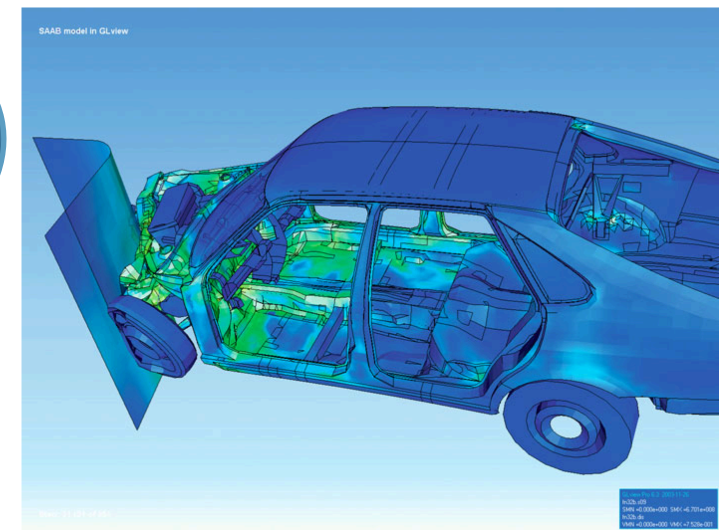
- Å-scale: Quantum Simulations (first principles: Density-Functional Theory, ...)
- nm-scale: Molecular Dynamics (empirical interatomic potentials)



What Level of Simulation?

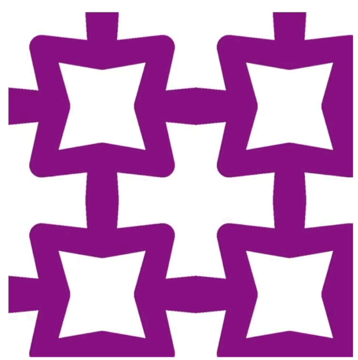
- Å-scale: Quantum Simulations (first principles: Density-Functional Theory, ...)
- nm-scale: Molecular Dynamics (empirical interatomic potentials)
- mm-scale+: Continuum Mechanics Models (partial differential equations)

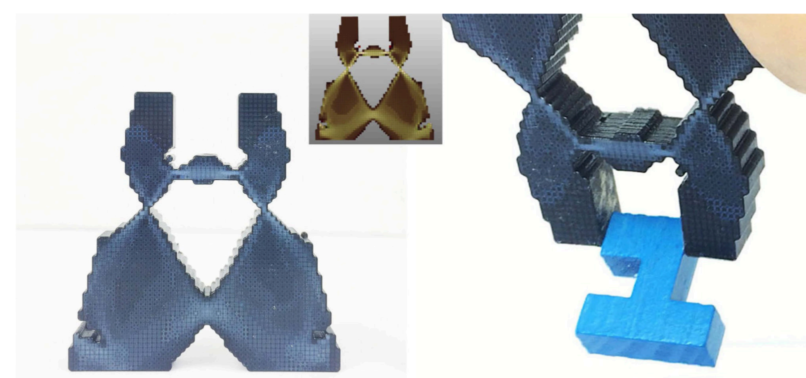




What Level of Simulation?

- Å-scale: Quantum Simulations (first principles: Density-Functional Theory, ...)
- nm-scale: Molecular Dynamics (empirical interatomic potentials)
- mm-scale+: Continuum Mechanics Models (partial differential equations)
- large scales: Homogenized Elasticity Models



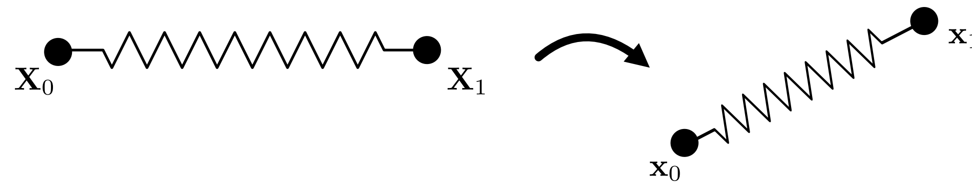


What Level of Simulation?

- Å-scale: Quantum Simulations (first principles: Density-Functional Theory, ...)
- nm-scale: Molecular Dynamics (empirical interatomic potentials)
- ***mm-scale+: Continuum Mechanics Models (partial differential equations)***
- large scales: Homogenized Elasticity Models

We will focus on continuum mechanics models.

Linear Springs (Hooke's Law) in nD



- Express a spring's energy in terms of undeformed/deformed points

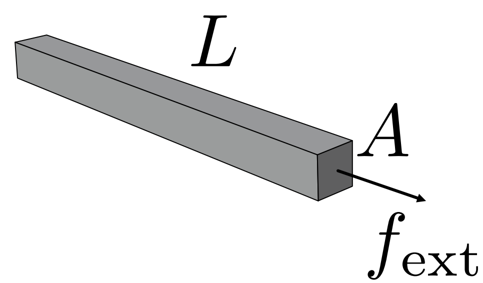
$$E_{\text{spring}} = \frac{1}{2}k\Big(\|\mathbf{x}_1 - \mathbf{x}_0\| - \|\mathbf{X}_1 - \mathbf{X}_0\|\Big)^2$$

- Force on point $\mathbf{x}_1$ applied by spring points along spring axis

$$-\frac{\partial E_{\text{spring}}}{\partial \mathbf{x}_1} = k\Big(\|\mathbf{x}_1 - \mathbf{x}_0\| - \|\mathbf{X}_1 - \mathbf{X}_0\|\Big)\frac{\mathbf{x}_0 - \mathbf{x}_1}{\|\mathbf{x}_0 - \mathbf{x}_1\|}$$

Material Properties

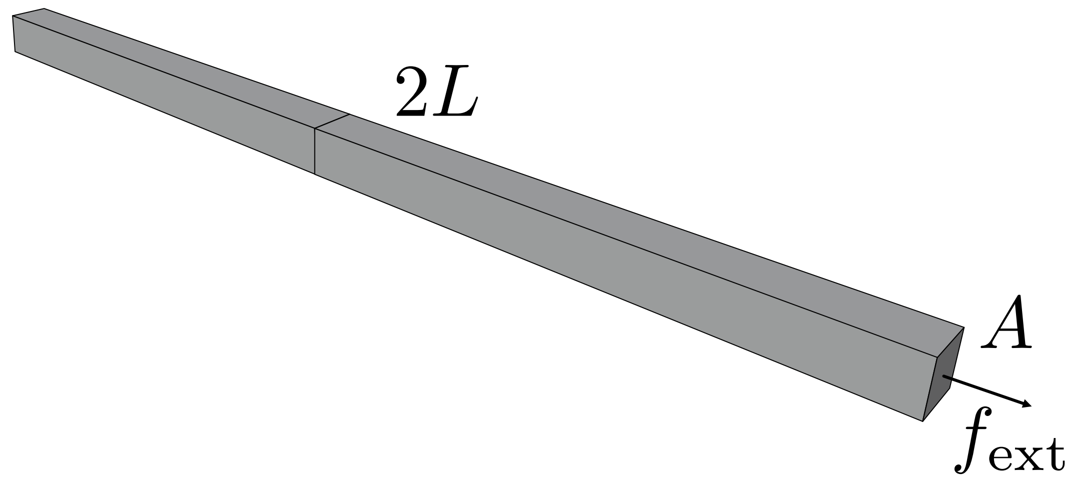
- Consider a rectangular elastic rod with rest length L, area A and spring constant k:



- According to Hooke's law, it changes length by $\Delta L = \frac{f_{\text{ext}}}{k}$

Material Properties

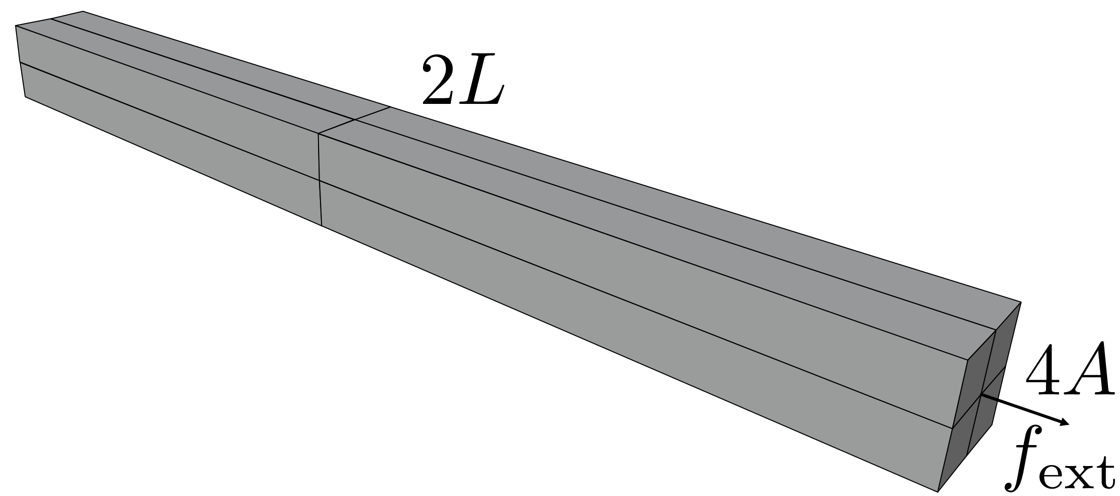
- What happens when we glue it to an identical copy?



- Both rods change length by $\Delta L \Longrightarrow$ total length change is $2\Delta L$.
- Spring constant changes to $\tilde{k} = \frac{f_{\text{ext}}}{2\Delta L} = \frac{1}{2}\frac{f_{\text{ext}}}{\Delta L} = \frac{1}{2}k$.

Material Properties

- Now let's glue together 4 copies of the lengthened rod:

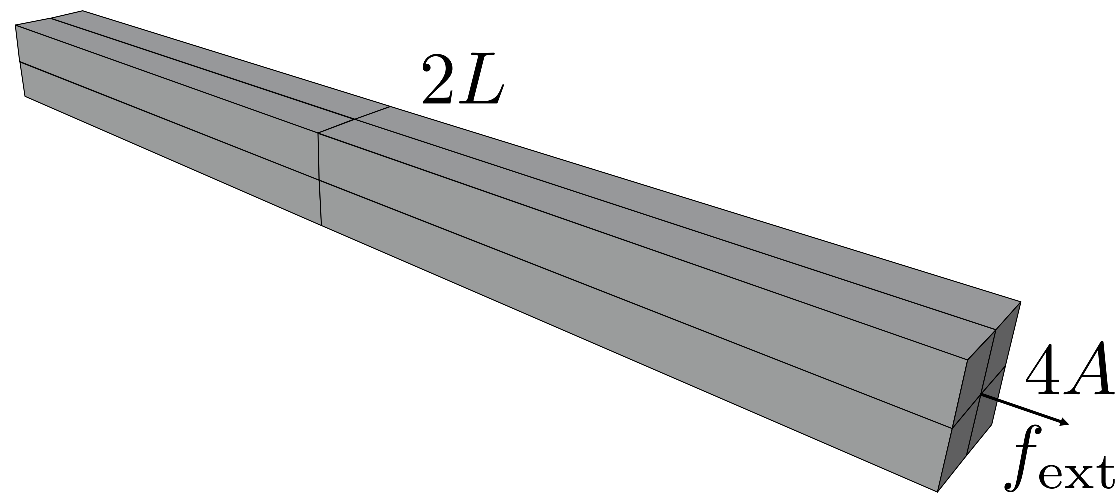


- Our force is spread across all 4 copies $\implies$ only $f_{\text{ext}}/4$ acts on each.

$$f_{\text{ext}}/4 = \tilde{k}\Delta L \quad \implies \quad f_{\text{ext}} = 4\tilde{k}\Delta L = \hat{k}\Delta L$$

- Spring constant changes to $\hat{k} = 4\tilde{k}$.

Material Properties



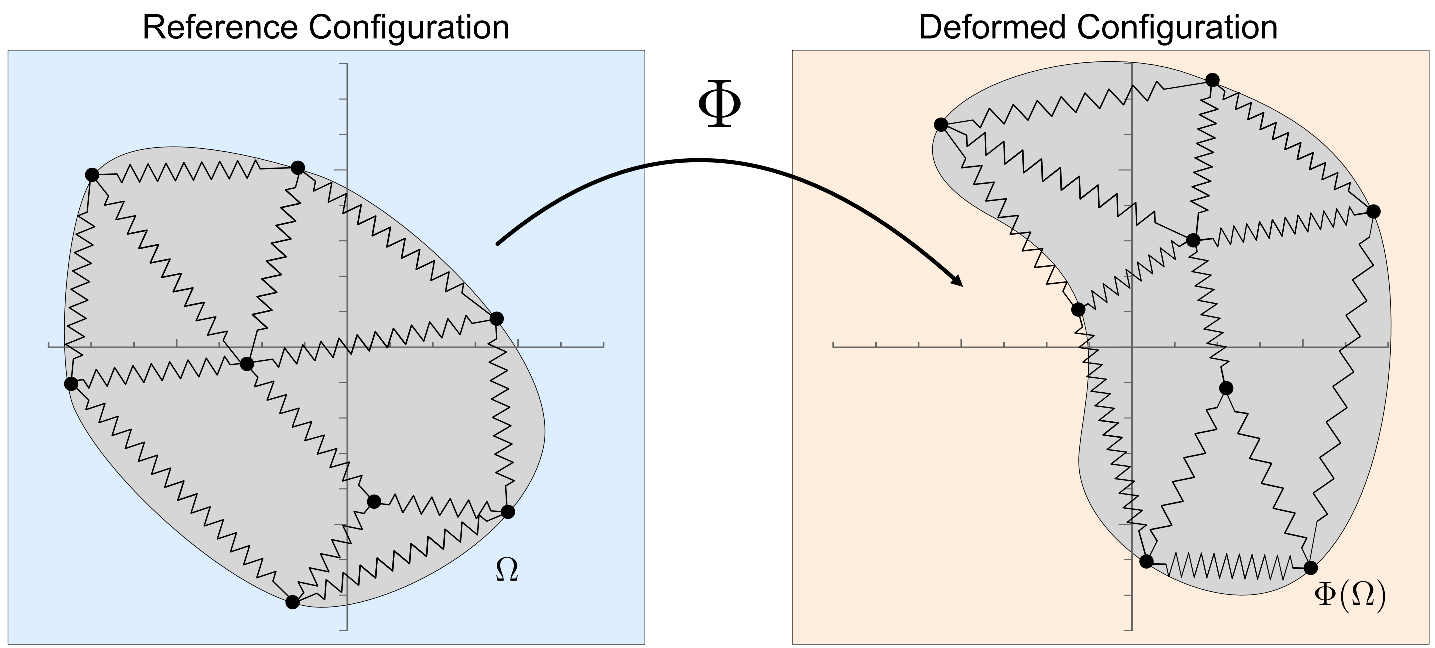
- Spring constant k confounds material properties with geometry.
  - Scaling length by s divides k by s.
  - Scaling cross-section area by s multiplies k by s.
  - Idea: use $Y = \frac{Lk}{A}$ as a geometry-independent measure of stiffness.

Young's modulus, Stress and Strain

- This material parameter "Y" (often denoted "E") is called the *Young's modulus*
  - Force per unit area (typical units: Pascals, **Megapascals**, Gigapascals)
  - Can recover spring stiffness by $k = \frac{YA}{L}$.
  - Better yet: work with
    - *strain* $\varepsilon := \Delta L / L$ (relative length change) instead of absolute length change
    - *stress* $\sigma := f_{\text{ext}} / A$ (applied force per unit area) instead of total force.
  - Then Hooke's law in 1D is just:

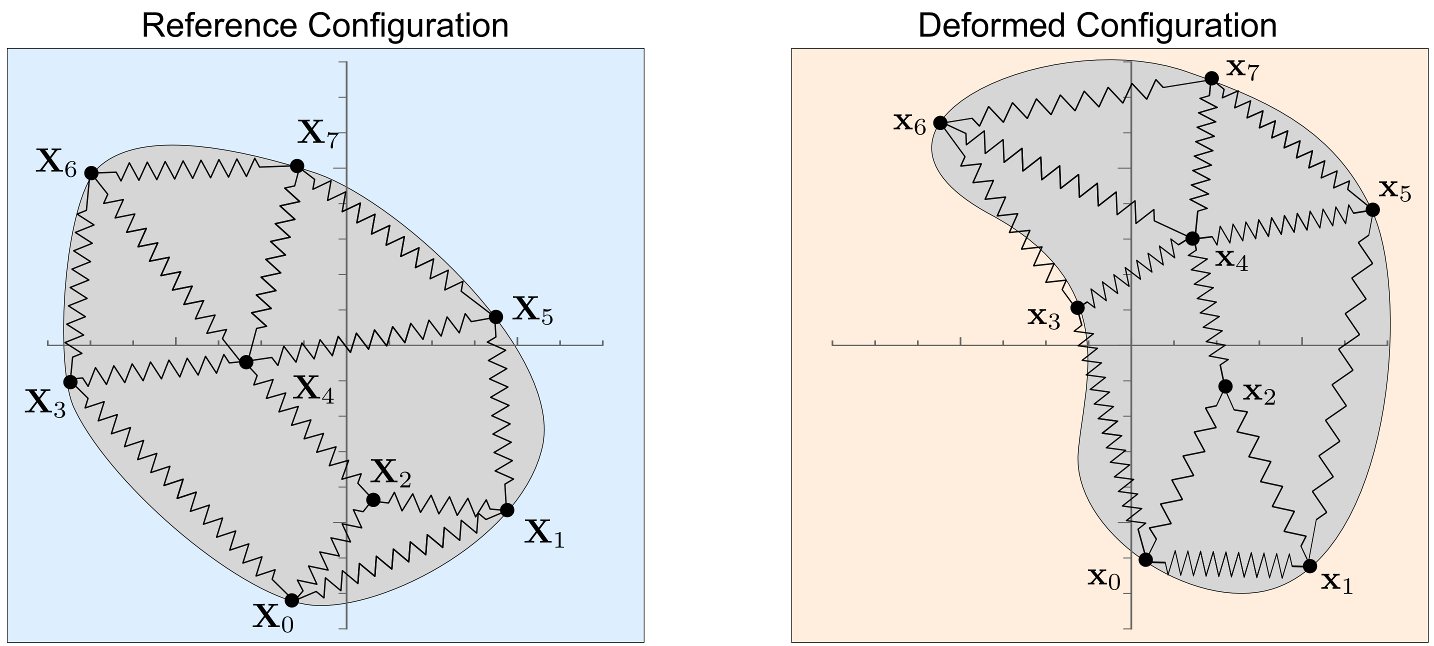
$$\sigma = Y\varepsilon$$

Recall: Spring Networks



- Rudimentary spring-based deformable object model:
    - Sample a set of points $\mathbf{X}_i$ and connect them with springs.
    - Now our deformed configuration is described by $\mathbf{x}_i = \Phi(\mathbf{X}_i)$.
      (Simulation problem is now optimizing a set of position variables.)

Spring System Energy

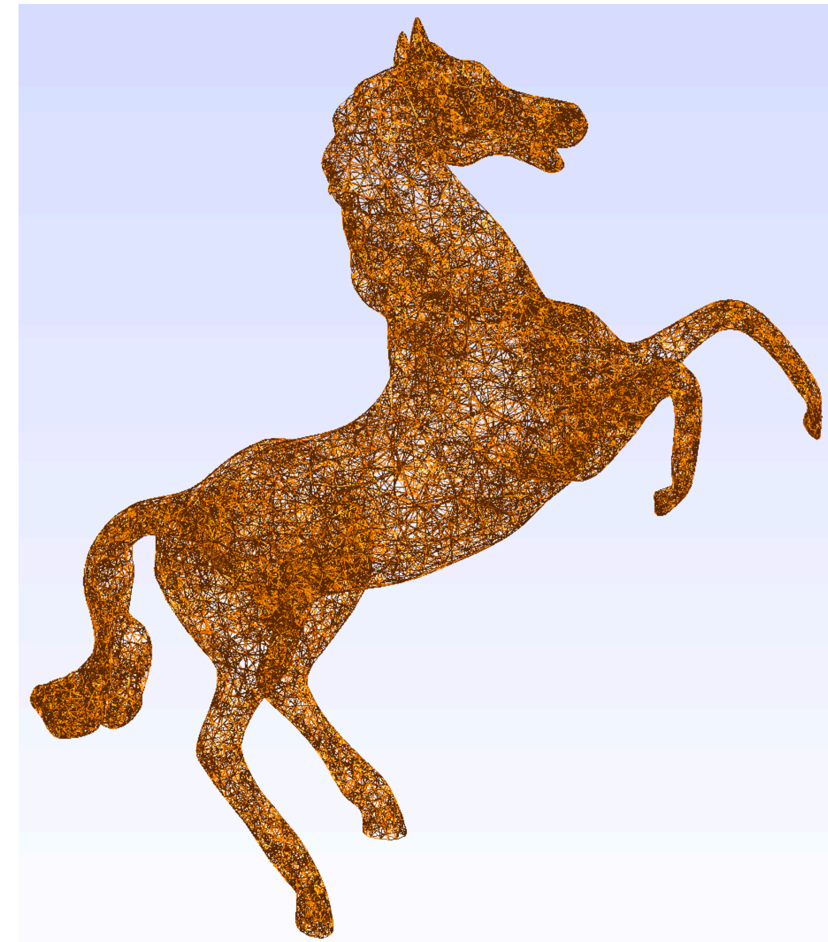


- Full elastic energy of spring system: just sum them up!

$$E_{\text{elastic}}(\mathbf{x}, \mathbf{X}) = \sum_{e_{ij}} \frac{1}{2} k_{ij} \big(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{X}_i - \mathbf{X}_j\| \big)^2$$

Spring Simulation Cons

- Behavior is extremely mesh dependent!
- Unclear what arrangement of springs will properly model bending/shearing.
- Alternative (coming up): a continuum-mechanics-based simulation.
  - Define elastic potential energy stored by arbitrary Φ
    - Define strain and stress for 2D/3D objects.
  - Discretize the function space Φ lives in (Finite Element Method)



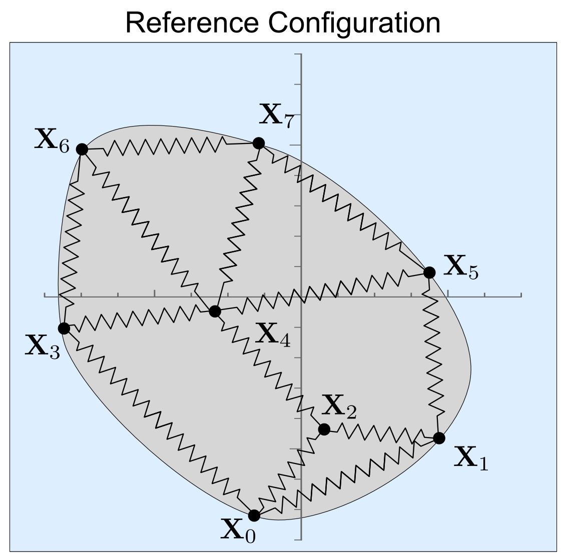
CS457 Geometric Computing

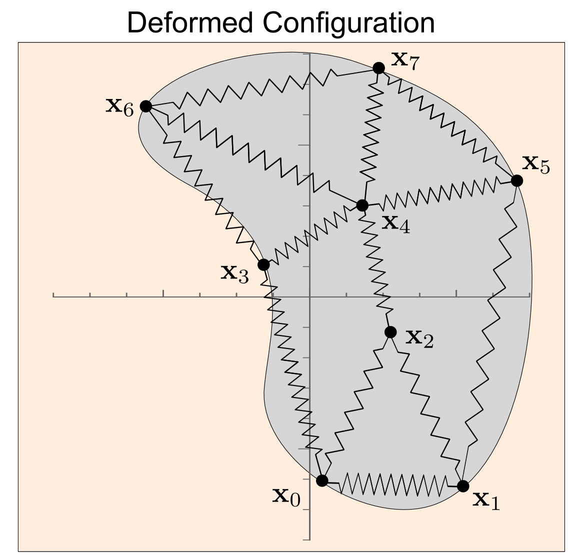
5B - Solid Mechanics I

Mark Pauly

Geometric Computing Laboratory - EPFL

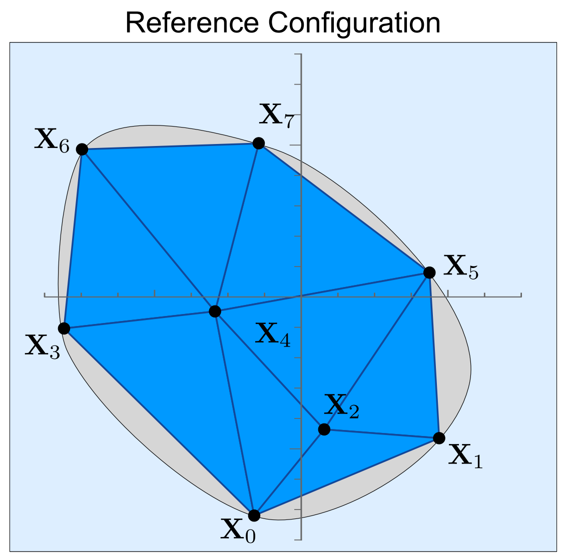
Continuum Mechanics

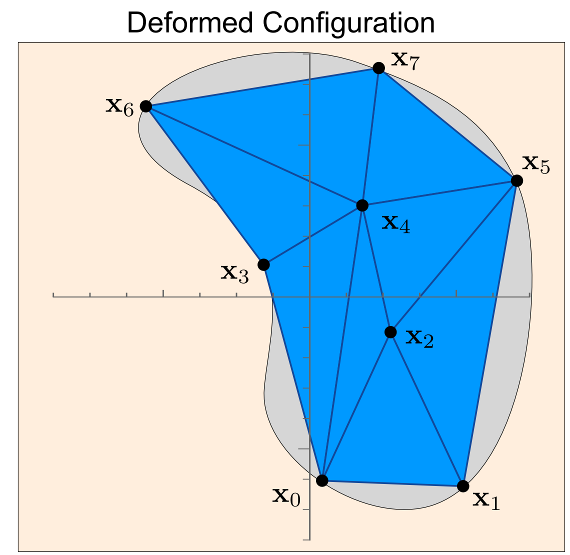




- Ultimately, we will replace 1D “edge springs”…

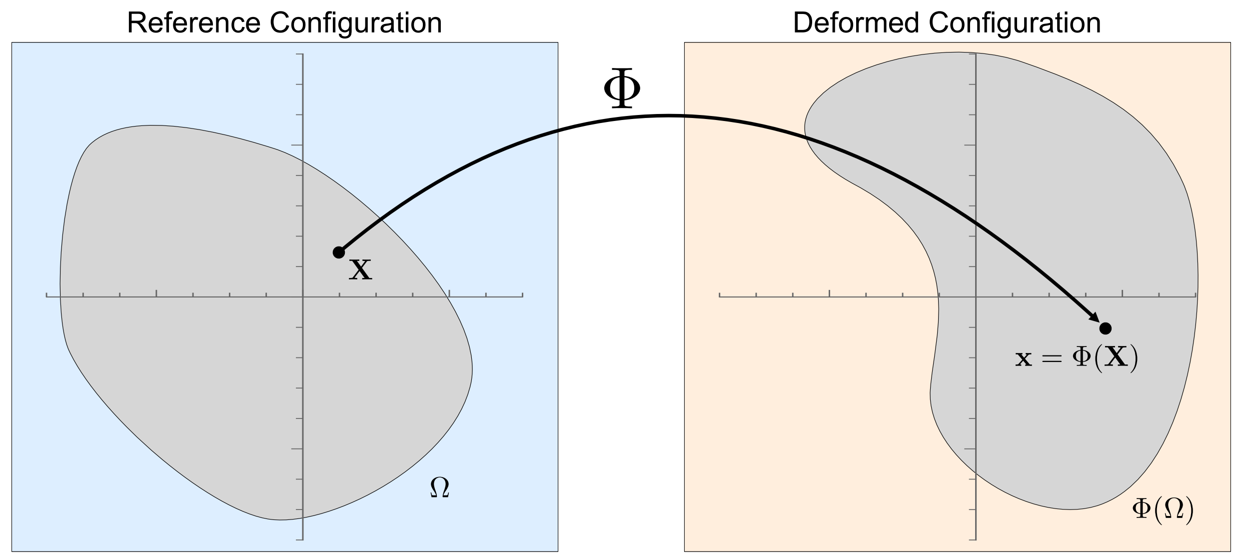
Continuum Mechanics





- Ultimately, we will replace 1D “edge springs” with 2D “triangle springs” and 3D “tetrahedron springs.”
- How do we ensure these “springs” are physically accurate/not badly mesh-dependent? Derive stiffnesses by discretizing a continuum mechanics model.

Continuum Mechanics

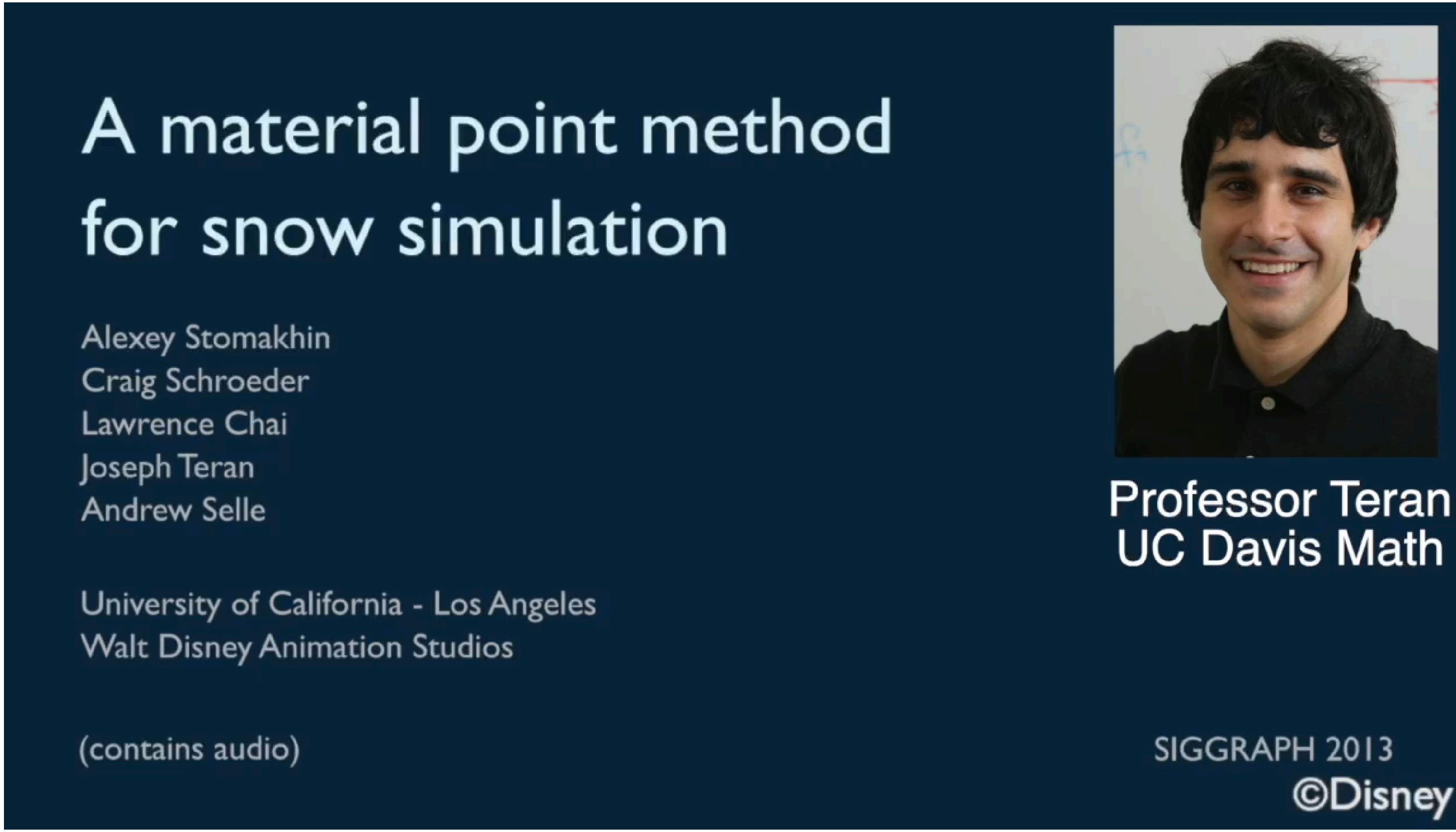


- We will study continuum mechanics models of the elastic energy stored in a volumetric solid due to an arbitrary deformation Φ.
- We will consider *hyperelastic* energies where the energy is purely a function of Φ (ignoring past state; path independent).
  - Energy returns to zero when the deformation returns to the identity map $\Phi(\mathbf{X}) = \mathbf{X}$.
  - Good approximation for the fabrication applications we care about.
  - Rules out *plastic* (irreversible) deformation–but these can be handled by updating the rest state.

Hyperelastic vs Non-Hyperelastic

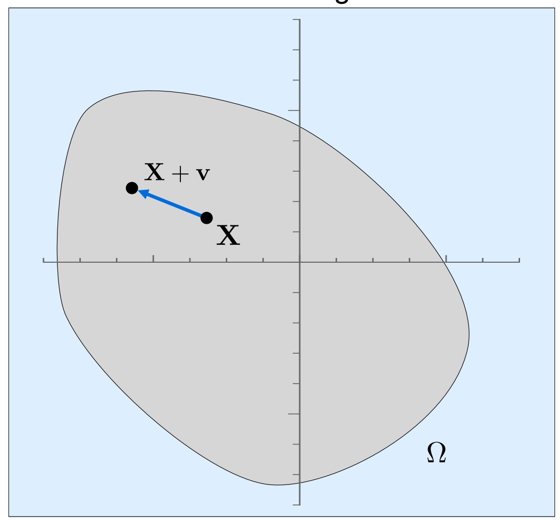
*Example of **Hyperelastic** Material: Silicone*

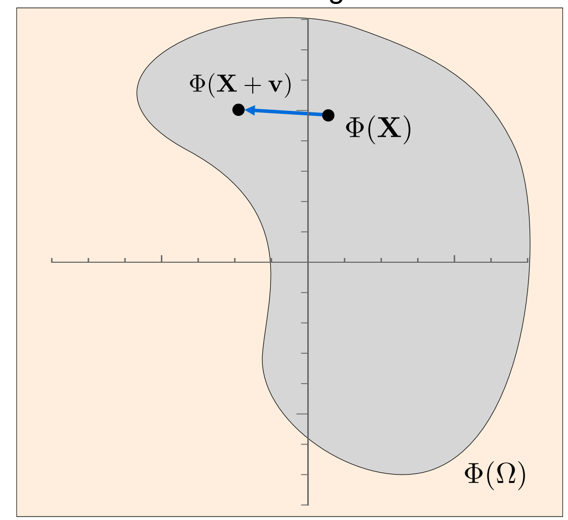
Hyperelastic vs Non-Hyperelastic



Example of ***non-Hyperelastic*** *Material: Snow. A material point method for snow simulation*

Distortions Induced by Φ





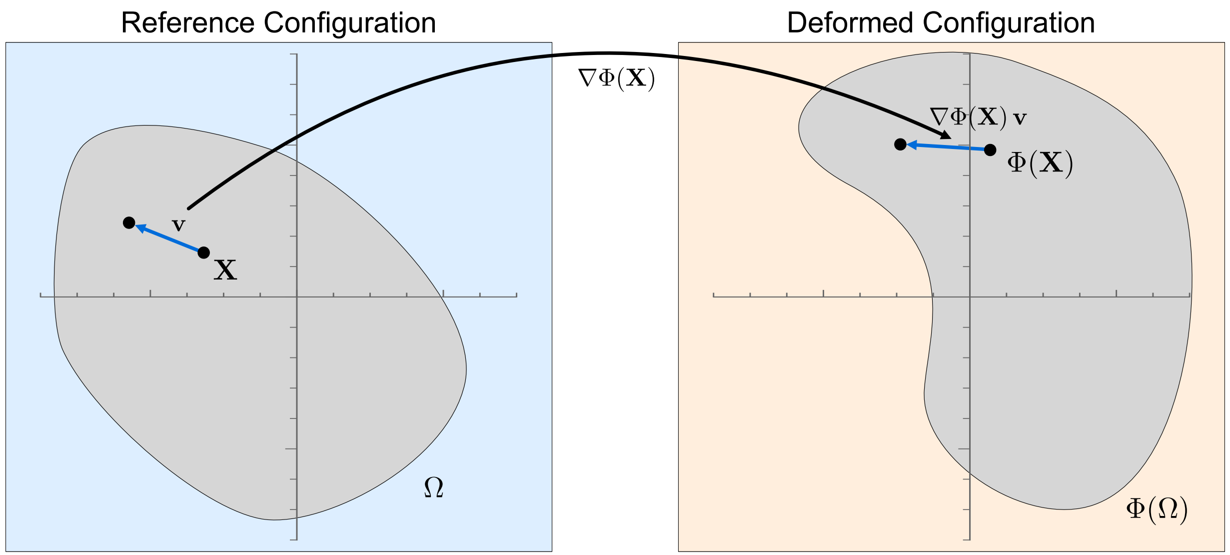
- Let's study how the mapping distorts small oriented "fibers" of the material.
  - Consider an infinitesimal vector $\mathbf{v}$ ($\|\mathbf{v}\| \ll 1$) connecting material points $\mathbf{X}$ and $\mathbf{X}+\mathbf{v}$ in rest config.
  - After deforming by Φ, this vector $\mathbf{v}$ is distorted into $\Phi(\mathbf{X}+\mathbf{v}) - \Phi(\mathbf{X})$.
  - From a Taylor expansion around $\mathbf{X}$, we see $\mathbf{v}$ is distorted into:

$$\left(\Phi(\mathbf{X}) + \nabla\Phi(\mathbf{X})\mathbf{v} + O(\|\mathbf{v}\|^2)\right) - \Phi(\mathbf{X}) \approx \nabla\Phi(\mathbf{X})\mathbf{v}.$$

  - This relationship $\mathbf{v} \mapsto \nabla\Phi(\mathbf{X})\mathbf{v}$ holds for *arbitrary* infinitesimal vectors $\mathbf{v}$.

$\nabla\Phi(\mathbf{X})$ encodes how fibers oriented in *all directions* at $\mathbf{X}$ get distorted!

Deformation Gradient



- Distortion is captured by $F := \nabla\Phi(\mathbf{X})$, the *deformation gradient* (*Jacobian*) of Φ.
- Elastic energy per unit volume can be formulated as an *energy density* function $\Psi(\nabla\Phi)$.
    - $\Psi : \mathbb{R}^{n\times n} \to \mathbb{R}$ (scalar-valued function of a matrix) encodes the material's elastic properties.
    - Ψ can optionally vary from point to point to model heterogeneous materials (*i.e.*, $\Psi(\nabla\Phi(\mathbf{X}), \mathbf{X})$).
    - We'll see later that Ψ must satisfy certain properties to make sense as a material model.

Elastic Energy and Forces

- Deformation gradient $\nabla\Phi(\mathbf{X})$, energy density function $\Psi(\nabla\Phi)$.
- The full elastic energy in the object is the integral of the energy density:

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

Compare to:
$E_{\text{spring}}(\mathbf{x},\mathbf{X}) = \sum_{e_{ij}} \frac{1}{2} k_{ij}\big(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{X}_i - \mathbf{X}_j\|\big)^2$

- 'Simple' formula for directional derivative along perturbation (displacement) $\delta\Phi$:

$$\left\langle \frac{\partial E_{\text{elastic}}}{\partial \Phi}, \delta\Phi \right\rangle := \frac{\mathrm{d}}{\mathrm{d}h}\bigg|_{h=0} \int_\Omega \Psi\Big(\nabla(\Phi + h\delta\Phi)\Big)\,\mathrm{d}\mathbf{X} = \boxed{\int_\Omega \Psi'(\nabla\Phi) : \nabla(\delta\Phi)\,\mathrm{d}\mathbf{X},}$$

- The matrix Ψ' is the derivative of Ψ with respect to its matrix argument:

$$[\Psi']_{ij} := \frac{\partial \Psi(F)}{\partial F_{ij}} \quad \implies \quad \Psi' : \delta F = \sum_{i,j} \frac{\partial \Psi(F)}{\partial F_{ij}} [\delta F]_{ij}.$$

Aside

Double Dot Products

The double dot product of two matrices produces a scalar result. It is written in matrix notation as $\mathbf{A} : \mathbf{B}$. Once again, its calculation is best explained with tensor notation.

$$\mathbf{A} : \mathbf{B} = A_{ij} B_{ij}$$

Since the i and j subscripts appear in both factors, they are both summed to give

$$\begin{aligned} \mathbf{A} : \mathbf{B} = A_{ij} B_{ij} = \; & A_{11} * B_{11} + A_{12} * B_{12} + A_{13} * B_{13} + \\ & A_{21} * B_{21} + A_{22} * B_{22} + A_{23} * B_{23} + \\ & A_{31} * B_{31} + A_{32} * B_{32} + A_{33} * B_{33} \end{aligned}$$

Double Dot Product Example

If $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 2 & 2 \\ 2 & 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ then

$$\begin{aligned} A_{ij} B_{ij} = \; & 1 * 1 + 2 * 4 + 3 * 7 + \\ & 4 * 2 + 2 * 5 + 2 * 8 + \\ & 2 * 3 + 3 * 6 + 4 * 9 \\ = \; & 124 \end{aligned}$$

Tensor Notation Basics

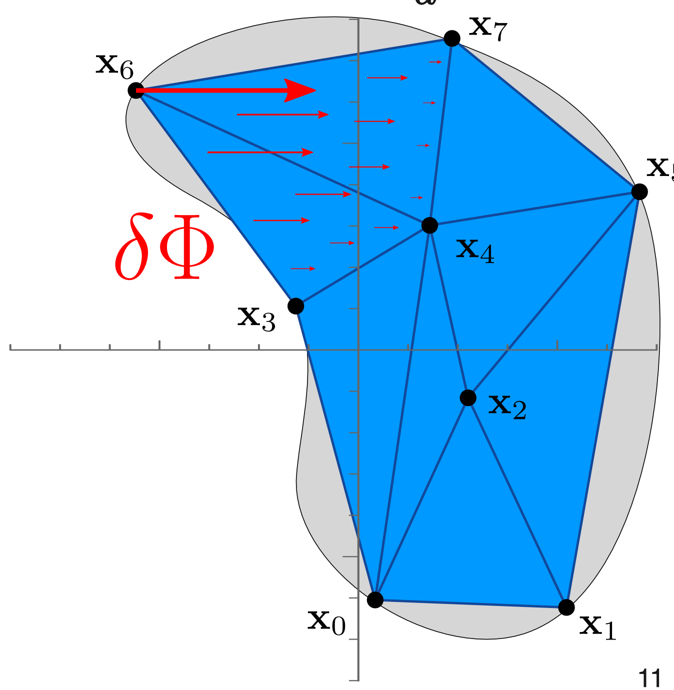
Elastic Energy and Forces

$$E_{\text{elastic}}[\Phi] = \int_{\Omega} \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X},$$
$$\left\langle \frac{\partial E_{\text{elastic}}}{\partial \Phi}[\Phi], \delta\Phi \right\rangle = \int_{\Omega} \Psi'(\nabla\Phi) : \nabla(\delta\Phi)\,\mathrm{d}\mathbf{X}.$$

- Physical interpretations: (negative) work done by elastic forces over displacement $\delta\Phi$.
- In our computations, Φ is controlled by a finite set of deformation variables x_a.
- Changing x_a induces perturbation $\delta\Phi_a$, allowing us to compute components of the gradient:

$$\frac{\partial E_{\text{elastic}}}{\partial x_a} = \left\langle \frac{\partial E_{\text{elastic}}}{\partial \Phi}, \delta\Phi_a \right\rangle$$

whose negation gives "elastic force" on variable x_a.



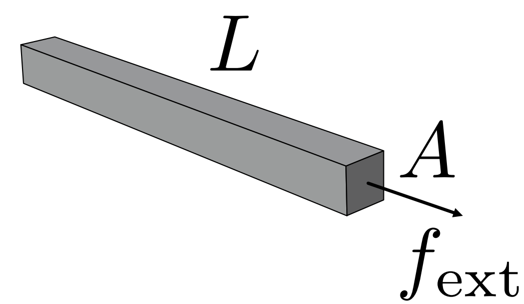
Remaining Questions

$$E_{\text{elastic}}[\Phi] = \int_\Omega \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X},$$
$$\left\langle \frac{\partial E_{\text{elastic}}}{\partial\Phi}[\Phi], \delta\Phi \right\rangle = \int_\Omega \Psi'(\nabla\Phi) : \nabla(\delta\Phi)\,\mathrm{d}\mathbf{X}.$$

- These formulas are powerful and “simple” (assuming comfort with tensors) but abstract.
  - How do we define $\Psi(F)$ to model a specific material?
  - How do we analyze the resulting deformation (will the object break?)
  - How to do all of this on a computer?
- To answer the first two, we will study the *stress* and *strain* occurring in our object.
- For computations, we will use the *finite element method* (FEM), discretizing the function space Φ lives in.

Strain and Stress

- Recall 1D Bar:



- We recommended working with the "geometry-invariant" properties:
  - *strain* $\varepsilon := \Delta L / L$ (relative length change) instead of absolute length change
  - *stress* $\sigma := f_{\text{ext}} / A$ (applied force per unit area) instead of total force.
- Then Hooke's law (linear material) in 1D is just:

$$\sigma = Y\varepsilon, \quad \text{energy density } \Psi = \frac{1}{2}\varepsilon\sigma = \frac{1}{2}Y\varepsilon^2.$$

- How do we generalize this to n dimensions?

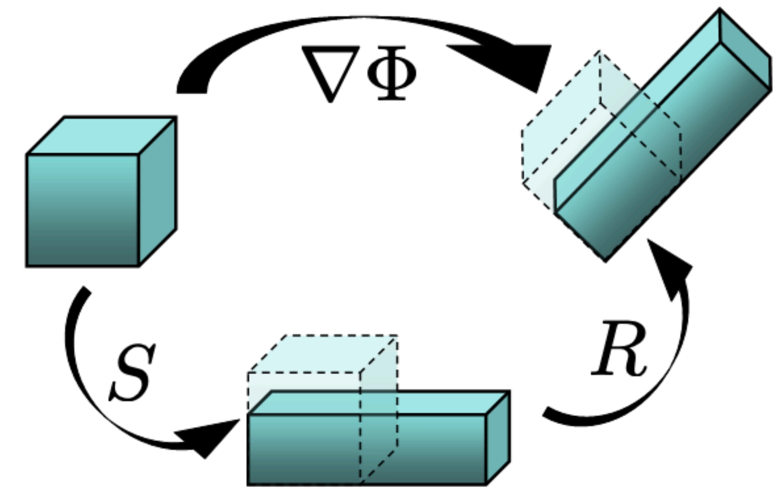
Can we just use $\nabla\Phi$ for strain?

Strain Measures

- Measuring distortion using the deformation gradient $\nabla\Phi$ has one main issue:
  - It is sensitive to *rotation* of material, which does not store or release elastic energy.
- To factor out rotation, we can use the ***polar decomposition***:

$$\nabla\Phi = RS,$$

  - S is a *symmetric matrix* representing pure stretching/compression along orthogonal axes.
  - R is a rotation matrix.
  - Always guaranteed to exist; is unique when $\nabla\Phi$ is nonsingular.

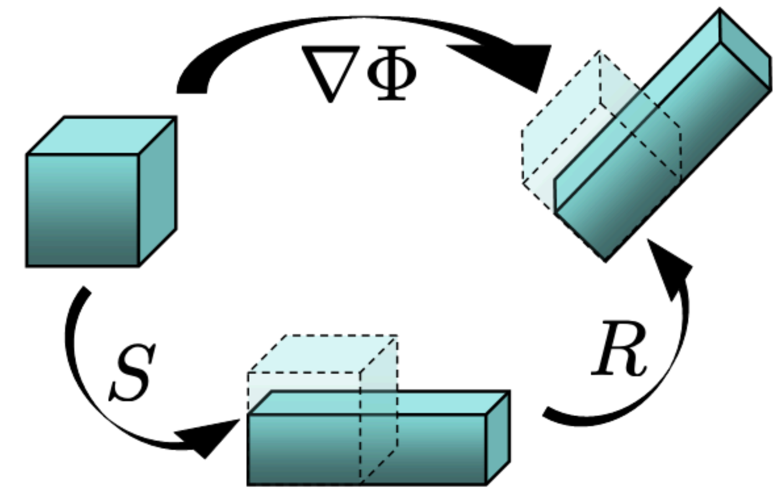


Strain Measures

- Measuring distortion using the deformation gradient $\nabla\Phi$ has one main issue:
  - It is sensitive to *rotation* of material, which does not store or release elastic energy.
- To factor out rotation, we can use the ***polar decomposition***:

$$\nabla\Phi = RS,$$

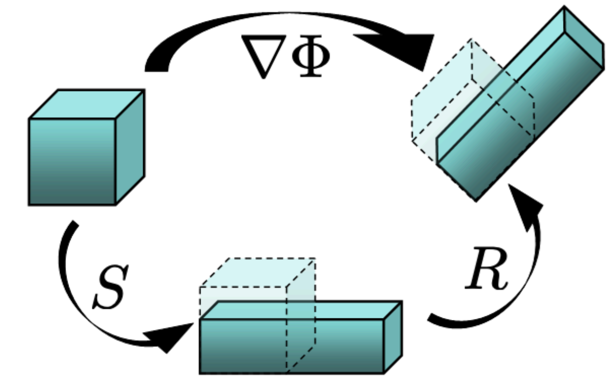
  - Now $S = I$ if and only if the material purely rotates.
  - *Biot strain* $\varepsilon_{\text{Biot}} := S - I$ is truly a measure of how much the material has distorted.
  - Computing and differentiating the polar decomposition can be inconvenient, so Biot strain is not commonly used.



Strain Measures

- Notice that we can compute the (rotation-invariant) *squared* length of a deformed material fiber as:

$$\|\nabla\Phi\mathbf{v}\|^2 = \mathbf{v}^\top(\nabla\Phi)^\top\nabla\Phi\mathbf{v}.$$



- The *change* in squared length is thus:

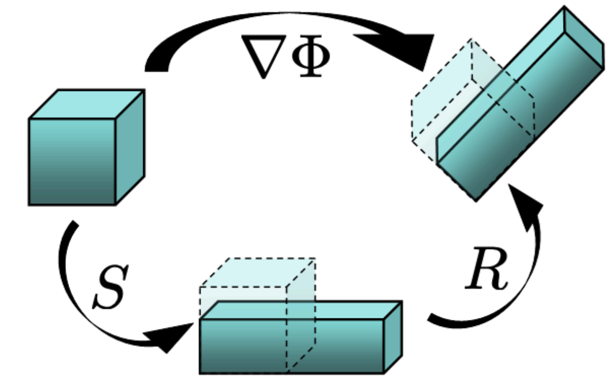
$$\|\nabla\Phi\mathbf{v}\|^2 - \|\mathbf{v}\|^2 \;=\; \mathbf{v}^\top(\nabla\Phi)^\top\nabla\Phi\mathbf{v} - \mathbf{v}^\top\mathbf{v} \;=\; \mathbf{v}^\top\Big((\nabla\Phi)^\top\nabla\Phi - I\Big)\mathbf{v}.$$

- This motivates the use of *Green-Lagrange strain*:

$$\varepsilon_{\text{Green}} := \frac{1}{2}\Big((\nabla\Phi)^\top\nabla\Phi - I\Big).$$

  - $\varepsilon_{\text{Green}}$ is rotation invariant since $(\nabla\Phi)^\top\nabla\Phi = (RS)^\top RS = SR^\top RS = S^2$.
  - Very popular due to its mathematical simplicity: no polar decomposition needed.

Strain Measures

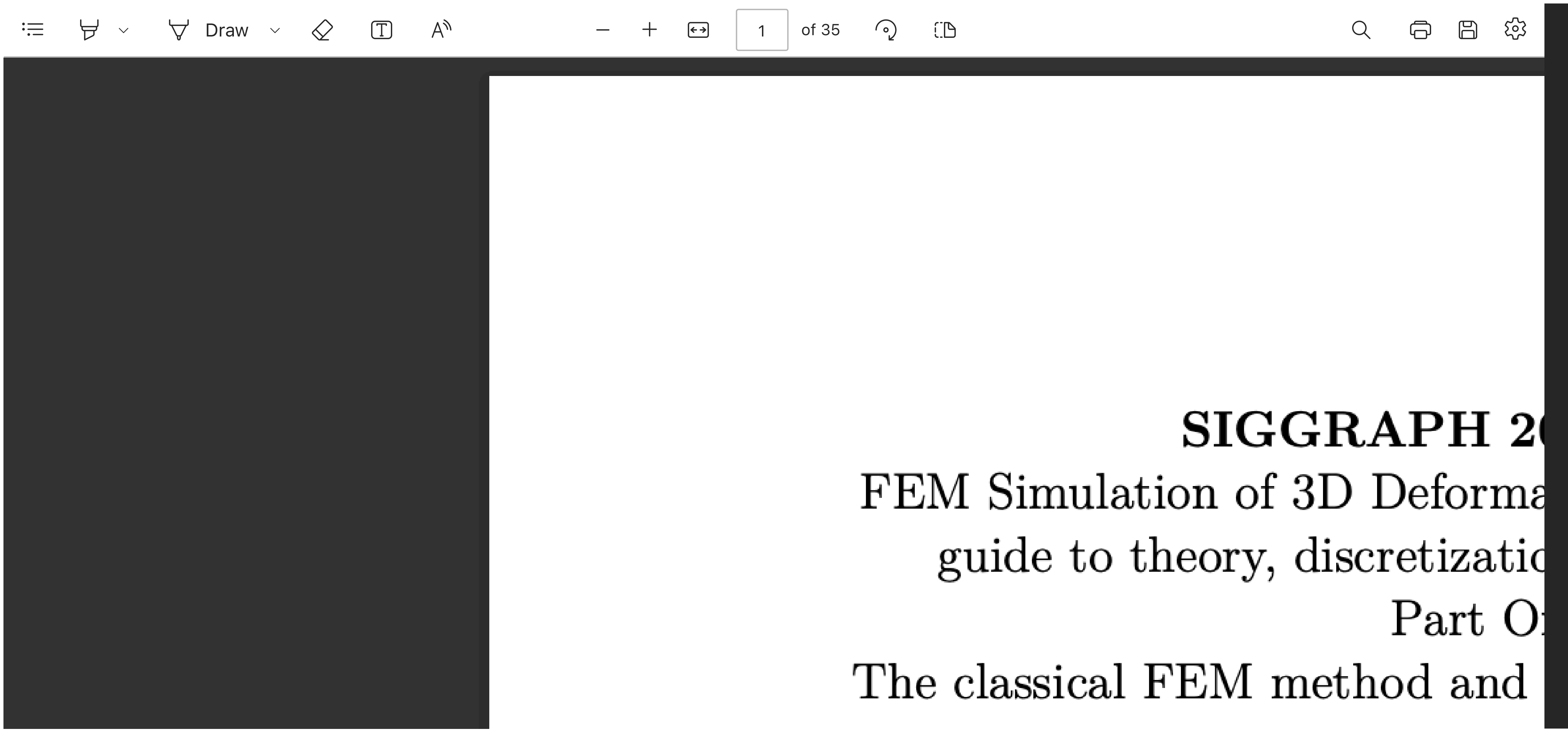


- Polar decomposition $\nabla\Phi = RS$
- Intuitive but somewhat difficult to compute measure:
  $\varepsilon_{\text{Biot}} = S - I.$
- Convenient, popular strain measure:
  $\varepsilon_{\text{Green}} := \frac{1}{2}\Big((\nabla\Phi)^\top \nabla\Phi - I\Big).$
- Many more strain measures have been used: Hencky Strain, Almansi Strain, ...
- All of these are equivalent at small strains (and will linearize to Cauchy strain tensor when we study linear elasticity)!

$$\varepsilon_{\text{Green}} = \frac{1}{2}(S^2 - I) = \frac{1}{2}\Big((\varepsilon_{\text{Biot}} + I)^2 - I\Big) = \varepsilon_{\text{Biot}} + \frac{1}{2}\varepsilon_{\text{Biot}}^2$$

- We can easily ensure energy density Ψ depends on pure deformation (not rotation), $\Psi(\nabla\Phi) = \Psi(RS) = \Psi(S)$ by defining Ψ in terms of strain!
  - This rotational invariance is needed for Ψ to make physical sense.

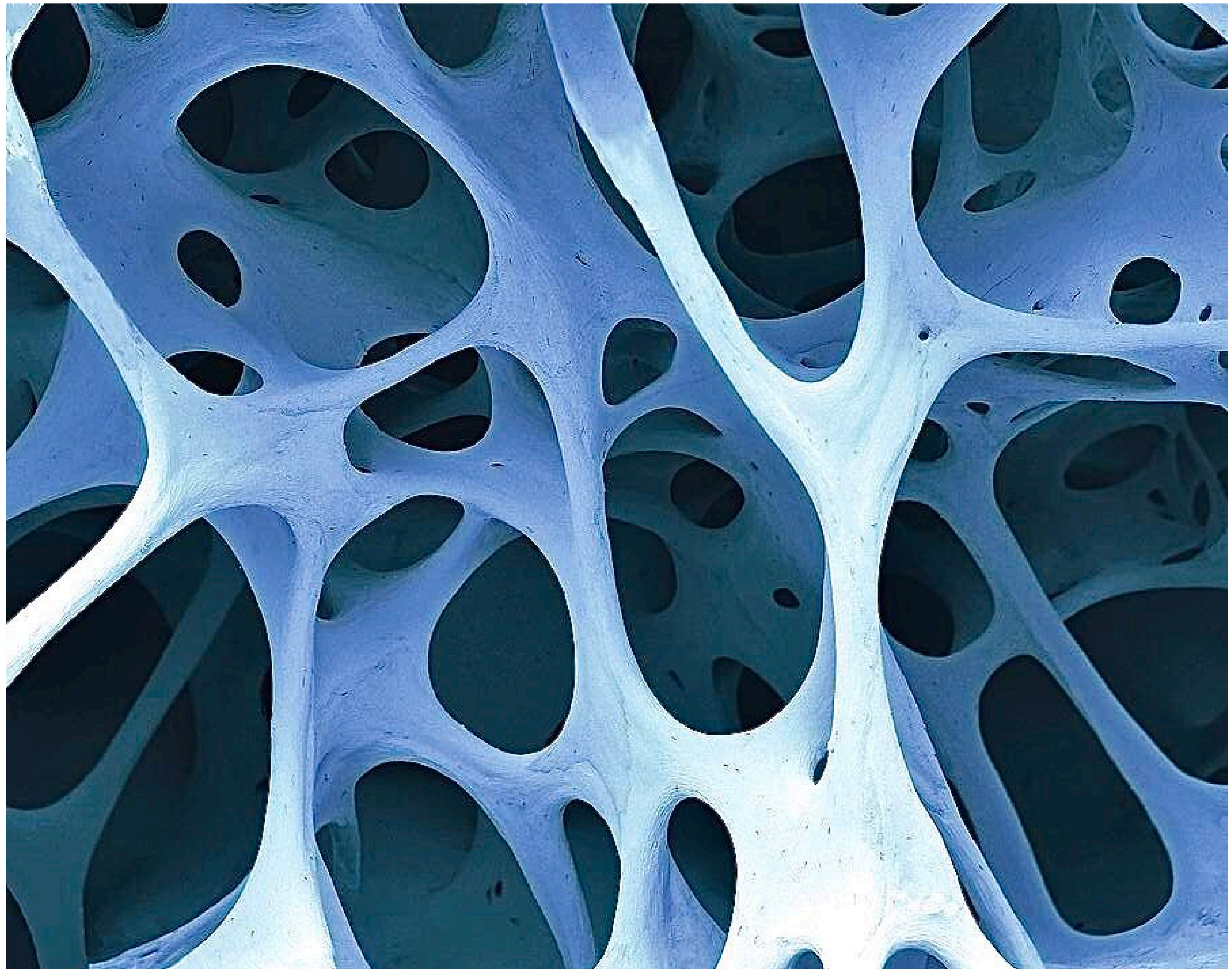
Reading



Siggraph 2012 Course Notes – femdefo.org

- More background: Bonet, Javier, and Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. 2nd ed., Cambridge University Press, 2008.

Optimal Structures: Bird Bone



Source

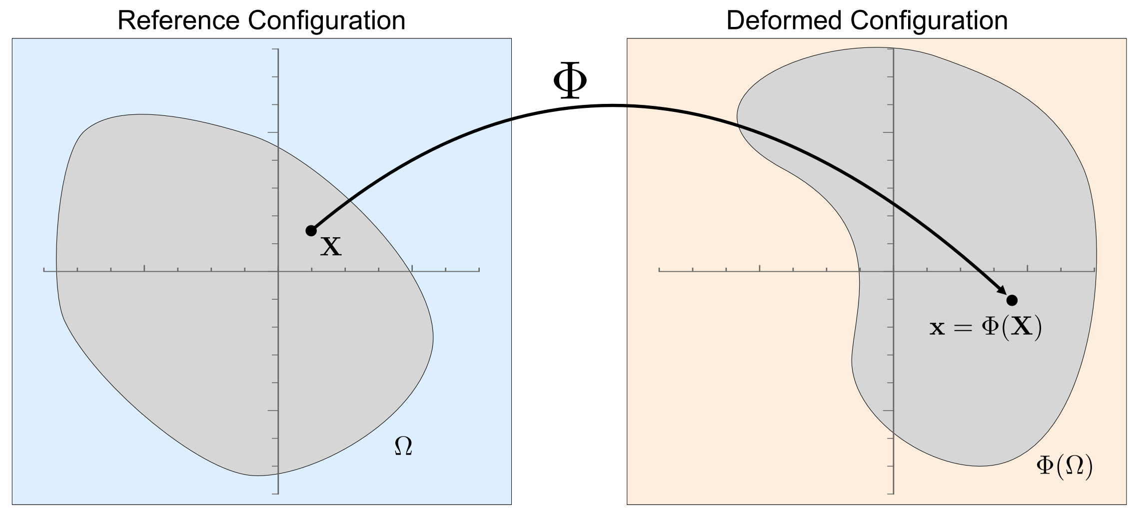
CS457 Geometric Computing

5C - Solid Mechanics II

Mark Pauly

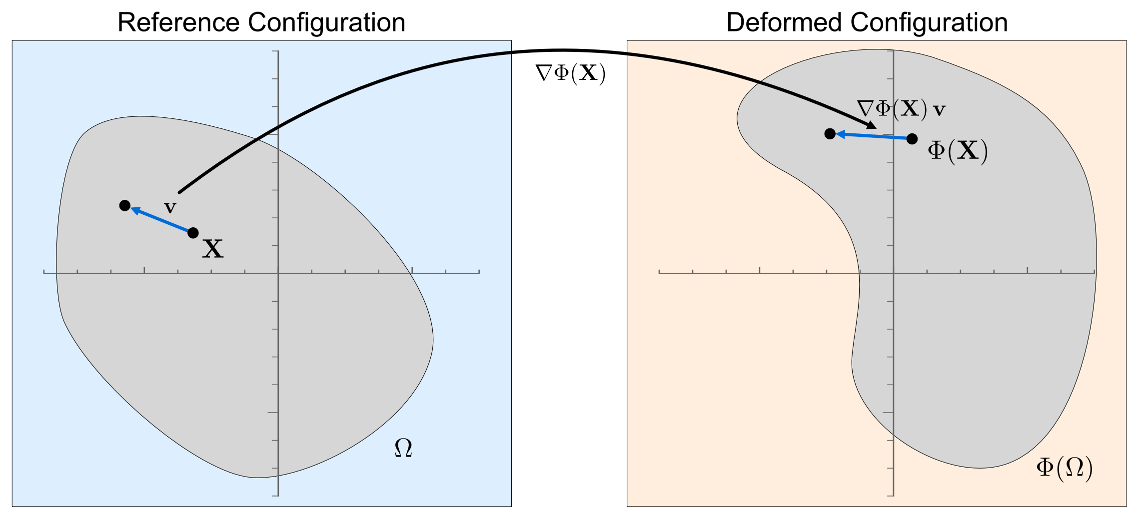
Geometric Computing Laboratory - EPFL

Recap: Deformation Gradient and Elastic Energy



- $\mathbf{\Phi} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defines deformation of a solid.

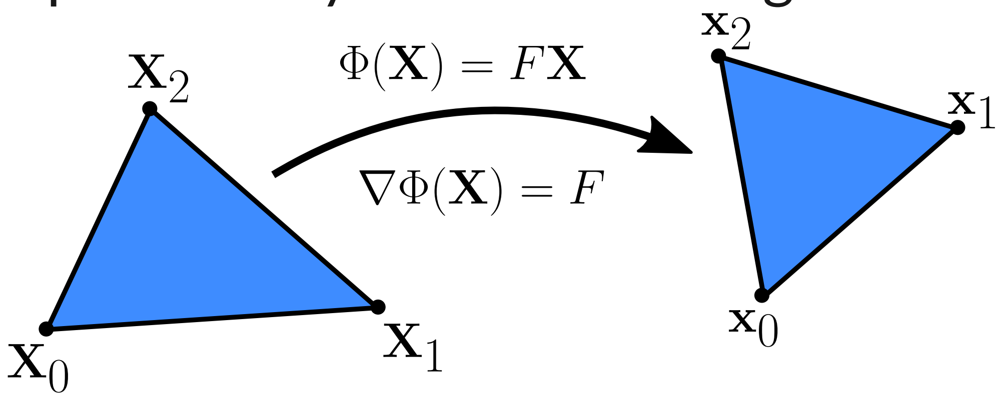
Recap: Deformation Gradient and Elastic Energy



- $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ defines deformation of a solid.
- $\nabla\Phi(\mathbf{X})$ is the *Jacobian* of Φ, $\frac{\partial \Phi_i}{\partial \mathbf{X}_j}$.
  - Encodes how short fibers oriented in any direction at $\mathbf{X}$ get distorted.
  - Usually called the ***deformation gradient*** and given the label F.
- Hyperelastic material model: elastic energy density function $\Psi(F)$
  - Function mapping 3×3 matrices to scalars, $\Psi : \mathbb{R}^{3\times 3} \to \mathbb{R}$.
  - Measures the energy stored in the material due the stretching caused by F.
- Elastic energy $E_{\text{elastic}}[\Phi] := \int_\Omega \Psi(\nabla\Phi)\,\mathrm{d}\mathbf{X}$

Recap: Strain

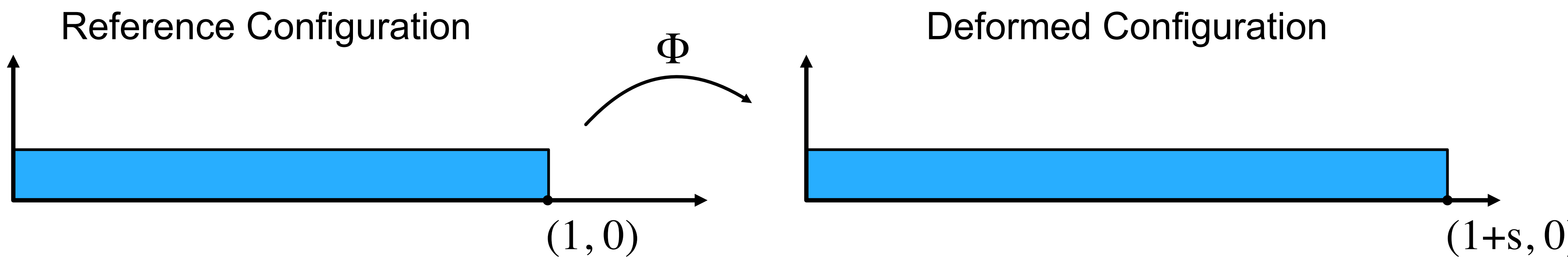
- How much is material stretched/compressed by a deformation gradient F?
- Example: linearly deformed triangle



  - Material stretch/compression is unaffected by rotating the triangle.
  - Deformations $\Phi(\mathbf{X}) = F\mathbf{X}$ and $\tilde{\Phi}(\mathbf{X}) = RF\mathbf{X}$ apply the same distortion.
  - When $F = R$, the deformation is (locally) pure rotation, causing no stretch.
- ***Strain tensors*** give us a measure of distortion that *factors out this rotation.*
  - Biot strain: $\varepsilon_{\text{Biot}} := S - I$ where S is from the ***polar decomposition*** $F = RS$
  - Green strain $\varepsilon_{\text{Green}} := \frac{1}{2}\left((\nabla\Phi)^\top \nabla\Phi - I\right) = \frac{1}{2}(S^2 - I)$
  - Strain tensor is zero if and only if the material transforms rigidly.

Elastic energy density Ψ will measure the “magnitude” of ε!

Simple 2D Strain Example



- 2D bar has been stretched horizontally by the *constant strain* deformation:

$$\Phi(\mathbf{X}) = \begin{bmatrix} 1+s & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X}$$

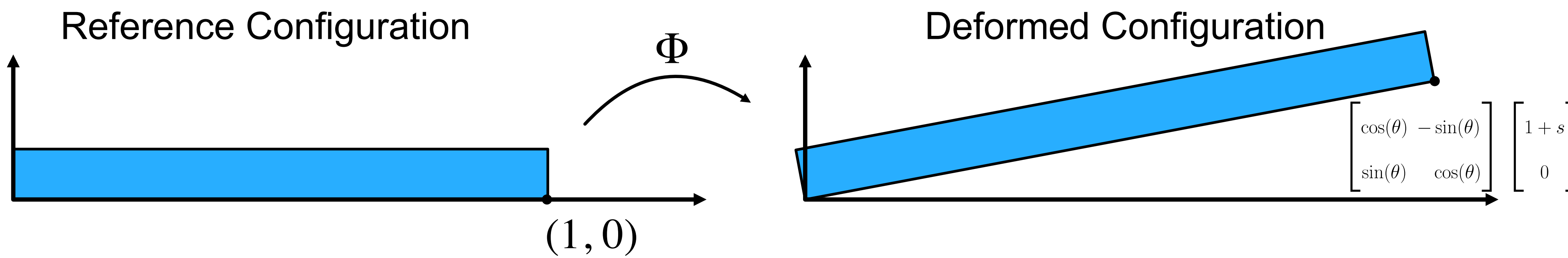
- Deformation gradient is just the constant matrix:

$$\nabla\Phi(\mathbf{X}) = \begin{bmatrix} 1+s & 0 \\ 0 & 1 \end{bmatrix} \xrightarrow{\text{polar decomposition}} \nabla\Phi(\mathbf{X}) = RS \text{ with } R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ S = \begin{bmatrix} 1+s & 0 \\ 0 & 1 \end{bmatrix}$$

- Strain measures agree to first order in s (and $[\varepsilon_{\text{Biot}}]_{00}$ matches 1D strain definition):

$$\varepsilon_{\text{Biot}} = S - I = \begin{bmatrix} s & 0 \\ 0 & 0 \end{bmatrix}, \quad \varepsilon_{\text{Green}} = \frac{1}{2}(S^2 - I) = \frac{1}{2}\left(\begin{bmatrix} (1+s)^2 - 1 & 0 \\ 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} s + \frac{s^2}{2} & 0 \\ 0 & 0 \end{bmatrix}.$$

Simple 2D Strain Example



- 2D bar has been deformed by the *constant strain* map:

$$\Phi(\mathbf{X}) = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}1+s & 0\\ 0 & 1\end{bmatrix}\mathbf{X}$$

- Deformation gradient is just the constant matrix:

$$\nabla\Phi(\mathbf{X}) = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}1+s & 0\\ 0 & 1\end{bmatrix} \xrightarrow{\text{polar decomposition}} \nabla\Phi(\mathbf{X}) = RS \ \text{ with } \ R = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix},\ S = \begin{bmatrix}1+s & 0\\ 0 & 1\end{bmatrix}$$

- Strain measures agree to first order in s (and $[\varepsilon_{\text{Biot}}]_{00}$ matches 1D strain definition):

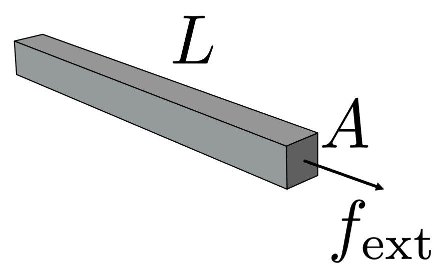
$$\varepsilon_{\text{Biot}} = S - I = \begin{bmatrix}s & 0\\ 0 & 0\end{bmatrix}, \quad \varepsilon_{\text{Green}} = \frac{1}{2}(S^2 - I) = \frac{1}{2}\left(\begin{bmatrix}(1+s)^2 - 1 & 0\\ 0 & 0\end{bmatrix}\right) = \begin{bmatrix}s + \frac{s^2}{2} & 0\\ 0 & 0\end{bmatrix}.$$

Stress Measures

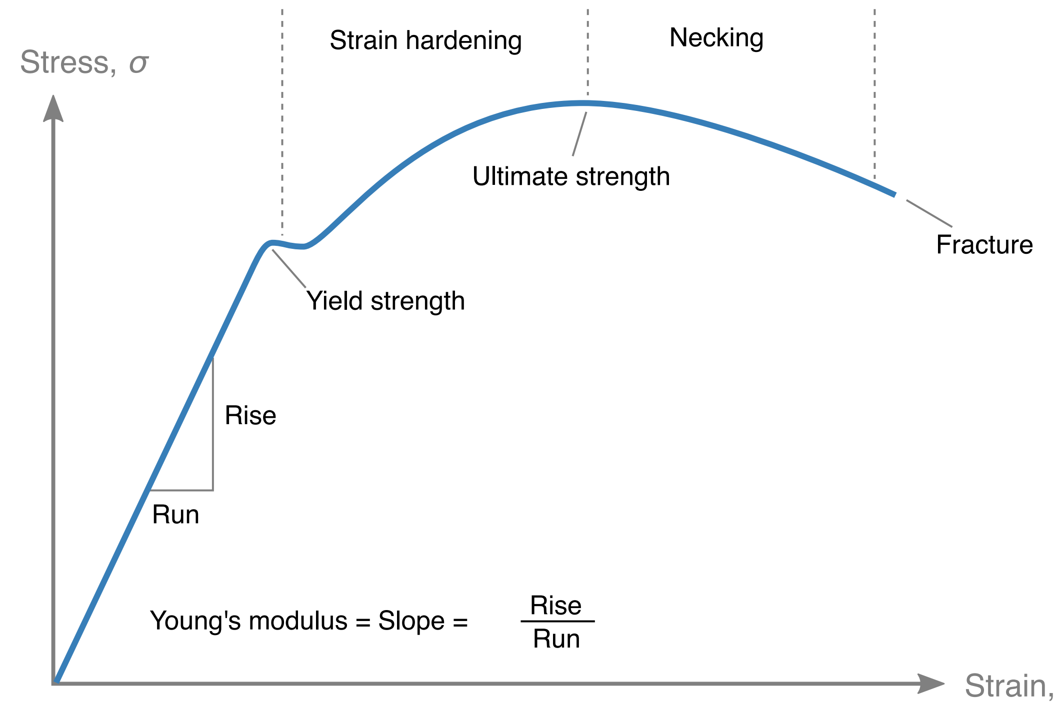
- When material is stretched by a strain ε , it responds with elastic restoring forces.
  - For beams, it was convenient to look at the restoring force *per unit area*
    - Stress $\sigma := f_{\text{ext}}/A$
    - Hooke's law in 1D:

$$\sigma = Y\varepsilon,$$

valid for small strains.

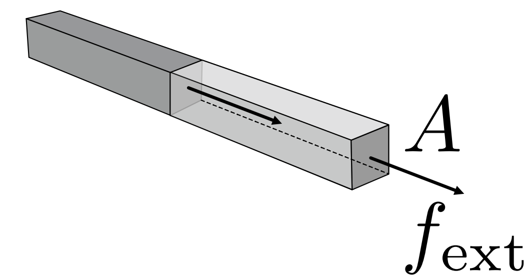


- Stresses are important because they tell us when an object will break.
  - This concept is *not actually needed to define a hyperelastic material model* Ψ.
  - The elastic energy density in a beam is $\frac{1}{2}Y\varepsilon^2$.
- How do we define stress for 3D solids?



Young's modulus is the initial stress/strain slope

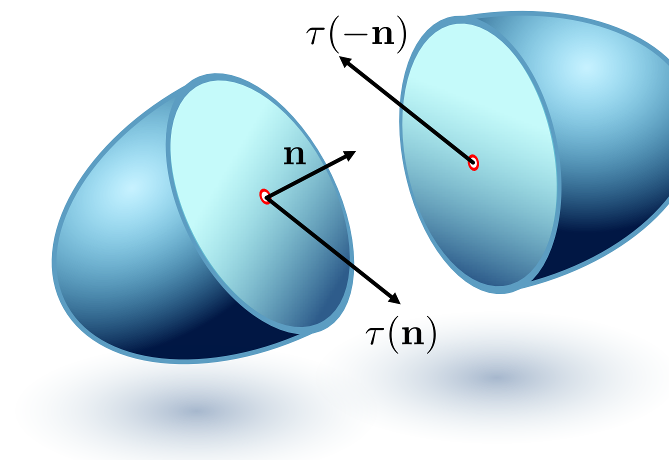
Stress in Beams



- Recall 1D Bar
  - Suppose we clamp the “back” and pull on the “front” with stress $\sigma := f_{\text{ext}}/A$ (applied force per unit area)
- Inside the bar at equilibrium, we have analogous forces
  - Material on one side of each slicing plane pulls on material on the opposite side.
  - In equilibrium, every orthogonal slicing plane experiences the same force over the same area.
  - We say this is a state of *constant stress* σ.
- Stress is a scalar quantity here since we consider only 1D (axial) forces that act on material planes with a single orientation (perpendicular to axis).

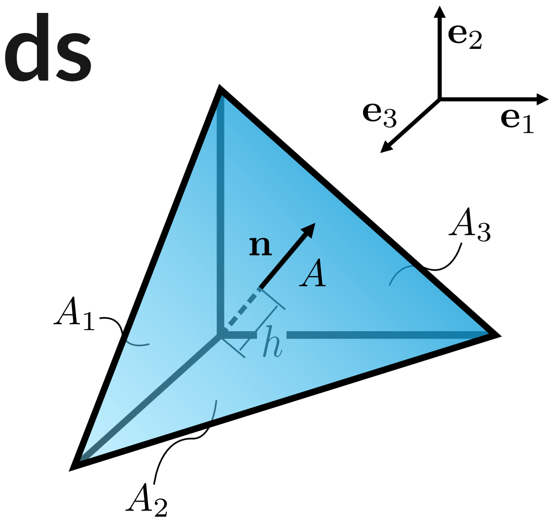
Stress in 3D Solids

- How does this generalize to 3D solids?
  - Arbitrary slicing plane orientation $\mathbf{n}$
  - Arbitrary force-per-unit-area ("traction") vector $\tau(\mathbf{n})$ acting *on* this face.



- $\tau(\mathbf{n})$ is some function of $\mathbf{n}$. What form must it take?
  - By Newton's third law $\tau(-\mathbf{n}) = -\tau(\mathbf{n})$.
  - Can we say more? Yes, much more!

Stress in 3D Solids



- Cauchy's tetrahedron argument (1822)
  - Let's calculate $\tau(\mathbf{n})$ at a point in the material by considering this special tiny tetrahedron:

- Net force acting on tetrahedron: sum forces acting on each face.

$$\begin{aligned}\mathbf{f} &= \tau(\mathbf{n})A + \tau(-\mathbf{e}_1)A_1 + \tau(-\mathbf{e}_2)A_2 + \tau(-\mathbf{e}_3)A_3 \\ &= \tau(\mathbf{n})A - \tau(\mathbf{e}_1)A_1 - \tau(\mathbf{e}_2)A_2 - \tau(\mathbf{e}_3)A_3\end{aligned}$$

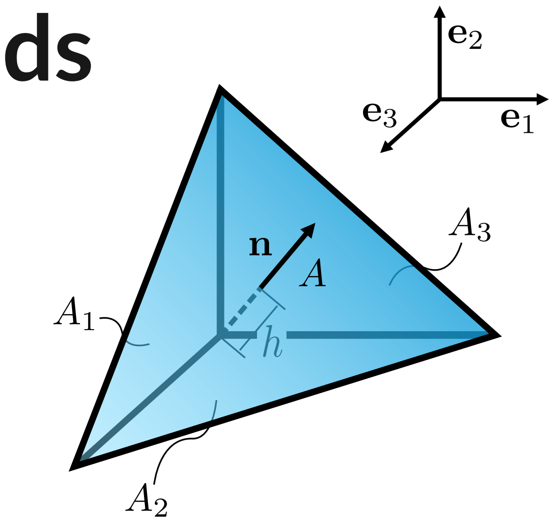
- We can calculate each face area (projection of $\mathbf{n}A$ onto each coordinate plane):

$$A_1 = (\mathbf{n}\cdot\mathbf{e}_1)A, \quad A_2 = (\mathbf{n}\cdot\mathbf{e}_2)A, \quad A_3 = (\mathbf{n}\cdot\mathbf{e}_3)A.$$

$$\implies \quad \mathbf{f} = \Big(\tau(\mathbf{n}) - \tau(\mathbf{e}_1)(\mathbf{n}\cdot\mathbf{e}_1) - \tau(\mathbf{e}_2)(\mathbf{n}\cdot\mathbf{e}_2) - \tau(\mathbf{e}_3)(\mathbf{n}\cdot\mathbf{e}_3)\Big)A$$

Stress in 3D Solids

- Cauchy's tetrahedron argument (1822)
  - Let's calculate $\tau(\mathbf{n})$ at a point in the material by considering this special tiny tetrahedron:



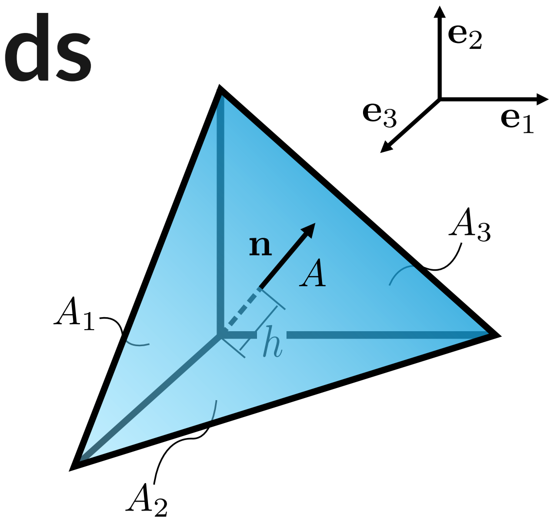
- Net force acting on tetrahedron: sum forces acting on each face.

$$\mathbf{f} = \Big(\tau(\mathbf{n}) - \tau(\mathbf{e}_1)(\mathbf{n}\cdot\mathbf{e}_1) - \tau(\mathbf{e}_2)(\mathbf{n}\cdot\mathbf{e}_2) - \tau(\mathbf{e}_3)(\mathbf{n}\cdot\mathbf{e}_3)\Big)A$$

- This produces acceleration:

$$\mathbf{a} = \frac{\mathbf{f}}{m} = \frac{\mathbf{f}}{\rho V} = \frac{1}{\rho\frac{1}{3}Ah}\mathbf{f} = \frac{3}{\rho h}\Big(\tau(\mathbf{n}) - \tau(\mathbf{e}_1)(\mathbf{n}\cdot\mathbf{e}_1) - \tau(\mathbf{e}_2)(\mathbf{n}\cdot\mathbf{e}_2) - \tau(\mathbf{e}_3)(\mathbf{n}\cdot\mathbf{e}_3)\Big).$$

Stress in 3D Solids



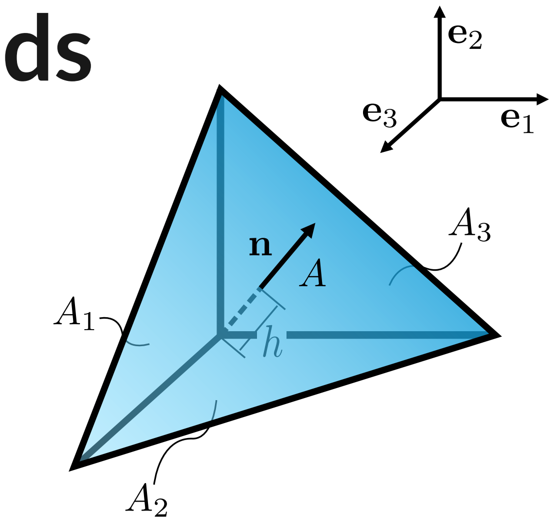
- Cauchy's tetrahedron argument (1822)
  - Let's calculate $\tau(\mathbf{n})$ at a point in the material by considering this special tiny tetrahedron:

- Acceleration: $\mathbf{a} = \frac{3}{\rho h}\Big(\tau(\mathbf{n}) - \tau(\mathbf{e}_1)(\mathbf{n}\cdot\mathbf{e}_1) - \tau(\mathbf{e}_2)(\mathbf{n}\cdot\mathbf{e}_2) - \tau(\mathbf{e}_3)(\mathbf{n}\cdot\mathbf{e}_3)\Big)$.

- In the limit as the tetrahedron shrinks to zero, $h \to 0$.
  For the acceleration to be finite, expression in parentheses must vanish!

$$\begin{aligned}\tau(\mathbf{n}) &= \tau(\mathbf{e}_1)(\mathbf{n}\cdot\mathbf{e}_1) + \tau(\mathbf{e}_2)(\mathbf{n}\cdot\mathbf{e}_2) + \tau(\mathbf{e}_3)(\mathbf{n}\cdot\mathbf{e}_3) \\ &= [\tau(\mathbf{e}_1) \mid \tau(\mathbf{e}_2) \mid \tau(\mathbf{e}_3)]\begin{pmatrix}\mathbf{n}\cdot\mathbf{e}_1 \\ \mathbf{n}\cdot\mathbf{e}_2 \\ \mathbf{n}\cdot\mathbf{e}_3\end{pmatrix} = \sigma\mathbf{n}\end{aligned}$$

Stress in 3D Solids



- Cauchy's tetrahedron argument (1822)
  - Let's calculate $\tau(\mathbf{n})$ at a point in the material by considering this special tiny tetrahedron:

$$\sigma = [\tau(\mathbf{e}_1) \mid \tau(\mathbf{e}_2) \mid \tau(\mathbf{e}_3)]$$

- We have shown traction $\tau(\mathbf{n}) = \sigma\mathbf{n}$.
  - Traction acting *on* slicing plane with orientation $\mathbf{n}$ is a linear function of $\mathbf{n}$, represented by matrix σ
- σ is called the *Cauchy stress tensor*; it's a 3×3 matrix for 3D problems.
- Requiring finite *angular* acceleration on an infinitesimal cube shows σ is symmetric.
  - Symmetry means σ has three orthogonal eigenvectors $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$, the "directions of principal stress"
  - $\mathbf{n}_1$ is the normal of the slicing plane with the greatest tension (least compression) $\sigma\mathbf{n}_1$ acting across it.
  - $\mathbf{n}_3$ is the normal of the slicing plane with the greatest compression (least tension) $\sigma\mathbf{n}_3$ acting across it.
- Note: plane orientations $\mathbf{n}$ and areas are measured in the *deformed configuration*!

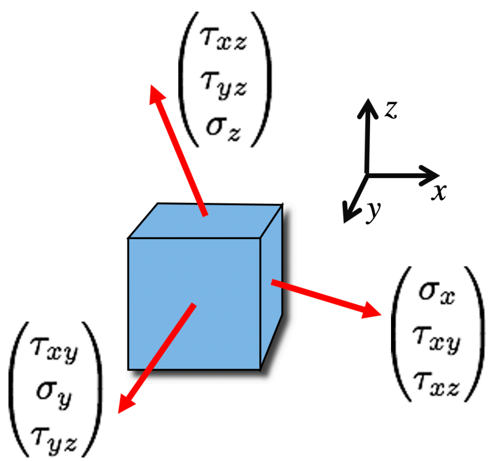
Physical Interpretation of Stress Components

- Cauchy stress tensor

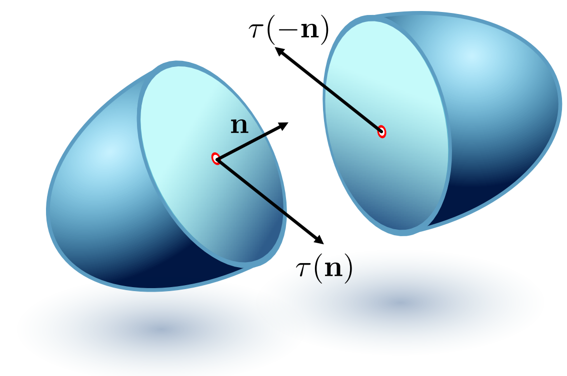
$$\sigma = [\tau(\mathbf{e}_1) \mid \tau(\mathbf{e}_2) \mid \tau(\mathbf{e}_3)] = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix}$$

  - normal stress components σ_i
  - shear stress components τ_{ij}

- Entries of σ give the traction acting on faces of a unit cube:



Stress from Hyperelastic Material Model



- Force acting per unit area on a slicing plane in the *deformed configuration*:

$$\tau(\mathbf{n}) = \sigma\mathbf{n},$$

  where σ is the *Cauchy Stress Tensor* (a symmetric matrix).

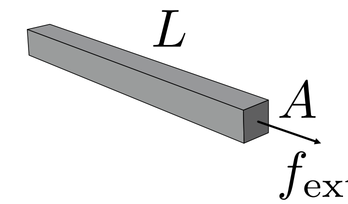
- How is stress related to energy density Ψ for a hyperelastic material?
  - One can show that:

$$\sigma = \Psi'(\nabla\Phi)\frac{\nabla\Phi^\top}{\det(\nabla\Phi)},$$

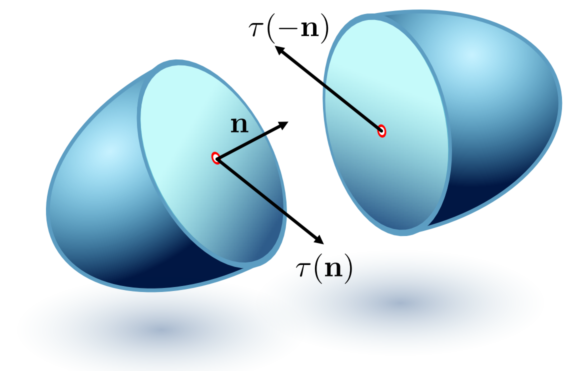
    where matrix $\Psi' = \frac{\partial\Psi(F)}{\partial F}$ is called the first Piola-Kirchhoff (PK1) stress.

- Physical interpretation: $\Psi'\,\mathbf{N}$ gives the force per unit *undeformed* area acting on plane described by normal $\mathbf{N}$ in the *undeformed* configuration. It is *asymmetric* in general.

Recap: Strain and Stress



- Uniaxial bars:
  - 1D *strain* $\varepsilon := \Delta L / L$ (relative length change)
  - 1D *stress* $\sigma := f_{\text{ext}} / A$ (applied force per unit area)



- 3D solids:
  - *Strain tensors* $\varepsilon_{\text{Biot}}$, $\varepsilon_{\text{Green}}$, *etc.*
    - $\mathbf{3} \times \mathbf{3}$ matrices encoding how the material is stretched along any direction.
    - *e.g.*, $\varepsilon_{\text{Green}} := \frac{1}{2}\left((\nabla\Phi)^\top \nabla\Phi - I\right)$ lets us compute squared length change in any direction $\mathbf{v}$ as $\mathbf{v}^\top \varepsilon_{\text{Green}} \mathbf{v}$.
  - *Stress tensors*: σ, Ψ', .
    - $\mathbf{3} \times \mathbf{3}$ matrices encoding the force per unit area (traction) acting on an oriented slicing plane.
    - *Cauchy stress tensor* σ gives the force per unit area (traction) acting on a plane with normal $\mathbf{n}$ in the *deformed configuration* as $\sigma\mathbf{n}$.
    - *1st Piola-Kirchoff stress* $\Psi' = \frac{\partial \Psi(F)}{\partial F}$ gives the force per unit area (traction) acting on a plane with normal $\mathbf{N}$ in the *reference configuration* as $\Psi'\mathbf{N}$.