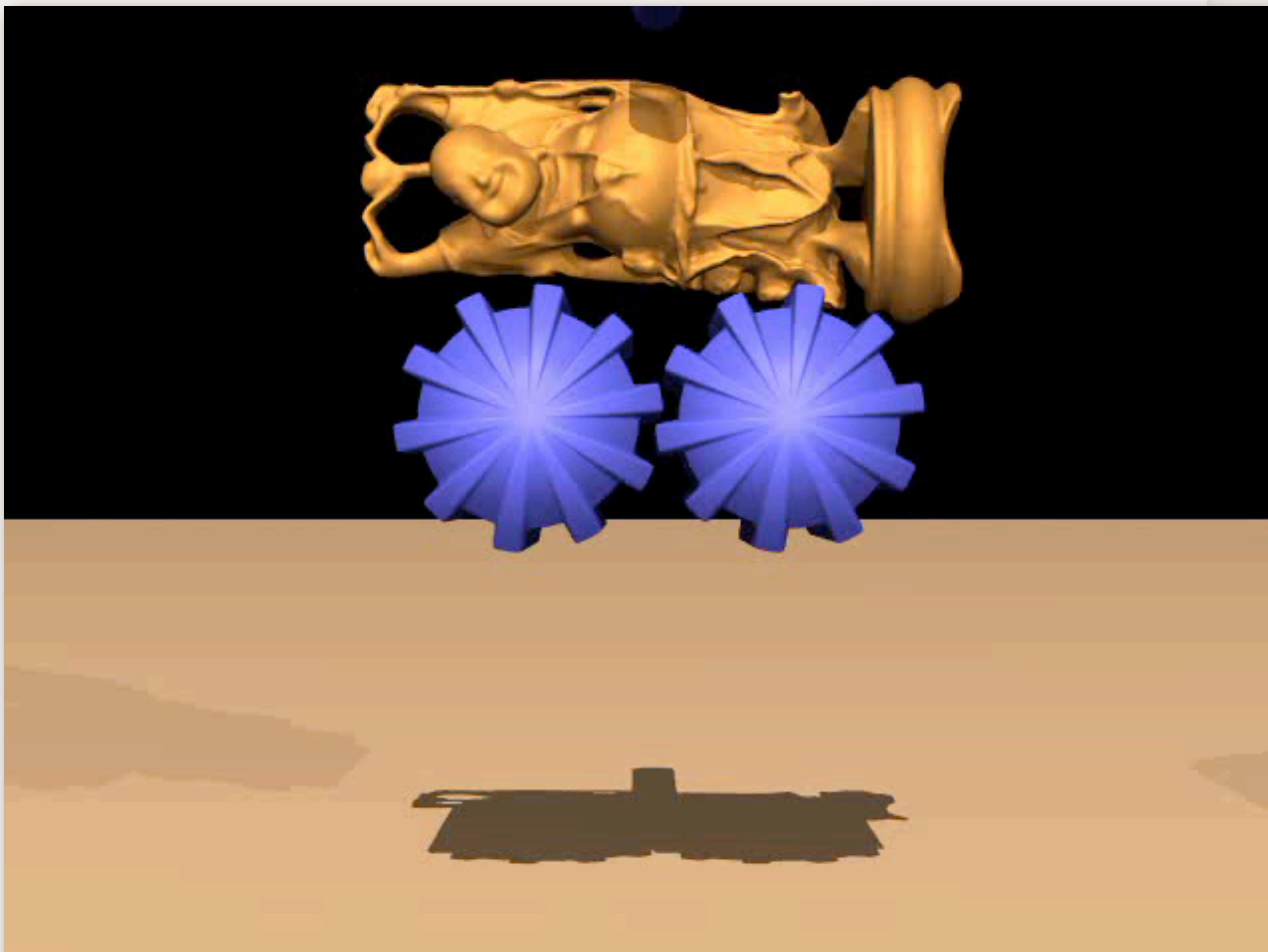
Deformable Materials

Adrien Treuille

Deformable Materials



Taking a Hard Look at Soft Things

Spring with rest length 1:

Deforms by Δx:

$$\text{energy} = E\frac{1}{2}(\Delta x)^2$$

$$\text{force} = -\frac{d\ \text{energy}}{dx}$$

$$\text{force} = -E\Delta x$$

Deformations

Spring deformed by Δx:

$$\text{force} = -E\Delta x$$

stress: σ — Young's modulus — strain: ε

Hooke's Law:

$$\sigma = -E\epsilon$$

Steel: $E = 10^{11}$ N/m²

Rubber: $E = 10^{7} \sim 10^{8}$ N/m²

Hooke's Law

$$\sigma = -E\epsilon$$

- Want to generalize in two ways:
  - Continuum Deformations
  - 3D

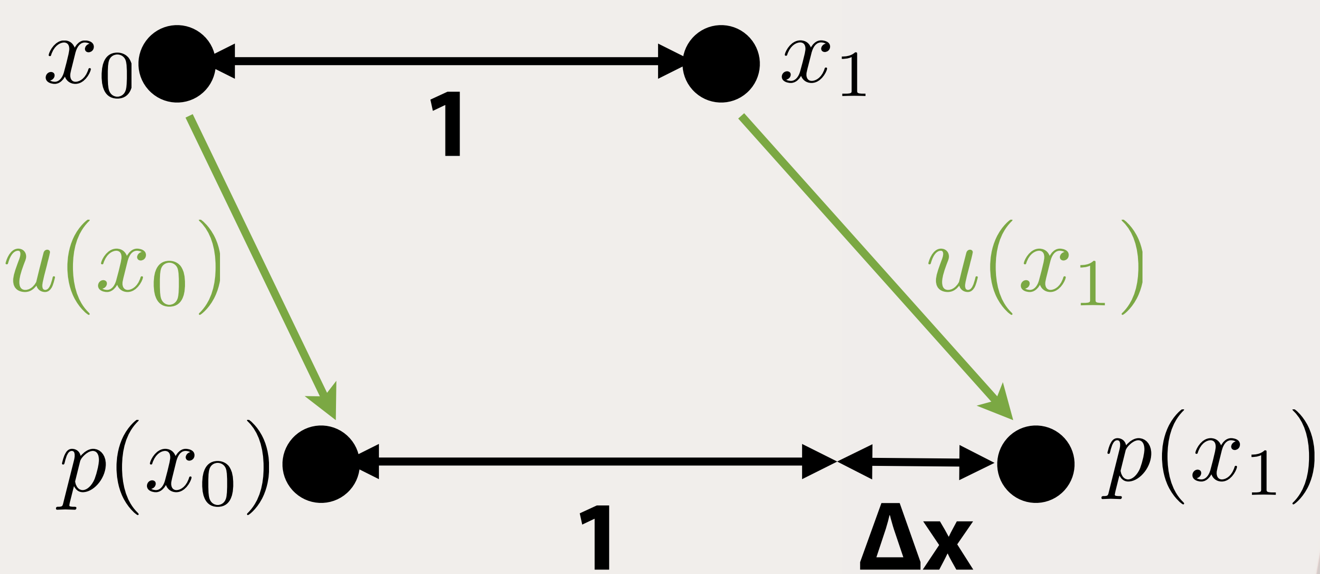
Hooke's Law

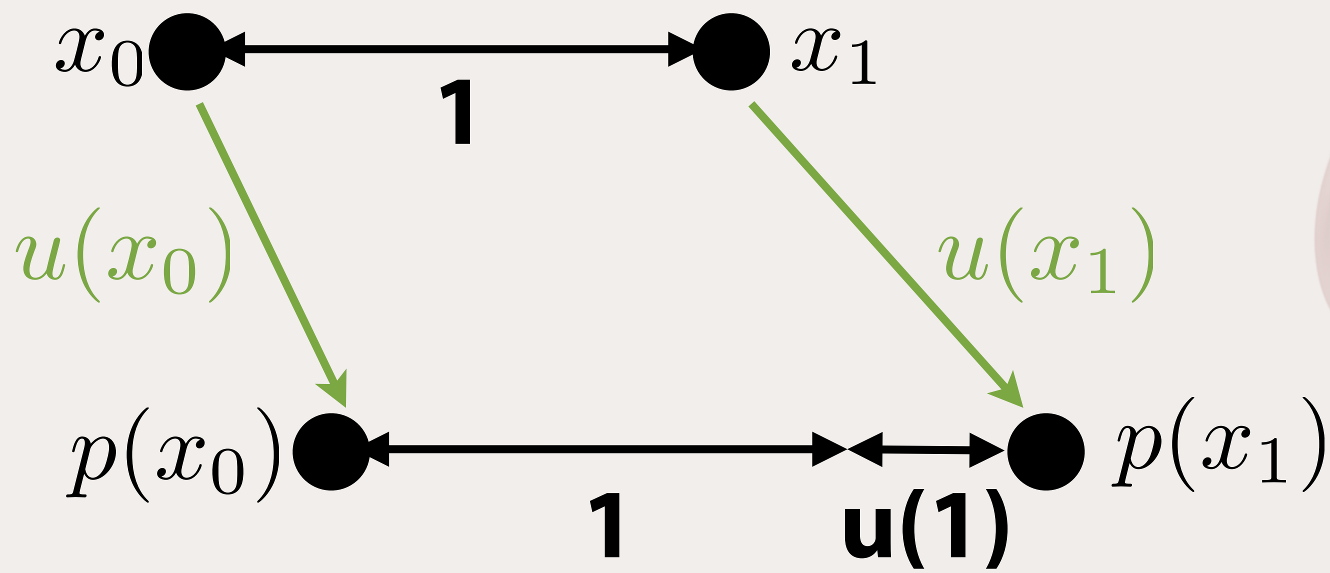
$$\sigma = -E\epsilon$$

- Want to generalize in two ways:
  - Continuum Deformations
  - 3D

Continuum Deformations

- **Given a displacement field *p*(*x*):**
- **Which defines a deformation field:**
  - ***u*(*x*) = *p*(*x*) - *x***
  - **(Like velocities in a fluid.)**





- **Suppose:** $x_0 = 0$ **and** $x_1 = 1$
- $p(0) = u(0)$
- $p(1) = 1 + u(1)$
- **energy** $= \frac{1}{2} E\,(p(1) - p(0) - 1)^2$
- $p(1) \approx 1 + u(0) + \nabla u(0)$
- **energy** $\approx \frac{1}{2} E\,(1 + u(0) + \nabla u(0) - u(0) - 1)^2$
- **energy** $\approx \frac{1}{2}\,\mathrm{E}\,\nabla u^2$
- **force** $= -\mathrm{E}\nabla u$

Therefore...

- force = -$E \nabla u$

$$\sigma = -E\epsilon$$

In 1D, $\varepsilon = \nabla u$.

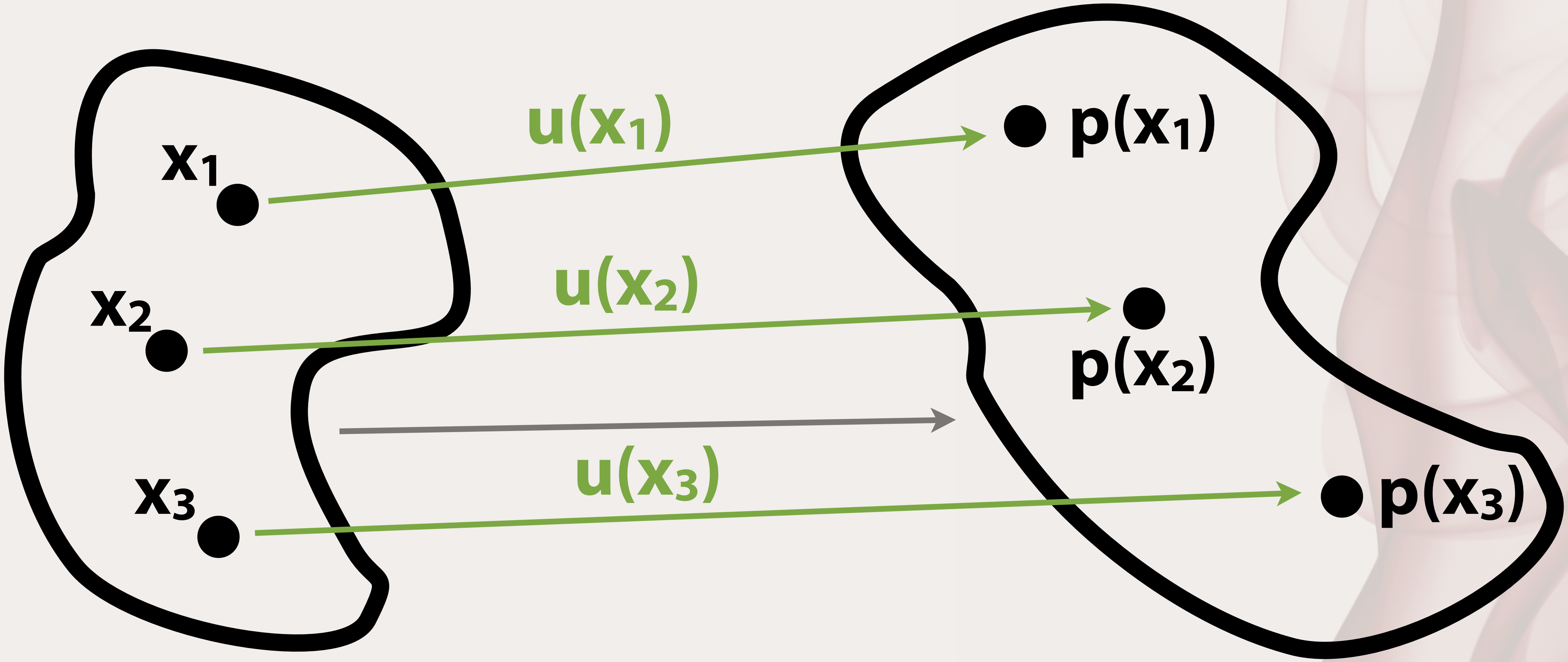
(Would like to generalize to 3D.)

Hooke's Law

$$\sigma = -E\epsilon$$

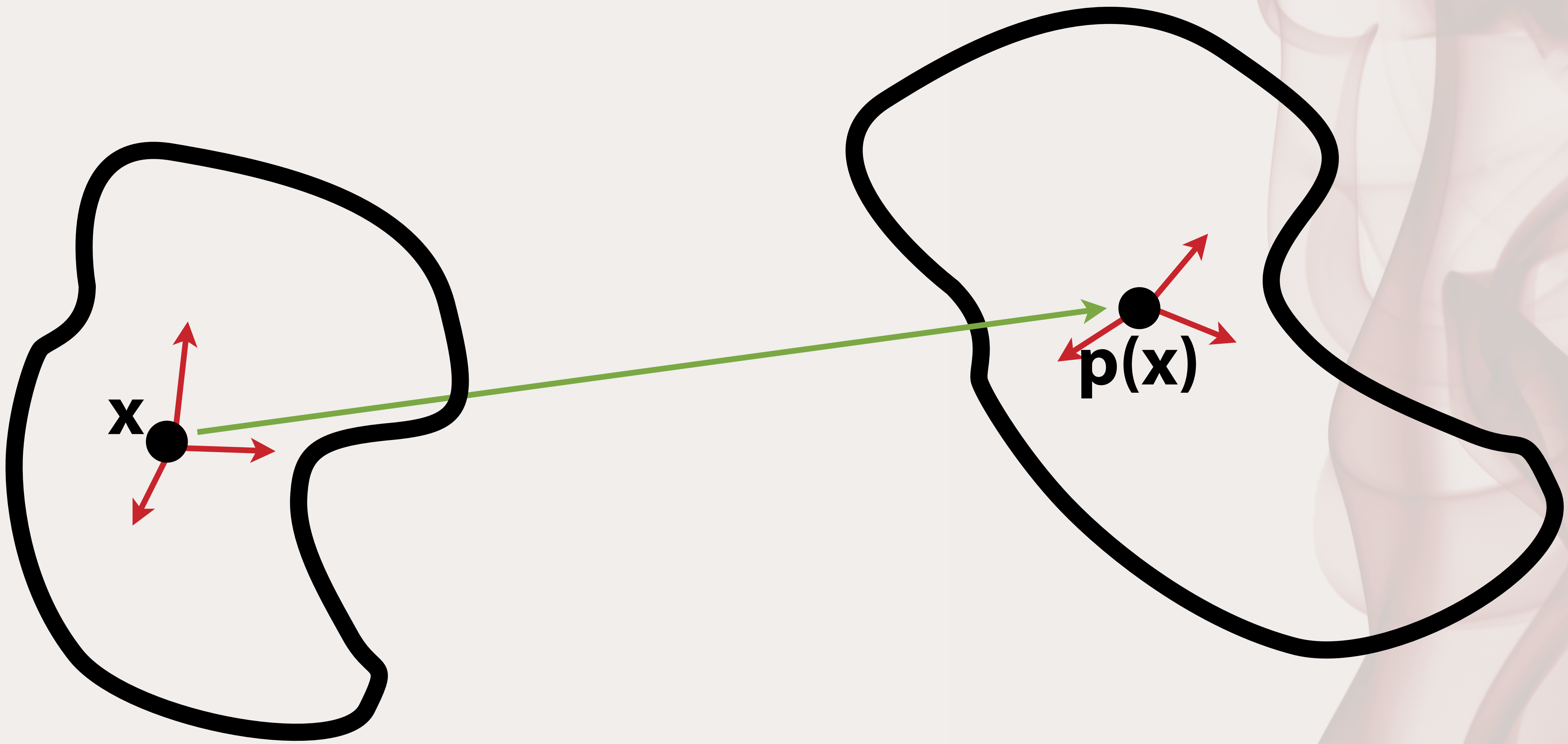
- Want to generalize in two ways:
  - Continuum Deformations
  - **3D (things will get a little silly)**

In 3D...



$$u(x) = p(x) - x$$

Coord Sys Transform

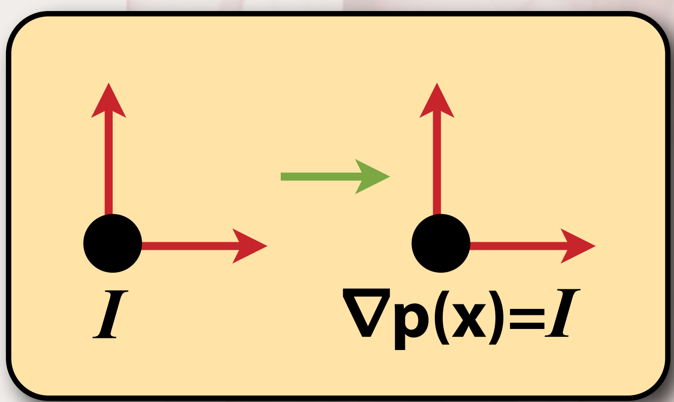


point: x
local coordinates: I

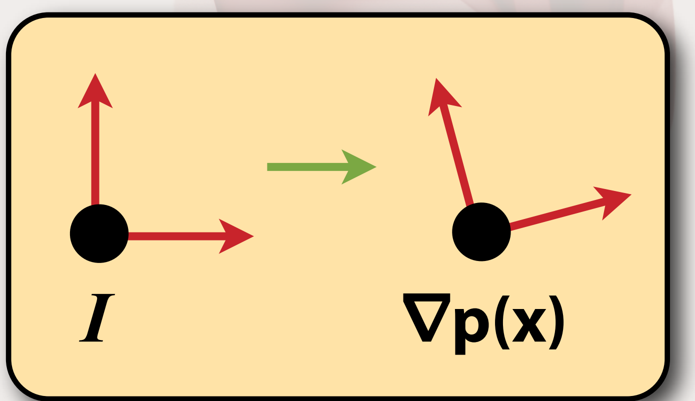
point: $p(x)$
local coordinates: ∇p

Defining Strain

- **Strain is invariant to translation.**
  - Ignore $p(x)$
  - Define in terms of local coordinate system transform: $\nabla p(x)$.



- **Strain is invariant to rotation.**
  - If $[\nabla p(x)]^T \nabla p(x) = I$,
  - Then $\varepsilon = 0$



- **Natural to define strain as:**
  - $\varepsilon = \frac{1}{2}([\nabla p(x)]^T \nabla p(x) - I)$
  - 6 DOFs

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{xy} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{xz} & \epsilon_{yz} & \epsilon_{zz} \end{bmatrix}$$

Defining Strain

$$\epsilon = \frac{1}{2}\left[\nabla p(x)\right]^T \nabla p(x) - I$$

$$u(x) = p(x) - x$$

$$\nabla u(x) = \nabla p(x) - I$$

$$\epsilon = \frac{1}{2}\left[\nabla u(x) + I\right]^T \left[\nabla u(x) + I\right] - I$$

Green's Strain: $$\epsilon_G = \frac{1}{2}\left(\nabla u + [\nabla u]^T + [\nabla u]^T \nabla u\right)$$

Cauchy's Strain: $$\epsilon_C = \frac{1}{2}\left(\nabla u + [\nabla u]^T\right)$$ **(no rotation)**

1D Strain: $$\epsilon_{1D} = \nabla u$$

Defining Strain

$$\epsilon = \frac{1}{2}\left[\nabla p(x)\right]^T \nabla p(x) - I$$

$$u(x) = p(x) - x$$

$$\nabla u(x) = \nabla p(x) - I$$

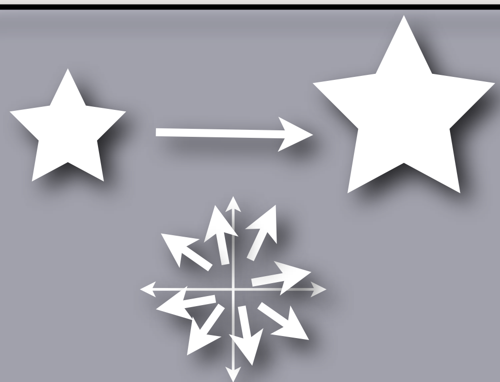
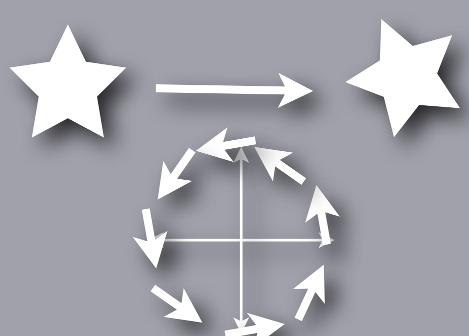
$$\epsilon = \frac{1}{2}\left[\nabla u(x) + I\right]^T \left[\nabla u(x) + I\right] - I$$

Green's Strain: $$\epsilon_G = \frac{1}{2}\left(\nabla u + [\nabla u]^T + [\nabla u]^T \nabla u\right)$$

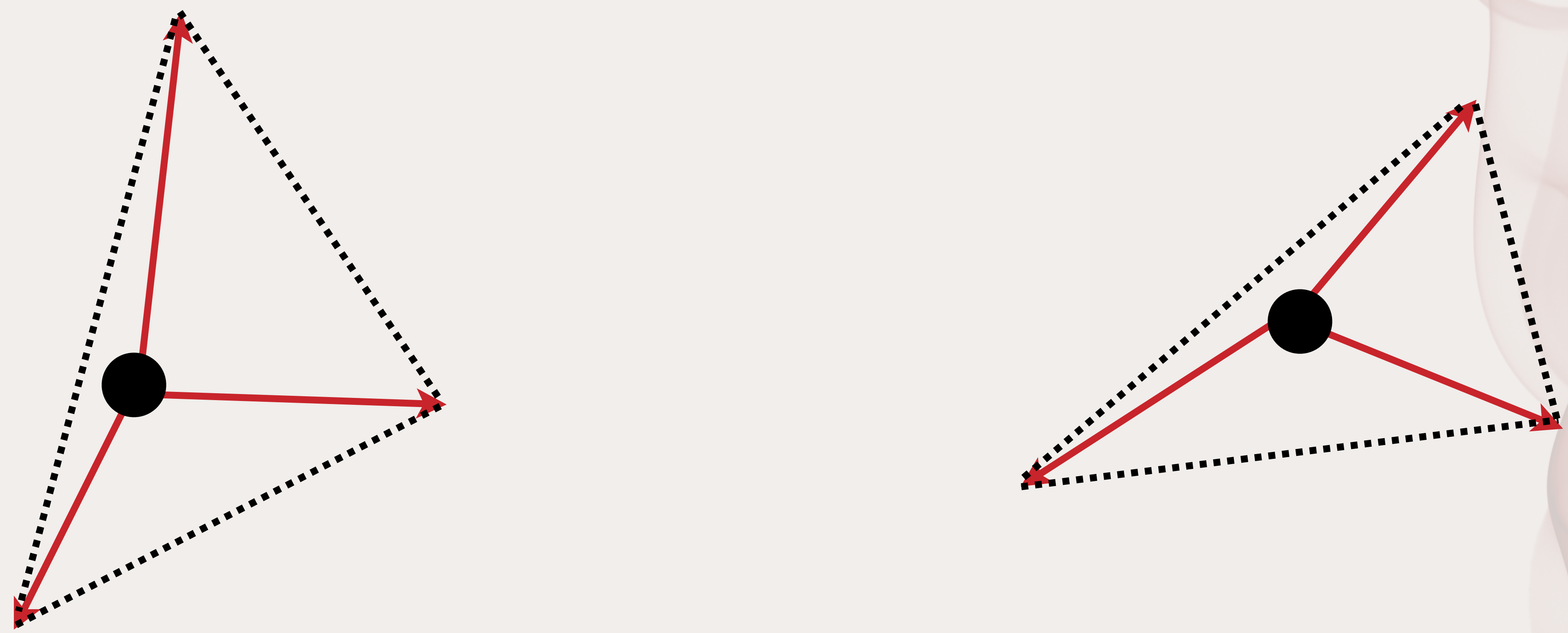
Cauchy's Strain: $$\epsilon_C = \frac{1}{2}\left(\nabla u + [\nabla u]^T\right)$$ **(no rotation)**

1D Strain: $$\epsilon_{1D} = \nabla u$$

Cauchy's vs. Green's Strain

	Cauchy's Strain	**Green's Strain**
	$\epsilon_C = \frac{1}{2}\left(\nabla u + [\nabla u]^T\right)$	$\epsilon_G = \frac{1}{2}\left(\nabla u + [\nabla u]^T + [\nabla u]^T \nabla u\right)$
 $u(x) = x$ $\nabla u(x) = I$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\frac{3}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
 $\nabla u = \begin{bmatrix} -1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ ✖	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Question



Question

- How could we reduce the cost of simulation for a very finely discretized surface?
- Are there cheap ways of getting volumetric behavior without a full tetrahedralization?
- How can collision constraints be integrated?
- How to simulate plasticity?

Student Answers

- **How could we reduce the cost of simulation for a very finely discretized surface?**

  - make a more coarse tetrahedral mesh, and then track the barycentric coordinates of the original mesh points
    - then, just simulate the coarser mesh!
    - how do we get the coarser mesh:
      - for example, in planar (or low curvature) regions, use larger elements
      - how can we "coalesce" tetrahedra into larger elements
      - what other principles are there?

  - if we can precompute (or if we just know) how some object reacts to forces
    - we can use this _proxy_ instead of the real thing!

- **Are there cheap ways of getting volumetric behavior without a full tetrahedralization?**

  - use springs inside the mesh instead of doing the whole stress/strain thing
  - forget the interior/ only simulate on the surface!!!
    - ...and maybe we can even simulate the surface too
    - maybe we can use angular springs

  - build a "tree" of the structure : the mesh refines itself in areas of deformation

- **How can collision constraints be integrated?**

  - is there a local way (looking at deformation gradients) to detect self collisions?
  - apply forces to points on the mesh (rather than the whole object itself)
  -

- **How to simulate plasticity?**

  - non monotonically increasing stress-strain relationship!
    - implementation: if it stresses (gradient of the displacement) too much, stop applying forces!