CSE 291 (SP24) Physical Simulation Elasticity: Part 2

Albert Chern

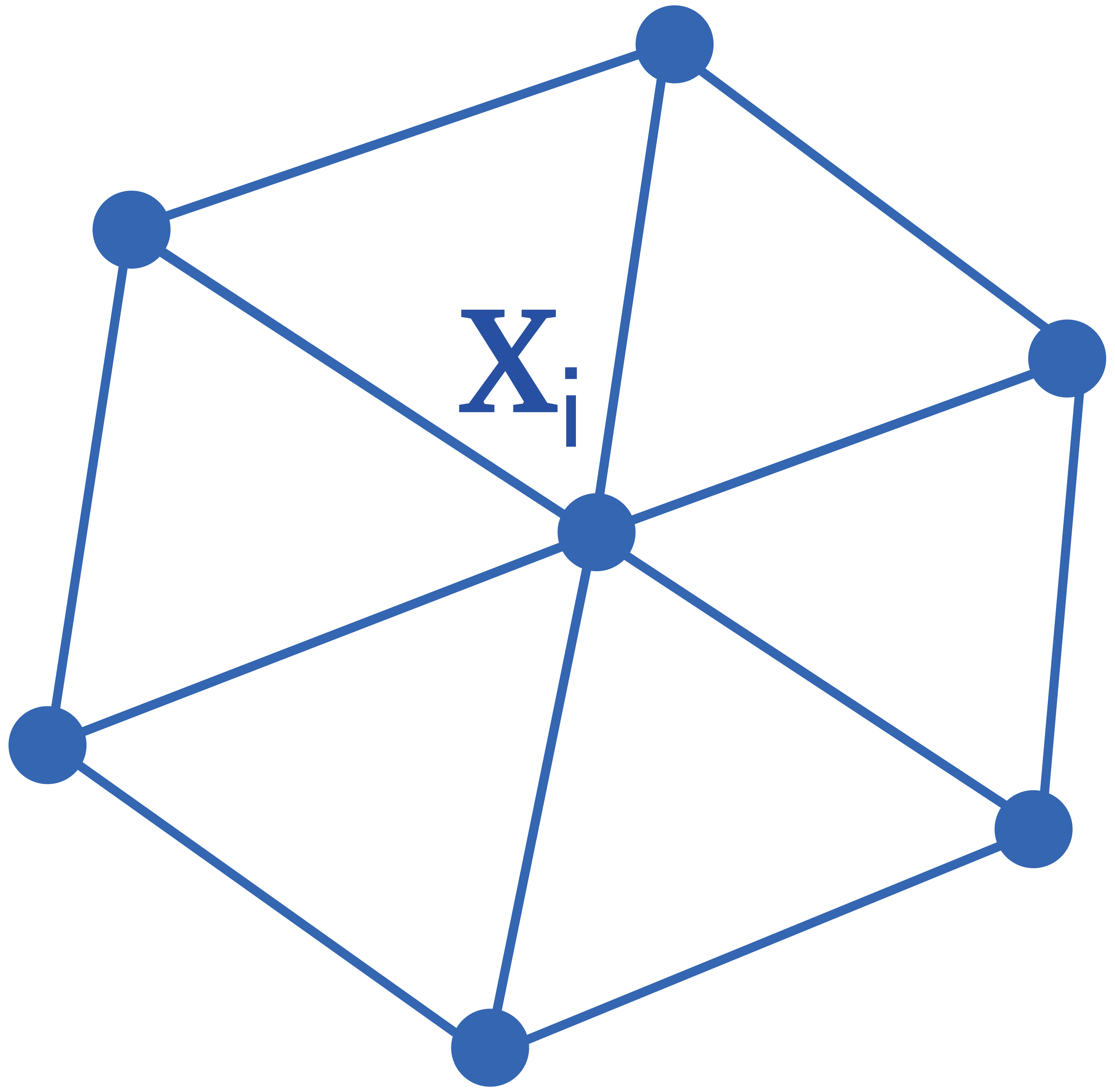
UC San Diego
JACOBS SCHOOL OF ENGINEERING

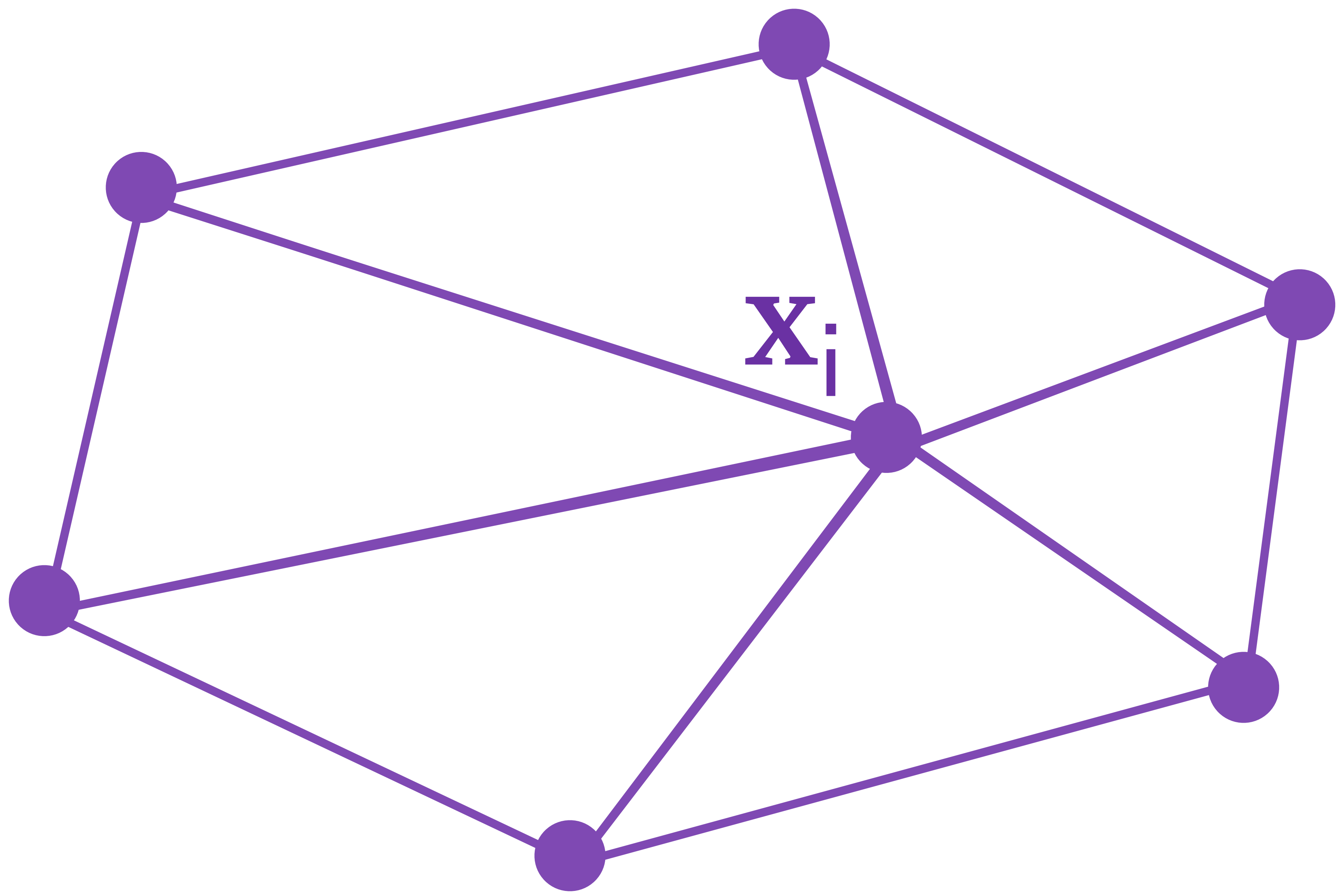
Finite Element Elasticity

- Finite element elasticity
- More on Stress–Strain relation

Finite element simulation

- Discretize the deformable body by triangle mesh (2D) or tetrahedral mesh (3D) (n = dim(M))
- Each vertex i stores a fixed rest position $\mathbf{X}_i$ (material coordinate) and a variable world position $\mathbf{x}_i$ (representing value of flow map)

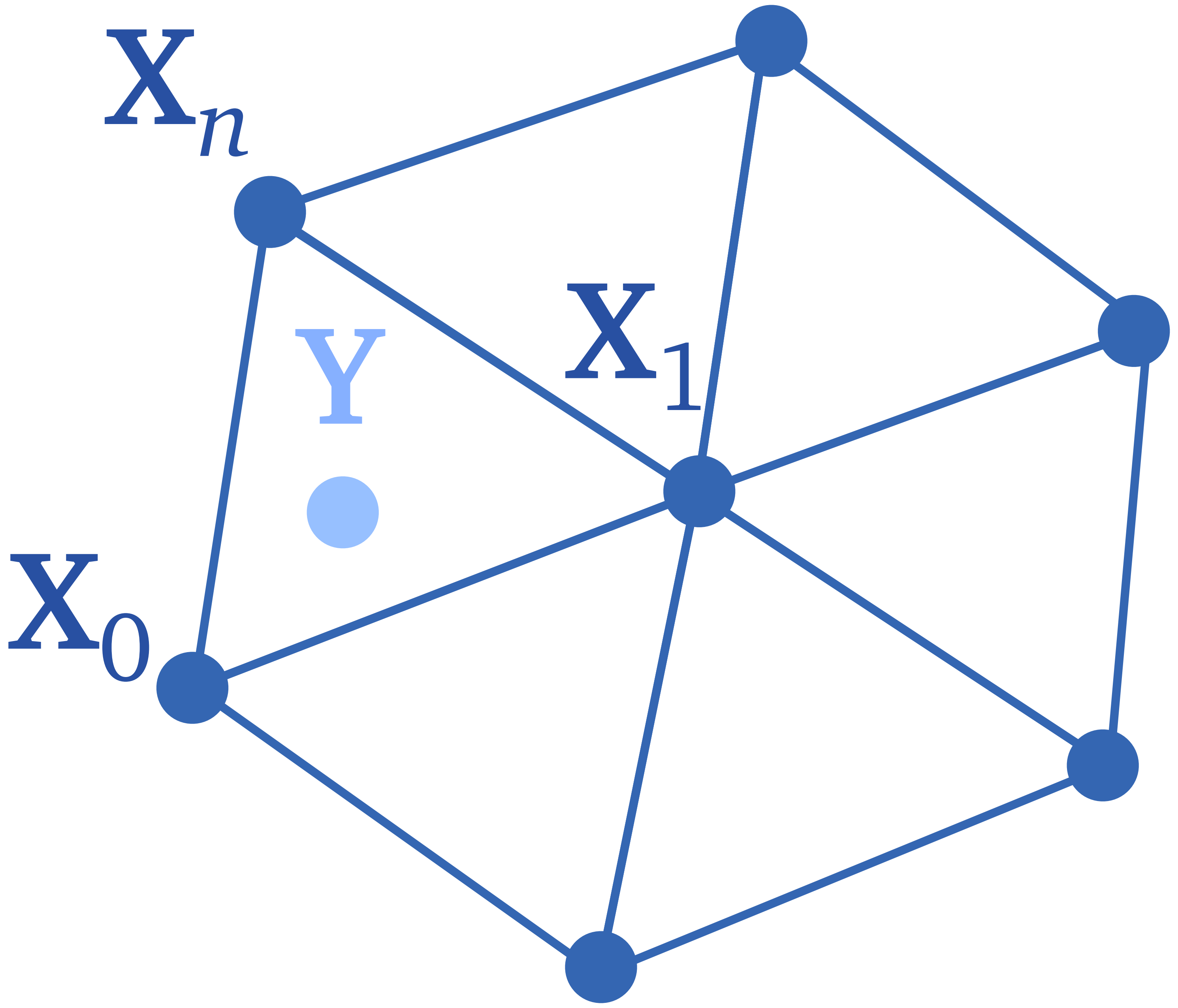


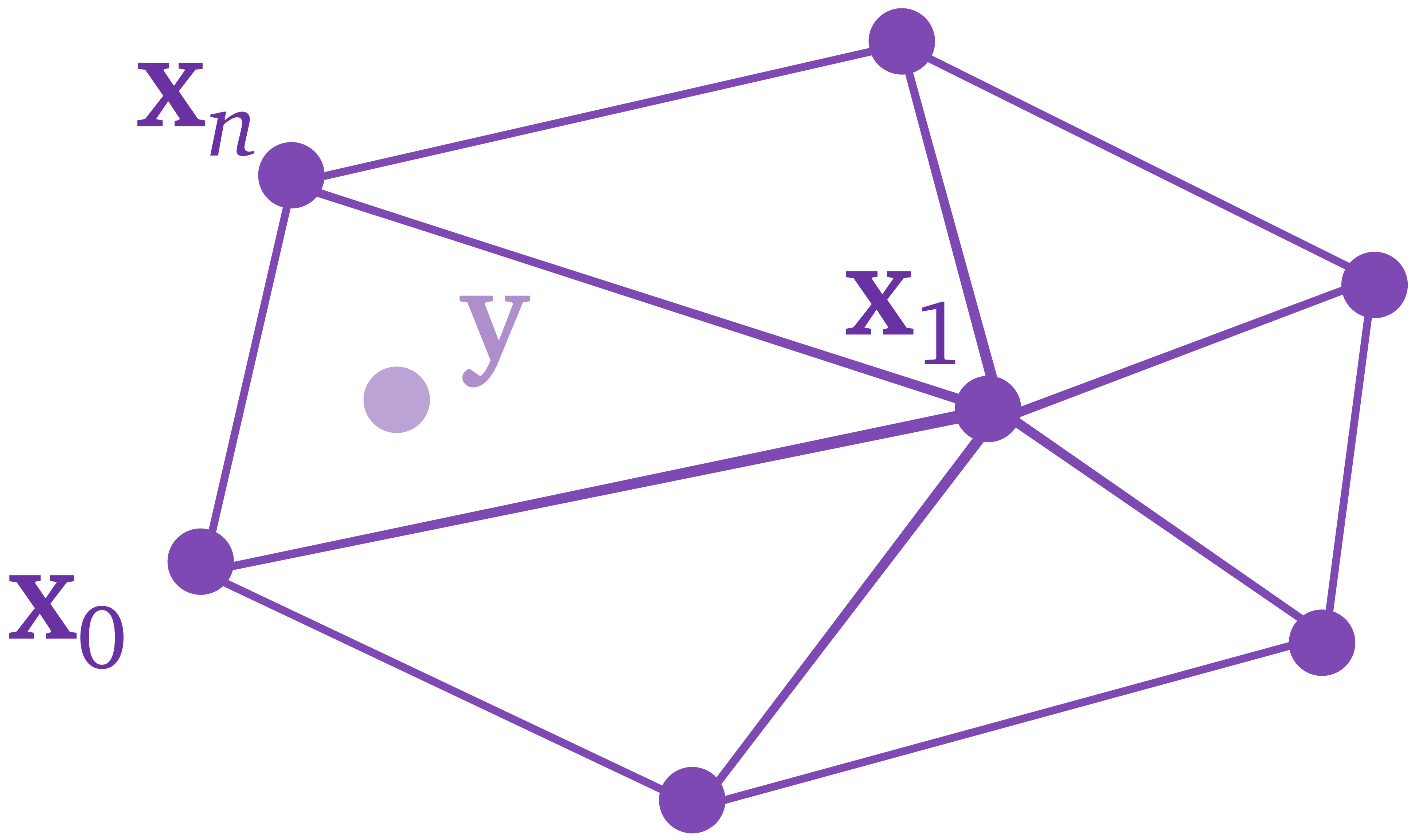


Linear interpolation

- The data on the vertices can be linearly interpolated into a piecewise linear flow map.

$$\begin{bmatrix} | \\ \mathbf{y} \\ | \\ 1 \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix}^{-1} \begin{bmatrix} | \\ \mathbf{Y} \\ | \\ 1 \end{bmatrix}$$





Deformation gradient

- The data on the vertices can be linearly interpolated into a piecewise linear flow map.

$$\begin{bmatrix} | \\ \mathbf{y} \\ | \\ 1 \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix}^{-1} \begin{bmatrix} | \\ \mathbf{Y} \\ | \\ 1 \end{bmatrix}$$

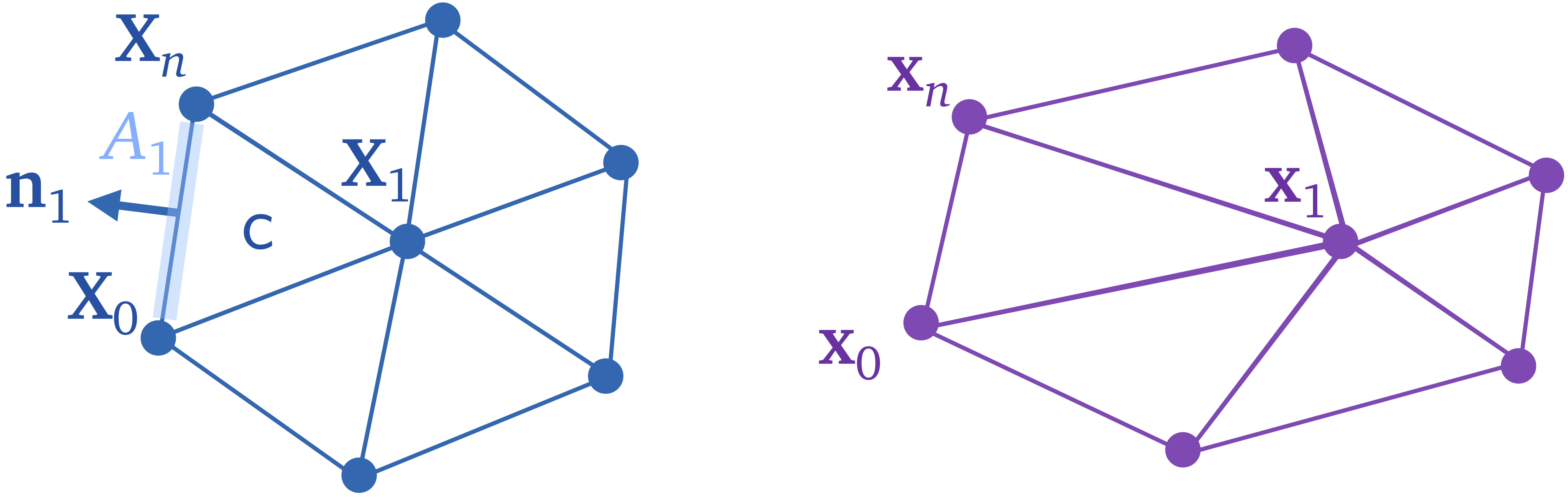
- The deformation gradient is a piecewise constant matrix

$$\begin{bmatrix} & & & 0 \\ & \mathbf{F} & & | \\ & & & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_n \\ | & | & & | \\ 1 & 1 & \cdots & 1 \end{bmatrix}^{-1}$$

Deformation gradient

- If $A_j \mathbf{n}_j$ is the area normal of the opposite face of *j*-th vertex and V is the volume of the cell. Then

$$\mathbf{F}_{\mathrm{c}} = -\frac{1}{nV_{\mathrm{c}}} \sum_{j=0}^{n} \begin{bmatrix} | \\ \mathbf{x}_j \\ | \end{bmatrix} \begin{bmatrix} - & A_{\mathrm{c},j}\mathbf{n}_{\mathrm{c},j}{}^{\intercal} & - \end{bmatrix}$$



Strain and stress computation

- Now in each cell we have deformation gradient $\mathbf{F}$
- We can compute the Cauchy–Green tensor per cell $\mathbf{C} = \mathbf{F}^{\mathsf{T}}\mathbf{F}$
- Like the smooth theory, build $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$
- Look up some stress–strain relation

$$\mathbf{S} = 2\mu\mathbf{E} + \lambda\,\mathrm{tr}(\mathbf{E})\mathbf{I}$$

$\mathbf{x}$ per vtx $\longrightarrow$ $\mathbf{F}$ per cell $\longrightarrow$ $\mathbf{C}$ per cell

$\vdots$

$\mathbf{S}$ per cell

Stress computation

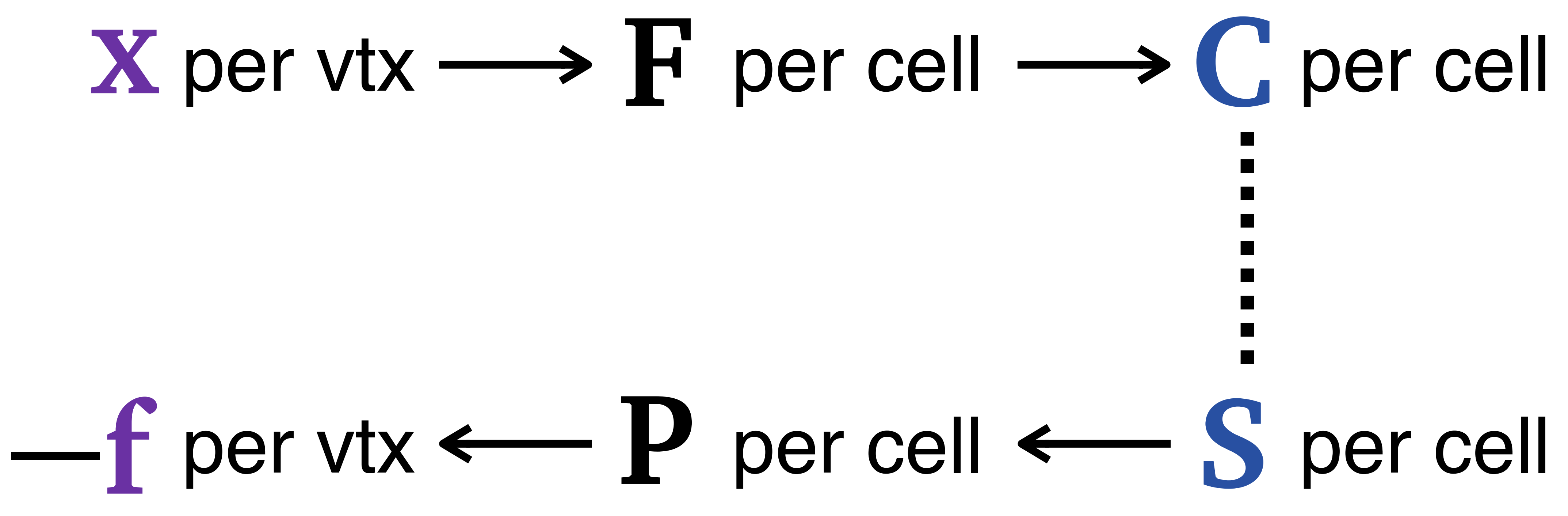
- Now in each cell we have deformation gradient $\mathbf{F}$
- We can compute the Cauchy–Green tensor per cell $\mathbf{C} = \mathbf{F}^{\mathsf{T}}\mathbf{F}$
- Like the smooth theory, build $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$
- Look up some stress–strain relation

$$\mathbf{S} = 2\mu\mathbf{E} + \lambda\,\mathrm{tr}(\mathbf{E})\mathbf{I}$$

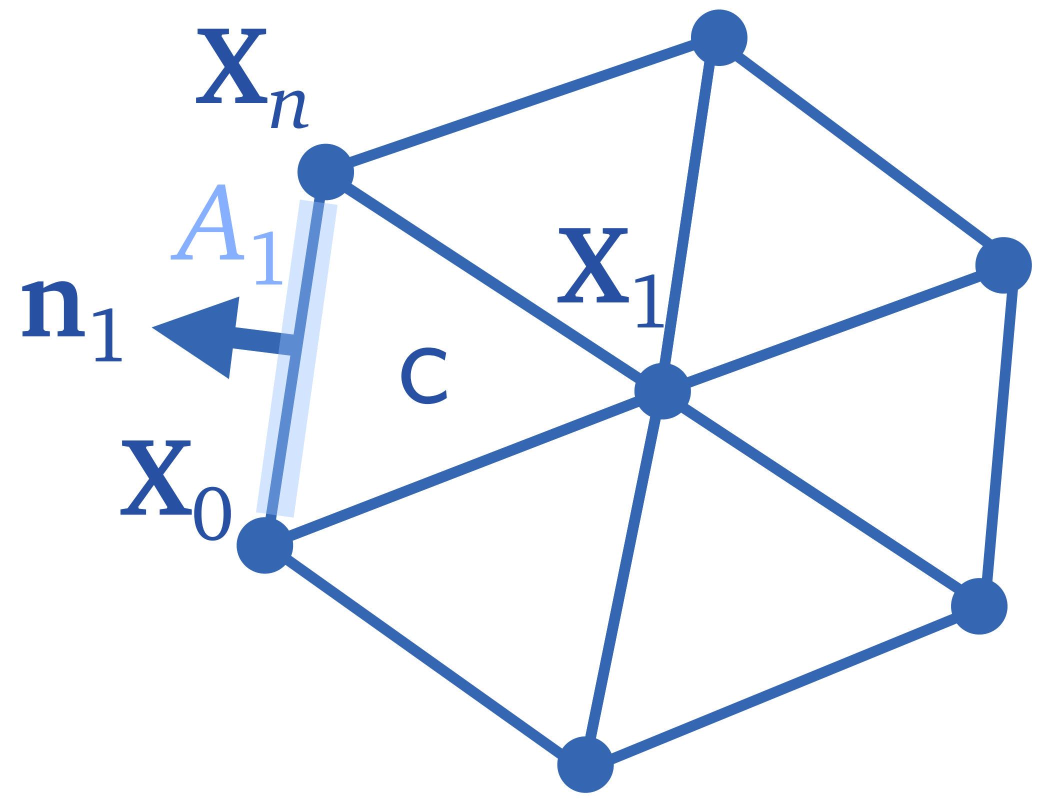
- 1st-Piola stress

$$\mathbf{P} = \mathbf{F}\mathbf{S}$$

- Compute force by taking adjoint of gradient

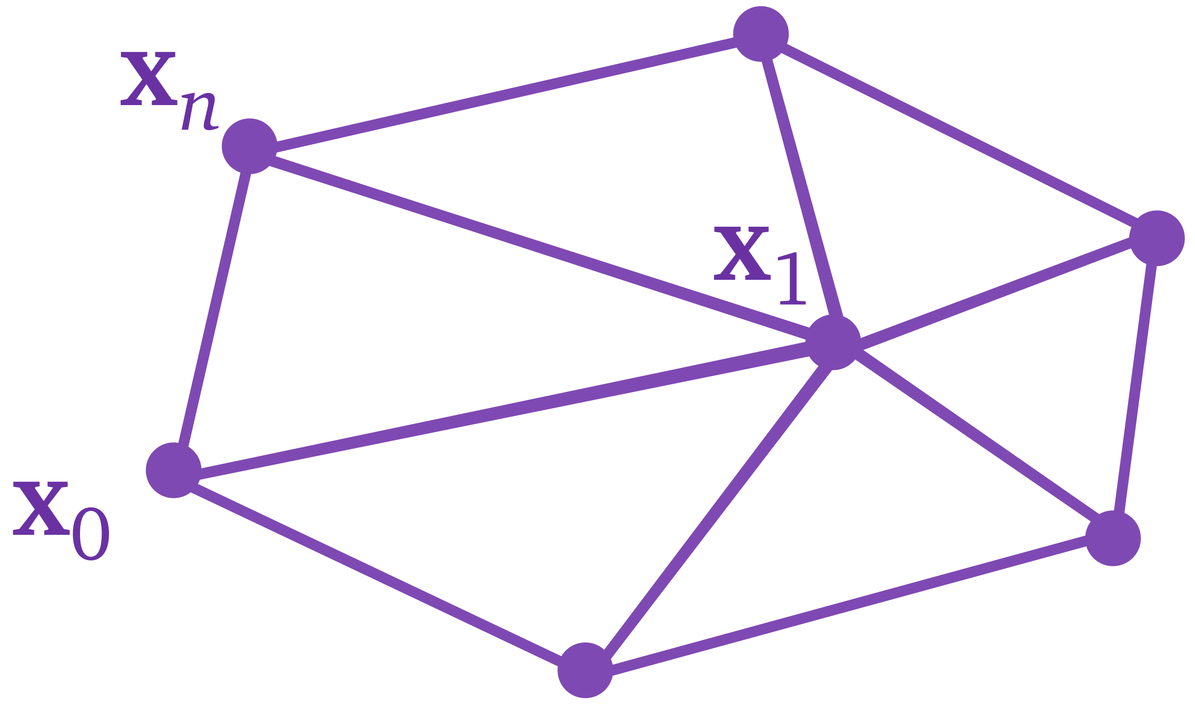


Total force computation



- The differential of $\mathbf{F}$ with respect to $\mathbf{x}$

$$\mathring{\mathbf{F}}_c = -\frac{1}{nV_c} \sum_{v \prec c} \begin{bmatrix} | \\ \mathring{\mathbf{x}}_v \\ | \end{bmatrix} \begin{bmatrix} - & A_{c,v} \mathbf{n}_{c,v}^\intercal & - \end{bmatrix}$$



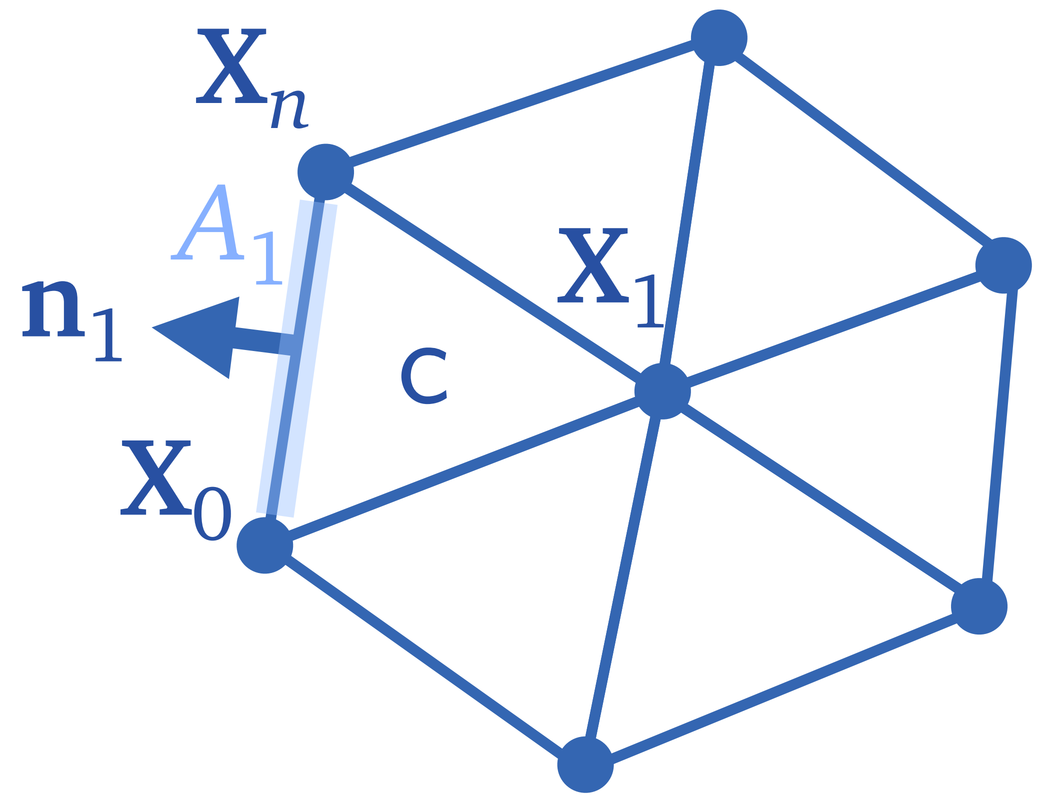
- Adjoint: accumulate traction force to vertex

$$\sum_c \mathrm{tr}(\mathbf{P}_c \mathring{\mathbf{F}}_c^\intercal) V_c = \sum_v -\mathbf{f}_v^\intercal \mathring{\mathbf{x}}_v$$

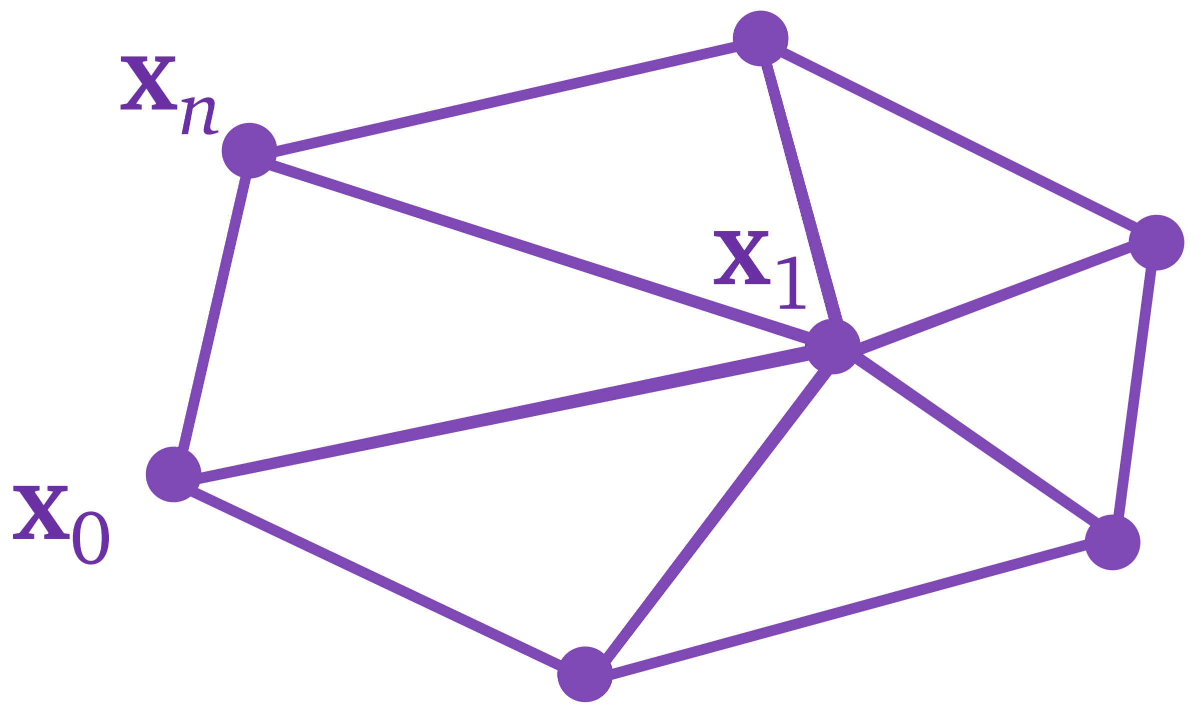
$$\mathbf{f}_v = \frac{1}{n} \sum_{c \succ v} \mathbf{P}_c \mathbf{n}_{c,v} A_{c,v}$$

Equation of motion

$$\mathbf{f}_{\mathsf{v}} = \frac{1}{n}\sum_{\mathsf{c}\succ\mathsf{v}} \mathbf{P}_{\mathsf{c}}\mathbf{n}_{\mathsf{c},\mathsf{v}}A_{\mathsf{c},\mathsf{v}}$$



$$m_{\mathsf{v}}\ddot{\mathbf{x}}_{\mathsf{v}} = \mathbf{f}_{\mathsf{v}}$$

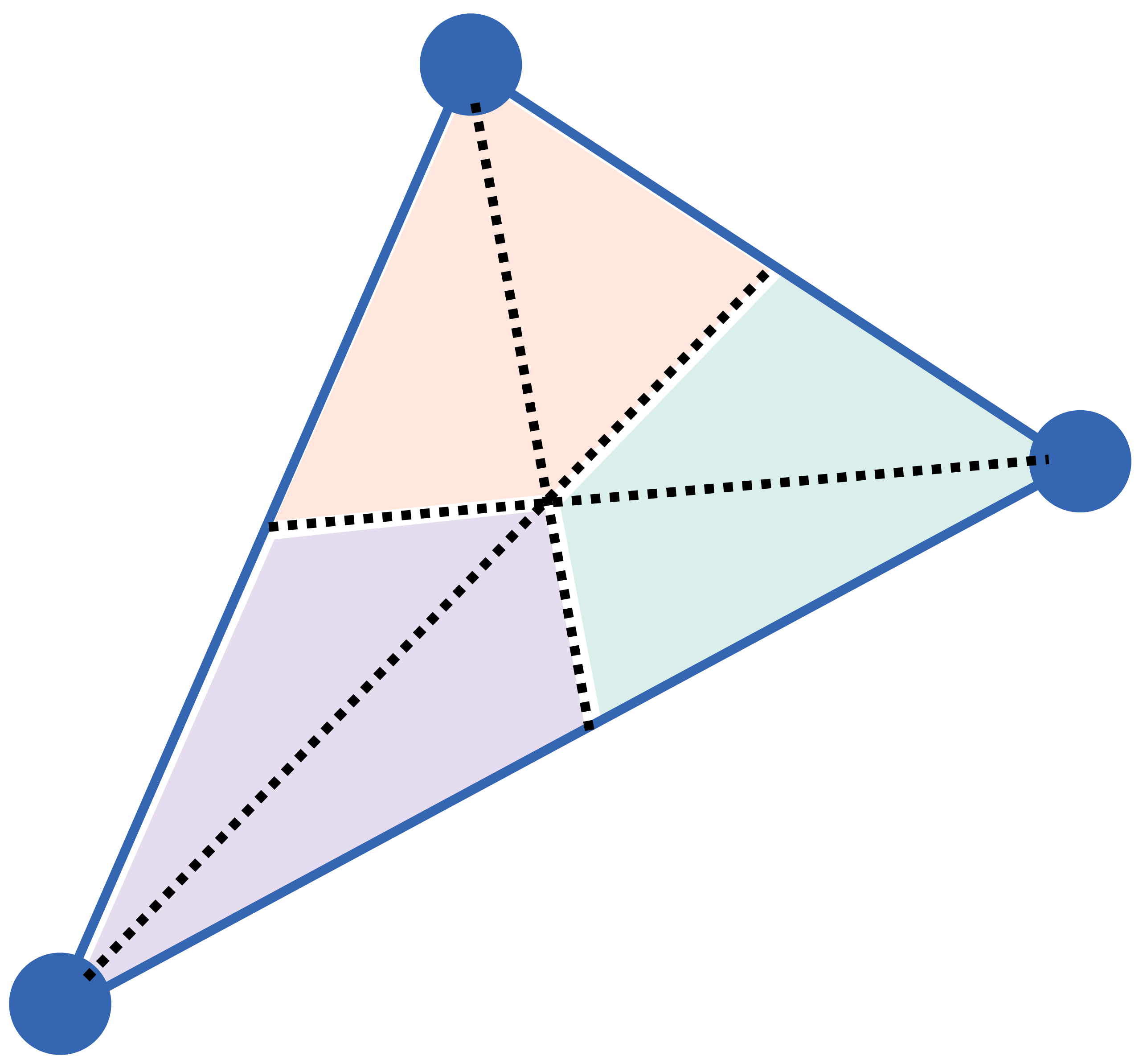


Mass computation

- The total mass of each vertex should be proportional to the vertex volume approximated by

$$m_{\mathrm{v}} = \sum_{\mathrm{c} \succ \mathrm{v}} \frac{1}{n+1} V_{\mathrm{c}}$$

- This is called lumped mass



Time integration

$$m_v \ddot{\mathbf{x}}_v = \mathbf{f}_v + \mathbf{f}_v^{\text{ext}}$$

- RK4 or Symplectic Euler method
  - Just need to evaluate force $(\mathbf{f}_v)_v$ given current position $(\mathbf{x}_v)_v$
  - Stepsize $\Delta t = O(\text{edge lengths})$

- Implicit Euler (with incremental potential): stable

$$\mathbf{x}^{(n+1)} = \underset{\mathbf{x}\in\mathbb{R}^m}{\operatorname{argmin}} \sum_v \frac{m_v}{2\Delta t^2} |\mathbf{x}_v - \mathbf{x}_v^{\text{pred}}|^2 + \mathcal{U}(\mathbf{x})$$

Time integration

- Implicit Euler (with incremental potential): stable

$$\mathbf{x}^{(n+1)} = \underset{\mathbf{x}\in\mathbb{R}^m}{\operatorname{argmin}} \sum_{\mathrm{v}} \frac{m_{\mathrm{v}}}{2\Delta t^2} |\mathbf{x}_{\mathrm{v}} - \mathbf{x}_{\mathrm{v}}^{\mathrm{pred}}|^2 + \mathcal{U}(\mathbf{x})$$

  - For gradient descent (or Newton) method with line search
  - Need evaluation of $\mathcal{U}(\mathbf{x}) = \sum_{\mathrm{c}} U(\mathbf{F}^{\mathsf{T}}\mathbf{F})\, V_{\mathrm{c}}$
  - Need evaluation of differential of potential (same as force evaluation)
  - Need an (approximated) Hessian for the potential

Time integration

- Need an (approximated) Hessian for the potential

$$\mathbf{x} \text{ per vtx} \xrightarrow{[\mathrm{grad}]} \mathbf{F} \text{ per cell} \xrightarrow{\mathbf{F}^\mathsf{T}(\cdot)+(\cdot)^\mathsf{T}\mathbf{F}} \mathbf{C} \text{ per cell}$$

$$-\mathbf{f} \text{ per vtx} \xleftarrow{[\mathrm{div}]} \mathbf{P} \text{ per cell} \xleftarrow{\mathbf{F}(\cdot)} \mathbf{S} \text{ per cell}$$

Laplacian

$$\mathbf{L} = \begin{bmatrix} & -\mathrm{div} & \end{bmatrix} \begin{bmatrix} V_c & & \\ & \ddots & \\ & & \ddots \end{bmatrix} \begin{bmatrix} & \mathrm{grad} & \end{bmatrix}$$

can serve as an approximated Hessian

- True Hessian: replace the central term by the cell-wise Hessian of the energy

More on Stress–Strain relation

- Finite element elasticity
- More on Stress–Strain relation

Designing potential energy

- Given the Cauchy–Green $\mathbf{C} = \mathbf{F}^\intercal \mathbf{F}$

$$\sharp_M C = \sharp_M F^* \flat_W F \in \Gamma(\mathrm{End}(TM))$$

  - (as endomorphism that measures the deviation between induced metric from the world and the pre-defined material metric)

- Design a potential energy function $U(\mathbf{C})$
  - Note that it's a function on symmetric matrices
  - The energy is said to be *isotropic* if

$$U(\mathbf{C}) = U(\mathbf{R}^\intercal \mathbf{C} \mathbf{R})$$

for rotation matrices $\mathbf{R}$ $\qquad R^* \flat_M R = \flat_M$

Designing potential energy

- If the material is isotropic $U(\mathbf{C}) = U(\mathbf{R}^{\mathsf{T}}\mathbf{C}\mathbf{R})$

  then the energy is only a function of the eigenvalues (modulo permutation)

$$\text{eigenvalues}(\mathbf{C}) = \{\lambda_1, \lambda_2, \lambda_3\}$$

- By the way, these eigenvalues are the square of the eigenvalues of **Y** in polar decomposition $\mathbf{F} = \mathbf{R}\mathbf{Y}$; equivalently, square of singular values of **F**. They are the square of principal stretching.

Designing potential energy

- Can we model *U* like $U(\mathbf{C}) = u(\lambda_1, \lambda_2, \lambda_3)$?
  - Generally this wouldn't respect symmetry under label permutation
  - View the eigenvalues as the roots of a polynomial, and use the coefficient of the polynomial as our parameters

$$\{\lambda_1, \lambda_2, \lambda_3\} = \mathrm{roots}(t^3 - I_1 t^2 + I_2 t - I_3; t)$$

  - These coefficients are called the "invariants":

$$\begin{aligned} I_1 &= \lambda_1 + \lambda_2 + \lambda_3 &&= \mathrm{tr}(\mathbf{C}) \\ I_2 &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 &&= \tfrac{1}{2}(\mathrm{tr}(\mathbf{C})^2 - \mathrm{tr}(\mathbf{C}^2)) \\ I_3 &= \lambda_1\lambda_2\lambda_3 &&= \det(\mathbf{C}) \end{aligned}$$

  - Characteristic polynomial $t^3 - I_1 t^2 + I_2 t - I_3 = \det(t\mathbf{I} - \mathbf{C})$

Designing potential energy

$$I_1 = \lambda_1 + \lambda_2 + \lambda_3 = \mathrm{tr}(\mathbf{C})$$
$$I_2 = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 = \tfrac{1}{2}(\mathrm{tr}(\mathbf{C})^2 - \mathrm{tr}(\mathbf{C}^2))$$
$$I_3 = \lambda_1\lambda_2\lambda_3 = \det(\mathbf{C})$$

- We model $U(\mathbf{C}) = w(I_1, I_2, I_3)$
  - How do you do chain rule? (Blackboard)
- For example neo-Hookean model

how much material respond to 1D stretch and volume change

$$w(I_1, I_2, I_3) = \tfrac{\mu}{2}(I_1 - 3 - \ln I_3) + \tfrac{\lambda}{2}(\sqrt{I_3} - 1)^2$$

- Approximately

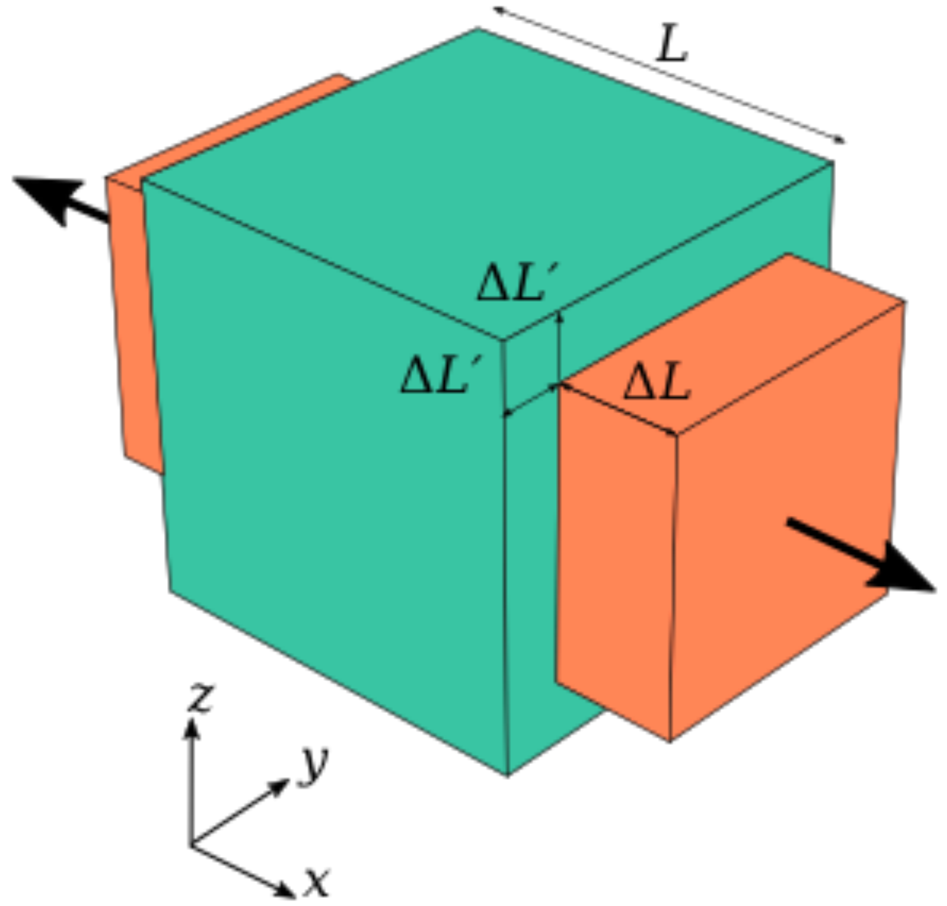
$$U(C) = \left(\tfrac{\lambda}{2}\,\mathrm{tr}(E)^2 + \mu\,\mathrm{tr}(E^2)\right) dV_M$$

Designing potential energy

- One can measure Young's modulus E, and Poisson ratio ν

$$E = \frac{F/A}{\Delta\ell/\ell}$$

similar to spring constant



$$\nu = -\frac{\Delta L'}{\Delta L}$$

usually between 0 and 0.5; it could be negative.

- Lamé constants

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$