Constitutive models

Constitutive model: determines **P** in terms of deformation

- ***Elastic material***: **P** depends only on current **F**
- ***Hyperelastic material***: work is independent of path
  ⇒ ***strain energy density function*** $\Psi(\mathbf{F})$

$$\mathbf{P} = \mathrm{d}\Psi/\mathrm{d}\mathbf{F}$$
$$P_{ij} = \mathrm{d}\Psi/\mathrm{d}F_{ij}$$

analogous to $\mathbf{f} = -\nabla U$

Constitutive models

Different choices of $\Psi \Rightarrow$ different models of elastic materials

- No deformation: $\Psi(\mathbf{I}) = 0$
- Rotation independence: $\Psi(\mathbf{R}\,\mathbf{F}) = \Psi(\mathbf{F})$

$\Rightarrow$ Decompose $\mathbf{F} = \mathbf{R}\,\mathbf{S}$. Ψ depends only on $\mathbf{S}$, min when $\mathbf{S} = \mathbf{I}$

Corotated linear elasticity

Quadratic in **S**:

$$\Psi(\mathbf{F}) = \mu \, \|\mathbf{S} - \mathbf{I}\|^2 + \lambda/2 \, \mathrm{tr}^2 \, (\mathbf{S} - \mathbf{I})$$

$$\Rightarrow \ \mathbf{P}(\mathbf{F}) = 2 \, \mu \, \mathbf{R} \, (\mathbf{S} - \mathbf{I}) + \lambda \, \mathbf{R} \, \mathrm{tr} \, (\mathbf{S} - \mathbf{I})$$

μ, λ: ***Lamé parameters***

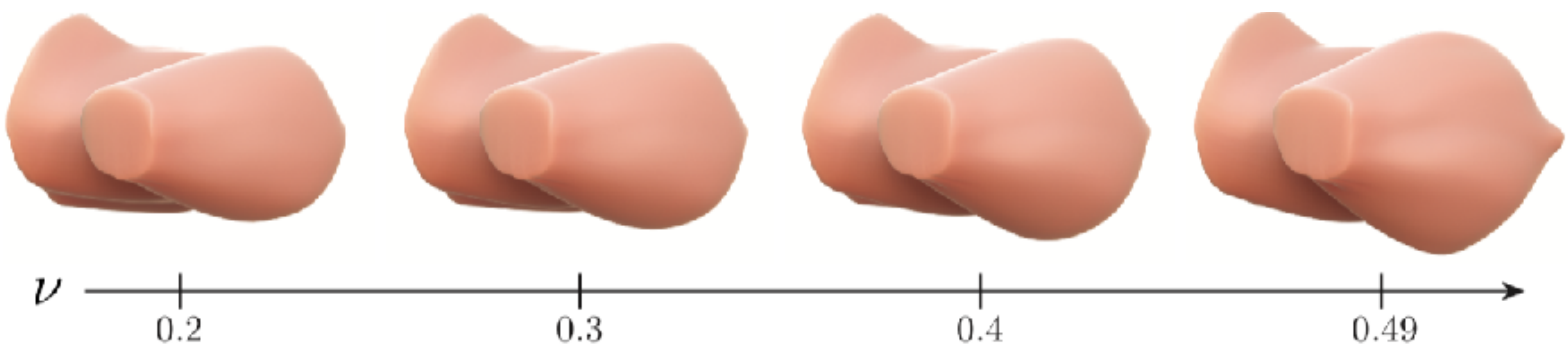
- μ: resistance to stretching, shearing
- λ: resistance to volume change (because $\mathrm{tr}\,(\mathbf{S} - \mathbf{I}) \approx \det \mathbf{F} - 1$)

Related to Young's modulus, Poisson's ratio

Material parameters

Conversion from Young's modulus k, Poisson's ratio ν:

$$\mu = \frac{k}{2(1+\nu)}, \quad \lambda = \frac{k\nu}{(1+\nu)(1-2\nu)}$$



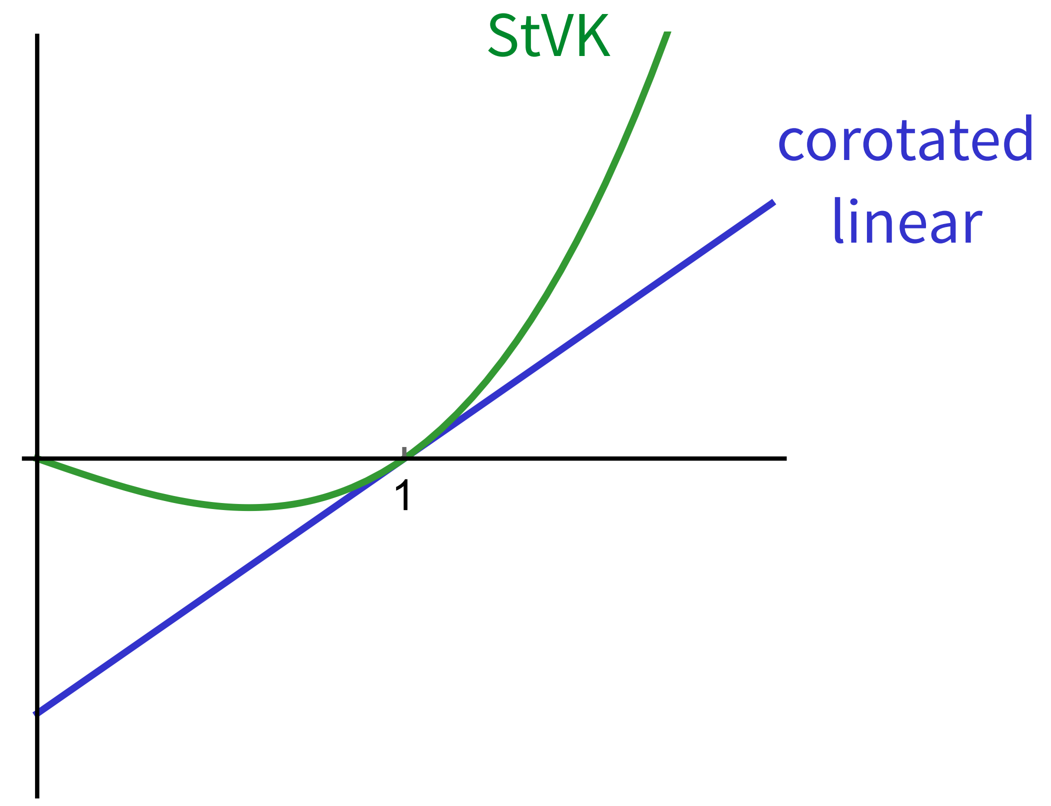
[Smith et al. 2018]

Other constitutive models

St. Venant–Kirchhoff (with $\mathbf{E} = ½\,(\mathbf{F}^\mathsf{T}\mathbf{F} - \mathbf{I})$):

$$\Psi(\mathbf{F}) = \mu\,\|\mathbf{E}\|^2 + \lambda/2\ \mathrm{tr}^2\,\mathbf{E}$$

- Easier to compute: no need for polar decomposition
- Ψ, $\mathbf{P}$ polynomial in $\mathbf{F}$
- Poor behaviour in compression

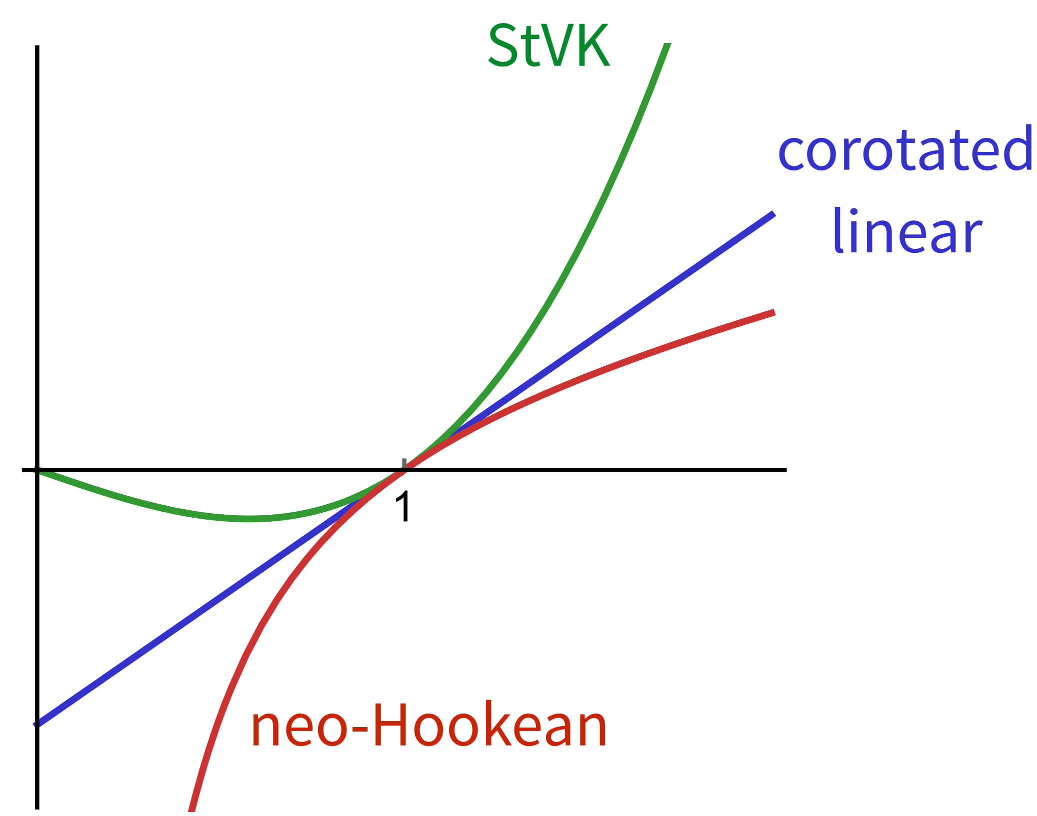


Other constitutive models

Neo-Hookean (with $\mathbf{C} = \mathbf{F}^\mathsf{T}\mathbf{F}$, $I_\mathbf{C} = \operatorname{tr} \mathbf{C}$, $J = \det \mathbf{F}$):

$$\Psi(\mathbf{F}) = \mu/2\ (I_\mathbf{C} - 3) - \mu \log J + \lambda/2\ (\log J)^2$$

- Correctly models volume change using $J = \det \mathbf{F}$
- Good for nearly incompressible materials (e.g. rubber, flesh)
- Undefined under collapse, inversion ($J \leq 0$)
- ***Stable neo-Hookean*** [Smith et al. 2018]



Anisotropy

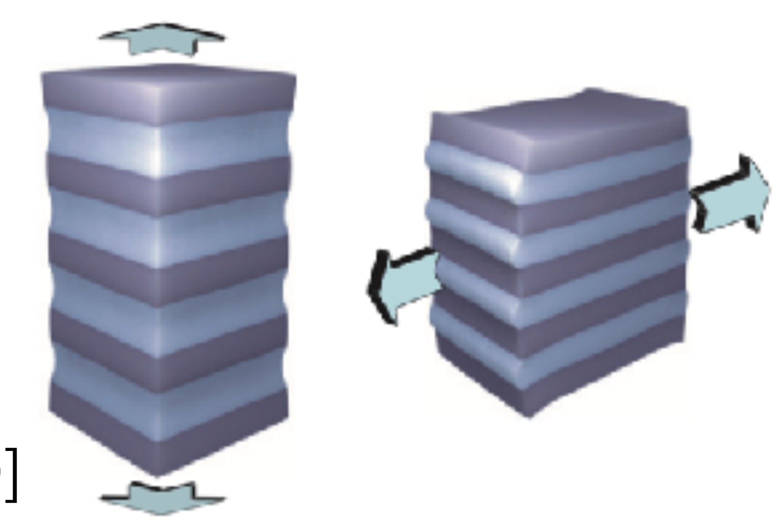
All these materials are ***isotropic***: independent of direction of stretching

$$\Psi(\mathbf{F}) = \Psi(\mathbf{F}\,\mathbf{Q})$$

$\Rightarrow \Psi(\mathbf{F}) = \Psi(\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T) = \Psi(\mathbf{\Sigma})$: Ψ only depends on principal strains

Equivalently, only depends on invariants of $\mathbf{C} = \mathbf{F}^T\mathbf{F}$

Examples of ***anisotropic*** materials:
woven fabrics, composites



[Kharevych et al. 2009]

From theory to simulation

State $\boldsymbol{\varphi}(\mathbf{X})$, $\dot{\boldsymbol{\varphi}}(\mathbf{X})$

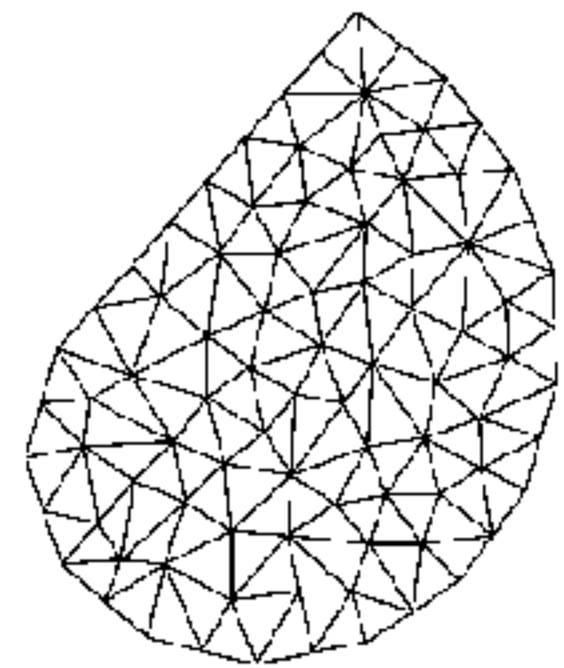
→ Deformation gradient $\mathbf{F}(\mathbf{X}) = d\boldsymbol{\varphi}/d\mathbf{X}$

→ Stress $\mathbf{P}(\mathbf{F}) = d\Psi/d\mathbf{F}$

→ Force density $\rho\, \ddot{\boldsymbol{\varphi}} = \text{div}\, \mathbf{P} + \mathbf{f}^{ext}$
(everything in terms of material space!)

- How to discretize $\boldsymbol{\varphi}(\mathbf{X})$?
- How to compute $\mathbf{F}$?
- How to compute div $\mathbf{P}$?

Meshes

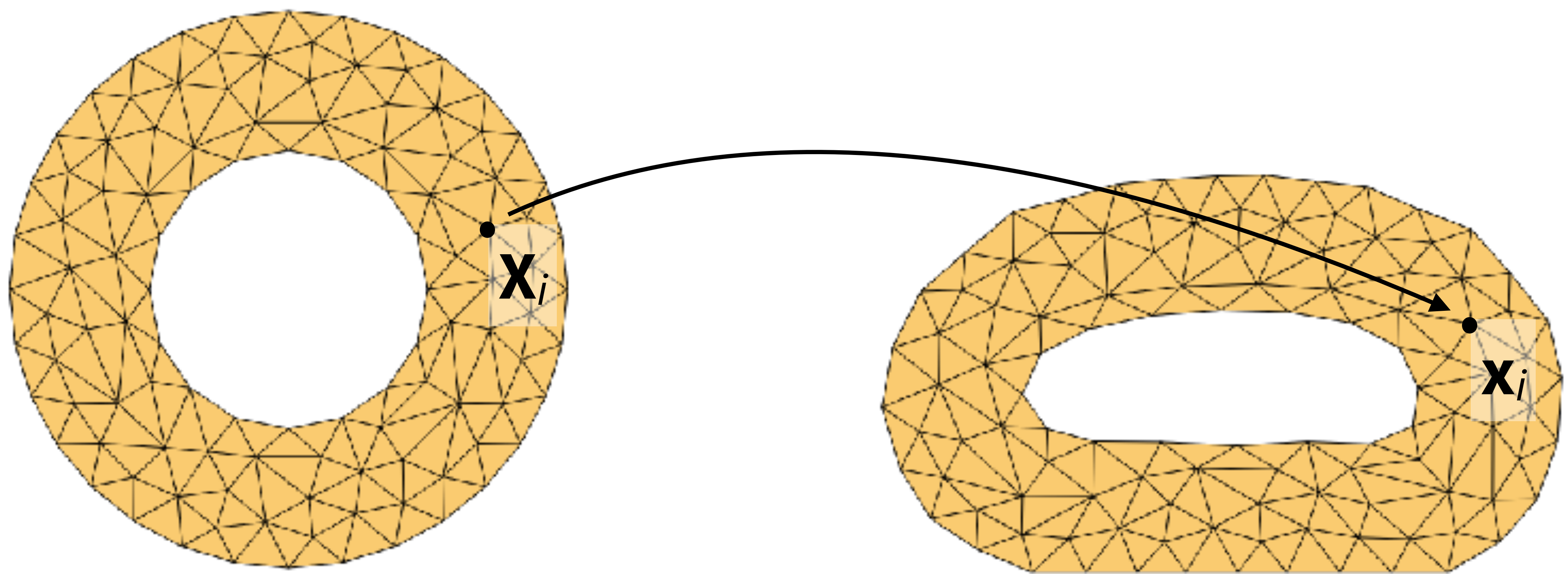


Divide space into ***elements*** (usually triangles, tetrahedra), put samples at vertices (a.k.a ***nodes***)

- ***Reconstruction***: linear / polynomial within each element
- ***Differentiation***: naively, gradient only defined in element interior
- Elements should be “well-shaped”: close to regular

Software to generate mesh for given shape: ***Triangle***, ***TetGen***

Meshes for elasticity



Create triangulated mesh in material space

- Each vertex stores fixed $\mathbf{X}_i$, varying $\mathbf{x}_i = \boldsymbol{\varphi}(\mathbf{X}_i)$, $\mathbf{v}_i = \dot{\boldsymbol{\varphi}}(\mathbf{X}_i)$
- Each element stores indices of vertices i_1, i_2, i_3 (, i_4)

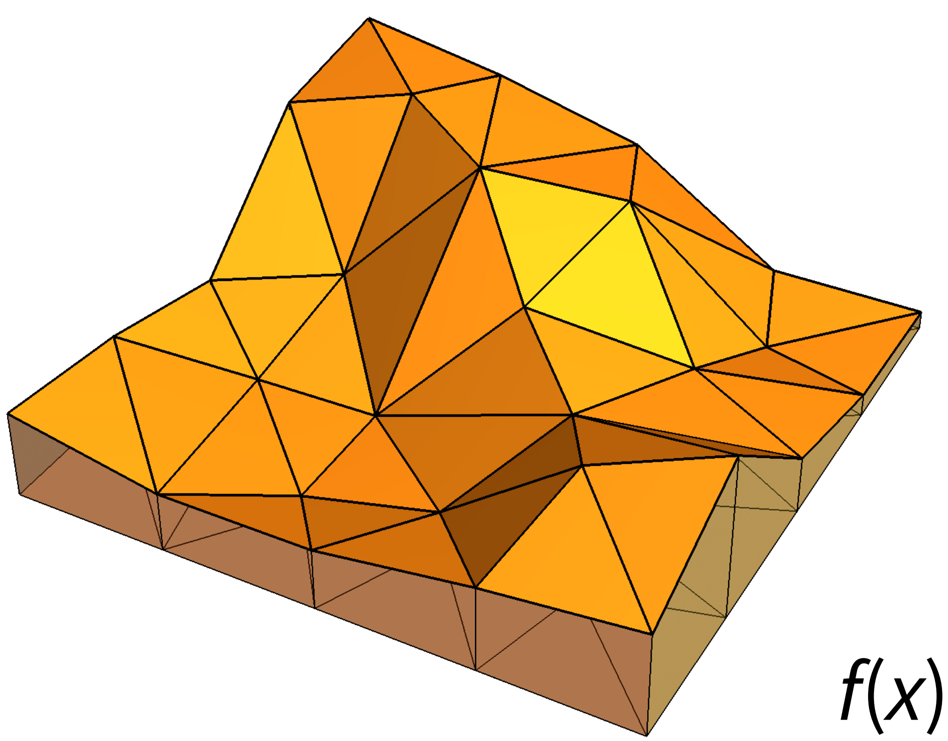
Reconstruct $\boldsymbol{\varphi}(\mathbf{X})$ by piecewise linear interpolation. (Other shapes e.g. quads, higher-order interpolation also possible)

Shape functions

Interpolation can be expressed as linear combination of ***shape functions***:

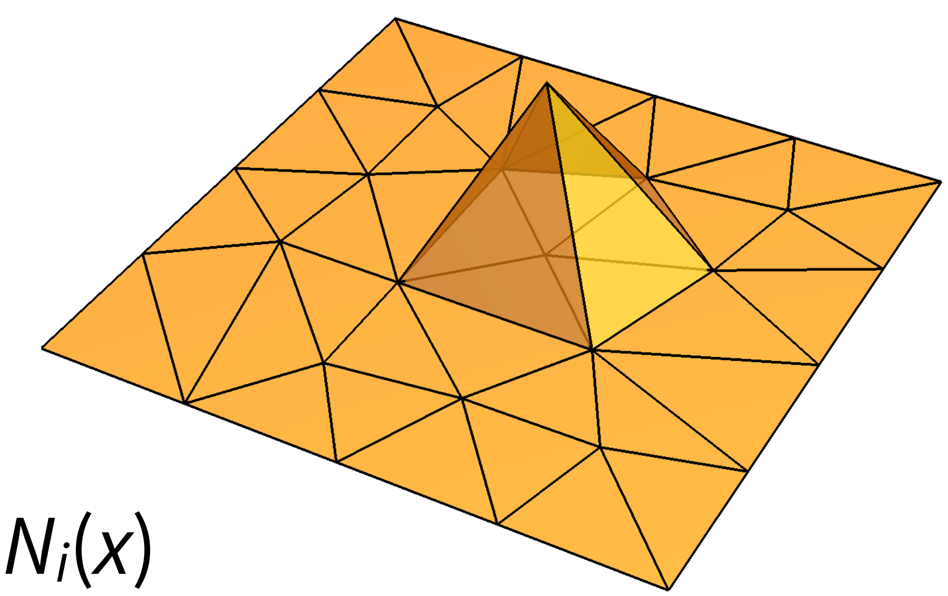
$$f(x) = f_1\, N_1(x) + f_2\, N_2(x) + \cdots + f_n\, N_n(x)$$

where $N_i(x_i) = 1,\ \ N_i(x_j) = 0$ for $j \neq i$



$f(x)$

$$= \sum f_i$$



$N_i(x)$

Shape functions

In particular,

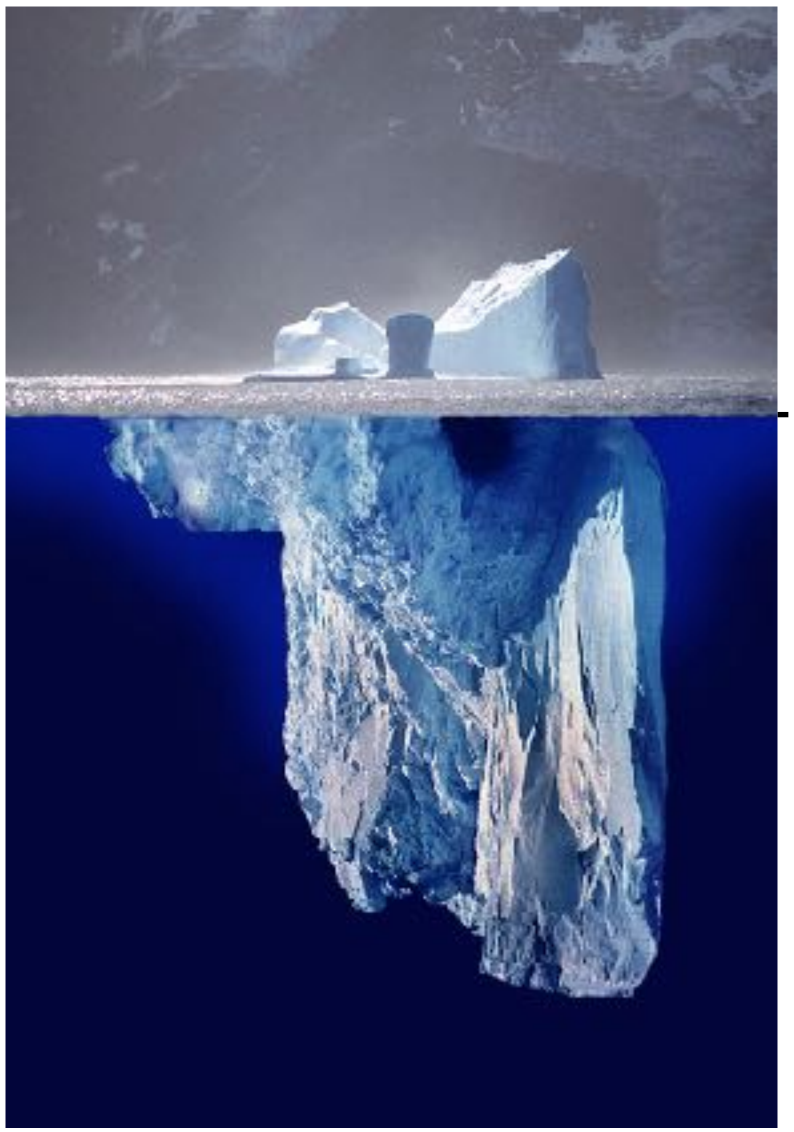
$$\boldsymbol{\varphi}(\mathbf{X}) = \mathbf{x}_1 \, N_1(\mathbf{X}) + \mathbf{x}_2 \, N_2(\mathbf{X}) + \cdots$$

$$\mathbf{F}(\mathbf{X}) = \mathbf{x}_1 \, \mathrm{d}N_1/\mathrm{d}\mathbf{X} + \mathbf{x}_2 \, \mathrm{d}N_2/\mathrm{d}\mathbf{X} + \cdots$$

Piecewise linear interpolation ⇒ **F**, **P** constant on each element

But what is div **P**?
Zero in element interior, undefined on element boundaries

The finite element method



FEM in graphics: Sifakis and Barbič, Part 1, Ch 4: “Discretization”

FEM theory: Bathe, *Finite Element Procedures* (especially Ch 3.3, 4.2)