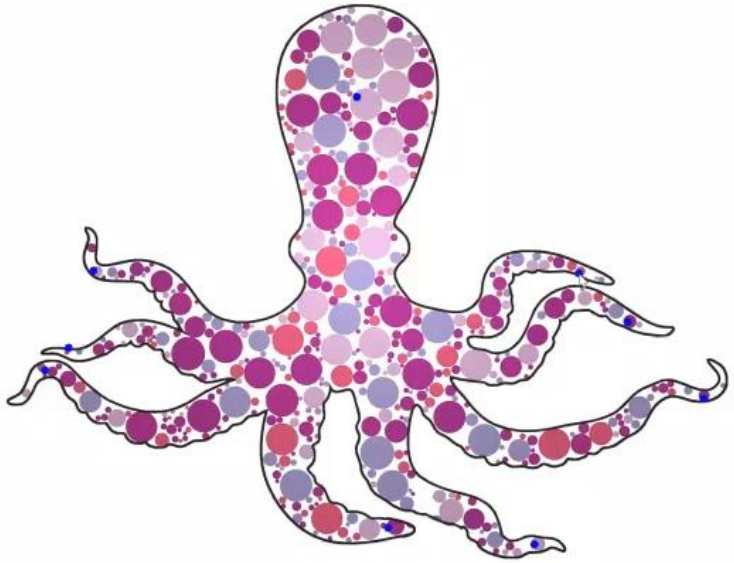
Physics-based models of deformation

Stelian Coros

Over the past few weeks…



- What characteristics are we looking for in a deformation model?

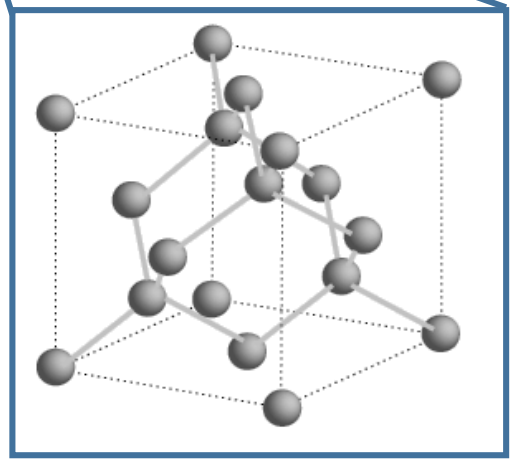
Elasticity – Definition

Elasticity: The ability of a material to resist a deforming force and to return to its original size and shape when that force is removed.

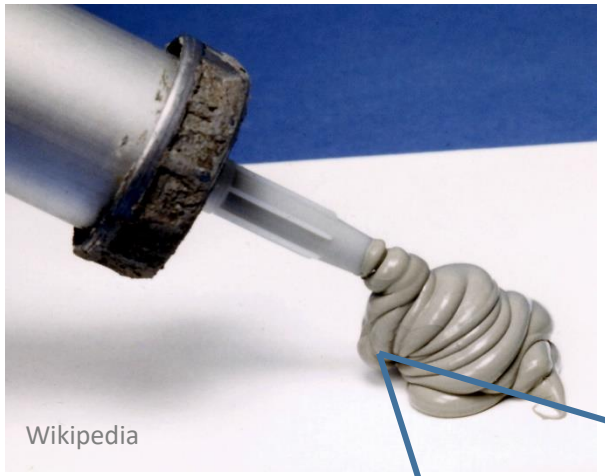
Elasticity – Mechanisms

Silicon [Si]

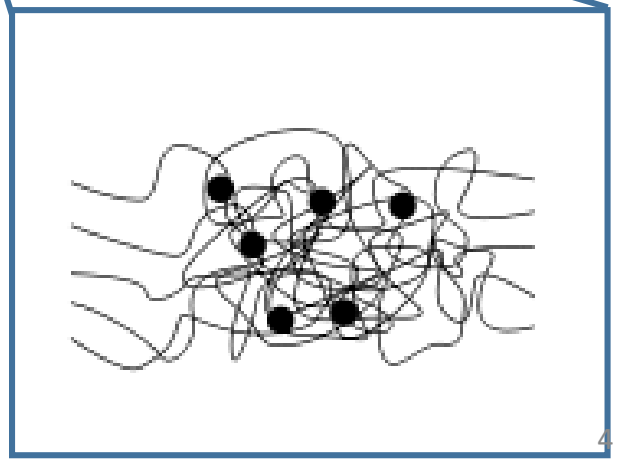
Crystal Structure



Silicone $[Si(CH_3)_2O]_n$

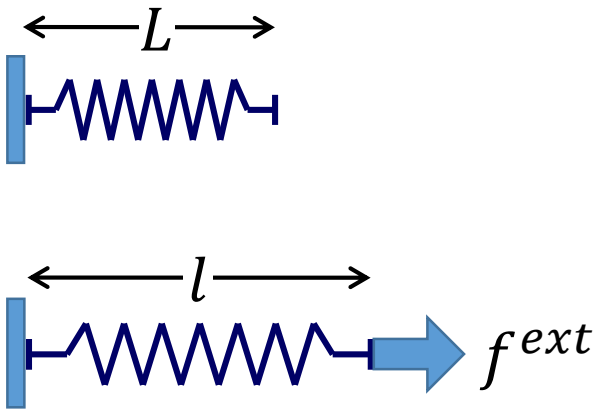


Polymer Structure



Modeling a simple elastic object

Hookean Springs in 1D



Length of undeformed spring	L
Length of deformed spring	l
Spring stiffness	k

Elasticity: *Ability of the spring to return to its initial length when the deforming force is removed.*

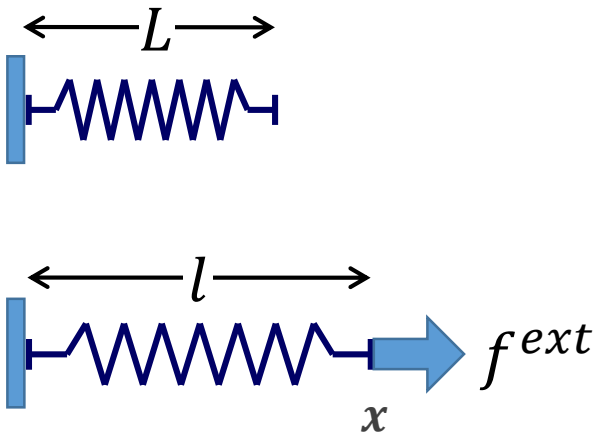
Spring Force:

- *Ceiiinosssttuv.* (Hooke, 1676)
- *Ut tensio, sic vis.* (Hooke, 1678)

  $\rightarrow$ *Force is linear w.r.t. extension!*

$$f^{ext} = k(l - L)$$ Hooke's Law

Hookean Springs in 1D



Length of undeformed spring	L
Length of deformed spring	l
Spring stiffness	k

For elastic springs, forces are **conservative**, i.e., no energy is lost during deformation.

Work done by forces to deform the spring:

$$W = \int_L^l f^{ext}(x)dx = \int_L^l k(x-L)dx$$

Stored energy of the spring is

$$E = W = \frac{1}{2}k(l-L)^2$$

Force f^{int} exerted by spring follows as negative gradient of E (how come?):

$$f^{int} = -\frac{dE}{dx} = -k(l-L)$$

Hookean Springs – Generalization

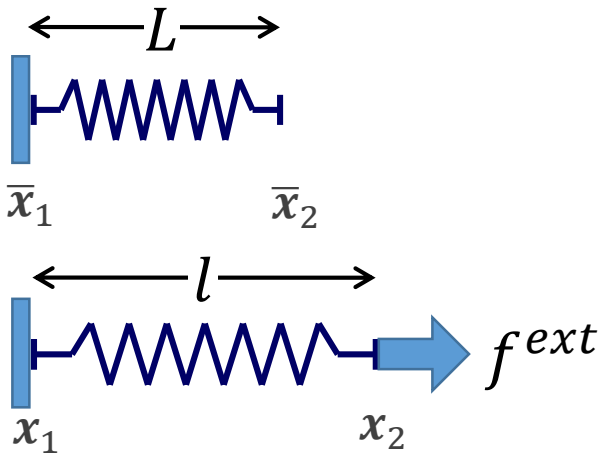
- **Inconvenience**: springs with same material but different lengths will have different stiffness characteristics

  A spring with rest length $L_1 = L$ deforms by $\delta l = L/2$
  A spring with rest length $L_2 = 100L$ deforms by the same δl
  Same stiffness k, same internal force according to Hook's law. Which one feels stiffer?
  Note relative deformation is VERY different!

- **Fix**: use relative deformation $\varepsilon = \frac{l-L}{L}$ and **material** stiffness $\tilde{k}$. Then energy density is $\Psi = \frac{1}{2}\tilde{k}\varepsilon^2$, and total stored energy is an integrated quantity:

$$E = \frac{1}{2}\tilde{k}\varepsilon^2 L$$

Hookean Springs in $\mathbf{R}^n$



Length of undeformed spring	L
Length of deformed spring	l
Spring stiffness	k

The configuration of a spring is determined by the position of its two endpoints. We will distinguish between

- Deformed positions $\boldsymbol{x}_1,\ \boldsymbol{x}_2 \in R^n$
- Undeformed positions $\bar{\boldsymbol{x}}_1,\ \bar{\boldsymbol{x}}_2 \in R^n$

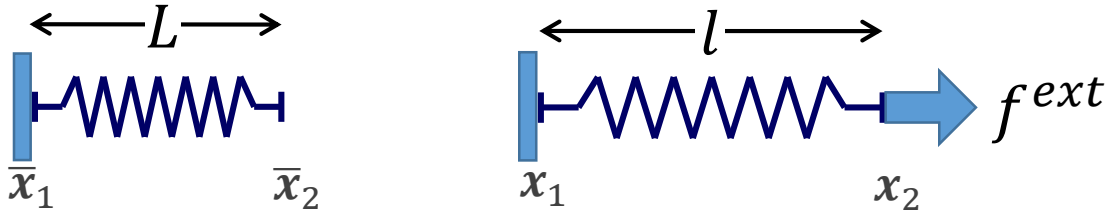
Lengths are functions of positions, i.e.,

$$l = |\boldsymbol{x}_2 - \boldsymbol{x}_1|_2 \text{ and } L = |\bar{\boldsymbol{x}}_2 - \bar{\boldsymbol{x}}_1|_2$$

Everything else stays the same

Hookean Springs in nD

- 1 mass point, 1 spring



Deformation Measure

$$\varepsilon = \frac{l}{L} - 1$$

Elastic Energy

$$E = \frac{1}{2}\tilde{k}\varepsilon^2 L$$

Forces

$$\boldsymbol{f}_{int} = -\frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}}$$

Working it out…

$l = |\boldsymbol{e}|_2 = (\boldsymbol{e}^T\boldsymbol{e})^{\frac{1}{2}}$
with $\boldsymbol{e} = \boldsymbol{x}_2 - \boldsymbol{x}_1$

$$\boldsymbol{f}_1 = -\frac{\partial E(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial \boldsymbol{x}_1} = -\frac{\partial E(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial l}\frac{\partial l}{\partial \boldsymbol{x}_1}$$

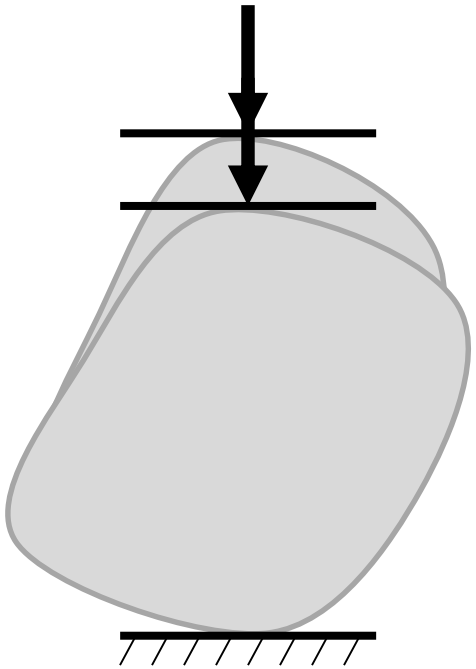
$$\frac{\partial E}{\partial l} = \tilde{k}\varepsilon$$

$$\frac{\partial l}{\partial \boldsymbol{x}_1} = \frac{1}{2}(\boldsymbol{e}^T\boldsymbol{e})^{-\frac{1}{2}}\frac{\partial(\boldsymbol{e}^T\boldsymbol{e})}{\partial \boldsymbol{x}_1}$$

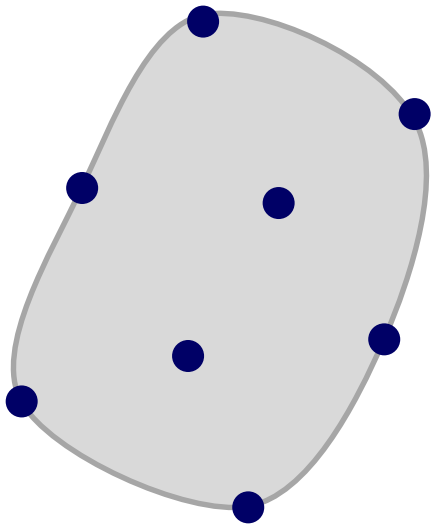
$$\boldsymbol{f}_1 = -\tilde{k}\left(\frac{l}{L} - 1\right)\frac{\boldsymbol{x}_2 - \boldsymbol{x}_1}{|\boldsymbol{x}_2 - \boldsymbol{x}_1|}$$

$$\boldsymbol{f}_1 = -\boldsymbol{f}_2$$

Modeling complex elastic objects

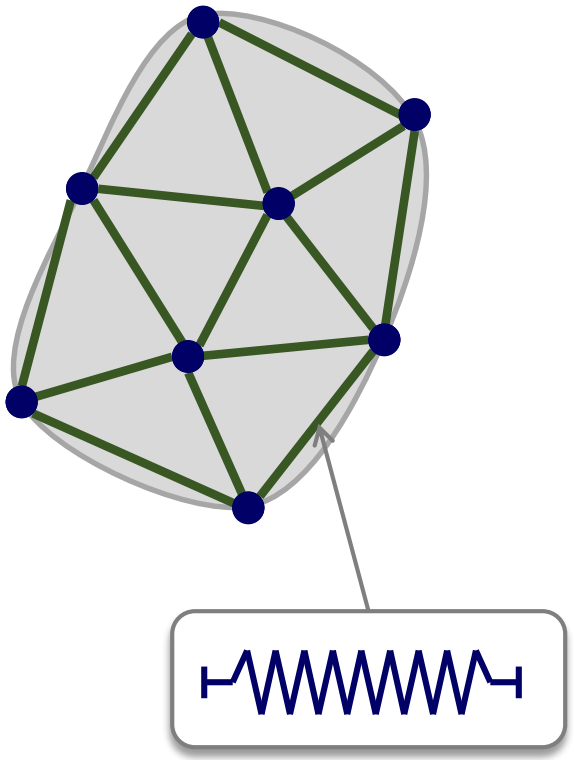


Modeling complex elastic objects



Straightforward concept: sample object with mass points, connect them with springs…

Modeling complex elastic objects

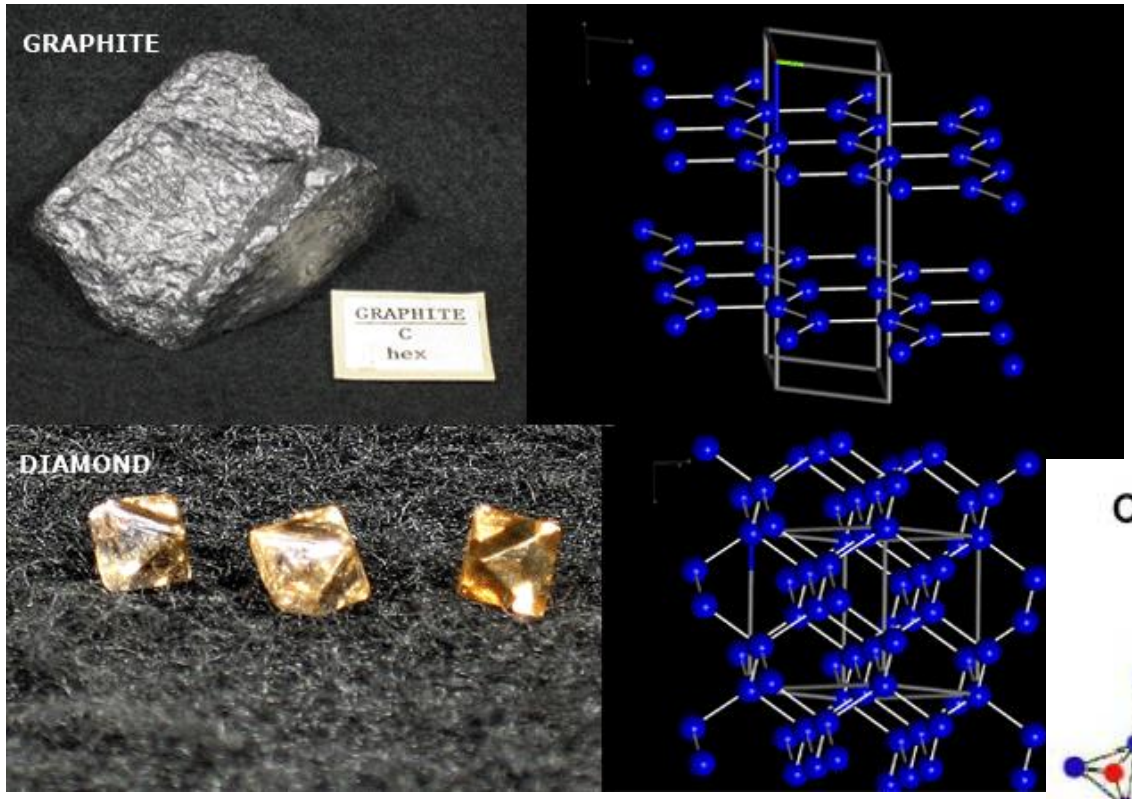


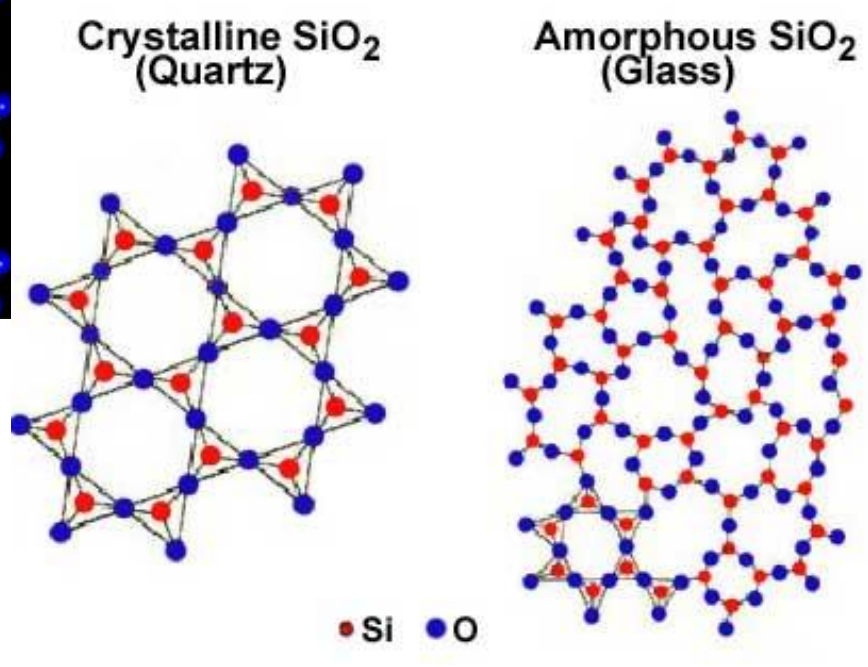
Straightforward concept: sample object with mass points, connect them with springs…

Representation: 2D triangle mesh (or 3D tetrahedral mesh, of course)

- Vertices $\mathbf{x}_i \in \mathbf{R}^2$
- Edges E_{ij} are springs connecting vertices $\mathbf{x}_i$ and $\mathbf{x}_j$

“Mass-spring” systems in the wild





Spring Networks

Energy of spring network

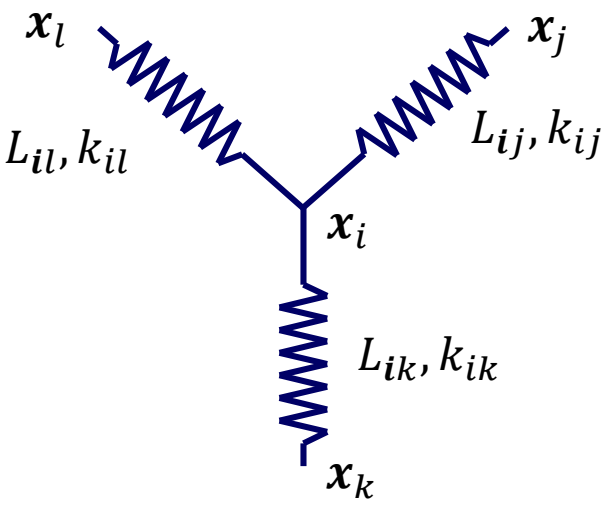
$$E = \sum_k E_k$$

Total spring force at given node

$$\boldsymbol{f}_i^{int} = -\frac{\partial E}{\partial \boldsymbol{x}_i} = -\sum_k \frac{\partial E_k}{\partial \boldsymbol{x}_i}$$

Total force at given node

$$\boldsymbol{f}_i = \boldsymbol{f}_i^{int} + \boldsymbol{f}_i^{ext}$$



Force equilibrium

Given applied forces $\boldsymbol{f}^{ext}$, how to compute resulting configuration $\boldsymbol{x}$?

For static equilibrium, the acceleration has to be zero for all nodes,

$$\boldsymbol{a_i}(\boldsymbol{x}) = \mathbf{0}\ \forall i$$

From Newton's second law, we know that

$$\boldsymbol{f}_i(\boldsymbol{x}) = m_i \boldsymbol{a}_i(\boldsymbol{x}) = \mathbf{0}$$

Static Equilibrium Conditions

$$\boldsymbol{f}_i^{int}(\boldsymbol{x}) + \boldsymbol{f}_i^{ext} = \mathbf{0}\ \forall i$$

Equilibrium Conditions – Energy

If internal and external forces derive from potential fields,

$$\boldsymbol{f}_i^{int} = -\frac{\partial E^{int}}{\partial \boldsymbol{x}_i} \qquad \boldsymbol{f}_i^{ext} = -\frac{\partial E^{ext}}{\partial \boldsymbol{x}_i}$$

then, static equilibrium conditions

$$\boldsymbol{f}_i^{int}(\boldsymbol{x}) + \boldsymbol{f}_i^{ext} = \mathbf{0}\ \forall i$$

are equivalent to $\boldsymbol{x}$ being a **stationary point** for the total energy

$$E(\boldsymbol{x}) = E^{int}(\boldsymbol{x}) + E^{ext}(\boldsymbol{x}),\ \text{i.e.,}\ \frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}} = \mathbf{0}$$

Problem Statement

Task: find minimizer $\boldsymbol{x}^*$ of function $E(\boldsymbol{x})$,

$$\boldsymbol{x}^* = \text{argmin}_x \, E(\boldsymbol{x}) \; \rightarrow E(\boldsymbol{x}^*) \leq E(\boldsymbol{x}) \; \forall \; \boldsymbol{x}$$

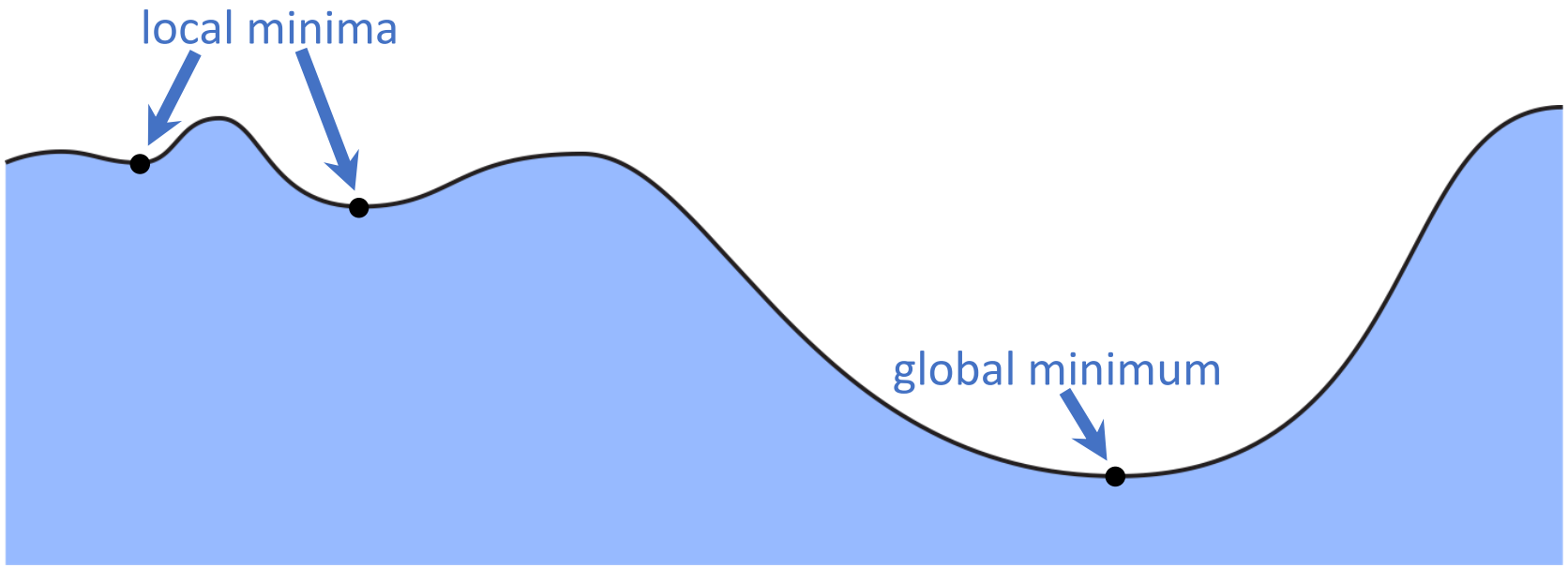
In general:

- $E(\boldsymbol{x})$ is a nonlinear function
- $E(\boldsymbol{x})$ is multivariate, i.e., $\boldsymbol{x} \in R^n$ with $n \geq 2$
- $E(\boldsymbol{x})$ may have local minima and maxima (numerical artifacts or expected behavior of physical system)?

Local vs. Global Minima

Global minimum is absolute best among all possibilities

Local minimum is best "among immediate neighbors"



Somewhat philosophical question: does a local minimum *"solve"* the problem?

Depends on the problem!

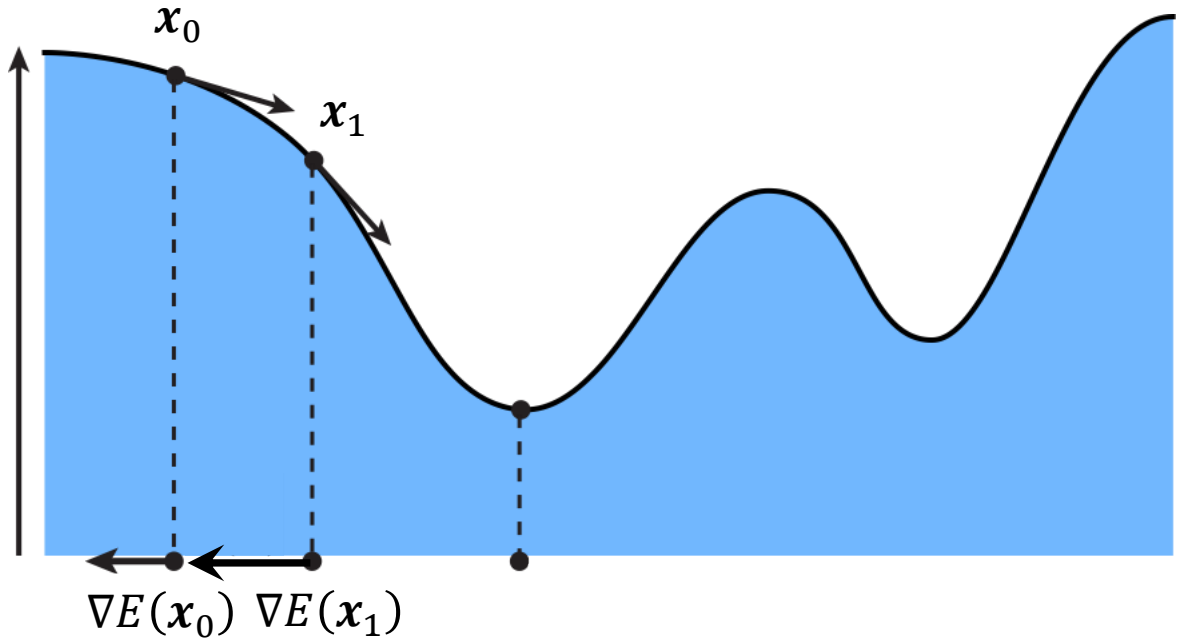
Solution Strategy

Given a point $\boldsymbol{x}_0$ how do we get to a minimum?

- "Walk" into a direction that decreases $E(\boldsymbol{x})$
- Which direction would you choose?

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha \nabla E(\boldsymbol{x}_n)$$

The derivative

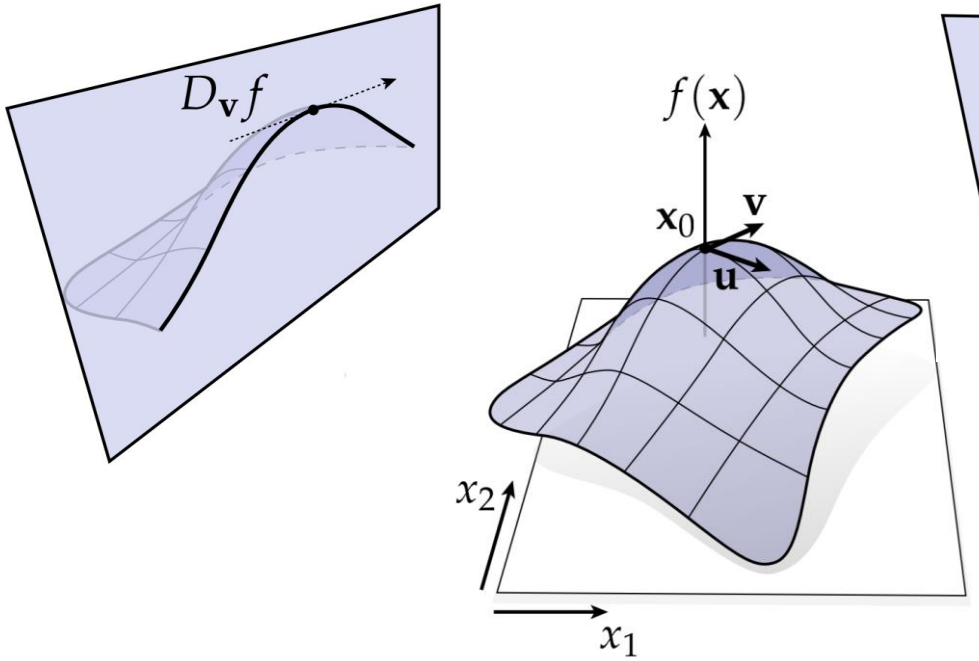


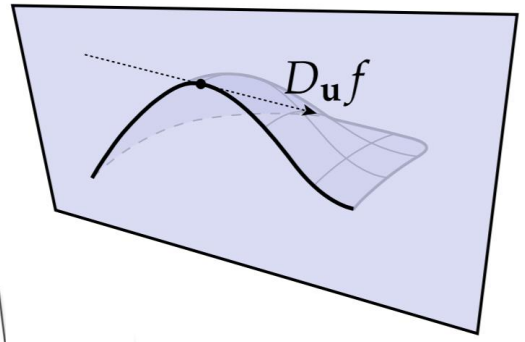
Directional Derivatives and the Gradient

In 1D: derivative == slope == rise over run

In higher dimensions: take a slice through this function along some direction

- Then apply the usual derivative concept
- This is called the directional derivative





take a small step along u

$$D_{\mathbf{u}} f(\mathbf{x}_0) := \lim_{\varepsilon \to 0} \frac{f(\mathbf{x}_0 + \varepsilon \mathbf{u}) - f(\mathbf{x}_0)}{\varepsilon}$$

Directional Derivatives and the Gradient

Starting from Taylor's series

$$f(\mathbf{x}) \approx \underbrace{f(\mathbf{x}_0)}_{c\in\mathbb{R}} + \underbrace{\langle \nabla f(\mathbf{x}_0), \mathbf{x}-\mathbf{x}_0\rangle}_{\mathbf{b}\in\mathbb{R}^n} + \underbrace{\langle \nabla^2 f(x_0)(\mathbf{x}-\mathbf{x}_0), \mathbf{x}-\mathbf{x}_0\rangle/2}_{\mathbf{A}\in\mathbb{R}^{n\times n}}$$

constant linear quadratic

easy to see that

take a small step along u

$$D_{\mathbf{u}} f(\mathbf{x}_0) := \lim_{\varepsilon\to 0} \frac{f(\mathbf{x}_0+\varepsilon\mathbf{u}) - f(\mathbf{x}_0)}{\varepsilon} = \langle \nabla f(x), \mathbf{u}\rangle$$

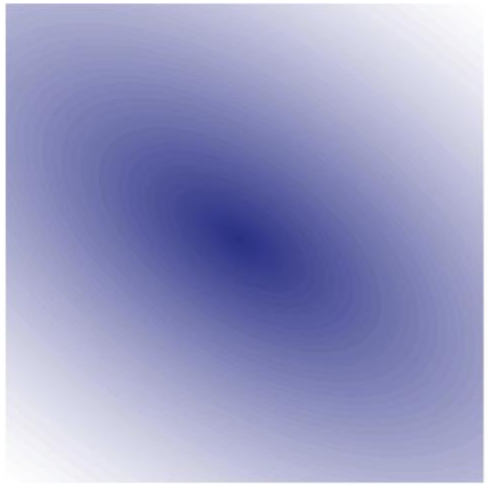
Directional Derivatives and the Gradient

Given a multivariate function $f(\boldsymbol{x})$, gradient assigns a vector $\nabla f(\boldsymbol{x})$ at each point

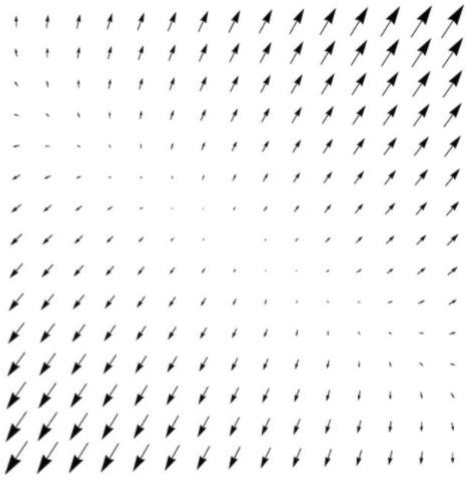
Inner product between gradient and any unit vector gives directional derivative "along that direction"

- Another way to think about it is that inner product outputs the component of the gradient along a unit vector…

Out of all possible unit vectors, what is the one along which the function changes most drastically?



$f(\mathbf{x})$



$\nabla f(\mathbf{x})$

Gradient in coordinates

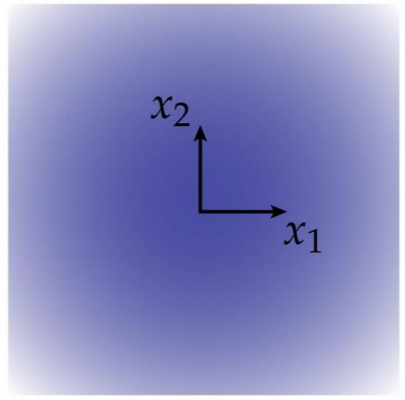
Most familiar definition: list of partial derivatives

$$f(\mathbf{x}) := x_1^2 + x_2^2$$

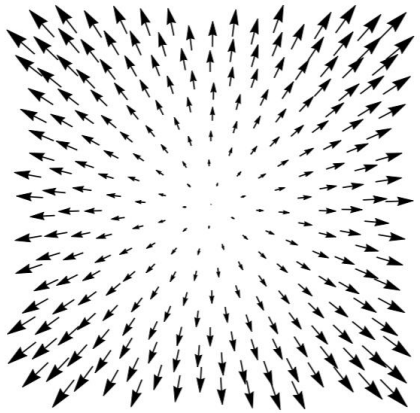
$$\frac{\partial f}{\partial x_1} = \frac{\partial}{\partial x_1} x_1^2 + \frac{\partial}{\partial x_1} x_2^2 = 2x_1 + 0$$

$$\frac{\partial f}{\partial x_2} = \frac{\partial}{\partial x_2} x_1^2 + \frac{\partial}{\partial x_2} x_2^2 = 0 + 2x_2$$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = 2\mathbf{x}$$



$f(\mathbf{x})$

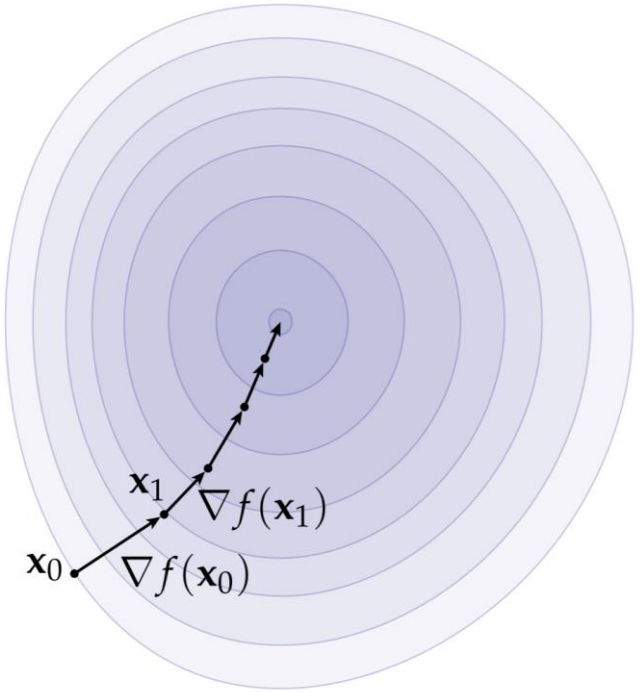


$\nabla f(\mathbf{x})$

Gradient is direction of steepest ascent

Function value

- gets largest if we move in direction of gradient
- doesn't change if we move orthogonally
- decreases *fastest* if we move exactly in opposite direction



Steepest Descent – Algorithm I

Algorithm: steepest_descent

Input: x //initial guess

α //step length

ε //threshold

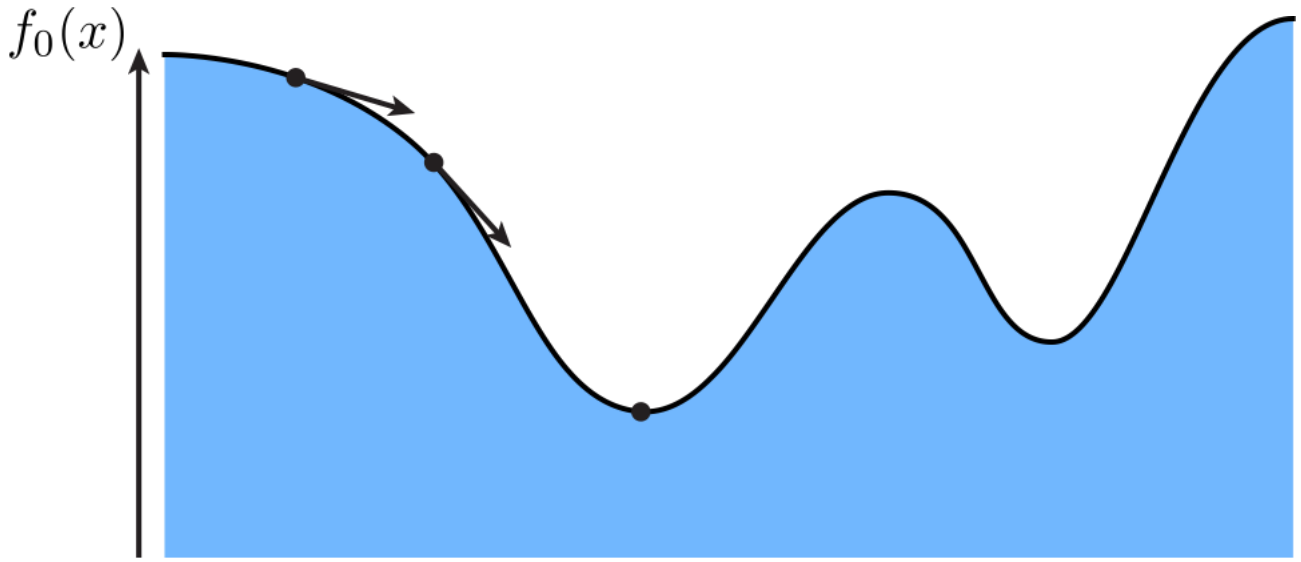
while abs$(\nabla E(x)) > \varepsilon$ **do**

$x = x - \alpha \nabla E(x)$;

end do;

Gradient Descent (1D)

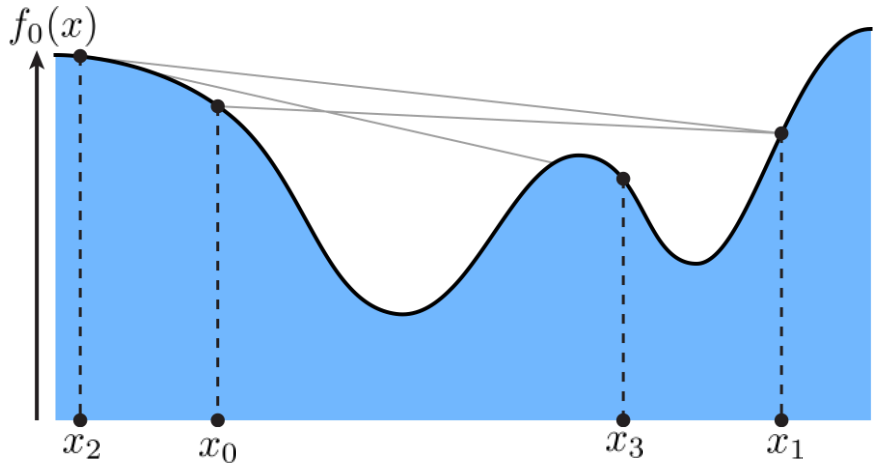
- Basic idea: follow the gradient “downhill” until it’s zero
- (Zero gradient is the 1st-order optimality condition)



Steepest Descent – Observations

- Progress is initially good, but slows down when approaching minimum
- Number of steps to convergence depends on step length α
  - Too small alpha leads to slow progress
  - Too large alpha leads to divergence

Gradient Descent Algorithm (1D)



Steepest Descent – Observations

- Progress is initially good, but slows down when approaching minimum
- Number of steps to convergence depends on step length α
  - Too small alpha leads to slow progress
  - Too large alpha leads to divergence

How can we improve this basic steepest descent algorithm?

Steepest Descent – Improvements

- **Idea**: enforce monotonicity, i.e., $E(x_{n+1}) < E(x_n)\ \forall\ n$.
- **Goal**: in each step, find α such that

$$\text{(I)} \quad E(x_n - \alpha \nabla E(x_n)) < E(x_n)$$

- From Taylor series, we know that $\exists\ \alpha > 0$ such that (I) holds

e.g. $f(x + dx) = \sum_{i=0}^{\infty} \frac{f^{(n)}(x)}{n!} dx^n$

$$= f(x) + f'(x)dx + O(dx^2)$$

$$\approx f(x) + f'(x)dx \quad \textbf{if } dx \text{ is } \textit{sufficiently} \text{ small}$$

- **Approach**: reduce α until condition (I) is satisfied

Line Search

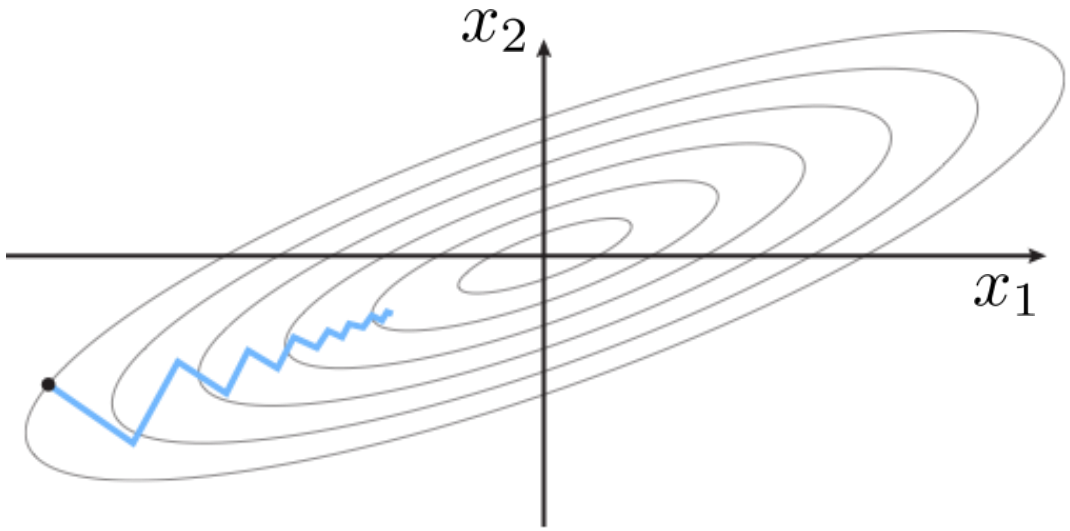
Algorithm: line_search
Input: x, dx, α, β
while $E(x - \alpha * dx) > E(x)$ **do**
    $\alpha = \alpha * \beta;$
end do;

Algorithm: steepest_descent
Input: $x, dx, \alpha, \beta, \varepsilon$
while abs$(\nabla E(x)) > \varepsilon$ **do**
    $dx = \nabla E(x);$
    $\alpha = \text{line\_search}(x, dx, \alpha, \beta);$
    $x = x - \alpha\, dx;$
end do;

x	current state
dx	search direction
α	initial step length
$0 < \beta < 1$	scaling factor

Gradient Descent Algorithm (nD)

- Basic challenge in nD:
  - solution can “oscillate”
  - takes many, many small steps
  - very slow to converge



How can we improve on this?

Newton's method

Alternative approach:

- Find dx such that $\nabla E(x_n + dx) = 0$

- Taylor series on gradient, to first order

$$\nabla E(x_n + dx) = \nabla E(x_n) + \nabla^2 E(x_n) dx = 0$$

$$dx = -\nabla^2 E(x_n)^{-1} \nabla E(x_n)$$

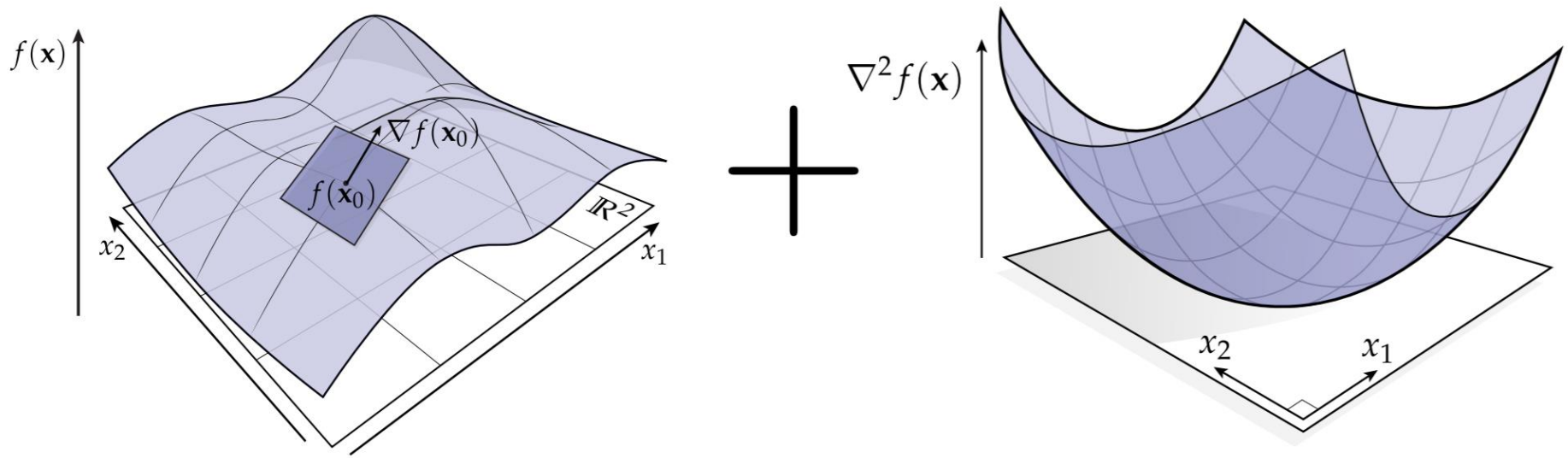
The Hessian visualized

Gradient: linear approximation

Hessian: quadratic approximation

$$f(\mathbf{x}) \approx \underbrace{f(\mathbf{x}_0)}_{c \in \mathbb{R}} + \underbrace{\langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle}_{\mathbf{b} \in \mathbb{R}^n} + \underbrace{\langle \nabla^2 f(x_0)(\mathbf{x} - \mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle / 2}_{\mathbf{A} \in \mathbb{R}^{n \times n}}$$

constant linear quadratic



Newton's Method

Algorithm: steepest_descent

Input: $x, \alpha, \beta, \varepsilon$

while abs$(\nabla E(x)) > \varepsilon$ **do**

$dx = -\nabla E(x)$;

$\alpha = \text{line\_search}(x, dx, \alpha, \beta)$;

$x = x + \alpha\, dx$;

end do;

Algorithm: newton

Input: $x, \alpha, \beta, \varepsilon$

while abs$(\nabla E(x)) > \varepsilon$ **do**

$dx = -\nabla^2 E^{-1}(\mathrm{x}) * \nabla E(x)$;

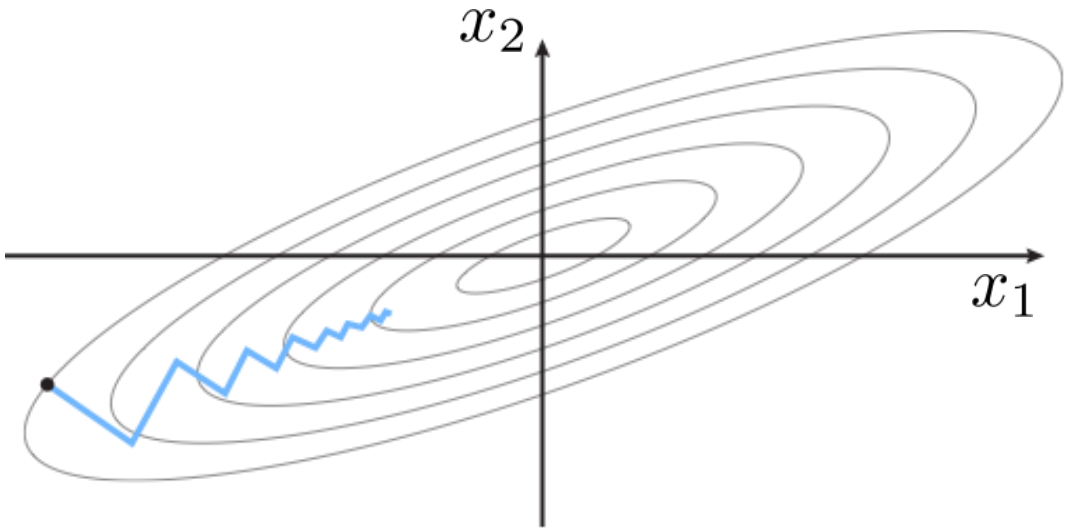
$\alpha = \text{line\_search}(x, dx, \alpha, \beta)$;

$x = x + \alpha dx$;

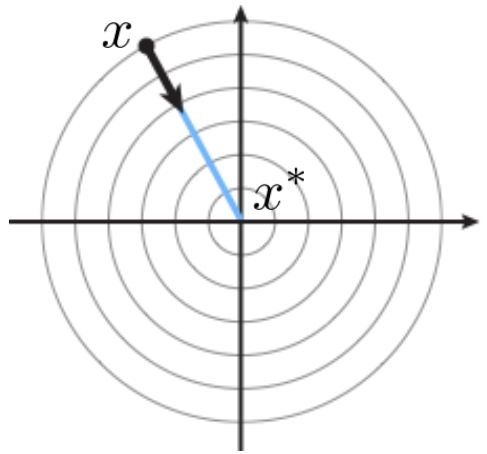
end do;

Observations & Interpretations

- Newton's method converges much faster than Steepest Descent



Gradient Descent



Newton's method: coordinate transformation that makes the energy landscape look like a "round bowl"

Spring Networks - Hessian

- Given $\boldsymbol{x}$, we can compute E_{int}, E_{ext} and $\partial E / \partial \boldsymbol{x}$.

  $\rightarrow$ *Sufficient for gradient descent*.

- For NM, we also need second derivatives, i.e., the Hessian $\mathbf{H} = \frac{\partial^2 E}{\partial \boldsymbol{x}^2}$
- Hessian for given spring

$$\mathbf{H}_{ij}^{k} = \frac{\partial^2 E^k}{\partial \boldsymbol{x_i} \partial \boldsymbol{x_i}}$$

Forces & Force Jacobians

- So, what does $\mathbf{H} = \frac{\partial^2 E}{\partial x^2} = -\frac{\partial F}{\partial x}$ represent?

- Much more intuitive to think in terms of blocks of the Jacobian:

$$\frac{\partial F_i}{\partial x_j}$$

- "how does the force on particle i change when the position of particle j changes"

Forces & Force Jacobians

- How do we compute the force Jacobian?
  - Analytic formulas
  - Numerical approach
    - Finite Differences, very useful for prototyping/debugging
  - Automatic & Symbolic differentiation
    - e.g. Maple

```
> fe1:=x^2;                          <- Expression giving the square of x
                                   fe1 := x^2
> fe2:=x*sin(x)-x;                   <- More complicated Expression
                                fe2 := x sin(x) - x
> dfe1:=diff(fe1,x);                 <- Derivative of fe1 with respect to x
                                  dfe1 := 2 x
> dfe2:=diff(fe2,x);                 <- Derivative of fe2 with respect to x
                             dfe2 := sin(x) + x cos(x) - 1
```

Gradients of Matrix-Valued Expressions

EXTREMELY useful to be able to differentiate matrix-valued expressions!

For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and **symmetric** matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

MATRIX DERIVATIVE	LOOKS LIKE
$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{y}) = \mathbf{y}$	$\frac{d}{dx}xy = y$
$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{x}) = 2\mathbf{x}$	$\frac{d}{dx}x^2 = 2x$
$\nabla_{\mathbf{x}}(\mathbf{x}^T A\mathbf{y}) = A\mathbf{y}$	$\frac{d}{dx}axy = ay$
$\nabla_{\mathbf{x}}(\mathbf{x}^T A\mathbf{x}) = 2A\mathbf{x}$	$\frac{d}{dx}ax^2 = 2ax$
$\ldots$	$\ldots$

Excellent resource: Petersen & Pedersen, "The Matrix Cookbook"

At least once in your life, work these out meticulously in coordinates!

Checking Derivatives

How do you know if your derivatives are correct?

- Compare analytical derivatives against numerical derivatives computed with finite differences.
- Taylor series for elastic energy:

$$E(\boldsymbol{x} + \boldsymbol{dx}) = E(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{dx} + O(\boldsymbol{dx}^2)$$

- FD approximation for elastic gradient

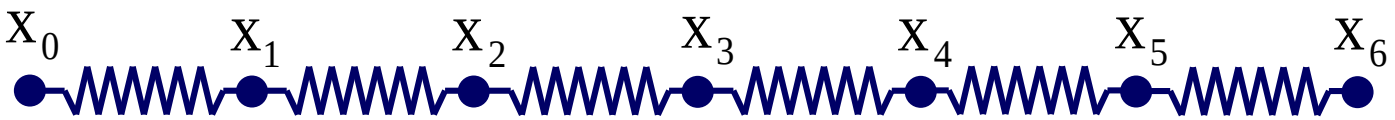
$$\boldsymbol{g}^{FD}(\boldsymbol{x})\boldsymbol{dx} = E(\boldsymbol{x} + \boldsymbol{dx}) - E(\boldsymbol{x})$$

- Can assume gradients are correct if $|\boldsymbol{g}^{FD} - \boldsymbol{g}| < \varepsilon$
- Even better, verify quadratic decrease in residual

$$r(\boldsymbol{dx}) = E(\boldsymbol{x} + \boldsymbol{dx}) - E(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x})\boldsymbol{dx}$$

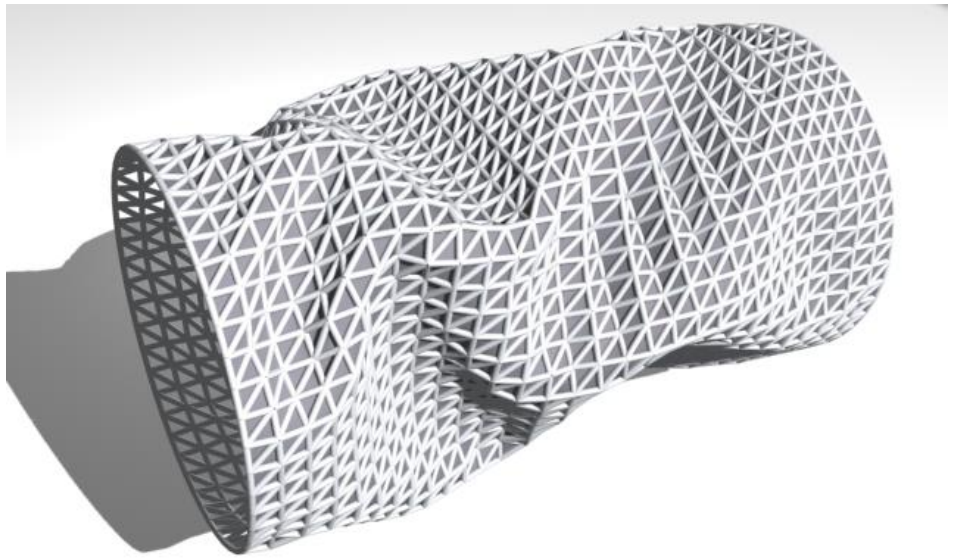
Forces & Force Jacobians

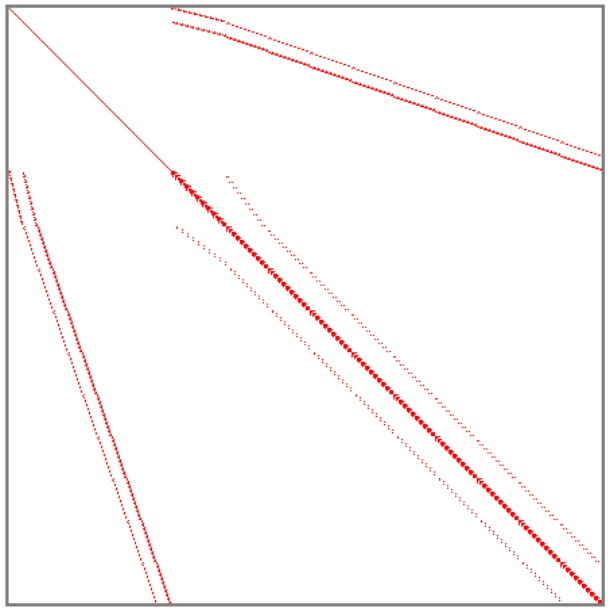
- What does the structure of the force Jacobian look like?



Forces & Force Jacobians

- Block i,j is non-zero only if there is a spring between particles i and j. In general, connectivity structure is very sparse - most entries are therefore zero!





Spring Networks – Hessian

- What can be said about $\mathbf{H}$?
  - $\mathbf{H}$ is symmetric: $\mathbf{H}_{ij} = \mathbf{H}_{ji}^{T}$
  - $\mathbf{H}$ is sparse: $\mathbf{H}_{ij}$ is only nonzero if there is a spring between nodes i and j
  - $\mathbf{H}$ is singular, *why?*
    - remove null-space by specifying constraints/boundary conditions

Newton's Method in nD

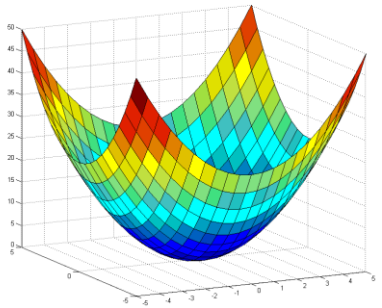
Observations & Remarks

- In general $\boldsymbol{dx} \neq -\boldsymbol{g}$ but we require $\boldsymbol{dx}^T \boldsymbol{g} < 0$
- All robustness measures (e.g. line search) remain necessary
- Additionally have to ensure robust solution of linear system

Indefinite Systems

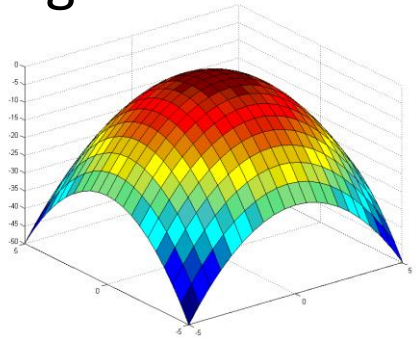
- Some linear system solvers (e.g. cholesky factorization and CG) only work if $\mathbf{A}$ is positive semi-definite (PSD).
- What does it mean for the hessian of a physical energy to be indefinite?

Positive Definite



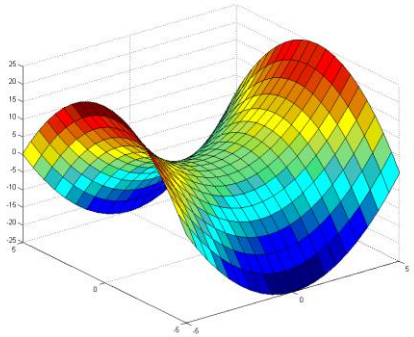
$\lambda_1, \lambda_2 > 0$

Negative Definite



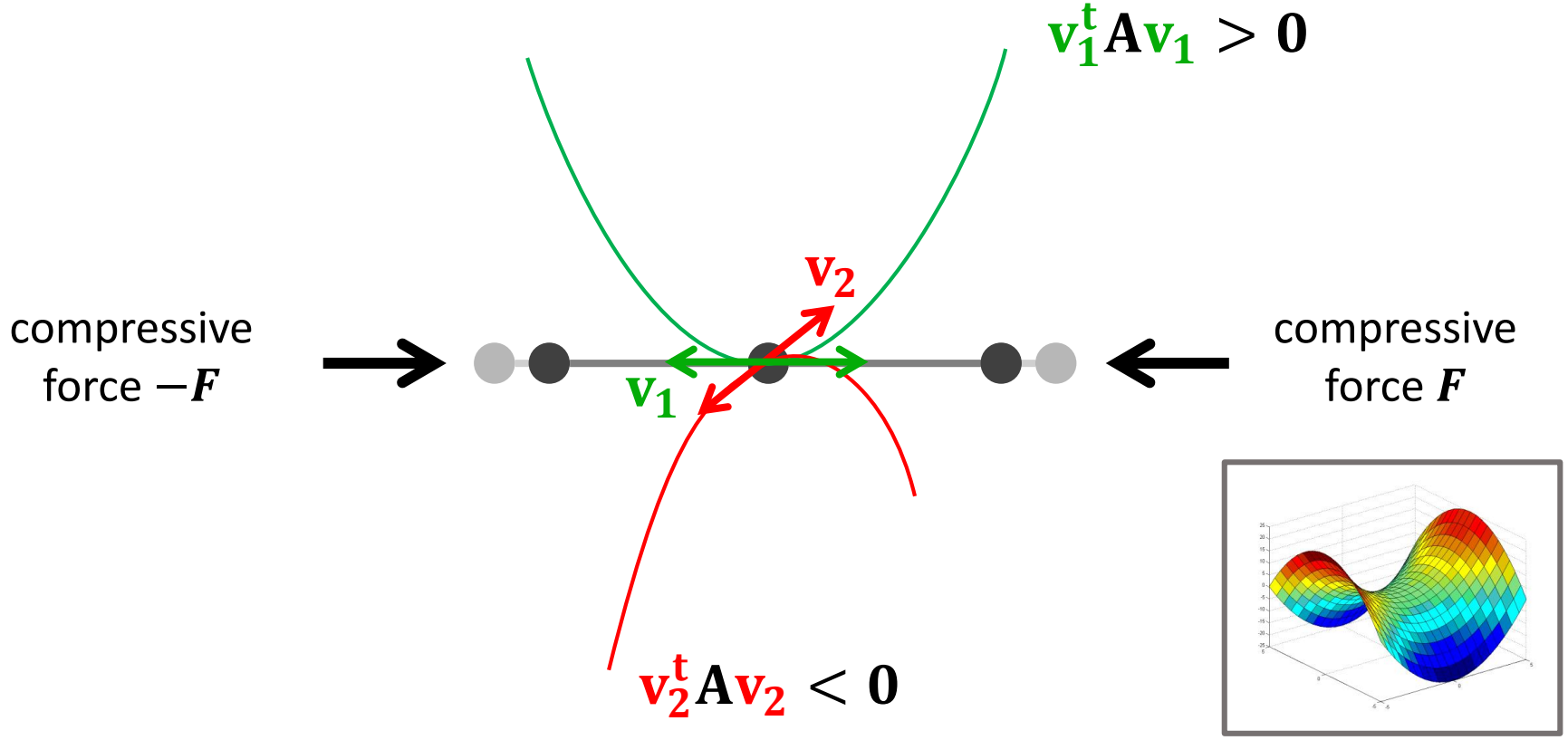
$\lambda_1, \lambda_2 < 0$

Indefinite



$\lambda_1 > 0, \lambda_2 < 0$

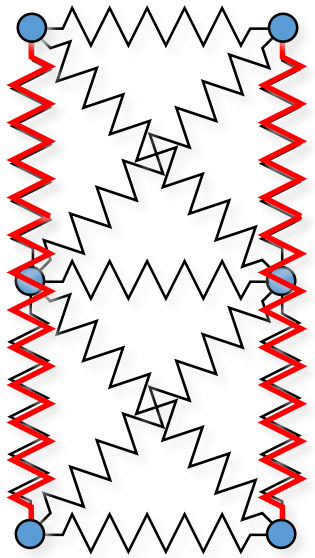
Indefinite systems: compressed springs



⇨ $\mathbf{A}$ is indefinite, we are at a saddle point!

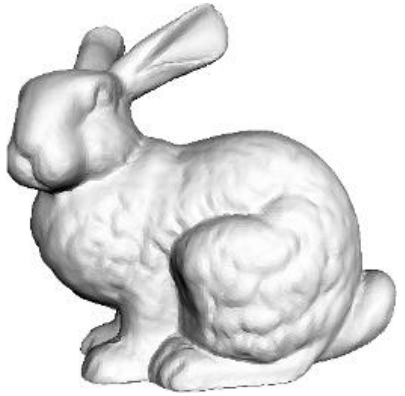
Mass Spring Systems

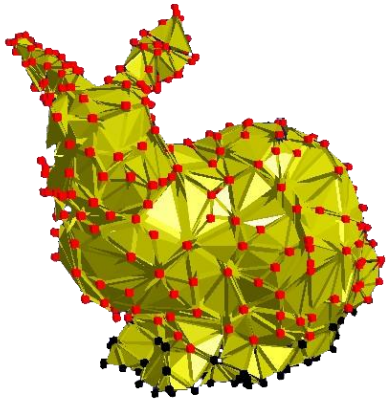
- Can be used to model arbitrary deformable objects, and are easy to understand and implement, but…
  - Behavior depends on mesh tessellation
    - Find good spring layout
    - Find good spring constants
    - Different types of springs interfere
  - Limited accuracy
    - No explicit volume or area preservation



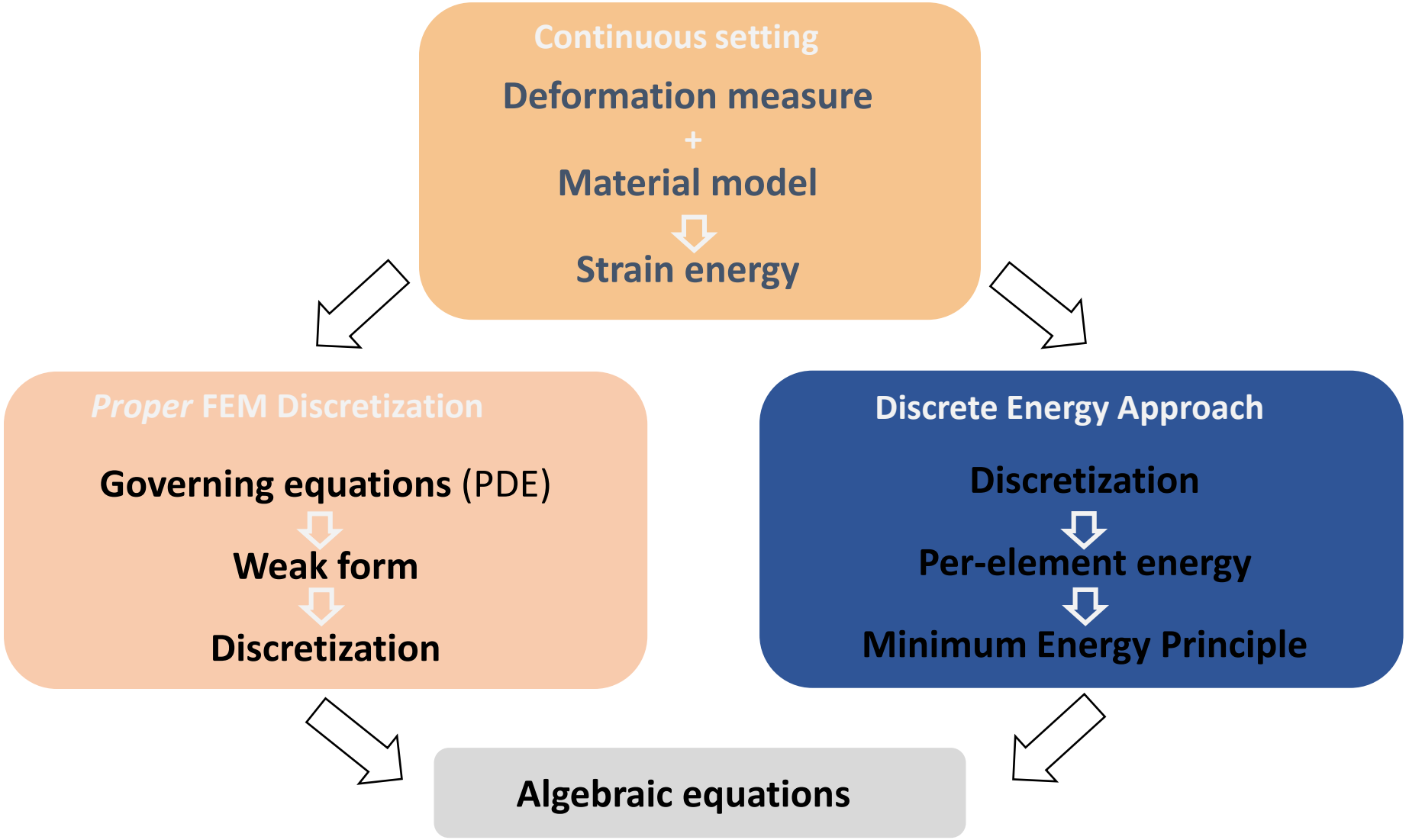
Alternative…

- Start from continuum mechanics principles
- Discretize with Finite Elements
  - Decompose model into simple elements
  - Setup & solve system of algebraic equations
- Advantages
  - Accurate and controllable material behavior
  - Largely independent of mesh structure





General Concept



Mass Spring vs Continuum Mechanics

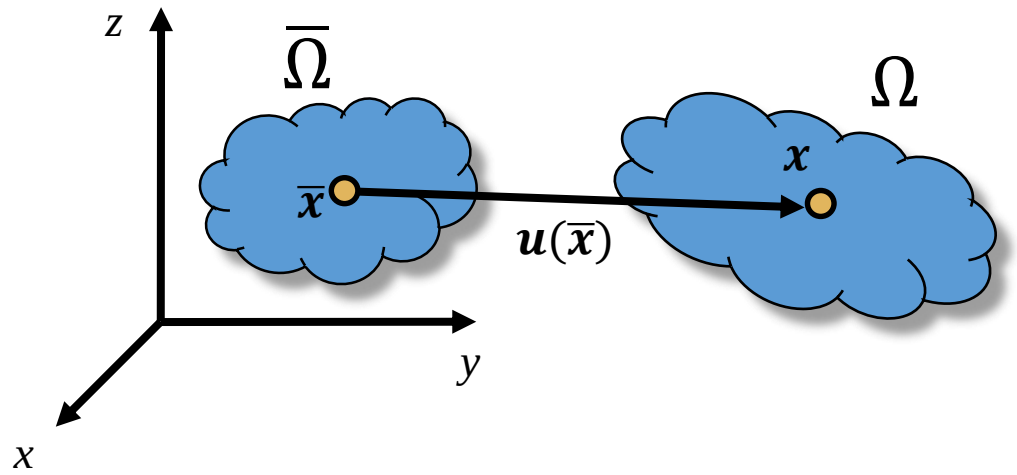
- Mass spring systems require:
  1. Measure of Deformation
  2. Material Model
  3. Deformation Energy
  4. Internal Forces

- Simulation approaches that start from continuum mechanics principles rely on conceptually identical pieces of information

Formalizing Continuous Deformations

- For a deformable body, keep track of the
  - undeformed state $\overline{\Omega} \subset \mathbf{R}^3$
  - deformed state $\Omega \subset \mathbf{R}^3$
- Displacement field $\boldsymbol{u}$ describes Ω in terms of $\overline{\Omega}$

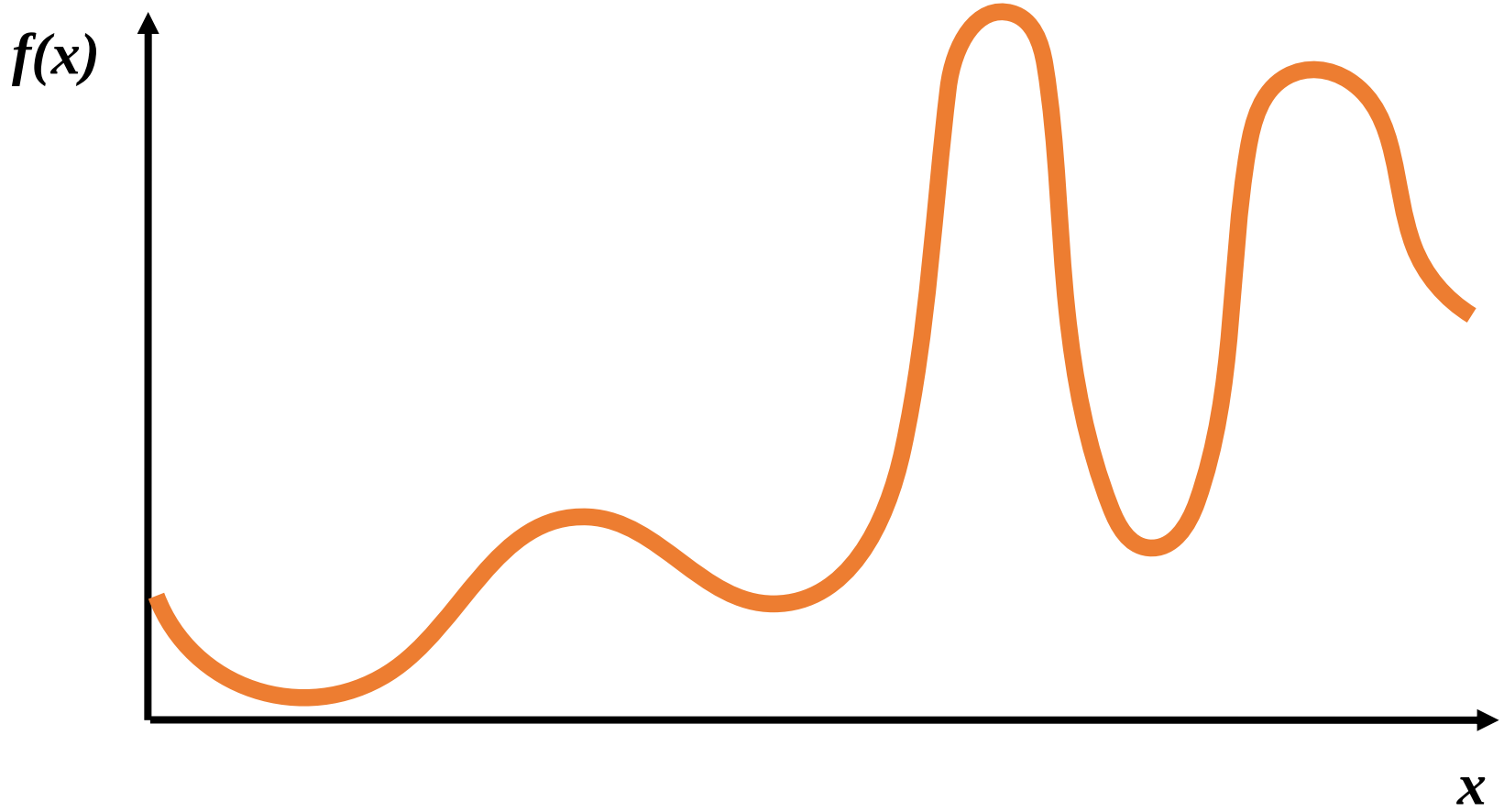
$$\boldsymbol{u}(\overline{\boldsymbol{x}}): \overline{\Omega} \rightarrow \Omega \ , \quad \boldsymbol{x}(\overline{\boldsymbol{x}}) = \overline{\boldsymbol{x}} + \boldsymbol{u}(\overline{\boldsymbol{x}})$$



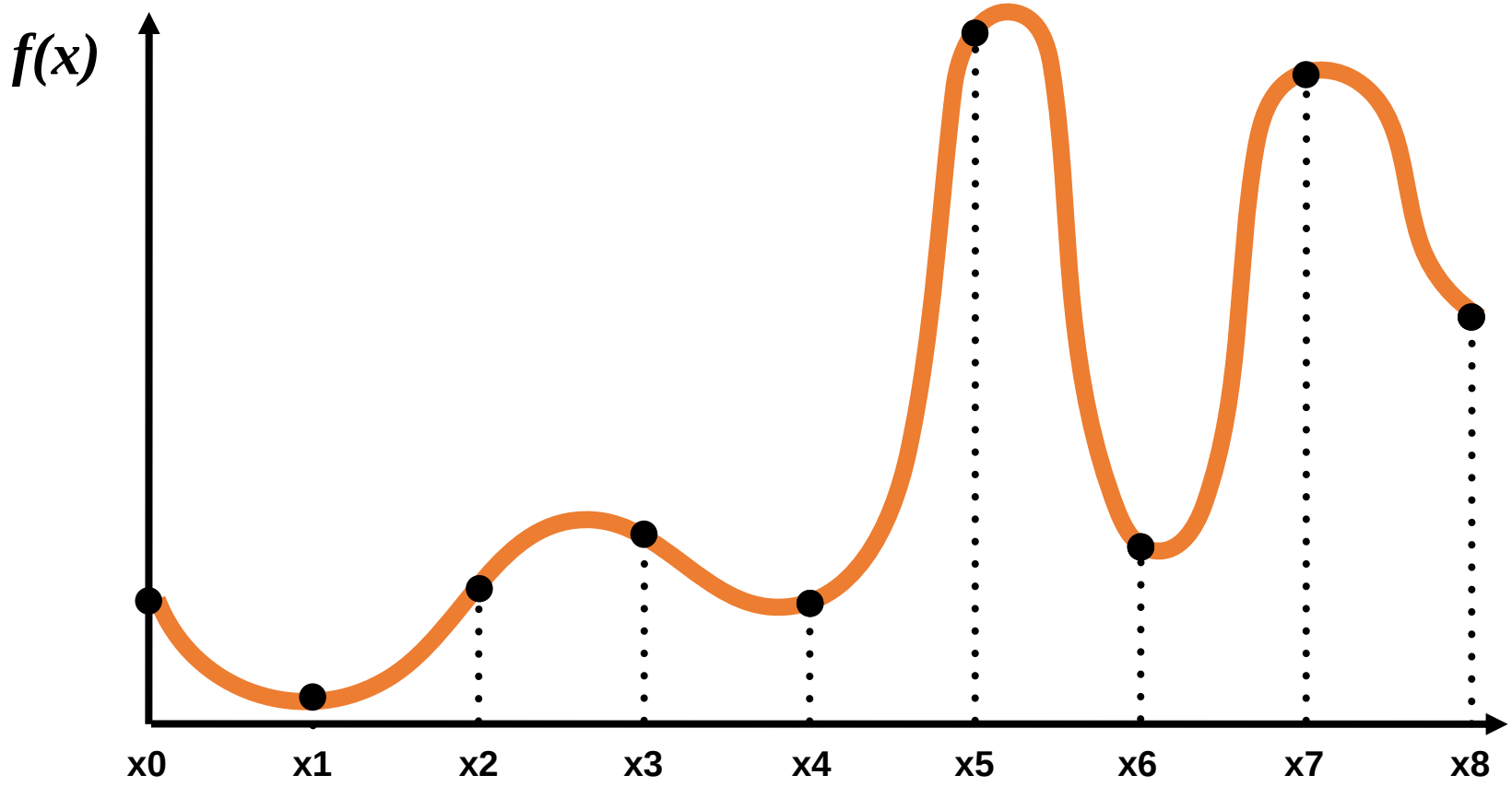
$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{pmatrix}$$

u is displacement in x direction
v is displacement in y direction
w is displacement in z direction

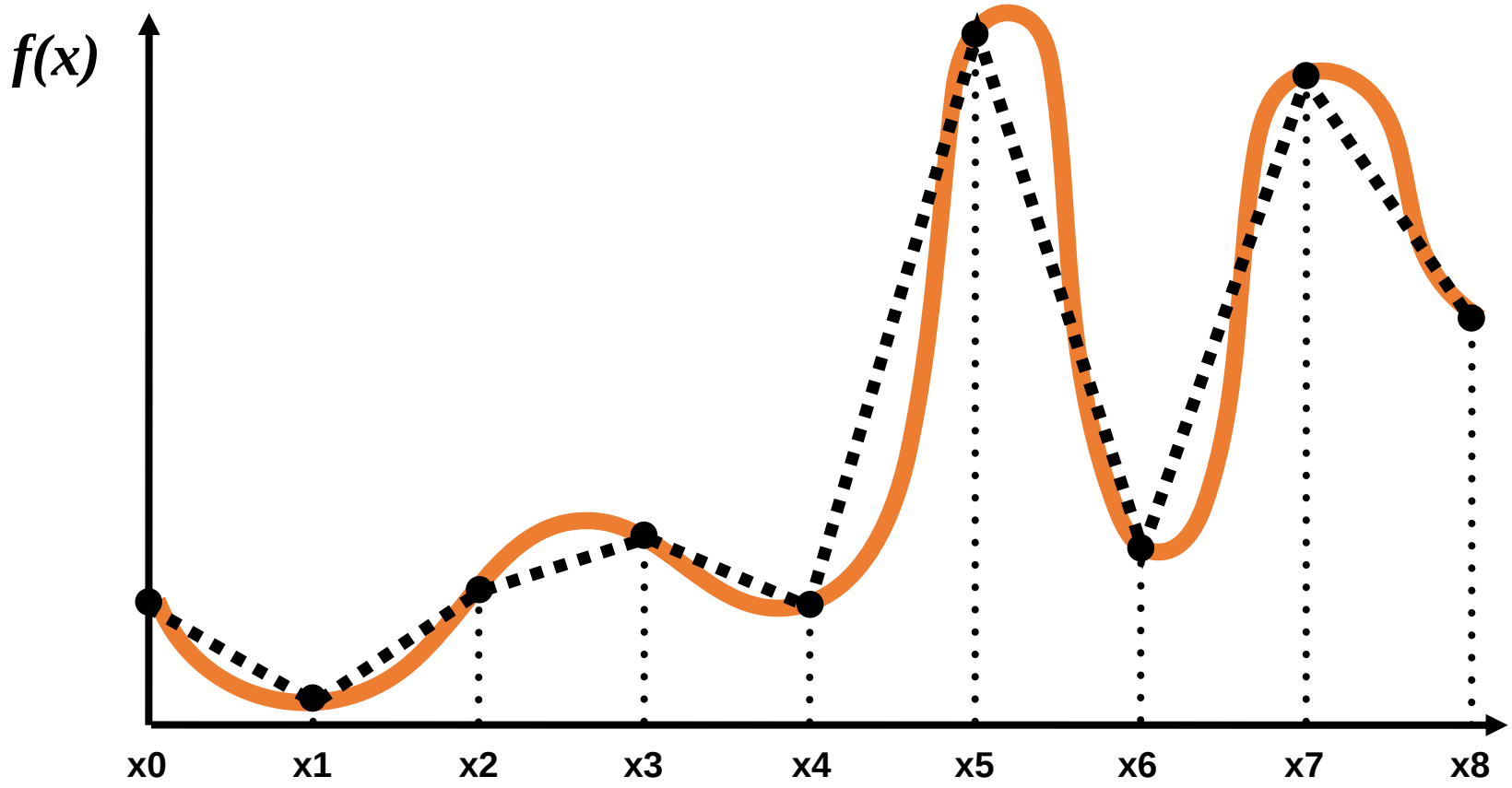
A continuous function



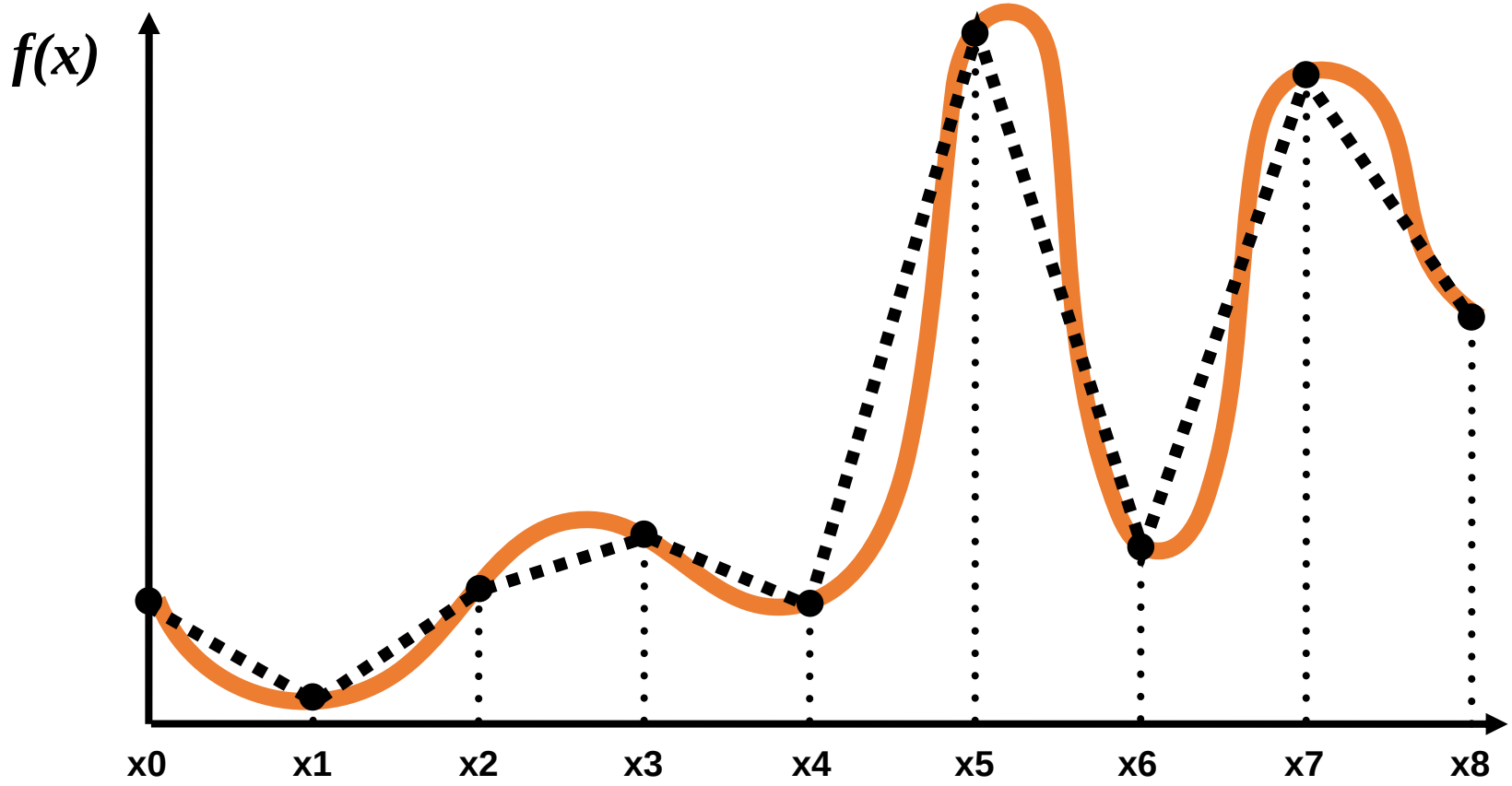
A continuous function – computer representation



A continuous function – computer representation



Let’s work it out in 1D



$$\mathbf{f}_{\text{approx}}(\mathbf{x}) = \sum N_i(\mathbf{x})\mathbf{f}(\mathbf{x}_i)$$

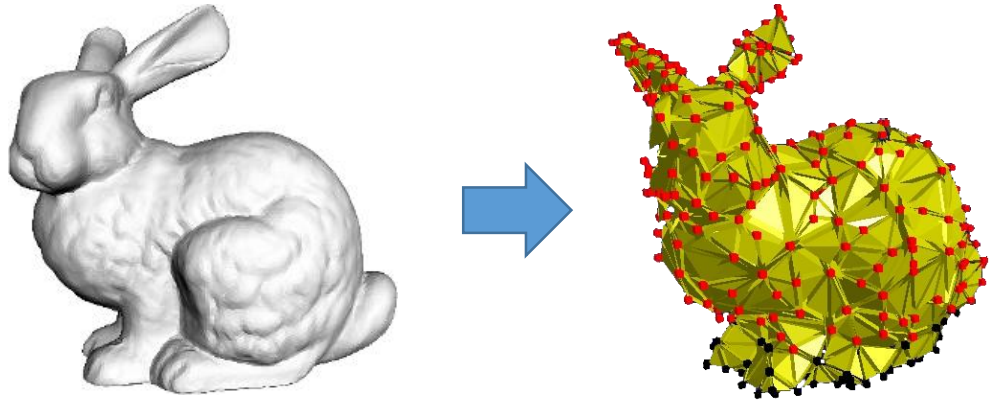
Approximating continuous functions

- Key idea: use basis functions and sampled function values to approximate continuous functions we care about (displacement field, deformed state, etc).
- For example:

$$\mathbf{x}(\bar{x}, \bar{y}, \bar{z}) = \sum N_i(\bar{x}, \bar{y}, \bar{z})\mathbf{x}_i$$

Finite element discretization

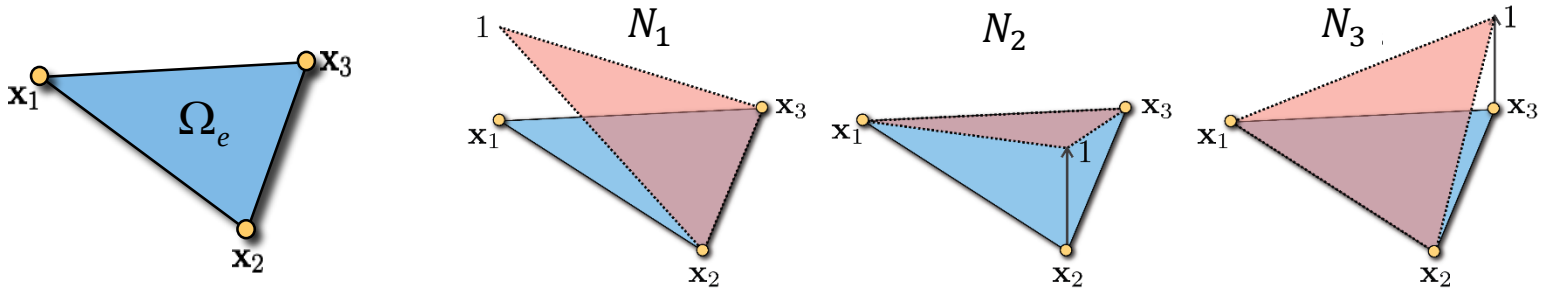
Decompose complex models into simple elements



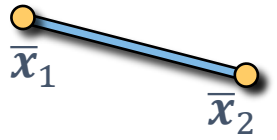
A finite element consists of

- a closed subset $\Omega_e \subset \boldsymbol{R}^d$ (*in d dimensions*)
- n nodal basis functions defined wrt undeformed domain $N_i: \overline{\Omega}_e \rightarrow \boldsymbol{R}$
- n vectors of nodal variables $\overline{\boldsymbol{x}}_i \in \boldsymbol{R}^d$ describing the reference geometry
- n vectors of degrees of freedom (e.g., deformed positions $\boldsymbol{x}_i$)

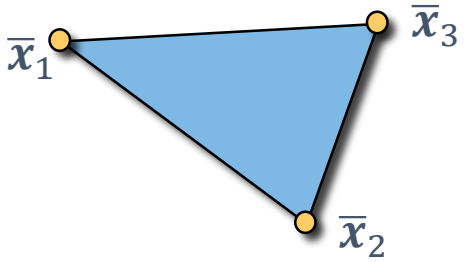
Example: 3-node triangle with linear basis functions



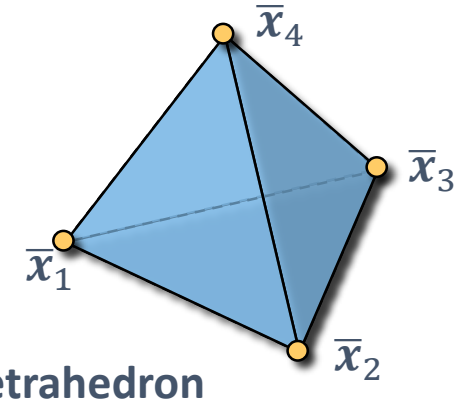
Linear Simplicial Elements



1D: **line segment**



2D: **triangle**



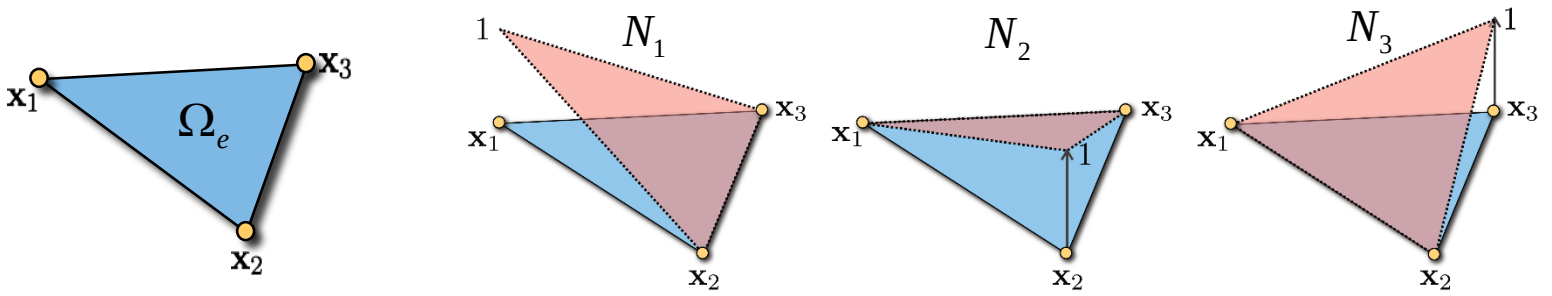
3D: **tetrahedron**

- Simplicial elements admit linear basis functions
- Basis functions are uniquely defined through
- reference geometry $\overline{\boldsymbol{x}}_i$ and
- interpolation requirement $N_i(\overline{\boldsymbol{x}}_j) = \delta_{ij}$

$\overline{\boldsymbol{x}}_i = \bar{x}_i$	in 1D
$\overline{\boldsymbol{x}}_i = (\bar{x}_i, \bar{y}_i)$	in 2D
$\overline{\boldsymbol{x}}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$	in 3D

Computing Basis Functions – 2D example

Example: 3-node elements with linear basis functions



- Basis functions are linear on element: $N_i(x,y) = a_i x + b_i y + c$
- Due to $N_i(\mathbf{x}_j) = \delta_{ij}$, we have

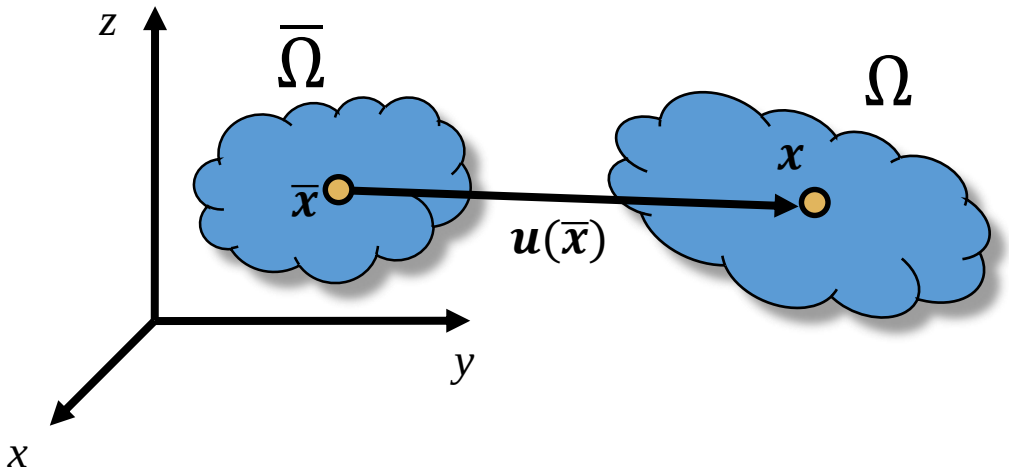
$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \end{bmatrix} \Rightarrow \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \end{bmatrix}$$

Reasoning about deformations

Displacement field $\boldsymbol{u}$ describes Ω (deformed configuration) in terms of $\overline{\Omega}$ (undeformed configuration): $\boldsymbol{u}$ tells us where each point in the material domain ends up in world coordinates

$$\boldsymbol{u}(\overline{\boldsymbol{x}}): \overline{\Omega} \rightarrow \Omega \ , \quad \boldsymbol{x}(\overline{\boldsymbol{x}}) = \overline{\boldsymbol{x}} + \boldsymbol{u}(\overline{\boldsymbol{x}})$$

Through interpolation, we need only keep track of the position of a finite set of points $\boldsymbol{x}$ to know (approximately) the deformed configuration of the entire simulation domain.

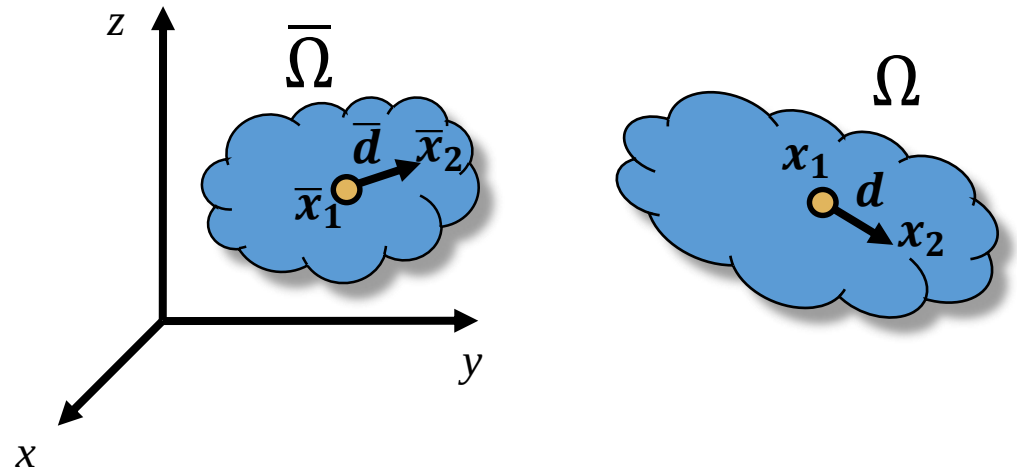


Reasoning about deformations

We know what happens to *points* in material space ($\boldsymbol{x}(\bar{\boldsymbol{x}}) = \bar{\boldsymbol{x}} + \boldsymbol{u}(\bar{\boldsymbol{x}})$)

But what happens to *vectors*?

Consider material points $\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_2$ and $\bar{\boldsymbol{d}} = (\bar{\boldsymbol{x}}_2 - \bar{\boldsymbol{x}}_1)$ such that $\bar{d}$ is infinitesimal. What is the corresponding deformed vector $\boldsymbol{d}$?



$$
\begin{aligned}
\boldsymbol{d} &= (\boldsymbol{x}_2 - \boldsymbol{x}_1) \\
&= \bar{\boldsymbol{x}}_2 + \boldsymbol{u}(\bar{\boldsymbol{x}}_2) - \bar{\boldsymbol{x}}_1 - \boldsymbol{u}(\bar{\boldsymbol{x}}_1) \\
&= \bar{\boldsymbol{d}} + \boldsymbol{u}\left(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{d}}\right) - \boldsymbol{u}(\bar{\boldsymbol{x}}_1) \\
&\approx \bar{\boldsymbol{d}} + \boldsymbol{u}(\bar{\boldsymbol{x}}_1) - \nabla \boldsymbol{u}\bar{\boldsymbol{d}} - \boldsymbol{u}(\bar{\boldsymbol{x}}_1) \\
&= \underbrace{(I + \nabla \boldsymbol{u})}_{\text{Deformation gradient } \mathbf{F}}\bar{\boldsymbol{d}}
\end{aligned}
$$

Displacement field and the deformation gradient

- Displacement field maps points in material space to points in world coordinates: $\boldsymbol{x}(\overline{\boldsymbol{x}}) = \overline{\boldsymbol{x}} + \boldsymbol{u}(\overline{\boldsymbol{x}})$

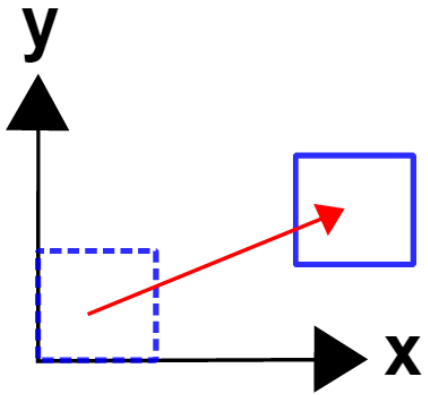
- Deformation gradient $\boldsymbol{F} = (\boldsymbol{I} + \nabla \boldsymbol{u})$ maps undeformed vectors to deformed vectors as $\boldsymbol{d} = \boldsymbol{F}\overline{\boldsymbol{d}}$

- Since $\boldsymbol{x}(\overline{\boldsymbol{x}}) = \overline{\boldsymbol{x}} + \boldsymbol{u}(\overline{\boldsymbol{x}})$ we can write

$$\boldsymbol{F} = (\boldsymbol{I} + \nabla \boldsymbol{u}) = \frac{\partial}{\partial \overline{\boldsymbol{x}}}\left(\overline{\boldsymbol{x}} + \boldsymbol{u}(\overline{\boldsymbol{x}})\right) = \frac{\partial \boldsymbol{x}}{\partial \overline{\boldsymbol{x}}}$$

Displacement Field and Deformation Gradient

- In general, displacement field is not explicitly described. Nevertheless, toy examples:

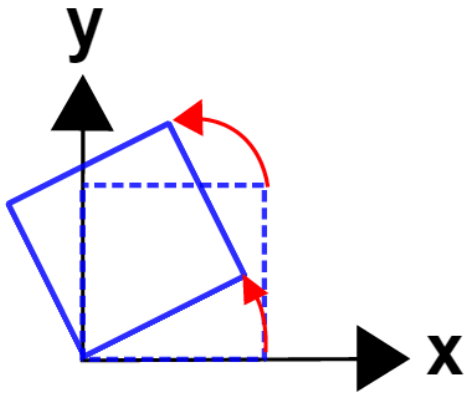


$$x = X + 5$$
$$y = Y + 2$$

$$\mathbf{F} = \mathbf{I}$$

Displacement Field and Deformation Gradient

- In general, displacement field is not explicitly described. Nevertheless, toy examples:

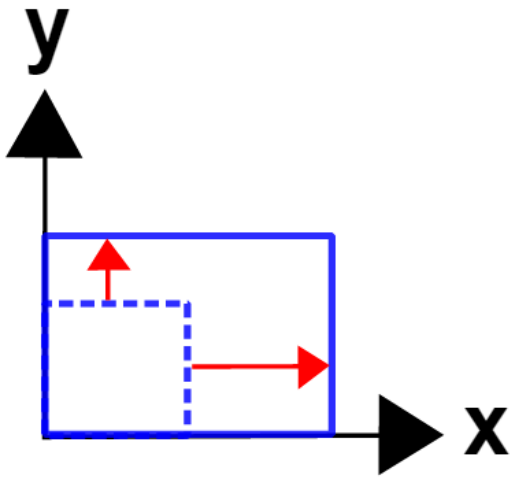


$$x = X\cos\theta - Y\sin\theta$$
$$y = X\sin\theta + Y\cos\theta$$

$$\mathbf{F} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Displacement Field and Deformation Gradient

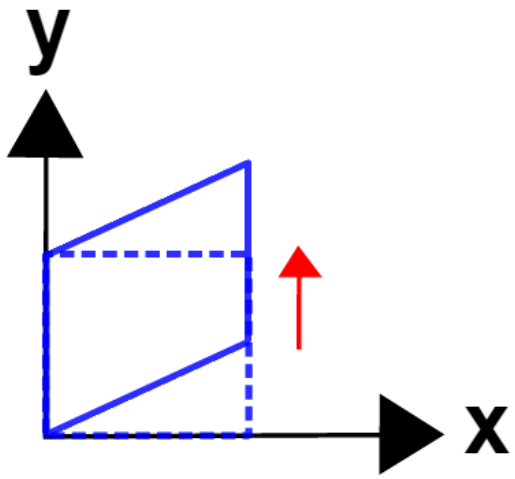
- In general, displacement field is not explicitly described. Nevertheless, toy examples:



$$x = 2.0X + 0.0\,Y$$
$$y = 0.0X + 1.5\,Y$$
$$\mathbf{F} = \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 1.5 \end{bmatrix}$$

Displacement Field and Deformation Gradient

- In general, displacement field is not explicitly described. Nevertheless, toy examples:

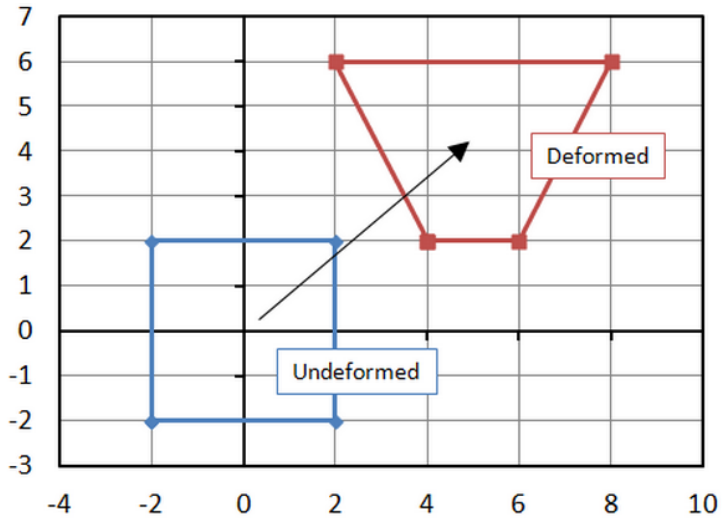


$$x = 1.0X + 0.0Y$$
$$y = 0.5X + 1.0Y$$

$$\mathbf{F} = \begin{bmatrix} 1.0 & 0.0 \\ 0.5 & 1.0 \end{bmatrix}$$

Displacement Field and Deformation Gradient

- In general, displacement field is not explicitly described. Nevertheless, toy examples:



$$x = X + \frac{1}{4}XY + 5$$

$$y = Y + 4$$

$$\mathbf{F} = \begin{bmatrix} 1 + \frac{1}{4}Y & \frac{1}{4}X \\ 0 & 1 \end{bmatrix}$$

Deformation Gradient and Finite Elements

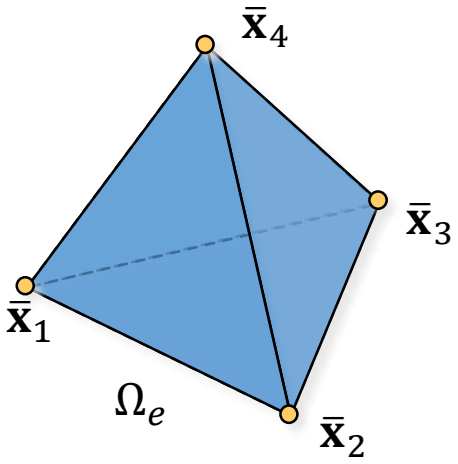
- Recall, we use basis functions to define continuous geometry of element as

$$\bar{\mathbf{x}}(\bar{x},\bar{y},\bar{z}) = \sum N_i(\bar{x},\bar{y},\bar{z})\bar{\mathbf{x}}_i \quad \text{and} \quad \mathbf{x}(\bar{x},\bar{y},\bar{z}) = \sum N_i(\bar{x},\bar{y},\bar{z})\mathbf{x}_i$$

- And we compute the deformation gradient as

$$\mathbf{F} = \frac{\partial \mathbf{x}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} = \sum_i \mathbf{x}_i \left(\frac{\partial N_i}{\partial \bar{\mathbf{x}}}\right)^t$$

- Notes:
  - In 3D, $\mathbf{F} \in \mathbf{R}^{3\times 3}$ and $\mathbf{F}$ is linear in $\mathbf{x}_i$
  - If N_i are linear, $\mathbf{F}$ is constant on element
  - Notation and confusion: $\frac{\partial N_i}{\partial \bar{\mathbf{x}}} \neq \frac{\partial N_i}{\partial \bar{\mathbf{x}}_i}$



Notation & Confusion

$$\mathbf{F} = \frac{\partial \mathbf{x}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} = \sum_i \mathbf{x}_i \left(\frac{\partial N_i}{\partial \bar{\mathbf{x}}} \right)^t \qquad \frac{\partial N_i}{\partial \bar{\mathbf{x}}} \neq \frac{\partial N_i}{\partial \bar{\mathbf{x}}_i}$$

Continuous case:

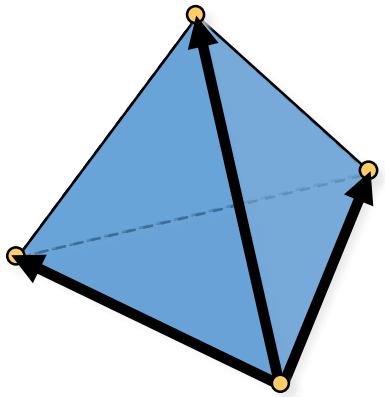
- Undeformed configuration $\bar{\boldsymbol{x}}(\bar{x}, \bar{y}) = (\bar{x}, \bar{y})^T$

Discretized:

- Undeformed configuration $\bar{\boldsymbol{x}}(\bar{x}, \bar{y}) = \sum_i N_i(\bar{x}, \bar{y}) \bar{\boldsymbol{x}}_i$
- Deformed configuration $\boldsymbol{x}(\bar{x}, \bar{y}) = \sum_i N_i(\bar{x}, \bar{y}) \boldsymbol{x}_i$
- With, for example, $N_i(\bar{\boldsymbol{x}}) = N_i(\bar{x}, \bar{y}) = a_i \bar{x} + b_i \bar{y} + c_i$
- Linear coefficients (a_i, b_i, c_i) are a function of $\bar{\boldsymbol{x}}_i$, but it's not what $\frac{\partial N_i}{\partial \bar{\mathbf{x}}}$ represents

Another way to compute the deformation gradient, if constant throughout element (i.e. linear basis functions)

- Deformation gradient maps undeformed vectors to deformed vectors
- Choose d vectors (e.g. element edge vectors) for which the map is known, assemble in matrix form and solve for $\mathbf{F}$: $\mathbf{F} = \boldsymbol{e}\bar{\boldsymbol{e}}^{-1}$



Interpreting F

- Polar decomposition $\boldsymbol{F} = \boldsymbol{RU}$, with $\boldsymbol{R}$ orthonormal and $\boldsymbol{U}$ positive definite
- If $\boldsymbol{F}$ is non-singular, i.e., $\det \boldsymbol{F} \neq 0$, then its PD exists and is unique.

How to compute the polar decomposition?

- Start from an SVD: $\boldsymbol{F} = \boldsymbol{P\Sigma Q}^T$ with $\boldsymbol{\Sigma}$ diagonal and $\boldsymbol{P}, \boldsymbol{Q}$ orthonormal
- Then $\boldsymbol{F} = \boldsymbol{P}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\Sigma Q}^T = \boldsymbol{RU}$ where $\boldsymbol{R} = \boldsymbol{PQ}^T$ is orthonormal (pure rotation) and $\boldsymbol{U} = \mathbf{Q}\boldsymbol{\Sigma Q}^T$ is positive definite (pure deformation)

Interpreting F

Singular Value Decomposition $\boldsymbol{F = P\Sigma Q^T}$

- Recall that $\boldsymbol{F}$ transforms vectors from undeformed to deformed space as $\boldsymbol{d = F\bar{d}}$
- Consider sequence of transformations for $\boldsymbol{F = P\Sigma Q^T = RQ\Sigma Q^T}$
  - $\boldsymbol{d^1 = Q^T\,\bar{d}}$: $\boldsymbol{Q^T}$ rotates from undeformed (material) space to intermediate space of principal stretches → preserves length.
  - $\boldsymbol{d^2 = \Sigma d^1}$: $\boldsymbol{\Sigma}$ scales components of $\boldsymbol{d^1}$ → changes length.
  - $\boldsymbol{d^3 = Q d^2}$: $\boldsymbol{Q}$ transforms $\boldsymbol{d^2}$ back to undeformed space → preserves length.
  - $\boldsymbol{d = R d^3}$: $\boldsymbol{R}$ rotates from undeformed to deformed space → preserves length.

Interpreting F

- Consider unit tetrahedron with
  - Vertices: $\bar{\boldsymbol{x}}_1 = (0,0,0)^T, \bar{\boldsymbol{x}}_2 = (1,0,0)^T, \bar{\boldsymbol{x}}_3 = (0,1,0)^T, \bar{\boldsymbol{x}}_4 = (0,0,1)^T$
  - Edges: $\boldsymbol{e}_{21} = (1,0,0)^T, \boldsymbol{e}_{31} = (0,1,0)^T, \boldsymbol{e}_{41} = (0,0,1)^T$
- Volume of undeformed tetrahedron is defined through scalar triple product

$$\bar{V} = \frac{1}{6}(\boldsymbol{e}_{21} \times \boldsymbol{e}_{31})^T \boldsymbol{e}_{41}$$

- Consider deformed tetrahedron with edges $\boldsymbol{F}\boldsymbol{e}_{i1}$. Then

$$V = \frac{1}{6}(\boldsymbol{F}\boldsymbol{e}_{21} \times \boldsymbol{F}\boldsymbol{e}_{31})^T \boldsymbol{F}\boldsymbol{e}_{41} = \frac{1}{6}(\boldsymbol{F}_{:1} \times \boldsymbol{F}_{:2})^T \boldsymbol{F}_{:3} = \frac{1}{6}\det \boldsymbol{F}$$

- The determinant of $\boldsymbol{F}$ indicates the change in volume, i.e.,

$$V = \det \boldsymbol{F}\, \bar{V}$$

Modeling deformations

Deformation gradient tells us how vectors change, but how can we describe deformations (strain) at any given material point?

Back to spring deformation

- Deformation measure (strain): $\left(\frac{l}{l_0} - 1\right)$

Kind of like F

- Undeformed spring: $\frac{l}{l_0} = 1$
- Undeformed* infinitesimal continuum volume: $\mathbf{F} = \mathbf{I}$

Deformation

- Deformation measure (strain): $\left(\frac{l}{l_0} - 1\right)$
- Desirable property: if spring is undeformed, strain is 0
- Can we find a similar measure that would work for an arbitrary volume ?

Deformation

- Deformation gradient $\boldsymbol{F} = (\boldsymbol{I} + \nabla \boldsymbol{u})$ maps undeformed vectors to deformed vectors as $\boldsymbol{d} = \boldsymbol{F}\overline{\boldsymbol{d}}$

- Deformation measure should capture change in length squared for any and all directions:

$$\boldsymbol{d}^T\boldsymbol{d} - \overline{\boldsymbol{d}}^T\overline{\boldsymbol{d}} = \overline{\boldsymbol{d}}^T(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I})\overline{\boldsymbol{d}}$$

- Green strain

$$E = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}) = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T + \nabla \boldsymbol{u}^T \nabla \boldsymbol{u})$$

Green Strain and Cauchy Strain

- Green strain is quadratic in displacements

$$E = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T + \nabla\boldsymbol{u}^T\nabla\boldsymbol{u})$$

- Neglecting quadratic terms leads to the linear Cauchy strain

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{t}}) = \frac{1}{2}(\mathbf{F} + \mathbf{F}^t) - \mathbf{I}$$

Note:

- both Cauchy and Green strain are invariant under translation
- Green strain is invariant under rotation, but Cauchy strain is not (what happens for a new deformation gradient $\boldsymbol{F'} = \boldsymbol{RF}$, obtained just by rotating an already deformed configuration? What does this mean?)

3D Linear Strain

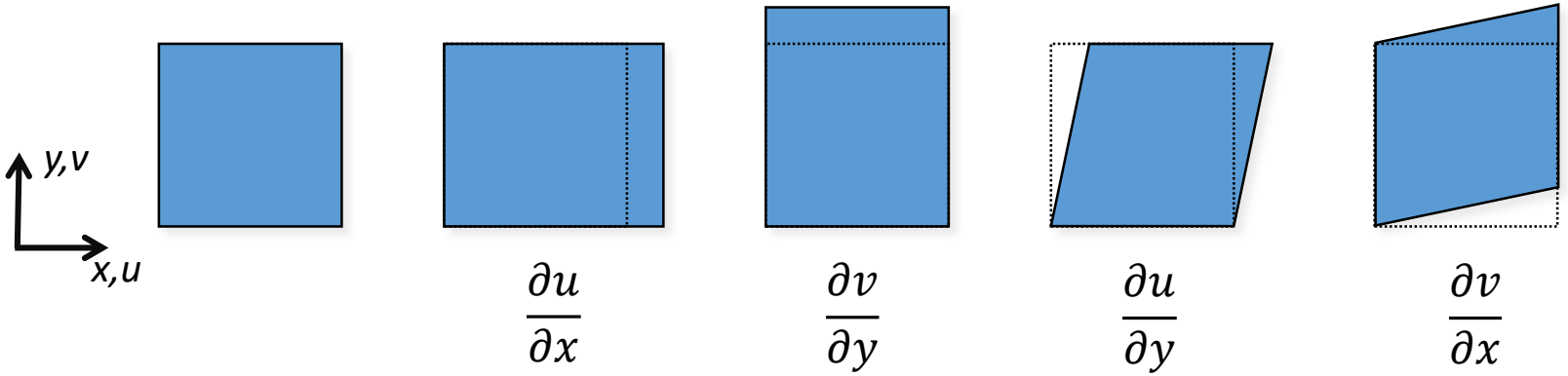
- Linear Cauchy strain

$$\varepsilon = \frac{1}{2}\begin{pmatrix} 2\partial_x u & \partial_y u + \partial_x v & \partial_z u + \partial_x w \\ \partial_x v + \partial_y u & 2\partial_y v & \partial_z v + \partial_y w \\ \partial_x w + \partial_z u & \partial_y w + \partial_z v & 2\partial_z w \end{pmatrix} =: \begin{pmatrix} \varepsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{xy} & \varepsilon_y & \gamma_{yz} \\ \gamma_{xz} & \gamma_{yz} & \varepsilon_z \end{pmatrix}$$

ε_i : normal strains

γ_i : shear strains

- Geometric interpretation



Mass Spring vs Continuum Mechanics

- Mass spring systems:
  1. Measure of Deformation $\left(\frac{l}{l_0} - 1\right)$
  2. Material Model
  3. Deformation Energy
  4. Internal Forces
- Continuum Mechanics:
  1. Measure of Deformation: Green or Cauchy strain
  2. Material Model

Material Model

- Material model links strain to deformation energy (and also to stresses/internal forces)
- e.g. linear isotropic material (*generalized Hooke's law*)
  - Energy density $\Psi = \frac{1}{2}\lambda \mathrm{tr}(\boldsymbol{\varepsilon})^2 + \mu \mathrm{tr}(\boldsymbol{\varepsilon}^2)$

$$\mathrm{tr}(\boldsymbol{\varepsilon}) = \sum \varepsilon_{ii}$$

  - Lame parameters λ and μ are material constants related to fundamental physical parameters, the Poisson Ratio and Young's modulus

Material Parameters

V · T · E **Elastic moduli for homogeneous isotropic materials** [hide]

Bulk modulus (K) · Young's modulus (E) · **Lamé's first parameter** (λ) · Shear modulus (G, μ) · Poisson's ratio (ν) · P-wave modulus (M)

Conversion formulas [hide]

Homogeneous isotropic linear elastic materials have their elastic properties uniquely determined by any two moduli among these; thus, given any two, any other of the elastic moduli can be calculated according to these formulas.

	$K =$	$E =$	$\lambda =$	$G =$	$\nu =$	$M =$	Notes
$(K,\ E)$	K	E	$\frac{3K(3K-E)}{9K-E}$	$\frac{3KE}{9K-E}$	$\frac{3K-E}{6K}$	$\frac{3K(3K+E)}{9K-E}$	
$(K,\ \lambda)$	K	$\frac{9K(K-\lambda)}{3K-\lambda}$	λ	$\frac{3(K-\lambda)}{2}$	$\frac{\lambda}{3K-\lambda}$	$3K-2\lambda$	
$(K,\ G)$	K	$\frac{9KG}{3K+G}$	$K-\frac{2G}{3}$	G	$\frac{3K-2G}{2(3K+G)}$	$K+\frac{4G}{3}$	
$(K,\ \nu)$	K	$3K(1-2\nu)$	$\frac{3K\nu}{1+\nu}$	$\frac{3K(1-2\nu)}{2(1+\nu)}$	ν	$\frac{3K(1-\nu)}{1+\nu}$	
$(K,\ M)$	K	$\frac{9K(M-K)}{3K+M}$	$\frac{3K-M}{2}$	$\frac{3(M-K)}{4}$	$\frac{3K-M}{3K+M}$	M	
$(E,\ \lambda)$	$\frac{E+3\lambda+R}{6}$	E	λ	$\frac{E-3\lambda+R}{4}$	$\frac{2\lambda}{E+\lambda+R}$	$\frac{E-\lambda+R}{2}$	$R=\sqrt{E^2+9\lambda^2+2E\lambda}$
$(E,\ G)$	$\frac{EG}{3(3G-E)}$	E	$\frac{G(E-2G)}{3G-E}$	G	$\frac{E}{2G}-1$	$\frac{G(4G-E)}{3G-E}$	
$(E,\ \nu)$	$\frac{E}{3(1-2\nu)}$	E	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{E}{2(1+\nu)}$	ν	$\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$	
$(E,\ M)$	$\frac{3M-E+S}{6}$	E	$\frac{M-E+S}{4}$	$\frac{3M+E-S}{8}$	$\frac{E-M+S}{4M}$	M	$S=\pm\sqrt{E^2+9M^2-10EM}$ There are two valid solutions. The plus sign leads to $\nu \geq 0$. The minus sign leads to $\nu \leq 0$.
$(\lambda,\ G)$	$\lambda+\frac{2G}{3}$	$\frac{G(3\lambda+2G)}{\lambda+G}$	λ	G	$\frac{\lambda}{2(\lambda+G)}$	$\lambda+2G$	
$(\lambda,\ \nu)$	$\frac{\lambda(1+\nu)}{3\nu}$	$\frac{\lambda(1+\nu)(1-2\nu)}{\nu}$	λ	$\frac{\lambda(1-2\nu)}{2\nu}$	ν	$\frac{\lambda(1-\nu)}{\nu}$	Cannot be used when $\nu=0 \Leftrightarrow \lambda=0$
$(\lambda,\ M)$	$\frac{M+2\lambda}{3}$	$\frac{(M-\lambda)(M+2\lambda)}{M+\lambda}$	λ	$\frac{M-\lambda}{2}$	$\frac{\lambda}{M+\lambda}$	M	
$(G,\ \nu)$	$\frac{2G(1+\nu)}{3(1-2\nu)}$	$2G(1+\nu)$	$\frac{2G\nu}{1-2\nu}$	G	ν	$\frac{2G(1-\nu)}{1-2\nu}$	
$(G,\ M)$	$M-\frac{4G}{3}$	$\frac{G(3M-4G)}{M-G}$	$M-2G$	G	$\frac{M-2G}{2M-2G}$	M	
$(\nu,\ M)$	$\frac{M(1+\nu)}{3(1-\nu)}$	$\frac{M(1+\nu)(1-2\nu)}{1-\nu}$	$\frac{M\nu}{1-\nu}$	$\frac{M(1-2\nu)}{2(1-\nu)}$	ν	M	

https://en.wikipedia.org/wiki/Lam%C3%A9_parameters

Material Model

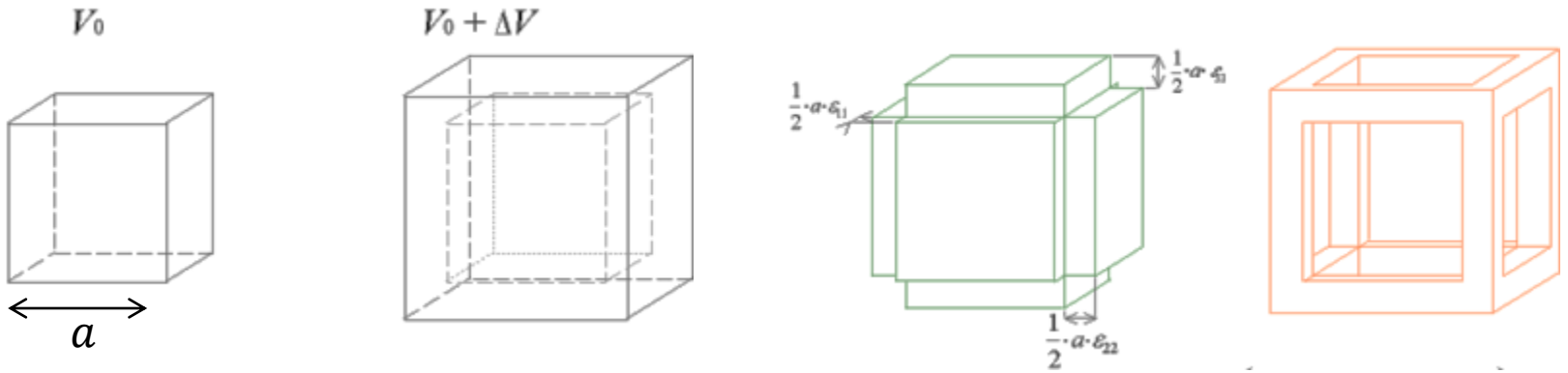
- Material model links strain to deformation energy
- E.g. linear isotropic material (*generalized Hooke's law*)
  - Energy density $\Psi = \frac{1}{2}\lambda \mathrm{tr}(\boldsymbol{\varepsilon})^2 + \mu \mathrm{tr}(\boldsymbol{\varepsilon}^2)$

    $\mathrm{tr}(\boldsymbol{\varepsilon}) = \sum \varepsilon_{ii}$
  - Lame parameters λ and μ are material constants
  - Obtain deformation energy by integrating energy density over element: $W^e = \int_{\Omega_e} \Psi$. If we are using linear basis functions (e.g. linear elements), then $\mathbf{F}$ is constant, so Ψ is constant and $W^e = \Psi(\mathbf{F}) \cdot \bar{V}^e$
- Interpretation
  - $\mathrm{tr}(\boldsymbol{\varepsilon}^2) = \mathrm{tr}(\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}) = \|\boldsymbol{\varepsilon}\|_F^2$ penalizes all strain components equally
  - $\lambda \mathrm{tr}(\boldsymbol{\varepsilon})^2$ penalizes dilatations, i.e., volume changes

Volumetric Strain

- Consider a cube with side length a
- For a given deformation $\boldsymbol{\varepsilon}$, the added volume is

$$\Delta V = a(1+\varepsilon_{11}) \cdot a(1+\varepsilon_{22}) \cdot a(1+\varepsilon_{33}) - a^3$$
$$= a^3(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33}) + O(\boldsymbol{\varepsilon}^2) \approx a^3 \mathrm{tr}(\boldsymbol{\varepsilon})$$



http://en.wikipedia.org/wiki/Infinitesimal_strain_theory

Solving Statics Problems

- Necessary condition for static equilibrium

$$\mathbf{f}_i(\mathbf{x}) = \mathbf{f}_i^{ext} + \mathbf{f}_i^{el}(\mathbf{x}) = 0 \; \forall i$$

- Given $\mathbf{x}$ and $\mathbf{f}(\mathbf{x})$, find $\Delta\mathbf{x}$ such that $\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{0}$
- First order approximation → $\mathbf{K}(\mathbf{x})\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})$

$$\mathbf{f}_i^{el} = -\frac{\partial W}{\partial \boldsymbol{x}_i}$$

Stiffness matrix

$$\mathbf{K} = \frac{\partial \mathbf{f}^{el}}{\partial \mathbf{x}}$$

Newton's method

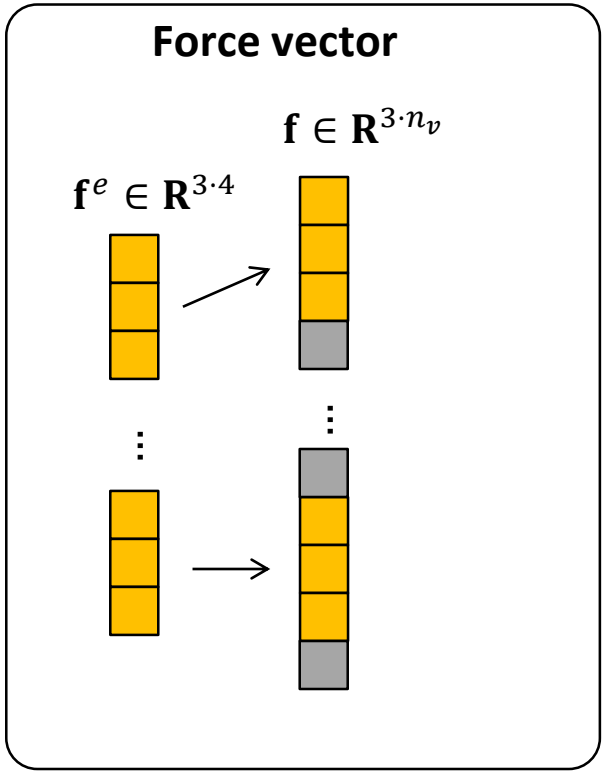
While not converged

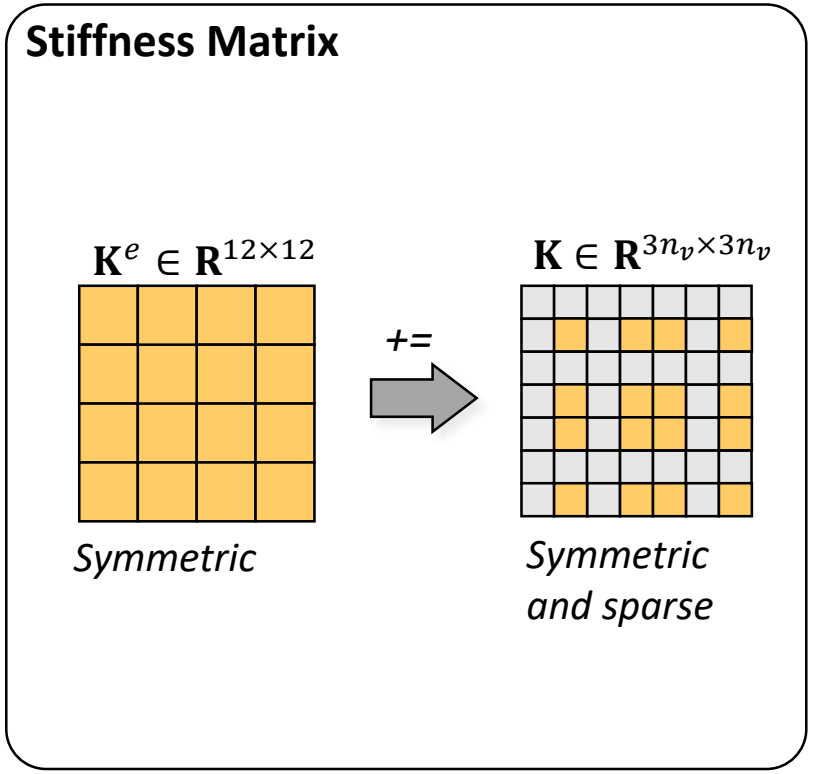
Compute	$\mathbf{f}(\mathbf{x}), \mathbf{K}(\mathbf{x})$
Solve	$\mathbf{K}(\mathbf{x})\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})$
line search	$\alpha = \text{linesearch}(\mathbf{x}, \Delta\mathbf{x})$
Update	$\mathbf{x} \mathrel{+}= \alpha\Delta\mathbf{x}$

end

Solving The Static Equilibrium Problem

Assemble element contributions into global vector and matrix

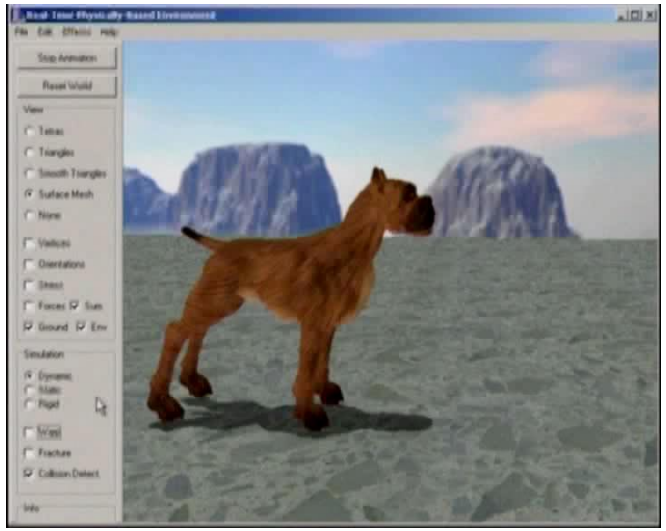




Linear Elasticity – Behavior

- For linear elasticity, W is quadratic in $\mathbf{x}$, $\mathbf{f}$ is linear in $\mathbf{x}$, and $\frac{\partial^2 W}{\partial \boldsymbol{x}^2}$ is constant → only solve one linear system for static equilibrium

- Problem: Cauchy strain is not invariant under rotations → inaccuracies for large rotations deformations
- Solution: use nonlinear deformation measure (e.g. Green strain instead of Cauchy strain)



Nonlinear Elasticity

- Idea: replace Cauchy strain with Green strain

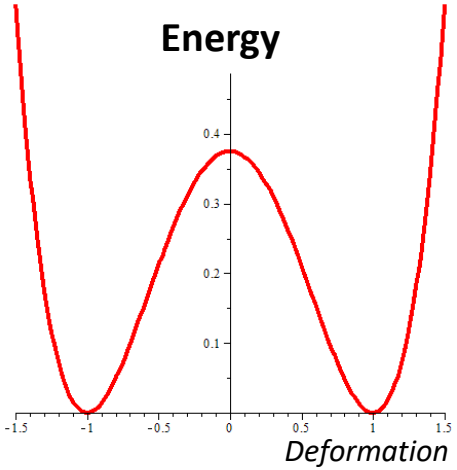
  → *St. Venant-Kirchhoff material* (StVK)

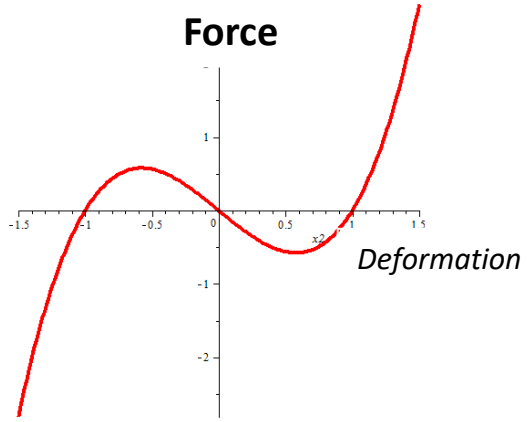
- Energy $\Psi_{StVK} = \frac{1}{2}\lambda \mathrm{tr}(\mathbf{E})^2 + \mu \mathrm{tr}(\mathbf{E}^2)$

- Note:
  - Energy is quartic in $\mathbf{x}$, forces are cubic
  - Green strain is rotation invariant
  - Solve system of nonlinear equations

StVK Limitations

- *Problem*: StVK softens under compression





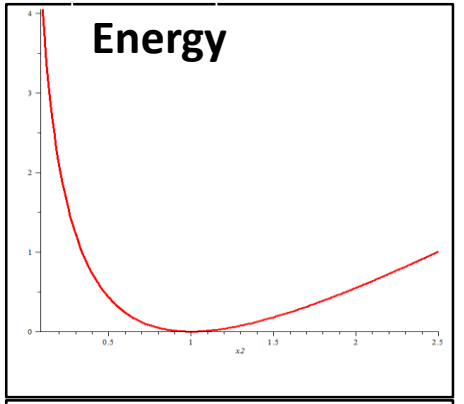
- Once inverted, elements are happy to stay there…
- *Work around*: add volume term

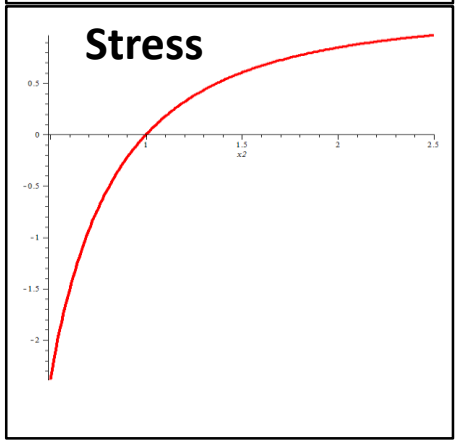
$$\Psi_{StVK} = \frac{\lambda}{2}\mathrm{tr}(\mathbf{E})^2 + \mu\mathrm{tr}(\mathbf{E}^2) \quad \rightarrow \quad \Psi_{Mod} = \eta(\det(\mathbf{F}) - 1)^2 + \mu\mathrm{tr}(\mathbf{E}^2)$$

Advanced nonlinear materials

- Green Strain $\mathbf{E} = \frac{1}{2}(\mathbf{F}^t\mathbf{F} - \mathbf{I}) = \frac{1}{2}(\mathbf{C} - \mathbf{I})$
- Split into *deviatoric (i.e. volume-preserving shape changing/distortion)* and *volumetric (dilation, volume changing)* deformations

  Volumetric: $J = \det(\mathbf{F})$ Deviatoric: $\mathbf{C} = \mathbf{F}^t\mathbf{F}$





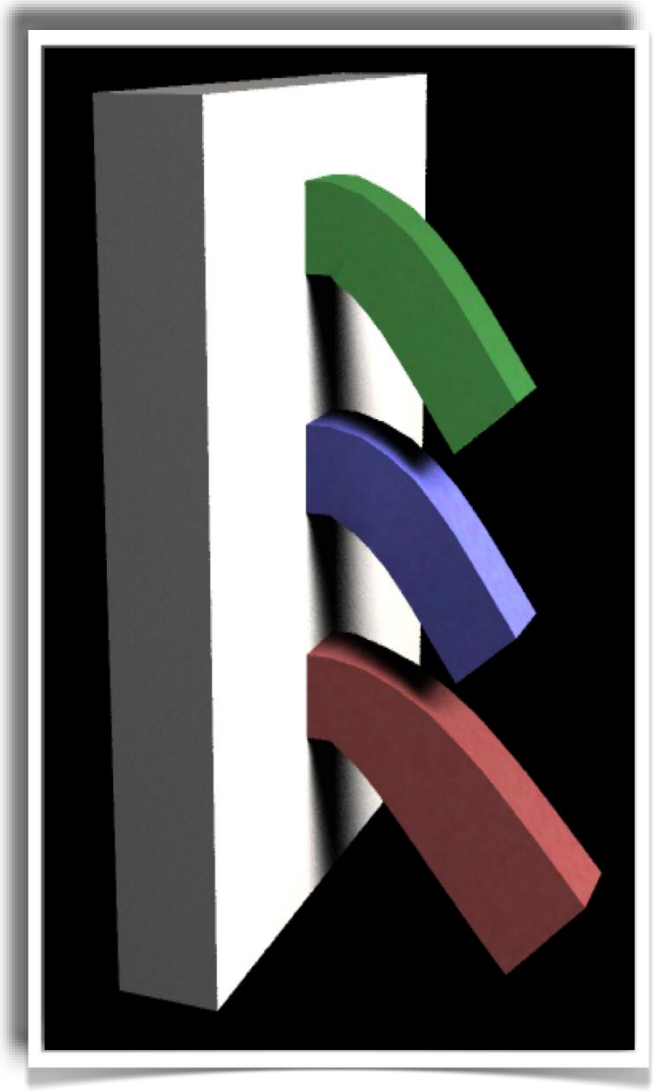
- *Compressible* Neo-Hookean material:

$$\Psi_{NH} = \frac{\mu}{2}(\mathrm{tr}(\mathbf{C}) - 3) - \mu \ln J + \frac{\lambda}{2}\ln(J)^2$$

Observations:

- the first term penalizes all deformations equally (since $\mathrm{tr}(\boldsymbol{C}) = |\boldsymbol{F}|_F^2$)
- the third term goes to infinity for increasing compression (*faster than the second*)
- the stress-strain behavior is initially linear, but goes into plateau for larger deformations
- Rule of thumb: NH is good for deformations of up to 20%

Different Models



St. Venant-Kirchoff

Neo-Hookean

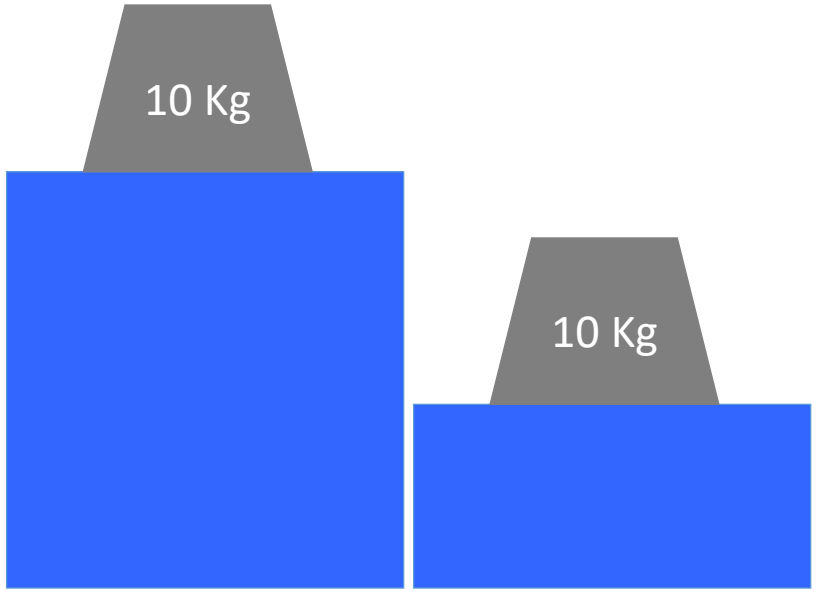
Linear

FEM recipe

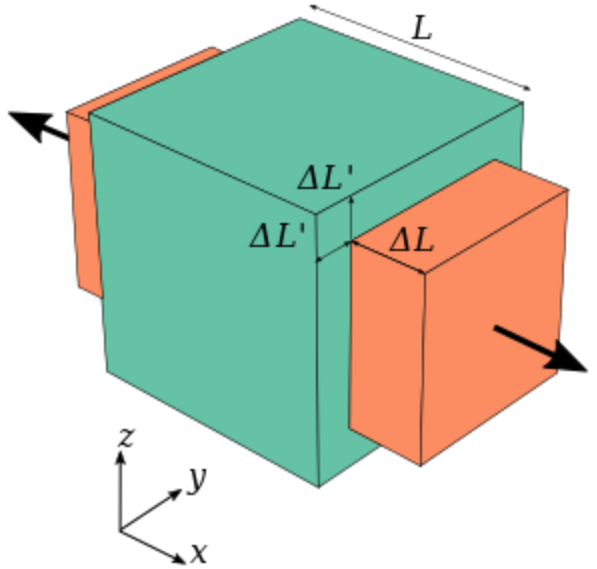
- Discretize into elements (triangles/tetraderons, etc)
- For each element
  - Compute deformation gradient
  - Use material model to define energy density $\Psi(\mathbf{F})$
  - Integrate over elements to compute energy: *W*
  - Compute nodal forces as the negative gradient of deformation energy

Young's Modulus and Poisson Ratio

Lame parameters λ and μ are material constants related to the fundamental physical parameters: Poisson's Ratio and Young's modulus (http://en.wikipedia.org/wiki/Lamé_parameters)



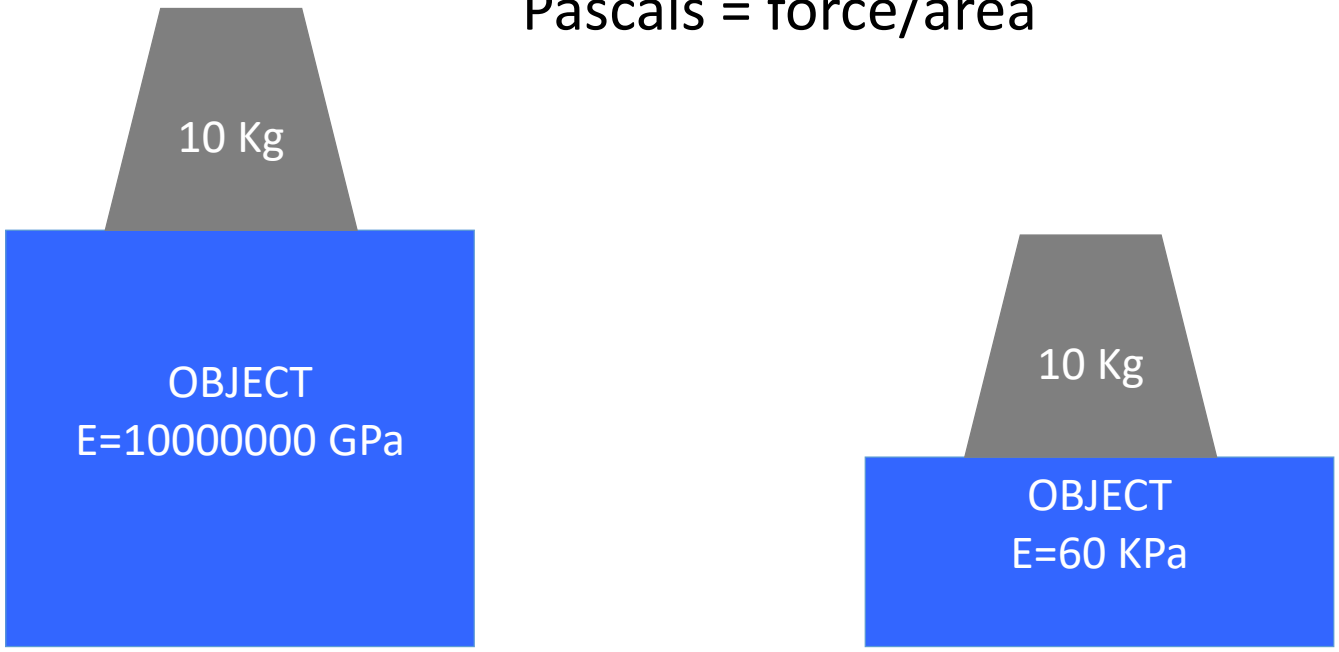
Young's modulus (E), measure of stiffness



Poisson's ratio (ν), captures transverse deformation relative to axial deformation

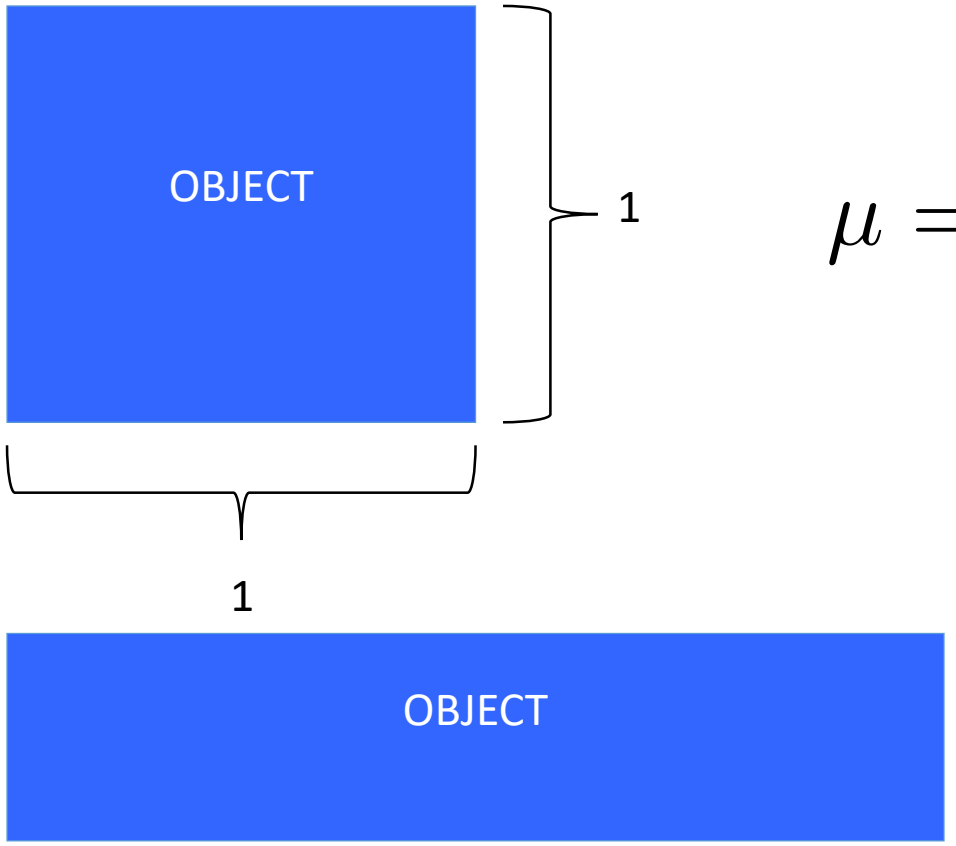
What Do These Parameters Mean

- Stiffness is pretty intuitive



What Do These Parameters Mean

Poisson's Ratio quantifies the extent to which volume is preservation

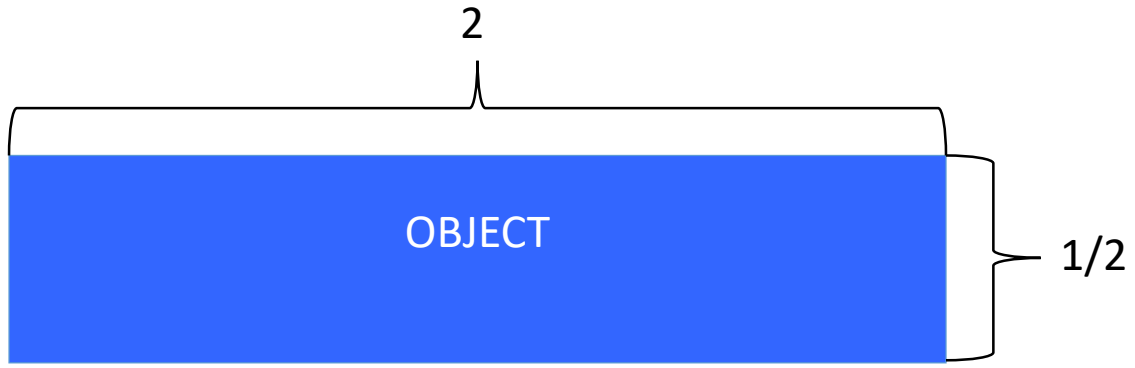


$$\mu = -\frac{\Delta y}{\Delta x}$$

What Do These Parameters Mean

Poisson's Ratio controls volume preservation

$$\mu = -\frac{\Delta y}{\Delta x}$$

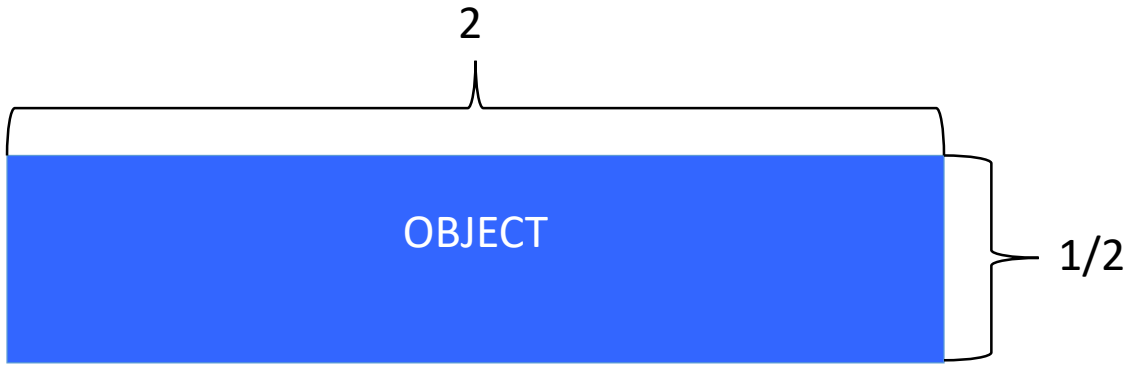


What Do These Parameters Mean

Poisson's Ratio controls volume preservation

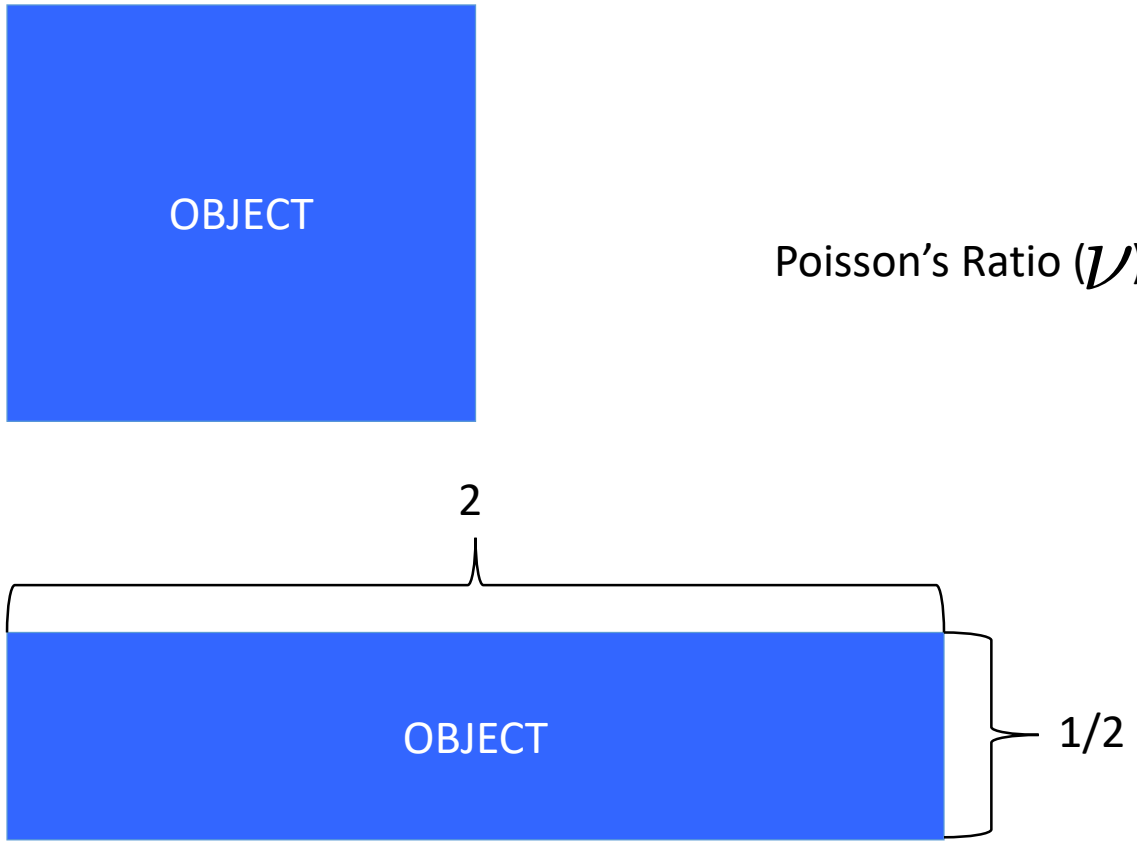
$$\mu = -\frac{\Delta y}{\Delta x}$$

Poisson's Ratio (ν) = ?



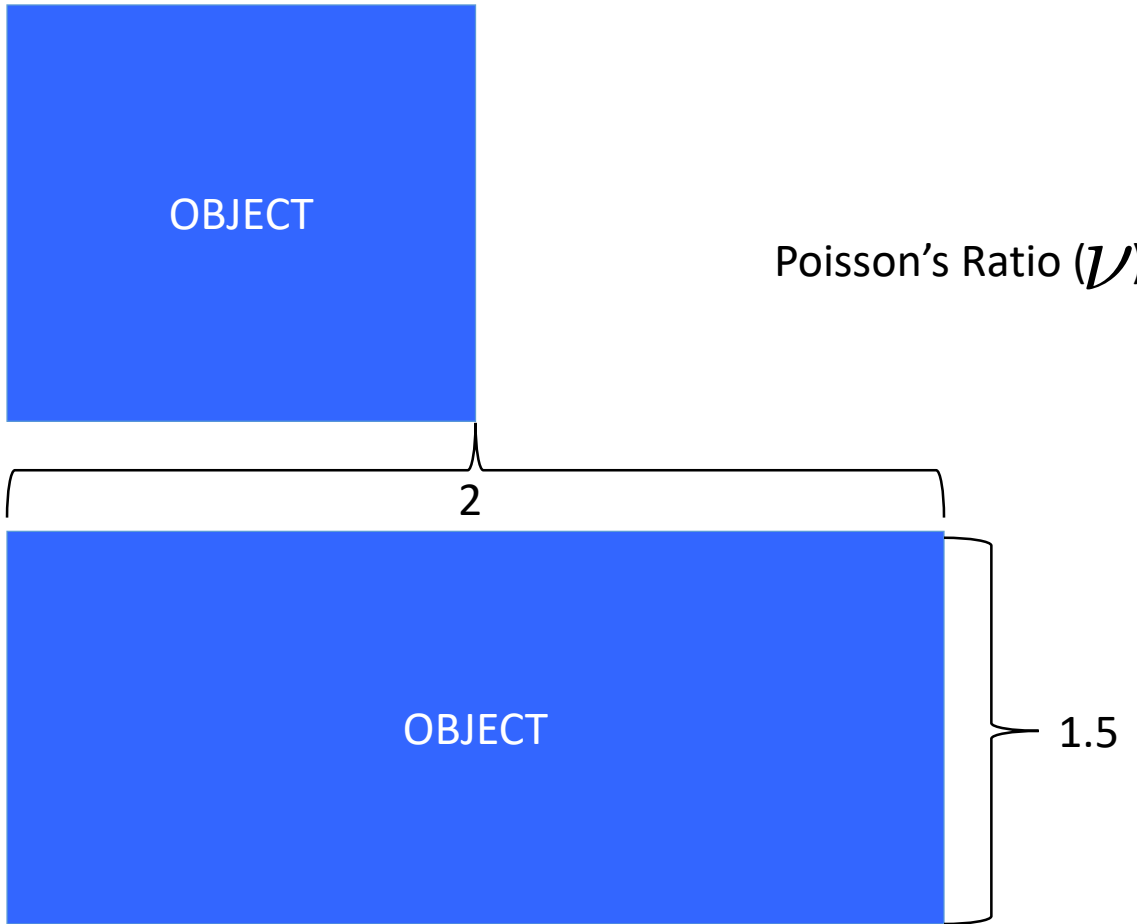
What Do These Parameters Mean

Poisson's Ratio controls volume preservation

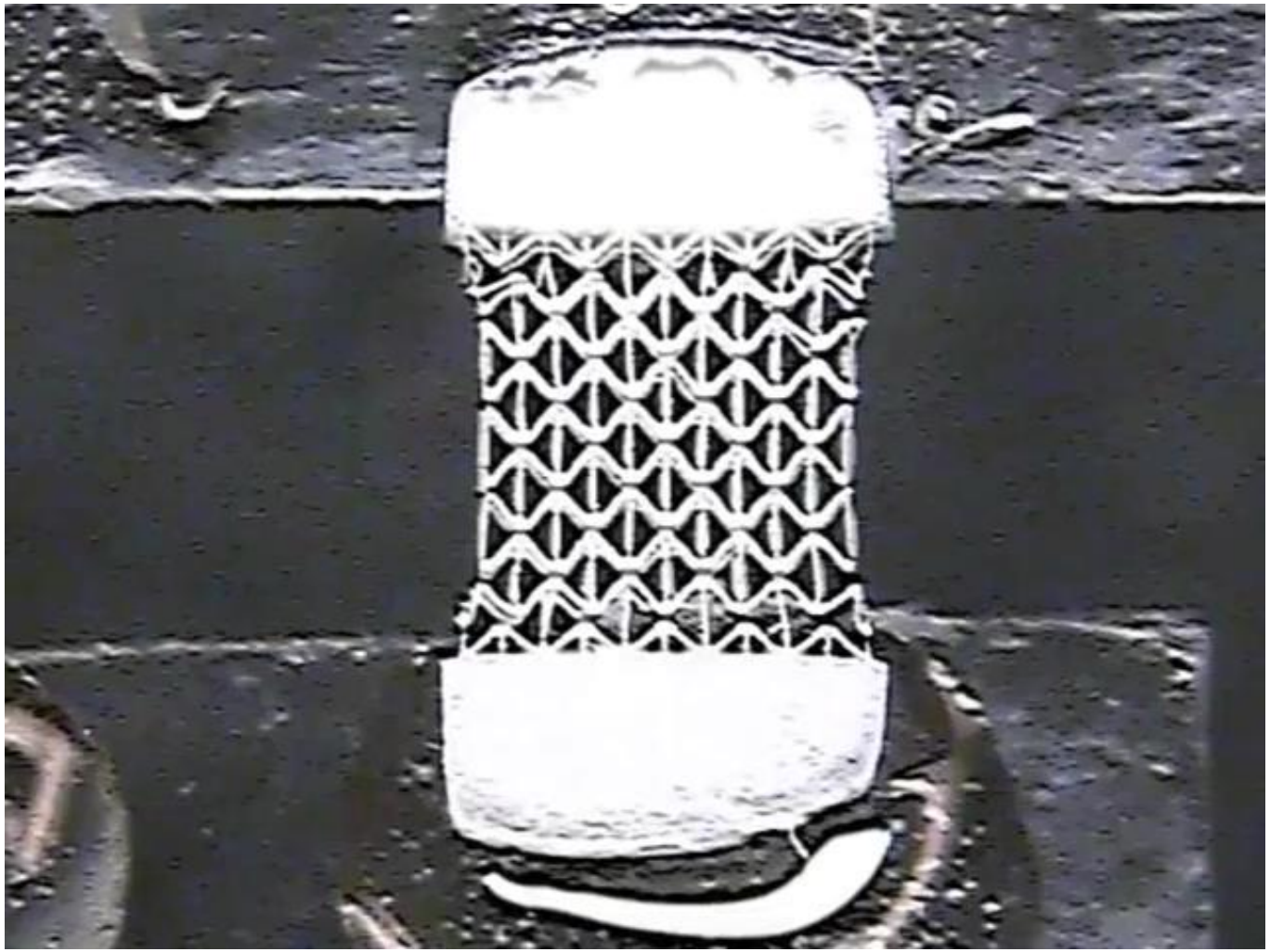


What Do These Parameters Mean

Poisson's ratio is between -1 and 0.5



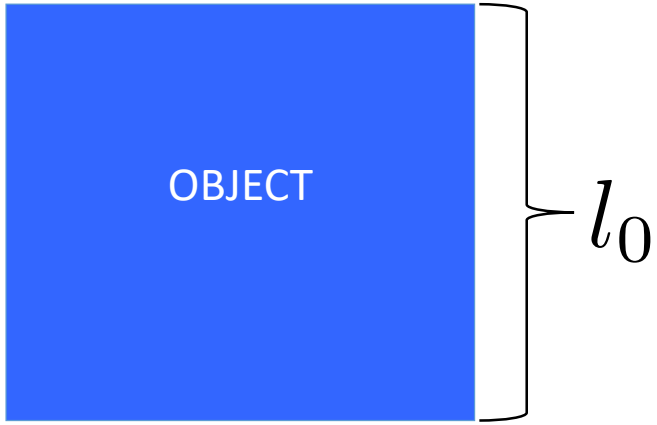
Negative Poisson's Ratio materials



Measurement

- Where do material parameters come from?

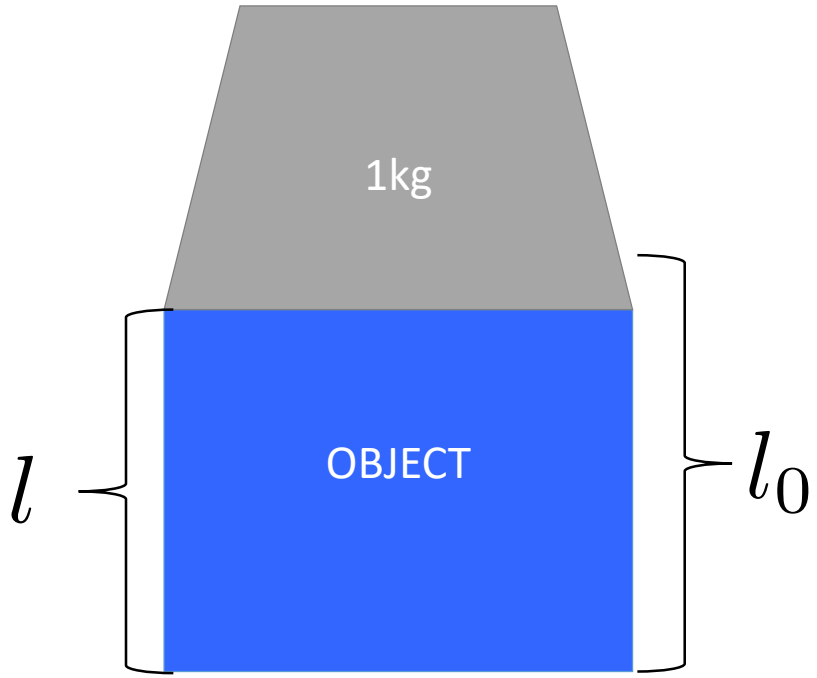
Simple Measurement: Stiffness



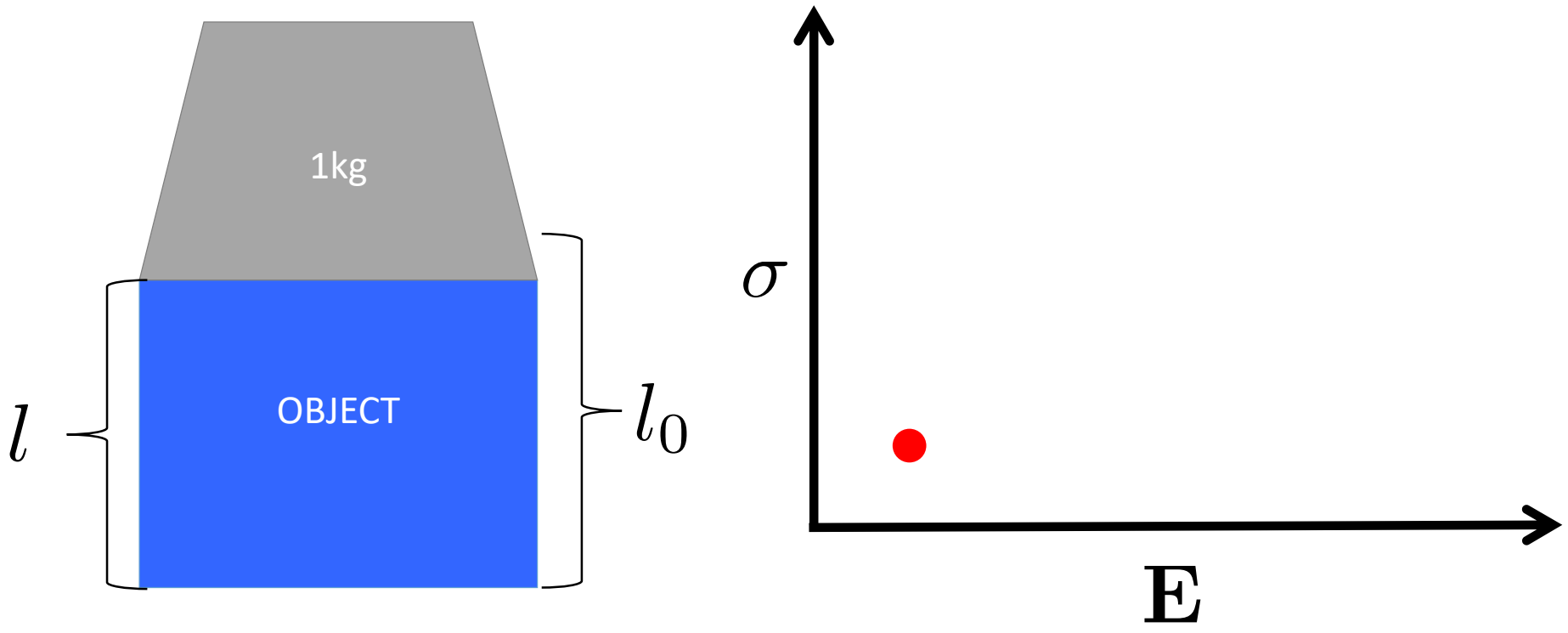
Simple Measurement

What's the Force (Stress)?

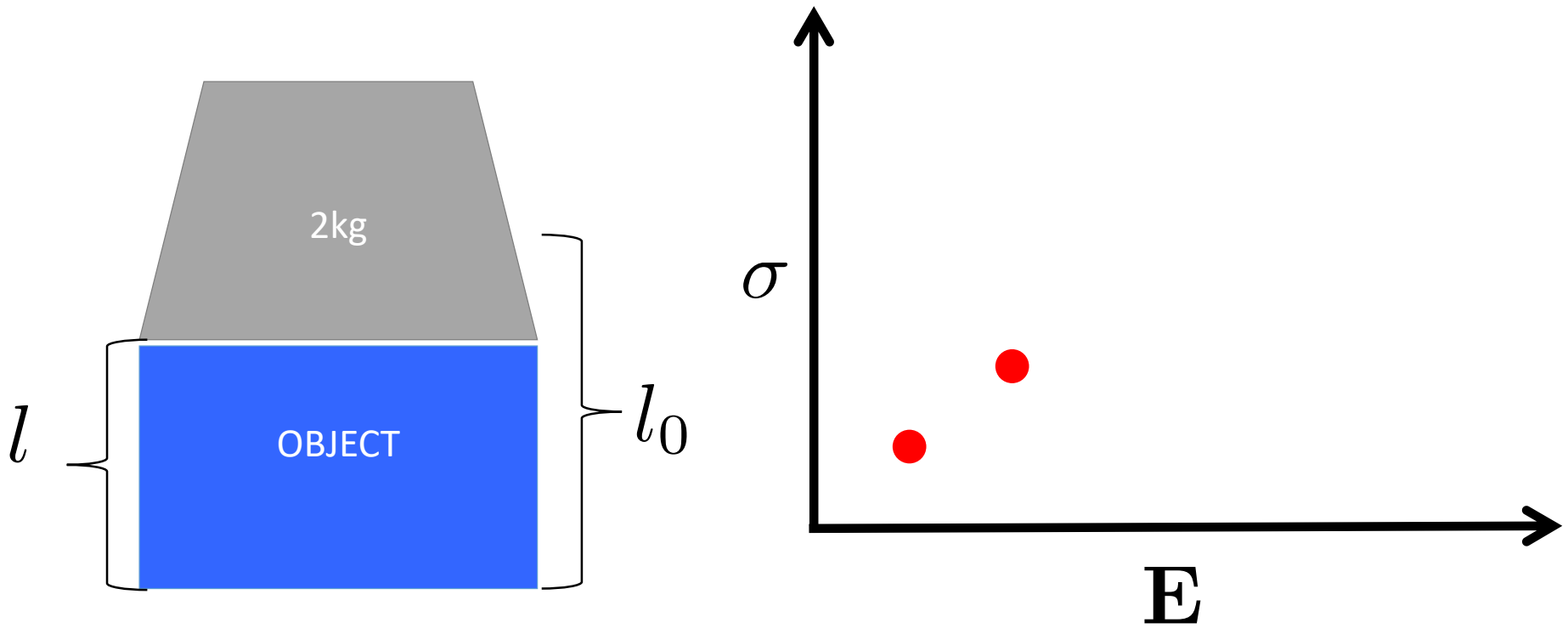
What's the Deformation (Strain)?



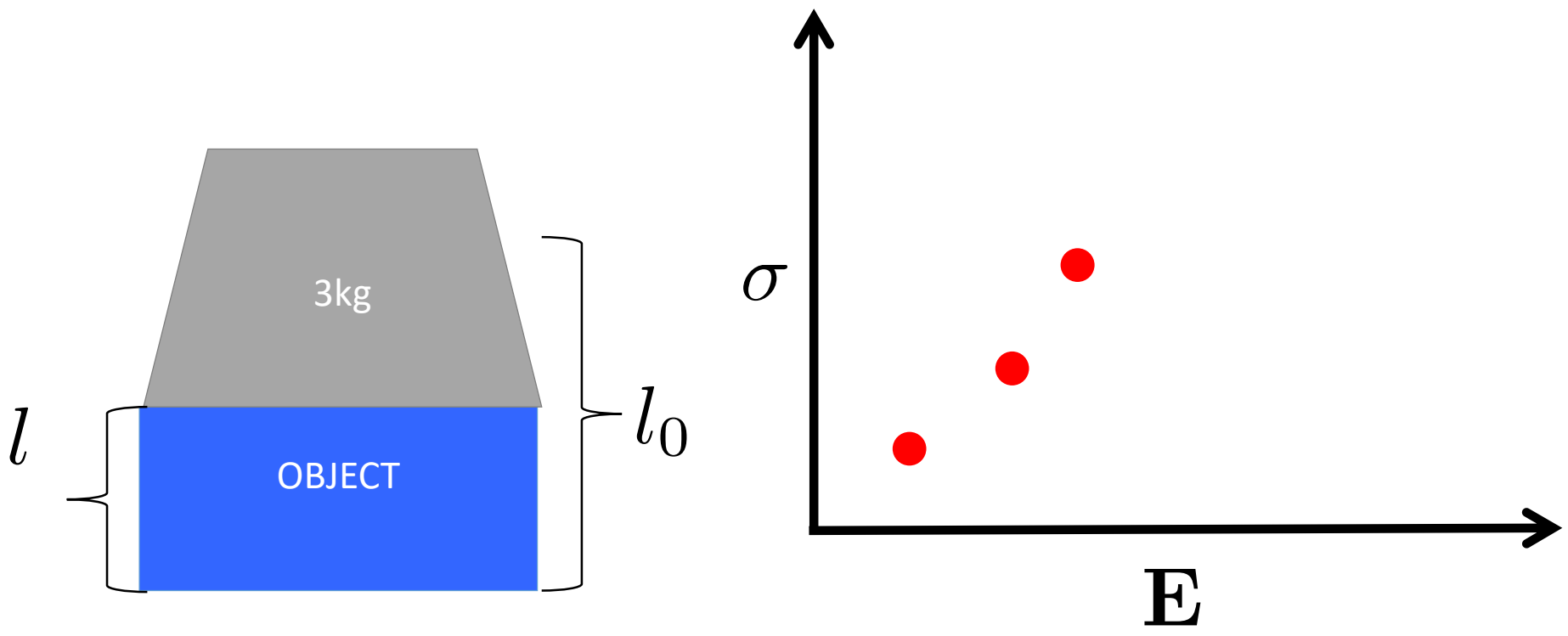
Simple Measurement



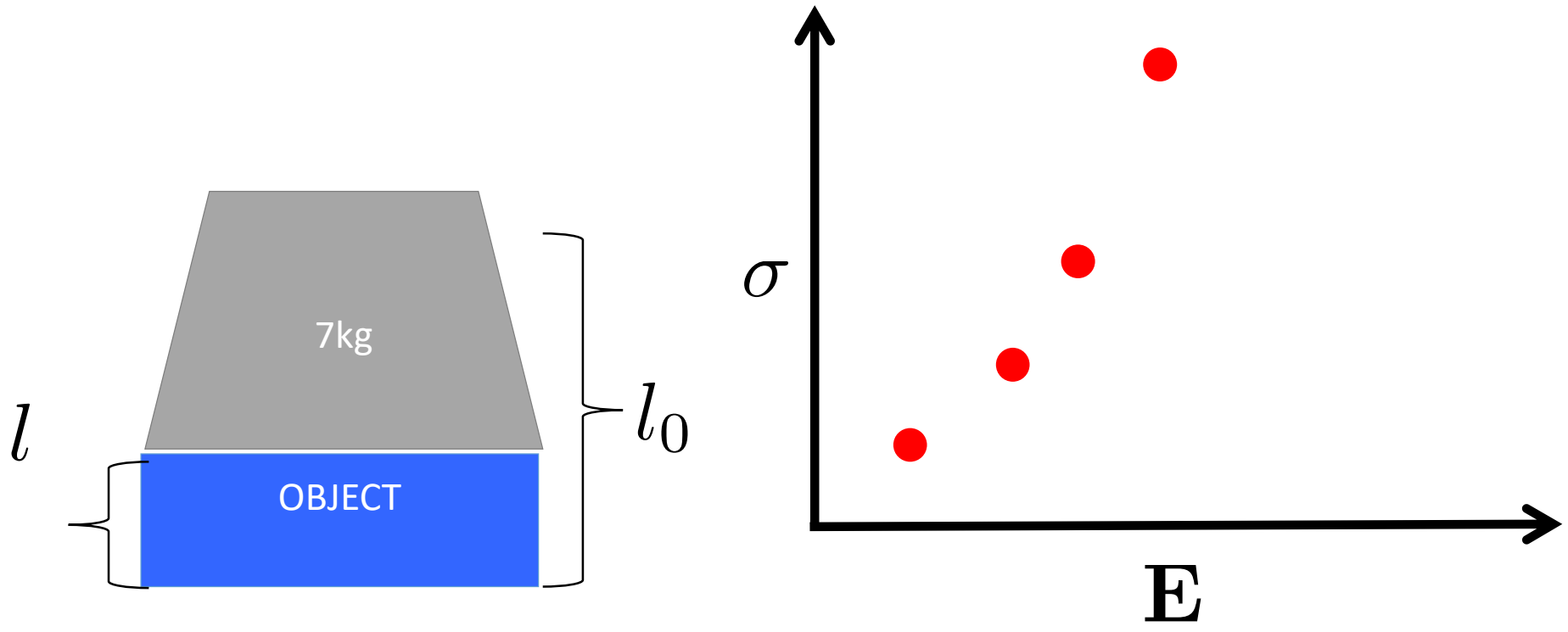
Simple Measurement



Simple Measurement

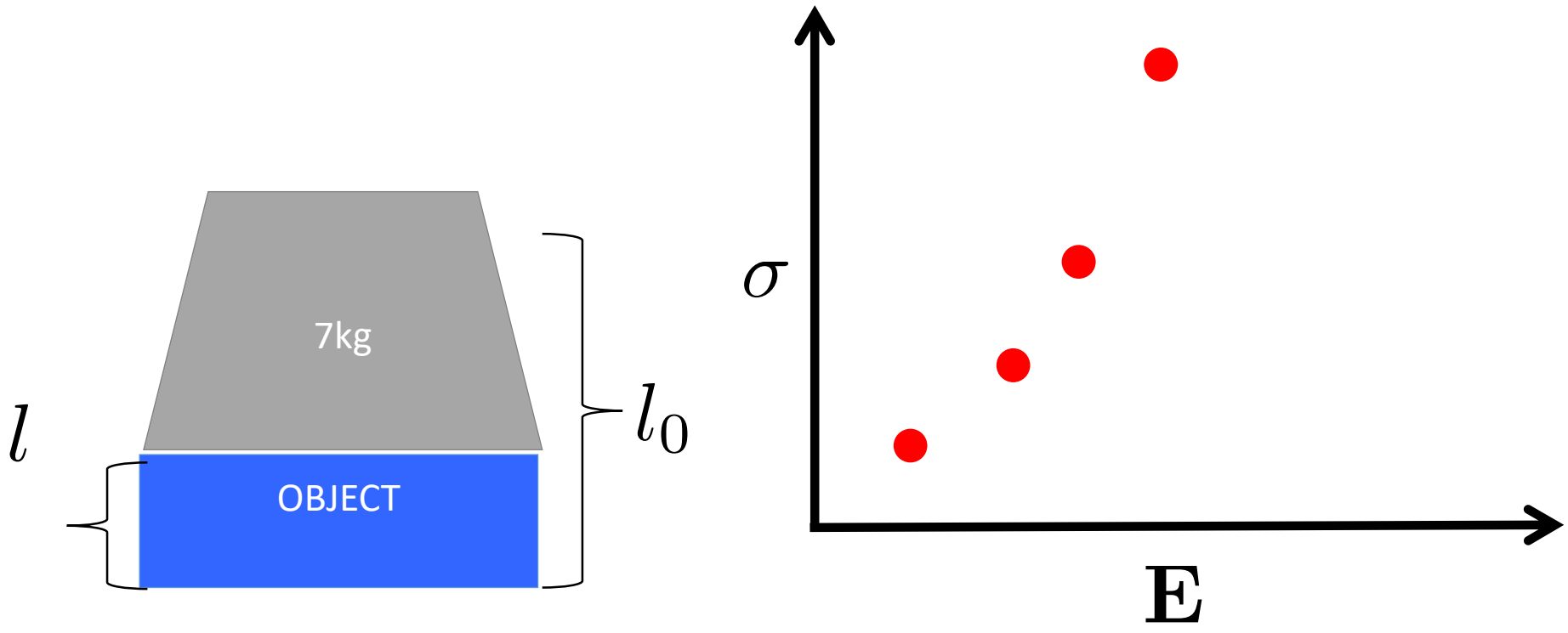


Simple Measurement



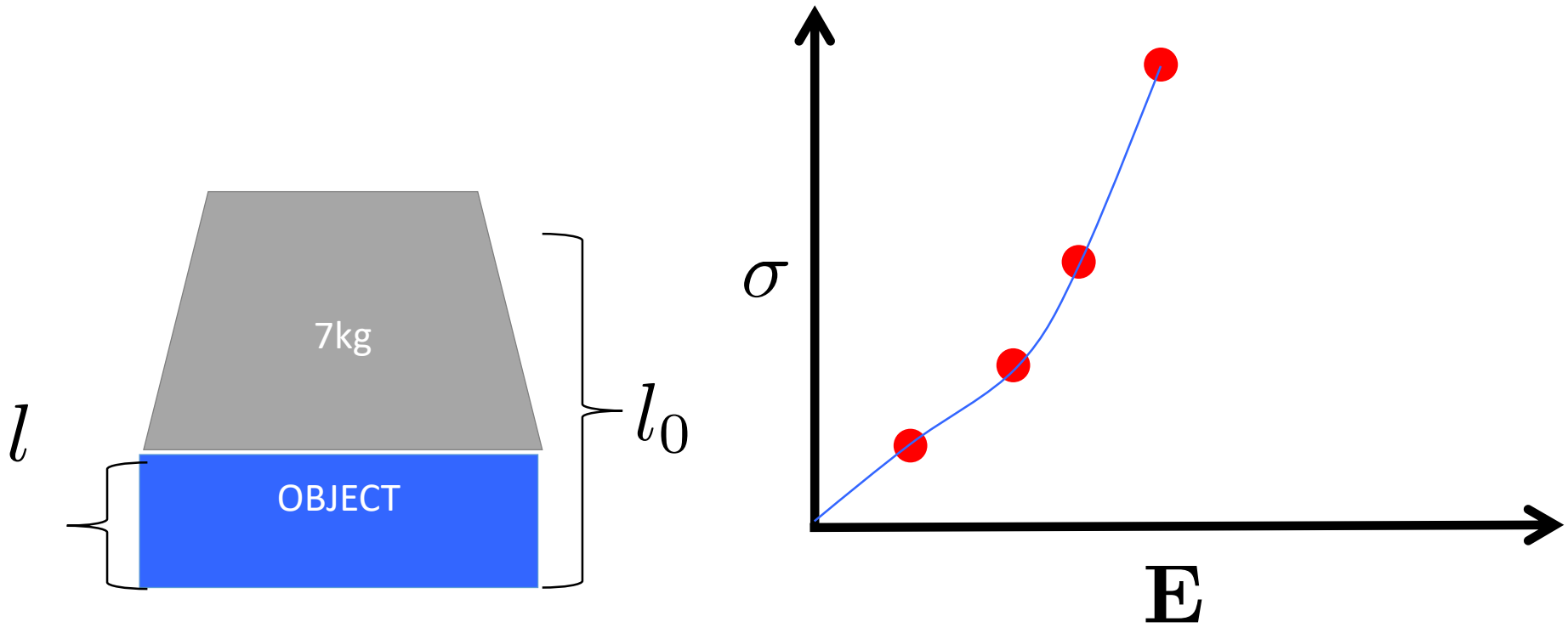
Simple Measurement

How do we get the stiffness ?



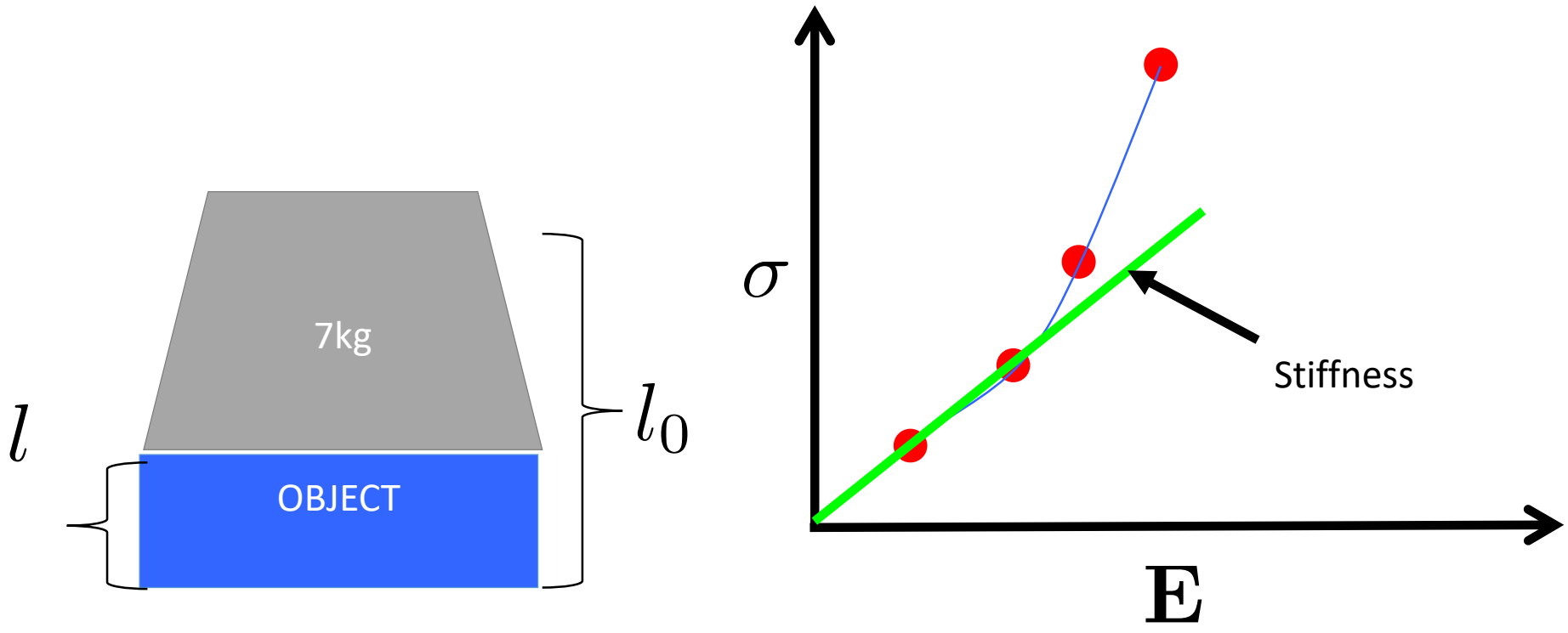
Simple Measurement

How do we get the stiffness ?

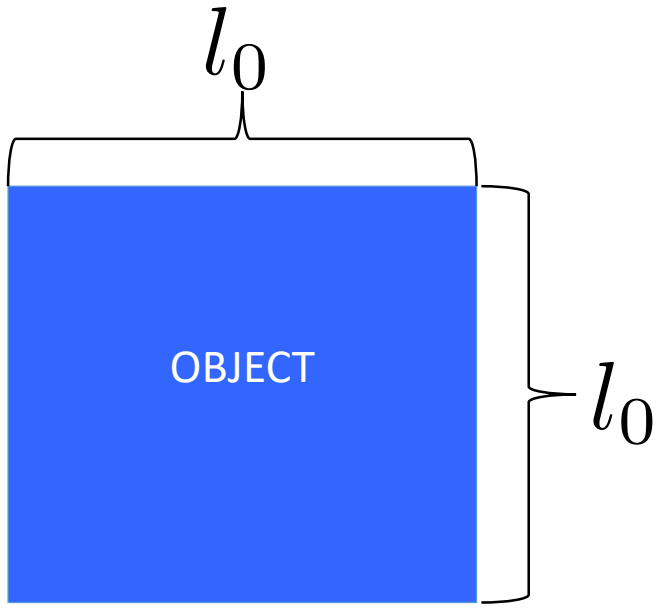


Simple Measurement

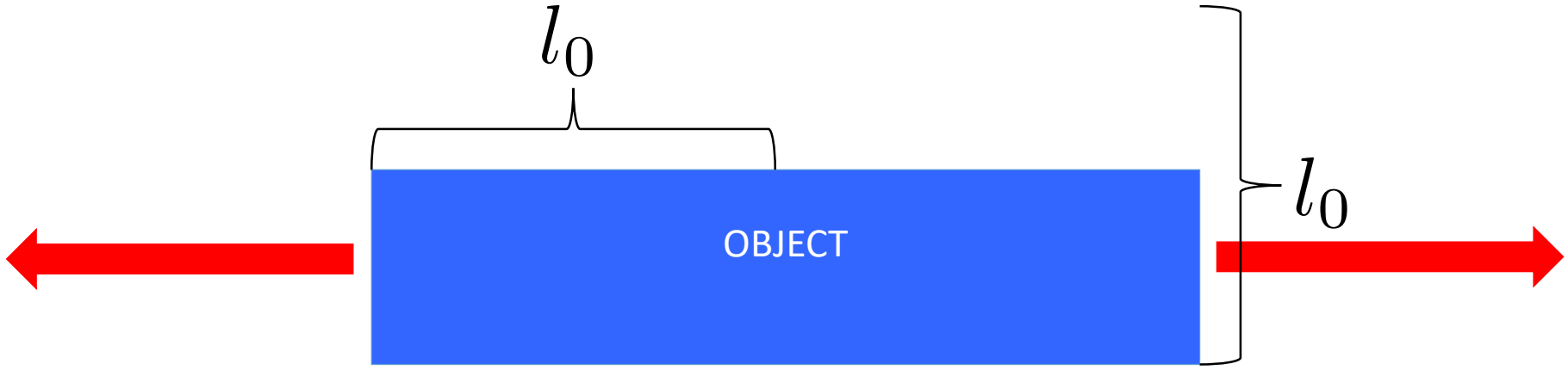
How do we get the stiffness ?



Simple Measurement: Poisson's Ratio

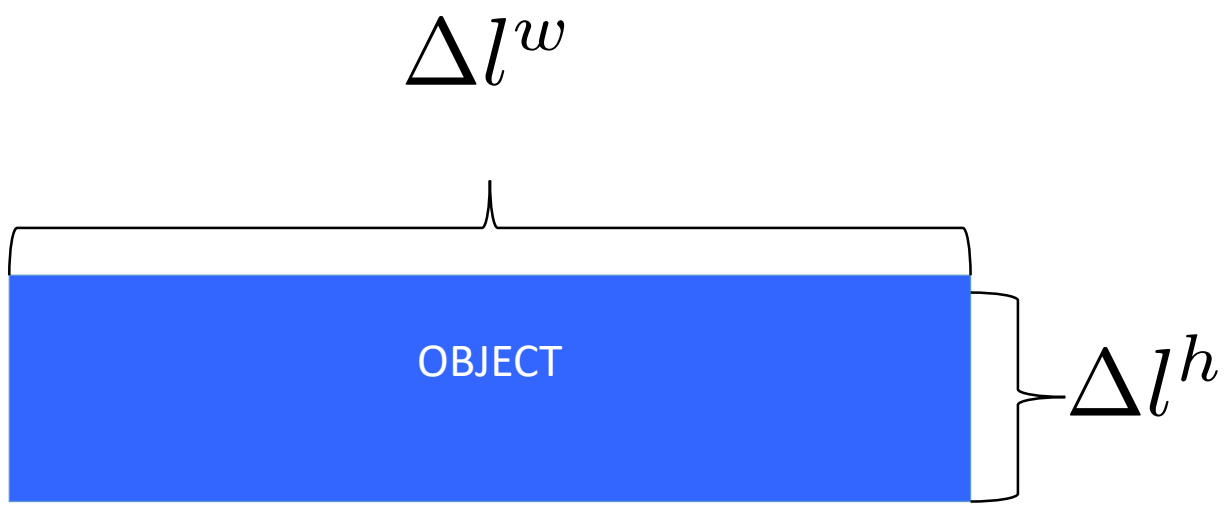


Simple Measurement: Poisson's Ratio



Simple Measurement: Poisson's Ratio

Compute changes in width and height



Simple Measurement: Poisson's Ratio

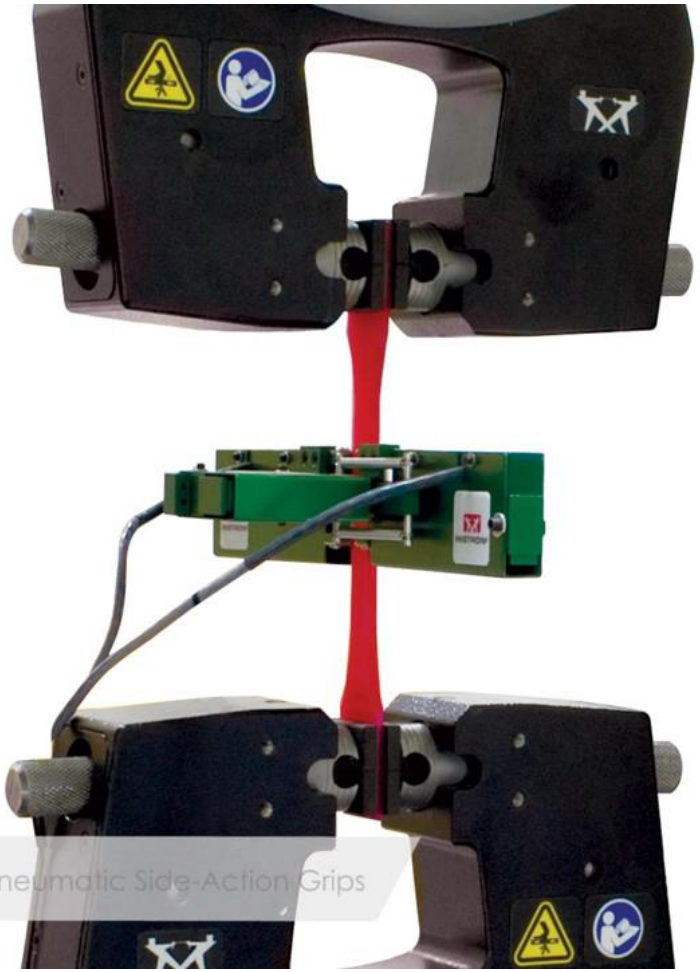
$$\nu = \frac{\Delta l_h}{\Delta l_w}$$

Poisson's Ratio

$$\frac{\Delta l^w}{l_0^w}$$

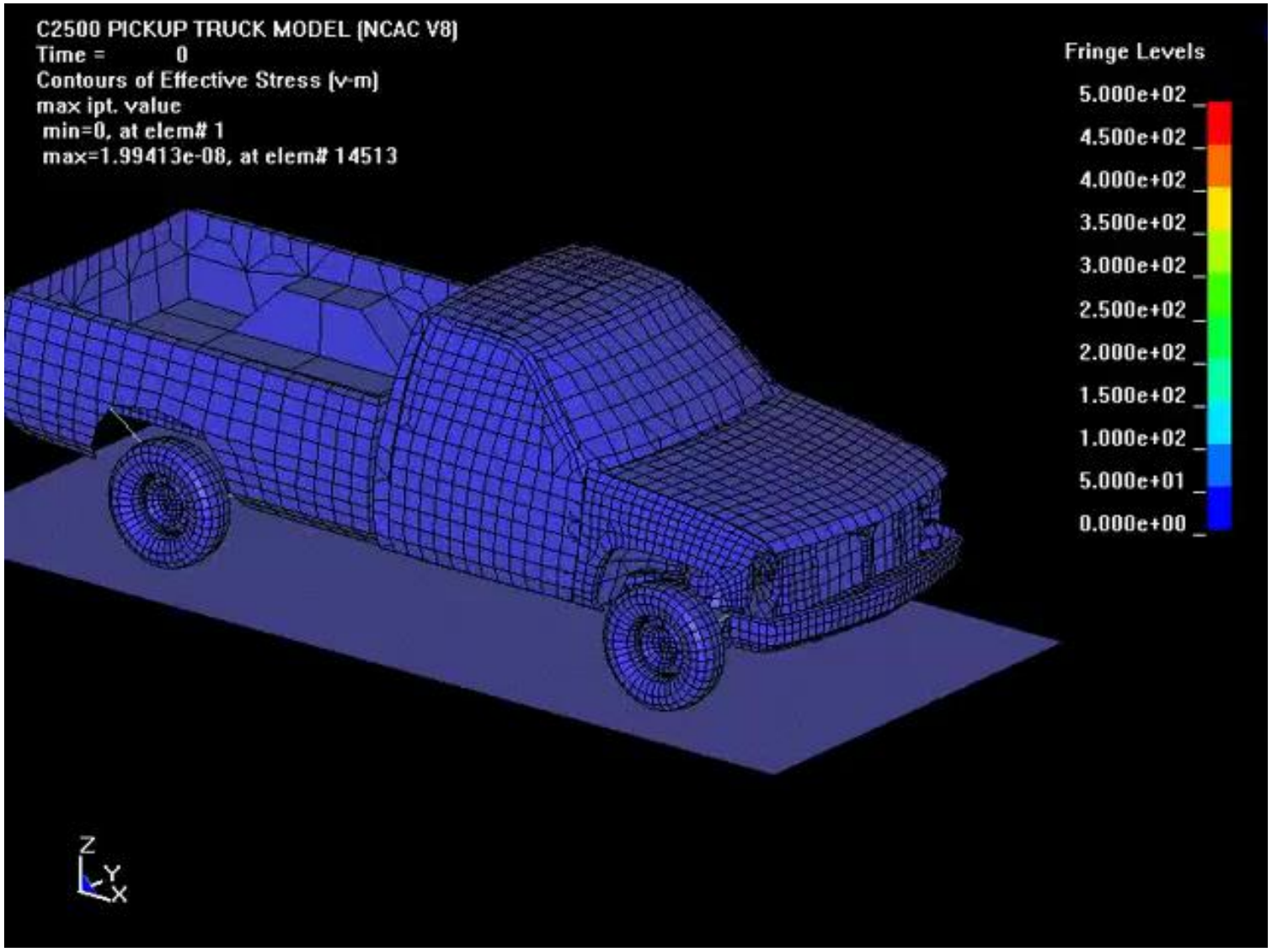
$$\frac{\Delta l^h}{l_0^h}$$

Measurement Devices



Bi-Axial Extensometer with 5 kN Pneumatic Side-Action Grips

Suppose you want to simulate this…



The Limits of Hyperelasticity

- Real-world materials are not perfectly hyperelastic
  - Viscosity (*stress relaxation, creep*)
  - Plasticity (*irreversible deformation*)
  - Mullins effect (*stiffness depends on strain history*)
  - Fatigue, damage, …

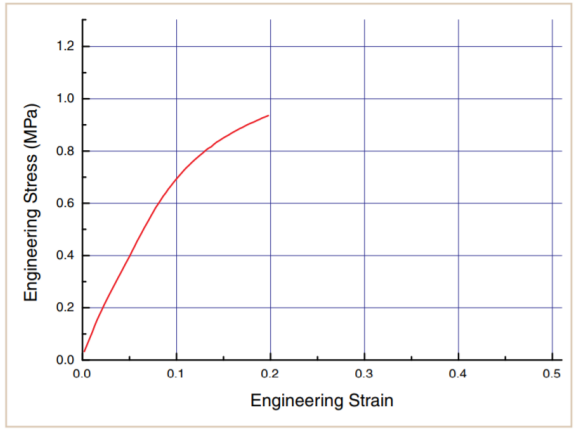


Figure 11, 1st Loading of a Thermoplastic Elastomer

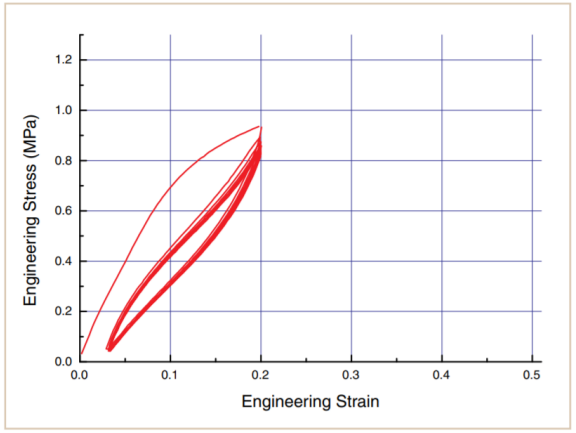


Figure 12, Multiple Strain Cycles of a Thermoplastic Elastomer

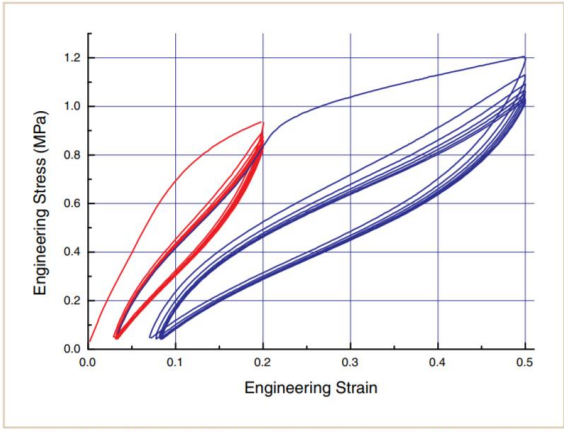


Figure 13, Multiple Strain Cycles of a Thermoplastic Elastomer at 2 Maximum Strain Levels

Further Reading

Textbooks

- Bonet and Wood, Nonlinear Continuum Mechanics
- Ogden, Nonlinear Elastic Deformations