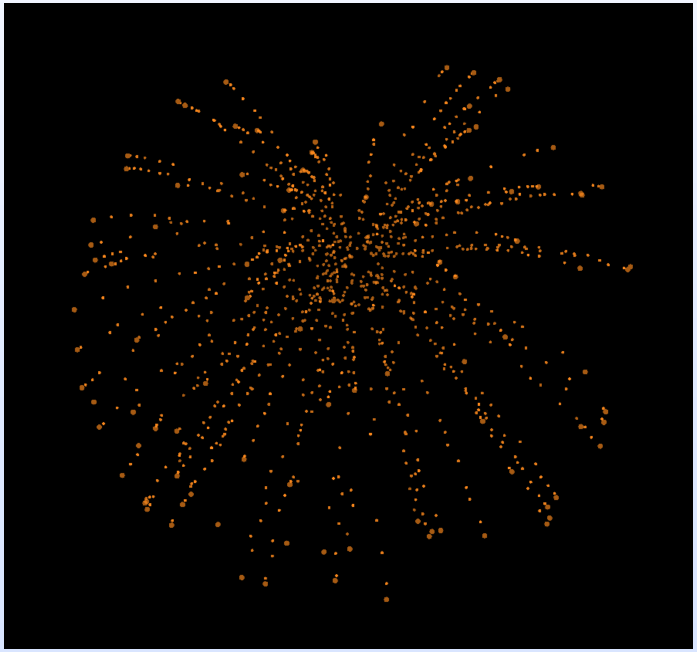
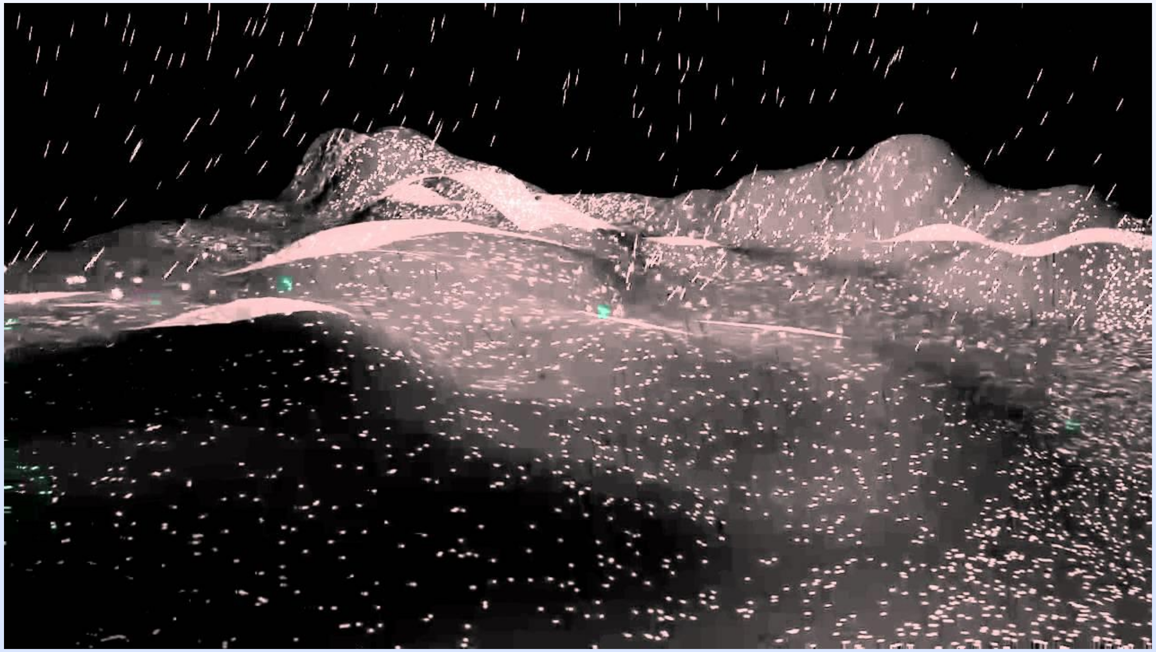
Mass-Spring Systems

What we've seen so far

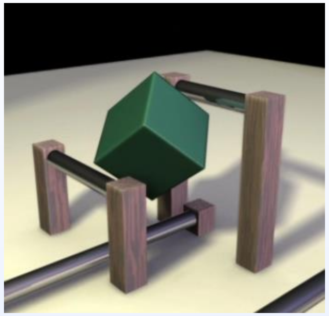
Individual particles moving under the influence of various types of forces



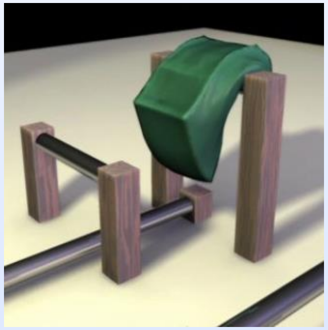


What if we wanted to simulate physical phenomena that are more complex?

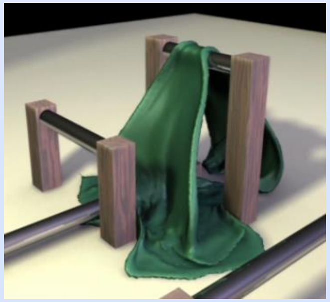
Material diversity



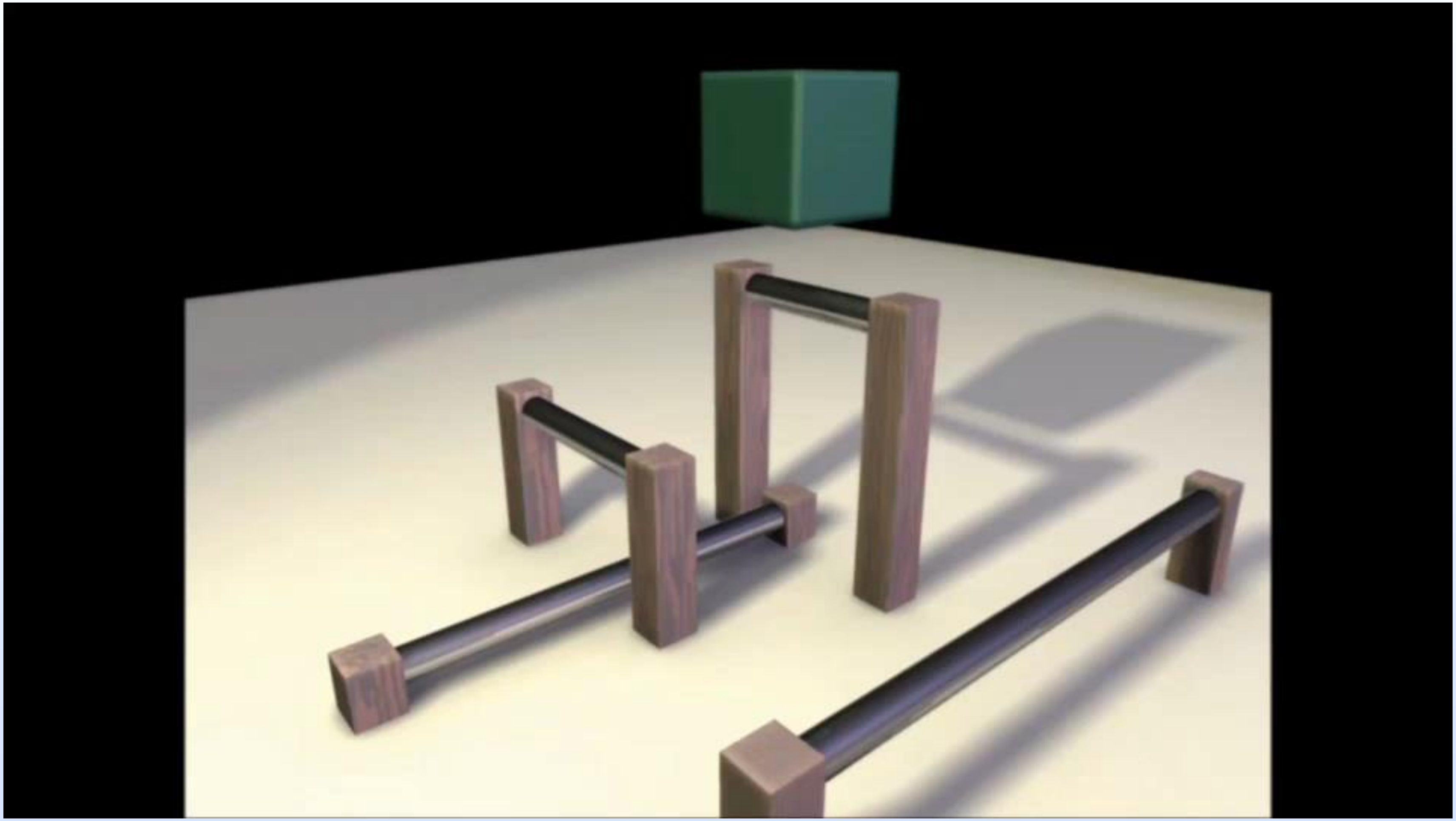
- Deformable objects
  - deform under applied forces
  - resist deformation



- Common material properties
  - **Elastic**: deformations are reversible
  - **Viscous**: amplitude of oscillations is reduced
  - **Plastic**: irreversible deformations
  - Different combinations, different extents



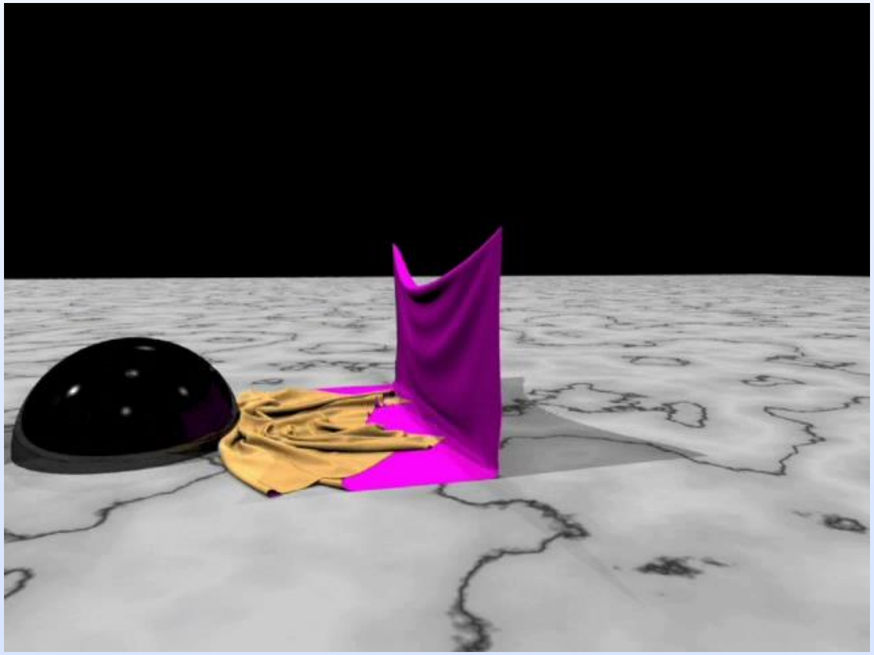
What if we wanted to simulate physical phenomena that are more complex?



What if we wanted to simulate physical phenomena that are more complex?

Model diversity

Cloth simulation



Bridson et al., 2002

Hair simulation

Selle et al., 2008

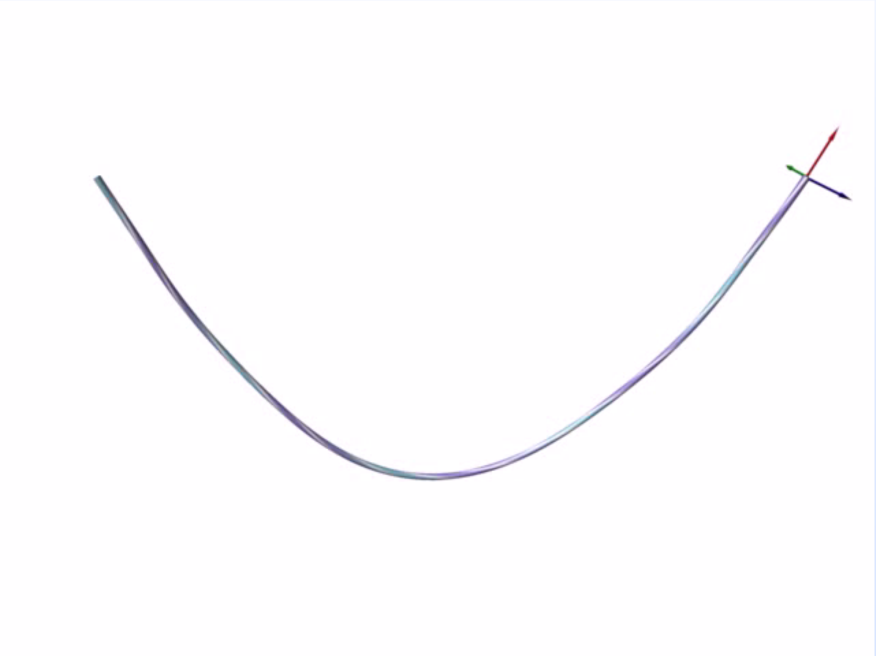
What if we wanted to simulate physical phenomena that are more complex?

Model diversity

Thin Shells

[Bridson et al. ‘03]

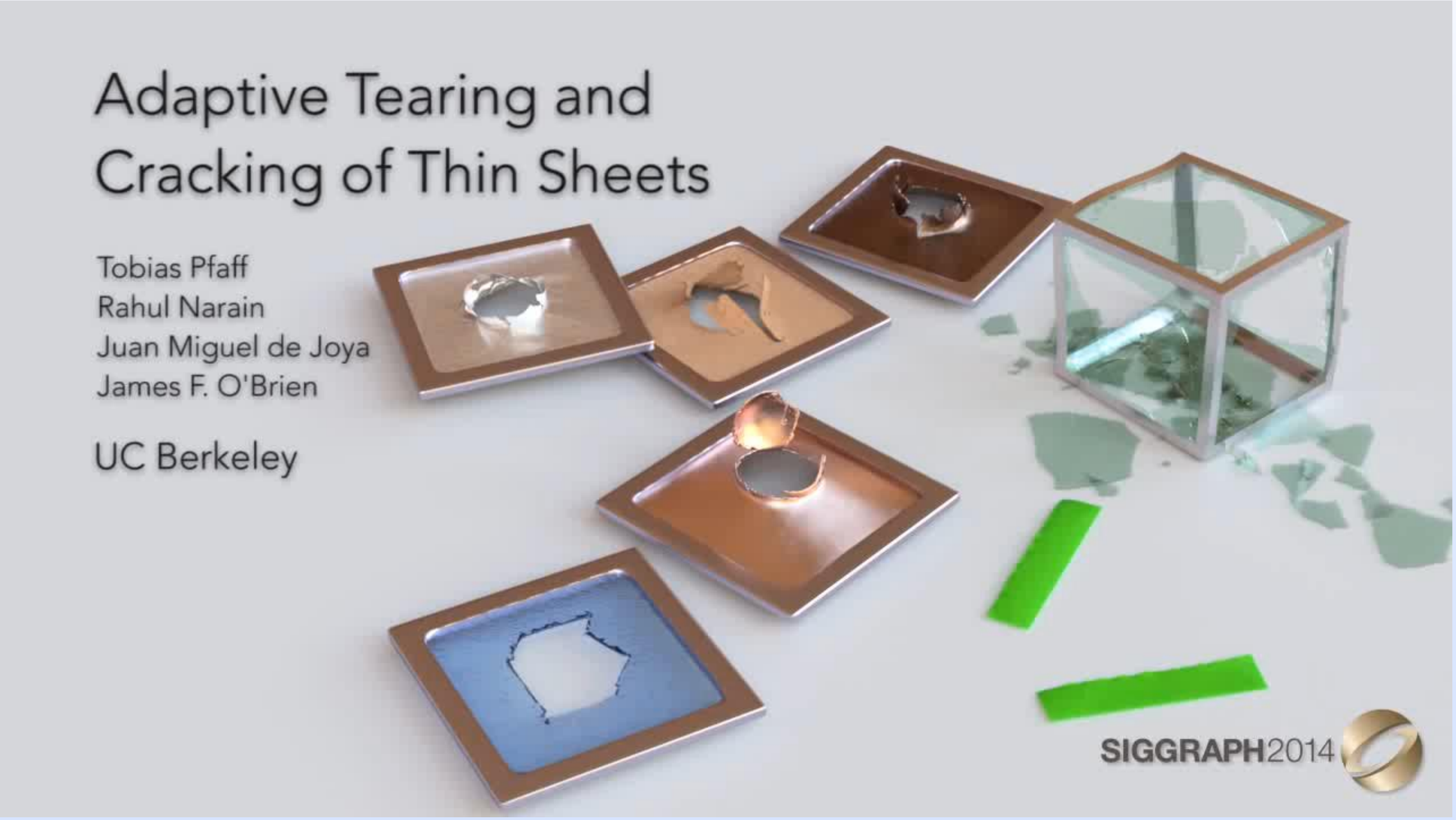
Rods



[Bergou et al. ‘08]

What if we wanted to simulate physical phenomena that are more complex?

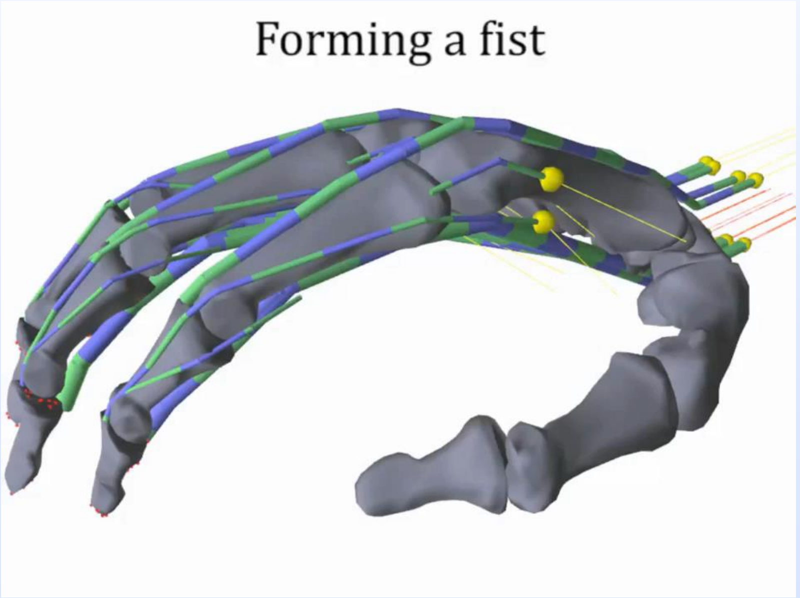
Diversity of phenomena



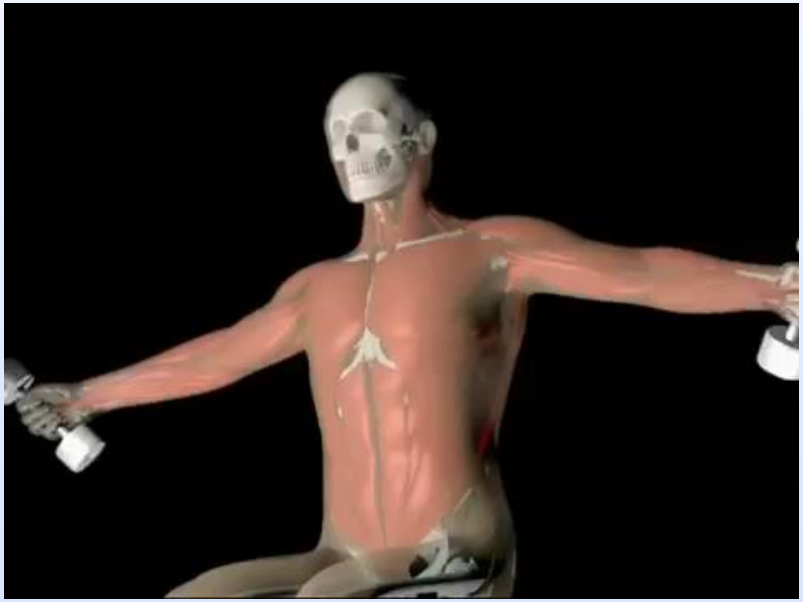
What if we wanted to simulate physical phenomena that are more complex?

Beyond passive objects

Muscles & biological tissues

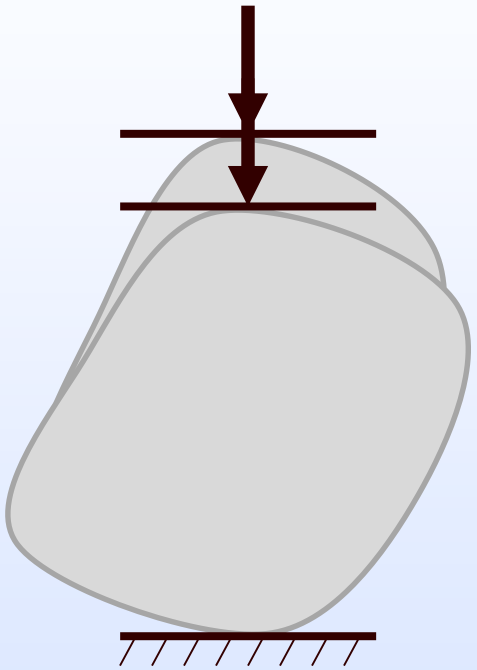


[Sachdeva et al. ‘15]

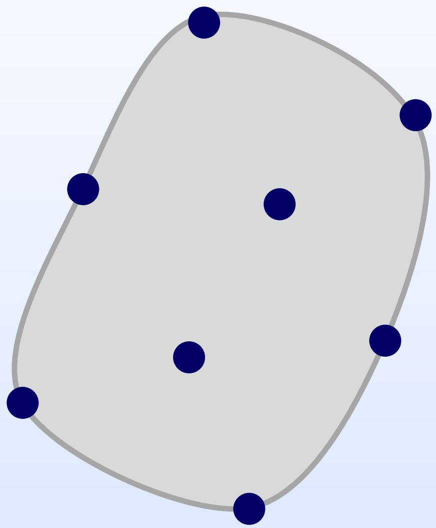


[Sifakis et al. ‘15]

Modeling complex phenomena



We can model many complex phenomena using mass-spring systems



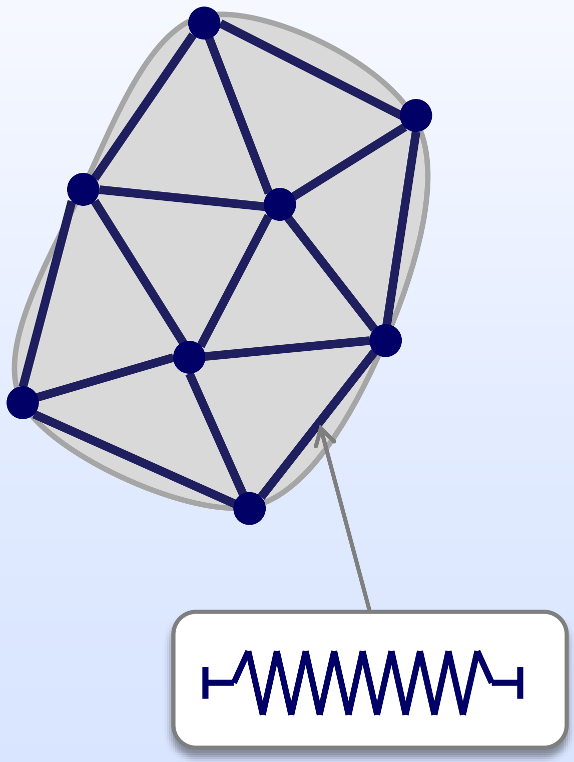
Spatial discretization: sample object with mass points

- Total mass of object: *M*
- Number of mass points: *p*
- Mass of each point: *m=M/p* (*uniform distribution*)

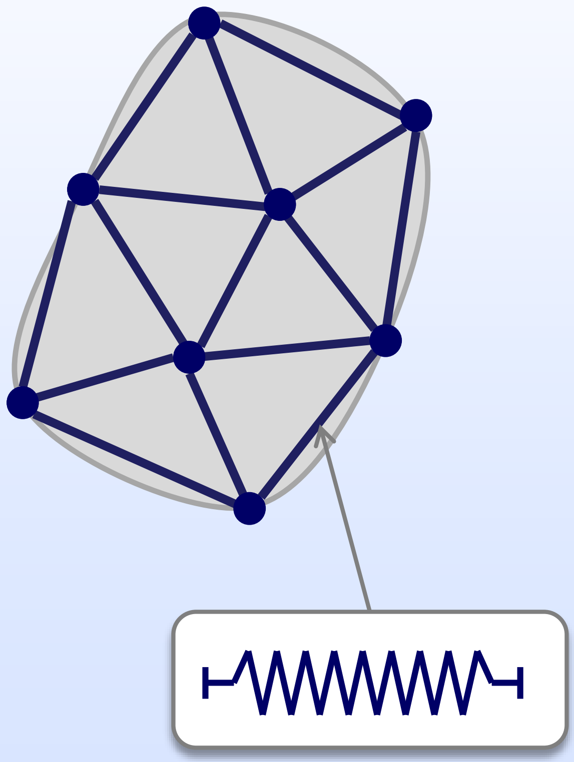
Each point is a particle, just like before. It has

- Mass m_i
- Position $\mathbf{x}_i(t)$
- Velocity $\mathbf{v}_i(t)$

We can model many complex phenomena using mass-spring systems



We can model many complex phenomena using mass-spring systems



External forces

– Gravity

– Contact & Collision Forces

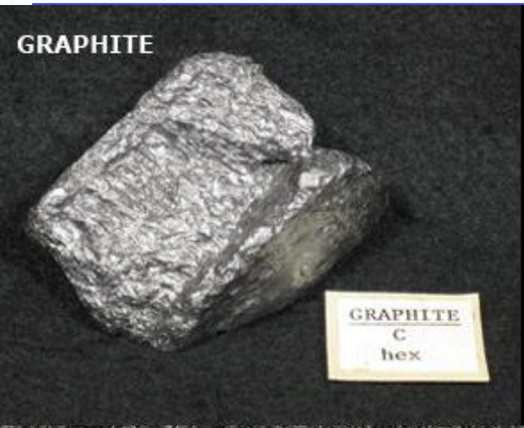
Internal forces

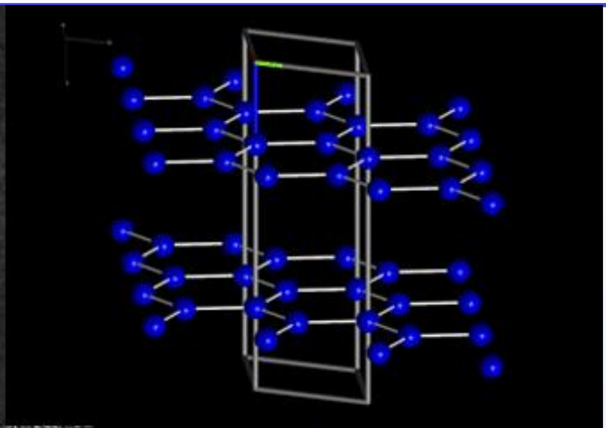
– Elastic spring forces

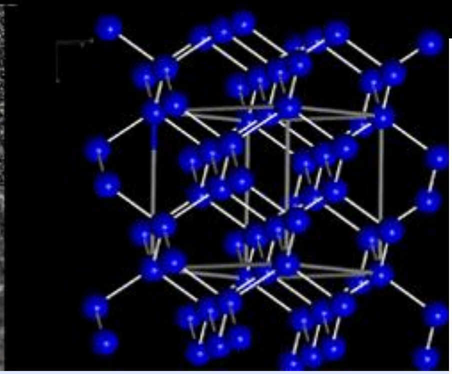
– Viscous damping forces

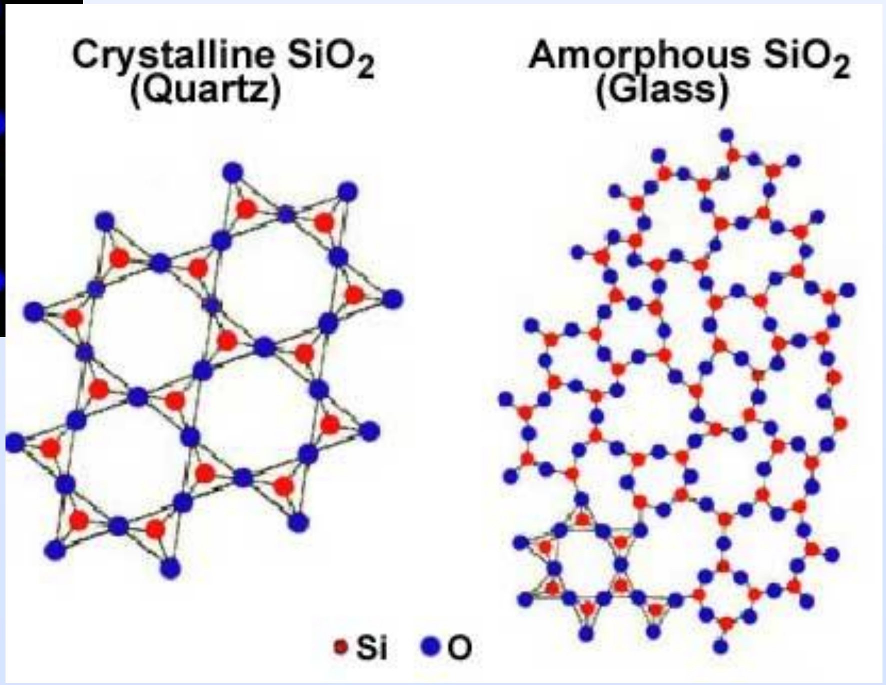
– Should always sum up to zero!

“Mass-spring” systems in the wild

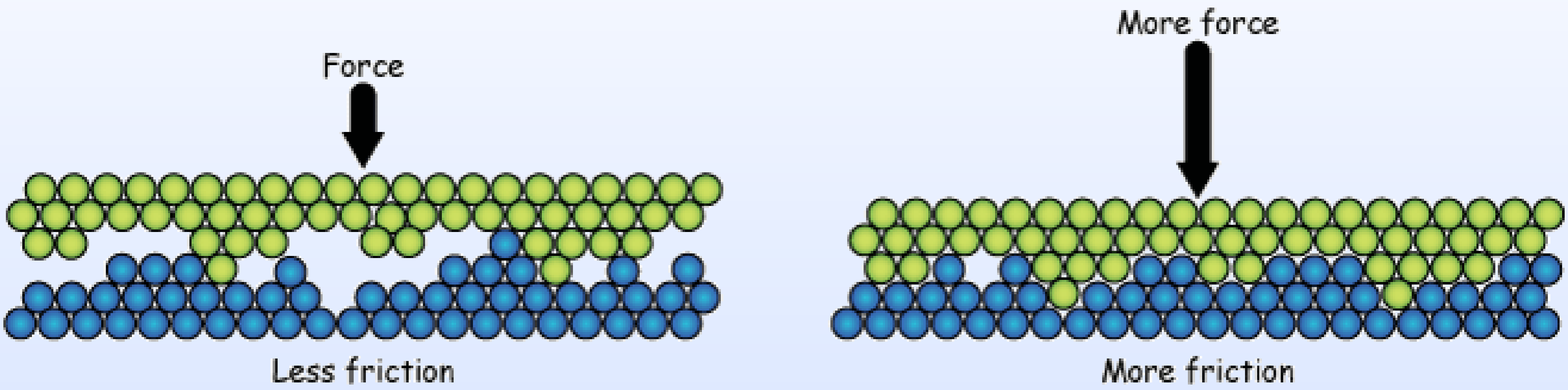




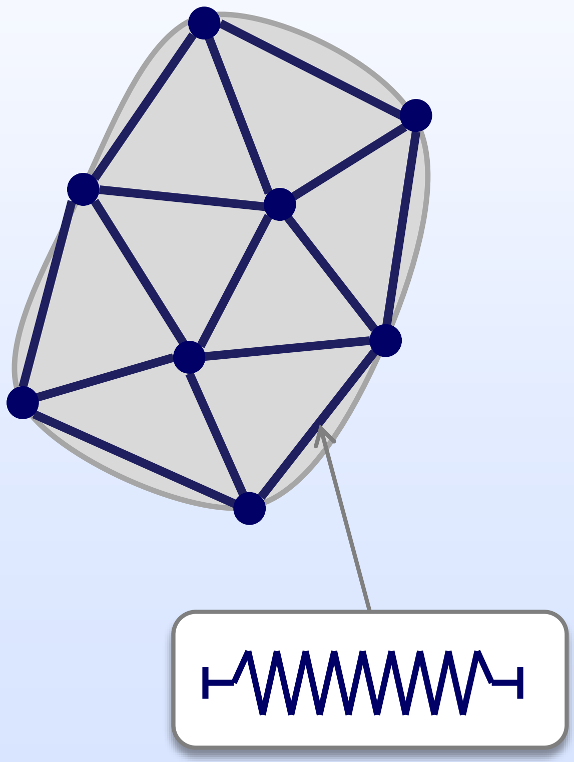




“Mass-spring” systems in the wild



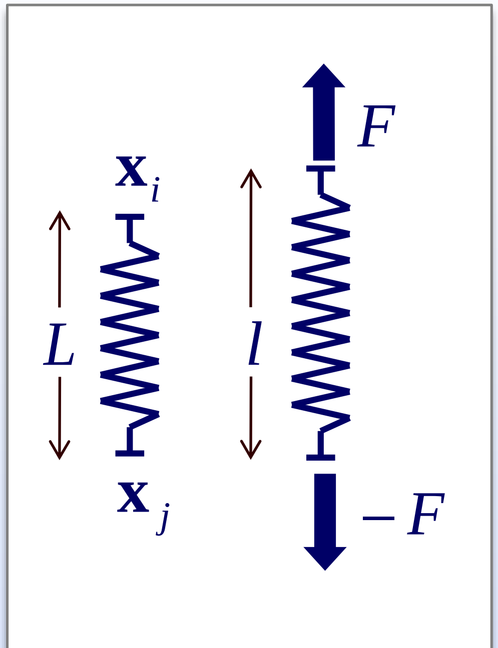
We can model many complex phenomena using mass-spring systems



External forces
- Gravity
- Contact & Collision Forces

Internal forces
- Elastic spring forces
- Viscous damping forces
- Should always sum up to zero!

We can model many complex phenomena using mass-spring systems



Zero length spring

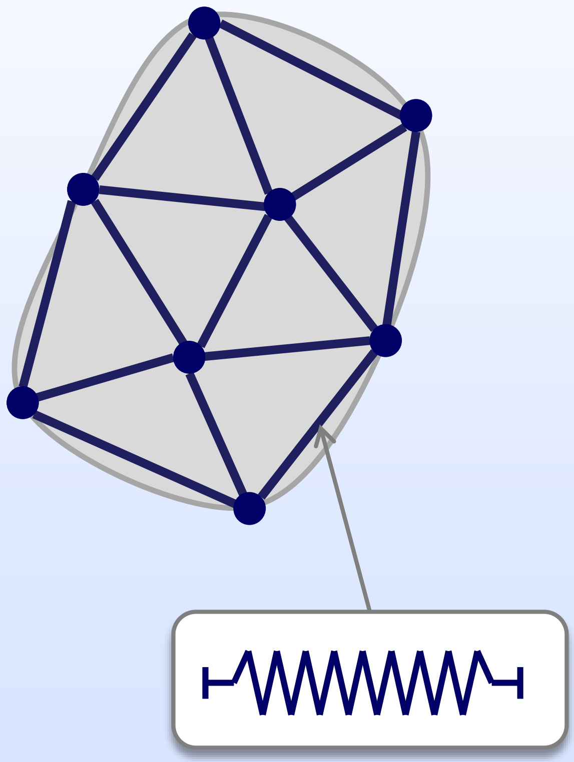
$$F_{spring} = -k(x_i - x_j)$$

Non-zero length spring

$$F_{spring} = -k\left(\frac{|x_i - x_j|}{L} - 1\right)\frac{x_i - x_j}{|x_i - x_j|}$$

Initial spring length L
Spring stiffness k

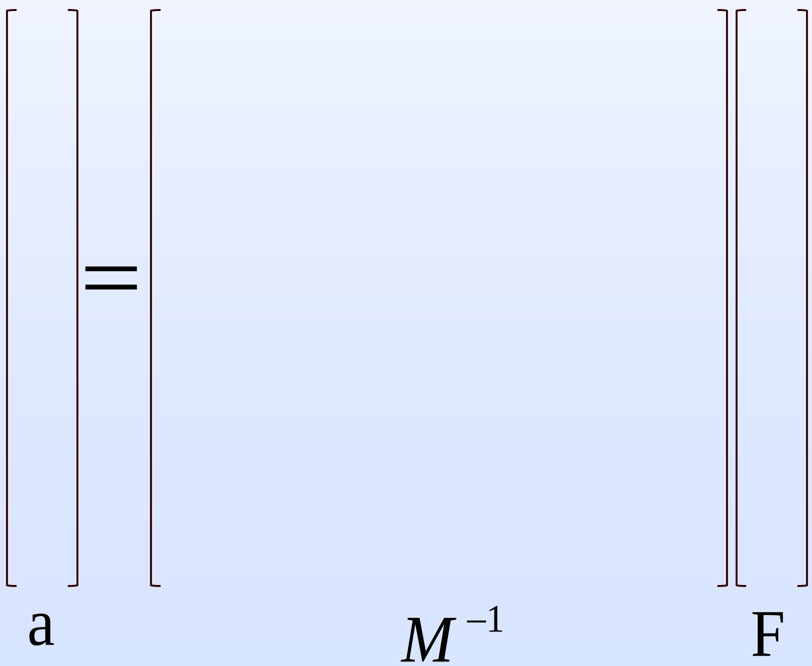
We can model many complex phenomena using mass-spring systems

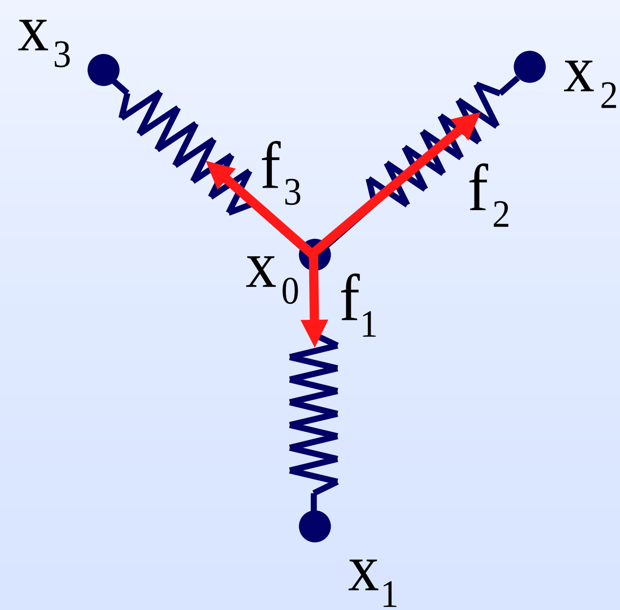


Modularity is key here. If you know how to model one spring, you know how to model complex objects!

Back to Equations of Motion

$$a = M^{-1}F(x, v)$$





Back to Equations of Motion

$$a = M^{-1} F(x, v)$$

$$a^i = \frac{1}{m^i} \sum f^i(x^i, v^i) = \frac{1}{m^i} F^i(x^i, v^i)$$

Symplectic Euler

1) Compute & sum up forces: $F(x_n, v_n)$

2) Update velocities: $v_{n+1} = v_n + hM^{-1}F(x_n, v_n)$

3) Update positions: $x_{n+1} = x_n + hv_{n+1}$

Backward Euler

- Recall need to solve:

$$x_{n+1} = x_n + h v_{n+1}$$

$$v_{n+1} = v_n + h M^{-1} F(x_{n+1}, v_{n+1})$$

- which boils down to solving systems of linear equations:

$$\underbrace{\left(M - h\frac{\partial F}{\partial v} - h^2 \frac{\partial F}{\partial x} \right)}_{\text{A}} \underbrace{\Delta v}_{x} = \underbrace{M\left(v_n - v^k\right) + hF}_{\text{b}}$$

Forces & Force Jacobians

- We've seen what F is and how it is computed. What are $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial v}$?
- A bit of tensor calculus overview:

$\frac{\partial \alpha}{\partial \beta}$: derivative of every component of α with respect to every component of β, where α and β can be scalars, vectors or matrices

Forces & Force Jacobians

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

- What is $\frac{\partial \mathbf{A}}{\partial \mathbf{x}}$?

A vector of matrices… we will see it soon enough…

Forces & Force Jacobians

- So, what does $\frac{\partial F}{\partial x}$ represent?
- Much more convenient to think in terms of blocks of the Jacobian:

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

- "how does the net force on particle i change when the position of particle j changes"
  - Need to look at derivatives of individual spring forces

Forces & Force Jacobians

- But first, some useful derivatives

$$\frac{dA}{\partial x} = 0$$

$$\frac{dx}{\partial x} = I$$

$$\frac{dAx}{\partial x} = A$$

$$\frac{dx^T A}{\partial x} = A^T$$

$$\frac{dx^T Ax}{\partial x} = x^T (A + A^T)$$

$$\frac{dx^T x}{\partial x} = 2x^T$$

$$\frac{da^T x}{\partial x} = \frac{dx^T a}{\partial x} = a^T$$

$$u = x_i - x_j$$

$$\frac{du}{\partial x_i} = -\frac{du}{\partial x_j} = I$$

$$\frac{d\,|u|}{\partial x_i} = \frac{u}{|u|}$$

$$\frac{du \times v}{\partial x_i} = -v_\times$$

$v_\times$ is a skew-symmetric cross product matrix

Forces & Force Jacobians

- Zero length spring

$$F = -k(x_i - x_j) \qquad \frac{\partial F}{\partial x_i} = -\frac{\partial F}{\partial x_j} = -kI$$

- Non-zero Length Spring

$$F = -k\underbrace{\left(\frac{|x_i - x_j|}{L} - 1\right)}_{\varepsilon}\underbrace{\frac{x_i - x_j}{|x_i - x_j|}}_{u} = -k\varepsilon\frac{u}{|u|}$$

$$\frac{\partial F}{\partial x_i} = -\frac{\partial F}{\partial x_j} = -k\left(\frac{1}{L}\frac{uu^T}{u^T u} + \frac{\varepsilon}{|u|}\left(I - \frac{uu^T}{u^T u}\right)\right)$$

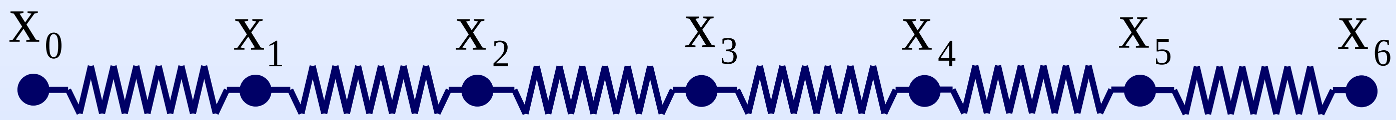
Forces & Force Jacobians

- 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
```

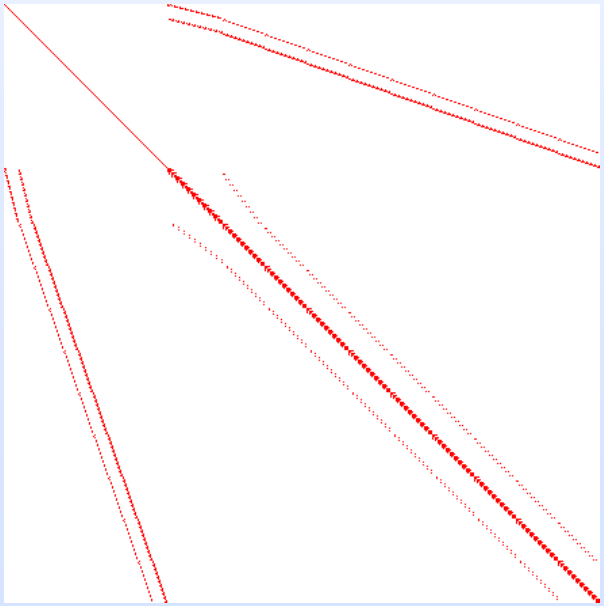
Forces & Force Jacobians

- We now know how to compute individual entries of the Jacobian, but what does it look like globally?



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!



Backward Euler

- Boils down to solving systems of linear equations:

$$\underbrace{\left(M - h\frac{\partial F}{\partial v} - h^2\frac{\partial F}{\partial x}\right)}_{\text{A}}\underbrace{\Delta v}_{x} \underset{=}{=} \underbrace{M\left(v_n - v^k\right) + hF}_{\text{b}}$$

- Matrix A is large, sparse, symmetric, (sometimes positive definite)
  - these characteristics will inform the choice of algorithm we can/should use to solve the systems of equations