COL865: Special Topics in Computer Applications

Physics-Based Animation

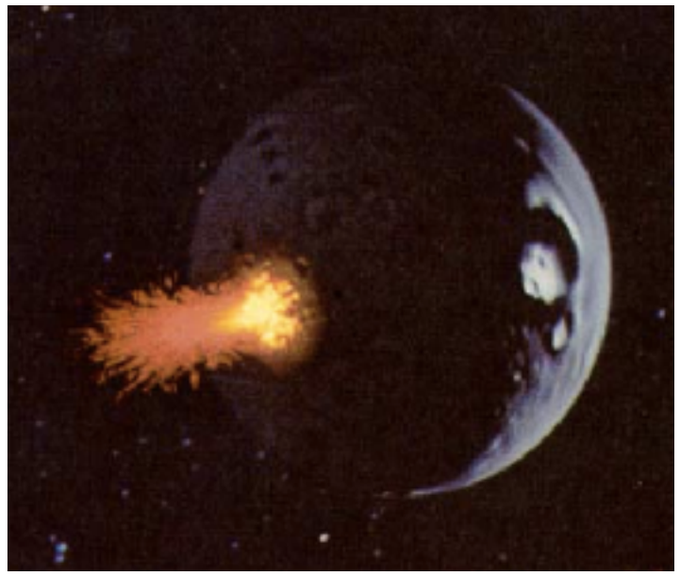
4 — Mass-spring systems

Particle systems

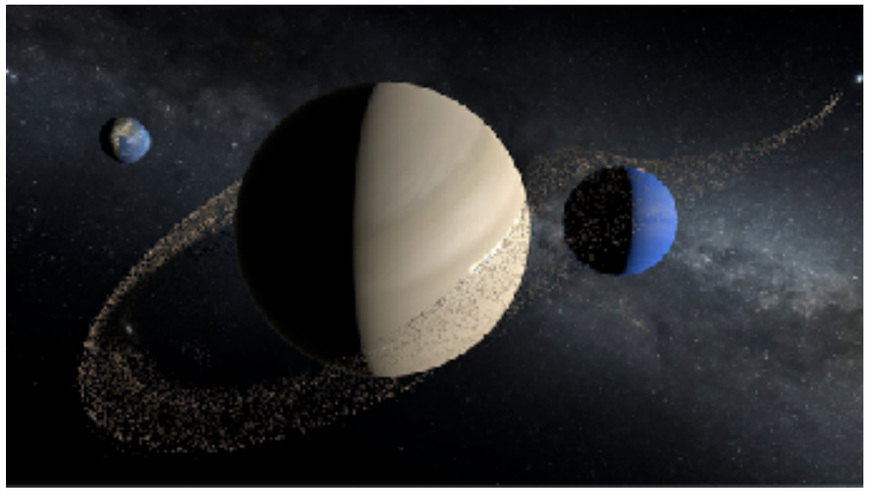
Consider n point masses in $\mathbb{R}^3$

ith particle has mass m_i, position $\mathbf{x}_i$, velocity $\mathbf{v}_i$

- External forces on particles: gravity, air resistance, …
- Internal forces between particles
  - Inverse-square attraction: solar system, galaxies
  - Springs between pairs of particles



[Reeves 1983]



Universe Sandbox

Springs

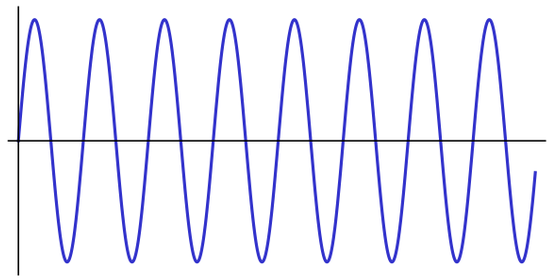
A spring is defined by its rest length ℓ_0 and spring constant k_s.

- In 1D: $f = -k_s(\ell - \ell_0)$
- In general: $\mathbf{f}_{ij} = -k_s\,(\|\mathbf{x}_i - \mathbf{x}_j\| - \ell_0)\,\hat{\mathbf{d}}_{ij}$
  where $\hat{\mathbf{d}}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)/\|\mathbf{x}_i - \mathbf{x}_j\|$

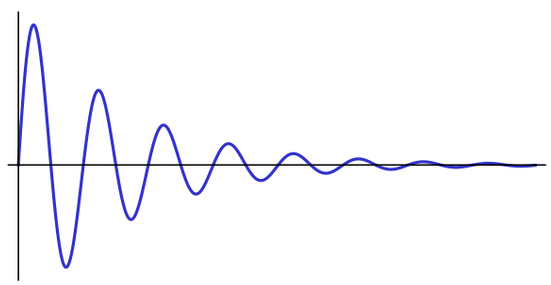
Or, define potential energy $U_{ij} = \frac{1}{2}\,k_s\,(\|\mathbf{x}_{ij}\| - \ell_0)^2$

- Then $\mathbf{f}_{ij} = -\partial U_{ij}/\partial \mathbf{x}_i$

Also add a damping force $-k_d\,(\mathbf{v}_{ij} \cdot \hat{\mathbf{d}}_{ij})\,\hat{\mathbf{d}}_{ij}$



$k_d = 0$



$k_d > 0$

Particle dynamics

Total force $\mathbf{f}_i$ on particle i = sum of all forces acting on it
(which can depend on positions, velocities of all other particles!)

Equations of motion:

$$\frac{\mathrm{d}^2\mathbf{x}_i}{\mathrm{d}t^2} = \frac{\mathbf{f}_i}{m_i} \quad \text{for all } i$$

or in 1st-order form:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} &= \mathbf{v}_i\,, \\ \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} &= \frac{\mathbf{f}_i}{m_i} \quad \text{for all } i \end{aligned}$$

System of ODEs in $2n$ vector-valued variables $\mathbf{x}_i$, $\mathbf{v}_i$

State space

Collect all particle positions into a single vector $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$

Similarly, collect velocities $\mathbf{v}$ and net forces $\mathbf{f}$

Equations of motion:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{bmatrix}, \qquad \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1/m_1 \\ \mathbf{f}_2/m_2 \\ \vdots \\ \mathbf{f}_n/m_n \end{bmatrix}$$

Particle dynamics, compactly

Introduce the inertia matrix

$$\mathbf{M} = \begin{bmatrix} m_1\mathbf{I} & & & \\ & m_2\mathbf{I} & & \\ & & \ddots & \\ & & & m_n\mathbf{I} \end{bmatrix}$$

So…

$$\mathrm{d}\mathbf{x}/\mathrm{d}t = \mathbf{v},$$
$$\mathrm{d}\mathbf{v}/\mathrm{d}t = \mathbf{M}^{-1}\,\mathbf{f}(\mathbf{x},\mathbf{v})$$

Forward Euler

$$\mathrm{d}\mathbf{y}/\mathrm{d}t = \varphi(\mathbf{y}) \quad \rightarrow \quad \mathbf{y}^1 = \mathbf{y}^0 + \Delta t\, \varphi(\mathbf{y}^0)$$

Apply it to the equations of motion:

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix}, \qquad \varphi(\mathbf{y}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}, \mathbf{v}) \end{bmatrix}$$

We get a simple update rule:

$$\mathbf{x}^1 = \mathbf{x}^0 + \Delta t\, \mathbf{v}^0$$
$$\mathbf{v}^1 = \mathbf{v}^0 + \Delta t\, \mathbf{M}^{-1}\, \mathbf{f}(\mathbf{x}^0, \mathbf{v}^0)$$

Forward Euler

$$\mathbf{x}^1 = \mathbf{x}^0 + \Delta t\, \mathbf{v}^0$$
$$\mathbf{v}^1 = \mathbf{v}^0 + \Delta t\, \mathbf{M}^{-1}\, \mathbf{f}(\mathbf{x}^0, \mathbf{v}^0)$$

Reinterpret in terms of particles:

$$\mathbf{x}_i^1 = \mathbf{x}_i^0 + \Delta t\, \mathbf{v}_i^0$$
$$\mathbf{v}_i^1 = \mathbf{v}_i^0 + \Delta t\, \mathbf{f}_i^0 / m_i$$

Implementation:

- Attach a force accumulator $\mathbf{f}_i$ to each particle
- For each force term (e.g. spring): compute force, add to affected particles
- For each particle i: update $\mathbf{x}_i$ and $\mathbf{v}_i$

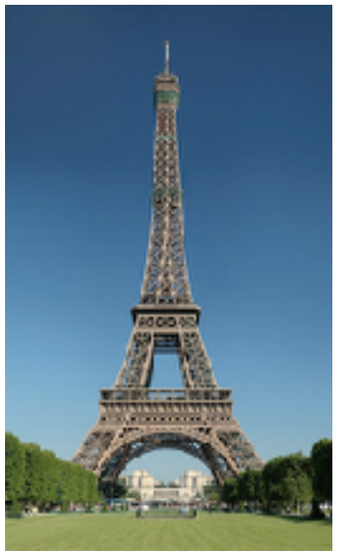
Details in Pixar notes

Mass-spring systems

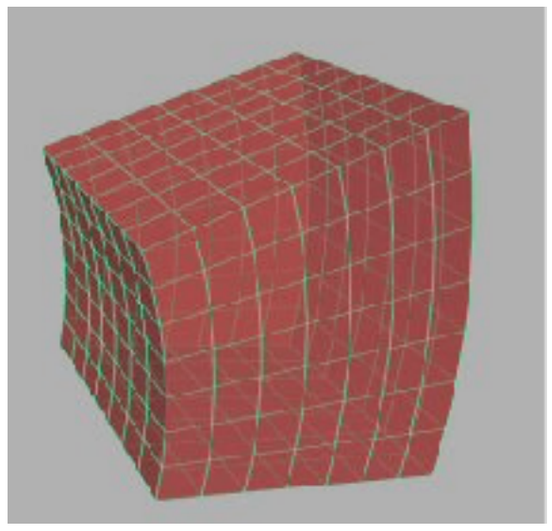
Mass-spring systems

Collection of particles connected by springs

- Natural model for a network of elastic rods / beams



- Also used as a phenomenological model for elastic deformable objects

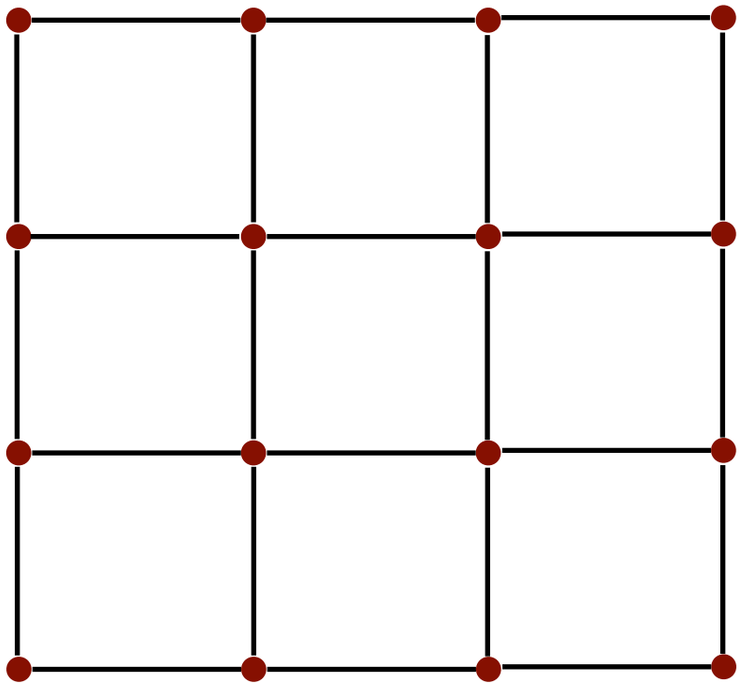


Mass-spring cloth [Provot ’95]

Particles in a rectangular grid

Different types of springs model different modes of elastic response

- **Structural springs**
  (modeling woven threads)

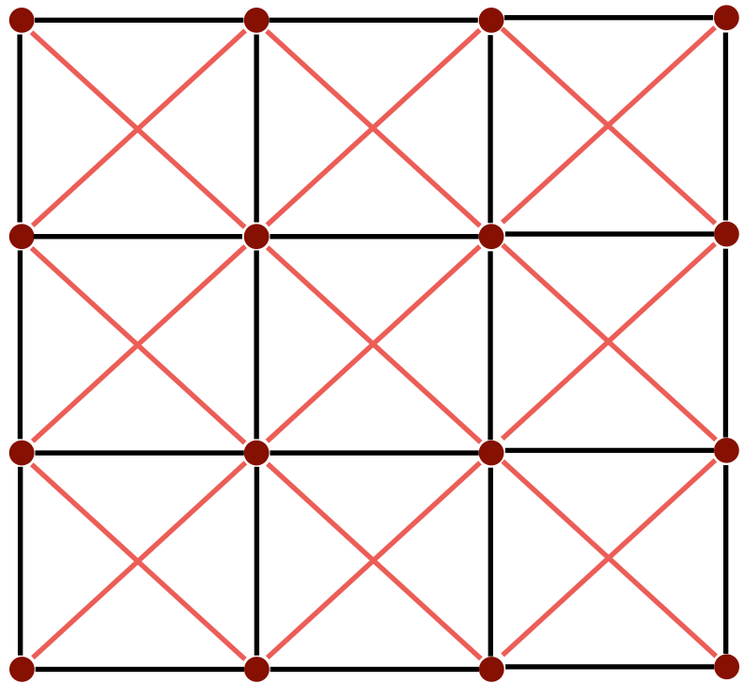


Mass-spring cloth [Provot ’95]

Particles in a rectangular grid

Different types of springs model different modes of elastic response

- **Structural springs**
- **Shear springs**
  (to prevent quads from collapsing)

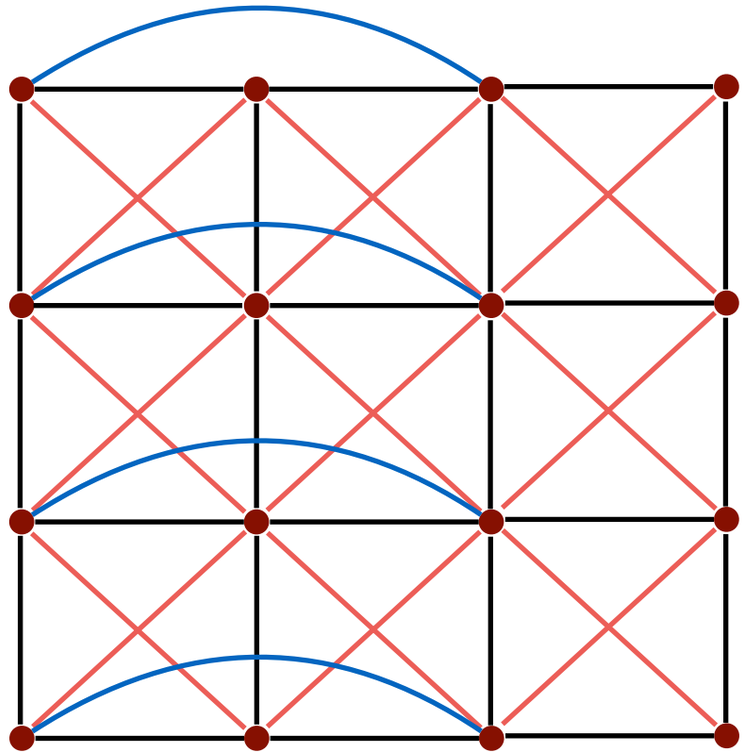


Mass-spring cloth [Provot ’95]

Particles in a rectangular grid

Different types of springs model different modes of elastic response

- **Structural springs**
- **Shear springs**
- **Flexion springs**
  (to resist excessive bending)

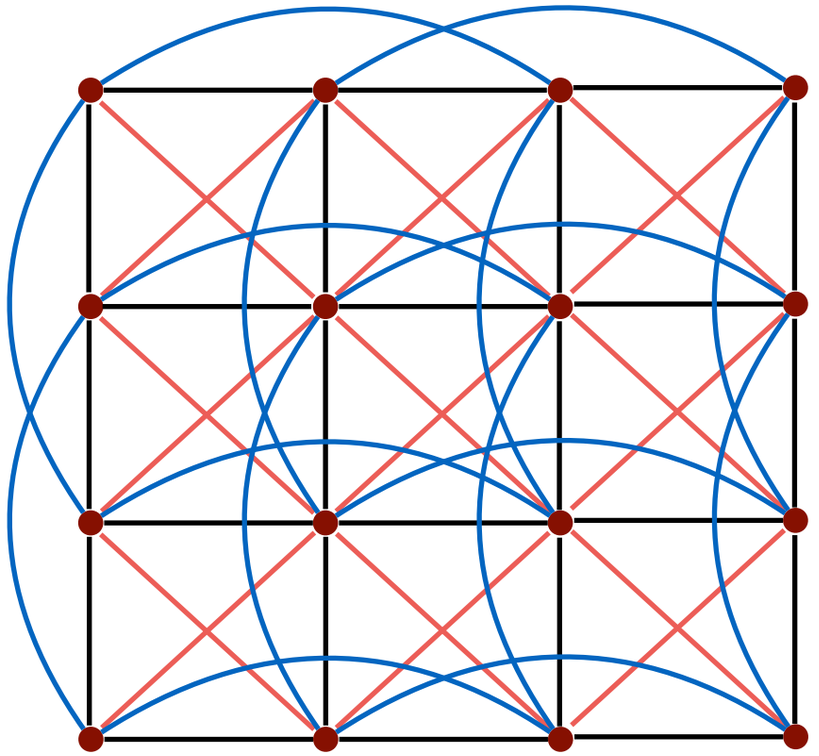


Mass-spring cloth [Provot ’95]

Particles in a rectangular grid

Different types of springs model different modes of elastic response

- **Structural springs**
- **Shear springs**
- **Flexion springs**




CLOTH
CG:SKEELOGY
cg.skeelogy.com
NEGATIVE EXAMPLE: NO SHEAR AND BEND SPRINGS


CLOTH
CG:SKEELOGY
cg.skeelogy.com
NEGATIVE EXAMPLE: NO BEND SPRINGS

CLOTH
CG:SKEELOGY
cg.skeelogy.com

Mass-spring cloth [Provot ’95]

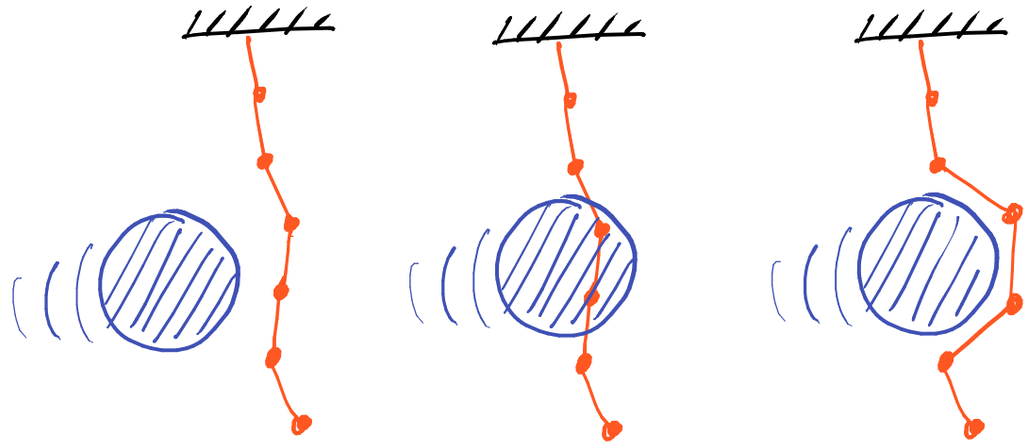
Most other forces discussed by Provot can be improved:

- Viscous damping
  - Use **spring damping** instead: conserves linear, angular momentum
- Wind forces
  - Better model: **quadratic** normal forces [Wejchert & Haumann 1991]
- Elongation constraints
  - Now better known as “strain limiting”, will discuss later as part of **constrained dynamics**

Collision handling

Collisions

Cloth shouldn’t pass through itself or through other objects



Collision handling is a big topic!
Today, only a basic scheme for collisions with simple fixed shapes

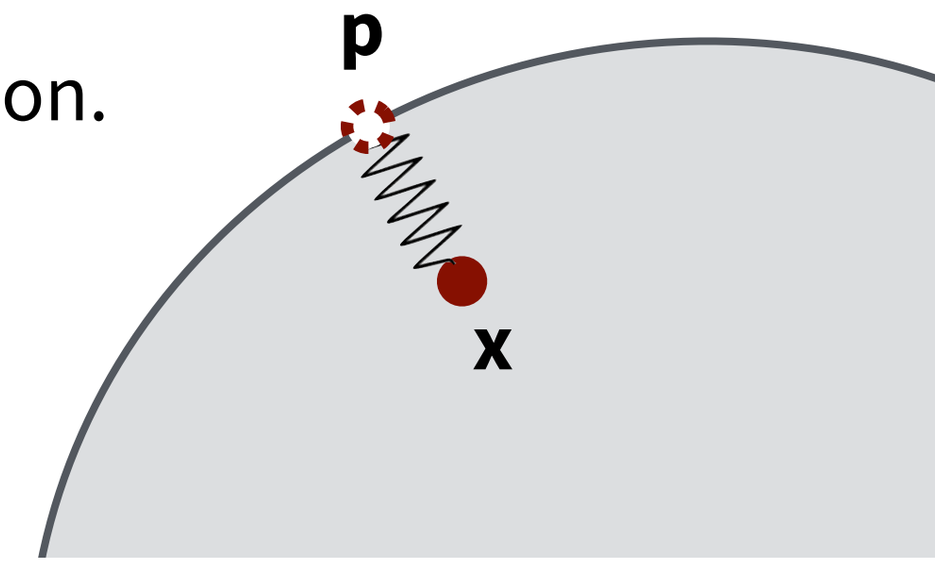
1. **Collision detection**: check if any particle is inside shape
   - Easy for simple shapes, e.g. sphere, plane, box
2. **Collision resolution**: do something to push it out
   - Simplest approach: penalty forces.
     Create a virtual zero-length spring pulling the particle out

Penalty forces

Find point **p** on surface closest to particle's current position **x**, create temporary spring with zero ℓ_0, large k_s

This models the **normal force** due to collision.

Better: choose **p** at some thickness outside the actual surface

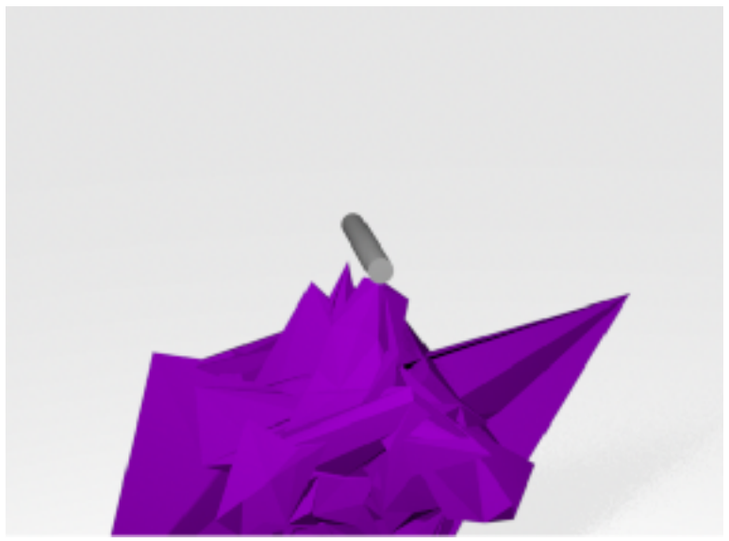


For **friction**, add tangential force due to Coulomb's law:

$$\|\mathbf{f}_t\| \le \mu \, \|\mathbf{f}_n\|, \quad \mathbf{f}_t \text{ is parallel to } -\mathbf{v}$$

Instability

Increasing the spring constant makes
makes forward Euler explode!
Have to decrease Δt to recover stability



[Dinev et al. 2018]

If $k_d = 0$, there is **no** stable time step

Other methods e.g. backward Euler have much better stability, even if same order of accuracy!