07 Deformation models
Working out our spring force from the energy

Start with the spring energy

\[ E_{ij}(x) = \frac{1}{2} k_s (\|x_i - x_j\| - l_0)^2 \] (this is the contribution of one spring to the total system energy)

Force is minus the gradient of energy

\[ f_i(x) = -\frac{\partial E}{\partial x_i}(x) \] (remember \( x \) is a big vector of all the positions; this partial derivative is zero for all the particles that are not connected to this particular spring)

Take the computation one step at a time:

- derivative of \( x_i - x_j \) is \( I \) wrt. \( x_i \) and \(-I\) wrt. \( x_j \)
- derivative of \( \|\mathbf{v}\| \) wrt. \( \mathbf{v} \) is \( \hat{\mathbf{v}} \)
- derivative of \( E_{ij} \) wrt \( \|\mathbf{v}\| \) is \( k_s (\|\mathbf{v}\| - l_0) \)
- put it all together: \( f_i = -\frac{\partial E}{\partial x_i} = -k_s (\|x_i\| - l_0) \hat{x}_{ij} \) and \( f_j = -\frac{\partial E}{\partial x_j} = k_s (\|x_j\| - l_0) \hat{x}_{ij} \)

where \( x_{ij} = x_i - x_j \)
Alternative “variational” notation

**Derivative is a linear transformation; write down the output**

- instead of \( \frac{\partial f}{\partial x} = A \) write \( \delta f = A \delta x \)

- when the matrix \( A \) is awkward to write down this can be neater…

- \( \delta x_{ij} = \delta x_i - \delta x_j \)

- \( \delta \|v\| = \hat{v} \cdot \delta v \)

- \( \delta E = k_s(l - l_0) \delta l \)

- substitute to get \( \delta E = k_s(\|x_{ij}\| - l_0) \hat{x}_{ij} \cdot \delta x_i - k_s(\|x_{ij}\| - l_0) \hat{x}_{ij} \cdot \delta x_j \)

- read off \( f_i \) and \( f_j \)
Deformable models

Mass-spring models can get you somewhere
  • but only so far
  • they were used a lot back in the Old Days

They have their limitations
  • hard to separate different stiffnesses (e.g. bend/shear springs contribute to stretch)
  • hard to control preservation of volume in deformations
  • hard to make them agree with measurements

Let's keep the idea of deriving forces from energies
  • define energies to get the behavior we want
  • borrow energies from other fields to get more accurate models
Example: hinge energy

We made a rope before using linear springs
- connect springs between every other point
- when rope bends, the springs fight one another, indirectly cause bending resistance

More direct approach
- just make the energy depend on the bending angle $\theta$ (well, $\sin \frac{\theta}{2}$)

$$E = k \sin \frac{\theta}{2} = \frac{k}{2} (1 - \cos \theta) \quad \text{equiv.} \quad E = -\frac{k}{2} \cos \theta.$$
A deforming object is described by a time varying function

- \( x = \phi(X, t) \)
- maps the *rest position* of a chunk of material to its current *deformed* position
- aka. a map from *material* space to *world* space
- varies as a function of time
The material of the deformable object “wants” to return to the rest shape

- how do we describe this behavior exactly?
- bits of material can’t communicate at a distance or “know” where they are in space
- all interactions are \textit{local} — the motion of a point depends only on its neighborhood

Result: deformation models are based only on the \textit{derivative} of $\phi$

- $\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ or $\delta \mathbf{x} = \mathbf{F} \delta \mathbf{X}$
- $\mathbf{F}$ is a matrix—2x2 or 3x3 depending on the dimension of the simulation
This is all very abstract — how do I compute it for a deforming mesh?

- very much like the computation used to get tangent vectors on a surface for shading
- in 2D, a triangle defines a unique affine map; in 3D a tetrahedron does the same
- can get that linear map by looking at triangle edge vectors

\[
[x_1 - x_0 \ x_2 - x_0] = F [X_1 - X_0 \ X_2 - X_0]
\]

\[
D = FD_0
\]

\[
F = DD_0^{-1}
\]