Notes

• Assignment #1
  • Uniform distribution in a disk:
    • Either pick points uniformly in square and discard those outside disk
    • Or take $r=\sqrt{\text{rand}}$, $\theta=2\pi \cdot \text{rand}$
  • Shortcut: if you implement Backward Euler and Forward Euler, just reuse them in Trapezoidal Rule

Constrained Dynamics

• Want “natural” dynamics but subject to constraint
• Last time: work with regular system, but add extra forces/impulses to satisfy constraint (at least approximately)
• Now: get rid of constraint equation altogether
  • Parameterize system so constraint automatically satisfied
  • Last time, the hard part was satisfying the constraint
  • This time, the hard part is satisfying physics!

Generalized Coordinates

• Say “positions” of system in vector $x$
• Constraint $C(x)=0$
• Find parameterization of the constraint manifold $x=X(q)$
  • $C(X(q))=0$ for all $q$
  • For every $x$ with $C(x)=0$, there is a $q$ s.t. $x=X(q)$
• The $q$ vector is the generalized coordinates
• Example: pendulum - 6 $x$ coordinates with 5 constraints, or 1 $q$ gen. coordinate (angle)
• No redundancy: cannot drift, should be fast for lots of constraints

Problems Ahead

• Math can get fairly nasty if the parameterization isn’t simple
  • Many people use Maple/Mathematica/etc. to crunch the expressions, generate code
• Parameterization could have pitfalls
  • [Gimbal-lock]
  • Some degenerate redundancies (multiple values of $q$ mapping to same $x$)
  • End up with ill-conditioned system (in the limit, underdetermined: more than one direction for $q$ to evolve)
General flavour

• Just look at constraint-free reparameterization (e.g. going to spherical coordinates)
• Say \( x=X(q) \), and inverse map also is well defined: \( q=Q(x) \)

\[
\dot{q} = \frac{\partial Q}{\partial x} \dot{x} = \frac{\partial X}{\partial q} \dot{q}
\]

\[
\ddot{q} = \frac{\partial Q}{\partial x} \ddot{x} + \frac{\partial^2 Q}{\partial x^2} : \dot{x} \dot{x}^T
\]

Getting rid of x’s...

\[
\ddot{q} = \frac{\partial Q}{\partial x} M^{-1} F + \frac{\partial^2 Q}{\partial x^2} : \left( \frac{\partial X}{\partial q} \dot{q} \dot{q}^T \frac{\partial X^T}{\partial q} \right)
\]

\[
= \frac{\partial X}{\partial q}^{-1} M^{-1} F + \frac{\partial^2 Q}{\partial x^2} \left( \frac{\partial X}{\partial q} \dot{q} \dot{q}^T \frac{\partial X^T}{\partial q} \right)
\]

\[
= \tilde{M}^{-1} \left( \tilde{F} + \tilde{F}_{\text{inertial}} \right)
\]

Let’s go on

• Not so important for passive motion
  • Critical for robotics
  • Important for human animation, but often better to directly specify joint angles (active motion anyways!)
  • If joint angles scripted, still can have character translate/rotate thorough space
    • Same as a rigid body (angular momentum is conserved) except inertia tensor in object space changes in time
  • For the occasional use of constraints in passive motion, easier to use soft constraints and/or Lagrange multipliers

Deformable Objects

• In reality, no such thing as a rigid body
• Lots of things aren’t anywhere close
  • All of your body except your bones
  • Clothing, and most other thin objects
  • Damaged/fractured objects
  • Water and other fluids
  • …
• Lots of degrees of freedom
  • -> painful to animate
• The math we need: continuum mechanics
Lagrangian vs. Eulerian

- Continuum: motion of a chunk of matter depends on nearby matter
- Two ways of looking at it
- Lagrangian: (e.g. particle systems)
  - Identify chunks of matter, track their positions (and velocities, accelerations, etc.) over time
- Eulerian: (will come later)
  - Forget identities of chunks of matter, instead just focus on how matter flows through space
  - Track velocity (and material properties) at fixed points in space
- [draw it - rigid chunk example]

Examples

- Elastic object, small deformation
  - Elastic means when force is removed, will try to return to original shape
  - E.g. [solid rubber ball]
  - Lagrangian works great
  - Eulerian - might have difficulty
- Completely fluid object, large deformation
  - E.g. [coffee]
  - Lagrangian has problems
  - Eulerian - works great

Elastic objects

- Simplest model: masses and springs
- Split up object into regions
- Integrate density in each region to get mass (if things are uniform enough, perhaps equal mass)
- Connect up neighbouring regions with springs
  - Careful: need chordal graph
- Now it's just a particle system
  - When you push on a node, neighbours pulled along with it, etc.

Masses and springs

- But: how strong should the springs be? Is this good in general?
  - [anisotropic examples]
- General rule: we don’t want to see the mesh in the output
  - Avoid “grid artifacts”
  - We of course will have numerical error, but let’s avoid obvious patterns in the error
1D masses and springs

- Look at a homogeneous elastic rod, length 1, linear density \( \rho \)
- Parameterize by \( p \) (\( x(p) = p \) in rest state)
- Split up into intervals/springs
  - \( 0 = p_0 < p_1 < \ldots < p_n = 1 \)
  - Mass \( m_i = \rho (p_{i+1} - p_{i-1})/2 \) (+ special cases for ends)
  - Spring \( i+1/2 \) has rest length and force
    \[
    f_{i+1/2} = k_{i+1/2} \frac{x_{i+1} - x_i - L_{i+1/2}}{L_{i+1/2}} = k_{i+1/2} \frac{x_i - x_{i+1} - L_{i-1/2}}{L_{i-1/2}}
    \]

Young’s modulus

- So each spring should have the same \( k \)
  - Note we divided by the rest length
  - Some people don’t, so they have to make their constant scale with rest length
- The constant \( k \) is a material property (doesn’t depend on our discretization) called the Young’s modulus
  - Often written as \( E \)
- The one-dimensional Young’s modulus is simply force per percentage deformation

Figuring out spring constants

- So net force on \( i \) is
  \[
  F_i = k_{i+1/2} \frac{x_{i+1} - x_i - L_{i+1/2}}{L_{i+1/2}} - k_{i-1/2} \frac{x_i - x_{i+1} - L_{i-1/2}}{L_{i-1/2}}
  \]
  \[
  = k_{i+1/2} \frac{x_{i+1} - x_i}{p_{i+1} - p_i} - k_{i-1/2} \frac{x_i - x_{i+1}}{p_i - p_{i+1}}
  \]
- We want mesh-independent response (roughly), e.g. for static equilibrium
  - Rod stretched the same everywhere: \( x_i = \alpha p_i \)
  - Then net force on each node should be zero (add in constraint force at ends…)

The continuum limit

- Imagine \( \Delta p \) (or \( \Delta x \)) going to zero
  - Eventually can represent any kind of deformation
  - [note force and mass go to zero too]
    \[
    \ddot{x}(p) = \frac{1}{\rho} \frac{\partial}{\partial p} \left( E(p) \left( \frac{\partial}{\partial \alpha} x(p) - 1 \right) \right)
    \]
  - If density and Young’s modulus constant,
    \[
    \frac{\partial^2 x}{\partial t^2} = \frac{E}{\rho} \frac{\partial^2 x}{\partial p^2}
    \]
Sound waves

- Try solution \( x(p,t)=x_0(p-ct) \)
- And \( x(p,t)=x_0(p+ct) \)
- So speed of sound in rod is \( \sqrt{\frac{E}{\rho}} \)
- Courant-Friedrichs-Levy (CFL) condition:
  - Numerical methods only will work if information transmitted numerically at least as fast as in reality (here: the speed of sound)
  - Usually the same as stability limit for explicit methods [what are the eigenvalues here]
  - Implicit methods transmit information infinitely fast

Why?

- Are sound waves important?
  - Visually? Usually not
  - However, since speed of sound is a material property, it can help us get to higher dimensions
  - Speed of sound in terms of one spring is \( c = \sqrt{\frac{kL}{m}} \)
  - So in higher dimensions, just pick \( k \) so that \( c \) is constant
  - \( m \) is mass around spring [triangles, tets]

Damping

- Figuring out how to scale damping is more tricky
- Go to differential equation (no mesh)
  \[
  \frac{\partial^2 x}{\partial t^2} = \frac{1}{\rho} \frac{\partial}{\partial p} \left( E \left( \frac{\partial x}{\partial p} - 1 \right) + D \frac{\partial v}{\partial p} \right)
  \]
- So spring damping should be
  \[
  f_{i+1/2} = k_{i+1/2} \frac{x_{i+1} - x_i - L_{i+1/2}}{L_{i+1/2}} + d_{i+1/2} \frac{v_{i+1} - v_i}{L_{i+1/2}}
  \]

Extra effects with springs

- (Brittle) fracture
  - Whenever a spring is stretched too far, break it
  - Issue with loose ends...
- Plasticity
  - Whenever a spring is stretched too far, change the rest length part of the way
Mass-spring problems

- [anisotropy]
- [stretching, Poisson’s ratio]
- [2D bending]
- More generally: implicit integration? contact/collision?