Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

An Introduction to Physics-based Animation

Adam Bargteil

University of Maryland,
Baltimore County

Tamar Shinar

University of California,
Riverside

I. *A Simple Start:* Particle Dynamics

Let's jump right in and consider the problem of tracing a particle through a velocity field

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change $\longrightarrow$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change $\longrightarrow$ Difference

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change $\longrightarrow$ Difference

Differential Equation

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change $\longrightarrow$ Difference

Ordinary Differential Equation

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Change $\longrightarrow$ Difference

First-order Ordinary Differential Equation

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Simple

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Simple

Powerful

Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

Simple

Powerful

Instructive

Euler's Method

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

$$\epsilon \longrightarrow \Delta t$$

The Derivative

$$\frac{d\mathbf{x}_p(t)}{dt} = \lim_{\epsilon \rightarrow 0} \frac{\mathbf{x}_p(t + \epsilon) - \mathbf{x}_p(t)}{\epsilon}$$

$$\epsilon \longrightarrow \Delta t$$

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$
$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t} = \mathbf{v}(\mathbf{x}_p, t)$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$

Euler's Method

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

$$\frac{d\mathbf{x}_p(t)}{dt} \stackrel{+}{=} \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$

The Great Tradeoff

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

The Great Tradeoff

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

As Δt decreases

The Great Tradeoff

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

As Δt decreases
the approximation gets better

The Great Tradeoff

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

As Δt decreases
the approximation gets better
but

The Great Tradeoff

$$\frac{d\mathbf{x}_p(t)}{dt} \approx \frac{\mathbf{x}_p(t + \Delta t) - \mathbf{x}_p(t)}{\Delta t}$$

As Δt decreases
the approximation gets better
but
the computational cost increases

Let's consider another problem

In the real world

$$\mathbf{f} = m\mathbf{a}$$

Another Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d^2 \mathbf{x}_p(t)}{dt^2} = \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

Another Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\frac{d^2 \mathbf{x}_p(t)}{dt^2} = \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

Second-order Ordinary Differential Equation

Another Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\mathbf{v}_p(0) = \mathbf{v}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{d\mathbf{v}_p(t)}{dt} = \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

Another Initial Value Problem

$$\mathbf{x}_p(0) = \mathbf{x}_0$$

$$\mathbf{v}_p(0) = \mathbf{v}_0$$

$$\frac{d\mathbf{x}_p(t)}{dt} = \mathbf{v}(\mathbf{x}_p, t)$$

$$\frac{d\mathbf{v}_p(t)}{dt} = \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

Coupled First-order Ordinary Differential Equations

Euler's Method (Again)

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p(t) + \Delta t \cdot \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}_p(t + \Delta t)$$

Euler's Method (Again)

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p(t) + \Delta t \cdot \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}_p(t + \Delta t)$$

Symplectic Euler


```
struct Particle {  
    double mass;  
    Eigen::Vector3d pos, vel, frc;  
};
```

```
foreach (p : particles) {  
    p.frc = 0.0;  
}
```

```
foreach (f : forces) {  
    foreach (p : forces.affectedParticles) {  
        p.frc += f.computeForce(p);  
    }  
}
```

```
foreach (p : particle) {  
    p.vel += dt * p.frc / p.mass;  
    p.pos += dt * p.vel;  
}
```


Check out Karl Sim's *Particle Dreams*

Let's Add Springs!

Springs

$$\mathbf{f} = -k\mathbf{d}$$

Springs

$$\mathbf{f}_p = -k\mathbf{x}_p$$

Springs

$$\mathbf{f}_p = -k\mathbf{x}_p$$

non-zero rest length?

Springs

$$\mathbf{f}_p = -k\mathbf{x}_p$$

$$\mathbf{f}_p = -k \left(\|\mathbf{x}_p\| \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k\mathbf{x}_p$$

$$\mathbf{f}_p = -k \left(\|\mathbf{x}_p\| \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

$$\mathbf{f}_p = -k \left(\|\mathbf{x}_p\| - r \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k \left(\|\mathbf{x}_p\| - r \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k \left(\|\mathbf{x}_p\| - r \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

$$\text{Strain} \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right)$$

Springs

$$\mathbf{f}_p = -k (\|\mathbf{x}_p\| - r) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

$$\text{Strain} \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right)$$

$$\mathbf{f}_p = -k \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

arbitrary connection?

Springs

$$\mathbf{f}_p = -k \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

Springs

$$\mathbf{f}_p = -k \left(\frac{\|\mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_p}{\|\mathbf{x}_p\|}$$

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_q = -\mathbf{f}_p$$

Damping

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

Damping

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_p = k_d \left(\frac{\mathbf{v}_q - \mathbf{v}_p}{r} \cdot \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|} \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

Damping

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_p = k_d \underbrace{\left(\frac{\mathbf{v}_q - \mathbf{v}_p}{r} \cdot \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|} \right)}_{\substack{\text{relative} \\ \text{velocity}}} \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

Damping

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_p = k_d \left(\underbrace{\frac{\mathbf{v}_q - \mathbf{v}_p}{r}}_{\text{relative velocity}} \cdot \underbrace{\frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}}_{\text{spring direction}} \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

Damping

$$\mathbf{f}_p = k \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_p = k_d \underbrace{\left(\frac{\mathbf{v}_q - \mathbf{v}_p}{r} \cdot \right)}_{\text{relative velocity}} \underbrace{\frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}}_{\text{spring direction}} \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$

$$\mathbf{f}_p = \left[k_s \left(\frac{\|\mathbf{x}_q - \mathbf{x}_p\|}{r} - 1 \right) + k_d \left(\frac{(\mathbf{v}_q - \mathbf{v}_p) \cdot (\mathbf{x}_q - \mathbf{x}_p)}{r \|\mathbf{x}_q - \mathbf{x}_p\|} \right) \right] \frac{\mathbf{x}_q - \mathbf{x}_p}{\|\mathbf{x}_q - \mathbf{x}_p\|}$$


```
foreach (p : particles) {  
    p.frc = 0.0;  
}
```

```
foreach (s : springs) {  
    Eigen::Vector3d d = particles[s->j].pos - particles[s->i].pos;  
    double l = d.norm();  
    Eigen::Vector3d v = particles[s->j].vel - particles[s->i].vel;  
    Eigen::Vector3d frc = (params.k_s*((l / s->r) - 1.0) +  
        params.k_d*(v.dot(d)/(l*s->r))) * (d/l);  
    particles[s.i].frc += frc  
    particles[s.j].frc -= frc  
}
```

```
foreach (p : particle) {  
    p.vel += dt * p.frc / p.mass;  
    p.pos += dt * p.vel;  
}
```


Live Demo

II. Mathematical Models

Newton's Laws

Newtonian Mechanics

- ❖ Published in *Principia*, 1687
- ❖ Includes three laws of motion:
 - ❖ inertia
 - ❖ $f = ma$
 - ❖ action / reaction
- ❖ Idealized particle or point mass

Newton's First Law

*A body persists at rest or in uniform motion in a straight line
unless acted upon by a force*

- ❖ Law of Inertia
 - ❖ Defines an inertial frame of reference

Newton's Second Law ($\mathbf{f} = m\mathbf{a}$)

The rate of change of momentum of a body is directly proportional to the force applied to the body

$$\frac{d}{dt}m\mathbf{v}(t) = m\mathbf{a} = \mathbf{f}(t)$$

- ❖ The basis for evolving a system of interacting particles

Newton's Second Law ($\mathbf{f} = m\mathbf{a}$)

❖ leads to a system of ODEs

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t)$$

$$\dot{\mathbf{v}}(t) = \mathbf{a}(t) = \frac{1}{m}\mathbf{f}(t)$$

$$\mathbf{f}(t) = \mathbf{f}_e(t) + \mathbf{f}_g(t) + \mathbf{f}_s(t)$$

$$m\mathbf{a} = \mathbf{f}_e + \mathbf{f}_g + \mathbf{f}_s$$

Newton's Second Law ($f = ma$)

- ❖ To model a system of particles,
- ❖ characterize all the forces on each particle
- ❖ Start with some initial conditions and apply $f = ma$ to evolve in time

$$m\mathbf{a} = \mathbf{f}_e + \mathbf{f}_g + \mathbf{f}_s$$

Newton's Third Law (Action/Reaction)

For every action, there is an equal and opposite reaction

- ❖ If body 1 applies force $\mathbf{f}$ to body 2, then body 2 applies force $-\mathbf{f}$ to body 1

Newton's Third Law (Action/Reaction)

For every action, there is an equal and opposite reaction

- ❖ If body 1 applies force $\mathbf{f}$ to body 2, then body 2 applies force $-\mathbf{f}$ to body 1
- ❖ Example: Two particles connected by a spring force, equal/opposite pair of forces

Newton's Third Law (Action/Reaction)

For every action, there is an equal and opposite reaction

- ❖ If body 1 applies force $\mathbf{f}$ to body 2, then body 2 applies force $-\mathbf{f}$ to body 1
- ❖ Example: In the collision of two particles, equal/opposite pair of forces prevents interpenetration

Newton's Third Law (Action/Reaction)

For every action, there is an equal and opposite reaction

- ❖ If body 1 applies force $\mathbf{f}$ to body 2, then body 2 applies force $-\mathbf{f}$ to body 1
- ❖ Example: Particle resting on a surface, equal / opposite pair of forces prevents interpenetration

Newton's Third Law (Action/Reaction)

For every action, there is an equal and opposite reaction

- ❖ If body 1 applies force $\mathbf{f}$ to body 2, then body 2 applies force $-\mathbf{f}$ to body 1

- ❖ Example: Particle resting on a surface, equal / opposite pair of forces prevents interpenetration

Alternative: Variational Mechanics

- ❖ Newtonian Mechanics is one formulation of classical mechanics
 - ❖ Based on vectors in Cartesian space
- ❖ Another set of approaches is called “variational” and is based on a principle of least action
 - ❖ variational approaches let you use any set of coordinates

Conservation Laws

Conserved Quantities

- ❖ Cannot be created or destroyed!
- ❖ Includes mass, linear momentum, angular momentum, and energy

Conservation Laws

- ❖ Used in deriving evolution equations
- ❖ Inform choice of discrete approximation to continuous equations
- ❖ Implications for visual quality, numerical accuracy, and stability

Conservation of Mass

- ❖ Mass not created or destroyed (inexact)
- ❖ Mass naturally conserved in particle-based methods
 - ❖ Particles carry mass with them as they move
- ❖ Grid-based methods sometimes have issues with proper mass conservation

Conservation of Momentum

- ❖ By Newton's second law, if there is no net force on a body, i.e., $\mathbf{f} = \mathbf{0}$

$$\frac{d}{dt}m\mathbf{v} = \mathbf{0}$$

$$\Rightarrow m\mathbf{v}(t) = \text{constant}$$

- ❖ So the momentum of the particle is conserved
- ❖ Similarly, if there is no net torque on a body, angular momentum is constant

Conservation of Momentum

- ❖ Newton's third law equal/opposite also implies conservation of linear and angular momentum

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{f}_1 + (-\mathbf{f}_1) = \mathbf{0}$$

Conservation of Momentum

- ❖ Newton's third law equal/opposite also implies conservation of linear and angular momentum

$$\frac{d}{dt}\mathbf{L}(t) = \mathbf{r}_1 \times \mathbf{f}_1 + \mathbf{r}_2 \times (-\mathbf{f}_1) = (\mathbf{r}_1 - \mathbf{r}_2) \times \mathbf{f}_1 = \mathbf{0}$$

Conservation of Momentum

- ❖ Same holds for a collection of interacting particles!

Conservation of Energy

- ❖ Initial energy (potential)

$$mgh_0$$

- ❖ Conservation of energy

$$\frac{1}{2}mv(t)^2 + mgh(t) = mgh_0$$

- ❖ Speed when hits

$$v(t) = \sqrt{2gh_0}$$

Conservation of Energy

- ❖ Initial energy (potential)

$$mgh_0$$

- ❖ Conservation of energy

$$\frac{1}{2}mv(t)^2 + mgh(t) = mgh_0$$

- ❖ Speed when hits

$$v(t) = \sqrt{2gh_0}$$

Conservation of Energy

- ❖ Different numerical schemes have different energy conservation properties
- ❖ In many schemes, energy grows or decays nonphysically
 - ❖ instability (blow up), or
 - ❖ motion too damped

Conservation Laws for Continua

- ❖ At large enough length scales, model matter as a continuum rather than set of discrete particles

Conservation Laws for Continua

- ❖ To derive evolution equations for continua, consider arbitrary control volume Ω

Conservation Laws for Continua

- ❖ Conservation of Mass

$$\frac{d}{dt} \int_{\Omega} \rho dV = - \oint_{\partial\Omega} \rho \mathbf{v} \cdot \mathbf{n} dS$$

- ❖ Continuity Equation

$$\Rightarrow \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

Conservation Laws for Continua

- ❖ Local conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0$$

flux

Rigid Bodies

Rigid Body Idealization

- ❖ If deformation is negligible, a rigid body approximation is more efficient than a soft body model

Rigid Body Idealization

- ❖ Elastic forces are replaced with constraints that particles in the body remain a fixed distance apart

Rigid Body Idealization

- ❖ $3n$ DoF are replaced with 6 DoF! position and orientation $\mathbf{x}(t), \mathbf{R}(t)$

Rigid Body Kinematics

❖ Center of mass

$$\mathbf{x}_{com} = \frac{\sum_{i=1}^N m_i \mathbf{x}_i}{\sum_{i=1}^N m_i}.$$

Rigid Body Coordinates

Linear and Angular Velocity

❖ Linear velocity

$$\mathbf{v}(t) = \dot{\mathbf{x}}(t)$$

Linear and Angular Velocity

- ❖ Angular velocity

$$\omega(t)$$

Rigid Body Coordinates

particle position

$$\mathbf{p}(t) = \mathbf{x}(t) + \underbrace{\mathbf{R}(t)\mathbf{r}_0}_{\mathbf{r}(t)}$$

particle velocity

$$\dot{\mathbf{p}}(t) = \mathbf{v}(t) + \boldsymbol{\omega}(t) \times \mathbf{r}(t)$$

Linear and Angular Momentum

❖ Linear Momentum

$$\mathbf{P}(t) = \sum_{i=1}^N m_i \mathbf{v}_i(t)$$

(c.o.m. origin) $\Rightarrow \mathbf{P}(t) = m \mathbf{v}(t)$

❖ Angular Momentum

$$\mathbf{L}(t) = \sum_{i=1}^N \mathbf{r}_i(t) \times m_i \mathbf{v}_i(t)$$

(c.o.m. origin) $\Rightarrow \mathbf{L}(t) = \mathbf{I}(t) \omega(t)$

$\mathbf{I}(t)$: inertia tensor

Rigid Body Inertia Tensor

$$\mathbf{I}(t) = \sum_{i=1}^N m_i \left(\mathbf{r}_i^T \mathbf{r}_i \delta - \mathbf{r}_i \mathbf{r}_i^T \right)$$

$$= \mathbf{R}(t) \underbrace{\sum_{i=1}^N m_i \left(\mathbf{r}_{0i}^T \mathbf{r}_{0i} \delta - \mathbf{r}_{0i} \mathbf{r}_{0i}^T \right)}_{\mathbf{I}_0} \mathbf{R}(t)^T$$

$$= \mathbf{R}(t) \mathbf{I}_0 \mathbf{R}(t)^T.$$

Linear and Angular Momentum

- ❖ No net force =>
 - ❖ linear momentum and velocity constant
- ❖ No net torque =>
 - ❖ angular momentum constant
 - ❖ angular velocity not necessarily constant

$$\mathbf{P}(t) = m\mathbf{v}(t)$$

$$\mathbf{L}(t) = \mathbf{I}(t)\omega(t)$$

Newton's Second Law for Rigid Bodies

$$\frac{d}{dt} \begin{pmatrix} \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{f}(t) \\ \tau(t) \end{pmatrix}, \quad \mathbf{f}(t) = \sum \mathbf{f}_i$$
$$\tau(t) = \sum \mathbf{r}_i \times \mathbf{f}_i$$

❖ Summary

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}(t) \\ R(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \omega^*(t)R(t) \\ \mathbf{f}(t) \\ \tau(t) \end{pmatrix}$$

Soft Bodies

Adding Elasticity and Damping

$$\mathbf{f} = m\mathbf{a}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

Adding Elasticity and Damping

$$m\mathbf{a} = \mathbf{f}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

Adding Elasticity and Damping

$$m\mathbf{a} = \mathbf{f}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

Adding Elasticity and Damping

$$m\mathbf{a} = \mathbf{f}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

Adding Elasticity and Damping

$$m\mathbf{a} = \mathbf{f}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

Adding Elasticity and Damping

$$m\mathbf{a} = \mathbf{f}$$

$$\mathbf{K}(\mathbf{x} - \mathbf{u}) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

in terms of displacements $\mathbf{d}$

$$\mathbf{K}(\mathbf{d}) + \mathbf{D}(\dot{\mathbf{d}}) + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}_{ext}$$

K

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix
- ❖ Sparse, Symmetric, Diagonally Dominant, Row / Col sums 0

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix
- ❖ Sparse, Symmetric, Diagonally Dominant, Row / Col sums 0
- ❖ Positive semi-definite

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix
- ❖ Sparse, Symmetric, Diagonally Dominant, Row / Col sums 0
- ❖ Positive semi-definite
- ❖ Negative Jacobian of forces

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix
- ❖ Sparse, Symmetric, Diagonally Dominant, Row / Col sums 0
- ❖ Positive semi-definite
- ❖ Negative Jacobian of forces
- ❖ Negative Hessian of energy

K

- ❖ Generalization of spring stiffness, called *stiffness* matrix
- ❖ Sparse, Symmetric, Diagonally Dominant, Row / Col sums 0
- ❖ Positive semi-definite
- ❖ Negative Jacobian of forces
- ❖ Negative Hessian of energy

$$\mathbf{K} = \frac{-\partial \mathbf{f}_i}{\partial \mathbf{x}_j} = \frac{-\partial \eta}{\partial \mathbf{x}_i \partial \mathbf{x}_j}$$

So how do we calculate elastic forces?

Elastic Forces Cookbook

- ❖ Deformation Function (world pos, rest position)
- ❖ Deformation Gradient (deformation function)
- ❖ Strain (deformation gradient)
- ❖ Stress (strain)
- ❖ Energy (stress, strain)
- ❖ Forces (energy)

Deformation and Its Gradient

Deformation and Its Gradient

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}(\mathbf{u}_0) + \mathbf{F}(\mathbf{u} - \mathbf{u}_0)$$

Deformation and Its Gradient

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}(\mathbf{u}_0) + \mathbf{F}(\mathbf{u} - \mathbf{u}_0)$$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{u}} = \mathbf{F}$$

Deformation and Its Gradient

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}(\mathbf{u}_0) + \mathbf{F}(\mathbf{u} - \mathbf{u}_0)$$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{u}} = \mathbf{F}$$

Strain is

Strain is

- ❖ a measure of deformation

Strain is

- ❖ a measure of deformation
- ❖ dimensionless (i.e. has no units)

Strain is

- ❖ a measure of deformation
- ❖ dimensionless (i.e. has no units)
- ❖ a function of the deformation gradient

Strain Metrics

- ❖ Green's finite $\epsilon = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I})$
- ❖ Cauchy's infinitesimal $\epsilon = \frac{1}{2} (\mathbf{F}^T + \mathbf{F}) - \mathbf{I}$
- ❖ Co-rotated $\epsilon = \frac{1}{2} (\tilde{\mathbf{F}}^T + \tilde{\mathbf{F}}) - \mathbf{I}$

where $\mathbf{F} = \mathbf{Q}\tilde{\mathbf{F}}$

Stress

Stress

- ❖ stress is a function of strain

Stress

- ❖ stress is a function of strain
- ❖ stress has units of (Newton / meter²)

Stress

- ❖ stress is a function of strain
- ❖ stress has units of (Newton / meter²)
- ❖ models of stress-strain relationships can be highly complex, especially with organic materials

Linear Stress-strain Relationships

General, linear

$$\sigma = \mathbf{C}\epsilon$$

Linear Stress-strain Relationships

General, linear

$$\sigma = \mathbf{C}\epsilon$$

isotropic material

$$\sigma = \lambda \text{Tr}(\epsilon) \mathbf{I} + 2\mu\epsilon$$

Elastic Potential, Traction, Force

$$\eta = \frac{1}{2} \sigma : \epsilon = \frac{1}{2} \sum_{i,j} \sigma_{ij} \epsilon_{ij}$$

Elastic Potential, Traction, Force

$$\eta = \frac{1}{2} \sigma : \epsilon = \frac{1}{2} \sum_{i,j} \sigma_{ij} \epsilon_{ij}$$

$$\tau = \sigma \mathbf{n}$$

Elastic Potential, Traction, Force

$$\eta = \frac{1}{2} \sigma : \epsilon = \frac{1}{2} \sum_{i,j} \sigma_{ij} \epsilon_{ij}$$

$$\tau = \sigma \mathbf{n}$$

$$\mathbf{f}_i = -\frac{\partial \eta}{\partial \mathbf{x}_i}$$

Elastic Potential, Traction, Force

$$\eta = \frac{1}{2} \sigma : \epsilon = \frac{1}{2} \sum_{i,j} \sigma_{ij} \epsilon_{ij}$$

$$\boldsymbol{\tau} = \boldsymbol{\sigma} \mathbf{n}$$

$$\mathbf{f}_i = - \frac{\partial \eta}{\partial \mathbf{x}_i}$$

$$\mathbf{f} = \oint_{\partial R} \boldsymbol{\sigma} \mathbf{n} \, dS$$

Plasticity

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_p$$

Fluids

Fluids

Fluids

take shape
of container

can't support
shear stress

Fluids

can support
normal stress

Navier-Stokes Equations

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Navier-Stokes Equations

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Navier-Stokes Equations

$$\overset{\text{m}}{\rho} \underbrace{(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}_{\text{a}} = - \underbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}_{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density

p : pressure

$\mathbf{f}$: forces

$\mathbf{u}$: velocity

μ : viscosity

Navier-Stokes Equations

$$\overset{\text{m}}{\rho} \overbrace{(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u}}^{\text{f}} + \mathbf{f}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Navier-Stokes Equations

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Navier-Stokes Equations

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Navier-Stokes Equations

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

ρ : density p : pressure $\mathbf{f}$: forces

$\mathbf{u}$: velocity μ : viscosity

Material Derivative

$$\overset{\text{m}}{\rho} \overset{\text{a}}{\underbrace{(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}} = - \overset{\text{f}}{\underbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}}$$
$$\nabla \cdot \mathbf{u} = 0$$

Material Derivative

$$\mathbf{u}(\mathbf{x}, t)$$

Material Derivative

$$\begin{aligned}\mathbf{a}_p(t) &= \frac{d}{dt} \mathbf{v}_p(t) \\ &= \frac{d}{dt} \mathbf{u}(\mathbf{x}_p(t), t) \\ &= \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \frac{d\mathbf{x}_p}{dt} \right)\end{aligned}$$

$\mathbf{u}(\mathbf{x}, t)$

Material Derivative

$\mathbf{u}(\mathbf{x}, t)$

$$\begin{aligned}\mathbf{a}_p(t) &= \frac{d}{dt} \mathbf{v}_p(t) \\ &= \frac{d}{dt} \mathbf{u}(\mathbf{x}_p(t), t) \\ &= \underbrace{\left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \frac{d\mathbf{x}_p}{dt} \right)}_{\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}}\end{aligned}$$

$$\frac{D\mathbf{u}}{Dt} = \mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}$$

Pressure Forces

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

- ❖ maintains fluid volume, resisting compression/expansion
- ❖ forces fluid from areas of high pressure to low pressure

Viscous Forces

$$\overset{\text{m}}{\rho} \overbrace{(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

- ❖ internal friction
- ❖ Laplacian measures difference from neighbors

μ	(mPa · s)
μ_{air}	1.8×10^{-2}
μ_{water}	1
μ_{honey}	5×10^3

Viscous Forces: Solid Boundaries

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

❖ no-slip boundary condition

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{V}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma$$

External Forces

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$
$$\nabla \cdot \mathbf{u} = 0$$

- ❖ Gravity, surface tension, interaction forces, control forces, embedded structures

Incompressibility

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$

$$\nabla \cdot \mathbf{u} = 0$$

- ❖ Conservation of mass $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$
- ❖ Constant density $\Rightarrow \nabla \cdot \mathbf{u} = 0$

Incompressibility

$$\overbrace{\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{m a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$

$$\nabla \cdot \mathbf{u} = 0$$

- ❖ Net flow through boundary must be zero

$$0 = \int_{\Omega} \nabla \cdot \mathbf{u} = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}$$

Incompressibility

$$\overset{\text{m}}{\rho} \overbrace{(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u})}^{\text{a}} = - \overbrace{\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}}^{\text{f}}$$

$$\nabla \cdot \mathbf{u} = 0$$

- ❖ Net flow through boundary must be zero

$$0 = \int_{\Omega} \nabla \cdot \mathbf{u} = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}$$

III. Spatial Discretization

Lagrangian vs. Eulerian

Reference Frames

- ❖ An Eulerian reference frame is fixed.
- ❖ A Lagrangian reference frame moves with the material.

Reference Frames

Eulerian

Lagrangian

Reference Frames

Eulerian

Lagrangian

Reference Frames

Eulerian

$$\frac{\partial y}{\partial t} + \mathbf{u} \cdot \nabla y$$

Lagrangian

$$\frac{dy}{dt}$$

Grids, Meshes, and Particles

Regular Grids

Advantages

- ❖ Simple
- ❖ Fast operations (e.g. point location)
- ❖ Can take advantage of structure for efficiency

Disadvantages

- ❖ Difficult to track shape over time
- ❖ Difficult to handle non-grid-aligned boundaries

Staggered Grid

Meshes (Simplicial Complexes)

Advantages

- ❖ Easy to map to previous points in time
- ❖ Can conform to boundaries

Disadvantages

- ❖ Difficult to generate meshes
- ❖ Difficult to perform some operations (e.g. point location)

Particles

Advantages

- ❖ Simple
- ❖ Easy to map to previous points in time

Disadvantages

- ❖ Difficult to perform integration because they don't partition space

Hybrid Structures

Advantages

- ❖ Advantages of underlying structures

Disadvantages

- ❖ Complexity
- ❖ Computational and accuracy costs from mapping between structures

Interpolation

Interpolation

- ❖ Samples stored at discrete points on grid, mesh, or particles
- ❖ Elsewhere, must be interpolated there

Interpolation

- ❖ Example: semi-Lagrangian advection

Interpolation

- ❖ Example: semi-Lagrangian advection

Interpolation

- ❖ Example: semi-Lagrangian advection

Interpolation

- ❖ Example: semi-Lagrangian advection

Linear Interpolation (1D)

- ❖ $f(0) = f_1$
- ❖ $f(1) = f_2$
- ❖ weights sum to 1
- ❖ weight is length opposite point

$$f(t) = (1 - t)f_1 + tf_2.$$

Bilinear Interpolation (2D)

- ❖ $f(0, 0) = f_1, \dots$

- ❖ $f(0, 1) = f_4$

- ❖ weights sum to 1

- ❖ weight is area opposite point

Bilinear Interpolation (2D)

$$f(s, t) = (1 - s)(1 - t)f_1 + s(1 - t)f_2 + stf_3 + (1 - s)tf_4$$

Bilinear Interpolation (2D)

$$f(s, t) = (1-s)(1-t)f_1 + s(1-t)f_2 + stf_3 + (1-s)tf_4$$

Bilinear Interpolation (2D)

$$f(s, t) = (1 - s)(1 - t)f_1 + s(1 - t)f_2 + stf_3 + (1 - s)tf_4$$

Bilinear Interpolation (2D)

$$f_5 = (1 - t)f_1 + tf_4$$

$$f_6 = (1 - t)f_2 + tf_3$$

$$f(s, t) = (1 - s)(1 - t)f_1 + s(1 - t)f_2 + stf_3 + (1 - s)tf_4$$

Trilinear Interpolation (3D)

$$\begin{aligned} f(s, t, u) = & \\ & (1 - s)(1 - t)(1 - u)f_1 \\ & + s(1 - t)(1 - u)f_2 \\ & + st(1 - u)f_3 \\ & + (1 - s)t(1 - u)f_4 \\ & + (1 - s)(1 - t)uf_5 \\ & + s(1 - t)uf_6 \\ & + stu f_7 \\ & + (1 - s)tu f_8 \end{aligned}$$

Trilinear Interpolation (3D)

$$\begin{aligned} f(s, t, u) = & \\ & (1-s)(1-t)(1-u)f_1 \\ & + s(1-t)(1-u)f_2 \\ & + st(1-u)f_3 \\ & + (1-s)t(1-u)f_4 \\ & + (1-s)(1-t)uf_5 \\ & + s(1-t)uf_6 \\ & + \boxed{stu}f_7 \\ & + (1-s)tu f_8 \end{aligned}$$

Trilinear Interpolation (3D)

$$\begin{aligned} f(s, t, u) = & \\ & (1 - s)(1 - t)(1 - u)f_1 \\ & + s(1 - t)(1 - u)f_2 \\ & + st(1 - u)f_3 \\ & + (1 - s)t(1 - u)f_4 \\ & + (1 - s)(1 - t)uf_5 \\ & + s(1 - t)uf_6 \\ & + stu f_7 \\ & + (1 - s)tu f_8 \end{aligned}$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

$$\alpha = \frac{\text{area}(p, b, c)}{\text{area}(a, b, c)},$$

$$\beta = \frac{\text{area}(p, c, a)}{\text{area}(a, b, c)},$$

$$\gamma = \frac{\text{area}(p, a, b)}{\text{area}(a, b, c)}.$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

$$\alpha = \frac{\text{area}(p, b, c)}{\text{area}(a, b, c)},$$

$$\beta = \frac{\text{area}(p, c, a)}{\text{area}(a, b, c)},$$

$$\gamma = \frac{\text{area}(p, a, b)}{\text{area}(a, b, c)}.$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

$$\alpha = \frac{\text{area}(p, b, c)}{\text{area}(a, b, c)},$$

$$\beta = \frac{\text{area}(p, c, a)}{\text{area}(a, b, c)},$$

$$\gamma = \frac{\text{area}(p, a, b)}{\text{area}(a, b, c)}.$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

$$\alpha = \frac{\text{area}(p, b, c)}{\text{area}(a, b, c)},$$

$$\beta = \frac{\text{area}(p, c, a)}{\text{area}(a, b, c)},$$

$$\gamma = \frac{\text{area}(p, a, b)}{\text{area}(a, b, c)}.$$

Barycentric Coordinates

$$f_p = f(\alpha, \beta, \gamma) = \alpha f_a + \beta f_b + \gamma f_c$$

$$\alpha = \frac{\text{area}(p, b, c)}{\text{area}(a, b, c)},$$

$$\beta = \frac{\text{area}(p, c, a)}{\text{area}(a, b, c)},$$

$$\gamma = \frac{\text{area}(p, a, b)}{\text{area}(a, b, c)}.$$

Barycentric Coordinates

❖ Coordinates at a vertex

$$\alpha = 1$$

$$\beta = 0$$

$$\gamma = 0$$

Barycentric Coordinates

❖ Coordinates at a vertex

$$\alpha = 0$$

$$\beta = 1$$

$$\gamma = 0$$

Barycentric Coordinates

❖ Coordinates at a vertex

$$\alpha = 0$$

$$\beta = 0$$

$$\gamma = 1$$

Barycentric Coordinates

❖ Coordinates on edge

$$\alpha = 1 - t$$

$$\beta = t$$

$$\gamma = 0$$

Barycentric Coordinates

❖ Coordinates on edge

$$\alpha = 0$$

$$\beta = 1 - t$$

$$\gamma = t$$

Barycentric Coordinates

❖ Coordinates on edge

$$\alpha = t$$

$$\beta = 0$$

$$\gamma = 1 - t$$

Barycentric Coordinates

❖ Inside / outside test

$$\alpha > 0, \beta > 0, \gamma > 0$$

Barycentric Coordinates

❖ Inside / outside test

$$\alpha < 0, \beta > 0, \gamma > 0$$

Barycentric Coordinates

❖ Inside / outside test

$$\alpha < 0, \beta > 0, \gamma > 0$$

Polynomial Interpolation

$$f(t) = at + b$$

Polynomial Interpolation

$$f(t) = at^2 + bt + c$$

Polynomial Interpolation

- ❖ Lagrange interpolation
- ❖ Newton interpolation
- ❖ Same polynomial
- ❖ Different cost of construction and evaluation

Least Squares Approximation

- ❖ Approximate noisy or overdetermined data

Least Squares Approximation

- ❖ Given m data points

$$(x_1, f_1), (x_2, f_2), \dots, (x_m, f_m)$$

- ❖ Given basis

$$\phi_1(x), \dots, \phi_n(x)$$

- ❖ Find coefficients $\alpha_1, \dots, \alpha_n$

$$\phi(x) = \alpha_1 \phi_1(x) + \dots + \alpha_n \phi_n(x)$$

Least Squares Approximation

$$(x_1, f_1), (x_2, f_2) \dots, (x_m, f_m)$$

$$\phi(x) = \alpha_1 \phi_1(x) + \dots + \alpha_n \phi_n(x)$$

- ❖ minimize sum of squared errors

$$\sum_{i=1}^m |\phi(x_i) - f_i|^2$$

Least Squares Approximation

$$\operatorname{argmin}_{\alpha} \|A\alpha - \mathbf{f}\|_2^2$$

❖ Normal equations

$$A^T A \alpha = A^T \mathbf{f}$$

Least Squares Approximation

- ❖ Weighted least squares

$$\sum_{i=1}^m w_i |\phi(x_i) - f_i|^2$$

- ❖ Regularized least squares

$$\operatorname{argmin}_{\alpha} \|A\alpha - \mathbf{f}\|_2^2 + \|\Gamma\alpha\|_2^2$$

Other Methods

- ❖ Bezier curves, splines
- ❖ Harmonic coordinates, mean value coordinates, Green coordinates

Finite Differences

Finite Difference Methods

- ❖ Used to discretize spatial derivatives

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

Finite Difference Methods

- ❖ Recall derivative of function

$$\frac{d}{dx} f(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

- ❖ Finite difference approximation

$$\frac{d}{dx} f(x) \approx \frac{f(x+h) - f(x)}{h}$$

forward difference

Finite Difference Methods

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

❖ Error decreases with decreasing h

❖ Taylor series

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \dots + \frac{h^n}{n!}f^{(n)}(x) + \dots$$

❖ Rearrange

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2}f''(x) + \dots$$

Finite Difference Methods

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- ❖ Error decreases with decreasing h
- ❖ Taylor series

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \dots + \frac{h^n}{n!}f^{(n)}(x) + \dots$$

- ❖ Rearrange

$$\frac{f(x+h) - f(x)}{h} = f'(x) + O(h)$$

first order
accurate

Forward and Backward Difference

- ❖ Forward difference

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- ❖ Backward difference

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Forward and Backward Difference

- ❖ Forward difference

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- ❖ Backward difference

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

Central Difference Approximation

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

❖ Taylor series

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + O(h^3),$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - O(h^3).$$

❖ Combine, rearrange

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + O(h^2)$$

second order accurate

Discretization Error

❖ Forward difference

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \dots$$

❖ Central difference

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6} f'''(x) + \dots$$

Higher Derivatives

$$f''(x) \approx \frac{f'(x + \frac{h}{2}) - f'(x - \frac{h}{2})}{h}$$

$$\approx \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h}$$

$$\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Higher Derivatives

$$f''(x) \approx \frac{f'(x + \frac{h}{2}) - f'(x - \frac{h}{2})}{h}$$

$$\approx \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h}$$

$$\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Higher Derivatives

$$\begin{aligned} f''(x) &\approx \frac{f'(x + \frac{h}{2}) - f'(x - \frac{h}{2})}{h} \\ &\approx \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h} \\ &\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \end{aligned}$$

3 - point stencil

Laplacian Operator

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

❖ In 2D $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$

$$\frac{\partial^2 u(x, y)}{\partial x^2} \approx \frac{u(x + h, y) - 2u(x, y) + u(x - h, y)}{h^2}$$

$$\frac{\partial^2 u(x, y)}{\partial y^2} \approx \frac{u(x, y + h) - 2u(x, y) + u(x, y - h)}{h^2}$$

Laplacian Operator

$$\Delta u \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}$$

5 - point stencil

Poisson Equation (1D)

$$u_{xx} = f, \quad x \in \Omega$$

$$u(x) = \bar{u}(x), \quad x \in \partial\Omega$$

❖ At each interior node $\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = f_i$

❖ Boundary nodes $u_0 = \bar{u}(x_0), u_7 = \bar{u}(x_7)$

Poisson Equation (1D)

$$u_{xx} = f, \quad x \in \Omega$$

$$u(x) = \bar{u}(x), \quad x \in \partial\Omega$$

linear
system

$$\frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{pmatrix} = \begin{pmatrix} f_1 - \frac{\bar{u}(x_0)}{h^2} \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 - \frac{\bar{u}(x_7)}{h^2} \end{pmatrix}$$

Finite Elements

Finite Elements

- ❖ Discretize the space of representable functions, instead of discretizing derivatives
- ❖ Method:
 - ❖ Discretize space into a finite set of elements
 - ❖ Choose a set of basis functions over the elements (e.g. piecewise linear)
 - ❖ *Galerkin* projection onto these basis functions
 - ❖ Solve the problem

Lets try an example:
Linear Finite Elements for Elasticity

Choose Element Type: Triangle

Choose Basis Functions: Piecewise Linear

Project Deformation Function onto the Piecewise Linear Function Space

Project Deformation Function onto the Piecewise Linear Function Space

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}(\mathbf{u}_0) + \mathbf{F} (\mathbf{u} - \mathbf{u}_0)$$

Solve the Problem

Solve the Problem

We know we can compute forces from **F**

Solve the Problem

We know we can compute forces from $\mathbf{F}$

But, how do we compute $\mathbf{F}$?

$$\left(\text{in } \mathbf{x}(\mathbf{u}) = \mathbf{x}(\mathbf{u}_0) + \mathbf{F} (\mathbf{u} - \mathbf{u}_0) \right)$$

$$\mathbf{u} = \mathbf{u}_0 + \alpha (\mathbf{u}_1 - \mathbf{u}_0) + \beta (\mathbf{u}_2 - \mathbf{u}_0)$$

$$\mathbf{u} = \mathbf{u}_0 + \alpha (\mathbf{u}_1 - \mathbf{u}_0) + \beta (\mathbf{u}_2 - \mathbf{u}_0)$$

$$\mathbf{u} = \mathbf{u}_0 + \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\mathbf{x} = \mathbf{x}_0 + \alpha (\mathbf{x}_1 - \mathbf{x}_0) + \beta (\mathbf{x}_2 - \mathbf{x}_0)$$

$$\mathbf{x} = \mathbf{x}_0 + \alpha (\mathbf{x}_1 - \mathbf{x}_0) + \beta (\mathbf{x}_2 - \mathbf{x}_0)$$

$$\mathbf{x} = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Rest / Material

$$\mathbf{u} = \mathbf{u}_0 + \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

World

$$\mathbf{x} = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Rest / Material

World

$$\mathbf{u} = \mathbf{u}_0 + \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\mathbf{x} = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix}^{-1} (\mathbf{u} - \mathbf{u}_0)$$

Rest / Material

World

$$\mathbf{u} = \mathbf{u}_0 + \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\mathbf{x} = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}_0 + \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix}^{-1} (\mathbf{u} - \mathbf{u}_0)$$

$$\mathbf{F} = \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix}^{-1}$$

Rest / Material

Canonical

World

$$\mathbf{F} = \begin{pmatrix} \mathbf{x}_{10} & \mathbf{x}_{20} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{10} & \mathbf{u}_{20} \end{pmatrix}^{-1}$$

Solve the Problem

$$\epsilon = \frac{1}{2} \left(\tilde{\mathbf{F}}^T + \tilde{\mathbf{F}} \right) - \mathbf{I}$$

where $\mathbf{F} = \mathbf{Q}\tilde{\mathbf{F}}$

Solve the Problem

$$\epsilon = \frac{1}{2} \left(\tilde{\mathbf{F}}^T + \tilde{\mathbf{F}} \right) - \mathbf{I}$$

$$\text{where } \mathbf{F} = \mathbf{Q}\tilde{\mathbf{F}}$$

$$\sigma = \lambda \text{Tr}(\epsilon) \mathbf{I} + 2\mu\epsilon$$

Solve the Problem

$$\epsilon = \frac{1}{2} \left(\tilde{\mathbf{F}}^T + \tilde{\mathbf{F}} \right) - \mathbf{I}$$

$$\text{where } \mathbf{F} = \mathbf{Q}\tilde{\mathbf{F}}$$

$$\sigma = \lambda \text{Tr}(\epsilon) \mathbf{I} + 2\mu\epsilon$$

$$\mathbf{f} = \mathbf{Q}\sigma\mathbf{n}_i$$

Solve the Problem

$$\epsilon = \frac{1}{2} \left(\tilde{\mathbf{F}}^T + \tilde{\mathbf{F}} \right) - \mathbf{I}$$

$$\text{where } \mathbf{F} = \mathbf{Q}\tilde{\mathbf{F}}$$

$$\sigma = \lambda \text{Tr}(\epsilon) \mathbf{I} + 2\mu\epsilon$$

$$\mathbf{f} = \mathbf{Q}\sigma\mathbf{n}_i$$

where $\mathbf{n}_i$
are the normals of the opposite faces
in rest space

IV. Temporal Integration

Explicit Integration

Explicit formula for $(t+1)$
in terms of quantities known at time t

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \Delta t \cdot \mathbf{v}(\mathbf{x}_p, t)$$

Note: everything on the right hand side
is evaluated at time t

Choose Your Integration Scheme Wisely

Trapezoidal Rule vs. Midpoint Method

Forward Euler

Trapezoidal Rule

Midpoint Method

Trapazoidal Rule

Average

Average

Midpoint Method

Trapezoidal Rule vs. Midpoint Method

- ❖ Both second-order Runge-Kutta methods (same accuracy)
- ❖ Very different behavior
 - ❖ Trapezoidal rule is smoother, more damped looking
 - ❖ Midpoint Method keeps more energy, but can be noisy / aliased

Position Updates for

$$\frac{d^2 \mathbf{x}_p(t)}{dt^2} = \frac{\mathbf{f}(\mathbf{x}_p, t)}{m_p}$$

Three Position Updates

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + dt \cdot \mathbf{v}_p(t)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \frac{dt}{2} \cdot (\mathbf{v}_p(t) + \mathbf{v}_p(t + \Delta t))$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + dt \cdot \mathbf{v}_p(t + \Delta t)$$

“Stiff” Problems

Consider the IVP:

$$\mathbf{x}(0) = \mathbf{x}_0$$

$$\mathbf{v}(0) = 0$$

$$\mathbf{f} = -k\mathbf{x}$$

“Stiff” Problems

Consider the IVP:

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{v}(0) = 0 \qquad \mathbf{f} = -k\mathbf{x}$$

After one time step:

$$\mathbf{x}(\Delta t) = \left(1 - \frac{\Delta t^2 k}{m}\right) \mathbf{x}_0$$

“Stiff” Problems

Consider the IVP:

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{v}(0) = 0 \qquad \mathbf{f} = -k\mathbf{x}$$

After one time step:

$$\mathbf{x}(\Delta t) = \left(1 - \frac{\Delta t^2 k}{m}\right) \mathbf{x}_0$$

$$\text{If } \Delta t > \sqrt{\frac{2m}{k}}$$

the spring will be more extended than when we started

If We Want to Take Bigger Timesteps

Implicit Integration

Replace:

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

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

With:

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

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

Implicit Integration

Replace:

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

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

With:

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

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

$$\mathbf{K}\mathbf{x} - \mathbf{K}\mathbf{x}_0 + \mathbf{D}\mathbf{v} + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

$$\mathbf{K}\mathbf{x} - \mathbf{K}\mathbf{x}_0 + \mathbf{D}\mathbf{v} + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

+

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

$$\mathbf{K}\mathbf{x} - \mathbf{K}\mathbf{x}_0 + \mathbf{D}\mathbf{v} + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

+

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

substitute into

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

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

$$\mathbf{K}\mathbf{x} - \mathbf{K}\mathbf{x}_0 + \mathbf{D}\mathbf{v} + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

+

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

substitute into

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

re-arrange

Applied to Soft Bodies

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \mathbf{D}(\mathbf{v}) + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

Linearize

$$\mathbf{K}\mathbf{x} - \mathbf{K}\mathbf{x}_0 + \mathbf{D}\mathbf{v} + \mathbf{M}\mathbf{a} = \mathbf{f}_{ext}$$

+

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t + \Delta t)$$

substitute into

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

re-arrange

$$(\mathbf{M} + \Delta t^2 \mathbf{K} + \Delta t \mathbf{D}) \mathbf{v}(t + \Delta t) = \mathbf{M}\mathbf{v}(t) + \Delta t (-\mathbf{K}(\mathbf{x}(t) - \mathbf{x}_0) + \mathbf{f}_{ext})$$

$$\left(\mathbf{M} + \Delta t^2 \mathbf{K} + \Delta t \mathbf{D}\right) \mathbf{v}(t + \Delta t) = \mathbf{M} \mathbf{v}(t) + \Delta t \left(-\mathbf{K} \left(\mathbf{x}(t) - \mathbf{x}_0\right) + \mathbf{f}_{ext}\right)$$

$$\underbrace{(\mathbf{M} + \Delta t^2 \mathbf{K} + \Delta t \mathbf{D})}_{\text{Linear System}} \mathbf{v}(t + \Delta t) = \mathbf{M} \mathbf{v}(t) + \Delta t (-\mathbf{K} (\mathbf{x}(t) - \mathbf{x}_0) + \mathbf{f}_{ext})$$

Linear System

Sparse, Symmetric

$$\underbrace{(\mathbf{M} + \Delta t^2 \mathbf{K} + \Delta t \mathbf{D})}_{\text{Linear System}} \mathbf{v}(t + \Delta t) = \underbrace{\mathbf{M} \mathbf{v}(t)}_{\text{Momentum}} + \Delta t (-\mathbf{K} (\mathbf{x}(t) - \mathbf{x}_0) + \mathbf{f}_{ext})$$

Linear System

Momentum

Sparse, Symmetric

Euler step of elastic
and external forces

$$\underbrace{(\mathbf{M} + \Delta t^2 \mathbf{K} + \Delta t \mathbf{D})}_{\text{Linear System}} \mathbf{v}(t + \Delta t) = \underbrace{\mathbf{M} \mathbf{v}(t)}_{\text{Momentum}} + \overbrace{\Delta t (-\mathbf{K} (\mathbf{x}(t) - \mathbf{x}_0) + \mathbf{f}_{ext})}^{\text{Euler step of elastic and external forces}}$$

Momentum

Linear System

Sparse, Symmetric

V. Constraints

Constraints

- ❖ Geometric relationships that must be satisfied

Constraints

- ❖ Constraint forces arises in response to other forces to maintain the constraint

Degrees of Freedom (DOF)

- ❖ Number of independent parameters describing configuration

$$3n$$

Degrees of Freedom (DOF)

- ❖ number of independent parameters describing configuration

$$3n - 1$$

$$6$$

Unilateral/Bilateral Constraints

$$g(\mathbf{x}, t) \geq 0$$

$$g(\mathbf{x}, t) = 0$$

Unilateral/Bilateral Constraints

$$\mathbf{g}(\mathbf{x}, t) \geq \mathbf{0}$$

$$y_1 - h \geq 0$$

$$\mathbf{g}(\mathbf{x}, t) = \mathbf{0}$$

$$(\mathbf{x}_1 + \mathbf{r}_1) - (\mathbf{x}_2 + \mathbf{r}_2) = \mathbf{0}$$

Soft vs. Hard Constraints

- ❖ **Soft constraint:**
force competes
with other forces
in the system

- ❖ **Hard constraint:**
force as strong as
necessary to
maintain constraint

Constraints: Solution Methods

- ❖ Maximal coordinates + auxiliary conditions
 - ❖ Include forces to maintain constraint
- ❖ Generalized coordinates (a.k.a., reduced coordinates, minimal coordinates)
 - ❖ Parameterize true DOF, respecting constraints

Penalty Methods

- ❖ Restoring force that acts to drive system to valid state

constraint: $(\mathbf{x}_1 + \mathbf{r}_1) - (\mathbf{x}_2 + \mathbf{r}_2) = \mathbf{0}$

penalty force: $\mathbf{f}_p = -k((\mathbf{x}_2 + \mathbf{r}_2) - (\mathbf{x}_1 + \mathbf{r}_1))$

$$m\mathbf{a} = \mathbf{f} + \mathbf{f}_p$$

Penalty Methods

- ✓ Simple to add to solver
 - Must tune parameters (k)
 - Introduce stiff forces $\rightarrow$ smaller time step or implicit integration
 - Constraint violation, oscillations

Lagrange Multiplier Methods

- ❖ Constraint force arises in response to other forces in the system
- ❖ Add unknowns to equations that represent strength of constraint force
- ❖ Add (differentiated) constraint equations

Constraint Implicit Surface

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

❖ Gradient

$$\nabla g(\mathbf{x}) = \begin{pmatrix} \frac{\partial g}{\partial x_1}(\mathbf{x}) \\ \frac{\partial g}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial g}{\partial x_n}(\mathbf{x}) \end{pmatrix}$$

Constraint Force

- ❖ Constraint force

$$\mathbf{F}_c$$

- ❖ Workless

$$\mathbf{F}_c \cdot \delta \mathbf{x} = 0$$

$$\Rightarrow \mathbf{F}_c = \lambda \nabla g(\mathbf{x})$$

Lagrange multiplier

Equations of Motion

❖ m constraints

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{0}$$

❖ Constraint force

$$\mathbf{F}_c = J^T \lambda = \begin{pmatrix} \nabla g_1 & \nabla g_2 & \dots & \nabla g_m \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix}$$

❖ Equations of motion

$$M\mathbf{a} = \mathbf{F} + \mathbf{F}_c = \mathbf{F} + J^T \lambda$$

How to find λ

- ❖ Constraint tells us valid positions

$$\mathbf{g}(\mathbf{x}) = \mathbf{0}$$

- ❖ Differentiate to get valid velocities

$$\dot{\mathbf{g}}(\mathbf{x}) = J(\mathbf{x})\mathbf{v} = \mathbf{0}$$

- ❖ Differentiate again to get valid accelerations

$$\ddot{\mathbf{g}}(\mathbf{x}) = \dot{J}\mathbf{v} + J\mathbf{a} = \mathbf{0}$$

How to find λ

❖ Equations of motion

$$M\mathbf{a} = \mathbf{F} + \mathbf{F}_c = \mathbf{F} + J^T \lambda$$

$$\mathbf{a} = M^{-1}(\mathbf{F} + J^T \lambda)$$

How to find λ

- ❖ Equations of motion

$$M\mathbf{a} = \mathbf{F} + \mathbf{F}_c = \mathbf{F} + J^T \lambda$$

$$\mathbf{a} = M^{-1}(\mathbf{F} + J^T \lambda)$$

- ❖ Plug into expression for
valid $\mathbf{a}$

$$\ddot{\mathbf{g}}(\mathbf{x}) = \dot{J}\mathbf{v} + J\mathbf{a} = \mathbf{0}$$

How to find λ

- ❖ Equations of motion

$$M\mathbf{a} = \mathbf{F} + \mathbf{F}_c = \mathbf{F} + J^T \lambda$$

$$\mathbf{a} = M^{-1}(\mathbf{F} + J^T \lambda)$$

- ❖ Plug into expression for valid $\mathbf{a}$

$$\ddot{\mathbf{g}}(\mathbf{x}) = \dot{J}\mathbf{v} + J\mathbf{a} = \mathbf{0}$$

- ❖ Rearrange

$$(JM^{-1}J^T)\lambda = -JM^{-1}\mathbf{F} - \dot{J}\mathbf{v}$$

How to find λ

❖ Rearrange

$$(JM^{-1}J^T)\lambda = \underbrace{-JM^{-1}\mathbf{F} - \dot{J}\mathbf{v}}_{\mathbf{b}}$$

❖ KKT System

$$\begin{pmatrix} M & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{b} \end{pmatrix}$$

Generalized Coordinates

- ❖ Instead of maximal coordinates $x_1, x_2, \dots, x_n$ along with auxiliary conditions and forces
- ❖ Generalized coordinates $q_1, q_2, \dots, q_N$ with $N < n$ that take constraints into account

Generalized Coordinates

- ❖ Example: rigid body

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$

+ constraints

$\mathbf{x}, R$

Transformation Equations

- ❖ Relate maximal and generalized coordinates

$$\begin{aligned}x_1 &= x_1(q_1, q_2, \dots, q_N) \\x_2 &= x_2(q_1, q_2, \dots, q_N) \\&\vdots \\x_n &= x_n(q_1, q_2, \dots, q_N)\end{aligned}\quad \mathbf{x} = \mathbf{x}(\mathbf{q})$$

- ❖ Jacobian

$$J(\mathbf{q}) = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}$$

Generalized Forces

x_1	f_1
$\vdots$	$\vdots$
x_n	f_n

q_1	g_1
$\vdots$	$\vdots$?
q_N	g_N

Generalized Forces

- ❖ Two sets of forces are dynamically equivalent if they do the same “virtual work”

$$\mathbf{f} \cdot \delta \mathbf{x} = \mathbf{g} \cdot \delta \mathbf{q}$$

$$\mathbf{f} \cdot J \delta \mathbf{q} = \mathbf{g} \cdot \delta \mathbf{q}$$

$$\mathbf{g} = J^T \mathbf{f}$$

Generalized Forces: Rigid Body

Jacobian

$$\dot{\mathbf{x}} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r} = \underbrace{\begin{pmatrix} I & \mathbf{r}^{*T} \end{pmatrix}}_J \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix}$$

generalized
force

$$\mathbf{g} = J^T \mathbf{f} = \begin{pmatrix} I \\ \mathbf{r}^* \end{pmatrix} \mathbf{f} = \begin{pmatrix} \mathbf{f} \\ \mathbf{r} \times \mathbf{f} \end{pmatrix}$$

Equations of Motion

- ❖ Lagrange equations of motion

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}} \right) = \mathbf{g} \qquad \left(\frac{d}{dt} (m\mathbf{v}) = \mathbf{f} \right)$$

$$T = \frac{1}{2} \dot{\mathbf{x}}^T M \dot{\mathbf{x}} = \frac{1}{2} (J \dot{\mathbf{q}})^T M J \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T J^T M J \dot{\mathbf{q}}$$

- ❖ Euler-Lagrange equations of motion, defining
Lagrangian $L = T - V$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{0}$$

Example: Pendulum

- ❖ Transformation equations

$$x(\theta) = l \sin \theta$$

$$y(\theta) = -l \cos \theta$$

- ❖ Velocity

$$\begin{pmatrix} \dot{x}(\theta) \\ \dot{y}(\theta) \end{pmatrix} = \begin{pmatrix} l \cos \theta \dot{\theta} \\ l \sin \theta \dot{\theta} \end{pmatrix} = J \dot{\theta}$$

- ❖ Lagrangian

$$L = \frac{1}{2} m l^2 \dot{\theta}^2 - m g l \cos \theta$$

- ❖ Equations of motion

$$\ddot{\theta} + \frac{g}{l} \sin \theta = 0$$

Constrained Rigid Body Systems

- ❖ Types of constraints
 - ❖ Articulation (joints)
 - ❖ Collisions
 - ❖ Resting and sliding contact

Impulse-Momentum Equations

- ❖ Acceleration-level formulation doesn't work well with discontinuous velocities, frictional contact
- ❖ Instead, integrate $\mathbf{f} = m\mathbf{a}$ to get **impulse-momentum** formulation

$$\int_{t_1}^{t_2} m\mathbf{a} \, dt = \int_{t_1}^{t_2} \mathbf{f} \, dt,$$
$$\Rightarrow m(\mathbf{v}(t_2) - \mathbf{v}(t_1)) = \mathbf{j}$$

Impulse-Momentum Equations

- ❖ Integrated, semi-discrete equations

$$M\mathbf{V}^{n+1} = M\mathbf{V}^n + \Delta t\mathbf{F} + J^T\lambda^{n+1}$$

- ❖ Combine with velocity-level constraint equation $J\mathbf{V} = 0$

$$\begin{pmatrix} M & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}^{n+1} \\ \mu^{n+1} \end{pmatrix} = \begin{pmatrix} M\mathbf{V}^n + \Delta t\mathbf{F} \\ \mathbf{0} \end{pmatrix}$$

$$\Rightarrow JM^{-1}J^T\mu^{n+1} = -J\mathbf{V}^n - \Delta tJM^{-1}\mathbf{F}$$

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

global
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T \mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T \mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

Impulse-Momentum Equations

- ❖ Instead of solving global, coupled system, common to split

update with non-
constraint forces

$$M\mathbf{V}^* = M\mathbf{V}^n + \Delta t\mathbf{F}$$

add constraint
impulses

$$M\mathbf{V}^{n+1} = M\mathbf{V}^* + J^T\mu^{n+1}$$

iterative
impulse
solve

repeat fixed
number of times or
until tolerance met

Global vs. Iterative Solve

❖ Global

- ❖ need to solve larger linear system
- ❖ LCP for inequality constraints

❖ Iterative

- ❖ may be slow to converge
- ❖ simple to do inequality constraints

Handling Drift With Stabilization

- ❖ Approach was based on velocity-level constraints lead to drift in positions

$$\begin{pmatrix} M & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}^{n+1} \\ \mu^{n+1} \end{pmatrix} = \begin{pmatrix} M\mathbf{V}^n + \Delta t \mathbf{F} \\ \mathbf{0} \end{pmatrix}$$

- ❖ Correct drift with stabilization
 - ❖ E.g., Baumgarte stabilization $\dot{\mathbf{g}}(\mathbf{x}) + \gamma \mathbf{g}(\mathbf{x}) = 0$
 - ❖ Modify positions and velocities so they satisfy constraints

Softening Constraints

- ❖ Also common to soften constraints

$$\begin{pmatrix} M & -J^T \\ -J & \gamma I \end{pmatrix} \begin{pmatrix} \mathbf{V}^{n+1} \\ \mu^{n+1} \end{pmatrix} = \begin{pmatrix} M\mathbf{V}^n + \Delta t \mathbf{F} \\ -\frac{\beta}{\Delta t} \mathbf{g}(\mathbf{X}^n) \end{pmatrix}$$

- ❖ Stabilizes constraints
- ❖ Regularizes system
 - ❖ Better numerical properties
 - ❖ Handle redundant constraints
- ❖ Adds some compliance to constraint

Collisions and Contact

Collision Detection

- ❖ Polygonal geometry

non-convex

convex

- ❖ Separating axis theorem

- ❖ Convex decomposition

Collision Detection

- ❖ Signed distance field

$$\phi(\mathbf{x})$$

- ❖ Zero level set

$$\phi(\mathbf{x}) = 0$$

- ❖ Fast inside/outside tests

- ❖ Penetration depth $\phi(\mathbf{x})$

- ❖ Normals $\nabla\phi(\mathbf{x})$

Accelerating Collision Detection

- ❖ Bounding volumes

- ❖ Hierarchical bounding volumes

- ❖ Spatial partitions

Discrete vs. Continuous Collision Detection

Collision Response: Inelastic

- ❖ Newton's third law (action / reaction)

$$m_1 v'_1 = m_1 v_1 + j$$

$$m_2 v'_2 = m_2 v_2 - j$$

- ❖ Assume inelastic (sticking)

$$v'_1 = v'_2$$

- ❖ Solve: $j = \left(\frac{1}{m_1} + \frac{1}{m_2} \right)^{-1} (v_2 - v_1)$

$$v'_1 = v'_2 = \left(\frac{m_1}{m_1 + m_2} \right) v_1 + \left(\frac{m_2}{m_1 + m_2} \right) v_2$$

Collision Response:Elastic

- ❖ Newton's third law (action / reaction)

$$m_1 v'_1 = m_1 v_1 + j$$

$$m_2 v'_2 = m_2 v_2 - j$$

- ❖ Assume elastic (bouncing)

$$(v'_2 - v'_1) = -\epsilon(v_2 - v_1) \quad \text{coefficient of restitution}$$

- ❖ Solve: $j = \left(\frac{1}{m_1} + \frac{1}{m_2} \right)^{-1} (1 + \epsilon)(v_2 - v_1)$

$$v'_1 = v_1 + \frac{1}{m_1} j, \quad v'_2 = v_2 - \frac{1}{m_2} j$$

Deformable Object Collisions

- ❖ Body compresses, stores energy and bounces
- ❖ Inelastic collision between particles and ground

$$m_1 \mathbf{v}'_1 = m_1 \mathbf{v}_1 + \mathbf{j}$$

$$m_2 \mathbf{v}'_2 = m_2 \mathbf{v}_2 - \mathbf{j}$$

$$\mathbf{v}'_2 = \mathbf{v}'_1$$

Increment [1]:

Deformable Object Collisions

- ❖ Body compresses, stores energy and bounces
- ❖ Inelastic collision between particles and ground

$$m_1 \mathbf{v}'_1 = m_1 \mathbf{v}_1 + \mathbf{j}$$

$$m_2 \mathbf{v}'_2 = m_2 \mathbf{v}_2 - \mathbf{j}$$

$$\mathbf{v}'_2 = \mathbf{v}'_1$$

Increment [1]:

Rigid Body Collisions

- ❖ Physically, similar to deformable body collisions
- ❖ But rigid idealization precludes storing energy
- ❖ Instead, algebraic collision laws for before / after collision
- ❖ Cases:
 - ❖ Inelastic (sticking)
 - ❖ Elastic: frictionless, with friction

Rigid Body Inelastic Collision

❖ Linear momentum

$$m_1 \mathbf{v}'_{C1} = m_1 \mathbf{v}_{C1} + \mathbf{j}$$

$$m_2 \mathbf{v}'_{C2} = m_2 \mathbf{v}_{C2} - \mathbf{j}$$

❖ Angular momentum

$$I_1 \omega'_1 = I_1 \omega_1 + \mathbf{r}_1 \times \mathbf{j}$$

$$I_2 \omega'_2 = I_2 \omega_1 - \mathbf{r}_2 \times \mathbf{j}$$

❖ Sticking

$$\mathbf{v}'_{P1} = \mathbf{v}'_{P2}$$

Rigid Body Frictionless Collision

❖ Linear momentum

$$m_1 \mathbf{v}'_{C1} = m_1 \mathbf{v}_{C1} + \mathbf{j}$$

$$m_2 \mathbf{v}'_{C2} = m_2 \mathbf{v}_{C2} - \mathbf{j}$$

❖ Angular momentum

$$I_1 \omega'_1 = I_1 \omega_1 + \mathbf{r}_1 \times \mathbf{j}$$

$$I_2 \omega'_2 = I_2 \omega_1 - \mathbf{r}_2 \times \mathbf{j}$$

❖ Elastic

$$\mathbf{j} = j \mathbf{n}$$

$$(\mathbf{v}'_{P2} - \mathbf{v}'_{P1}) \cdot \mathbf{n} = -\epsilon (\mathbf{v}_{P2} - \mathbf{v}_{P1}) \cdot \mathbf{n}$$

Rigid Body Frictional Collision

- ❖ Coulomb friction model

μ coefficient of friction

- ❖ Static $\|\mathbf{f}_t\| \leq \mu \|\mathbf{f}_n\|$

- ❖ Sliding $\mathbf{f}_t = -\mu \|\mathbf{f}_n\| \mathbf{t}$

Rigid Body Frictional Collision

- ❖ Example [Guendelman et al. 2003]

- ❖ Elastic in normal direction

$$(\mathbf{v}'_{P2} - \mathbf{v}'_{P1}) \cdot \mathbf{n} = -\epsilon(\mathbf{v}_{P2} - \mathbf{v}_{P1}) \cdot \mathbf{n}$$

- ❖ Assume sticking (+2 equations)

$$(I - \mathbf{n}\mathbf{n}^T)(\mathbf{v}'_{P2} - \mathbf{v}'_{P1}) = \mathbf{0}$$

- ❖ If not admissible, assume sliding (-2 unknowns)

$$\mathbf{j} = j_n \mathbf{n} - \mu j_n \mathbf{t}$$

Thanks!

For notes, slides, and source code visit:

<https://cal.cs.umbc.edu/Courses/PhysicsBasedAnimation>

adamb@umbc.edu

shinar@cs.ucr.edu