Rigid Body Collisions

treuille@cs.cmu.edu
Example
Example

These simulations could never have been created by hand.
Conservation and Forces

**Linear Momentum**

\[
\frac{d}{dt} \sum_i m_i \dot{x}_i = \sum_i f_i \\
\sum_i m_i \ddot{x}_i = F \\
(\ddot{x} = \frac{1}{M} \sum_i m_i x_i) \\
(M \ddot{x} = \sum_i m_i \ddot{x}_i) \\
M \ddot{x} = F
\]

**Angular Momentum**

\[
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{\mathbf{x}}_i = \sum_i \mathbf{r}_i \times f_i \\
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{\mathbf{x}}_i = \tau \\
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \mathbf{\omega} \times \mathbf{r}_i = \tau \\
\frac{d}{dt} \sum_i m_i \mathbf{r}_i^* \mathbf{r}_i^* \mathbf{\omega} = \tau
\]
Conservation and Forces

**Linear Momentum**

\[ \frac{d}{dt} \sum_i m_i \dot{x}_i = \sum_i f_i \]

\[ \sum_i m_i \ddot{x}_i = F \]

\[ \ddot{x} = \frac{1}{M} \sum_i m_i x_i \]

\[ M \dddot{x} = \sum_i m_i \dddot{x}_i \]

\[ M \dddot{x} = F \]

**Angular Momentum**

\[ \frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{x}_i = \sum_i \mathbf{r}_i \times f_i \]

\[ \frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{x}_i = \tau \]

\[ \frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \omega \times \mathbf{r}_i = \tau \]

\[ \frac{d}{dt} \sum_i m_i \mathbf{r}_i^* \mathbf{r}_i^* \omega = \tau \]

\[ \frac{d}{dt} I \omega = \tau \]
Conservation and Forces

**Linear Momentum**

\[
\frac{d}{dt} \sum_i m_i \dot{x}_i = \sum_i f_i
\]

\[
\sum_i m_i \ddot{x}_i = F
\]

\[
\ddot{x} = \frac{1}{M} \sum_i m_i x_i
\]

\[
M \ddot{x} = \sum_i m_i \ddot{x}_i
\]

**Angular Momentum**

\[
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{x}_i = \sum_i \mathbf{r}_i \times f_i
\]

\[
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \dot{x}_i = \tau
\]

\[
\frac{d}{dt} \sum_i m_i \mathbf{r}_i \times \mathbf{\omega} \times \mathbf{r}_i = \tau
\]

\[
\frac{d}{dt} \sum_i m_i \mathbf{r}_i^* \mathbf{r}_i^* \mathbf{\omega} = \tau
\]

\[
\frac{d}{dt} I \omega = \tau
\]
Rigid Body Equation of Motion

\[
\frac{d}{dt} \mathbf{X}(t) = \frac{d}{dt} \begin{pmatrix} x(t) \\ \mathbf{R}(t) \\ M \mathbf{v}(t) \\ \mathbf{I}(t) \mathbf{\omega}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{\omega}(t)^* \mathbf{R}(t) \\ F(t) \\ \tau(t) \end{pmatrix}
\]

\( P(t) \) – linear momentum

\( L(t) \) – angular momentum
Discrete Inertia

\[ I = \sum_i m_i r_i^* r_i^* \]

\[ I = \sum_i \begin{pmatrix} m_i \begin{bmatrix} -y^2 - z^2 & xy & xz \\ xy & -x^2 - z^2 & yz \\ xz & yz & -x^2 - y^2 \end{bmatrix} \end{pmatrix} \]
Continuous Inertia

\[ I(t) = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \]

**diagonal terms**

\[ I_{xx} = M \int_V (y^2 + z^2) \, dV \]

**off-diagonal terms**

\[ I_{xy} = -M \int_V xy \, dV \]

Tuesday, November 29, 11
Inertia Tensors Vary in World Space...

\[ I_{xx} = M \int_{V} (y^2 + z^2) \, dV \quad I_{xy} = -M \int_{V} xy \, dV \]
... but are Constant in Body Space

\[ I(t) = R(t)I_{\text{body}}R(t)^T \]
Approximating $I_{\text{body}}$: Bounding Boxes

Pros: Simple.
Cons: Bounding box may not be a good fit. Inaccurate.
Approximating $I_{body}$: Point Sampling

**Pros:** Simple, fairly accurate, no B-rep needed.

**Cons:** Expensive, requires volume test.
Computing $I_{\text{body}}$: Green’s Theorem (2x!)

Pros: Simple, exact, no volumes needed.
Cons: Requires boundary representation.
Code: [http://www.acm.org/jgt/papers/Mirtich96](http://www.acm.org/jgt/papers/Mirtich96)
Rigid Body Dynamics

Collision and Contact

David Baraff
Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces
Outline

• Detect Collisions
  • Compute Collision Type
  • Depending on Collision Type...
    • Apply Impulse Force
    • Compute Resting Contact Forces
Problem

- Positions **NOT OK**
Collision Detection

Assume we have some spatial collision detection algorithm. (This can be solved in less than $O(n^2)$ time.)
Simulations with Collisions

We want rigid bodies to behave as solid objects, and not interpenetrate. By applying constraint forces between contacting bodies, we prevent interpenetration from occurring. We need to:

a) Detect interpenetration
b) Determine contact points
c) Compute constraint forces
We want rigid bodies to behave as solid objects, and not interpenetrate. By applying constraint forces between contacting bodies, we prevent interpenetration from occurring. We need to:

- a) Detect interpenetration
- b) Determine contact points
- c) Compute constraint forces

Simulations with Collisions

\[ Y(t_0 + \Delta t) \]

\[ Y(t_0) \]
Simulations with Collisions

\[ Y(t_0) \]

\[ Y(t_0 + \Delta t) \]

\[ Y(t_0 + 2\Delta t) \]

An Illegal State

Back up to the Collision Time

Colliding Contact

\[ \mathbf{n} \cdot \mathbf{p}_a < 0 \]
Simulations with Collisions

An Illegal State $Y$

$Y(t_0 + 2\Delta t)$

$Y(t_0 + \Delta t)$

$Y(t_0)$

$Y(t_0 + 3\Delta t)$

illegal state

Tuesday, November 29, 11
Backing up to the Collision Time

$Y(t_0)$

$Y(t_0) + \Delta t$

$Y(t_0 + 2\Delta t)$

$Y(t_0 + 3\Delta t)$

illegal state

Colliding Contact

$p_a \cdot n < 0$

$p_a \cdot n < 0$
Outline

- Detect Collisions
- **Compute Collision Type**
  - Depending on Collision Type...
    - Apply Impulse Force
    - Compute Resting Contact Forces
Geometric Contact

- Vertex-Face
- Edge-Edge

Figure 3. (a) Vertex-plane contact (side view). (b) Edge-edge contact. (c) Contact geometry.

Physical Contact

- Impulse Collision ("bounce")
- Resting Contact
Physical Contact

- Impulse Collision ("bounce")
- Resting Contact
Physical Contact

\[ p_a(t) = \text{contact point on body A} \]
\[ p_b(t) = \text{contact point on body B} \]

\[ p_a(t_0) = p_b(t_0) \text{ but in general } \dot{p}_a(t_0) \neq \dot{p}_b(t_0) \]
Physical Contact
(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n < 0

Impulse collision.
Physical Contact

\[(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n < 0\]

Impulse collision.

\[(\dot{p}_a(t_0) - \dot{p}_b(t_0)) \cdot n = 0\]

Resting contact.
Physical Contact

\[(p_a(t_0) - p_b(t_0)) \cdot n < 0\]

Impulse collision.

\[(p_a(t_0) - p_b(t_0)) \cdot n = 0\]

Resting contact.

\[(p_a(t_0) - p_b(t_0)) \cdot n > 0\]

No collision.
Outline

• Detect Collisions
• Compute Collision Type

• Depending on Collision Type...
  • Apply Impulse Force
  • Compute Resting Contact Forces
Problem

- Positions OK
- Velocities NOT OK
Colliding Contact

\[ \hat{n} \cdot \dot{p}_a < 0 \]
Collision Process

Δt

no force

no force
A Soft Collision

force

velocity

\[ \Delta t \]
A Harder Collision

force

velocity

$\Delta t$
A Very Hard Collision

force

velocity

Δt

Tuesday, November 29, 11
A Rigid Body Collision

impulsive force

\[ f_{imp} = \infty \]

\[ \Delta t = 0 \]
Δv remains constant!
Colliding Contact

Impulse
(alters velocity)
Mathematically...
Computing Impulses

\[ \dot{p}_a^- \]

\[ \dot{p}_a^+ = ? \]

\[ \hat{n} \]

\[ B \]

\[ j\hat{n} \]

\[ n \cdot p_a \]

\[ \epsilon = 1 \]

\[ \epsilon = \frac{1}{2} \]

\[ \epsilon = 0 \]
Coefficient of Restitution

\[ \hat{n} \cdot \dot{p}_a^+ = -\varepsilon (\hat{n} \cdot \dot{p}_a^-) \]

\( \varepsilon = 1 \)
\( \varepsilon = \frac{1}{2} \)
\( \varepsilon = 0 \)
Computing $j$

\[
v_a^+(t_0) = v_a^-(t_0) + \frac{j\hat{n}(t_0)}{M_a}
\]

\[
\omega_a^+(t_0) = \omega_a^-(t_0) + I_a^{-1} \left( r_a \times j\hat{n}(t_0) \right)
\]

\[
\dot{p}_a^+(t_0) = v_a^+(t_0) + \omega_a^+(t_0) \times r_a
\]

\[
\downarrow
\]

\[
\dot{p}_a^+(t_0) = aj + b
\]
Computing $j$

$$\hat{n} \cdot \dot{p}_a^+ = -\epsilon (\hat{n} \cdot \dot{p}_a^-)$$

$cj + b = d$
Computing $j$

\[ \hat{n} \cdot (\dot{p}_a^+ - \dot{p}_b^+) = -\varepsilon (\hat{n} \cdot (\dot{p}_a^- - \dot{p}_b^-)) \]
Computing \( j \)

\[
\hat{n} \cdot \left( \dot{p}_a^+ - \dot{p}_b^+ \right) = -\epsilon \left( \hat{n} \cdot \left( \dot{p}_a^- - \dot{p}_b^- \right) \right)
\]

\[ cj + b = d \]
Outline

- Detect Collisions
- Compute Collision Type
- Depending on Collision Type...
  - Apply Impulse Force
  - Compute Resting Contact Forces
Colliding Contact

\[ \hat{n} \cdot \dot{p}_a = 0 \]
Problem

- Positions OK
- Velocities OK
- Accelerations NOT OK
Resting Contact

external forces

\( \dot{P}_a \)

force

(alters acceleration)
Example
Example
Resting Contact Forces

To avoid inter-penetration, the force strength \( f_n \) must prevent the vertex \( p_a \) from accelerating downwards. If \( B \) is fixed, this is written as:

\[
\hat{n} \cdot p_a \geq 0
\]

To prevent the constraint force from holding bodies together, the force must be repulsive:

\[
f \geq 0 \quad \text{and} \quad \hat{n} \cdot p_a \geq 0
\]

\[
f \hat{n} + b \geq 0
\]
Resting Contact Forces

To avoid inter-penetration, the force strength $f_n$ must prevent the vertex $p_a$ from accelerating downwards. If $B$ is fixed, this is written as:

$$n \cdot p_a \geq 0$$

To prevent the constraint force from holding bodies together, the force must be repulsive:

$$f \geq 0$$

$$n \cdot p_a \geq 0$$

$$af + b \geq 0$$

What is $f$?
Solution Outline
Solution Outline

- Similar to constraints before, we will compute constraint forces.
Solution Outline

• Similar to constraints before, we will compute constraint forces.

• Except...
Solution Outline

- Similar to constraints before, we will compute constraint forces.
- Except...
  - There will be inequalities.
Solution Outline

- Similar to constraints before, we will compute constraint forces.
- Except...
  - There will be inequalities.
  - There will be quadratic terms.
Conditions on the Constraint Force

To avoid inter-penetration, the force strength $f$ must prevent the vertex $p_a$ from accelerating downwards. If $B$ is fixed, this is written as

$$\hat{n} \cdot \ddot{p}_a \geq 0$$
To avoid inter-penetration, the force strength $f$ must prevent the vertex $p_a$ from accelerating downwards. If $B$ is fixed, this is written as

$$\hat{n} \cdot \ddot{p}_a \geq 0$$

To prevent the constraint force from holding bodies together, the force must be repulsive:

$$af + b \geq 0$$

Does the above, along with sufficiently constrain $f$?
Conditions on the Constraint Force

To prevent the constraint force from holding bodies together, the force must be repulsive:

\[ f \geq 0 \]

Does the above, along with

\[ \hat{n} \cdot \dot{p}_a \geq 0 \quad \Rightarrow \quad af + b \geq 0 \]

sufficiently constrain \( f \)?
3rd Constraint

- We require that the force at a contact point become zero if the bodies begin to separate.
Workless Constraint Force

To make \( f \) be workless, we use the condition:

\[
af + b = 0
\]

or

\[
af + b > 0, \quad f = 0
\]
Conditions on the Constraint Force

To make \( f \) be workless, we use the condition

\[
f \cdot (af + b) = 0
\]

The full set of conditions is

\[
af + b \geq 0
\]

\[
f \geq 0
\]

\[
f \cdot (af + b) = 0
\]
Multiple Contact Points

To make $f$ be workless, we use the condition

$$f \cdot (af + b) = 0$$

The full set of conditions is

$$af + b \geq 0$$

$$f \geq 0$$

Conditions on $f_1$:

$$f_1 \cdot (a_{11}f_1 + a_{12}f_2 + b_1) = 0$$

$$f_1 \geq 0$$

Conditions on $f_2$:

$$f_2 \cdot (a_{11}f_1 + a_{12}f_2 + b_1) = 0$$

$$f_2 \geq 0$$

Non-penetration:

$$a_{11}f_1 + a_{12}f_2 + b_1 \geq 0$$

Repulsive:

$$a_{11}f_1 + a_{12}f_2 + b_1 > 0$$
**Conditions on $f_1$**

**Non-penetration:**

$$a_{11} f_1 + a_{12} f_2 + b_1 \geq 0$$

**Repulsive:**

$$f_1 \geq 0$$

**Workless:**

$$f_1 \cdot (a_{11} f_1 + a_{12} f_2 + b_1) = 0$$
Quadratic Program for $f_1$ and $f_2$

Non-penetration:
\[
\begin{align*}
    a_{11}f_1 + a_{12}f_2 + b_1 &\geq 0 \\
    a_{21}f_1 + a_{22}f_2 + b_2 &\geq 0
\end{align*}
\]

Repulsive:
\[
\begin{align*}
    f_1 &\geq 0 \\
    f_2 &\geq 0
\end{align*}
\]

Workless:
\[
\begin{align*}
    f_1 \cdot (a_{11}f_1 + a_{12}f_2 + b_1) &= 0 \\
    f_2 \cdot (a_{21}f_1 + a_{22}f_2 + b_2) &= 0
\end{align*}
\]
In the Notes – Constraint Forces

Derivations of the non-penetration constraints for contacting polyhedra.

Derivations and code for computing the $a_{ij}$ and $b_i$ coefficients.

Code for computing and applying the constraint forces $f_i \hat{n}_i$. 
Example
Question

• What type of discrete geometric representation should we use for a deformable object?

• What sort of forces apply to deformable objects, i.e. in what ways do they resist deformation?

• How can we compute these forces?