ASSIGNMENT 5: BASIC COLLISION RESPONSE

22/02/2016
ASSIGNMENT 5: COLLISION RESPONSE

Implement collision response system for a rigid body.

- Basic requirement requires correct simulation of a convex object falling onto an infinite plane

Minimum requirements:
- Minimum scene: convex object 3D object e.g. cube colliding against infinite plane
- Rigid Body state (from previous)
- Unconstrained State update (from previous)
- Detect a point of collision and collision normal (from previous)
- Impulse magnitude calculation
- Application of impulsive force and torque

Additional Features + Options:
- Object-object collisions
- General objects i.e. beyond a cube
- c.o.m. and Inertia tensor computation
- Backtracking, adaptive timesteps, springs or some corrective measure
- Any additional features/customizations
- Cool demos

Required Reading: [Baraff] or [Hecker]
NOTES ON COMPLETING THIS ASSIGNMENT
BASIC STEPS (OVERVIEW)

1. Make sure you read the following or equivalent*:
   - [Baraff] Colliding Contact OR
   - [Hecker] Rigid Body Dynamics

2. Get a collision point, $p$ and normal, $\hat{n}$
   - This is input from narrow-phase

3. Calculate the Impulse Magnitude, $j$
   - You will need to know the inertial tensor (see Lecture 2)

4. Apply an Impulse, $J = j\hat{n}$, to update the angular and linear momenta of each colliding rigid body

*: Texts by Eberley, Millington, Ericson or Erleben et al will also cover this material but may take a slightly different approach
COLLISION RESPONSE

Calculate instantaneous change in rigid body state (linear and angular momenta)

Based on impulse $J$

We require:

- Point of contact, $p$
- Direction (contact normal), $\hat{n}$
- Magnitude of impulse, $j$

$p$ and $\hat{n}$ are obtained as part of contact modelling. $j$ is calculated based on a number of variables ...
**IMPULSE MAGNITUDE EQUATION**

N.B. YOU WILL NEED TO IMPLEMENT THIS

\[
\begin{align*}
j &= \frac{-(1 + \epsilon) \, v_{rel}^-}{m_A^{-1} + m_B^{-1} + \hat{n} \left( I_A^{-1} (r_A \times \hat{n}) \right) \times r_A + \hat{n} \left( I_B^{-1} (r_B \times \hat{n}) \right) \times r_B} \\
I_A^{-1} &= R_A R_{bodyA}^T R_A^T \quad \text{and} \quad I_B^{-1} = R_B R_{bodyB}^T R_B^T \\
r_A &= p_A - x_A \quad \text{and} \quad r_B = p_B - x_B \\
v_{rel}^- &= \hat{n} (\dot{p}_A^- - \dot{p}_B^-) \\
\dot{p}_A &= v_A + \omega_A \times (p_A - x_A) \quad \text{and} \quad \dot{p}_B = v_B + \omega_B \times (p_B - x_B)
\end{align*}
\]

- \( j \): impulse magnitude
- \( \epsilon \): coefficient of restitution
- \( m_A, m_B \): mass of object A and B
- \( R_A, R_B \): orientation of A and B
- \( \omega_A, \omega_B \): angular velocity of A and B
- \( x_A, x_B \): centre of mass position of A and B
- \( v_A, v_B \): velocity of A and B
- \( v_{rel}^- \): pre-collision relative velocity of contact points
- \( I_A, I_B \): world space inertial tensor
- \( I_{bodyA}, I_{bodyB} \): object space inertial tensor
- \( \hat{n} \): contact plane normal
- \( p_A, p_B \): contact point on A and B
**IMPULSE MAGNITUDE EQUATION**

**WHERE THE VARIABLES COME FROM**

\[
j = \frac{-(1 + \epsilon) \mathbf{v}_{rel}}{m_A^{-1} + m_B^{-1} + \hat{n}(I_A^{-1}(r_A \times \hat{n})) \times r_A + \hat{n}(I_B^{-1}(r_B \times \hat{n})) \times r_B}
\]

- \( I_A^{-1} = R_A I_{bodyA} R_A^T \) and \( I_B^{-1} = R_B I_{bodyB} R_B^T \)

\[
\mathbf{r}_A = \mathbf{p}_A - \mathbf{x}_A \quad \text{and} \quad \mathbf{r}_B = \mathbf{p}_B - \mathbf{x}_B
\]

\[
\mathbf{v}_{rel} = \hat{n} (\mathbf{p}_A^\top - \mathbf{p}_B^\top)
\]

\[
\mathbf{\dot{p}}_A = \mathbf{v}_A + \mathbf{\omega}_A \times (\mathbf{p}_A - \mathbf{x}_A) \quad \text{and} \quad \mathbf{\dot{p}}_B = \mathbf{v}_B + \mathbf{\omega}_B \times (\mathbf{p}_B - \mathbf{x}_B)
\]

**Constants**

**Intermediate Variables**

**Rigid Body State**

**Contact Model**

\( j \): impulse magnitude

\( \epsilon \): coefficient of restitution

\( m_A, m_B \): mass of object A and B

\( R_A, R_B \): orientation of A and B

\( \mathbf{\omega}_A, \mathbf{\omega}_B \): angular velocity of A and B

\( \mathbf{x}_A, \mathbf{x}_B \): centre of mass position of A and B

\( \mathbf{v}_A, \mathbf{v}_B \): velocity of A and B

\( \mathbf{v}_{rel} \): pre-collision relative velocity of contact points

\( \mathbf{I}_A, \mathbf{I}_B \): world space inertial tensor

\( \mathbf{I}_{bodyA}, \mathbf{I}_{bodyB} \): object space inertial tensor

\( \mathbf{n} \): contact plane normal

\( \mathbf{p}_A, \mathbf{p}_B \): contact point on A and B
COLLISION RESPONSE: INPUTS AND OUTPUTS

**SCENE DESCRIPTION**
- Assignment 2
  - Constants: $I_{body}$, $m$, $\epsilon$

**SIMULATION (ODE)**
- Assignment 2
  - State Variables: $x$, $v$, $R$, $\omega$

**COLLISION DETECTION**
- Assignment 4
  - Contact Model: $p$, $\hat{n}$

**COMPUTE IMPULSE**
  - Impulse magnitude: $j$

**APPLY IMPULSE**
- Assignment 2
  - Updated State: $v^+$, $\omega^+$

Intermediate vars:
- $r$, $I_{world}$, $\dot{p}$
IMPLEMENTATION

For ease of coding, Baraff et al break down the impulse equation as follows:

\[ j = \frac{N}{t_1 + t_2 + t_3 + t_4} \]

where

- \( N = -(1 + \epsilon) v_{rel} \)
- \( t_1 = \frac{1}{m_A} \)
- \( t_2 = \frac{1}{m_B} \)
- \( t_3 = \hat{n} \left( (I_A^{-1}(r_A \times \hat{n})) \times r_A \right) \)
- \( t_4 = \hat{n} \left( (I_B^{-1}(r_B \times \hat{n})) \times r_B \right) \)

```c
void collision(Contact *c, double epsilon)
{
    triple padot = pt_velocity(c->a, c->p); /* \dot{p}_a^{c}(t) */
    pbdot = pt_velocity(c->b, c->p); /* \dot{p}_b^{c}(t) */
    n = c->n,
        /* \dot{n}(t) */
    ra = p - c->a->x,
        /* \dot{r}_a */
    rb = p - c->b->x;
        /* \dot{r}_b */

    double vrel = n * (padot - pbdot), /* v_{rel} */
    numerator = -(1 + epsilon) * vrel;

    /* We'll calculate the denominator in four parts */
    double term1 = 1 / c->a->mass,
        term2 = 1 / c->b->mass,
        term3 = n * ((c->a->Iinv * (ra ^ n)) ^ ra),
        term4 = n * ((c->b->Iinv * (rb ^ n)) ^ rb);

    /* Compute the impulse magnitude */
    double j = numerator / (term1 + term2 + term3 + term4);
    triple force = j * n;

    /* Apply the impulse to the bodies */
    c->a->P += force;
    c->b->P -= force;
    c->a->L += ra ^ force;
    c->b->L -= rb ^ force;

    /* recompute auxiliary variables */
    c->a->v = c->a->P / c->a->mass;
    c->b->v = c->b->P / c->b->mass;
    c->a->omega = c->a->Iinv * c->a->L;
    c->b->omega = c->b->Iinv * c->b->L;
}
```
For efficiency, we typically store:

- $\frac{1}{m}$ (inverse mass) instead of $m$
- and $I^{-1}_{body}$ (inverse of inertia tensor in body space) instead of $I_{body}$

For immovable objects (e.g. ground plane) use

- Mass: $\frac{1}{m} = 0$ i.e. infinite mass
- Inertial Tensor: $I^{-1}_{body} = 0$ i.e. infinite moment of inertia

- $t_2$ and $t_4$ are 0 in this case
### APPLYING IMPULSE

**Impulse magnitude is given by:**

\[
j = \frac{-(1 + \epsilon) \, v_{rel}}{m_A^{-1} + m_B^{-1} + \hat{n}(I_A^{-1}(r_A \times \hat{n})) \times r_A + \hat{n}(I_B^{-1}(r_B \times \hat{n})) \times r_B}
\]

**The actual impulse vector is simply** \( J = j\hat{n} \)

**This is applied to the objects as follows:**

- **Change in Linear momentum is directly equal to the impulse:**
  \[
  \Delta P = J \iff \Delta v = Jm^{-1}
  \]

- **Change in Angular momentum is equal to the imulsive torque (\( \tau_{\text{IMPULSE}} \)):**
  \[
  \Delta L = (r \times J) \iff \Delta \omega = I^{-1}(r \times J)
  \]
REFERENCES

REQUIRED READING (either one of):

[Hecker] Chris Hecker has a few Gamasutra Articles on the topic that are relatively easy reading:

- In particular, “Physics Part 4 – The 3rd Dimension”
- Available http://chrishecker.com/Rigid_Body_Dynamics


- Rigid Body Dynamics Notes - Collision and Contact

ALSO SEE:
