Collision Thoughts

December 25, 2017

We have two objects, A and B which collide. We assume the two objects are uniform density, rigid bodies, and convex polygons.

1 Two objects collide in 1D space

$$\begin{aligned} p_A^0 &= m_A v_A^0 && \text{(Momentum of object A)} \\ p_B^0 &= m_B v_B^0 && \text{(Momentum of object B)} \\ c_r &= \frac{v_B^1 - v_B^0}{v_A^1 - v_A^0} && \text{(Coefficient of restitution).} \end{aligned}$$

The coefficient of restitution, $c_r \in [0, 1]$, gives a measure of elasticity of the collision. $c_r = 0$ denotes a completely inelastic collision (no change in velocity occurs) and $c_r = 1$ denotes a completely elastic collision (kinetic energy is conserved).

Given *a priori* values for $m_A, m_B, v_A^0, v_B^0, c_r$, we are interested in determining v_A^1 and v_B^1. To do this, we solve the linear system

$$\begin{aligned} m_A v_A^0 + m_B v_B^0 &= m_A v_A^1 + m_B v_B^1 \\ c_r &= \frac{v_B^1 - v_B^0}{v_A^1 - v_A^0}. \end{aligned}$$

This yields

$$\begin{aligned} v_A^1 &= \frac{m_A v_A^0 + m_B v_B^0 + c_r m_B (v_B^0 - v_A^0)}{m_A + m_B} \\ v_B^1 &= \frac{m_B v_B^0 + m_A v_A^0 + c_r m_A (v_A^0 - v_B^0)}{m_B + m_A}. \end{aligned}$$

Intuitively, we hope the outcome does not explicitly depend on the velocities of the objects, instead in terms of momentum and mass only. We can refomulate the results as follows. (With $m_A + m_B = m_A + m_B$.)

$$
\begin{aligned}
p_A^1 &= \left((1 - c_r)p_A^0 + (1 + c_r)p_B^0\right)\left(\frac{m_A}{m_A + m_B}\right) \\
p_B^1 &= \left((1 + c_r)p_A^0 + (1 - c_r)p_B^0\right)\left(\frac{m_B}{m_A + m_B}\right).
\end{aligned}
$$

1.1 Sanity Checks on some example cases:

1. $m_A = m_B =: m, \quad v_A^0 = -v_B^0 =: v^0$

$$
\begin{aligned}
p_A^1 &= -c_r p^0 \\
p_B^1 &= c_r p^0.
\end{aligned}
$$

Here, the interpretation of c_r is upheld as the linear damping of the final momenta of two identical objects with equi-opposite initial momenta.

2. $m_A = m_B =: m$

$$
\begin{aligned}
p_A^1 &= \frac{1 - c_r}{2} p_A^0 + \frac{1 + c_r}{2} p_B^0 \\
p_B^1 &= \frac{1 + c_r}{2} p_A^0 + \frac{1 - c_r}{2} p_B^0
\end{aligned}
$$

The resulting momenta are weighted averages of the initial momenta. When $c_r = 1$, the two momenta swap so that $p_B^1 = p_A^0$ and vice versa. When $c_r = 0$, the resulting momenta are equal since the two objects are same mass and are stuck together.

3. $p_A^0 = -p_B^0 =: p^0$

$$
\begin{aligned}
p_A^1 &= -2c_r p^0 \left(\frac{m_A}{m_A + m_B}\right) \\
p_B^1 &= 2c_r p^0 \left(\frac{m_B}{m_A + m_B}\right)
\end{aligned}
$$

Both objects entered the collision with equi-opposite momentum entering the collision, and so both objects leave the collision with equi-opposite momentum, whose magnitude is linearly damped by the coefficient of restitution, and scaled by what proportion of the total mass that object accounted for.

4. $c_r = 1$ (Perfectly elastic collision)

$$\begin{aligned} p_A^1 &= 2p_B^0 \left(\frac{m_A}{m_A + m_B} \right) \\ p_B^1 &= 2p_A^0 \left(\frac{m_B}{m_A + m_B} \right). \end{aligned}$$

   Recall since this is a 1D collision, the two objects' momenta had opposite directions initially. This result shows in a perfectly elastic collision, the objects reverse directions and the magnitude of the resulting momentum is linearly proportional to the proportion of the total mass that object held.

5. $c_r = 0$ (Perfectly inelastic collision)

$$\begin{aligned} p_A^1 &= (p_A^0 + p_B^0) \left(\frac{m_A}{m_A + m_B} \right) \\ p_B^1 &= (p_A^0 + p_B^0) \left(\frac{m_B}{m_A + m_B} \right). \end{aligned}$$

   Here, the two objects have the same velocity since they are stuck together, so their final momenta vary only by what their difference in mass is.

Computational Considerations for Simulation

We make the initial assumption that all collisions are instantaneous. Formally, we let δ_t be the time step used in our simulation. All events transpire at some discrete time point t^i where $i = i, 1, 2, \ldots$ and $t^i - t^{i-1} = \delta_t$ for all $i \in \mathbb{N}$. Then, consider two circles A and B. We will adopt the notation of $trait_A$ and $trait_B$ to be analogous traits of the two circles and superscripts denote the time step at which this quantity is measured.

The traits relevant to our discussion here are x^i, v^i, a^i, p^i, R (radius) and m. Let us consider the simplest case:

1.

	m	**R**	$\mathbf{x}^0$	$\mathbf{v}^0$	$\mathbf{a}^0$
A	m	1	-1	1	0
B	m	1	1	-1	0

   Clearly at t^1 the two balls will be tangent to each other at the origin. That is, the instant of (instantaneous) collision occurs at a measured time point. Then, we adjust the momenta so that

$$\begin{aligned} v_A^1 &= -c_r \\ v_B^1 &= c_r. \end{aligned}$$

So at t^2, we have

	m	**R**	$\mathbf{x}^2$	$\mathbf{v}^2$	$\mathbf{a}^2$
A	m	1	$-c_r$	$-c_r$	0
B	m	1	c_r	c_r	0

2.

	m	**R**	$\mathbf{x}^0$	$\mathbf{v}^0$	$\mathbf{a}^0$
A	m	1	-1	2	0
B	m	1	1	-1	0

Here, we have a more complicated collision, since the instant of collision occurs between t^0 and t^1.

With zero acceleration, we have

$$t_{collide} = \frac{x_A^0 - x_B^0}{v_B^0 - v_A^0} = t_0 + \frac{2}{3}\delta_t.$$

Regardless of which instant the collision occurs, we know

$$v_A^1 = \frac{-3c_r}{2} + \frac{1}{2}.v_B^1 \qquad\qquad = \frac{3c_r}{2} + \frac{1}{2}.$$

But since the collision occurs before t_1, we need a position update at a resolution $\delta_t/3$.

$$\begin{aligned} x_A^1 &= x_A^0 + \left(\frac{2v_A^0}{3} + \frac{v_A^1}{3}\right)\delta_t = -1 + \left(\frac{4}{3} + \frac{1-3c_r}{6}\right)\delta_t = -1 + \left(\frac{3}{2} - \frac{c_r}{2}\right)\delta_t \\ x_B^1 &= x_B^0 + \left(\frac{2v_B^0}{3} + \frac{v_B^1}{3}\right)\delta_t = 1 + \left(\frac{-2}{3} + \frac{1+3c_r}{6}\right)\delta_t = 1 + \left(\frac{-1}{2} + \frac{c_r}{2}\right)\delta_t. \end{aligned}$$

1.2 Error Analysis

If we instead assume $\frac{2}{3}\delta_t \approx \delta_t$, we then get

$$\begin{aligned} x_A^1 &= x_A^0 + v_A^0\delta_t = -1 + 2\delta_t \\ x_B^1 &= x_B^0 + v_B^0\delta_t = 1 - \delta_t. \end{aligned}$$

The question remains: *How large can the error in this approximation get?*

2 Computational Cost of Collision Resolution

We use the Velocity-Verlet algorithm [1] which updates

$$\begin{aligned} x^1 &= x^0 + v^0\delta_t + \frac{1}{2}a^0\delta_t^2 \\ v^1 &= v^0 + \frac{1}{2}\left(a^0 + a^1\right)\delta_t \\ a^1 &= \frac{1}{m}\sum_i f_i. \end{aligned}$$

So, if a collision is detected between objects A and B at time step t^1, then we must identify the $t^* \in [t^0, t^1]$ at which

$$x_A^0 + v_B^0 t^* + \frac{1}{2}a_A^0\left(t^*\right)^2 = x_B^0 + v_B^0 t^* + \frac{1}{2}a_B^0\left(t^*\right)^2$$

which is found by computing (note the superscripts are all 0 and left off for clarity of presentation)

$$t_\pm^* = \frac{v_B - v_A \pm \sqrt{(v_A - v_B)^2 - 2(a_A - a_B)(x_A - x_B)}}{a_A - a_B}$$

and choosing the root contained in the interval $[t_0, t_1]$.

We compute the computational cost of each of these steps by a more granular approach than simple FLOP count. Since not all flops are created equal, we use the following conversions (normalized to units of `adds`)

$+$	1 `add`
-	1 `add`
$<, >, \leq, \geq$	1 `add`
$*$	1.5 `add`
$\div$	4 `add`
$\sqrt{\cdot}$	6 `add`

With this standard, the velocity verlet update of position, velocity and acceleration per object is 17.5 `adds` per dimension. Since we are concerned with 2D physical simulation, this should cost 35 `adds`.

Computing the two roots using the quadratic equation costs 21.5 `adds` to find the two roots and then 2 `adds` to determine which root is the correct one, resulting in a total of 23.5 `adds`. Note that we only need to compute this in one dimension as long as we are only concerned with rigged convex polygons.

[1] https://en.wikipedia.org/wiki/Verlet_integration#Velocity_Verlet

Though this computation might be more complex than this, since we need to know exactly where the two objects are going to collide. What if we look where the centers of mass are closest?