Rigid Body Dynamics

Rigid body dynamics is the physical simulation of objects subject to external forces such as gravity and collisions. These bodies are rigid in the sense that they will not deform due to these forces. Newton's second law of motion is used to integrate the position of an object over time and collision forces are handled using two different techniques: impulses and springs. Two methods of integration are demonstrated, Euler, for simplicity, and the fourth order Runge-Kutta method (RK4) for accuracy.

Euler's Method

All numerical integration methods make use of initial value and a function that expresses the rate of change of that quantity with respect to time.

$$y_{i+1} = y_i + f(t, y_i)$$

The simplest form of numerical integration is known as Euler's method. This method estimates the future value by adding a small distance along the slope at the given point to the previous estimate. In our case, we will be doing two euler integrations simultaneously. One to estimate the velocity and another to estimate position using the velocity estimate just calculated.

$$\begin{array}{ll} v = v_0 + at & \quad v_{i+1} = v_i + tf'(v_i) \\ x = x_0 + vt & \quad x_{i+1} = x_i + tf'(x_i) \end{array}$$

The initial velocity and position can be set arbitrarily and we solve for acceleration using newtons second law of motion:

$$f = ma \qquad \text{or} \qquad a = \frac{f}{m}$$

Rotational Motion and the Inertial Tensor

These equations handle translational motion, rotational motion is similar, however the rotational mass of an object is less intuitive as we dont frequently discuss the moment of inertia or how much torque is required to rotate an object. This quantity is further complicated by the ability to rotate around any arbitrary axis.

$$\alpha = \tau I^{-1}$$
$$\omega = \omega_0 + \alpha t$$
$$\mathrm{M} = \mathrm{M}_0 + M * \begin{bmatrix} 0 & -\omega.z & \omega.y \\ \omega.z & 0 & -\omega.x \\ -\omega.y & \omega.x & 0 \end{bmatrix} * t$$

Inertial Tensor

The inertial tensor, I, is a 3x3 matrix that describes how hard it is to rotate an object around any arbitrary axis. Equations exist to calculate the inertia tensor for common objects such as spheres, cubes, cylinders, hoops etc. And it is possible to calculate an intertial tensor using calculus for an object of arbitrary shape and uniform density. An important note is that the inertia tensor depends on the bodies orientation. If the body rotates, the inertial tensor changes. For this reason we calculate the inertial tensor in an initial orientation and solve for the rotated tensor using the current orientation matrix.

$$\text{sphere} = \begin{bmatrix} \frac{2}{5}mr^2 & 0 & 0 \\ 0 & \frac{2}{5}mr^2 & 0 \\ 0 & 0 & \frac{2}{5}mr^2 \end{bmatrix}$$

$$\text{cube} = \begin{bmatrix} \frac{m}{12}(y^2+z^2) & 0 & 0 \\ 0 & \frac{m}{12}(x^2+z^2) & 0 \\ 0 & 0 & \frac{m}{12}(x^2+y^2) \end{bmatrix}$$

$$\text{I} = \begin{bmatrix} \int(y^2+z^2)\ dm & -\int xy\ dm & -\int xz\ dm \\ -\int xy\ dm & \int(x^2+z^2)\ dm & -\int yz\ dm \\ -\int xz\ dm & -\int yz\ dm & \int(x^2+y^2)\ dm \end{bmatrix}$$

Collision Detection

At this point the objects will translate and rotate in response to an applied force and torque. We must still apply these forces to bodies in response to collisions with other objects and planes defining the world.

The distance from a point to a plane can be determined as follows:

$d = \frac{|Ax+By+Cz+D|}{\sqrt{A^2+B^2+C^2}}$

If the plane's normal vector is of unit length, then this simplifies to:

$d = n \cdot (x, y, z) + D$

In between every iteration of integration we check if the distance is between some value epsilon. If $d < \epsilon$ then we consider the objects to have collided. If it has moved too far into the plane, we discard all calculated results and restart the simulation with a smaller time step. In order to perform this computation in real time, we greatly simplify the object by only considering the eight verticies of an axis aligned bounding box or by considering a single point of a bounding sphere.

Impulse Response

After it is determined that a collision has occurred, we can apply the impulse response. Since the world is considered immovable it is assumed to have infinite mass. This greatly simplifies the impulse response equation and we present this version first. This equation is similar to that of vector reflection, but has added terms to account for mass.

$$\rho = \frac{-(1+e)v_{rel}}{\frac{1}{m}+n\cdot(I^{-1}*(r\times n))}$$

* The constant e represents the coefficient of restitution, a value that represents the amount of energy lost in the collision.

* The v_{rel} value indicates the relative velocity between the two colliding objects, in this case since the world plane will always have a zero velocity, it is simply the dot product of the velocity of the colliding object with the plane's normal.

* The radius value, r, is the distance between the point of collision and the center of mass of the object. The resulting ρ is in units of momentum ($\rho = mv$)

The complete impulse response equation follows:

$$\rho = \frac{-(1+e)v_{rel}}{\frac{1}{m_a}+\frac{1}{m_b}+n\cdot(I_a^{-1}*(r_a\times n))+n\cdot(I_b^{-1}*(r_b\times n))}$$

We apply the final results to our velocity and angular velocity state variables as follows:
$v = v_0 + \frac{\rho}{m}$
$\omega = \omega_0 + I^{-1}(\rho \times r)$

Spring based collisions

A spring force can be modeled as follows:
$F = -kd$ where
k = spring constant
d = displacement

Spring based collision response uses stiff springs to apply the force required to prevent two objects from interpenetrating. We let $displacement = \frac{1}{d^2}$ in order to apply an exponentially large force when the distance between the plane and the object approach zero. Collisions between bodies can be handled in the same way and often resting contact is best handled using spring forces instead a series of impulse responses.

Runge Kutta Method of order four (RK4)

This method is similar to euler's method except a better estimate for the rate of change is used. Euler integration fails to accurately describe any differential equation that involves exponential change due to it's linear method of estimation. These numerical inaccuracies over time can cause issues. For example, using euler integration in orbiting simulations such as n-body problems can lead to overestimations that result in a body slowly drifting away from it's desired orbit over time. The lorenz equation is another situation where simple euler integration fails to maintain the correct trajectories. However, RK4 requires that we provide a function for future accelerations and velocities. Creating a function for these future values is not immediately intuitive.

RK4:
$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Where:
$k_1 = f(t_i, y_i)$
$k_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_1)$
$k_3 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}s_2)$
$k_4 = f(t_i + h, y_i + hs_3)$
$h = stepsize$
$t_i = timevalue$