A Simple Python Simulator

In classical physics, the EOM of a system can be determined from the potential in the system. Here, I will investigate a discrete particle simulation The project consists of three main python files.

- **config.py**: holds all the configuration information for the project.
- **driver.py**: computes the particle dynamics and saves them to a file.
- **visualize.py**: creates animations to view the simulations.

Physics Relationships

Looking at Potential Energy, we have:

$$\overrightarrow{\mathbf{F}} = -\nabla \mathbf{U} = \langle -\frac{\partial \mathbf{U}}{\partial x}, -\frac{\partial \mathbf{U}}{\partial y} \rangle \tag{1}$$

$$\overrightarrow{\mathbf{F}} = m\overrightarrow{\mathbf{a}} \tag{2}$$

The time derivative of velocity is acceleration so assuming that the potential energy is time independant. The change in velocity and position from $t_n \to t_{n+1}$ is:

$$\Delta\overrightarrow{\mathbf{v}} = \overrightarrow{\mathbf{a}}(t_{n+1} - t_n) \tag{3}$$

$$\Delta\overrightarrow{\mathbf{x}} = \overrightarrow{\mathbf{v}}(t_{n+1} - t_n) \tag{4}$$

Inelastic vs. Elastic Collisions

In both types of collisions, the total momentum before the collision is equal to the total colision after the collision.

Perfectly elastic collisions also conserve kinetic energy. This constraint results makes the objects bounce off one another.

On the other hand, perfectly Inelastic collisions do not conserve kinetic energy. Instead a perfectly Inelastic collisions

Real collisions are somewhere in between perfectly elastic ans perfectly inelastic. These collisions can be modelled with a coeffiecient of restitution (COR).

For a collision, we have:

$$m_1\overrightarrow{v_{1b}} + m_2\overrightarrow{v_{2b}} = m_1\overrightarrow{v_{1a}} + m_2\overrightarrow{v_{2a}} \tag{5}$$

$$C_R = -\frac{\overrightarrow{v_{1a}} - \overrightarrow{v_{2a}}}{\overrightarrow{v_{1b}} - \overrightarrow{v_{2b}}} \tag{6}$$

The coeffiecient of restitution then is defined as the amount of relative velocity that the system keeps after the collision. In one dimension using the C_R and the conservation of momentum, we obtain:

$$\overrightarrow{v_{1a}} = \frac{m_1\overrightarrow{v_{1b}} + m_2\overrightarrow{v_{2b}} + m_2C_R(\overrightarrow{v_{2b}} - \overrightarrow{v_{1b}})}{m_1 + m_2} \tag{7}$$

$$\overrightarrow{v_{2a}} = \frac{m_1\overrightarrow{v_{1b}} + m_2\overrightarrow{v_{2b}} + m_1C_R(\overrightarrow{v_{1b}} - \overrightarrow{v_{2b}})}{m_1 + m_2} \tag{8}$$

Since the above equations are in one dimendion, we need to project the velocities into coordinates normal and parallel to the collision.

Given a velocity $\overrightarrow{v} = v_x\hat{\imath} + v_y\hat{\jmath}$ with a collision vector, v_c, and normal vector, v_n.

$$\overrightarrow{v_c} = (\overrightarrow{v} \cdot \widehat{v}_c)\widehat{v}_c \tag{9}$$

$$\overrightarrow{v_n} = (\overrightarrow{v} \cdot \widehat{v}_n)\widehat{v}_n \tag{10}$$

So we can get the collision components and treat the colision as one dimensional. Once I find the change in velocity from the collision, I can resemble the vector in terms of $\hat{\imath}$ and $\hat{\jmath}$.