

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:

$$\vec{\mathbf{F}} = -\nabla \mathbf{U} = \left\langle -\frac{\partial \mathbf{U}}{\partial x}, -\frac{\partial \mathbf{U}}{\partial y} \right\rangle \quad (1)$$

$$\vec{\mathbf{F}} = m \vec{\mathbf{a}} \quad (2)$$

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

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

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

Inelastic vs. Elastic Collisions

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

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

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

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

For a collision, we have:

$$m_1 \vec{v}_{1b} + m_2 \vec{v}_{2b} = m_1 \vec{v}_{1a} + m_2 \vec{v}_{2a} \quad (5)$$

$$C_R = -\frac{\vec{v}_{1a} - \vec{v}_{2a}}{\vec{v}_{1b} - \vec{v}_{2b}} \quad (6)$$

The coefficient 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:

$$\vec{v}_{1a} = \frac{m_1 \vec{v}_{1b} + m_2 \vec{v}_{2b} + m_1 C_R (\vec{v}_{2b} - \vec{v}_{1b})}{m_1 + m_2} \quad (7)$$

$$\vec{v}_{2a} = \frac{m_1 \vec{v}_{1b} + m_2 \vec{v}_{2b} + m_2 C_R (\vec{v}_{1b} - \vec{v}_{2b})}{m_1 + m_2} \quad (8)$$

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

Given a velocity $\vec{v} = v_x \hat{i} + v_y \hat{j}$ with a collision vector, \vec{v}_c , and normal vector, \vec{v}_n .

$$\vec{v}_c = (\vec{v} \cdot \hat{v}_c) \hat{v}_c \quad (9)$$

$$\vec{v}_n = (\vec{v} \cdot \hat{v}_n) \hat{v}_n \quad (10)$$

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