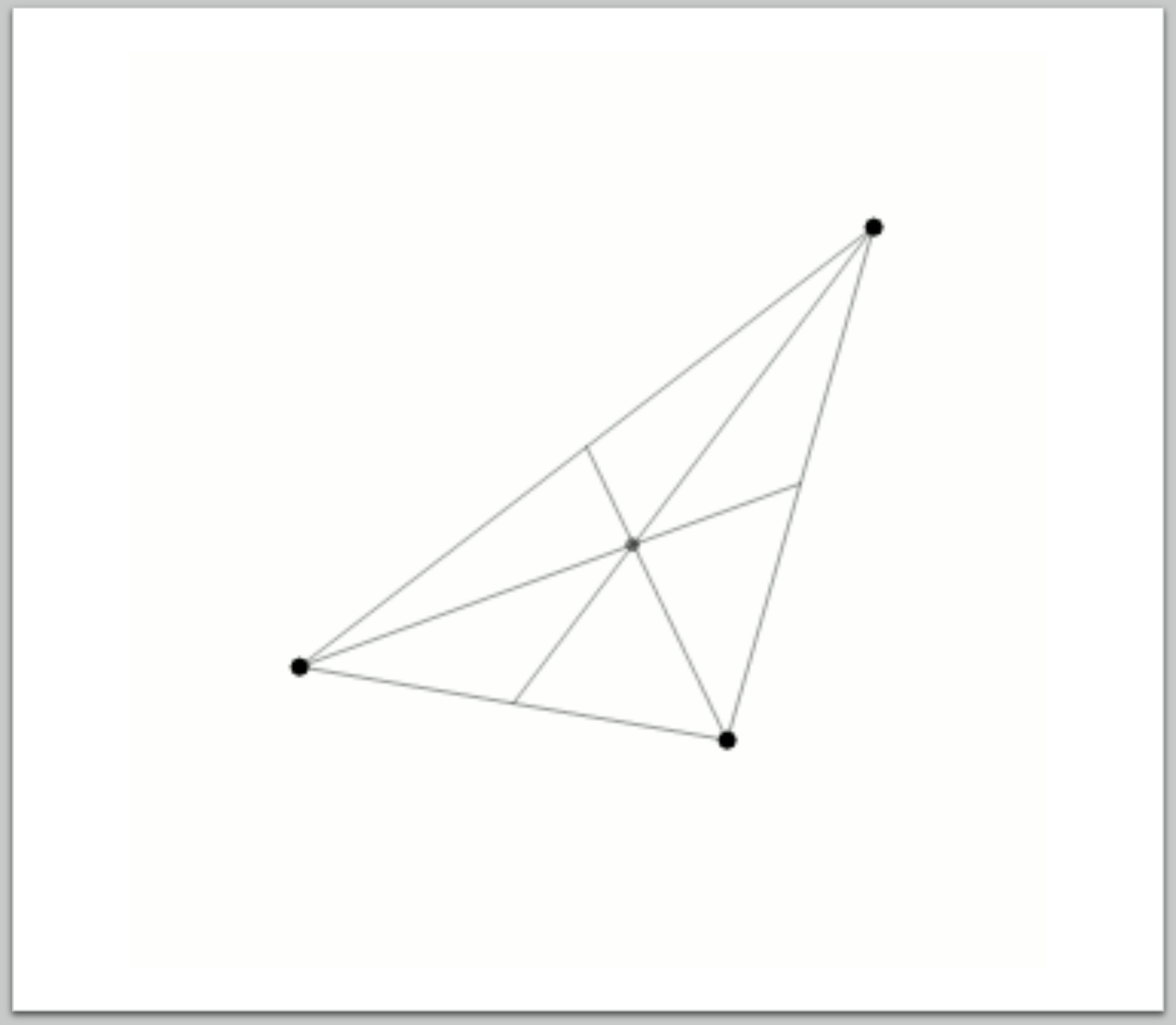
2-Body System Simulation and Prediction using Regression

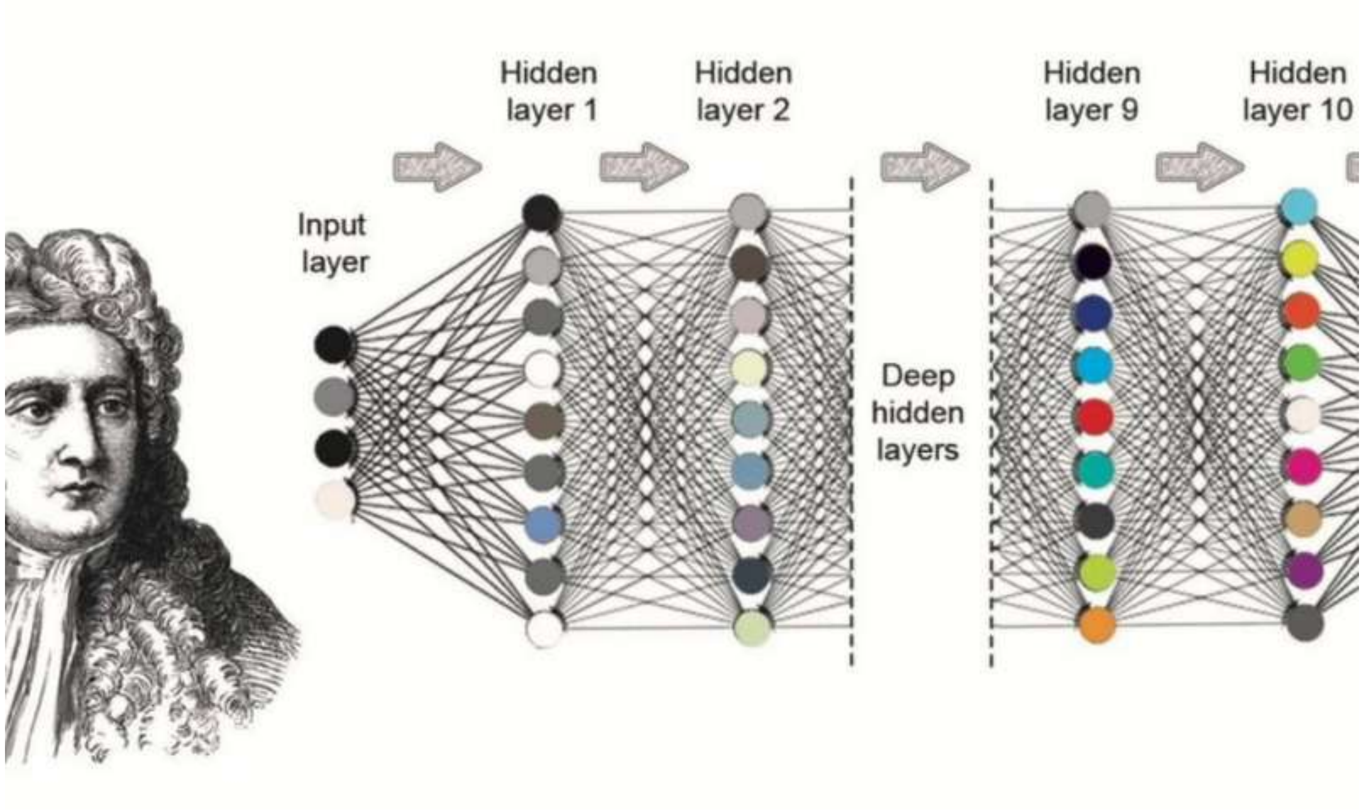
Edwin Tomy George

Three Body Problem

- 1600s - 1700s
- Chaotic nature
- Cannot be solved with equations in most cases
- Currently solved with computer simulations



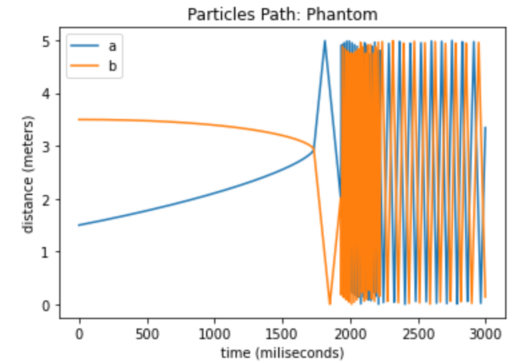
Solving the three-body problem faster using a deep neural network

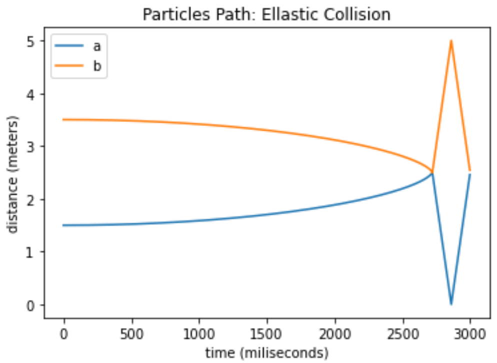


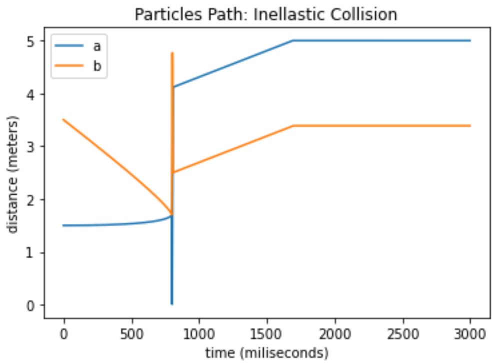
(Green et at. 2019) **https://arxiv.org/abs/1910.07291**

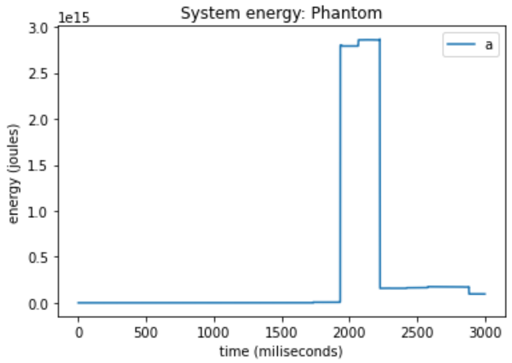
- Simulate 2 particle system where particles are affected by each other’s gravity
- Closed one dimensional space of length 5m
- Particle masses between 10 billion and 50 billion kg
- Initial velocities between -1 and 1 m/s
- Initial positions between 0 and 5m

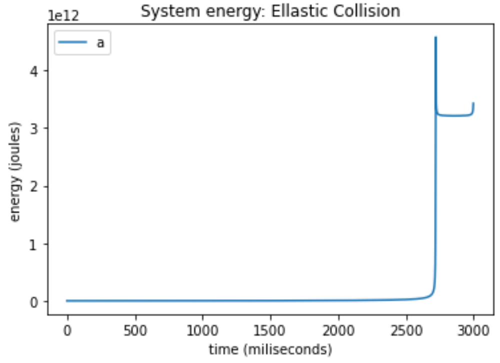
Initial Modelling

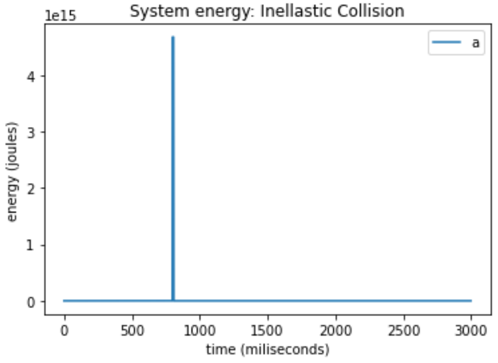




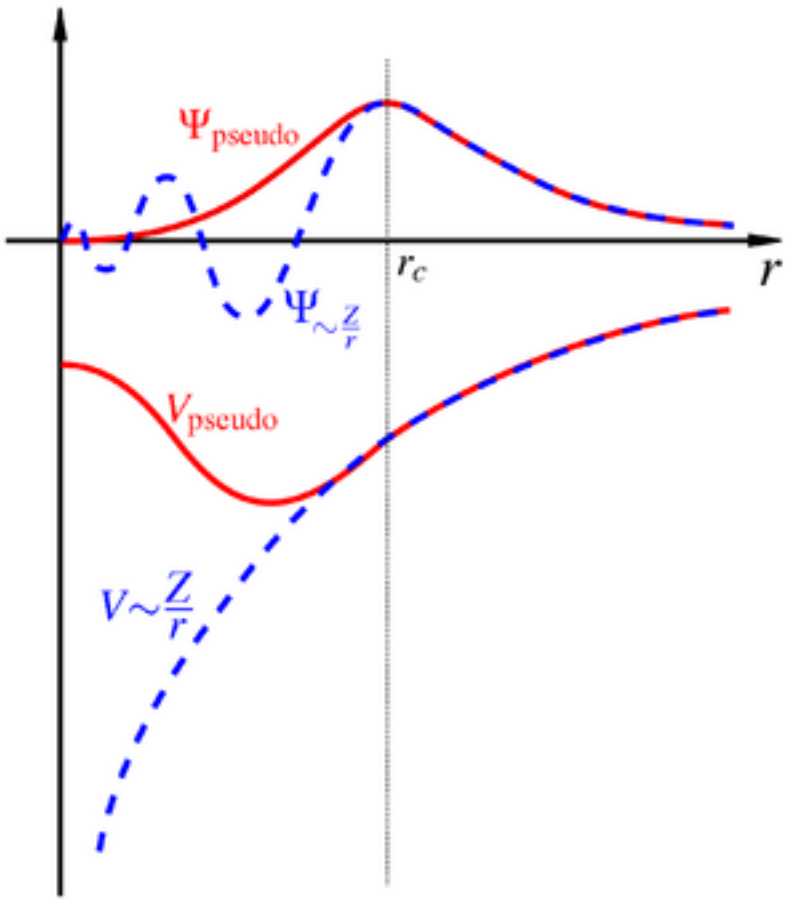








Pseudopotential



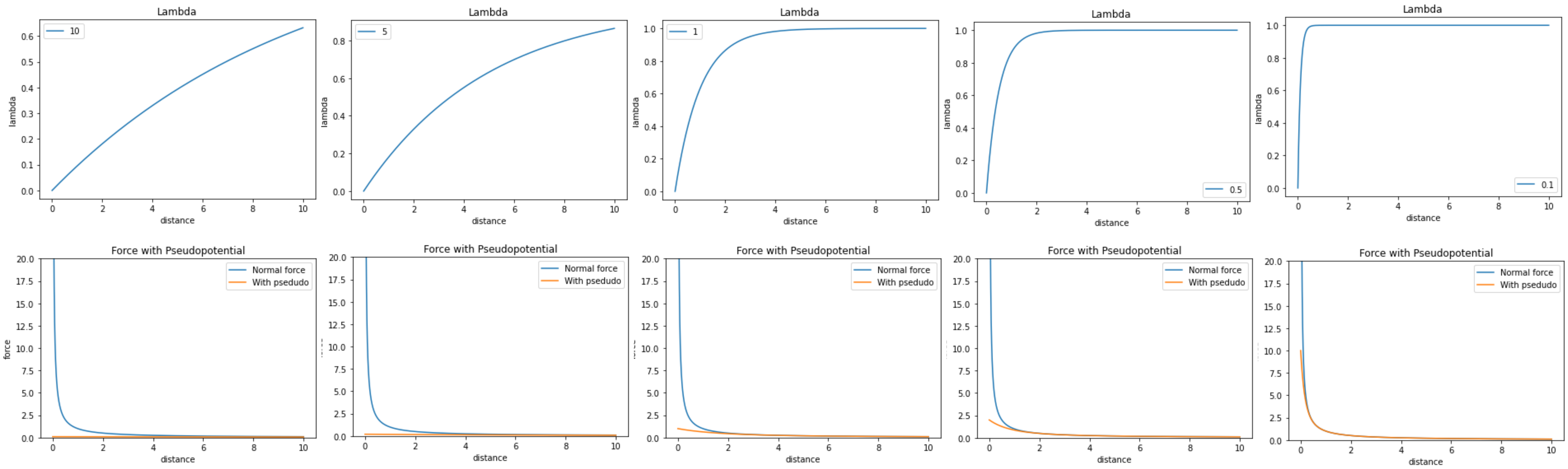
$$F = \frac{Gm_1m_2}{r^2}$$

$$F_{pseudo1} = \frac{Gm_1m_2}{r^2} * soften$$

$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

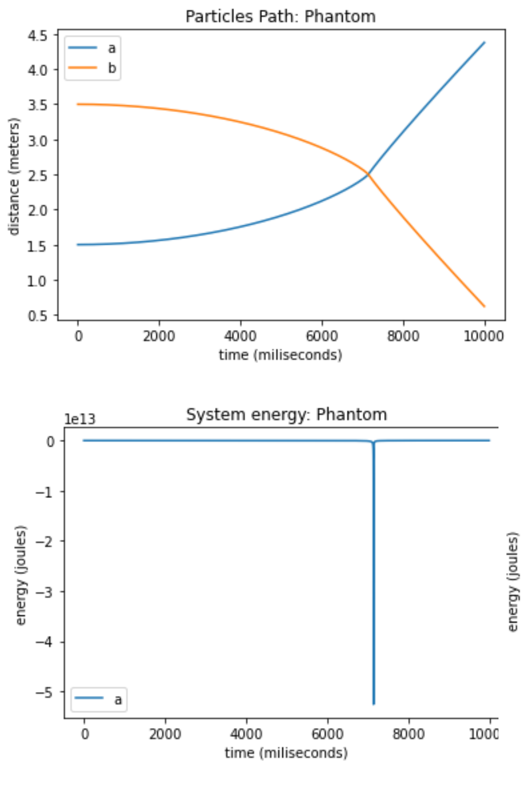
$$F_{pseudo1} = \frac{Gm_1m_2}{r^2} * soften$$

- $soften = 1 - e^{\frac{-r}{lambda}}$

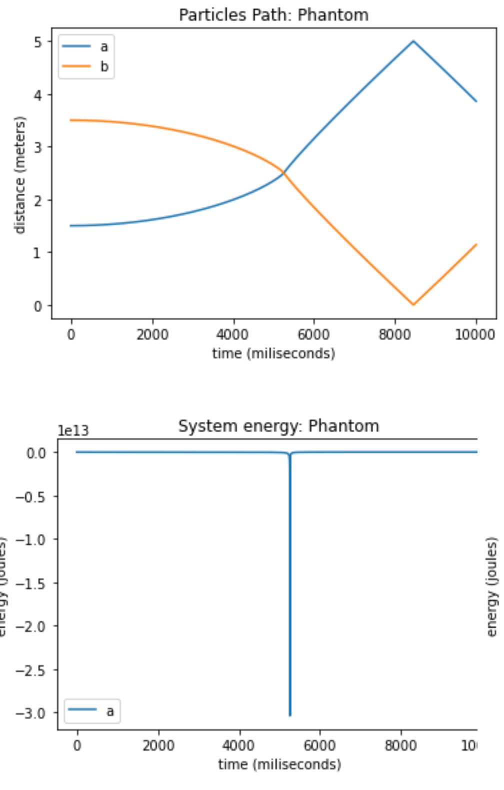


$$F_{pseudo1} = \frac{Gm_1m_2}{r^2} * soften$$

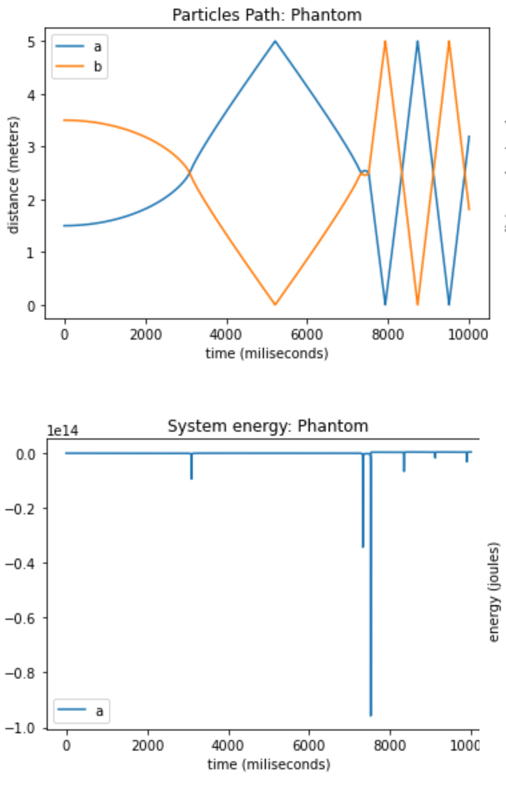
- $soften = 1 - e^{\frac{-r}{lambda}}$



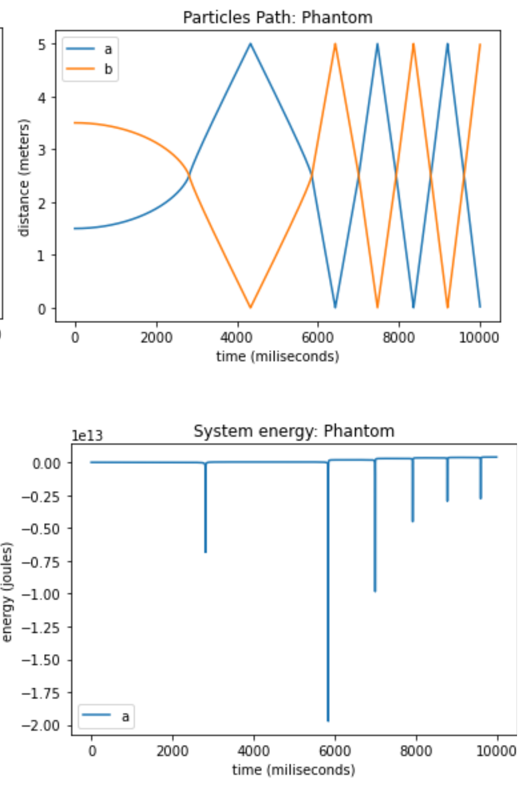
lambda = 10



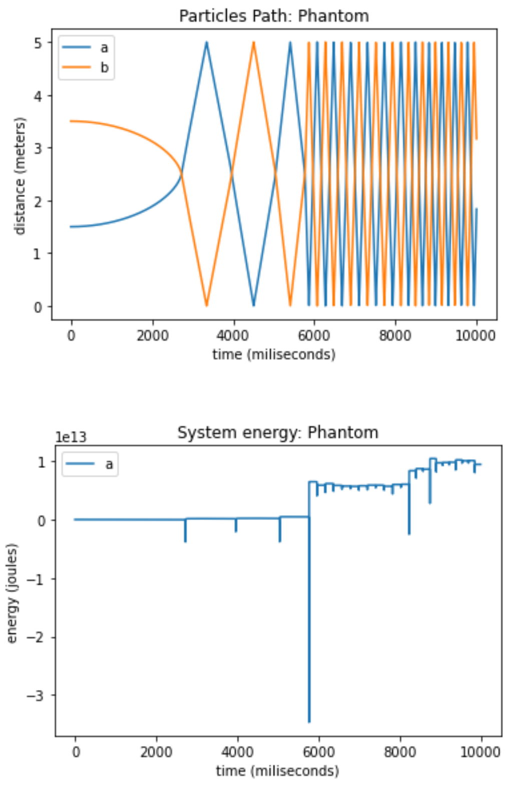
lambda = 5



lambda = 1



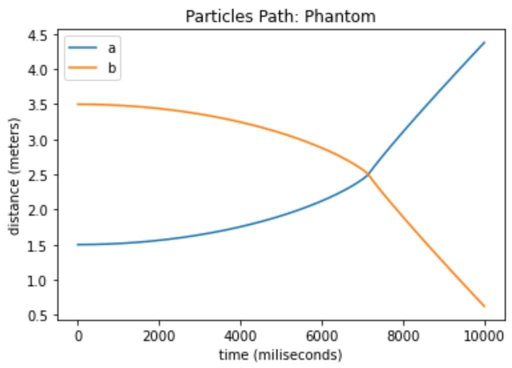
lambda = 0.5

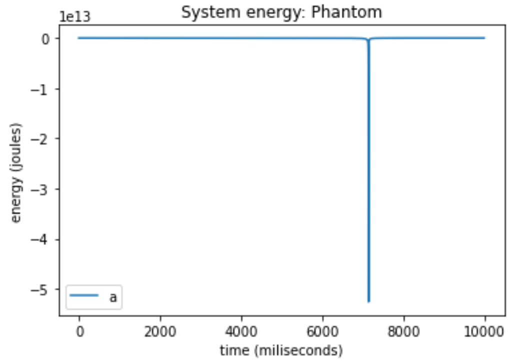


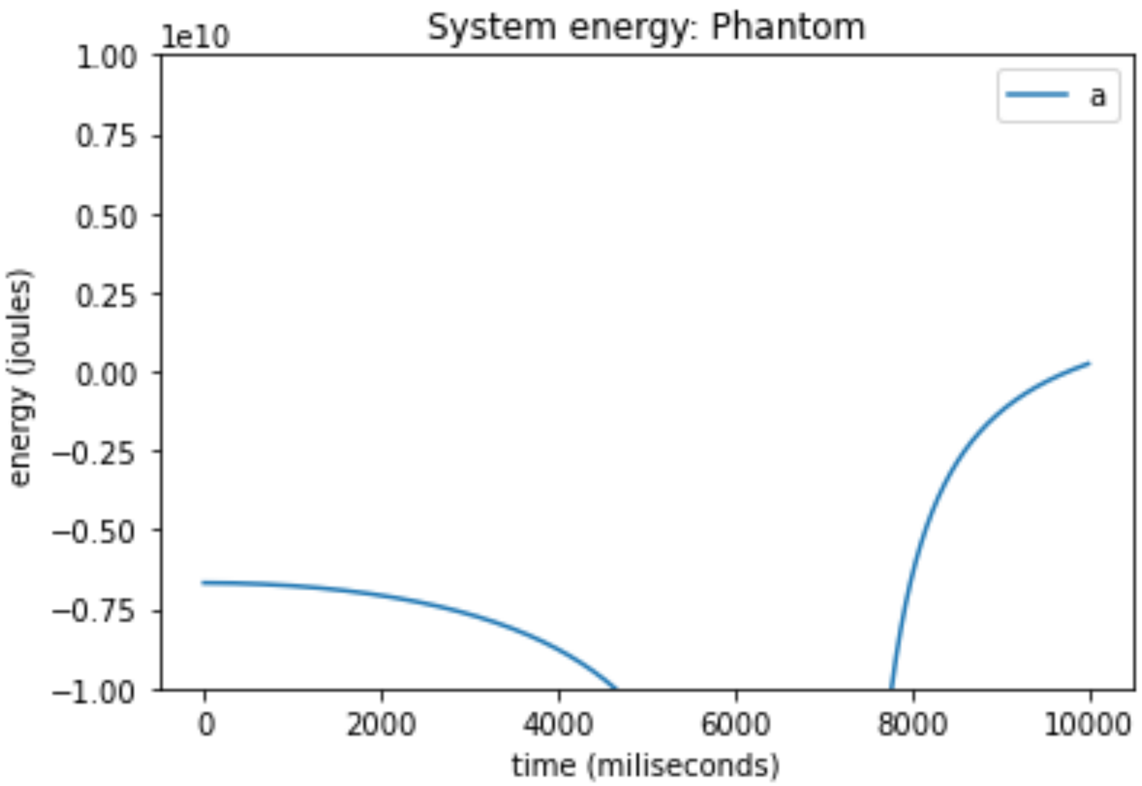
lambda = 0.1

$$F_{pseudo1} = \frac{Gm_1m_2}{r^2} * soften$$

- $soften = 1 - e^{\frac{-r}{lambda}}$

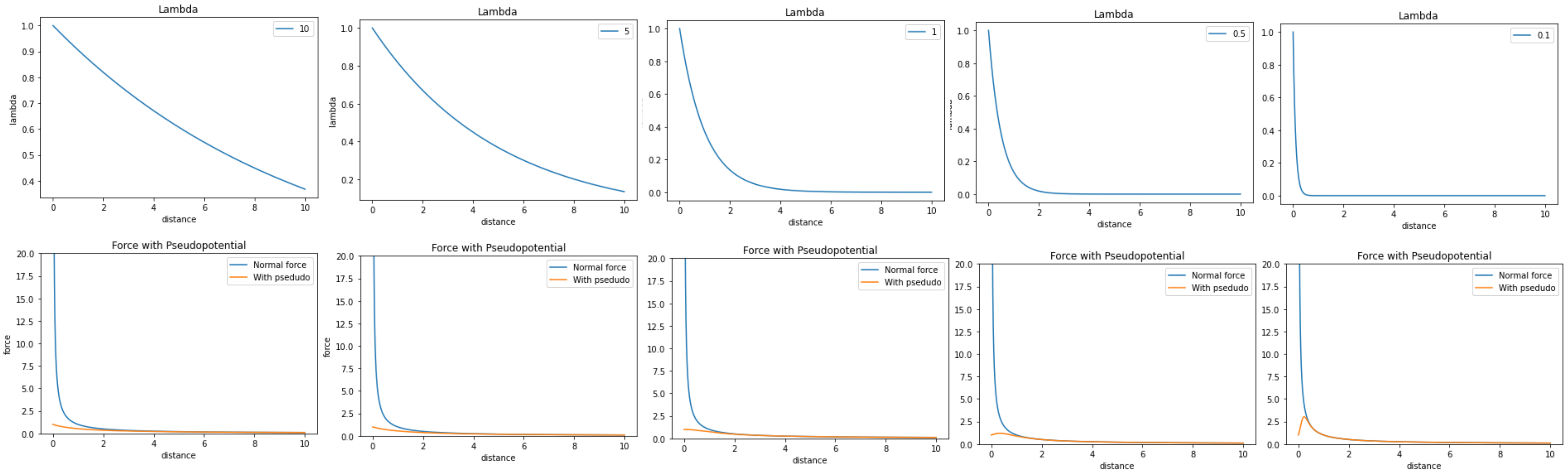






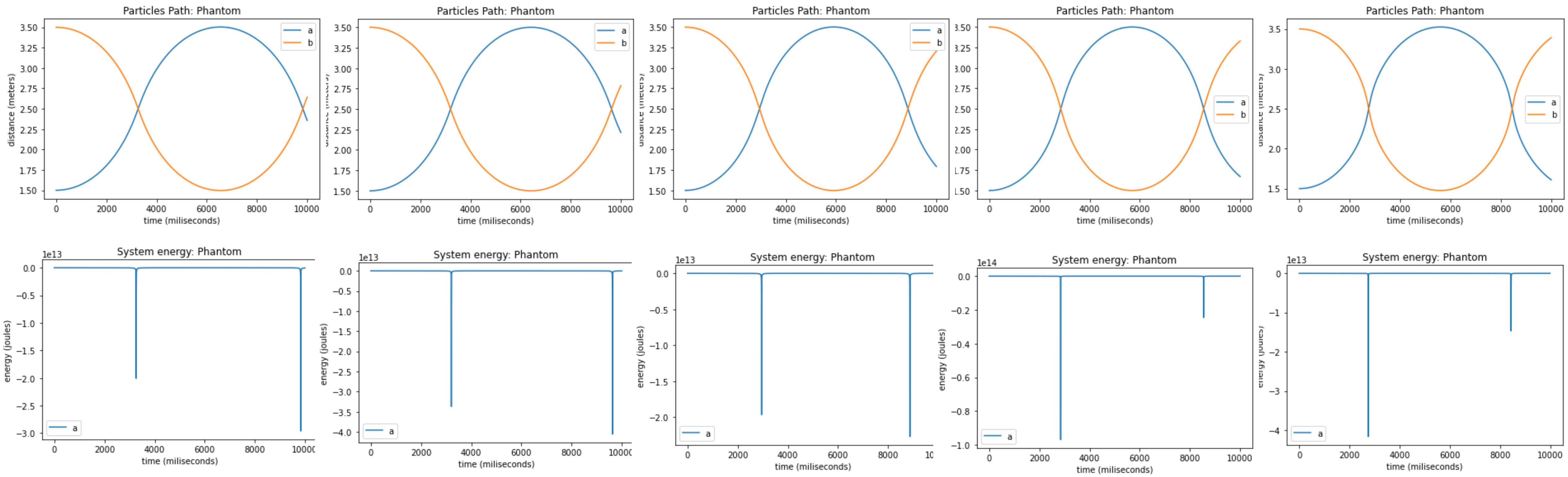
$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

- $soften = e^{\frac{-r}{lambda}}$



$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

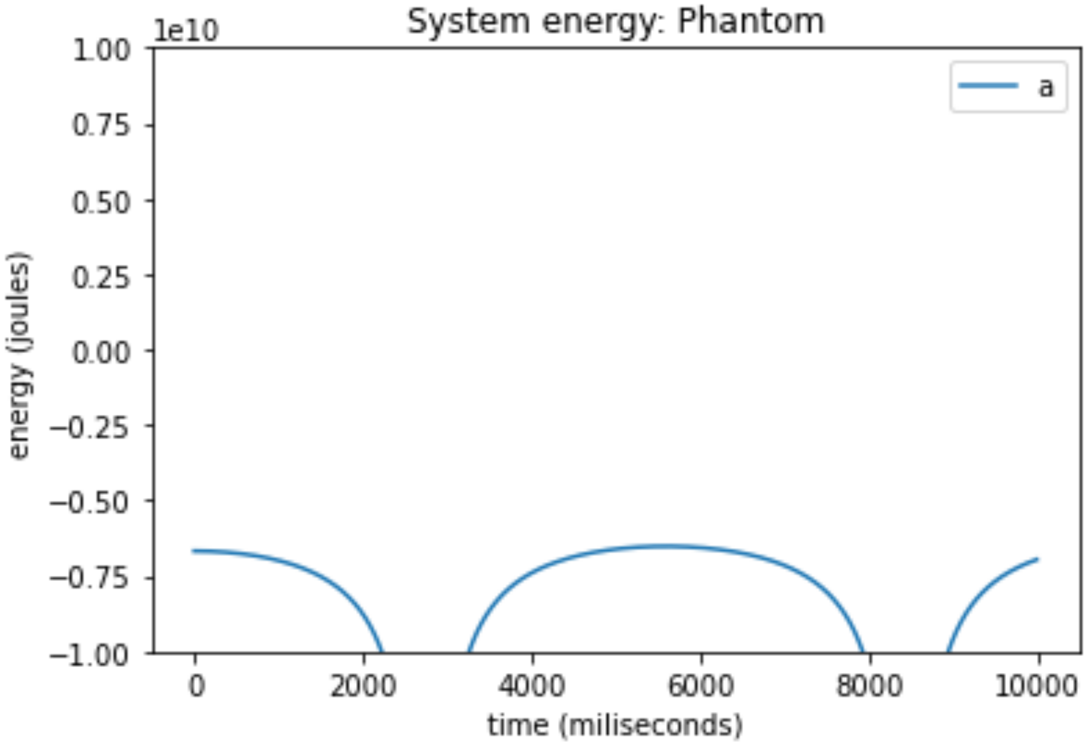
- $soften = e^{\frac{-r}{lambda}}$

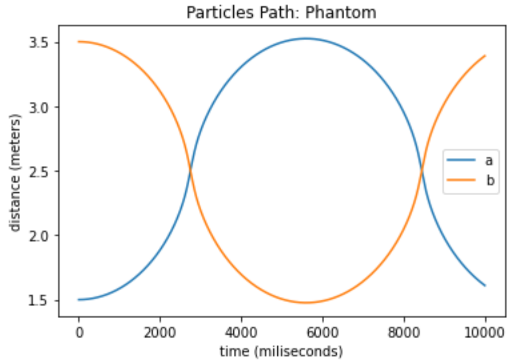


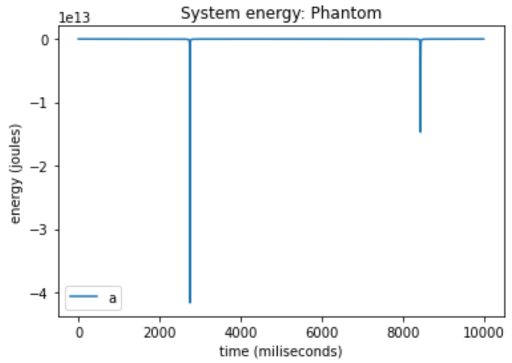
lambda = 10 | lambda = 5 | lambda = 1 | lambda = 0.5 | lambda = 0.1

$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

- $soften = e^{\frac{-r}{lambda}}$



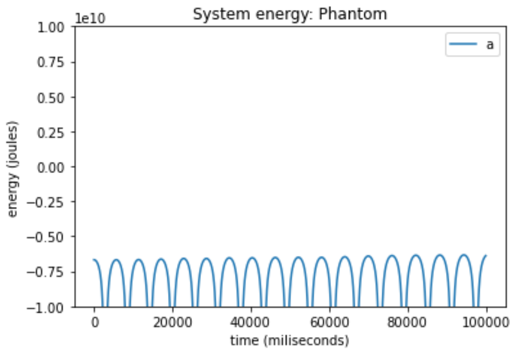


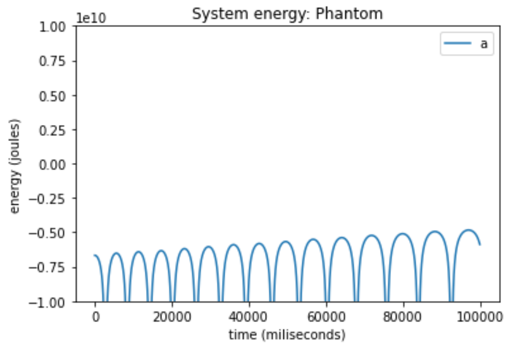


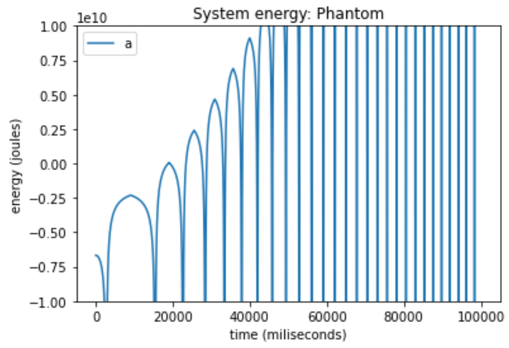
$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

- $soften = e^{\frac{-r}{lambda}}$

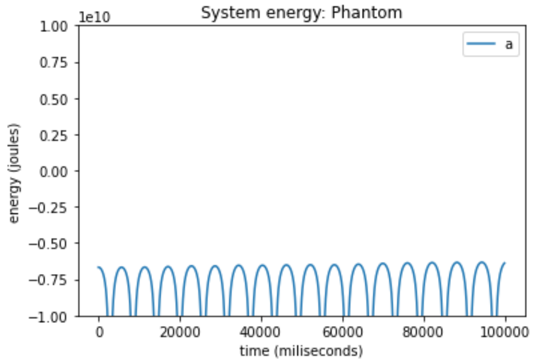
1 millisecond

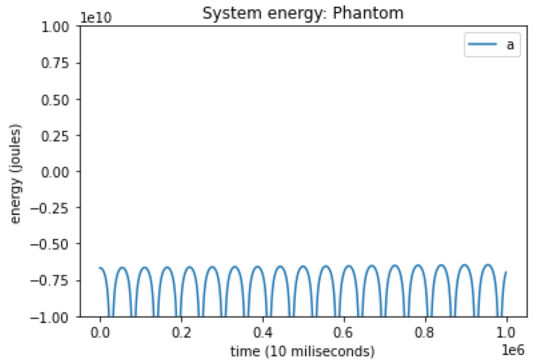


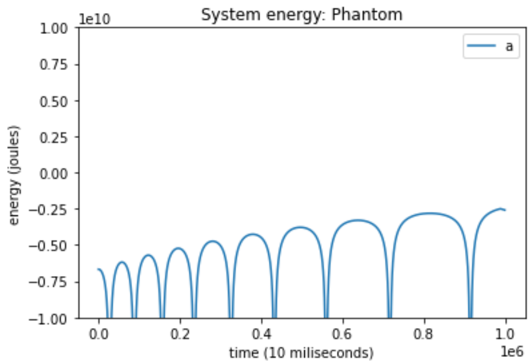




100 microseconds







lambda = 0.5

lambda = 0.1

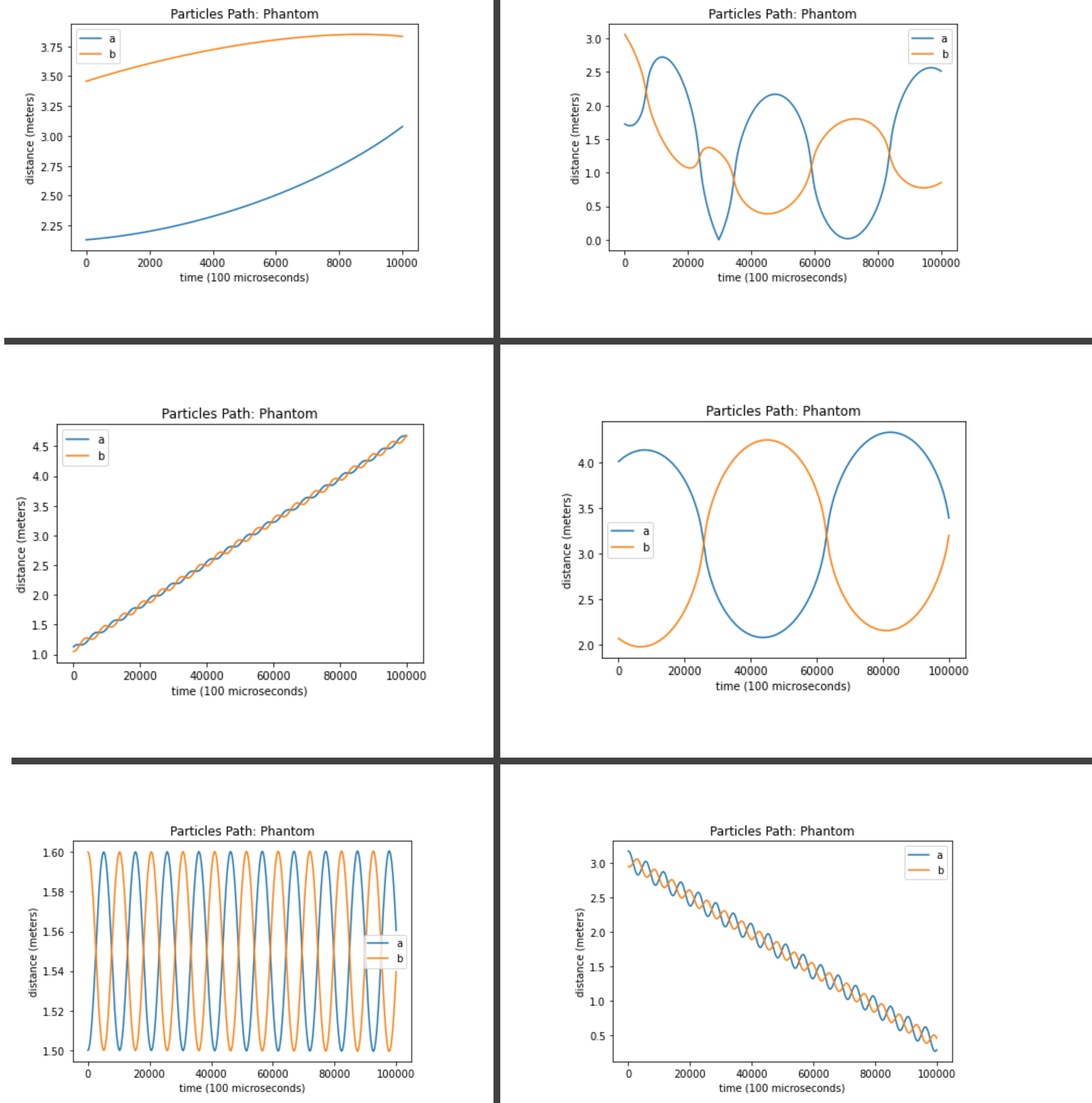
lambda = 0.01

$$F_{pseudo2} = \frac{Gm_1m_2}{r^2 + soften}$$

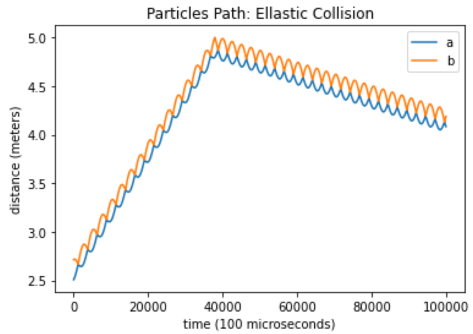
- $soften = e^{\frac{-r}{lambda}}$

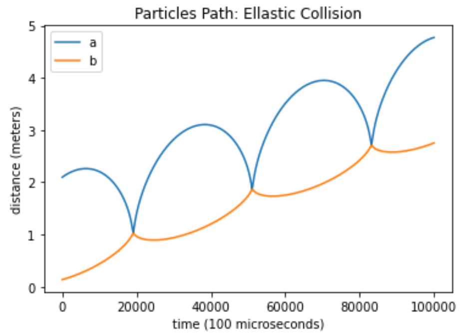
Time step	Time to simulate 100 seconds of trajectory
1 millisecond = 10^{-3} seconds	2.48 seconds
100 microsecond = 10^{-4} seconds	17.55 seconds
10 microsecond = 10^{-5} seconds	205.56 seconds
1 microsecond = 10^{-6} seconds	+30 minutes

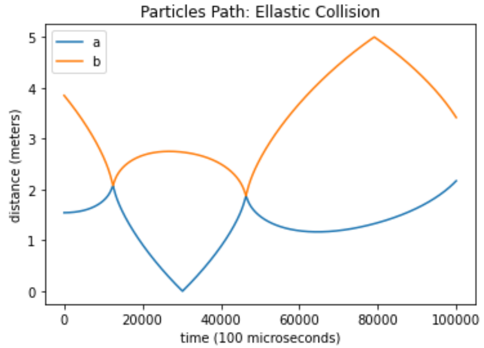
Graph Trajectories: Phantom Particles

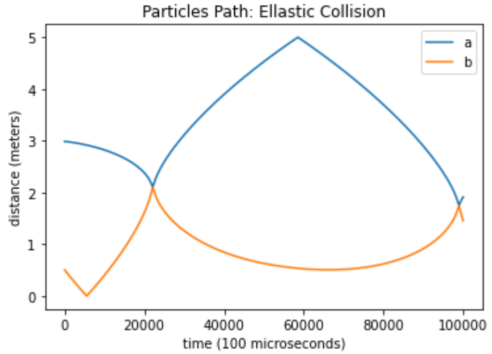


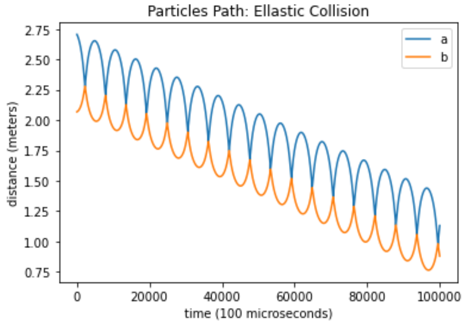
Graph Trajectories: Elastic Collision

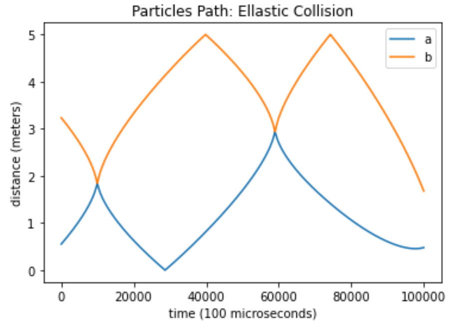










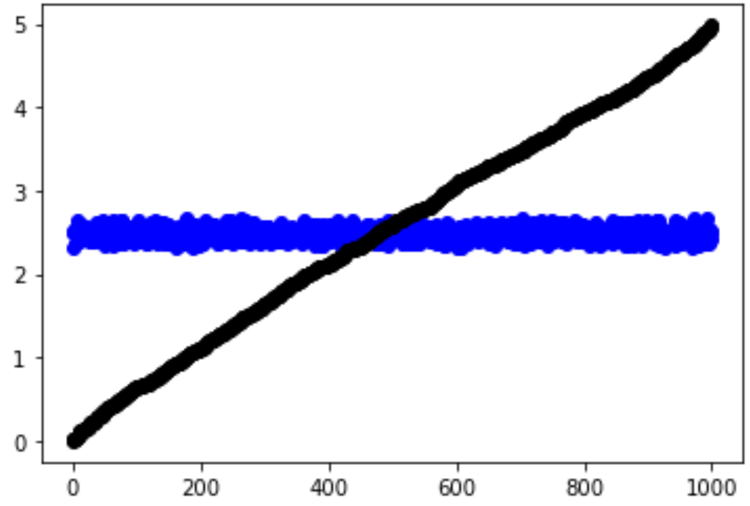


Objectives of the Regression

- Final position of particle in phantom and elastic collisions
- Time of collision
- If they collide or not

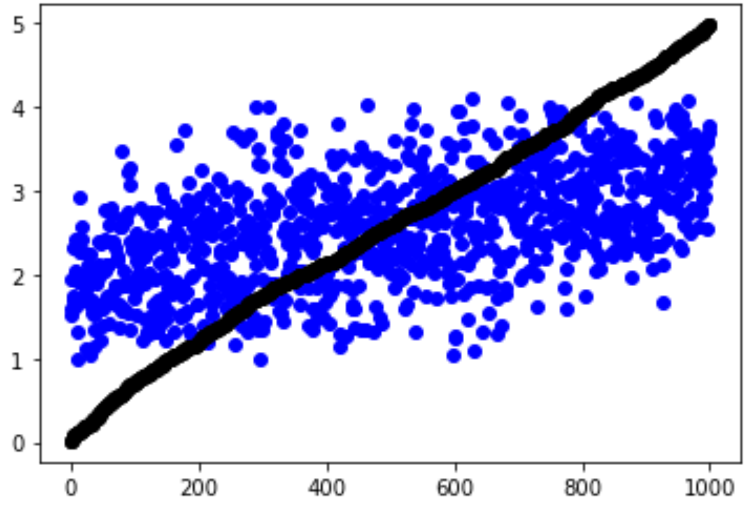
Initial Linear Regression

Phantom Collision

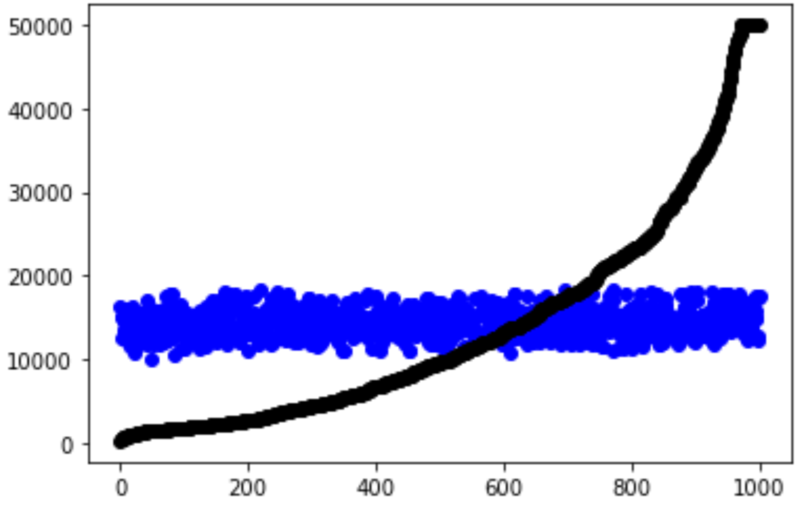


Root Mean Squared Error: 1.36

Elastic Collision



Root Mean Squared Error: 1.16



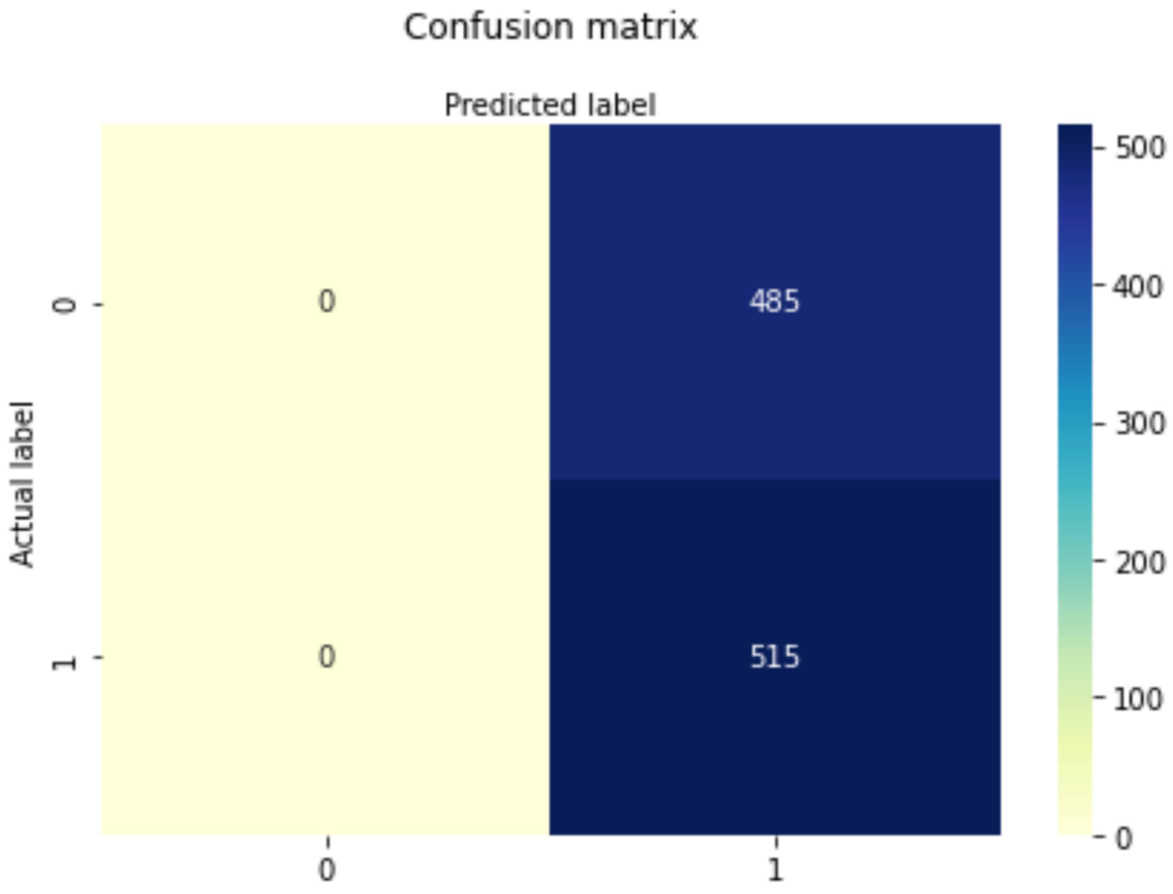
Root Mean Squared Error: 12595

Blue dots = predicted final positions of first particle
Black dots = actual final positions of first particle

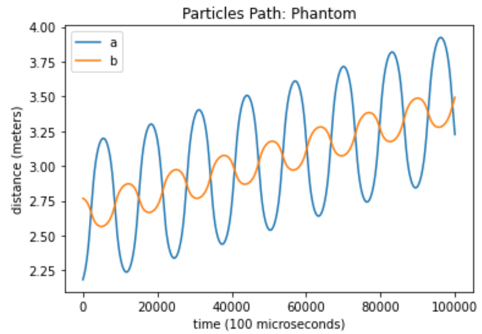
Blue dots = predicted collision time
Black dots = actual collision time

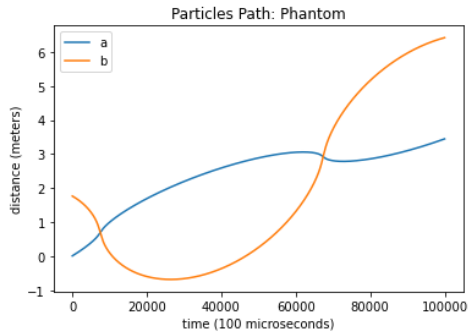
Initial Logistic Regression

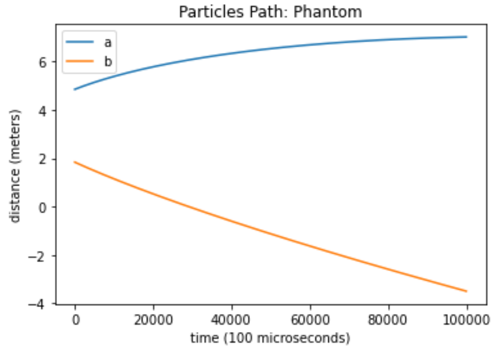
Elastic Collision: If particles collide

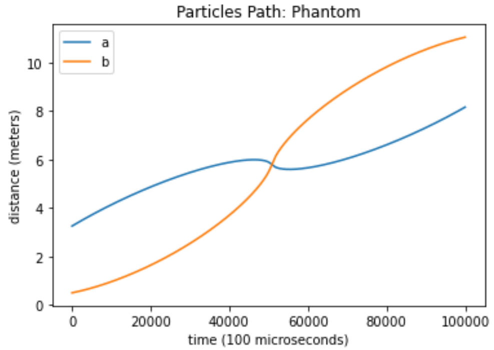


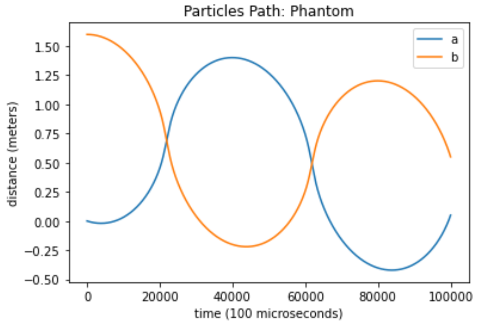
Graph Trajectories: Phantom Particles w/o bounds

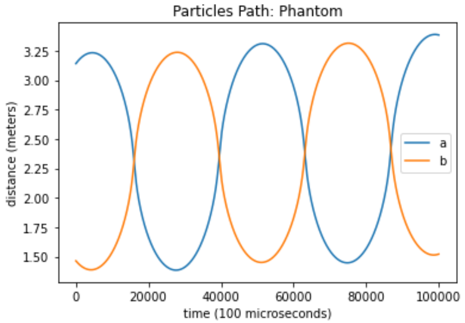




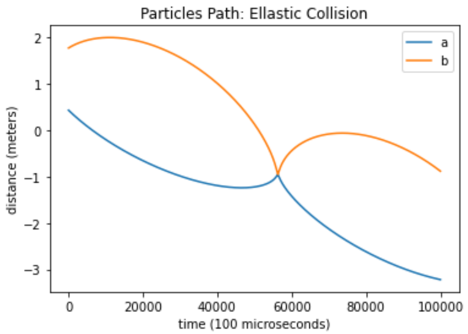


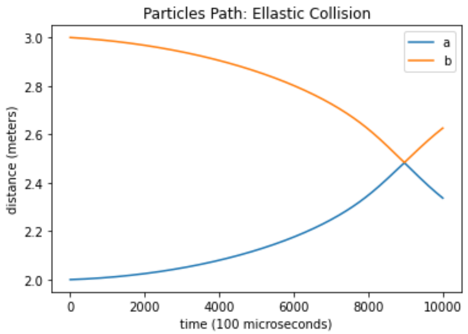


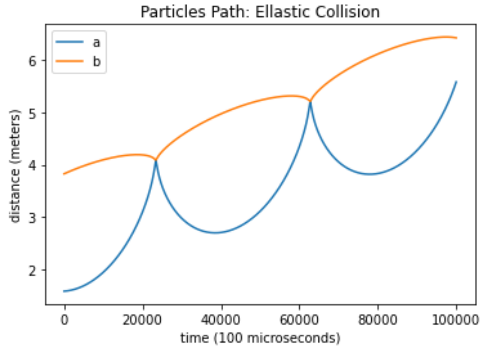


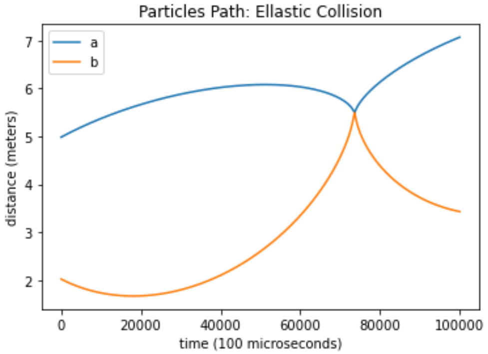


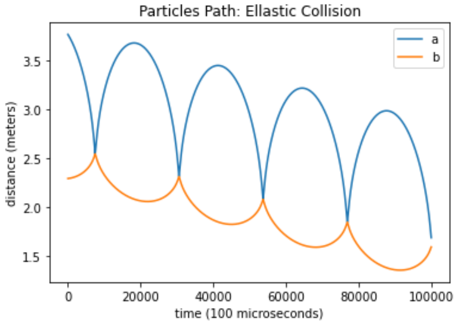
Graph Trajectories: Elastic Collision w/o bounds

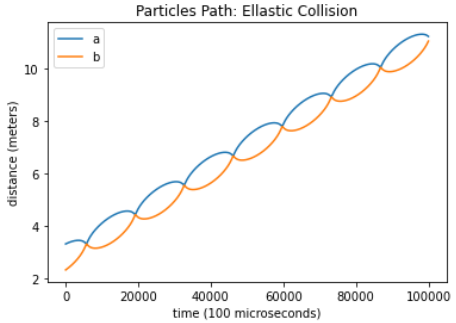






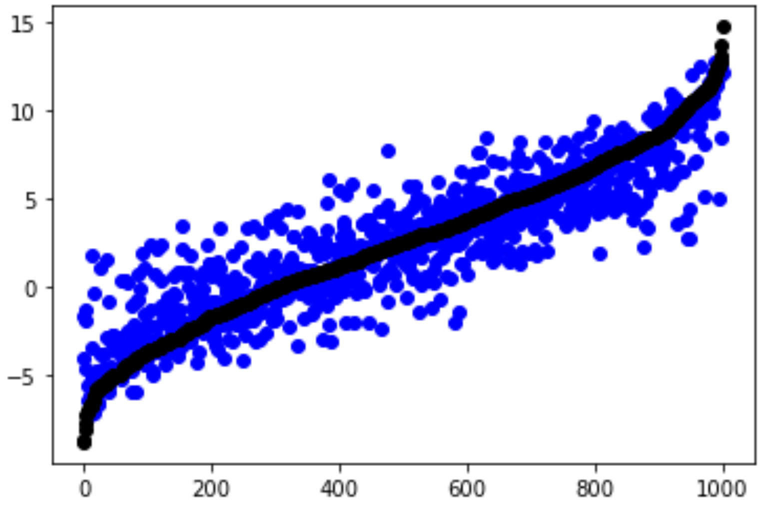






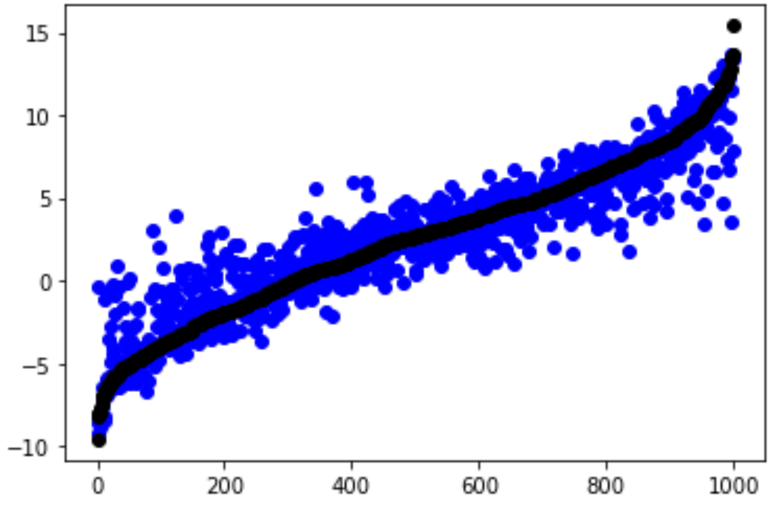
Linear Regression of simulations w/o bounds

Phantom Collision

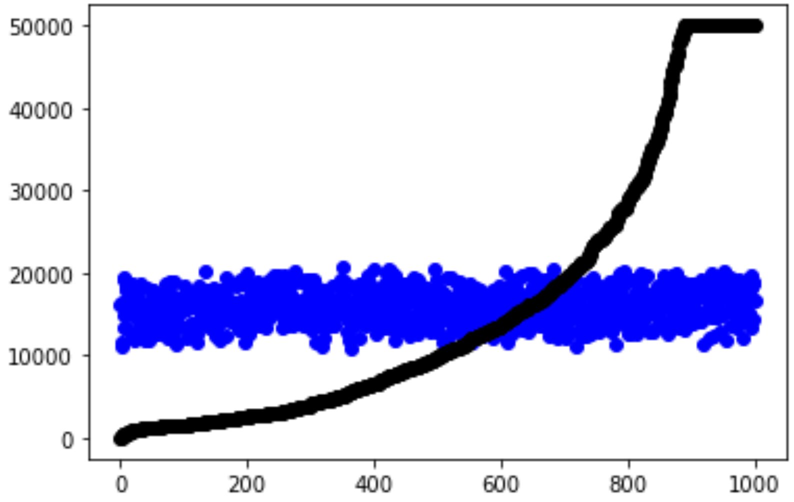


Root Mean Squared Error: 1.90

Elastic Collision



Root Mean Squared Error: 1.64



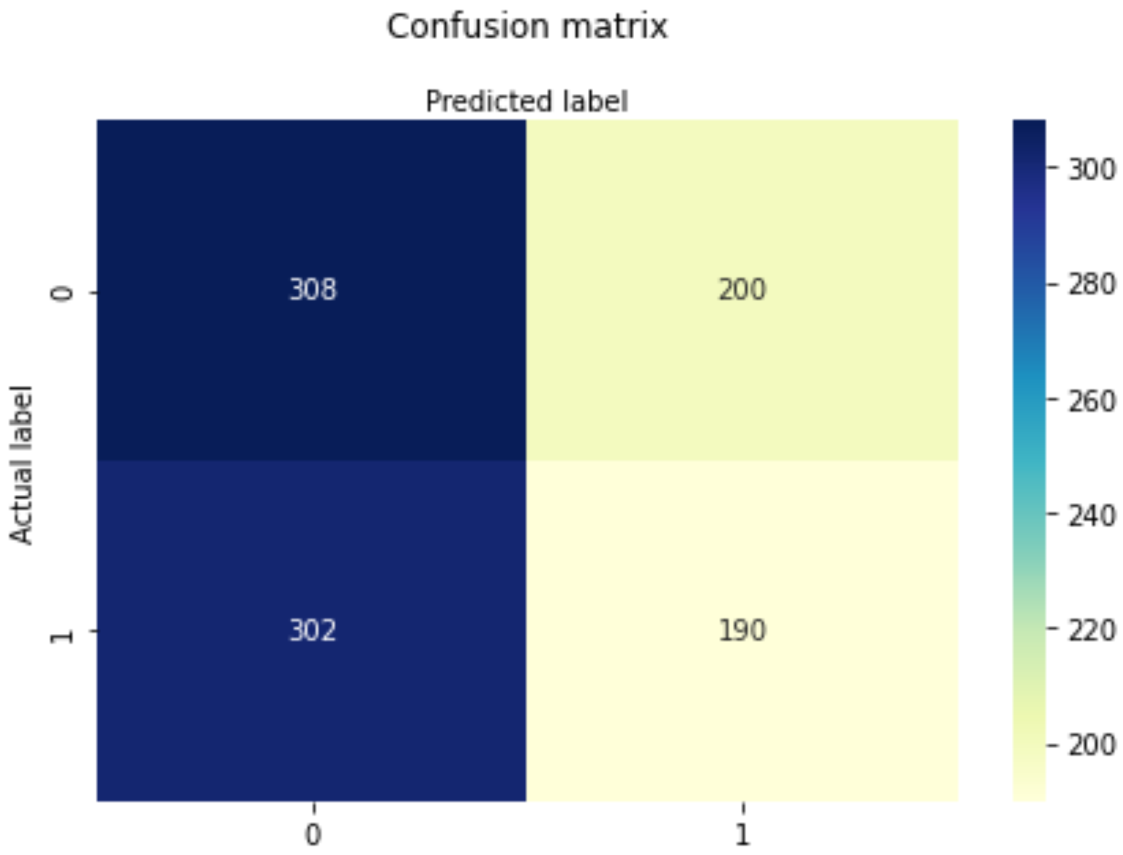
Root Mean Squared Error: 15936

Blue dots = predicted final positions of first particle
Black dots = actual final positions of first particle

Blue dots = predicted collision time
Black dots = actual collision time

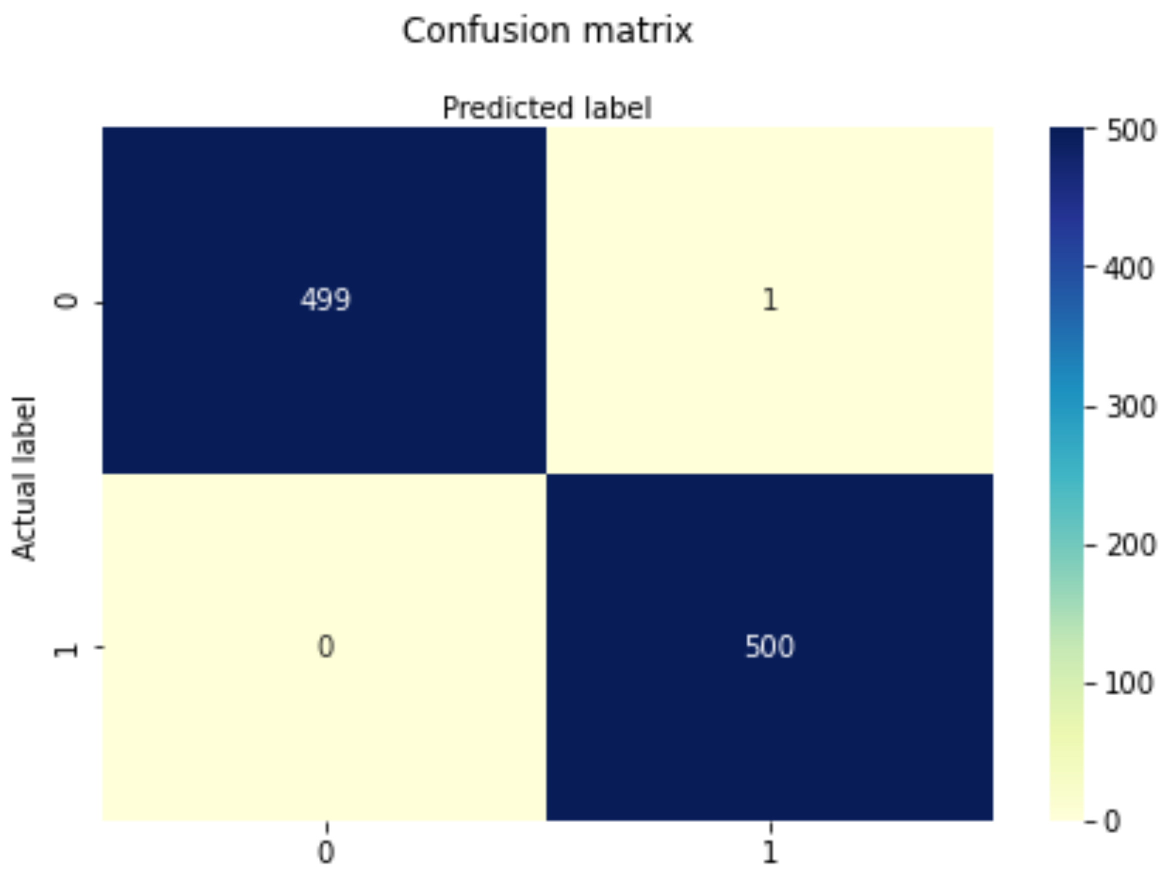
Logistic Regression w/o bounds

Elastic Collision: If particles collide



Logistic Regression less parameters

Elastic Collision: If particles collide



Thank you for your attention!