COMPASS

COMPASS ASSIGNMENT II

September 4, 2025

§1 Before you proceed

These are sections from the book *Computational Physics Problem Solving with Python* you might find helpful:

- 13.5 – 2- and 3-Body Planetary Orbits

§2 Problem 1: Infinity

Given the Figure 8 solution for the three-body problem, where you have the initial positions, velocities and masses for the three bodies:

§2.1 Your tasks

1. Simulate the solution using Forward Euler, RK2 and Velocity Verlet integrators, for T = 200, with timesteps dt = 0.1 and 0.001
2. Plot the orbit of the 3 bodies (See the second session's Jupyter file), and graph energy and angular momentum against time for each simulation
3. Explain why the graphs are different for each simulation (The differences between integrators)
4. In the figure 8 website, you'll find angular momentum and energy for the figure 8 solution, which method was most accurate to the given values?

§2.2 Resources

- To understand how to solve a problem that gives you initial conditions (The N-Body problem is one) Chapter 12: Ordinary Differential Equations (Part 4 - Solving Systems of IVPs)
- Verlet integrator tutorials:
  - Verlet Integration
  - Verlet Integration for Physics Simulations
  - Lecture 3: The Velocity-Verlet Method
- What is the Three Body Problem and How Do You Solve It?

§3 Problem 2: REBOUND Flower

Given the Broucke A12 solution for the three-body problem, where you have the initial positions and velocities for the three bodies, with each body's mass = 1.0:

§3.1 Your tasks

1. Simulate the solution using REBOUND for T = 200 eleven times, the first one should be the same initial conditions, the next ten you should add 0.05 to the x-coordinate initial value of the first body every new simulation (The body in the first simulation will have initial position

(-0.337076702, 0.00), int the second it will have (-0.287076702, 0.00), the third (-0.237076702, 0.00) and so on)

2. Plot the orbit of the 3 bodies (See the third session's Jupyter file), and graph energy and angular momentum against time for each simulation
3. Explain why the graphs are different for each simulation (Search online on the sensitivity of the N-Body problem to the initial conditions)
4. EXTRA POINTS: Use matplotlib to animate the orbit of the 3 bodies and save it as a .gif

§3.2 Resources

- An Astrophysicist's guide on using REBOUND (Her channel is loads of fun, I recommend you check it out!) Learn How to Run a Simple N-Body SImulation (REBOUND Tutorial)
- The official REBOUND tutorials: REBOUND Youtube Tutorials
- Animating matplotlib:
  * Making Animations in Python using Matplotlib!
  * Matplotlib Animations in Python

§4 Problem 3: Newton's law of gravitation

- A probe is placed somewhere on the line joining the centers of Earth and the Moon. At what distance x from Earth's center will the net gravitational force on the probe be zero?
  - $M_{Earth} = 5.97 \times 10^{24} kg$
  - $M_{Moon} = 7.35 \times 10^{22} kg$
  - $distance_{Earth-Moon} = 3.84 \times 10^{8} kg$
  - $G = 6.67 \times 10^{-11} m^3 kg^{-1} s^{-2}$
- Two masses, $m_1 = 2.0$ kg and $m_2 = 3.0$ kg, sit at two vertices of an equilateral triangle with side length a = 0.80 m. A 1.0 kg test mass is at the third vertex; find the magnitude and direction (angle relative to one side) of the net gravitational force on the test mass due to m_1 and m_2.

§4.1 Resources

- Newton's law of gravitation | Physics | Khan Academy
- Newton's Law of Universal Gravitation

§5 Extra Resources

These are optional but recommended:

- · The video says Verlet Integration, but it really covers everything related to integrators: Verlet Integration
- Another method to apply Verlet: Verlet integration with Pygame

- Proving Verlet Integration: Math for Game Developers - Verlet Integration
- Kepler's laws: Kepler's Third Law of Planetary Motion Explained, Physics Problems, Period & Orbital Radius
- A dissertation on the 3-body problem, extremely in-depth and an amazingly fun to read article! It solves the problem in a slightly different way than what we used though. The three body problem | Form and Formula
- A video that solves the 3-body problem with a method similar to the one above. Notes: It uses built in integrators, which is more limited due to it taking specific parameters (Easier to write, harder to configure) Solving the 3-Body Problem in Python!