Simulating Kepler's first law of planetary motion
Introduction to the "leapfrog" method

Fani Dosopoulou and Mark Vondracek

Purpose

Johannes Kepler (1571 – 1630) was a German astronomer who realized that **circular orbits wouldn't work** while investigating the orbital motion of Mars in close detail.

Kepler's first law of planetary motion describes the motion of planets around the Sun:

> The orbit of a planet is an ellipse with the Sun at one of the two foci.

The goal of this activity is to introduce high school physics students to modeling simple physical systems on their computers like the classic two-body problem in Kepler's first law. This is done by numerically simulating Kepler's first law of planetary motion that the students have been already taught in introductory astronomy classes.

The numerical modeling is done in the Python programming language. Learning Python as early as possible gives students a significant advantage when applying for colleges, undergraduate (or even high school) research positions, and ultimately jobs. This activity is given as part of a series of lessons found at my personal website : https://sites.google.com/site/fanidosopoulou/

This project/assignment is intended to introduce the students to the basic techniques of numerical simulations. The students will learn how to turn a system of equations of motion into a set of discrete algebraic equations that a computer can handle and then writing a computer program capable of solving these equations.

This class and project will focus on a system of equations of motion using orbital mechanics as a physical example. The students are given Python code that is mostly complete and are asked to fill in several missing code fragments. For simulating a planetary orbit, students fill in the blanks for a simple energy conserving numerical integrator. We choose the numerical method "Leapfrog" and we implement it numerically using the programming language Python and the environment Canopy both installed in the school computer labs. These consist the two fundamental start-up tools one needs to begin basic numerical coding. This procedure will introduce students for the first time to these tools. The students will then run a simulation of a planet at some distance from the Sun being launched with some velocity. Providing students with a code that misses only small parts allows them to focus more on

Reach for the Stars is a GK-12 program supported by the National Science Foundation under grant DGE-0948017. However, any opinions, findings, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the Foundation.

the simulation and the physics instead of the coding. However, the coding requirements should still pose a challenge to students who have had little exposure to programming.

Overview

1. An introduction to the two-body problem (10 min) given in detail in my personal website: https://sites.google.com/site/fanidosopoulou/
2. An introduction to numerical techniques and the numerical method "Leapfrog" given in detail in my personal website : https://sites.google.com/site/fanidosopoulou/ (10 min).
3. An introduction to basic functions and definitions in Python in case students are not already familiar with the language. (5 min)
4. Students work through the Canopy environment. The students take time to open up their computers, download the files needed and set-up their account in Canopy (15 min).
5. Have students go through all sections of the website https://sites.google.com/site/fanidosopoulou/ . The activity is meant to be as self-guided as possible. The teacher should take this time to walk around the class and help students who are stuck and/or go through difficult sections as a class. This will depend on the programming level of the class.
6. Students that finish early should attempt some of the challenges found in the more advanced material section.

Student Outcomes

Students will be able to learn the fundamental dynamics of the two-body problem.
Students will be introduced to the basic techniques of numerical simulations.
Students will be able to run a simple program in Python.
Students will be able to translate the equations of motion for a planet around the Sun into Python code.
Students will be able to use a computational model of an orbiting planet to simulate different planets of various eccentricities around stars with different masses.

Standards Addressed

K-PS2-1 Motion and Stability: Forces and Interactions

Plan and conduct an investigation to compare the effects of different strengths or different directions of pushes and pulls on the motion of an object.

MS-ESS1-2 Earth's Place in the Universe

Develop and use a model to describe the role of gravity in the motions within galaxies and the solar system.

MS-PS2-2 Motion and Stability: Forces and Interactions

Plan an investigation to provide evidence that the change in an object's motion depends on the sum of the forces on the object and the mass of the object.

MS-PS2-4 Motion and Stability: Forces and Interactions

Construct and present arguments using evidence to support the claim that gravitational interactions are attractive and depend on the masses of interacting objects.

HS-PS2-4 Motion and Stability: Forces and Interactions

Use mathematical representations of Newton's Law of Gravitation and Coulomb's Law to describe and predict the gravitational and electrostatic forces between objects.

Time

The lesson was designed for a 1.5 hour class. Some students may need more or less time.

Level

This activity is geared toward junior and senior level physics/computer science students.

Materials and Tools

1. A computer (or laptop) for every student (or small groups of students). Any type of operating system will work (Windows, Mac OSX, or Linux). However, the lesson will not work on tablets, mobile devices, or chromebooks.

2. Have students (or the IT department) install a Python distribution, either Anaconda or Canopy, on the machines. Note that the lessons are written assuming the students are using the Canopy distribution.

Preparation

1. The instructor should check to make sure the Python installation was successful on all of the computers.

2. The instructor should also be comfortable enough with the activity to help students who are stuck.

Prerequisites

Students (and the teacher) should complete the first few units of Codecademy. This can be done as homework or, ideally, in the classroom. The teacher can also use other resources for learning the Python language.

Background

1. Students should be familiar with the basic dynamics of the two-body problem.

2. Students should also have some basic knowledge of the Python language

Teaching Notes

1. As this is a self-guided activity, the teacher should mostly be helping students who are stuck and introducing the physics of the two-body problem at the beginning of the class.

2. It is important that the students have some exposure prior to Python before starting this activity. The teacher should be sure that students have had time to go through the first few units of Codecademy or a similar learning device.

Assessment

Students will be asked to fill in some simple parts of the numerical code that uses the most basic numerical method called "Leapfrog" to solve the system of equations of motion involved in the classic Kepler problem in orbital mechanics. The students will do this by following the instructions of the assignment that can be found at my personal website:
https://sites.google.com/site/fanidosopoulou/

If students fill in these parts incorrectly they learn though the errors that pop up in the Canopy environment the usual mistakes someone makes while coding. Once they fill in the parts correctly they see a plot that visualizes the numerical solution to the two-body problem. More specifically, if the students successfully completed the activity, they should see when they run the Python program a plot that shows the elliptical orbits of the two bodies as well as the ellipse that describes the relative motion of the planet-Sun system.

More specifically, students will

1) observe the agreement between the expected behavior of the system according to the analytical solution (Kepler's first law of planetary motion predicts an ellipse) and the numerical solution.

2) understand how the two bodies move around their center of mass (in ellipses) as well as how the relative motion of the two bodies is shaped (an ellipse as well). Students will be asked to describe what common characteristic they notice all these elliptical orbits have (the same eccentricity) and why they think this is the case according to the laws of physics. Discussion based on their answers will follow.

Students can submit their completed Python scripts to be checked against the instructor's completed (and working) script.

Additional Information

N/A