AST101: Our Corner of the Universe
Lab 6: Simulating Planets' Orbits (I)

Name:

Lab section:

Group Members:

---

1 Objective

To use a computer simulation to explore the properties of planets' orbits, and in the process learn a little about how computers simulate things.

2 Introduction

As we've discussed in class, the development of orbital mechanics proceeded in two steps. First, Johannes Kepler worked out the correct *shapes* of the orbits of the planets, but didn't know *why* they moved in that way. Later, Isaac Newton discovered the laws of gravity and motion that explain *why*, not just *how*, the planets move in elliptical orbits.

In this lab, you will work directly with a computer program that contains within it Newton's laws of motion and gravity. You can use this program to simulate all kinds of orbits! In this lab, you will use this program to simulate different kinds of orbits and see how they behave.

This lab will also expose you to a little computer code, written in the language Python. You do not need to know anything about computer programming beforehand. However, one step in this lab requires you to look at some computer code and think about what it does. We're not going to require you to *write* code, but we want to give you a little exposure to what it looks like. Python is a language used by many scientists and web developers (and other people) for all kinds of things; it is one of the most popular programming languages today.

2.1 Kepler's laws of planetary motion

First, Kepler realized that if planets moved in **elliptical** orbits rather than circular ones, their movements fit the observations very well. He formulated three laws of orbits that describe the motion of the planets. Paraphrased, they are:

1. Planets move in elliptical orbits with the Sun at one focus
2. The line connecting a planet to the sun sweeps out equal area in equal time, so planets move faster when they are closer to the Sun, and slower when they are further away
3. The time it takes a planet to orbit the Sun depends only on the length of its semimajor axis. In particular, if the semimajor axis is multiplied by a factor A, the period P is multiplied by $A^{3/2}$.

2.2 Newton's laws of gravity and motion

Newton realized several things about moving objects:

1. Forces cause objects to accelerate by an amount $a = F/m$ in the direction of those forces.
2. Gravity is one such force. All objects attract each other with a force equal to

$$F_{\text{grav}} = \frac{Gm_1m_2}{r^2}$$

   where r is the distance between the two objects, and m_1 and m_2 are their masses.

2.3 Computer programs

You won't need to write your own code here – only modify my code to see what happens, and then look at some parts of it to see how it works. You can't break anything; if something goes wrong, you can just click the three-lines menu in the top left and choose "Reset".

If you press "Run" and your code gives you an error (a big red bar at the bottom), either hit Ctrl-Z to undo your last few changes and then try again, or refresh your browser. Your GTA might be able to assist you as well (most of the graduate students will know some Python).

Computers, at heart, do *math* on *variables*. Look at line 5: this assigns the value `1.017` to the variable `start_distance`. A computer program is just a list of math instructions in order, along with other instructions that do things like draw on the screen and repeat some segments of math.

It is important that you don't change the names of any of the variables here: if you rename `start_distance` to `begin_distance`, even though those mean the same thing in English, the computer won't know what is going on later.

It is also important that you don't add extra spaces before the beginning of any lines, or remove any that are there. Spaces before a line mean something special in Python, and if you remove them, the program will mean something different than it did before (and probably won't work).

Also, notice the text that's displayed in green and follows a # sign. These are *comments* – the computer ignores them. They're just in the code to give you "signposts" to describe what parts of the code do.

3 Exploring Kepler's laws

Question 1. Go to the orbit simulator and set it to fullscreen. Then run the simulator. You've played with it before for your prelab, but now we're going to make sure that our computer program's orbits follow Kepler's laws.

Kepler's first law says that planets orbit the Sun in elliptical orbits with the Sun at a focus. Does the orbit that you see follow Kepler's first law?

Question 2. Now, try different starting velocities for the planet, and sketch the planet's orbit around its star. Notice that the starting point (1.02 AU) will always either be perihelion or aphelion. For each of the following, record:

- A picture of the planet's orbit, labeling its starting point, and mark where aphelion and perihelion are
- How close, or how far, the planet gets to the Sun on the other side (the computer will tell you this; it's the other perihelion/aphelion point)
- The eccentricity of the planet's orbit
- Whether anything weird happened that wasn't at all what you expected

If your planet moves too slowly or too quickly, change the value of `seconds_per_year` that controls how many seconds in the simulation corresponds to one year of real time.

Try starting velocities of

1. 2.5 AU/year

2. 5 AU/year

3. The value you found during your prelab that matches Earth's actual orbit

4. 8 AU/year

5. 10 AU/year

Question 3. What happened when you gave the Earth a large starting velocity of 10 AU/year? Is this what you expected? Do you think that the Earth would really move in this way if its velocity were suddenly increased to 10 AU/year?

Question 4. Remember that the focus of an ellipse is close to the center of the long axis if the ellipse is not very eccentric, but close to the edge if it is strongly eccentric. Look back at your drawings for the previous question. Is this true in the orbits you observed?

Question 5. Now, let's look at Kepler's second law, which compares *the speed of a planet* at *different places in its orbit*. By changing the starting distance from the Sun and the starting velocity, put your planet into an orbit with eccentricity at least 0.6.

Does your planet move more quickly near perihelion or aphelion? In evaluating this, look at both the animation and the graph to make sure that they agree. How does your graph show this?

Question 6. Does your planet spend more time near perihelion or aphelion?

Question 7. Does your planet appear to follow Kepler's second law? How do you know?

Now, let's look at Kepler's third law, which compares *the time that different planets take to go around the Sun*.

Kepler's original formulation of the third law says that

$$\frac{(\text{orbital period})^2}{(\text{length of long axis})^3} = \text{constant}$$

This means that if you calculate this fraction for any planet orbiting the Sun, it should be the same.

Question 8. How can you determine the length of the long axis if you know the perihelion and aphelion distances? (A diagram on scratch paper may help you figure this out!)

Question 9. Simulate two different orbits with different starting distances and velocities. (Remember, if your planet flies off, your starting velocity is too high!)

Aphelion	Perihelion	Long axis	Orbital period	$\frac{(\text{orbital period})^2}{(\text{long axis})^3}$

Do your two orbits appear to follow Kepler's third law? (You won't need to square and cube things on the exam, but you can do it here!)

4 Does the Mass Matter?

The mass of the planet is set to the mass of Earth by default: 0.000003 times the mass of the Sun. Change the mass of the planet to the mass of Jupiter: 0.001 times the mass of the Sun.

Question 10. Did the planet's orbit change? Describe how it changed, or explain why it didn't.

Question 11. Now, change the *star's* mass. Does this affect the orbit? Why or why not?

Kepler's description of planet orbits – where the Sun doesn't move and the planet orbits around it in an ellipse – are based only on the motions of the planets in our solar system. They are true only if:

- A much less massive object orbits a much more massive one (like Earth around the Sun)
- The gravity of the larger object being orbited is the only force that causes much of an effect on the smaller object.

Earlier, you saw that changing the mass of the planet orbiting the Sun doesn't affect its orbit. But what happens if you make it *very* massive – almost as massive as the Sun itself?

Question 12. Increase the mass of the planet to at least ten percent of the star's mass. What happens *now?* Is this what you expect?

Question 13. Is this behavior something that matches Kepler's description of planetary orbits? If it's not, which of the assumptions above are no longer true?

Question 14. Suppose that you looked in space and saw a star wiggling back and forth. We can see stars with telescopes, but can't see planets since they don't make their own light. If you saw a star wiggling back and forth, what could you conclude about it?

Question 15. We know that many stars are *binary stars* – there are two stars that orbit each other closely. Since we know that gravity affects stars in the same way as it affects planets, we can use this program to simulate binary stars too.

Set the mass of the "planet" to nearly the same as the mass of the "star": both of them will now represent stars. See if you can create an orbit where the two stars orbit each other. Show your orbit to your TA when you've got it.

5 How's This All Work?

This program may look complicated. But most of the code has to do with making the animation and table you see! The only code that actually performs the simulation – where the actual *physics* is – is between line 50 and 65.

All of the laws of nature that control the motion of orbits are the following:

- Forces cause things to accelerate, where the acceleration is equal to the force divided by the mass of that thing **(Newton's second law of motion)**
- Any two objects attract each other with a force equal to $F = \frac{Gm_1m_2}{r^2}$, where the force is directed toward the other object **(The law of gravity)**
- Forces come in pairs: if the star's gravity exerts a force on the planet, the planet's gravity exerts a force back on the star that goes **(Newton's third law of motion)** in the opposite direction and has the same size
- In a small interval of time, an object's position changes by an amount equal to its velocity multiplied by the amount of time. (You know this already: if you are traveling at 50 miles per hour, and you do this for two hours, you will go 100 miles...) **(Definition of velocity)**
- In a small interval of time, an object's velocity changes by an amount equal to its acceleration multiplied by the amount of time. **(Definition of acceleration)**

The computer simulation in this program works as follows. First, we divide time up into very tiny intervals (the length is given on line 40, if you want to change it). Then, repeat the following:

1. Determine the distance from the planet to the star
2. Determine the *direction* from the planet to the star
3. Using Newton's law of gravity and the distance and direction you just found, determine the force the two objects feel from each other
4. Using Newton's second law of motion, determine the acceleration that this causes on each object
5. Using the definition of velocity, change the positions of the star and the planet by the amount that they will move during the small interval
6. Using the definition of acceleration, change the velocities of the star and the planet by the amount that they will change during the small interval

This is all it does!

Question 16. Look at the code from lines 50-65. Look at the code and try to identify what each group of lines does. Write those line numbers by the steps in the above list.

Question 17. Newton's third law in this context says: *"If the star applies a force to the planet, then the planet applies a force to the star that is of equal size, but in the opposite direction."* Look at the computer code that calculates the gravitational forces on the planet and on the star. Does this simulation appear to follow Newton's third law? How can you tell? (How do you say "the opposite direction" to a computer?)

Question 18. As you saw earlier, this simulation has all of the features that Kepler observed in the orbits of the actual planets. However, Kepler's laws don't appear *anywhere* in this code; there are no mentions of ellipses, long and short axes, moving faster near perihelion, or anything like that!

You've already seen how this program works. Why do you think it correctly simulates the features of the planets' orbits, even though it doesn't have any of the substance of Kepler's laws inside it?

Call your TA over to join in your discussion. Ask them to sign your paper after talking to you.