Kepler's Laws: Universal or Not?

Jake Ribich

12 December 2016

Abstract

While it may seem that the heavens are an unorderly place, each and every planetary body follows the same set of patterns. These patterns were at one point only conjectured, but they now are considered universal laws. This project focuses exclusively on the planets within our solar system.

Kepler's first law states that the orbit of a comet or planet is always a planar conic in shape. Whether an ellipse, hyperbola, or parabola, each planetary motion can be modeled and predicted by the mathematics of each type of path. Kepler's second law is an extension of the first in the fact that the shape of the orbit is still defined, but is also dependent on the line segment connecting the planet to the sun. It states that this line segment sweeps over equal areas in equal periods of time. Finally, the third law only holds true for objects with closed, or elliptical, orbits. This final law relates the square of the orbital period to the cube of the semimajor axis of the orbit. Every planet within our solar system follows a closed orbit and we can therefore illustrate the third law with ease.

1 Background and Introduction

Before the year 1543, it was accepted that the earth was located at the center of the universe. Based partially on scientific observation and partially on religios theory, this claim went virtually unquestioned for thousands of years. However, in 1543 Nicolaus Copernicus published the extremely controversial idea that this claim was untrue. He proposed the idea of heliocentricity, or the notion that the universe was centered around the sun, not around the earth. Between the years of 1609 and 1619, Johannes Kepler published a work which would strongly support this idea of heliocentricity. The work contained what would later be called *Kepler's Laws of Planetary Motion*, and went on to explain and describe the motions of the heavenly bodies. [3]

2 Goals and How They Were Met

My goal with this project was to model these three laws, illustrating the motions of the planets while also mathematically proving them. I chose to use Vpython for all of my programs because of how object-oriented it is, as well as how user-friendly it is towards those with little to no programming experience. My original plan was to create one program to illustrate each of the three laws, and then to explain the math behind them separately. Over time, however, I found it difficult to create a program that would demonstrate the third law due to the mechanics used for creating orbits in Vpython. Because of this, I made the decision to make two different versions of Kepler's second law instead of attempting to create a working third law. I still cover the underlying mathematics and theory and also go into a little more detail on the difficulties that I faced.

3 Kepler's First Law

This law states that a planet revolves around the sun in an elliptical orbit with the sun at one focus.

3.1 Underlying Theory and Math

After finding the equation of the orbit, proving this first law is actually relatively simple. The derived equation of orbit written in polar coordintes is:

$$\frac{1}{r} = \frac{1 + e\cos(\theta)}{p(1+e)}$$

In this case, p represents the perihelion which is the smallest value of r, or the nearest that the planet gets to the sun. We can then manipulate this equation of orbit in the following way:

$$p(1+e) = r(1 + e\cos(\theta))$$

$$p(1+e) = r + ex$$

$$r = e[p(1 + \frac{1}{e}) - x]$$

[4] Where r is the the planets distance from the sun, and the expression in brackets represents the the perpendicular distance from the line D, the directerix, whose x coordinate is $x = p(1 + \frac{1}{e})$.

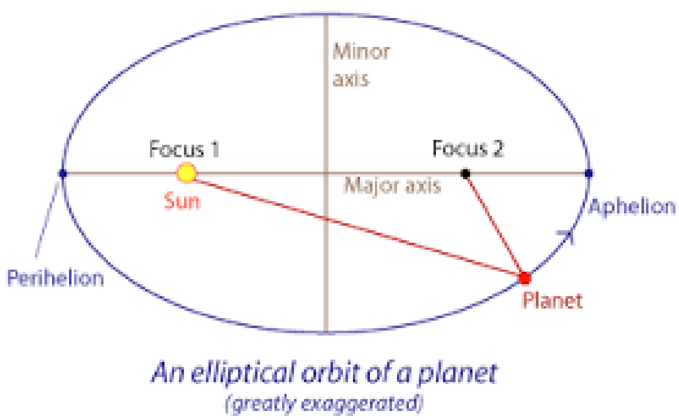

Figure 1: This figure contains a few the mentioned values, but because it is an elliptical orbit the directerix D would be represented as a vertical line to one side of the image. The red line connecting the sun to the planet is r and you can see the perihelion p labeled on the left.

This affirms that the planet lies on a conic shape with a focus at the origin(the sun) and an eccentricity of e.

3.2 Program Details

This model is relatively simple in the sense that only a few objects need to be constructed at any given time. Four examples configurations are illustrated, each using the same sun but with two different of planets of varying size and distance from the sun. The program asks for user input in the form of a button press, and this input is used to decide which planets or combination of planets will travel around their orbits. VPython allows for the easy creation of each of the celestial bodies. After bestowing the size, position, and physical features of each object, the universal law of gravitation is used to determine the vector quantities of the forces that are exerted. Updating each object's momentum by applying this force at various points in time, the movement of the planets around the sun can be demonsrated with relative scale.

4 Kepler's Second Law

This law states that the line segment joining the sun to a planet sweeps out equal areas in equal times.

4.1 Underlying Theory and Math

In order to determine the area swept out by the line segment, we can treat the area as a triangle. The line segment itself makes up the larger leg, the change in position makes up the smaller leg, and the hypotenuse connects the planets new position to the sun. While this may seem inaccurate due to the fact that the smaller leg will be a curve and not a straight line, we can change this by making the movement of the planet infinitesimally small. By doing this, the shape of the area approaches that of a triangle, and this is what we will use for the calculations.[1] The change in area can then be represented by:

$$dA = \frac{1}{2}r^2\Delta\theta dt$$

Where r is the distance from the sun to the planet, and $r\Delta\theta$ is the change in position of the planet. The areal velocity, or rate at which area is swept out, we then define as:

$$K = \frac{1}{2}r^2\Delta\theta$$

Differentiating with respect to time:

$$K' = \frac{r}{2}(r\theta'' + 2r'\theta')$$

$$K' = 0$$

[4] Previous calculations have allowed us to determine that the quantity in parenthesis has the value of 0, and therefore K is a constant. This means that area is swept out at a constant rate so equal areas are swept in equal times. Here is a visual in order to better understand the mathematics:

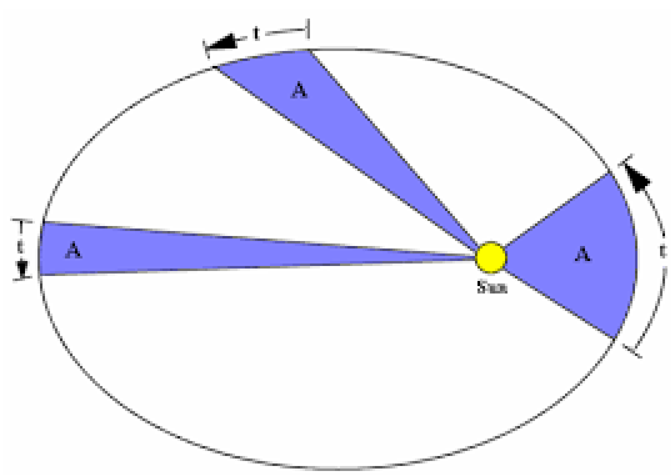

Figure 2: As stated previously, it takes the same amount of time t for the planet to make each of the three movements. Because of the varying force experienced by the planet, and therefore the varying speed at which it travels, the area swept out by all of these movements is equal to the same amount A

4.2 Program Details

The program illustrating Kepler's second law takes advantage of the make trail function as well as the curve object in order to show the amount of area covered over acertain period of time. The orbit of the planet is traced out by this trail function, and at regular points in time a curve is drawn from the planet to the sun. During each equal period of time, the color of these curves changes in order to symbolize the new area being swept out. In the first version of this program, the amount of time periods and initial speed are displayed. The areas of each section must be determined just by looking at them and reasoning out that they are equal.

The second verion of the program is almost identical to the first in terms of the visual that is displayed. It allows the user to select from a few different orbital paths, which then produce the same trail and curves as before. The only visual difference is that instead of displaying the initial speed, the program will display the calculated area of each section of the orbit[1].

5 Kepler's Third Law

This law states that the square of the period of revolution of a planet is proportional to the cube of the length of the major axis of its orbit.

5.1 Underlying Theory and Math

Using the equation of the orbit that was derived for the first law, we can determine that the semiaxis in the x direction, denoted by a, is equal to the value $\frac{p}{1-e}$, which can then be rewritten as $p = a(1-e)$. Through substitution and manipulation[2], we arrive at the equation of the orbit in rectangular coordinates:

$$(\frac{x+ae}{a})^2 + (\frac{y}{a\sqrt{1-e^2}})^2 = 1$$

Newton's law of universal gravitation can be related to the perihelion (minimum value of r) in the following way:

$$p = \frac{4K^2}{GM(1+e)}$$

[1]These calculations were done by hand for the same reasons that are explained in the program difficulties section of the third law

[2]This manipulation consists of drawn out algebraic computations and simplifications that are not necessary in order to understand the fundamental theory behind this law

This is later manipulated[3] until we arrive at a relationship that supports, and actually is, Kepler's third law:

$$T^2 = \frac{4\pi^2}{GM}a^3$$

[4]

5.2 Program Difficulties

My original plan was to re-use a majority of the code from the first law as the foundation for this law. This way, the same visual representations would havel been in effect, however, a display or print function would have been added in order to display both the length of the major axis as well as the period of the orbit. The program was supposed to allow the user to input the desired length of the major axis. Using this input in its calculations, the period of revolution would then be displayed alongside the user's original input and the visual representation would be adapted in order to accurately model the result.

The problem with this plan was the unexpected difficulty in calculating the length of the major axis. Creating orbits in Vpython is relatively simple, but analyzing the exact shape of this orbit proved to be much harder. This is because in order to set up a successful orbit, the original positions and velocities of the planetary bodies must be defined as well as the force acting between the two bodies. With only this information, the size of the axises and the location of the center of the elliptical orbit cannot easily be determined without a trial and error sort of approach. This means that in order to create a program like I orginally wanted to, I would have had to use trial and error and hand calculations for every different length of major axis, which would have taken an unreasonable amount of time for the small amount of results that would have been produced

[3]Refer to the previous footnote

References

[1] Kepler's Three Laws . (2016). Physicsclassroom.com. Retrieved 13 December 2016

http://www.physicsclassroom.com/class/circles/Lesson-4/Kepler-s-Three-Laws

[2] Kepler's Laws. (2016). Hyperphysics.phy-astr.gsu.edu. Retrieved 13 December 2016

http://hyperphysics.phy-astr.gsu.edu/hbase/kepler.html

[3] Kepler's laws of planetary motion. (2016). En.wikipedia.org. Retrieved 13 December 2016

https://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion

[4] Gavin R. Putland: A self-contained derivation of Kepler's laws from Newton's laws. (2016). Grputland.com. Retrieved 14 December 2016

http://www.grputland.com/2013/12/self-contained-derivation-of-keplers-laws-from-newton html