

Kepler's First Law

November 25, 2017

Kepler's First Law

Kepler's first law is an empirical observation that the planets move in elliptical orbits. Newton later proved, using his form of geometric calculus, that in all elliptical orbits, the force that attracts the orbiting body to the foci of the ellipse is the inverse-square of the distance. Newton also showed that the reverse of this is true, that orbiting bodies governed by an inverse-square force follow an elliptical path. This project intends to demonstrate, that given a moving body and an inverse-square force towards a central object, the body will follow an elliptical path.

Firstly, we import some needed libraries and set up the environment for plotting.

```
In [1]: %matplotlib inline
```

```
In [2]: import numpy as np; import matplotlib.pyplot as plt
```

We need to create two lists in order to store the data that will end up getting plotted.

```
In [3]: x = []; y = [];
```

Now we will set some initial variables to determine how this simulation will behave. Feel free to change these values in order to get a different final result. 'bheight' is how far above the center the body will start. 'speed' is how fast the body will be going horizontally. 'steps' is the number of updates the simulation will churn through. 'c' stores the coordinates for the center of the orbit. 'body' stores the coordinates for the body's initial position. 'v' is the initial velocity of the body.

```
In [4]: bheight = -254.0 - 3959
        speed = 4.76
        steps = 1000
        c = (0.0, 0.0)
        body = (0.0, bheight)
        v = (speed, 0.0)
```

Now it is time to begin actually simulating the orbits. The general idea is that the forces on the body will be calculated, and then the body will be moved according to those forces. This will happen for n steps. After each step is completed, the resulting position of the body is stored into the x and y lists.

```
In [5]: # add the initial positions to the arrays
        x.append(body[0])
        y.append(body[1])
```

```

# take n steps,
for i in range(1,steps):

    # find the distance the body is from the center using the pythagorean theorem
    dx = c[0]-body[0]
    dy = c[1]-body[1]
    d = np.sqrt(dx*dx + dy*dy)

    # calculate the force acting on the body using the inverse-square law
    f = 1/d*d

    # find the force vector towards the center of the orbit
    direction = (dx, dy)

    # get the length of this vector so that we can make it unit length
    direction_length = np.sqrt(direction[0]*direction[0]+direction[1]*direction[1])

    # normed direction
    direction = (dx/direction_length, dy/direction_length)

    # now we can just multiply the unit length direction vector by the length of force
    force_vector = (f*direction[0],f*direction[1])

    # update velocity of the body
    v = (v[0]+force_vector[0],v[1]+force_vector[1])

    # update the body's position
    body = (body[0]+v[0],body[1]+v[1])

    # record the position
    x.append(body[0])
    y.append(body[1])

```

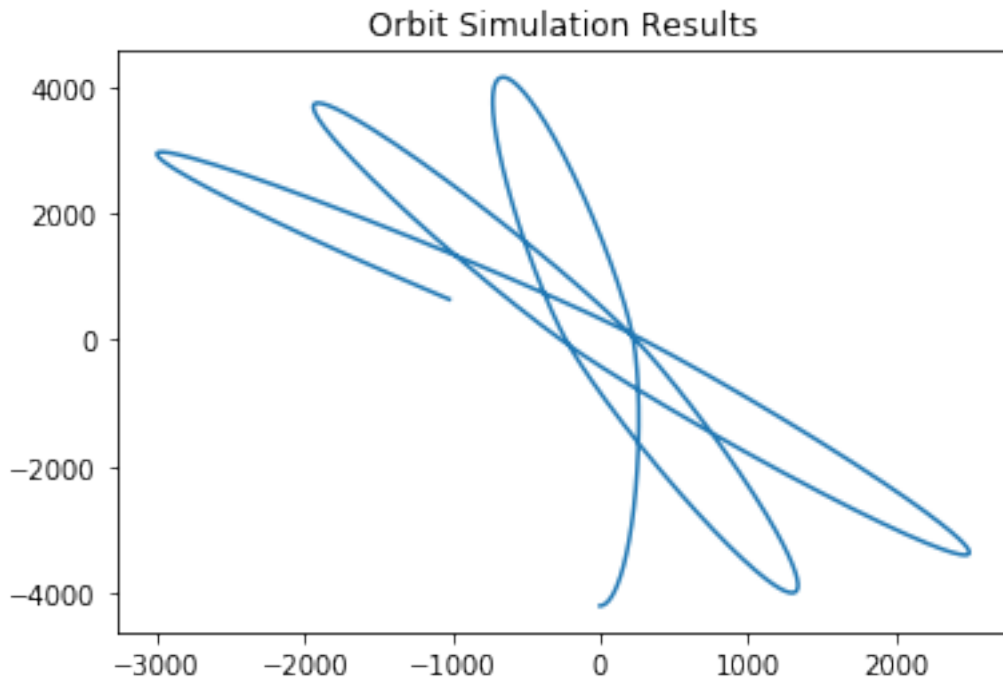
Now that the simulation has been performed, it is time to plot the data. However, we need to first convert our lists into numpy arrays.

```

In [6]: x = np.array(x)
        y = np.array(y)

In [7]: plt.plot(x,y)
        plt.title('Orbit Simulation Results');

```



As can be seen from the resulting plot, the orbits appear to follow an elliptical path. This means that, using Python, we have verified Kepler's and Newton's findings. To get a different orbit pattern, try altering some of the initial variables.