Simulating Proposed Geocentric Models of the Solar System

Daniel Zhang

Introduction

In this project, drawing inspiration from [2], I simulated 3 different proposed models of the solar system: the Ptolemaic, Platonic, and Brahe models. These are all geocentric models or models where the Earth is in the center [5]. The goal is to see the aftermath of simulating the solar system for some time, and whether or not the systems completely break down. The geocentric models will be compared to a regular solar system simulation, and plots of the trajectories and separations from the central mass will be used to determine whether or not the system is held together.

Methods

We will be using the n-body lab code provided by Dr. Slinker to model the systems. The initial positions of the Sun to Pluto will be reorganized from a dictionary of objects and positions to a dictionary of order and positions. The initial velocities are also reorganized by order, and the masses of the objects will stay the same. The Ptolemaic model and Platonic model are simulated this way, while the Brahe model assigns initial positions based on known distances from the Sun, and initial velocities are calculated using Kepler's Law.

Solar System Simulation

As you can imagine, this simulation of a regular solar system is pretty intuitive. The order of celestial objects is as follows from the center to the outermost object: Sun, Mercury, Venus, Earth, Moon, Mars, Jupiter, Saturn, Uranus, and Neptune. The trajectories of the planets are shown in Figure 1 and are expected, as the planets are traveling in elliptical orbits.

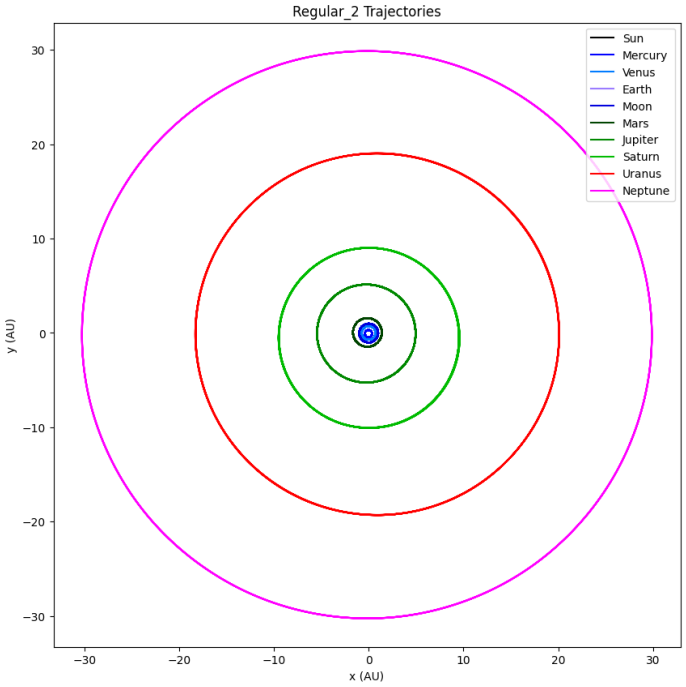

Figure 1: Solar System Trajectories (dt = 0.005, tmax = 500)

Figure 2 shows the separations of the celestial from the center object/mass, the Sun. All objects display sin wave-like separation lines, showing their elliptical orbits around the Sun. None of them are constantly increasing or decreasing, showing no collision or slingshot of any objects.

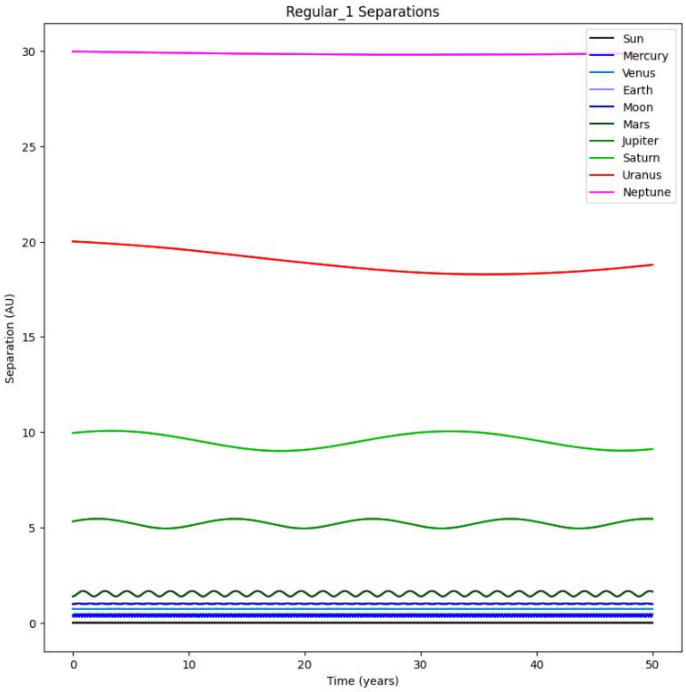

Figure 2: Solat System Separations (dt = 0.005, tmax = 500)

Ptolemaic Model

Proposed by the Roman astronomer and mathematician Ptolemy Claudius Ptolemy in the second century AD, this is the first geocentric orbit we will model. His model had the Earth in the middle of the solar system, which didn't move, and the celestial objects in the following order, from closest to farthest from the Earth: Moon, Mercury, Venus, Sun, Mars, Jupiter, and Saturn [1]. In this model, the planets travel in smaller circles called epicycles, whose center *P* was the circular orbit around the Earth, called the deferent. His big idea for the model was that the planets traveled around the deferent at different speeds instead of the constant speed stated by the Principle of Uniform Circular Motion [1].

The first simulation, shown in Figures 3 and 4, used dt = 0.00005 and tmax = 1 to find any immediately noticeable trajectories. Most of the objects seemed to follow generally elliptical orbits, but immediately we can see that Venus has shot out of orbit, as we see a large, constant increase in separation and non-elliptical trajectory.

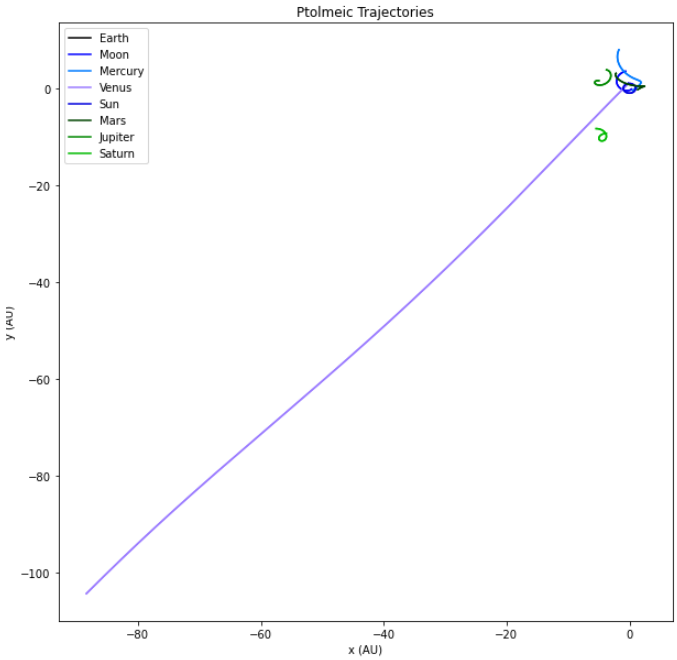

Figure 3: Ptolemaic Trajectories (dt = 0.00005, tmax = 1)

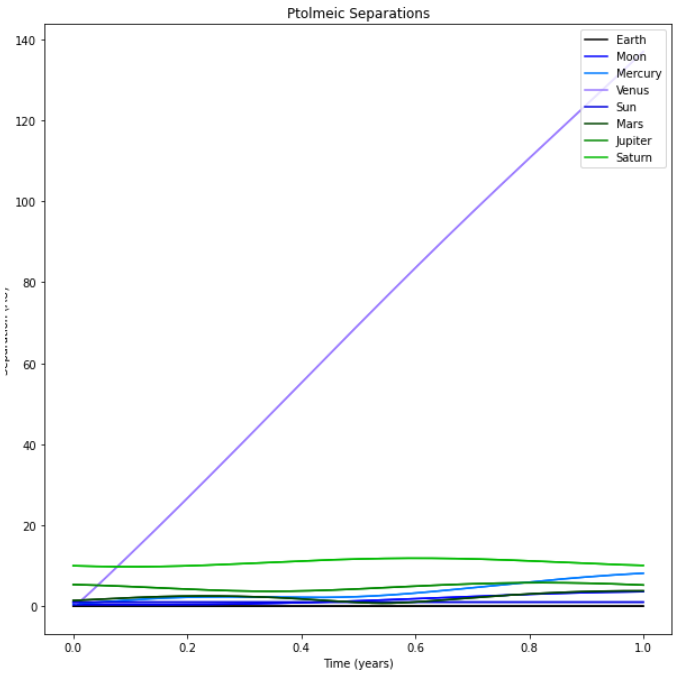

Figure 4: Ptolemaic Separations (dt = 0.00005, tmax = 1)

Mercury also seemed to have a rather large elliptical orbit that may have begun to shoot out, so tmax was increased to 50 in the next simulation. As shown in Figures 5 and 6, we can see that Mercury indeed began to shoot out of orbit, as the separation constantly increases, and the trajectory is no longer elliptical. Although not as dramatic as Venus, Mercury still shot out of its elliptical orbit.

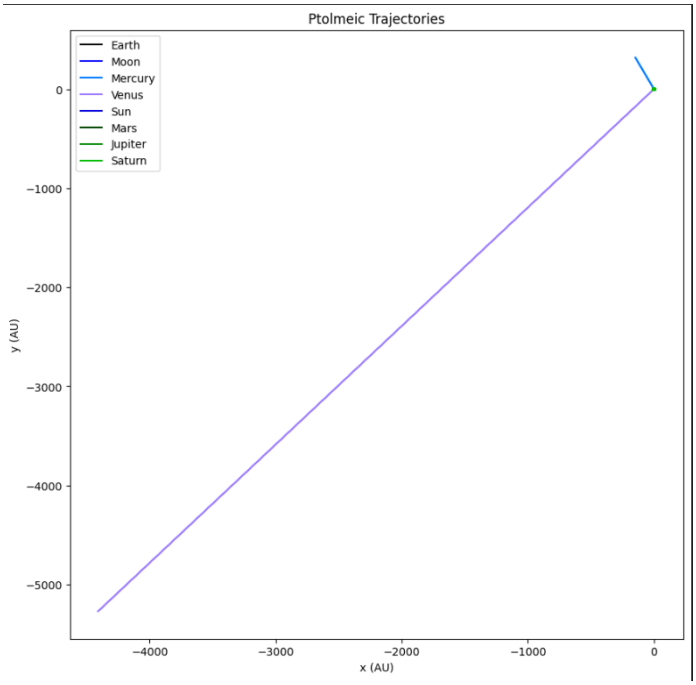

Figure 5: Ptolemaic Separations (dt = 0.00005, tmax = 50)

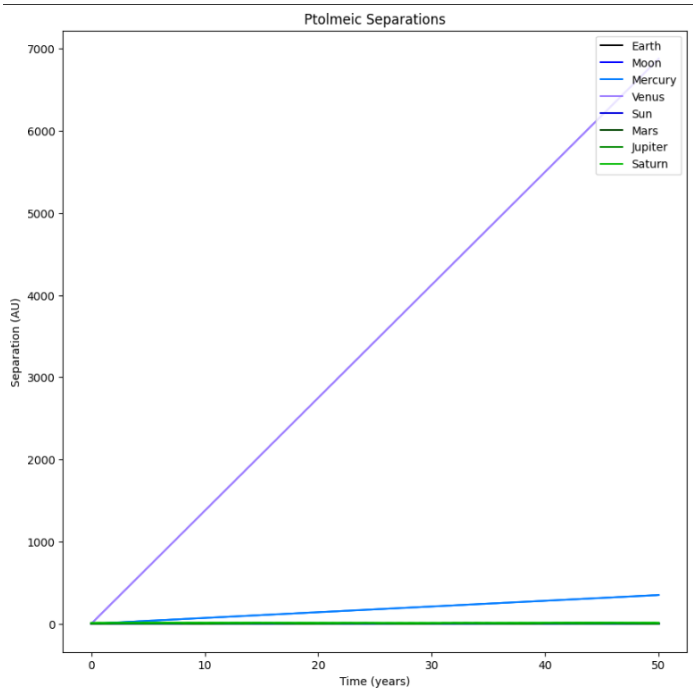

Figure 6: Ptolemaic Separations (dt = 0.00005, tmax = 50)

Platonic Model

Plato, an ancient Greek philosopher, proposed his model in the 4th century BCE, a geocentric model that was the dominant ideology in cosmological theory until the 1600s when Copernicus proposed the first heliocentric model. Plato believed that Earth was stationary at the center of the universe, and the stars and planets orbited in circles around it. The order of the celestial objects from closest to farthest was the Moon, Sun, Venus, Mercury, Mars, Jupiter, and Saturn[5].

At first, I began with dt = 0.0005 and tmax = 10 to find any noticeable patterns or deviations. This was different from the previous simulation where I started at tmax = 1, because there were no notable patterns with that small of a tmax. As shown in Figure 7, we can see right away that Mercury and Venus have already shot out of orbit, while the rest of the objects are maintaining an elliptical orbit. All of the celestial objects seem to have these mini half-orbits while they go through the whole orbit, which is shown by the squiggles in the elliptical orbits.

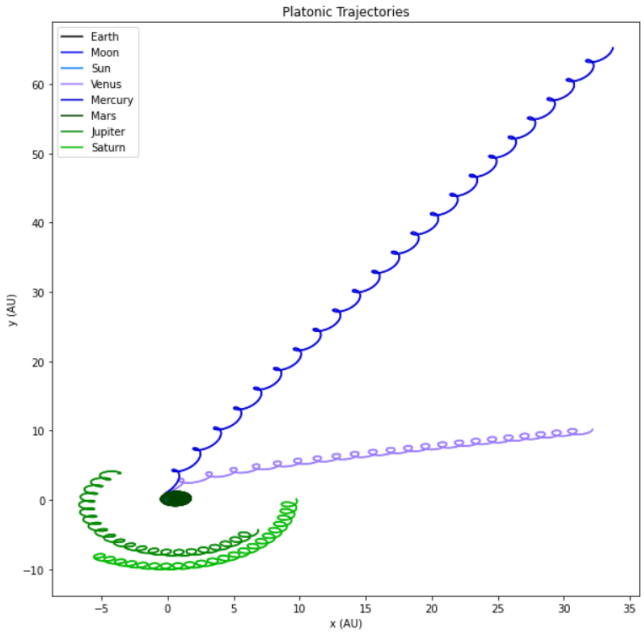

Figure 7: Platonic Trajectories (dt = 0.0005, tmax = 10)

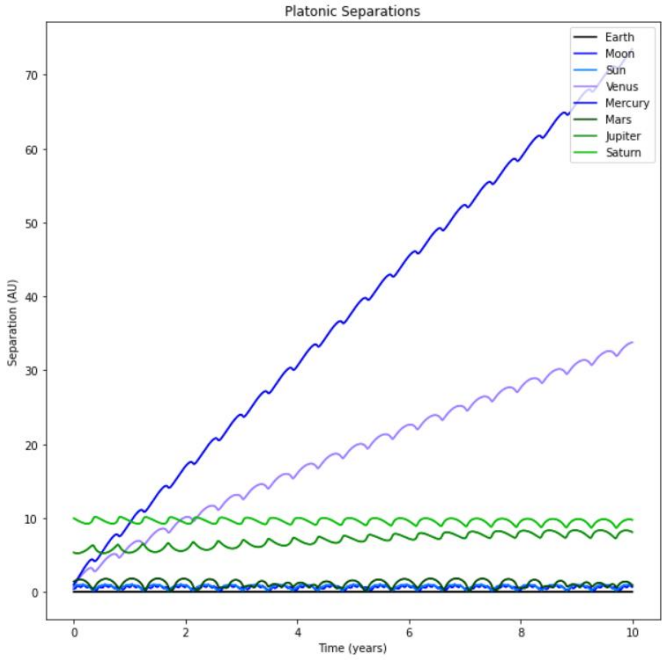

Figure 8: Platonic Separations (Dt = 0.0005, tmax = 10)

Brahe Model

The last geocentric model I modeled was proposed by Tycho Brahe, a Danish astronomer, in the late 60th century. He also believed that the Earth was at the center of the universe, but he also believed that the Copernican model had some truth to it. His model has the Earth in the center of the universe, with the Moon and Sun orbiting around the Earth [4]. However, he had the rest of the planets orbiting the Sun instead of the Earth.

To model this system, initial positions were calculated with respect to their actual distances from the Sun [3], since the system looks similar to our actual solar system, in terms of the order of the celestial objects. Earth's initial position is set at the center (0,0,0), and the Moon is set at (0.00257,0,0), where 0.00257 AU is roughly the actual distance of the Moon from the Earth. The Sun is next in line, set at 1 AU, its actual distance away from the Sun. Now, here's where things become a little bit funky. The rest of the positions are calculated by adding the Sun's distance away from the Earth, 1 AU, and the planet's actual distance from the Sun in our solar system. I set initial positions this way just to simplify the process, but also try to set it as close to the proposed model as possible.

The initial velocities were not kept the same for the celestial objects like the rest of the models and were instead calculated for each object using Kepler's Law for planetary motion and the equation for Uniform Circular Motion:

$$T^2 = \frac{4pi^2}{GM} * a^3$$
$$V = 2pi * r/T$$

Again, to find any deviations right away, I started at dt = 0.0005 and tmax = 5, since tmax = 10 was too difficult to differentiate the trajectories of the different objects. As shown by Figures 9 and 10, we can see that the objects have immediately shot out of their elliptical orbits. This seemingly shows that the objects are no longer in any sort of orbit, and the system has broken down.

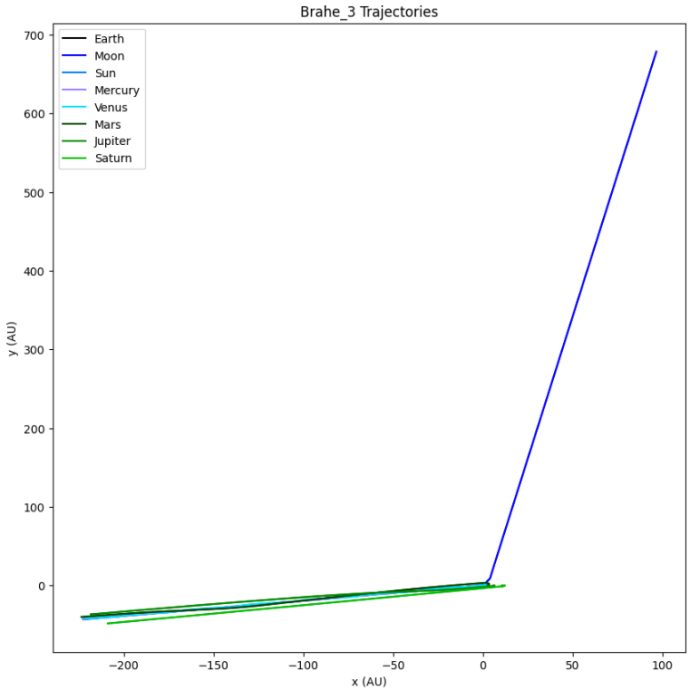

Figure 9: Brahe Trajectories (dt = 0.0005, tmax = 5)

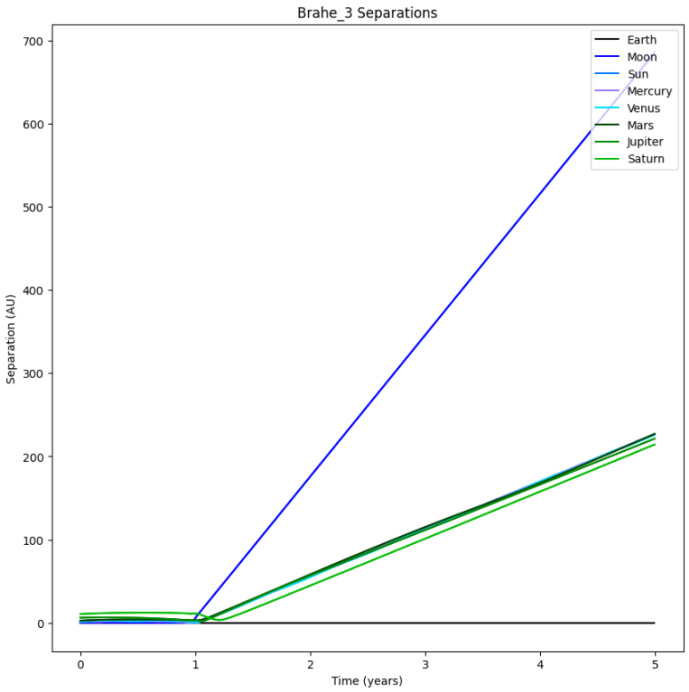

Figure 10: Brahe Separations (dt = 0.0005, tmax = 5)

However, if we decrease dt to 0.00005, we see some glimpses of elliptical orbits in Figures 11 and 12. Some objects like the Moon, Venus, and Mercury already show their elliptical orbits. The other objects have a bunch of squiggles, but there seems to be what looks like the beginning of an elliptical orbit for most objects. This can be displayed the best by Jupiter, as the trajectory looks like it could become an ellipse with its curved path, and its separation plot line seems to show the dip of a sine wave.

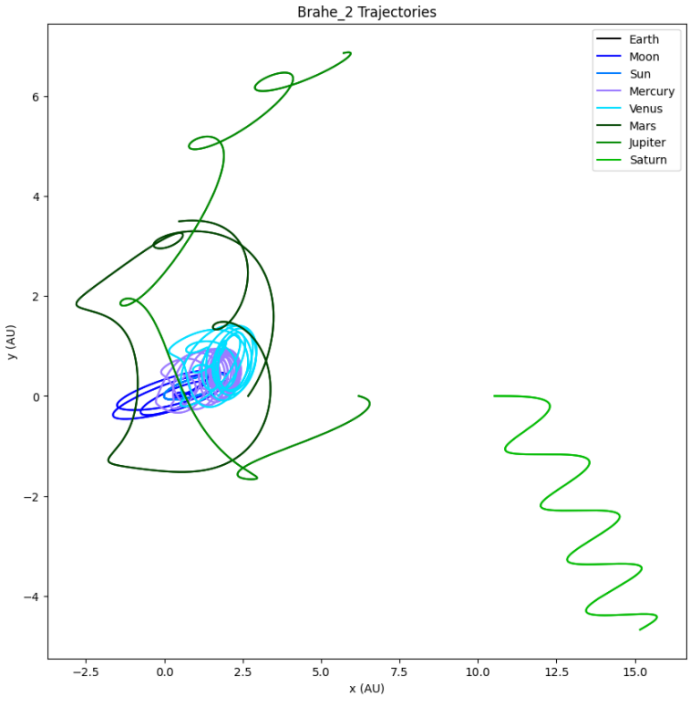

Figure 11: Brahe Trajectories (dt = 0.00005, tmax = 5)

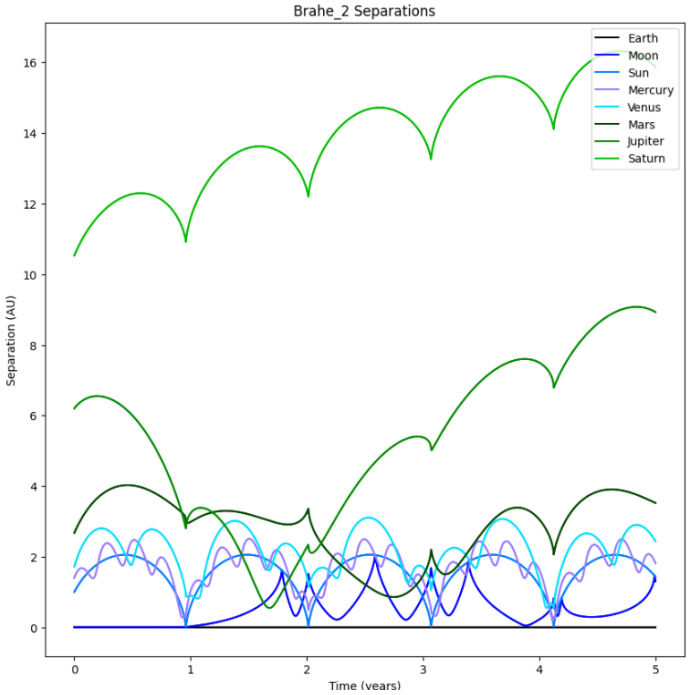

Figure 12: Brahe Separations (dt = 0.00005, tmax = 5)

Increasing tmax to 50, we find that Figures 13 and 14 seem to confirm the elliptical orbits, as we can see the general sin-wave outlines of the object separations. I'm not exactly sure how to explain this, maybe the immediate "de-orbit" can be explained by the squiggles within the trajectories.

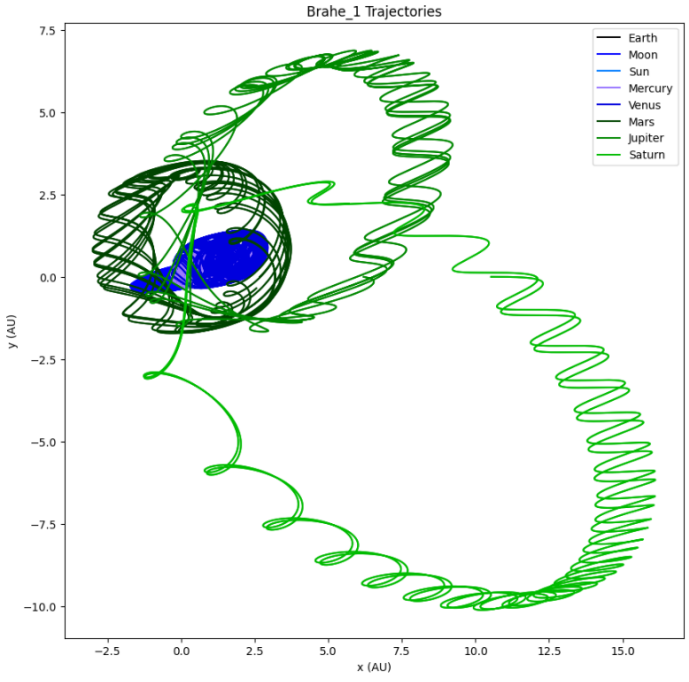

Figure 13: Brahe Trajectories (dt = 0.00005, tmax = 50)

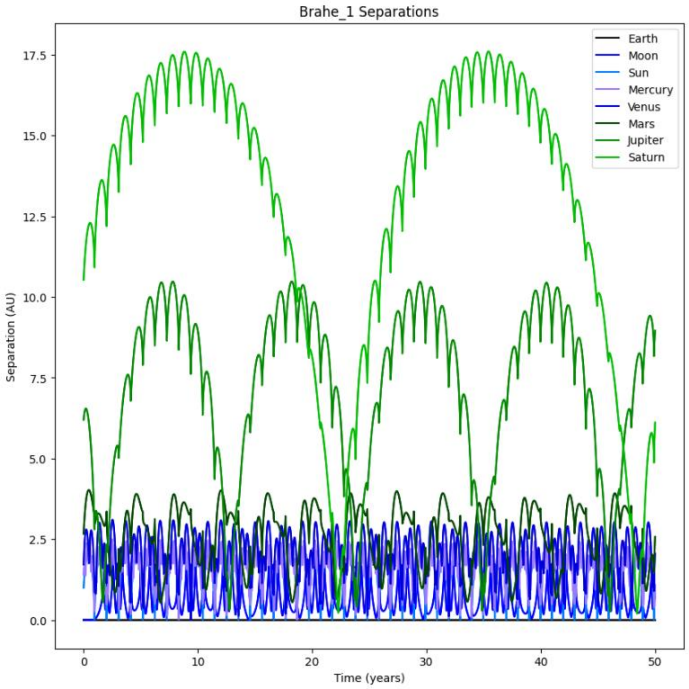

Figure 14: Brahe Separations (dt = 0.00005, tmax = 50)

Results and Limitations

Simulating the geocentric models, we find that both the Ptolemaic and Platonic models both mostly stay together, with the exception of 2 planets shooting out of orbit, and the Brahe model sticking together, albeit with some odd elliptical orbits. The Brahe model sticking together makes a decent amount of sense if you draw out the theoretical orbits and positions of the planets [1], it looks similar to our solar system. Also, if you put the sun as the frame of reference and as the center of the universe, the order of the planets is correct.

I found it somewhat surprising that the systems did not immediately dissolve into chaos and have celestial objects shooting everywhere. However, it is possible, and very likely, that within the systems, there were many collisions between the objects. This would make sense, as with a mass as large as the Sun not at the center of the universe, the gravitational pull would drag the objects around and close to the Sun, eventually colliding with each other.

Some limitations to these simulations include computational resources and keeping the initial velocities of the celestial objects. I used the fourth_order_step function for all the simulations for greater precision. However, this was more computationally intensive, and simulations with a large tmax or very small dt could take as long as 2.5 hours to complete. Therefore, it was not possible to simulate more precise models.

Code Explanation

The main body of the n_body code provided by Dr. Slinker stayed the same. I created a numpy array with the order of the positions of the planets from the sun. I wrote the function ‘assign_pos’ that took in the empty position dictionary of the system to be simulated, the position array, and

the celestial object array. Since the celestial object array is in order from the center to the outermost object, a for loop was used to assign the first object with the first position array, the second object with the second position array, and so on. This can be seen in the code below, along with an example of how it is used with the Ptolemaic simulations.

```
positions = np.array([
    np.array([0.002841029214124732, -0.0008551488389783957, -0.0001372196345812671]),
    np.array([0.3401540875319301, -0.2044550792740463, -0.04772012105321857]),
    np.array([0.5524983482189795, -0.4769264575796522, -0.03838223296316453]),
    np.array([-0.1683241372257412, 0.9674441923084423, -0.0001669835242727353]),
    np.array([-0.1666917632267577, 0.9694171153182533, -0.0002967360187374639]),
    np.array([1.358690797965978, -0.2758171771328441, -0.03917699408597988]),
    np.array([-3.726726780726139, 3.793259959947861, 0.06756053344765718]),
    np.array([-5.405313185445199, -8.35009362433022, 0.3603093935541996]),
    np.array([19.30657051495468, 5.250507992762163, -0.2306220390711898]),
    np.array([27.52968718187136, -11.8437607152196, -0.3905499561179334]),
    np.array([7.400602906905451, -31.91988616953689, 1.274938896927729])
])
```

```
def assign_pos (init_pos_dict,pos_array,obj_arr): #works
    index = 0
    for i in obj_arr:
        init_pos_dict[i] = pos_array[index]
        index=index + 1
```

```
ptol_objects = np.array(["Earth","Moon","Mercury","Venus","Sun","Mars","Jupiter","Saturn"])
ptol_init_pos={}
assign_pos(ptol_init_pos,positions,ptol_objects)
```

The line_colors array was used to make sure that in the trajectory and separation plots, the different celestial objects were clearly distinguishable from each other. Each index is a hex code for a different color.

```
line_colors = np.array(['#000000',
                        '#0000ff',
                        '#007bff',
                        '#9b7bff',
                        '#0000dd',
                        '#004400',
                        '#008800',
                        '#00bb00',
                        '#ff0000',
                        '#ff00ff',
                        '#0000aa']) #hex values for colors to differentiate
```

To generalize the simulation, I created the run_simulation function to run the simulation for each model. This way, I would only have to pass through the celestial object array and initial conditions to run the simulation, instead of copy-and-pasting the simulation code over and over again. The function takes in the system's name, the array of objects in order, the initial positions of the objects' masses, and the initial velocities of the objects, dt, and tmax. The code block is shown below:

```
def
run_simulation(name,object_array,init_pos_dict,mass_dict,init_vel_dict,dtt
,ttmax):
    #gather a few pieces of information based on choosen parameters
    nmax=object_array.size
    dt2=dtt*dtt
    m=np.zeros(nmax)
    x=np.zeros((nmax,3,1))
    v=np.zeros((nmax,3,1))
    for i in range(0,nmax):
        m[i]=mass_dict[object_array[i]]
        x[i,:,0]=init_pos_dict[object_array[i]]
        v[i,:,0]=init_vel_dict[object_array[i]]

    #evolve the system; this is the part where the simulation is actually
carried out
    for t in range(0,(int)(ttmax/dtt)):
        x,v=fourth_order_step(x,v,m,nmax,newtong,t%10==0)

    f=plt.figure()
    f.set_figwidth(10)
    f.set_figheight(10)
    #plt.axes().set_aspect('equal','datalim')
    lines = []
    for i in range(0,len(object_array)):
        lines+=plt.plot(x[i,0]-x[0,0],x[i,1]-x[0,1],label =
object_array[i],color=line_colors[i])
        plt.plot(x[i,0]-x[0,0],x[i,1]-x[0,1],label =
object_array[i],color=line_colors[i])
    labels = [l.get_label() for l in lines] # Credit:
https://stackoverflow.com/questions/14826119/multiple-legends-in-
matplotlib-in-for-loop
    plt.legend(lines,labels)
    plt.xlabel("x (AU)")
    plt.ylabel("y (AU)")
    #plt.autoscale(enable=True, axis='x', tight=True)
    #plt.autoscale(enable=True, axis='y', tight=True)
    plt.title(name + " Trajectories")
    plt.show()
    file_name = name + "_trajectories.png"
```

```
    f.savefig(file_name,bbox_inches='tight')

    f2=plt.figure()
    f2.set_figwidth(10)
    f2.set_figheight(10)
    lines = []
    for i in range(0,len(object_array)):
        sep=np.sqrt(np.sum((x[i]-x[0])**2,axis=0))

plt.plot((ttmax/sep.size)*np.array(range(0,sep.size)),sep,color=line_color
s[i])

lines+=plt.plot((ttmax/sep.size)*np.array(range(0,sep.size)),sep,label=obj
ect_array[i],color=line_colors[i])
    labels = [l.get_label() for l in lines]
    plt.legend(lines, labels,loc = "upper right")
    plt.xlabel("Time (years)")
    plt.ylabel("Separation (AU)")
    plt.title(name+" Separations")
    plt.show()
    f2.savefig(name+"_separations.png",bbox_inches='tight')
```

To calculate runtime, I ran an initial simulation with a relatively small tmax and larger dt and used the time module to find the runtime. The runtime should be about proportional to tmax*N^2/dt, where N is the number of objects within the system. Using the equation runtime = C * tmax*N^2/dt, I wrote the function calcC to calculate the constant C for the system. The function calc_run_time is then used to calculate and print the expected runtime in hours, minutes, and seconds.

```
def calc_run_time(C,obj_arr,t_max,d_t):
    N = len(obj_arr)
    time = C*t_max*N**2/d_t
    hours = int(time/(60*60))
    minutes = int((time-hours*3600)/60)
    seconds = int(time%60)
    print("Expected runtime: ",hours," hours ",minutes," minutes ",seconds," seconds")

def calcC (obj_arr, t_max, d_t, runtime):
    N=len(obj_arr)
    C = runtime*d_t/(t_max * N**2)
    return C
```

Works Cited

[1] “Historical Astronomy: Concepts: Models of the Solar System.” Historical Astronomy Concepts: Models of the Solar System, themcclungs.net/astronomy/concepts/models.html#:~:text=In%20class%2C%20we%20discussed%20three,of%20these%20proposed%20by%20Brahe. Accessed 12 May 2023.

[2] Laskar, Jacques. “Jacques Laskar.” Jacques Laskar • Home, 11 June 2009, perso.imcce.fr/jacques-laskar/en/.

[3] “Planets.” NASA, 22 Mar. 2023, solarsystem.nasa.gov/planets/overview/.

[4] “Tychonic System.” Encyclopædia Britannica, www.britannica.com/science/Tychonic-system. Accessed 12 May 2023.

[5]Williams, Matt. “What Is the Geocentric Model of the Universe?” Universe Today, 4 Feb. 2016, www.universetoday.com/32607/geocentric-model/#:~:text=According%20to%20Plato%2C%20the%20Earth,stars%2C%20and%20the%20fixed%20stars.