Ksp 4.0 Midterm-Report
Solar System Dynamics

Suryansh patidar

June 25, 2023

Abstract

So till this time, I've learned a lot about celestial mechanics, the 3 body problem, and how to plot/simulate all those which is really interesting for me.

Contents

1 Celestial mechanics

1.1 Two Body Problem

Starting with the Celestial mechanics of H.Karttunen, i revised the concepts that were taught in PH111 and that I learned during my Astronomy Olympiad camp. String with the newton's laws and deriving all 3 Kepler laws, then going a bit into Lagrange points and finally completing with the Virial theorem and Jeans limit.

1.2 Three Body Problem

A pdf shared by the mentors for the 3 body problem and solving for it, especially the Lagrange points and the contours of Potential energy for the Reduced 3 body problem, that clearly shows the Lagrange points.

2 Plotting the trajectories of Orbits and the contours

2.1 Elliptical Orbit

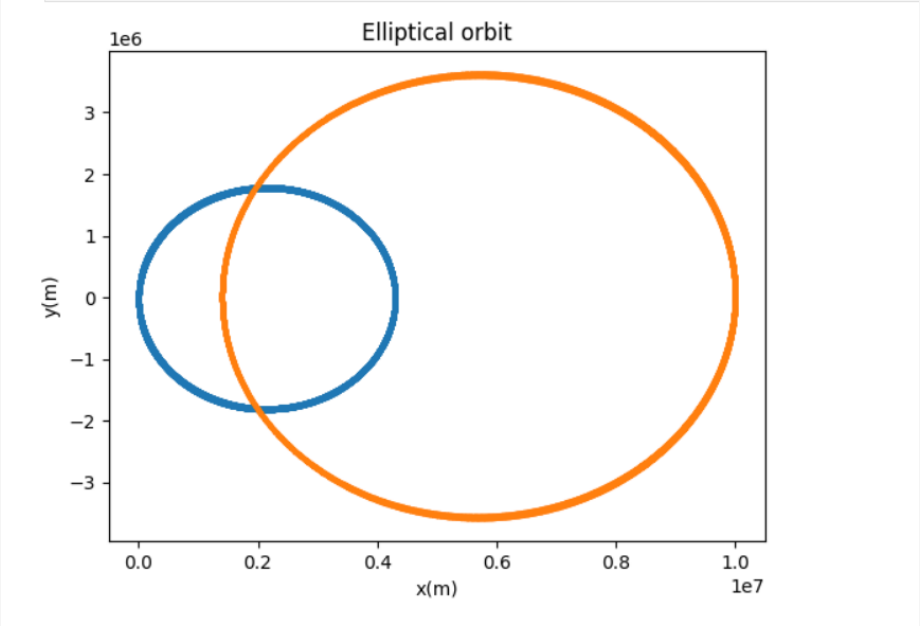
Starting with the normal Euler integration method, i wrote the whole code for plotting the orbit in 'Class' as suggested by mentors, so that I will be able to use that further when it will be required. Initially i was not getting the correct plot because of the parameters that i used was random and not for some realistic celestial bodies.

2.2 Hyperbolic Orbit

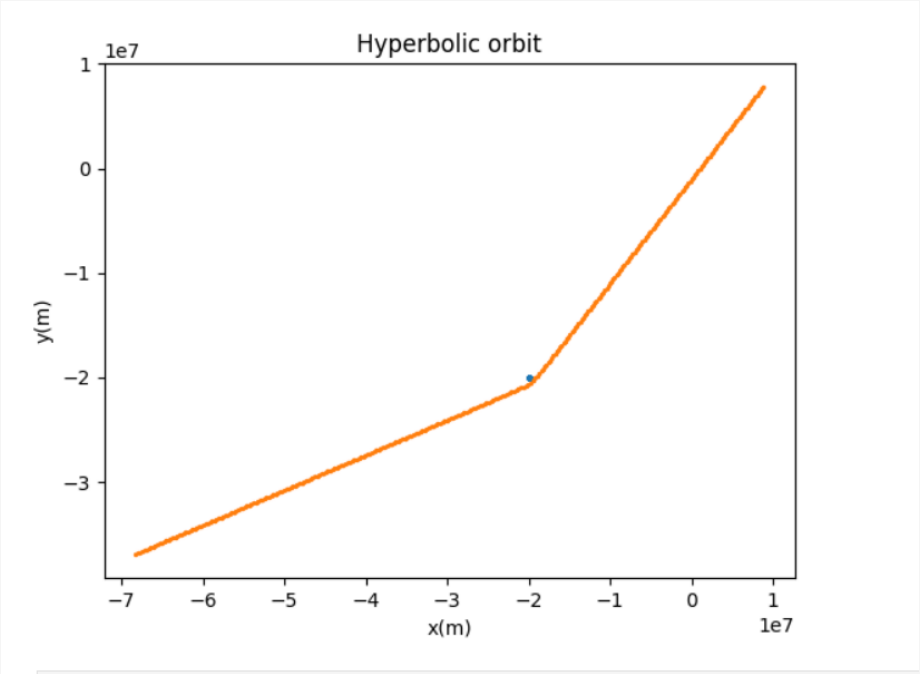
Using the same code as for the elliptical orbit, I just changed the parameter values for the two particle.
The format of code that I used for plotting the trajectory(both for Elliptical and Hyperbolic orbit) has been given below:

```
    class Particle:
       def __init__(self, mass, x,y,vx,vy):
            # For defining the mass, co-ordinates and
               velocity components for the particles.
        def update(self,time_step,f_x,f_y):
            # For updating the vel and x,y-coordinates
                according to the acceleration, that
               comes from the type of force for which
               we want to plot
    class GravitationalSimulation:
         def __init__(self, particle1, particle2):
            # Initializing particle1 and particle2, so
                that I can use its parameters for
               further code
        def calculate_force(self):
            # For storing the gravitational forces
               that i calculated(component-wise) in a
               numpy array
        def simulate(self, time_step, num_steps):
            # For simulating/ plotting the trajectory(
               scatter plot) by updating the positions
                according to the x and y forces
               calculated in above function
    # Parameter values that i used for Elliptical
       Orbit was
        "new_particle_1 = Particle(2e30,0,0,0,1e6)
         new_particle_2 = Particle(1e30,1e7,0,0,-2e6)"
    # And the parameter values that i used for
       Hyperbolic Orbit was
        "particle_1=Particle(2e30,-2e7,-2e7,0,0)
         particle_2=Particle(6e18,0.9e7,0.8e7,-2e7,-2
            e7)"
```

The plots that i got was the ones given below,



(a) **Elliptical Orbit**



(b) **Hyperbolic orbit**

2.3 Contour for the Reduced 3 body problem

Now moving to the 3 body problem and especially simulating for the Reduced 3 body problem by plotting the Contour of Potential energy that shows the Lagrange points.
The code was not that tough to write tho, but i was facing issues with the plotting , as i was not getting correct plot even after writing the code correctly. Then after changing the parameter values and making masses more comparable, i got a decent plot but not totally accurate. Although it is nearly same i guess in comparison to the actual one.
For the code, i only had to add these functions in above code for trajectories.

```
    def potential_energy(self,x,y):
        # For calculating the potential energy in a
            reduced 3 body problem
    def generate_contour_plot(self,x0,y0,x1,y1,n):
        # It will generate the contour according to
            the number of x and y values
        # Parameter values that i used for Contour
            plotting was
             "new_particle_1 = Particle(2e30,0,0,0,1e6)
              new_particle_2 = Particle(1e30,1e7,0,0,-2
                 e6)"
        # I chose this as to make the masses
            comparable so that i need not had to zoom
            in the figure to my area of interest that i
             might have to do in the case of sun-earth
            where masses are not comparable
```

Finally i got this as my contour, but i have to improve it so it to look more like the actual one.I'm not sure why it is not perfect, but i will work on it:

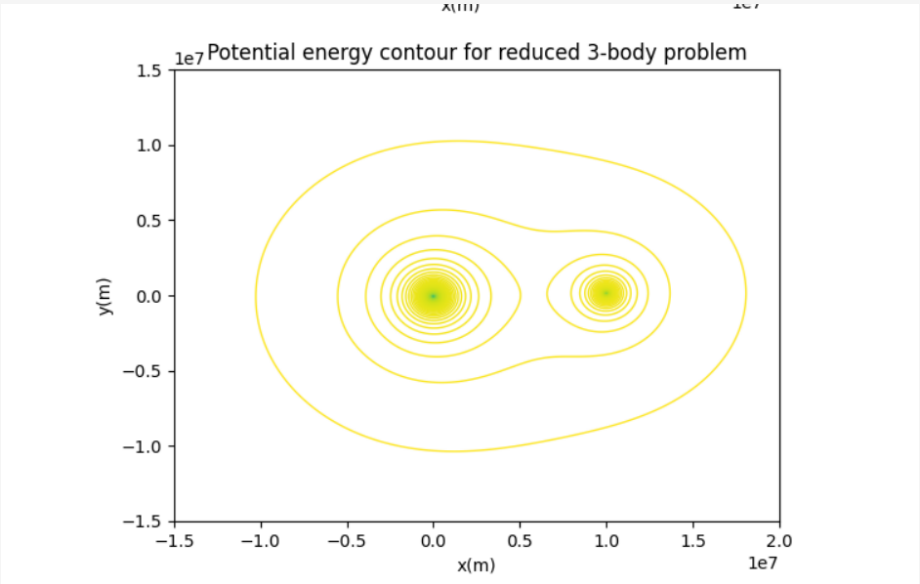


Figure 2: Contour plot of potential energy for the reduced 3 body