Circular Orbit Simulations

Shivam Kumaran SC17B122 18/09/2020

Problem 1

To make animation for a binary system in circular orbit. The bodies will be orbiting around the center of mass. The radius of the orbit of bodies are given by :

$$M_1 \times a_1 = M_2 \times a_2$$

And the distances between them is given by

$$a = a_1 + a_2$$

1 Python Code

```
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
from matplotlib import style as st
from matplotlib.animation import FuncAnimation
import matplotlib.animation as animation


phi_p = np.radians(180)
phi_s = 0


theta = np.radians(np.linspace(0, 360,360,endpoint=True))

a = 1
m_c_p = 1 #mass constant for planet taken Unity
m_c_s = float(input('Enter Mass of star:(assuming mass of planet unity):'


# scaling factor , as the actual astronomical distance is much large ,
# so for clarity , we need to scale up the size of the bodies
# we will scale up the size of both bodies using this value
m_scale =  200/m_c_s
```

```
m_p = m_scale * m_c_p   #mass of planet
m_s = m_scale * m_c_s   #mass of star



##orbit size of bodies
a_p = np.divide(a*m_s,(m_s+m_p))
a_s = np.divide(a*m_p,(m_s+m_p))

orb_p = [a_p*np.cos(theta + phi_p),a_p*np.sin(theta+phi_p)]
orb_s = [a_s*np.cos(theta + phi_s),a_s*np.sin(theta+phi_s)]
#com_pos = np.divide((m_p*orb_p+m_s*orb_s),(m_p+m_s))

st.use('dark_background')
fig = plt.figure(1, figsize=(12,12))
ax = plt.gca()
ax.set_xlim(-1.2*a,1.2*a)
ax.set_ylim(-1.2*a,1.2*a)
ax.set_aspect(1)
ax.axis('off')

def init():
    plt.plot([0],[0],'+r',markersize=50,zorder=3) # center of mass
    plt.plot(orb_p[0],orb_p[1] ,'-r', markersize=m_p,zorder=0) #planet or
    plt.plot(orb_s[0],orb_s[1],'-r', markersize=m_s,zorder=0) # star orbi
    #plt.show()


planet, = plt.plot([],[],'w.',markersize=m_p,zorder=3)
star, = plt.plot([],[],'w.',markersize=m_s,zorder=3)
planet_center, = plt.plot([],[],'y+',markersize=10,zorder=3)
star_center, = plt.plot([],[],'y+',markersize=10,zorder=3)


def move(i):
    planet.set_data(orb_p[0][i],orb_p[1][i])
    star.set_data(orb_s[0][i],orb_s[1][i])
    planet_center.set_data(orb_p[0][i],orb_p[1][i])
    star_center.set_data(orb_s[0][i],orb_s[1][i])
```

```
Writer = animation.writers['ffmpeg']
writer = Writer(fps=30, metadata=dict(artist='Me'), bitrate=1800)

sim_animation = FuncAnimation(fig,move,[i for i in range(len(theta))],ini


ch = str(input('Display Plot or Save Plot : [D/S]:'))
if(ch=='D'):
    plt.show()
elif(ch=='S'):
    matplotlib.use("Agg")
    sim_animation.save('ratio_'+str(m_c_s)+'.mp4', writer=writer)
```

2 Result

Animation made using the above formula for orbit radius and the center of Mass of the binary system. The radius of bodies taken to be linearly proportional to the masses GIF files were generated for the following mass ratio:

1. $M_1 = M_2$

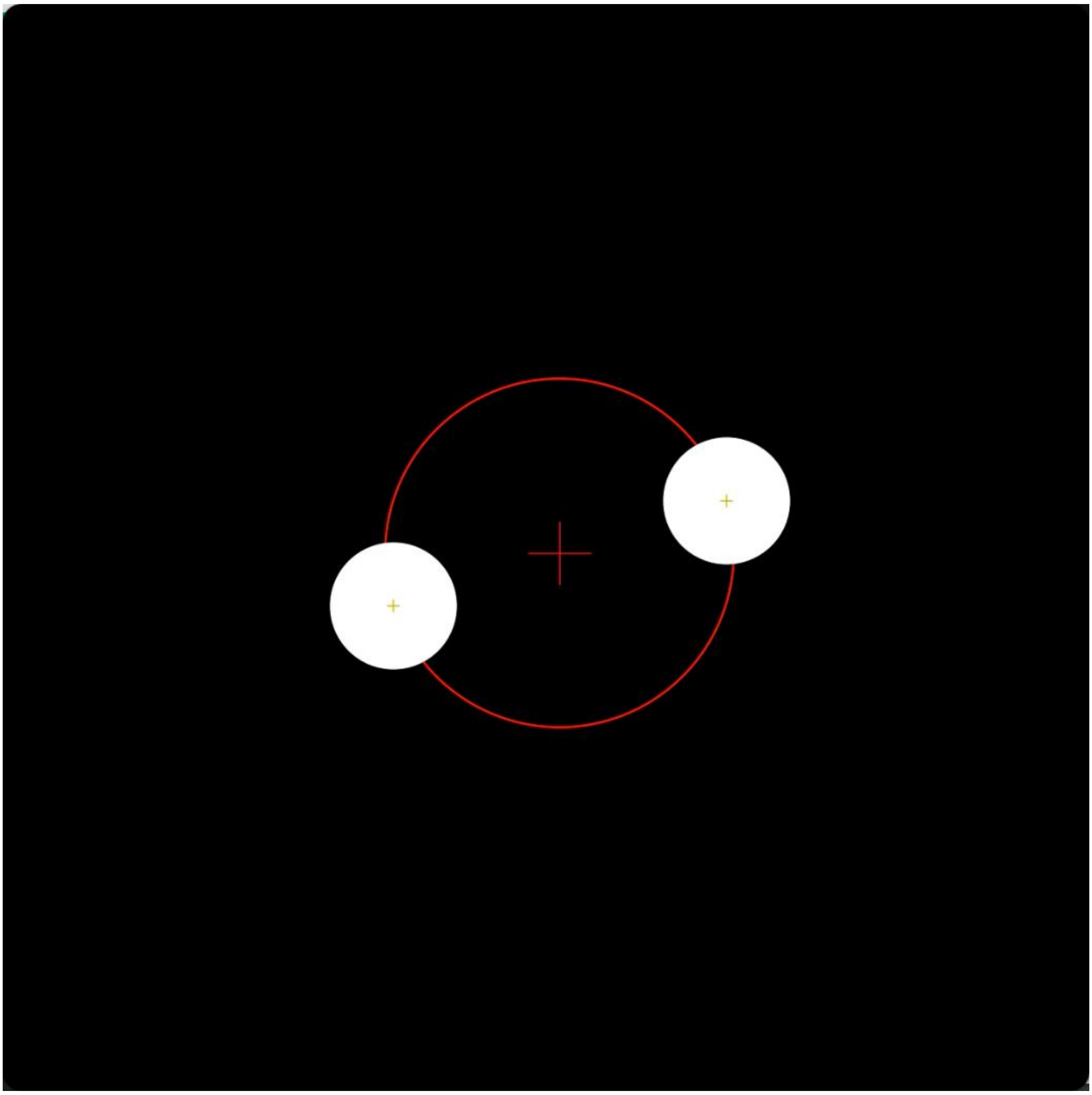

Figure 1: Equal Mass Bodies

- The radius of both the orbits are same.
- Orbital speed are also equal

2. $M_1 = 10 \times M_2$

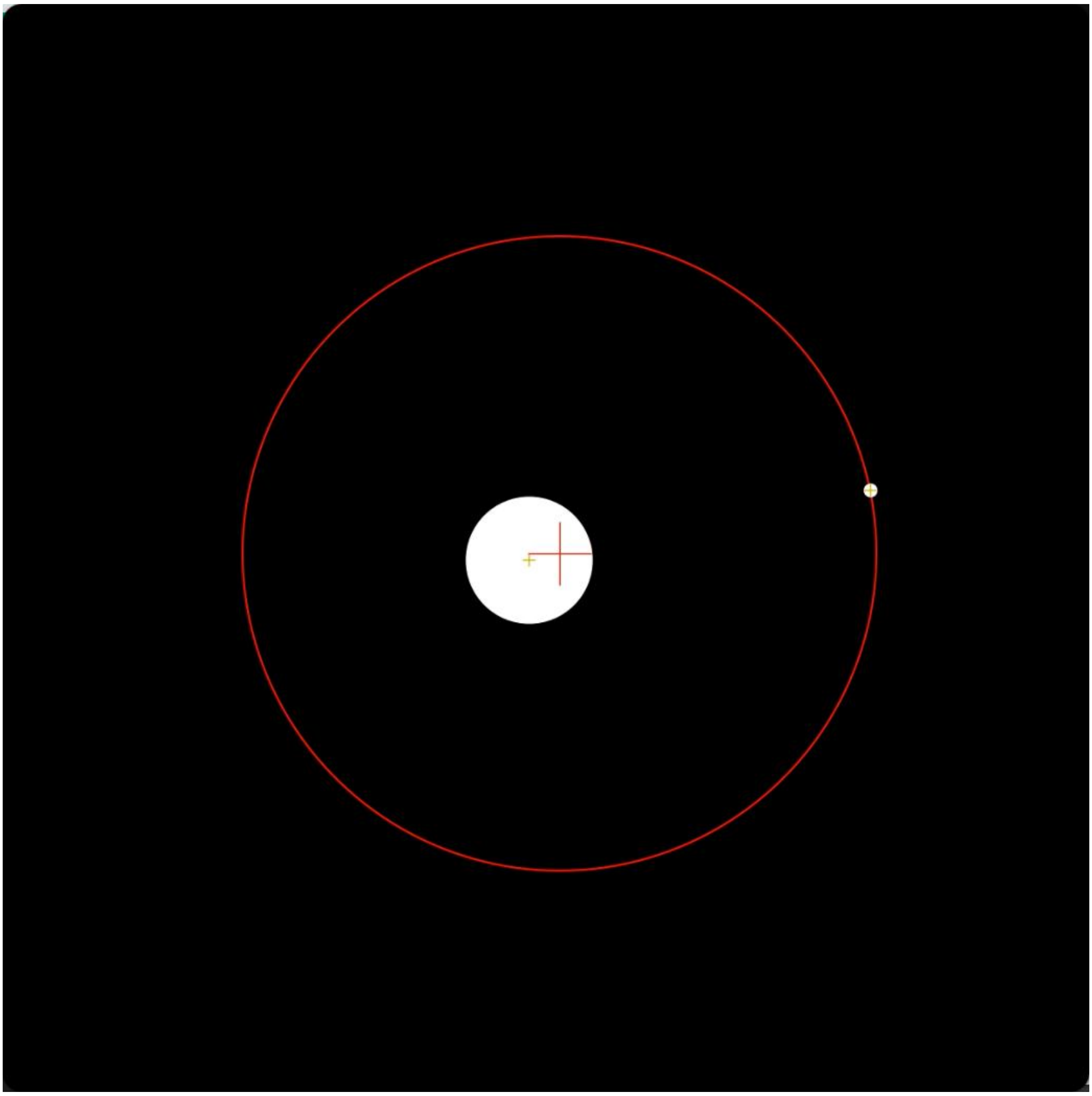

Figure 2: Mass Ratio 1:10

- Body having heavier mass have smaller radius
- Motion of both the bodies around center of mass is clearly visible
- Although, Larger body's center of orbit lies inside the body itself , its revolution around center of mass is also visible

3. $M_1 = 10^6 \times M_2$

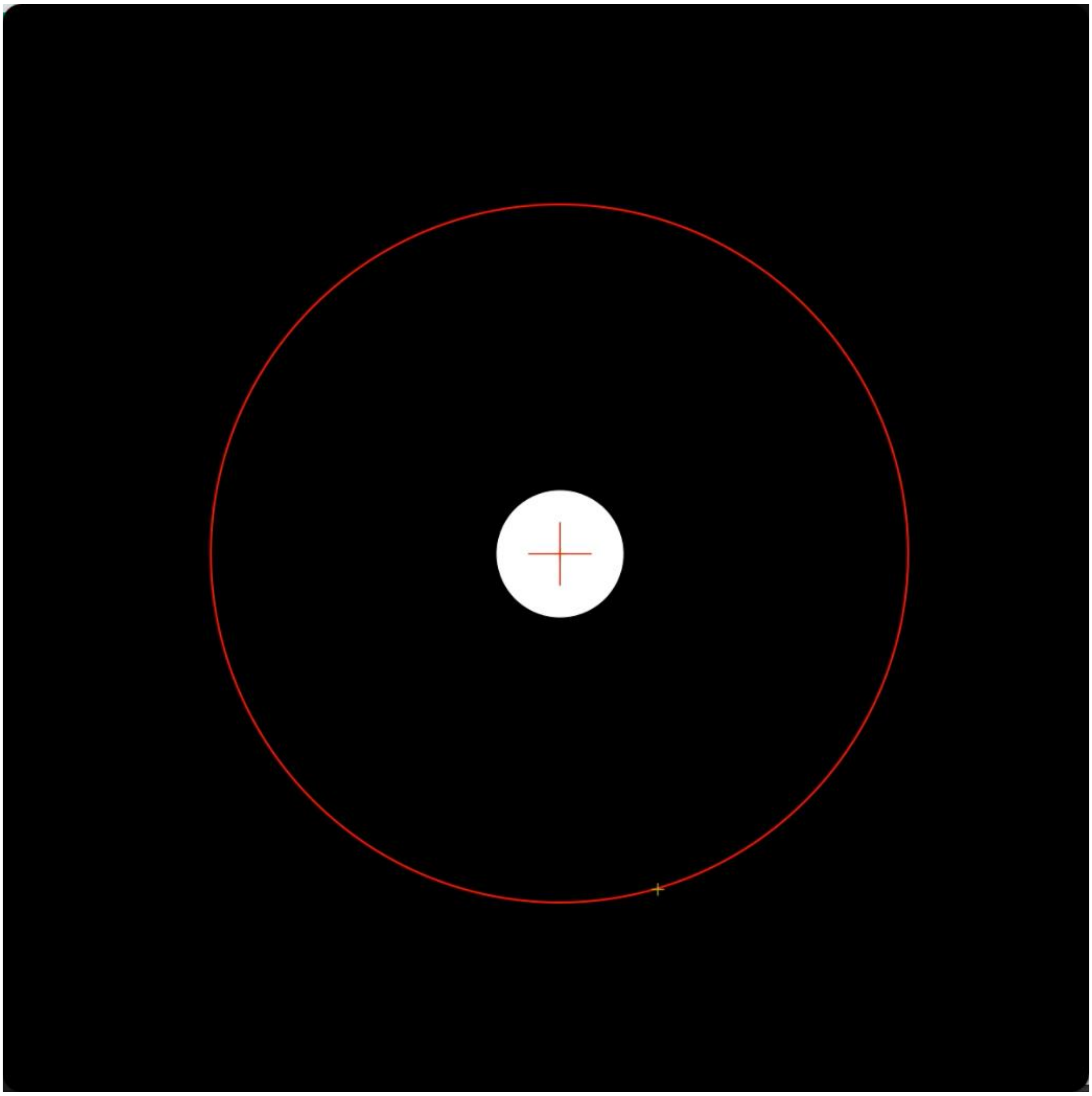

Figure 3: Caption

- Due to enormous size difference, It is very difficult to spot smaller boy in the plot
- Smaller body have very high speed of revolution
- Center of mass coincide almost completely with the center of larger body
- Hence orbital motion of Heavier body is negligible
- It appears like smaller body is orbiting large body.