# Orbiters

Louis George

November 16, 2021

I start with the following code:

```
MASS=400000
mass=3000
G=6.67408E-11
timeFrameLength=1
pos=[100,400]
POS=[200,200]
vel=[2,-1]
```

This is deciphered as the following:

$$\text{Centre Mass} = 400000$$
$$\text{Orbiter Mass} = 3000$$
$$\text{Gravitational Constant} = 6.67408 \times 10^{-11}$$
$$\text{Time Resolution} = 1$$
$$\text{Orbiter Position} = (100, 400)$$
$$\text{Centre Position} = (200, 200)$$
$$\text{Orbiter Velocity} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

The next block of code defines the function "Orbiter":

```
def Orbiter(pos,POS,veloc,MASS,mass):
    # finding the orbital radius
    rad=math.sqrt((POS[0]-pos[0])**2+(POS[1]-pos[1])**2)
    # getting the acceleration
    acc=(G*MASS)/abs(rad)**2
    # getting the new velocity vector
    for i in range(2):
        veloc[i]+=(acc*timeFrameLength)*((POS[i]-pos[i])/rad) #(pos[i]/rad)
                    being to make it go towards the object
    # getting the new position
    for i in range(2):
        pos[i]+=veloc[i]*timeFrameLength
    return [pos,veloc]
```

The first part of this code is working out the radius of the orbiter, using Pythagoras' Theorem:
$R = \sqrt{\Delta x^2 + \Delta y^2}$

We then find the acceleration due to gravity using the following equation:
$g = G\frac{M}{R^2}$
With $M$ being the mass at the centre, $g$ being the acceleration due to gravity and $R$ being the radius

of the orbiter.

The next part (the for loop) is getting the new velocity vector for this time-frame.

$$\vec{v} = \vec{v}_0 + \vec{a}\Delta T$$
$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \end{bmatrix} + \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \Delta T$$

I separated the acceleration into its components by multiplying it by a unit vector:
$\begin{bmatrix} \frac{\Delta x}{R} \\ \frac{\Delta y}{R} \end{bmatrix}$
Which points towards the centre object, worked out by finding the proportion each dimension contributes to the radius.

After this we can work out the change in displacement:
$\vec{s} = \vec{v}\Delta T$

Finally, the first iteration the initial values are plugged into the function, and every time frame after that, the function is used with the previous values and the displacement is displayed.