```python
In [1]: import numpy as np

        g=9.81 # m/s^2
        pendulum_length = 1 # meters
        theta_0 = 40 # degrees
        t_final = 18.5 # seconds
        dt=0.005 # seconds
        n_steps=int(t_final/dt)

        def pendulum_rk4(dt, n_steps, initial_theta_degrees, l):
            ''' RK4 integrator to solve the Simple Pendulum ODE system:
                d^2 theta / d theta ^2 = -(g/l)sin(theta)
                returns the x,y position, time, theta and d theta/ dt arrays with n_
            '''
            theta_old = initial_theta_degrees*(np.pi/180)

            theta_dot_old = 0

            pos_x = np.zeros(n_steps+1)
            pos_y = np.zeros(n_steps+1)
            times = np.zeros(n_steps+1)
            thetas = np.zeros(n_steps+1)
            theta_dots = np.zeros(n_steps+1)

            ## initial conditions
            pos_x[0] = np.sin(theta_old)*l
            pos_y[0] = -np.cos(theta_old)*l
            times[0] = 0
            thetas[0] = theta_old


            t_new = 0
            for i in range(n_steps):
                t_new += dt


                k1_theta = theta_dot_old
                k1_theta_dot = -(g/l)*np.sin(theta_old) ## use ODE equation for k_th

                k2_theta = theta_dot_old + k1_theta_dot*dt/2
                k2_theta_dot = -(g/l)*np.sin(theta_old + k1_theta*dt/2)

                k3_theta = theta_dot_old + k2_theta_dot*dt/2
                k3_theta_dot = -(g/l)*np.sin(theta_old + k2_theta*dt/2)


                k4_theta = theta_dot_old + k3_theta_dot*dt
                k4_theta_dot =  -(g/l)*np.sin(theta_old + k3_theta*dt)

                ## use k1,k2,k3,k4 to update theta and dtheta/ dt
                theta_new = theta_old + dt * (k1_theta + 2*k2_theta + 2*k3_theta + k
                theta_dot_new = theta_dot_old + dt * (k1_theta_dot + 2*k2_theta_dot
```

```python
        ### update arrays
        pos_x[i+1] = np.sin(theta_new)*l
        pos_y[i+1] = -np.cos(theta_new)*l
        times[i+1] = times[i]+dt
        thetas[i+1] = theta_new
        theta_dots[i+1] = theta_dot_new
        ###

        theta_old = theta_new
        theta_dot_old = theta_dot_new

    return pos_x, pos_y, times, thetas, theta_dots


pos_x, pos_y, times, thetas, theta_dots = pendulum_rk4(dt, n_steps, theta_0,

print(f"After {round(times[-1], 2)} seconds, the pedulum bob has height {rou
```

After 18.5 seconds, the pedulum bob has height 4.21 meters