```
import numpy as np
from matplotlib import animation
import pylab

def RungeKutta(f1, f2, f3, f4, a_init, b_init, c_init, d_init, t0, tf, h=0.1):
    A = []
    B = []
    C = []
    D = []
    T = np.arange(t0, tf, h)
    a = a_init
    b = b_init
    c = c_init
    d = d_init
    for t in T:
        A.append(a)
        B.append(b)
        C.append(c)
        D.append(d)
        k1 = h*f1(a,b,c,d,t)
        l1 = h*f2(a,b,c,d,t)
        m1 = h*f3(a,b,c,d,t)
        n1 = h*f4(a,b,c,d,t)
        k2 = h*f1(a+k1/2,b+l1/2,c+m1/2,d+n1/2,t+h/2)
        l2 = h*f2(a+k1/2,b+l1/2,c+m1/2,d+n1/2,t+h/2)
        m2 = h*f3(a+k1/2,b+l1/2,c+m1/2,d+n1/2,t+h/2)
        n2 = h*f4(a+k1/2,b+l1/2,c+m1/2,d+n1/2,t+h/2)
        k3 = h*f1(a+k2/2,b+l2/2,c+m2/2,d+n2/2,t+h/2)
        l3 = h*f2(a+k2/2,b+l2/2,c+m2/2,d+n2/2,t+h/2)
        m3 = h*f3(a+k2/2,b+l2/2,c+m2/2,d+n2/2,t+h/2)
        n3 = h*f4(a+k2/2,b+l2/2,c+m2/2,d+n2/2,t+h/2)
        k4 = h*f1(a+k3,b+l3,c+m3,d+n3,t+h)
        l4 = h*f2(a+k3,b+l3,c+m3,d+n3,t+h)
        m4 = h*f3(a+k3,b+l3,c+m3,d+n3,t+h)
        n4 = h*f4(a+k3,b+l3,c+m3,d+n3,t+h)
        a += (k1+2*k2+2*k3+k4)/6
        b += (l1+2*l2+2*l3+l4)/6
        c += (m1+2*m2+2*m3+m4)/6
        d += (n1+2*n2+2*n3+n4)/6
    return T, A, B, C, D

def main():
    MsG = 4*np.pi**2
    f1 = lambda x, y, vx, vy, t: vx
    f2 = lambda x, y, vx, vy, t: vy
    f3 = lambda x, y, vx, vy, t: -MsG*x/pow(x**2+y**2, 3/2)
    f4 = lambda x, y, vx, vy, t: -MsG*y/pow(x**2+y**2, 3/2)
    T, X, Y, Vx, Vy = RungeKutta(f1,f2,f3,f4, 1, 0, 0, 2*np.pi, 0.0, 10, 0.01)
    print("Calculated")
    fig, ax = pylab.subplots()
    line, = ax.plot(X[0], Y[0], ’ko’)
    pylab.ylim((-1.1, 1.1))
    pylab.xlim((-1.1, 1.1))
    def init():
        line.set_data(X[0], Y[0])
        return line,
    def anim(i):
        line.set_data(X[i], Y[i])
        return line,
    ani = animation.FuncAnimation(fig, anim, init_func=init, interval=20, frames=range(len(T)))
    pylab.show()

```

```
if __name__ == "__main__":
    main()
```