Symulacja ruchu planety za pomocą równań różniczkowych

Michał Palwal, Wojciech Kura, Karol Gosławski

Grudzień 2020

Spis treści

1 Wstęp

Od wieków ludzkość próbowała zrozumieć ruch ciał niebieskich, czyli punktów na niebie, poruszających się w nietypowy sposób na tle odległych gwiazd. Część z nich jest i była od tysiącleci dostrzegalna gołym okiem, lecz to dopiero w XVI w. Mikołaj Kopernik przyjrzał im się dokładniej i opracował bardziej realistyczny model naszego układu słonecznego formułując teorię heliocentryczną. Na podstawie jej założeń, w XVII wieku, Jan Kepler sformułował trzy prawa, znane dzisiaj jako prawa Keplera, w których między innymi jako pierwszy stwierdził, że planety poruszają się po orbitach eliptycznych, ze słońcem znajdującym się w jednym z ognisk danej elipsy. Prawa te zostały później potwierdzone przez Isaaca Newtona, który stwierdził że są one zgodne z jego zasadami dynamiki, oraz prawem powszechnego ciążenia.

2 Cel Pracy

Celem pracy było narysowanie orbity planetarnej za pomocą równań różniczkowych drugiego rzędu w Pythonie.

3 Baza i równania ruchu

3.1 Kod bazowy

Punktem wyjściowym do pracy nad projektem było napisanie kodu wykorzystującego równania różniczkowe drugiego rzędu do obliczenia ruchu cząstki w polu grawitacyjnym.

```
import math
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

from vpython import *

v0 = 22.
angle = 34.
g = 9.8
kf = 0.9
N = 25
v0x = v0*cos(angle*pi/180.)
v0y = v0*sin(angle*pi/180.)
T = 2.*v0y/g
H = v0y*v0y/g
R = 2.*v0x*v0y/g

graph1 = graph( title='Projectile with Air Resistance',
                xtitle='x', ytitle='y', xmax=R, xmin=-R/20.,
                ymax=8, ymin=-6.0 )
funct = gcurve(color=color.red)
def plotNumeric(k):
    vx = v0*cos(angle*pi/180.)
    vy = v0*sin(angle*pi/180.)
    x = 0.0
    y = 0.0
    dt = vy/g/N/2
    print("          x            y")
    for i in range(3*N):
        rate(30)
        vx=vx - k*vx*dt
        vy=vy - g*dt - k*vy*dt
```

```
x = x + vx*dt
y = y + vy*dt
funct.plot(pos=(x,y))
print("␣%13.10f␣␣%13.10f␣" %(x,y))
```

```
plotNumeric(kf)
```

[1]

3.2 Równania ruchu

Bazując na prawie powszechnego ciążenia Isaaca Newtona

$$F^{(g)} = -\frac{GmM}{r^2} \tag{1}$$

gdzie r oznacza odległość planety od gwiazdy, G to uniwersalna stała grawitacji, a m i M oznaczają odpowiednio masy planety i gwiazdy, lub ogólnie obiektu o masie mniej i większej.
Równanie ruchu dla planet wygląda tak samo jak dla każdego innego ciała, czyli

$$\mathbf{f} = m\mathbf{a} = m\frac{d^2\mathbf{x}}{dt^2} \tag{2}$$

gdzie siły $F^{(g)}$ to:

$$f_x = F^{(g)}cos\theta = F^{(g)}\frac{x}{r}, \tag{3}$$

$$f_y = F^{(g)}sin\theta = F^{(g)}\frac{y}{r}, \tag{4}$$

$$r = \sqrt{x^2 + y^2} \tag{5}$$

Równania ruchu przedstawić można za pomocą dwóch równoczesnych równań różniczkowych drugiego rzędu

$$\frac{d^2x}{dt^2} = -GM\frac{x}{r^3}, \frac{d^2y}{dt^2} = -GM\frac{y}{r^3}. \tag{6}$$

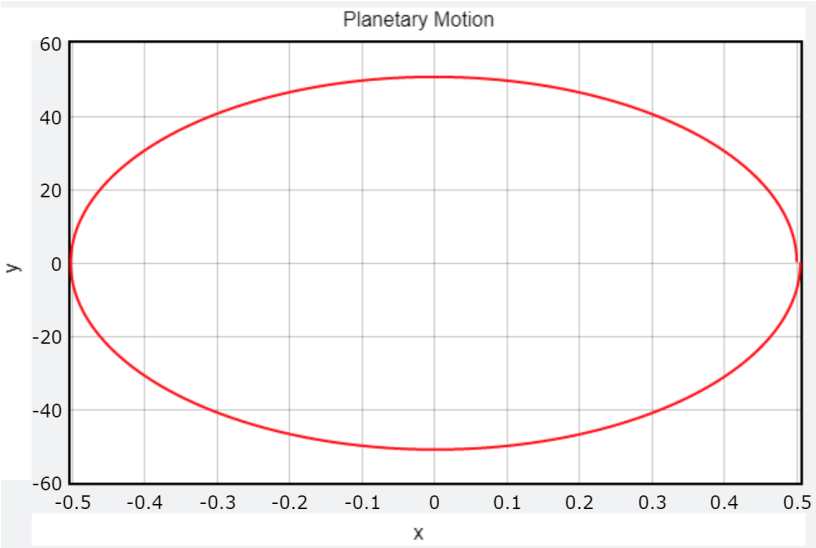
4 Rysowanie orbity

Bazując na tym kodzie rysującym ruch cząstki z uwzględnieniem oporu powietrza dostosowano dane początkowe i obliczenia tak, aby można było za pomocą nich symulować ruch planety po orbicie eliptycznej.

4.1 Kod

```
import math
import scipy

from vpython import *


GM = 1.
x0 = 0.5
y0 = 0.
vx0 = 0.
vy0 = 1.6
N = 500
r = 10
dt = 0.01


graph1 = graph( title='Planetary Motion',
                    xtitle='x', ytitle='y')
funct = gcurve(color=color.red)

def plotNumeric(dt):
    vx = vx0
    vy = vy0
    x = x0
    y = y0

    print("          x             y")
    for i in range(3*N):
        rate(30)
        vx_temp = vx    #Vn
        vy_temp = vy
        vx = vx - (GM/(r*r*r))*x*dt   #Vn+1
        vy = vy - (GM/(r*r*r))*y*dt
        x = x + vx_temp*dt            #Xn+1
        y = y + vy_temp*dt
        funct.plot(pos=(x,y))
        print(" %13.10f  %13.10f " %(x,y))



plotNumeric(dt)
```

4.2 Wykres



Rysunek 1: Wykres przestawiający eliptyczną orbitę ciała niebieskiego

5 Podsumowanie

Orbita została narysowana co świadczy o tym że udało się nam zaimplementować równanie różniczkowe zwyczajne drugiego rzędu. Zadanie zostało wykonane.

Literatura

[1] José Páez Rubin H. Landau and Cristian C. Bordeianu. *A Survey of Computational Physics: Introductory Computational Science*. Princeton University Press, 2011.

[2] Wikipedia. Copernican heliocentrism https://en.wikipedia.org/wiki/copernican_heliocentrism, dostęp 03.12.2020.

[3] Wikipedia. Newton's law of universal gravitation https://en.wikipedia.org/wiki/newton%27s_law_of_universal_gravitation, dostęp 03.12.2020.