Нижегородский государственный университет им. Н. И. Лобачевского Радиофизический факультет

Отчет по практическому заданию по курсу «Методы математического моделирования»»

Выполнил студент:

Холодный А.А.

Гр. 429

Нижний Новгород

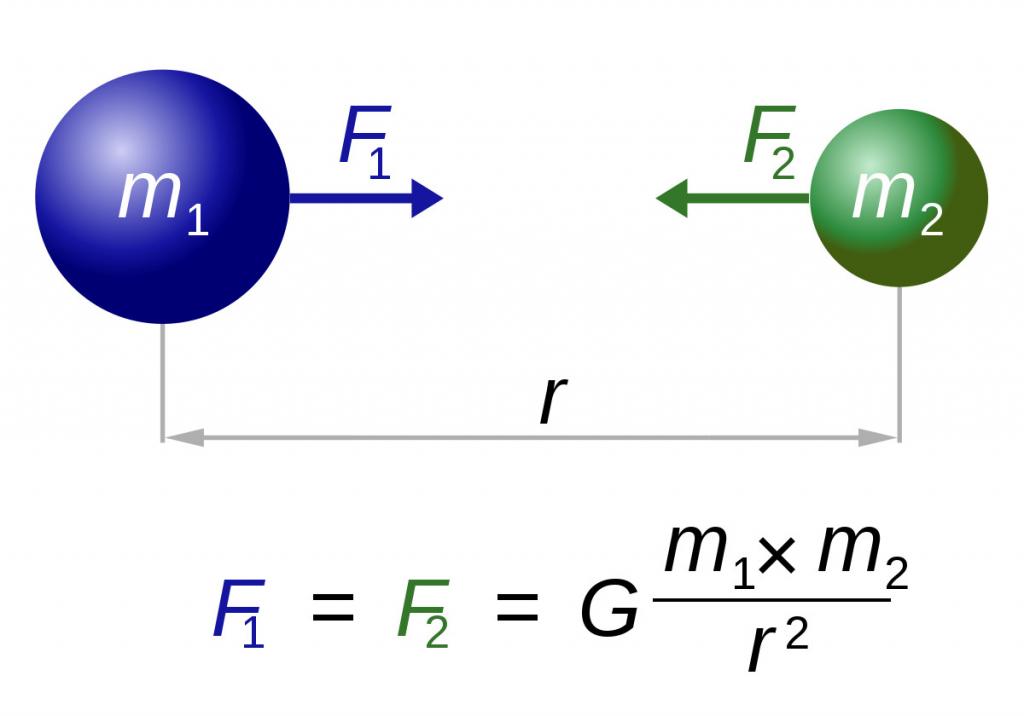
2023

**Задание 8.** Постройте модель Солнечной системы. Рассчитайте необходимые параметры траектории кометы, попавшей в Солнечную систему извне.

**Теория**

Все тела в солнечной системе взаимодействуют в меньшей мере друг с другом и в большей с Солнцем. Взаимодействие описывается законом всемирного тяготения.

**Классическая теория тяготения Ньютона** (***Закон всемирного тяготения Ньютона***) — закон, описывающий гравитационное взаимодействие в рамках классической механики. Этот закон был открыт Ньютоном около 1666 года, опубликован в 1687 году в «*Началах*» Ньютона.



Здесь**�G** — гравитационная постоянная, равная: 6,67430(15)·10−11 м³/(кг·с²).

**Программный код:**

import math  
from tkinter import \*  
  
#окно отрисовки  
root = Tk()  
c = Canvas(root, width=1920, height=1080, bg='black')  
c.pack()  
  
#константы  
coef = 100  
dt = 0.01  
GM = 4 \* math.pi \* math.pi  
  
#комета  
r\_kom = 0.05  
r\_k = r\_kom \* coef  
xk = -8  
yk = -3  
xk\_scr = xk \* coef  
yk\_scr = yk \* coef  
vxk = 4 \* math.pi  
vyk = 1 \* math.pi  
  
  
#радиусы планет  
r\_s = 0.7  
r\_merkury = 0.025  
r\_venera = 0.06  
r\_earth = 0.063  
r\_mars = 0.034  
r\_jupiter = 0.1  
r\_saturn = 0.058  
r\_uran = 0.0253  
r\_neptun = 0.0246  
  
#радиусы планет в пикселях  
rad\_s = r\_s \* coef  
rad\_merk = r\_merkury \* coef  
rad\_v = r\_venera \* coef  
rad\_e = r\_earth \* coef  
rad\_mars = r\_mars \* coef  
rad\_j = r\_jupiter \* coef  
rad\_sat = r\_saturn \* coef  
rad\_u = r\_uran \* coef  
rad\_n = r\_neptun \* coef  
  
#координаты солнечной системы  
x\_scr0 = 960  
y\_scr0 = 500  
  
#начальные скорости планет  
vx = 2 \* math.pi  
vy = 0.5 \* math.pi  
  
vx1 = 1.6 \* math.pi  
vy1 = 0.3 \* math.pi  
  
vx2 = 1.5 \* math.pi  
vy2 = 0.3 \* math.pi  
  
vx3 = 1.4 \* math.pi  
vy3 = 0.3 \* math.pi  
  
vx4 = 1.3 \* math.pi  
vy4 = 0.3 \* math.pi  
  
vx5 = 1.2 \* math.pi  
vy5 = 0.3 \* math.pi  
  
vx6 = 1.1 \* math.pi  
vy6 = 0.3 \* math.pi  
  
vx7 = 1.1 \* math.pi  
vy7 = 0.3 \* math.pi  
  
#начальные координаты планет относительно солнца  
x = 0  
x1 = 0  
x2 = 0  
x3 = 0  
x4 = 0  
x5 = 0  
x6 = 0  
x7 = 0  
  
y = -1  
y1 = -1.5  
y2 = -2  
y3 = -2.5  
y4 = -3  
y5 = -3.5  
y6 = -4  
y7 = -4.5  
  
#отрисовка солнечной системы  
c.create\_oval(x\_scr0 - rad\_s, y\_scr0 - rad\_s, x\_scr0 + rad\_s, y\_scr0 + rad\_s, fill='yellow')  
merkury = c.create\_oval(x\_scr0 + x \* coef - rad\_merk, y\_scr0 + y \* coef - rad\_merk, x\_scr0 + x \* coef + rad\_merk, y\_scr0 + y \* coef + rad\_merk, fill='gray')  
venera = c.create\_oval(x\_scr0 + x \* coef - rad\_v, y\_scr0 + y1 \* coef - rad\_v, x\_scr0 + x \* coef + rad\_v, y\_scr0 + y1 \* coef + rad\_v, fill='orange')  
earth = c.create\_oval(x\_scr0 + x \* coef - rad\_e, y\_scr0 + y2 \* coef - rad\_e, x\_scr0 + x \* coef + rad\_e, y\_scr0 + y2 \* coef + rad\_e, fill='blue')  
mars = c.create\_oval(x\_scr0 + x \* coef - rad\_mars, y\_scr0 + y3 \* coef - rad\_mars, x\_scr0 + x \* coef + rad\_mars, y\_scr0 + y3 \* coef + rad\_mars, fill='orange')  
jupiter = c.create\_oval(x\_scr0 + x \* coef - rad\_j, y\_scr0 + y4 \* coef - rad\_j, x\_scr0 + x \* coef + rad\_j, y\_scr0 + y4 \* coef + rad\_j, fill='brown')  
saturn = c.create\_oval(x\_scr0 + x \* coef - rad\_sat, y\_scr0 + y5 \* coef - rad\_sat, x\_scr0 + x \* coef + rad\_sat, y\_scr0 + y5 \* coef + rad\_sat, fill='gray')  
uran = c.create\_oval(x\_scr0 + x \* coef - rad\_u, y\_scr0 + y6 \* coef - rad\_u, x\_scr0 + x \* coef + rad\_u, y\_scr0 + y6 \* coef + rad\_u, fill='dark blue')  
neptun = c.create\_oval(x\_scr0 + x \* coef - rad\_n, y\_scr0 + y7 \* coef - rad\_n, x\_scr0 + x \* coef + rad\_n, y\_scr0 + y7 \* coef + rad\_n, fill='light blue')  
  
kometa = c.create\_oval(x\_scr0 + xk \* coef - r\_k, y\_scr0 + yk \* coef - r\_k, x\_scr0 + xk \* coef + r\_k, y\_scr0 + yk \* coef + r\_k, fill='gray')  
  
#движение планет  
def motion():  
 global vx, vy, x, y, vx1, vy1, x1, y1, vx2, vy2, x2, y2, vx3, vy3, x3, y3, vx4, vy4, x4, y4, vx5, vy5, x5, y5, vx6, vy6, x6, y6, vx7, vy7, x7, y7, vxk, vyk, xk, yk  
  
 r = (x \* x + y \* y)\*\*0.5 #ур-я движения для меркурия  
 ax = -GM \* x / r / r / r  
 ay = -GM \* y / r / r / r  
 vx += ax \* dt  
 vy += ay \* dt  
 x += vx \* dt  
 y += vy \* dt  
 dx = vx \* dt  
 dy = vy \* dt  
  
 r1 = (x1 \* x1 + y1 \* y1) \*\* 0.5 #ур-я движения для венеры  
 ax1 = -GM \* x1 / r1 / r1 / r1  
 ay1 = -GM \* y1 / r1 / r1 / r1  
 vx1 += ax1 \* dt  
 vy1 += ay1 \* dt  
 x1 += vx1 \* dt  
 y1 += vy1 \* dt  
 dx1 = vx1 \* dt  
 dy1 = vy1 \* dt  
  
 r2 = (x2 \* x2 + y2 \* y2) \*\* 0.5 #ур-я движения для земли  
 ax2 = -GM \* x2 / r2 / r2 / r2  
 ay2 = -GM \* y2 / r2 / r2 / r2  
 vx2 += ax2 \* dt  
 vy2 += ay2 \* dt  
 x2 += vx2 \* dt  
 y2 += vy2 \* dt  
 dx2 = vx2 \* dt  
 dy2 = vy2 \* dt  
  
 r3 = (x3 \* x3 + y3 \* y3) \*\* 0.5 #ур-я движения для марса  
 ax3 = -GM \* x3 / r3 / r3 / r3  
 ay3 = -GM \* y3 / r3 / r3 / r3  
 vx3 += ax3 \* dt  
 vy3 += ay3 \* dt  
 x3 += vx3 \* dt  
 y3 += vy3 \* dt  
 dx3 = vx3 \* dt  
 dy3 = vy3 \* dt  
  
 r4 = (x4 \* x4 + y4 \* y4) \*\* 0.5 #ур-я движения для юпитера  
 ax4 = -GM \* x4 / r4 / r4 / r4  
 ay4 = -GM \* y4 / r4 / r4 / r4  
 vx4 += ax4 \* dt  
 vy4 += ay4 \* dt  
 x4 += vx4 \* dt  
 y4 += vy4 \* dt  
 dx4 = vx4 \* dt  
 dy4 = vy4 \* dt  
  
 r5 = (x5 \* x5 + y5 \* y5) \*\* 0.5 #ур-я движения для сатурна  
 ax5 = -GM \* x5 / r5 / r5 / r5  
 ay5 = -GM \* y5 / r5 / r5 / r5  
 vx5 += ax5 \* dt  
 vy5 += ay5 \* dt  
 x5 += vx5 \* dt  
 y5 += vy5 \* dt  
 dx5 = vx5 \* dt  
 dy5 = vy5 \* dt  
  
 r6 = (x6 \* x6 + y6 \* y6) \*\* 0.5 #ур-я движения для урана  
 ax6 = -GM \* x6 / r6 / r6 / r6  
 ay6 = -GM \* y6 / r6 / r6 / r6  
 vx6 += ax6 \* dt  
 vy6 += ay6 \* dt  
 x6 += vx6 \* dt  
 y6 += vy6 \* dt  
 dx6 = vx6 \* dt  
 dy6 = vy6 \* dt  
  
 r7 = (x7 \* x7 + y7 \* y7) \*\* 0.5 #ур-я движения для нептуна  
 ax7 = -GM \* x7 / r7 / r7 / r7  
 ay7 = -GM \* y7 / r7 / r7 / r7  
 vx7 += ax7 \* dt  
 vy7 += ay7 \* dt  
 x7 += vx7 \* dt  
 y7 += vy7 \* dt  
 dx7 = vx7 \* dt  
 dy7 = vy7 \* dt  
  
 rk = (xk\*\*2 + yk\*\*2) \*\* 0.5 # ур-я движения для кометы  
 axk = -GM \* xk / rk / rk / rk  
 ayk = -GM \* yk / rk / rk / rk  
 vxk += axk \* dt  
 vyk += ayk \* dt  
 xk += vxk \* dt  
 yk += vyk \* dt  
 dxk = vxk \* dt  
 dyk = vyk \* dt  
  
 #двигаем планеты  
 c.move(merkury, dx \* coef, dy \* coef)  
 c.move(venera, dx1 \* coef, dy1 \* coef)  
 c.move(earth, dx2 \* coef, dy2 \* coef)  
 c.move(mars, dx3 \* coef, dy3 \* coef)  
 c.move(jupiter, dx4 \* coef, dy4 \* coef)  
 c.move(saturn, dx5 \* coef, dy5 \* coef)  
 c.move(uran, dx6 \* coef, dy6 \* coef)  
 c.move(neptun, dx7 \* coef, dy7 \* coef)  
  
 c.move(kometa, dxk \* coef, dyk \* coef)  
  
 root.after(20, motion)  
  
  
motion()  
root.mainloop()

**Результат:**

