Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
Московский государственный технический университет имени Н.Э. Баумана

Лабораторная работа №1
«Решение баллистической задачи»
по курсу:
«Моделирование»

Выполнил:
студент группы ИУ9-82
Иванов Георгий

Проверила:
Домрачева А.Б.

Москва, 2019

Содержание

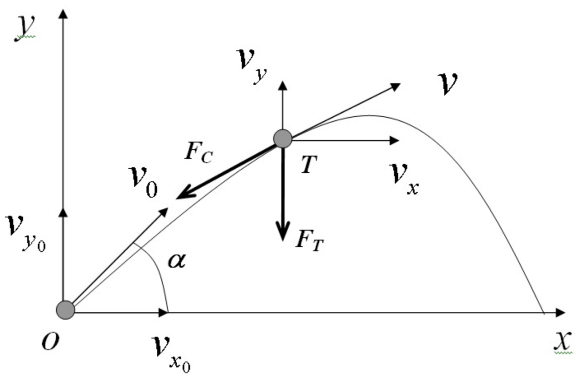
1 Постановка задачи

Необходимо определить дальность полёта снаряда, брошенного под углом α к поверхности Земли с некоторой начальной скоростью v_0. Рассчитывая методом Ньютона, необходимо учесть сопротивление воздуха (плотность $\rho = 1.29\frac{\text{кг}}{\text{м}^3}$. Сравнить результат для данного метода с сопротивлением равным нулю, то есть пренебрегая им, и методом Галилея. Ускорение свободного падения примем за константу $g = 9.8\frac{\text{м}}{\text{с}}$

В качестве начальных данных имеем: свинцовый шарик $\rho = 11340\frac{\text{кг}}{\text{м}^3}$ диаметром 0,1 м брошен под углом $\alpha = 45$ градусов с начальной скоростью $v_0 = 10\frac{\text{м}}{\text{с}}$

2 Теоретические сведения

2.1 Модель Галилея



Модель Галилея является упрощенной моделью для решения баллистической задачи. В этой модели мы только учитываем силу тяжести. В модели Галилея на входе переменная x, а на выходе переменная y, параметром системы будет являться g (ускорение свободного падения).

Распишем основные уравнения:

$$\begin{cases} x = (v_0 cos(\alpha))t \\ y = (v_0 sin(\alpha) - \dfrac{gt}{2})t \end{cases} . \tag{1}$$

Для нахождения уравнения зависимости выразим t и подставим во второе уравнение. Тогда получим:

$$\begin{cases} t = \dfrac{x}{v_0 cos(\alpha)} \\ y = xtg(\alpha) - \dfrac{gx^2}{2v_0 cos^2(\alpha)} \end{cases} . \tag{2}$$

2.2 Модель Ньютона

Основное отличие модели Ньютона от модели Галилея заключается в учитывании сопротивления воздуха. Сила сопротивления вохдуха рассчитывается по формуле $F_c = \beta v^2$, коэффициент которой вычисляется по формуле:

$$\beta = \frac{C\rho S}{2}, \tag{3}$$

где C - баллистический коэффициент ($C \approx 0.15$), S – площадь поперечного сечения снаряда ($S = \pi r^2$), ρ - плотность воздуха.

При $\beta \equiv 0$ модель Ньютона оказывается частным случаем методом Галилея.

Составляющие равнодействующей всех сил по осям X и Y равны:

$$\begin{cases} F_x = -\beta v_x \sqrt{v_x^2 + v_y^2} \\ F_y = -\beta v_y \sqrt{v_x^2 + v_y^2} - mg \\ \dfrac{dx}{dt} = v_x \\ \dfrac{dy}{dt} = v_y \end{cases}, \tag{4}$$

Применяя 2-й закон Ньютона, мы получаем систему, которая описывает движение тела, брошенного под углом к горизонту с учетом сопротивления воздуха. Это является моделью Ньютона.

$$\begin{cases} \dfrac{dv_x}{dt} = -\dfrac{\beta v_x}{m} \sqrt{v_x^2 + v_y^2} \\ \dfrac{dv_y}{dt} = -\dfrac{\beta v_y}{m} \sqrt{v_x^2 + v_y^2} - g \\ \dfrac{dx}{dt} = v_x \\ \dfrac{dy}{dt} = v_y \end{cases}, \tag{5}$$

Эта система должна быть дополнена начальными условиями:

$$\begin{cases} v_x(0) = v_0 cos(\alpha) \\ v_y(0) = v_0 sin(\alpha) \\ x(0) = 0 \\ y(0) = 0 \end{cases}. \tag{6}$$

Для численного решения системы дифференциальных уравнений воспользуемся методом Рунге-Кутта 4-го порядка, обеспечивающий достаточно высокую точность вычислений.

3 Практическая реализация

Ниже приведена реализация программы на языке Python 3, которая строит графики зависимости координаты y от координаты x в определенный промежуток времени для метода Галилея и методов Ньютона с учётом коэффициента сопротивления и без его учёта (Листинг 1).

```
#!python -v
# -*- coding: utf-8 -*-

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

```

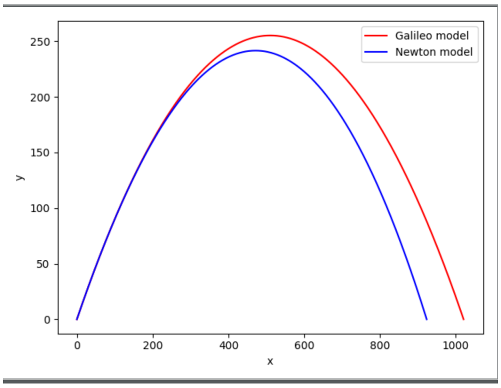
```python
C = 0.15  # баллистический коэффициент

rho_lead = 11340  # kg/m^3 - плотность меди
rho_air = 1.29  # kg/m^3 - плотность воздуха

v0_1 = 10  # начальная скорость

diam = 0.5  # диаметр шарика
rad = diam / 2  # радиус шарика
alpha = math.pi / 4  # угол в радианах

t_0 = 0  # время начала
t_max = 100  # время конца

beta = 0  # C * rho_air * S / 2
S = math.pi * (rad ** 2)  # площадь поперечного сечения
V = (4 / 3) * math.pi * (rad ** 3)  # объем шара
m = rho_lead * V  # масса шара

v0 = 150  # тестирование начальной скорости

eps = 1.e-4
g = 9.8  # m/sec^2

# Начальные условия
u0 = v0 * math.cos(alpha)
w0 = v0 * math.sin(alpha)
x0 = 0
y0 = 0


# функции по модели Галилея:
def x(t):
    return v0 * math.cos(alpha) * t


def y(t):
    return v0 * math.sin(alpha) * t - g * (t ** 2) / 2


# описание системы модели Ньютона:
def right_part(t, system):
    (u, w, x, y) = system
    factor = -beta * math.sqrt(u ** 2 + w ** 2) / m
    return np.ndarray((4,), buffer=np.array([u * factor, w * factor - g, u, w]))

```

```
# начальные условия
def init_system():
    return np.ndarray((4,), buffer=np.array([u0, w0, x0, y0]))


coords = solve_ivp(right_part, (t_0, t_max), init_system(), max_step=eps)
t_arr = coords['t']
coords = coords['y'][2:]


# удаляем все точки, где x < 0
def trim(arr):
    M = np.where(arr[1] >= 0)[-1][-1]
    return arr[:M, :M]


coords = trim(coords)

xvals, yvals = coords
gal_xvals = [x(t) for t in t_arr]
gal_yvals = [y(t) for t in t_arr]
gal_coords = np.ndarray((2, len(gal_xvals)), buffer=np.array([gal_xvals, gal_yvals]))
gal_coords = trim(gal_coords)
gal_xvals, gal_yvals = gal_coords

# Построение графика
plt.plot(gal_xvals, gal_yvals, 'r-', label='Galileo model')
plt.plot(xvals, yvals, 'b', label='Newton model')
plt.legend()
plt.ylabel('y')
plt.xlabel('x')
plt.show()
```

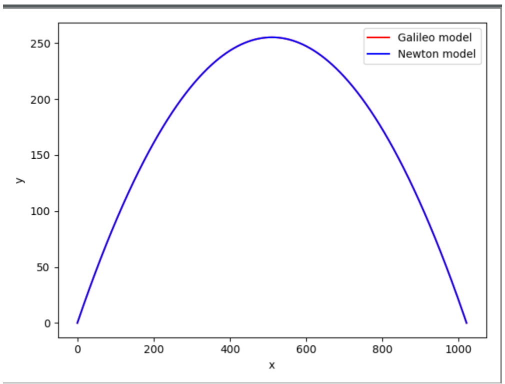
4 Результаты

Для начала разберем случай модели Ньютона, когда $\beta \neq 0$.

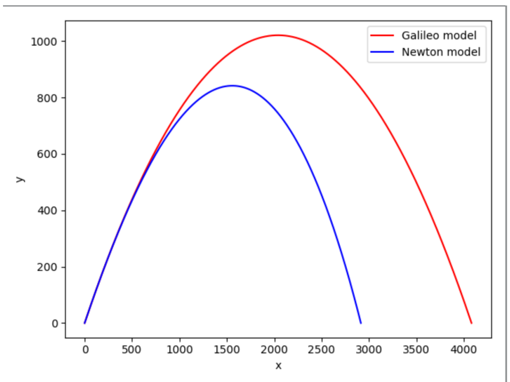
При $v_0 = 100$ и $\beta = 0.1$ координаты точки падения в модели Галилея: (x, y) = (1020.3985116434825, 0.009651530532778452), а в модели Ньютона: (x, y) = (923.052998696859, 0.008128764647009045). Графический результат работы программы приведен на рисунке ниже.



Далее разберем случай модели Ньютона, когда $\beta = 0$ и сравним с моделью Галилея. При $v_0 = 100$ координаты точки падения в модели Галилея: (x, y) = (1020.3985116434825, 0.009651530532778452) а в модели Ньютона: (x, y) = (1020.3985116411366, 0.00965152986713986).
Графический результат работы программы приведен на рисунке ниже.



Далее увеличим начальную скорость и разберем случай модели Ньютона, когда $\beta \neq 0$ и сравним с моделью Галилея. При $v_0 = 200$ и $\beta = 0.1$ координаты точки падения в модели Галилея: (x, y) = (4081.6151886864536, 0.017464300044139236), а в модели Ньютона: (x, y) = (2914.48262197702, 0.014622545483177918).
Графический результат работы программы приведен на рисунке ниже.



5 Вывод

В ходе выполнения лабораторной работы были изучены модели Ньютона и Галилея для решения баллистической задачи и была написана их реализация на языке Python 3.

Как модель Галилея, так и модель Ньютона позволяют описать движение тела, брошенного под углом к горизонту, однако модель Ньютона дает более точный результат, так как учитывает силу сопротивления воздуха. При одинаковых значениях параметров дальность полёта по модели Галилея больше, чем у модели Ньютона в силу того, что модель Ньютона учитывает сопротивление воздуха, которе направлена против скорости.

Кроме того, при $\beta = 0$ модель Галилея является частным случаем модели Ньютона.