|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Теоретическая информатика и компьютерные технологии

**Отчет по лабораторной работе № 1**

***по курсу «Моделирование»***

**«Решение баллистической задачи»**

Студент ИУ9-81Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Костриця М.И.

(Группа) (Подпись, дата) (Фамилия И.О.)

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Домрачева А. Б. (Подпись, дата) (Фамилия И.О.)

*Москва, 2023 г.*

# Цель работы

Целью данной лабораторной работы является определение дальности полёта снаряда, брошенного под углом с начальной скоростью :

1. Методом Галилея
2. Методом Ньютона с сопротивлением воздуха

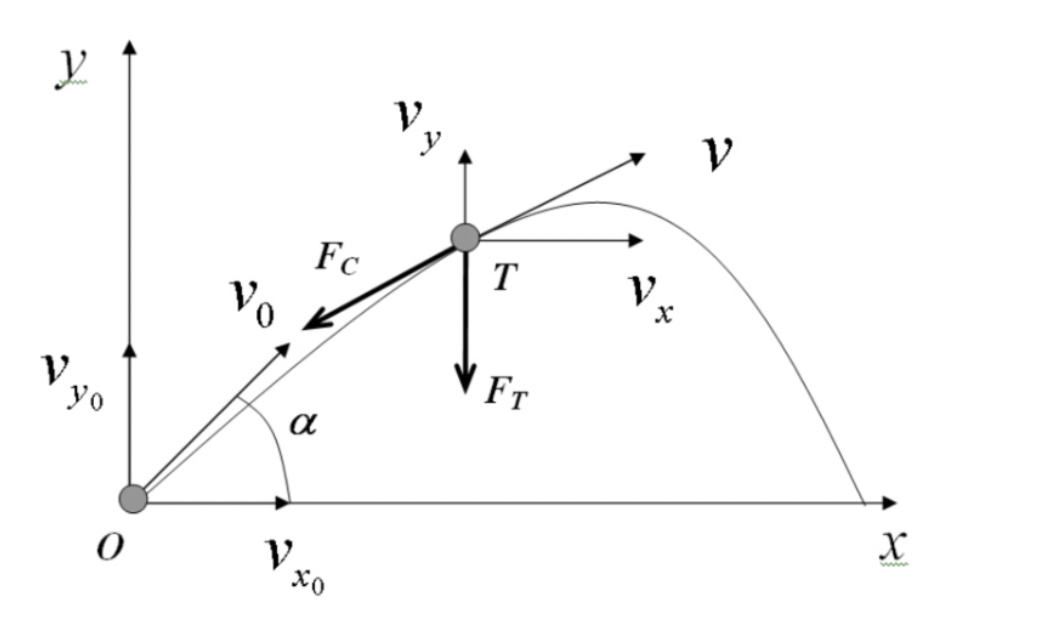
# Постановка задачи

Дано:

1. Плотность воздуха
2. Ускорение свободного падения
3. Железный шарик:
4. Угол броска

# Теоретическая часть

* 1. **Модель Галилея**

****

Модель Галилея является упрощенной моделью для решения баллистической задачи. В этой модели мы только учитываем силу тяжести. Основные уравнения:

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

* 1. **Модель Ньютона**

Основное отличие модели Ньютона от модели Галилея заключается в учитывании сопротивления воздуха. Сила сопротивления воздуха рассчитывается по формуле . вычисляется по формуле:

где - баллистический коэффициент , – площадь поперечного сечения снаряда , - плотность воздуха.

При модель Ньютона оказывается частным случаем методом Галилея. Составляющие равнодействующей всех сил по осям X и Y равны:

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.

Система должна быть дополнена начальными условиями:

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

1. **Практическая реализация**

from scipy.integrate import solve\_ivp

import numpy as np

import math

from matplotlib import pyplot as plt

#9 вариант

C = 0.15 # баллистический коэффициент

rho\_lead = 7874 # kg/m^3 - плотность железа

rho\_air = 1.29 # kg/m^3 - плотность воздуха

v0\_1 = 10 # начальная скорость

rad = 0.09 # радиус шарика

t\_0 = 0 # время начала

t\_max = 100 # время конца

S = math.pi \* (rad \*\* 2) # площадь поперечного сечения

beta = C \* rho\_air \* S / 2

V = (4 / 3) \* math.pi \* (rad \*\* 3) # объем шара

m = rho\_lead \* V # масса шара

eps = 1.e-2

g = 9.8 # m/sec^2

# Начальные условия

# Галилей:

def x(t, alph, v00):

return v00 \* math.cos(alph) \* t

def y(t, alph, v00):

return v00 \* math.sin(alph) \* t - g \* (t \*\* 2) / 2

# Ньютон:

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]))

# удаляем все точки, где x < 0

def trim(arr):

M = np.where(arr[1] >= 0)[-1][-1]

return arr[:M, :M]

if \_\_name\_\_ == "\_\_main\_\_":

alpha = math.radians(34) # угол в радианах

v0 = 125 # тестирование начальной скорости

data\_newton = []

data\_gal = []

for speed in np.arange(10, 125.0, 10.0):

v0 = speed

u0 = v0 \* math.cos(alpha)

w0 = v0 \* math.sin(alpha)

x0 = 0

y0 = 0

coords = solve\_ivp(right\_part, (t\_0, t\_max), np.ndarray((4,), buffer=np.array([u0, w0, x0, y0])), max\_step=eps)

#print(coords)

t\_arr = coords['t']

coords = coords['y'][2:]

coords = trim(coords)

xvals, yvals = coords

gal\_xvals = [x(t, alpha, v0) for t in t\_arr]

gal\_yvals = [y(t, alpha, v0) 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

data\_gal.append((gal\_xvals, gal\_yvals))

data\_newton.append((xvals, yvals))

#print("Galilei: (", gal\_xvals[-1], ", ", gal\_yvals[-1], ")", sep='')

#print("Newton : (", xvals[-1], ", ", yvals[-1], ")", sep='')

#plt.plot(gal\_xvals, gal\_yvals, 'r-', label='Galileo model')

#plt.plot(xvals, yvals, 'b', label='Newton model')

for i in range(len(data\_newton)):

label\_1 = 'Galileo model'

label\_2 = 'Newton model'

if i != 0:

label\_1 =''

label\_2 =''

plt.plot(data\_gal[i][0], data\_gal[i][1], 'r-', label=label\_1)

plt.plot(data\_newton[i][0], data\_newton[i][1], 'b', label=label\_2)

plt.legend()

plt.ylabel('y')

plt.xlabel('x')

plt.show()

1. **Результаты**

Для начала разберем случай модели Ньютона, когда . Графический результат работы программы приведен на рисунке ниже. При

****

Далее разберем случай модели Ньютона, когда 𝛽 = 0 при условии (1) и сравним с моделью Галилея.

****

****Рассмотрим, как изменяется дистанция полета в зависимости от угла в двух моделях

График полета шарика при условии:



****Рассмотрим, как изменяется высота полета в зависимости от угла в двух моделях

График изменения высоты полета в зависимости от скорости при угле в двух моделях

1. **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 сопротивления воздуха.

Выявлены следующие зависимости:

1) Дальности полета при разных скоростях и фиксированном угле

2) Дальности полета при разных углах и фиксированной скорости

3) Максимальной высоты полета при разных скоростях и фиксированном угле

4) Максимальной высоты полета при разных углах и фиксированной скорости