

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА  
Факультет информатики и систем управления  
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №1  
по курсу «Математическое моделирование»

«Построение динамической модели на примере баллистической задачи  
(модель Галилея, модель Ньютона)»

Выполнил:  
студент ИУ9-91  
Выборнов А. И.  
  
Руководитель:  
Домрачева А.Б.

Москва 2015

# 1. Постановка задачи

Моделировать движение артиллерийского снаряда. Выстрел произведен с начальной скоростью  $v_0 = 50\text{м/с}$ , под углом к горизонту  $\alpha = \pi/4$ . Считать, что снаряд изготовлен из свинца, имеет форму шара с радиусом  $r = 0.1\text{м}$ . Построить траекторию полета снаряда, указать точку падения снаряда и время полета.

Необходимо рассмотреть решения данной задачи с помощью моделей Галилея и Ньютона, а также проанализировать полученные результаты.

## 2. Теоретическая часть

Предположим снаряд вылетел из точки  $(0, 0)$ . Ось  $x$  направлена горизонтально, ось  $y$  вертикально.

### 2.1. Модель Галилея

В модели Галилея на тело действует только сила тяжести. Подобная задача решалась в рамках школьного курса физики следующим образом:

$$\begin{cases} v_x = v_0 \cos(\alpha) \\ v_y = v_0 \sin(\alpha) \\ x(t) = v_x t \\ y(t) = v_y t - \frac{gt^2}{2} \end{cases}$$

С помощью несложных преобразований можно получить зависимость времени и координаты  $y$  от координаты  $x$ :

$$\begin{cases} t(x) = \frac{x}{v_x} \\ y(x) = \frac{v_y x}{v_x} - \frac{gx^2}{2v_x^2} \end{cases}$$

Начиная с  $x = 0$  можно инкрементировать  $x$  на  $dx$ . Если график  $y(x)$  снова пересечёт ось  $x$ , то есть  $y(x) = 0$ , будем считать полученный  $x$  местом падения. Время полёта  $t(x)$  можно посчитать по приведённой выше формуле.

## 2.2. Модель Ньютона

В отличие от модели Галилея модель Ньютона учитывает силу сопротивления воздуха  $F_C = -\beta v^2$ , где  $\beta = 0.5CS\rho$  ( $C = 0.15$  — коэффициент аэродинамического сопротивления,  $S = \pi r^2$  — площадь поперечного сечения,  $\rho = 1.29 \text{ кг/м}^3$  — плотность воздуха).

Чтобы найти точку падения тела с помощью модели Галилея необходимо решить следующую систему:

$$\begin{cases} \frac{dv_x}{dt} = -\frac{\beta}{m}v_x\sqrt{v_x^2 + v_y^2} \\ \frac{dv_y}{dt} = -g - \frac{\beta}{m}v_y\sqrt{v_x^2 + v_y^2} \\ v_x(0) = v_0\cos(\alpha) \\ v_y(0) = v_0\sin(\alpha) \end{cases}$$

Решив эту систему дифференциальных уравнений получим зависимость скорости от времени. Ищем зависимость координат  $x$  и  $y$  от времени:

$$\begin{cases} x(t) = \int_0^t v_x(t)dt \\ y(t) = \int_0^t v_y(t)dt \end{cases}$$

Необходимо найти  $t$  при котором  $y(t) = 0$  и  $x(t) \neq 0$ . Получим время полёта снаряда.

Можно численно решать систему дифференциальных уравнений с помощью метода Рунге-Кутты. Данный метод итеративен и позволяет последовательно получать  $v_x(t_i)$ ,  $v_y(t_i)$ , где  $t_i = i * dt, i = 1, 2, \dots$ . Получив зависимость скорости от времени можно найти координаты:

$$\begin{cases} x(t_i) = \sum_{t=0}^{t_i} v_x(t) \\ y(t_i) = \sum_{t=0}^{t_i} v_y(t) \end{cases}$$

То есть искомая задача сводиться к получению скорости для каждого последовательного момента времени, до тех пор пока  $y(t_i) \neq 0$  не станет равен нулю. Полученное время также однозначно задаёт координату падения.

## 3. Реализация

В рамках лабораторной работы рассматривались обе модели (в рамках модели Ньютона был также рассмотрен случай когда  $C = 0$ ). Была написана программа на

языке python, которая решает данную задачу и визуализирует результаты. Полная версия исходного кода содержится в файле *lab1\_ball.py*

### 3.1. Модель Галилея

Ниже представлен код на Python, реализующий модель Галилея:

```
vx, vy = v0*cos(alpha), v0*sin(alpha)

t = lambda x: x / vx
y = lambda x: vy * x / vx - g * x**2 / (2 * vx**2)

x = 0
while y(x) >= 0:
    x += dx
```

После его выполнения координата падения —  $x$ , а время полёта —  $t(x)$ .

### 3.2. Модель Ньютона

Ниже представлен код на Python, реализующий модель Ньютона:

```
vx, vy = v0*cos(alpha), v0*sin(alpha)

dvx_dt = lambda vy, vx: - beta * vx * sqrt(vx**2+vy**2) / m
dvy_dt = lambda vx, vy: - g - beta * vy * sqrt(vx**2+vy**2) / m

x, y = 0, 0
i = 0

while y >= 0:
    vx, vy = runge_kutta_iteration(dvx_dt, vy, vx, dt), \
              runge_kutta_iteration(dvy_dt, vx, vy, dt)
    x += vx*dt
    y += vy*dt

    i += 1
```

После его выполнения координата падения —  $x$ , а время полёта —  $i * dt$ .

## 4. Результаты

Было проведено три эксперимента. Вычисления производились для значений  $dx = 10^{-3}$  и  $dt = 10^{-3}$ .

- Galilei — модель Галилея (координата  $x$  падения снаряда — 254.93 м., время полёта — 7.21 с.),
- Newton — модель Ньютона (координата  $x$  падения снаряда — 251.87 м., время полёта — 7.19 с.),
- Newton without drag — модель Ньютона с  $C = 0$  (координата  $x$  падения снаряда — 254.91 м., время полёта — 7.21 с.).

На рисунке 1 показаны траектории полёта снаряда, которые рассчитаны с помощью различных моделей, описанных выше. Также на рисунке 2 показаны траектории перед падением снарядов.

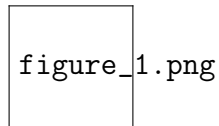


Рисунок 1 — Траектории полёта снарядов для различных моделей

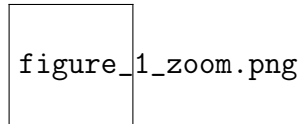


Рисунок 2 — Траектории полёта снарядов для различных моделей

## 5. Анализ результатов

Результаты обеих моделей без учёта сопротивления воздуха совпали, что говорит о корректности этих моделей. Как и ожидалось, сопротивление воздуха сократило дальность полёта снаряда.