

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования «Крымский федеральный университет им. В.И. Вернадского»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

**«Численное моделирование движения тела, брошенного под углом к  
горизонту, в среде с сопротивлением»**

обязательное задание по дисциплине

«Основы компьютерного моделирования»

студента 2 курса группы ПИ-б-о-201(1)

Шенгелай Всеволода Михайловича

направления подготовки 09.03.04 «Программная инженерия»

Кандидат технических наук,  
доцент, заведующий кафедрой  
компьютерной инженерии и  
моделирования

90  
(оценка)



(подпись, дата)

Милюков В.В.

Симферополь, 2021

## **Цели работы**

- Изучить и реализовать приемы численного интегрирования уравнений Ньютона на примере компьютерного моделирования движения тел в среде с сопротивлением;
- Изучить способы верификации компьютерных моделей на основании физических законов и аналитических моделей;
- Повысить навыки создания графических приложений на Python, изучить интеграцию графиков библиотеки Matplotlib в программу с графическим интерфейсом Tkinter.

### **Перед выполнением лабораторной работы:**

1. Были изучены теоретические сведения в методических указаниях к выполнению данной лабораторной работы; подробно рассмотрены приведенные практические примеры.
2. Прочитан соответствующий материал в электронном конспекте лектора по данному курсу.

**Представлена 1 программа в электронной форме, продемонстрирована её работоспособность.**

Программы, в которой производились все расчеты, были написаны на языке Python 3 в интерактивной среде разработки IDLE (Python 3.6 32-bit). Технические характеристики рабочего компьютера следующие: процессор Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz, 8 ГБ установленной оперативной памяти. Операционная система - Windows 10.

## Ход работы

### 1. Аналитическое исследование

Введём прямоугольную систему координат так, как показано на рисунке. В начальный момент времени тело массой  $m$  находится в начале координат. Вектор ускорения свободного падения  $\vec{g}$  направлен вертикально вниз и имеет координаты  $(0, -g)$ .

$\vec{v}_0$  - вектор начальной скорости. Разложим этот вектор по базису:  $\vec{v}_0 = v_{0x}\vec{i} + v_{0y}\vec{j}$ . Здесь  $v_{0x} = v_0 \cos \alpha$ ,  $v_{0y} = v_0 \sin \alpha$ , где  $v_0 = |\vec{v}_0|$  - модуль вектора скорости,  $\alpha$  - угол бросания.

Запишем второй закон Ньютона: 
$$\vec{a} = \frac{\vec{F}}{m}.$$

Ускорение в каждый момент времени есть (мгновенная) скорость изменения скорости, то есть производная от скорости по времени: 
$$\vec{a} = \frac{d\vec{v}}{dt}.$$

Следовательно, 2-й закон Ньютона можно переписать в следующем виде:  $m \frac{d\vec{v}}{dt} = \vec{F}$ , где  $\vec{F}$  - это равнодействующая всех сил, действующая на тело.

Так как на тело действуют сила тяжести и сила сопротивления воздуха, то

$$m \frac{d\vec{v}}{dt} = m\vec{g} + \vec{F}_c.$$

Мы будем рассматривать три случая:

1) Сила сопротивления воздуха равна 0:  $\vec{F}_c = 0$ .

2) Сила сопротивления воздуха противоположно направлена с вектором скорости, и её величина пропорциональна скорости:  $\vec{F}_c = -k\vec{v}, k > 0$ .

3) Сила сопротивления воздуха противоположно направлена с вектором скорости, и её величина пропорциональна квадрату скорости:  $\vec{F}_c = -k|\vec{v}|\vec{v}, k > 0$ .

Вначале рассмотрим 1-й случай.

В этом случае  $m \frac{d\vec{v}}{dt} = m\vec{g}$ , или  $\frac{d\vec{v}}{dt} = \vec{g}$ .

Запишем это равенство в скалярном виде:

$$\begin{cases} \frac{dv_x}{dt} = 0 \\ \frac{dv_y}{dt} = -g \end{cases}$$

Из  $\frac{d\vec{v}}{dt} = \vec{g}$  следует, что  $\vec{v} = \vec{v}_0 + \vec{g}t$  (равноускоренное движение).

Так как  $\vec{v} = \frac{d\vec{r}}{dt}$  ( $r$  - радиус-вектор), то  $\frac{d\vec{r}}{dt} = \vec{v}_0 + \vec{g}t$ .

Отсюда  $\vec{r} = \vec{r}_0 + \vec{v}_0 t + \frac{\vec{g}t^2}{2}$ .

Эта формула есть не что иное, как формула закона движения тела при равноускоренном движении.

Так как  $\vec{r}_0 = 0$ , то  $\vec{r} = \vec{v}_0 t + \frac{\vec{g} t^2}{2}$ .

Учитывая, что  $\vec{v}_0 = \{v_0 \cos \alpha, v_0 \sin \alpha\}$  и  $\vec{g} = \{0, -g\}$ , получаем из последнего векторного равенства скалярные равенства:

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

Проанализируем полученные формулы.  
Найдём *время полёта* тела. Приравняв  $y$  к нулю, получим

$$t_0 = \frac{2v_0 \sin \alpha}{g}$$

*Дальность полёта* равна значению координаты  $x$  в момент времени  $t_0$ :

$$L = \frac{v_0^2 \sin 2\alpha}{g}$$

Из этой формулы следует, что максимальная дальность полёта достигается при  $\alpha = 45^\circ$ .

Теперь найдём *уравнение траектории тела*. Для этого выразим  $t$  через  $x$

$$t = \frac{x}{v_0 \cos \alpha}$$

и подставим полученное выражение для  $t$  в равенство для  $y$ .

$$y = x \operatorname{tg} \alpha - \frac{gx^2}{2v_0^2 \cos^2 \alpha}$$

Полученная функция  $y(x)$  -- квадратичная функция, её графиком является парабола, ветви которой направлены вниз.

Теперь рассмотрим второй случай:  $\vec{F}_c = -k\vec{v}$ ,  $k > 0$ .

Второй закон приобретает вид  $m \frac{d\vec{v}}{dt} = m\vec{g} - k\vec{v}$ ,

отсюда  $\frac{d\vec{v}}{dt} = \vec{g} - \frac{k}{m}\vec{v}$ .

Запишем это равенство в скалярном виде:

$$\begin{cases} \frac{dv_x}{dt} = -\frac{k}{m}v_x \\ \frac{dv_y}{dt} = -g - \frac{k}{m}v_y \end{cases}$$

Мы получили два линейных дифференциальных уравнения.

Первое уравнение имеет решение

$$v_x(t) = v_{0x}e^{-\frac{k}{m}t}$$

в чём можно убедиться, подставив данную функцию в уравнение для  $v_x$  и в начальное условие  $v_x(0) = v_{0x}$ .

Здесь  $e = 2,718281828459...$  – число Эйлера.

Второе уравнение имеет решение

$$v_y(t) = v_{0y}e^{-\frac{k}{m}t} - \frac{gm}{k} \left(1 - e^{-\frac{k}{m}t}\right)$$

Так как  $\lim_{t \rightarrow +\infty} v_x(t) = 0$ ,  $\lim_{t \rightarrow +\infty} v_y(t) = -\frac{gm}{k}$ , то при наличии сопротивления воздуха движение тела стремится к равномерному, в отличие от случая 1, когда скорость неограниченно увеличивается. В следующем видеоролике говорится, что парашютист сначала движется ускоренно, а потом начинает двигаться равномерно (даже до раскрытия парашюта).

Найдём выражения для  $x$  и  $y$ .

Так как  $x(0) = 0$ ,  $y(0) = 0$ , то

$$x(t) = \int_0^t v_x(\tau) d\tau, \quad y(t) = \int_0^t v_y(\tau) d\tau$$

Отсюда

$$\begin{cases} x = v_{0x} \frac{m}{k} \left( 1 - e^{-\frac{k}{m}t} \right) \\ y = \frac{m}{k} \left[ \left( v_{0y} + \frac{mg}{k} \right) \left( 1 - e^{-\frac{k}{m}t} \right) - gt \right] \end{cases}$$

Нам осталось рассмотреть случай 3, когда  $\vec{F}_c = -k|\vec{v}|\vec{v}$ ,  $k > 0$ .

Второй закон Ньютона имеет вид

$$m \frac{d\vec{v}}{dt} = m\vec{g} - k|\vec{v}|\vec{v}, \quad \text{или} \quad \frac{d\vec{v}}{dt} = \vec{g} - \frac{k}{m}|\vec{v}|\vec{v}.$$

В скалярном виде это уравнение имеет вид:

$$\begin{cases} \frac{dv_x}{dt} = -\frac{k}{m} \sqrt{v_x^2 + v_y^2} \cdot v_x \\ \frac{dv_y}{dt} = -g - \frac{k}{m} \sqrt{v_x^2 + v_y^2} \cdot v_y \end{cases}$$

Это *система нелинейных дифференциальных уравнений*. Данную систему не удаётся решить в явном виде, поэтому необходимо применять численное моделирование.

## 2. Численное исследование

### 2.1 Для модели, в которой величина силы сопротивления пропорциональна скорости тела

В предыдущем разделе мы увидели, что в первых двух случаях закон движения тела можно получить в явном виде. Однако в третьем случае необходимо решать задачу численно. При помощи численных методов мы получим лишь приближённое решение, но нас вполне устроит и небольшая точность. (Число  $\pi$  или квадратный корень из 2, кстати, нельзя записать абсолютно точно, поэтому при расчётах берут какое-то конечное число цифр, и этого вполне хватает).

Будем рассматривать второй случай, когда сила сопротивления воздуха определяется формулой  $\vec{F}_c = -k\vec{v}$ ,  $k > 0$ . Отметим, что при  $k = 0$  получаем первый случай.

Скорость тела  $\vec{v} = \{v_x, v_y\}$  подчиняется следующим уравнениям:



$$\begin{cases} \frac{dv_x}{dt} = -\frac{k}{m}v_x \\ \frac{dv_y}{dt} = -g - \frac{k}{m}v_y \end{cases}$$

В левых частях этих уравнений записаны компоненты

$$a_x = \frac{dv_x}{dt}, \quad a_y = \frac{dv_y}{dt}$$

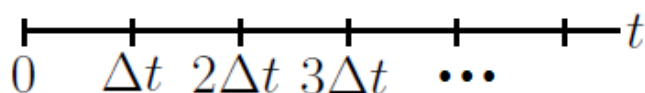
ускорения

Напомним, что ускорение есть (мгновенная) скорость изменения скорости, то есть производная от скорости по времени.

В правых частях уравнений записаны компоненты скорости. Таким образом, данные уравнения показывают, как скорость изменения скорости связана со скоростью.

Попробуем найти решения этих уравнений при помощи численных методов.

Для этого введём на временной оси *сетку*: выберем число  $\Delta t$  и будем рассматривать моменты времени вида  $j\Delta t$ :  $0, \Delta t, 2\Delta t, 3\Delta t, \dots$ .



Наша задача -- приблизительно вычислить значения  $v_x(t), v_y(t)$  в узлах сетки.

Заменим в уравнениях ускорение (*мгновенную скорость* изменения скорости) на *среднюю скорость* изменения скорости, рассматривая движение тела на промежутке времени  $\Delta t$ :

$$a_x(t) = \frac{dv_x(t)}{dt} \approx \frac{v_x(t + \Delta t) - v_x(t)}{\Delta t}$$

$$a_y(t) = \frac{dv_y(t)}{dt} \approx \frac{v_y(t + \Delta t) - v_y(t)}{\Delta t}$$

Теперь подставим полученные аппроксимации в наши уравнения.

$$\begin{cases} \frac{v_x(t + \Delta t) - v_x(t)}{\Delta t} = -\frac{k}{m}v_x(t) \\ \frac{v_y(t + \Delta t) - v_y(t)}{\Delta t} = -g - \frac{k}{m}v_y(t) \end{cases}$$

Отсюда

$$\begin{cases} v_x(t + \Delta t) = v_x(t) - \frac{k}{m}v_x(t)\Delta t \\ v_y(t + \Delta t) = v_y(t) - \left(g + \frac{k}{m}v_y(t)\right) \Delta t \end{cases}$$

Полученные формулы позволяют нам вычислить значения функций  $v_x(t)$ ,  $v_y(t)$  в следующем узле сетки, если известны значения этих функций в предыдущем узле сетки.

При помощи описанного метода мы можем получить таблицу приближённых значений компонент скорости.

Как найти закон движения тела, т.е. таблицу приближённых значений координат  $x(t)$ ,  $y(t)$  можно найти тем же образом.

Имеем

$$\begin{cases} \frac{dx}{dt} = v_x \\ \frac{dy}{dt} = v_y \end{cases}$$

Заменив мгновенную скорость на среднюю скорость на промежутке времени  $\Delta t$ , получим

$$\begin{cases} \frac{x(t + \Delta t) - x(t)}{\Delta t} \approx v_x(t) \\ \frac{y(t + \Delta t) - y(t)}{\Delta t} \approx v_y(t) \end{cases}$$

Отметим, что в правых частях уравнений можно взять полусумму значений  $v_x$  ( $v_y$ ) в точках  $t$  и  $t + \Delta t$ . Возможно, так точность будет выше.

Отсюда

$$\begin{cases} x(t + \Delta t) = x(t) + v_x(t)\Delta t \\ y(t + \Delta t) = y(t) + v_y(t)\Delta t \end{cases}$$

По этим формулам мы можем вычислить приближённые значения функций  $x(t)$  и  $y(t)$ .

## 2.2 Для модели с установившейся скоростью

Упростим расчёты, воспользовавшись методом, представленным в методическом пособии: будем полагать известным не коэффициент  $a$ , а установившуюся скорость падения тела  $U_m$ , тогда из условия компенсации силы тяжести силой вязкого трения получаем  $a * U_m^2 = mg$ , и  $a = mg/U_m^2$ .

Второй закон Ньютона в проекции на оси  $x$ ,  $y$  даёт:

$$\begin{cases} a_x = -\frac{g}{U_m^2} U * U_x \\ a_y = -\frac{g}{U_m^2} U * U_y - g \end{cases} \quad (1.1)$$

Связь между уравнениями осуществляется через выражение для модуля скорости  $U = \sqrt{U_x^2 + U_y^2}$ .

Таким образом, сила сопротивления вычисляется через силу тяжести, текущую и установившуюся скорости.

### 3. Реализация алгоритма численного моделирования на языке Python с поддержкой графического интерфейса

#### 3.1 Алгоритм для модели с коэффициентом аэродинамического сопротивления

Задаём массивы (списки)

```
listx = [], listy = []
```

Значение  $vx[j]$  равняется значению функции  $v_x(j\Delta t)$ , для других массивов аналогично.

Теперь остаётся написать цикл, внутри которого мы будем вычислять  $vx[j+1]$  через уже вычисленное значение  $vx[j]$ , и с остальными массивами то же самое. Цикл будет по  $j$  от 1 до момента, пока координата  $y$  не станет принимать отрицательные значения.

Не забываем инициализировать начальные значения  $v_x[0]$ ,  $v_y[0]$ ,  $x[0]$ ,  $y[0]$  по формулам  $v_{0x} = v_0 \cos \alpha$ ,  $v_{0y} = v_0 \sin \alpha$ ,  $x_0 = 0$ ,  $y_0 = 0$ .

```

time = 0
dt = t
y = 0
x = 0
α = a
Ux = U*math.cos(α*math.pi/180)
Uy = U*math.sin(α*math.pi/180)
F = k*U**2
Fx = F*math.cos((α+180)*math.pi/180)
Fy = F*math.sin((α+180)*math.pi/180)
ax = Fx/m
ay = (Fy/m) - g

```

Рисунок 1. Блок инициализации начальных значений переменных,  
участвующих в цикле

```

while y >= 0:
    time = time+dt
    listt.append(time)
    Ux = Ux+ax*dt
    Uy = Uy+ay*dt
    U = math.sqrt(Ux**2+Uy**2)
    listU.append(U)
    α = math.atan(Uy/Ux)*180/math.pi
    F = k*U**2
    Fx = F*math.cos((α+180)*math.pi/180)
    Fy = F*math.sin((α+180)*math.pi/180)
    ax = Fx/m
    ay = (Fy/m) - g
    a = math.sqrt(ax**2+ay**2)
    lista.append(a)
    x = x + Ux*dt
    x = round(x, 0)
    listx.append(x)
    y = y + Uy*dt
    y = round(y, 0)
    listy.append(y)
    #i = i+1
    #print(i)

```

Рисунок 2. Тело цикла, вычисляющего координаты x, y, ускорение a,  
скорость U в определённые моменты времени с шагом dt

### 3.2 Алгоритм для модели с установившейся скоростью

За основу этого метода можно взять алгоритм для модели с коэффициентом аэродинамического сопротивления. Убираем нахождение силы сопротивления  $F_c$  и её проекций на оси  $x$  и  $y$ . Ускорения для каждой из осей будем вычислять по формулам (1.1).

```
time = 0
dt = t
y = 0
x = 0
α = a
Ux = U * math.cos(α * math.pi / 180)
Uy = U * math.sin(α * math.pi / 180)

ax = 0
ay = 0
```

Рисунок 3. Блок инициализации начальных значений переменных, участвующих в цикле

```
while y >= 0:
    time = time + dt
    listt.append(time)
    Ux = Ux + ax * dt
    Uy = Uy + ay * dt
    U = math.sqrt(Ux ** 2 + Uy ** 2)
    listU.append(U)
    α = math.atan(Uy / Ux) * 180 / math.pi

    ax = -g / (Um**2) * U * Ux
    ay = -g / (Um**2) * U * Uy - g

    a = math.sqrt(ax ** 2 + ay ** 2)
    lista.append(a)
    x = x + Ux * dt
    print(x)
    x = round(x, 3)
    listx.append(x)
    y = y + Uy * dt
    print(y)
    y = round(y, 3)
    listy.append(y)
    # i = i+1
    # print(i)
```

Рисунок 4. Тело цикла, вычисляющего координаты  $x$ ,  $y$ , ускорение  $a$ , скорость  $U$  в определённые моменты времени с шагом  $dt$

## 4. Интерфейс программы

### 4.1 Обзор интерфейса программы для модели с заранее определённой установившейся скоростью тела

Интерфейс можно условно разделить на два поля: поле ввода информации для запуска алгоритма моделирования, и полотно для отображения графиков. Стартовое окно имеет следующий вид:

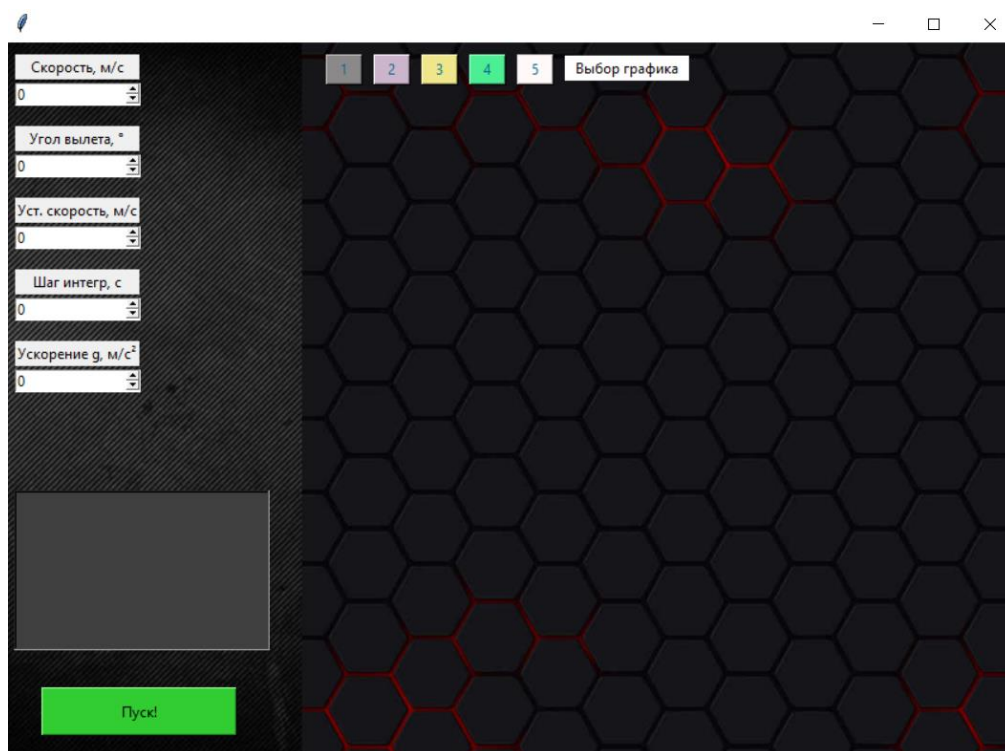


Рисунок 5. Стартовое окно программы

Для начала расчёта необходимо указать следующие параметры:

- Начальную скорость [м/с]
- Угол вылета в [°]

- Установившуюся скорость [м/с]
- Шаг интегрирования [с]
- Ускорение свободного падения [м/с<sup>2</sup>]

Для запуска в вакууме установившаяся скорость должна быть равна начальной скорости тела.

The image shows two side-by-side screenshots of a software interface for inputting data. Each interface has five input fields with up/down arrows and a 'Start' button at the bottom.

**Left Screenshot (Default Values):**

- Скорость, м/с: 0
- Угол вылета, °: 0
- Уст. скорость, м/с: 0
- Шаг интегр, с: 0
- Ускорение g, м/с<sup>2</sup>: 0
- Button: Пуск!

**Right Screenshot (Specific Values):**

- Скорость, м/с: 1500
- Угол вылета, °: 35
- Уст. скорость, м/с: 1350
- Шаг интегр, с: 0.01
- Ускорение g, м/с<sup>2</sup>: 9.8
- Button: Пуск!

Below the right screenshot, there is a text box displaying the following information in green text:

```
>Скорость U = 1500.0 м/с
>Угол а = 35.0°
>Скорость Um = 1350.0м/с
>Шаг интегрирования = 0.01
с
>Ускорение g = 9.8 м/с²
```

Рисунок 6. Ввод данных

На холсте могут быть отображены 5 графиков:

- график зависимости  $y(x)$
- график зависимости скорости от времени
- зависимость модуля ускорения от времени
- зависимость координаты  $x$  от  $t$



- зависимость координаты  $y$  от  $t$

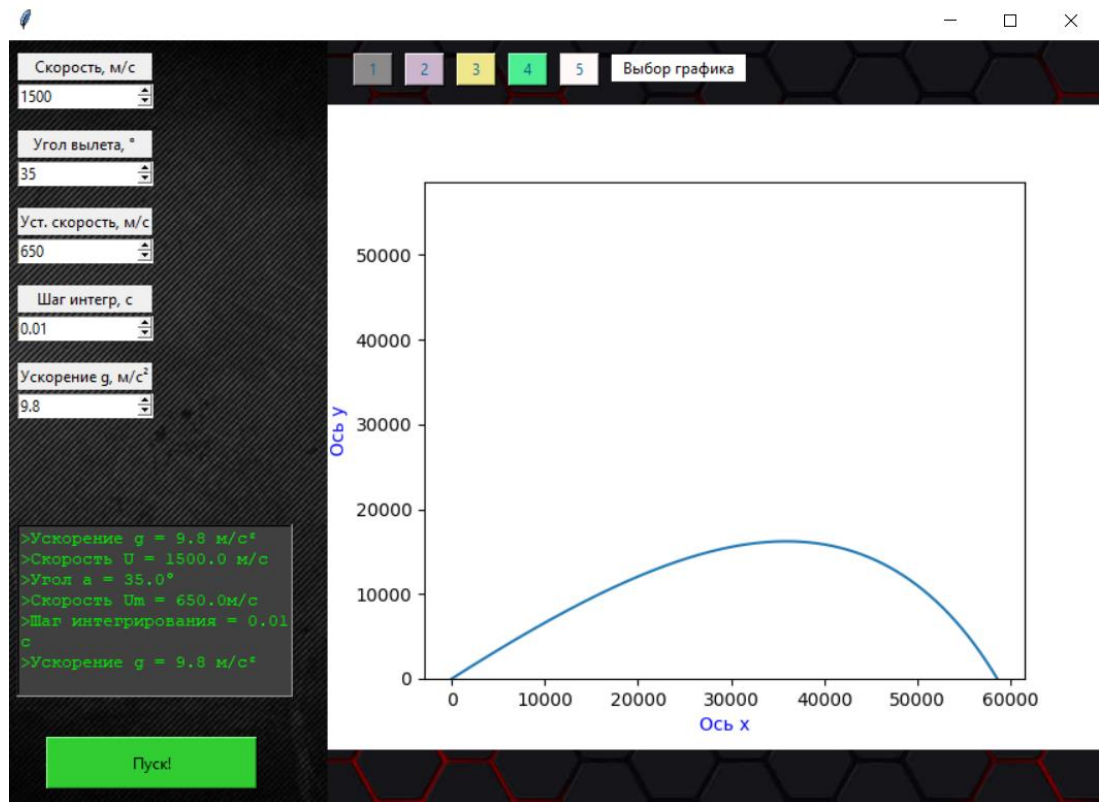


Рисунок 7. Пример графика зависимости  $y(x)$

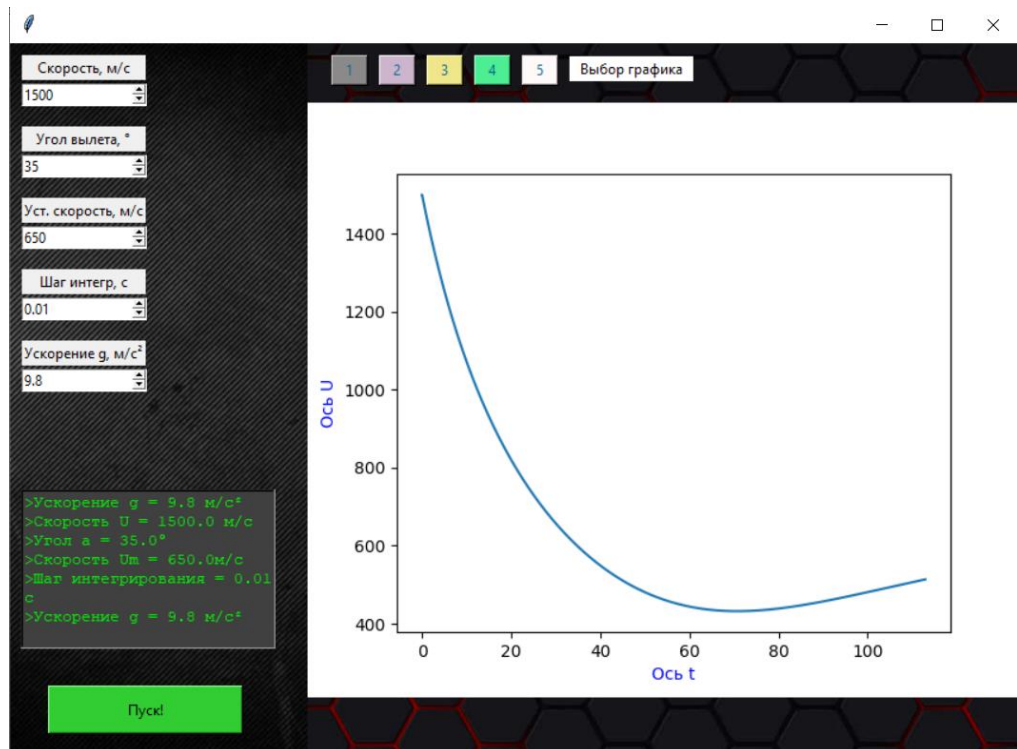


Рисунок 8. Пример графика зависимости скорости от времени

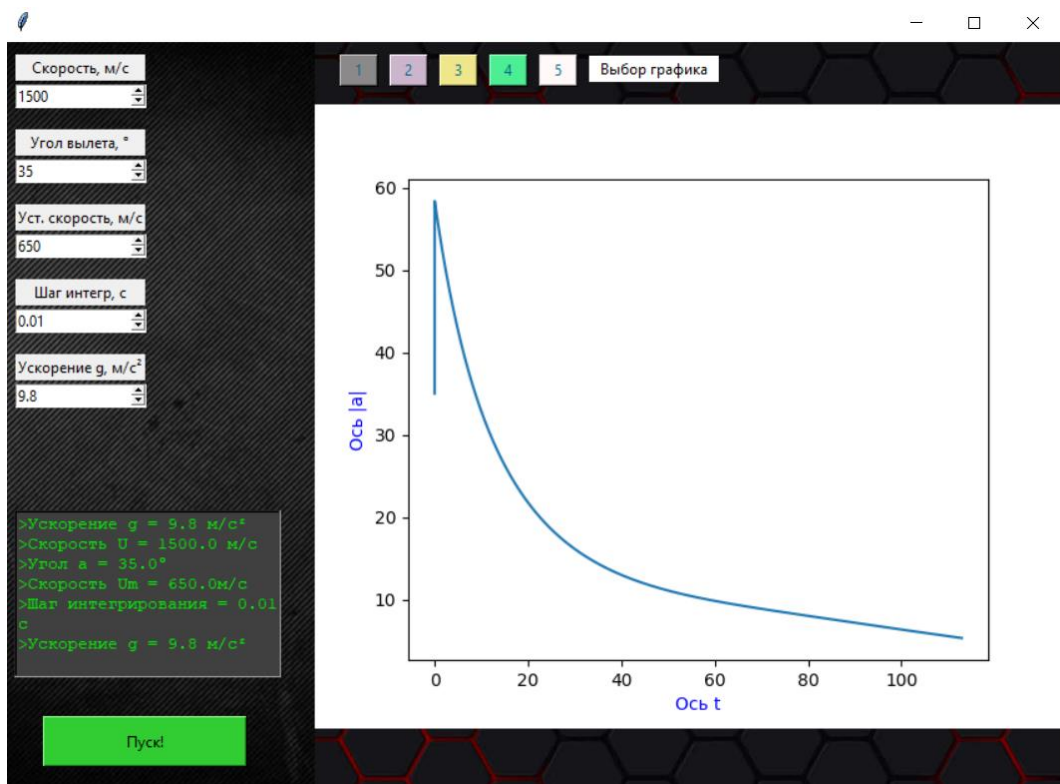


Рисунок 9. Пример графика зависимости модуля ускорения от времени

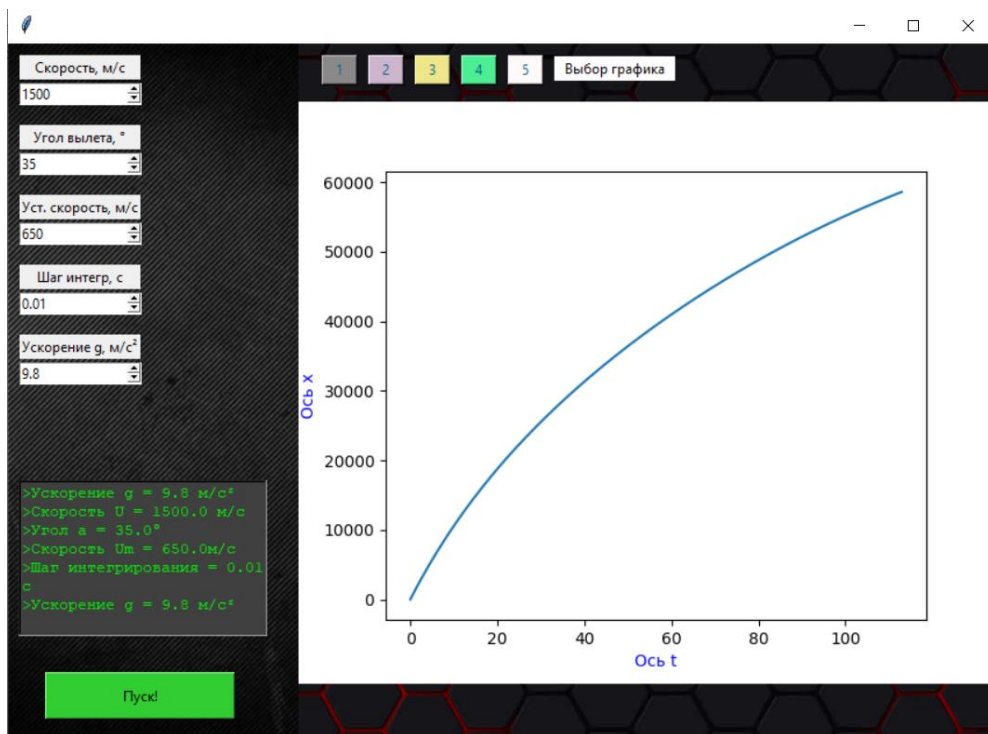


Рисунок 10. Пример графика зависимости координаты  $x$  от  $t$

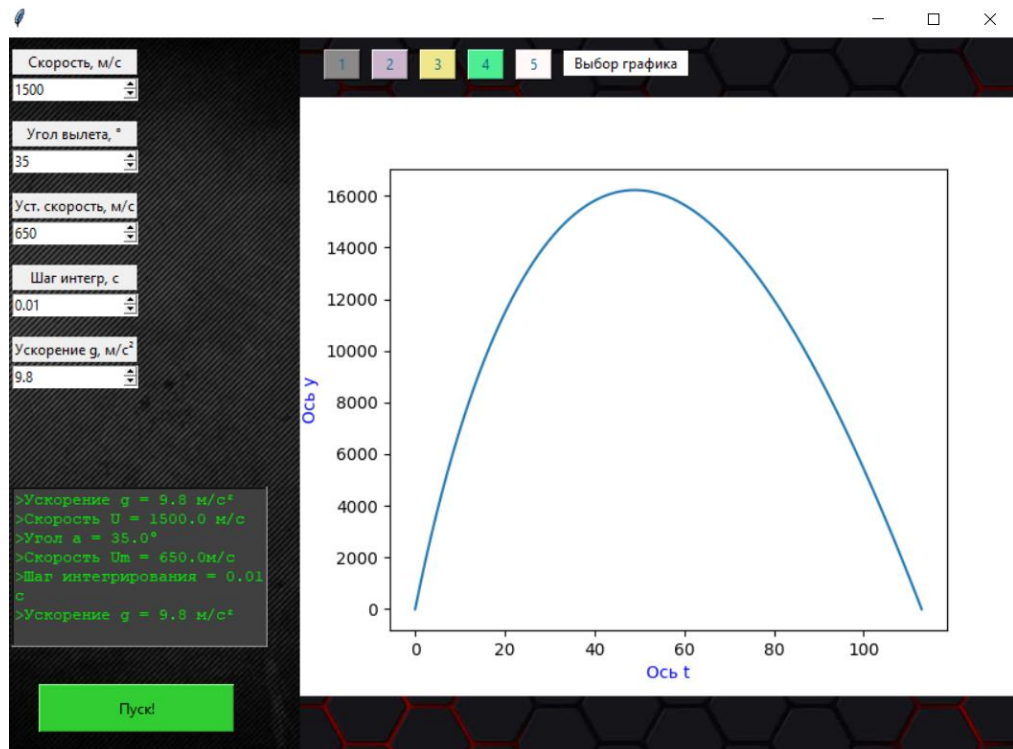


Рисунок 11. Пример графика зависимости координаты  $y$  от  $t$

#### 4.2 Обзор интерфейса программы для модели, в которой величина силы сопротивления пропорциональна скорости тела

Интерфейс можно условно разделить на два поля: поле ввода информации для запуска алгоритма моделирования, и полотно для отображения графиков. Стартовое окно имеет следующий вид:

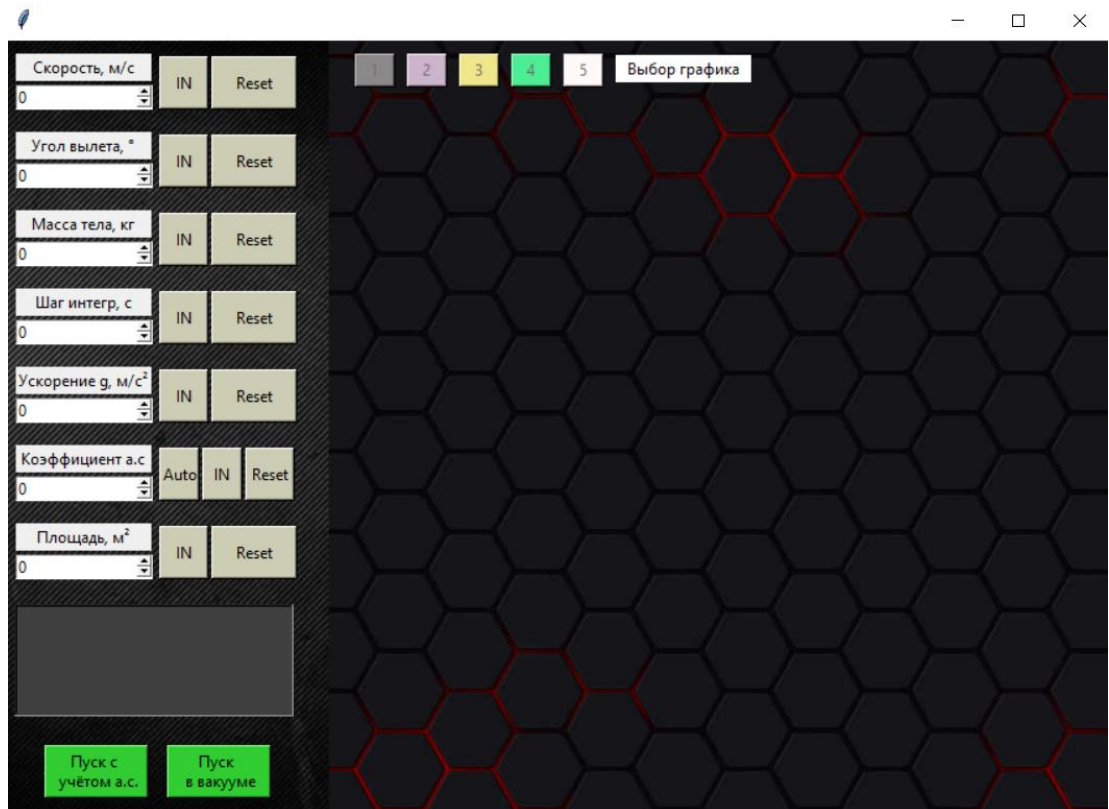


Рисунок 12. Стартовое окно программы

Программа предлагает два варианта броска тела: в вакууме и с учётом аэродинамического сопротивления. Для случая с учётом аэродинамического сопротивления необходимо указать следующие параметры:

- Начальную скорость [м/с]
- Угол вылета в [°]
- Массу тела в [кг]
- Шаг интегрирования [с]
- Ускорение свободного падения [м/с<sup>2</sup>]
- Коэффициент аэродинамического сопротивления [безразмерная величина]

Для запуска в вакууме достаточно указать начальную скорость и угол вылета – расчёт траектории движения тела будет производиться по упрощённому алгоритму с использованием математических функций библиотеки Matplotlib. (графики 2 – 5 будут недоступны). Запустить тело в вакууме также можно,



указав коэффициент а.с. равным 0 в пуске с аэродинамическим сопротивлением.

The figure shows three panels of a software interface for inputting data. Each panel has a dark background with white text and green buttons. The panels show different parameter settings for three different simulation scenarios.

**Panel 1 (Left):**

- Скорость, м/с: 0
- Угол вылета, °: 0
- Масса тела, кг: 0
- Шаг интерп, с: 0
- Ускорение g, м/с<sup>2</sup>: 0
- Коэффициент а.с.: 0 (Auto, IN, Reset buttons)
- Площадь, м<sup>2</sup>: 0
- Buttons: Пуск с учётом а.с. (green), Пуск в вакууме (green)

**Panel 2 (Middle):**

- Скорость, м/с: 200
- Угол вылета, °: 41
- Масса тела, кг: 0
- Шаг интерп, с: 0
- Ускорение g, м/с<sup>2</sup>: 0
- Коэффициент а.с.: 0 (Auto, IN, Reset buttons)
- Площадь, м<sup>2</sup>: 0
- Display: >Скорость U = 200.0 м/с, >Угол а = 41.0°
- Buttons: Пуск с учётом а.с. (green), Пуск в вакууме (yellow)

**Panel 3 (Right):**

- Скорость, м/с: 30
- Угол вылета, °: 30
- Масса тела, кг: 2
- Шаг интерп, с: 0.01
- Ускорение g, м/с<sup>2</sup>: 9.8
- Коэффициент а.с.: 0.4 (Auto, IN, Reset buttons)
- Площадь, м<sup>2</sup>: 2
- Display: >Ускорение g = 9.8 м/с<sup>2</sup>, >Коэффициент а.с. = 0.4, >Площадь = 2.0, >Пуск совершён
- Buttons: Пуск с учётом а.с. (yellow), Пуск в вакууме (green)

Рисунок 13. Ввод данных

На холсте могут быть отображены 5 графиков:

- график зависимости  $y(x)$
- график зависимости скорости от времени
- зависимость модуля ускорения от времени
- зависимость координаты  $x$  от  $t$
- зависимость координаты  $y$  от  $t$

Полученные графики будут аналогичны графикам, которые представлены на рисунках 7-11

## 5. Верификация данных

Проверить результаты расчёта скорости можно на аналитической модели движения тела в среде с сопротивлением воздуха.

Для вычисления погрешности алгоритма вычисления скорости можно использовать среднеквадратическую погрешность

$$\varepsilon = \frac{\sqrt{\sum (v_{\text{точн}} - v_{\text{числ}})^2}}{\sqrt{\sum v_{\text{точн}}^2}}.$$

Погрешность будет зависеть от выбора алгоритма численного интегрирования и величины шага по времени. При достижении требуемой погрешности можно определить минимальный шаг по времени для дальнейших вычислений.

Ведём данные для реального тела – пули для винтовки Бердана

- $m = 0,02416 \text{ кг}$
- $S = 0,0000907165625 \text{ м}^2$
- $k = 0,0165159$
- $\rho = 1,2754 \text{ кг/м}^3$
- $U_0 = 437 \text{ м/с}$
- $U_{\text{уст}} = 352 \text{ м/с}$

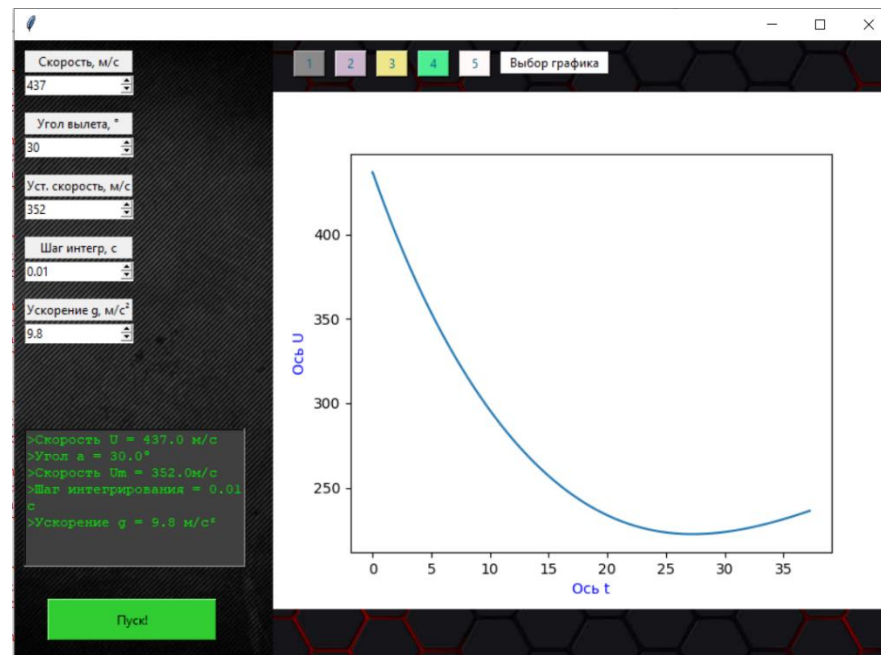


Рисунок 14. Верификация алгоритма

Графики зависимости скорости от времени для модели с коэффициентом а.с. (красная линия) и модели с установившейся скоростью (синяя линия) совпали.

### Выводы:

- 1) В ходе выполнения обязательного задания был численно промоделирован бросок тела под углом к горизонту;
- 2) Изучены способы верификации компьютерных моделей на основании физических законов и аналитических моделей;
- 3) Совершенствованы приемы объектно-ориентированного программирования в физических задачах.