

ВИЗУАЛИЗАЦИЯ ЧИСЛЕННОГО РЕШЕНИЯ ФИЗИЧЕСКОЙ ЗАДАЧИ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ SCIPY

И. А. Савеня¹⁾, Г. Ч. Шушкевич²⁾

¹⁾ Гродненский государственный университет Янки Купалы,
ул. Ожешко, 22, 208, г. Гродно, Беларусь, saveny_ia_21@student.grsu.by

²⁾ Гродненский государственный университет Янки Купалы,
ул. Ожешко, 22, 208, г. Гродно, Беларусь, gsys@grsu.by

В данной работе разработан алгоритм и программный код для моделирования полета снаряда. Траектория движения описана системой дифференциальных уравнений. Для реализации данной задачи выбран язык программирования Python с использованием научной библиотеки SciPy. С помощью разработанного программного обеспечения проведён анализ различных сценариев полёта, произведена интерпретация полученных результатов. На основе численного эксперимента построены графики траектории движения.

Ключевые слова система дифференциальных уравнений; визуализация результатов; язык программирования Python; библиотека SciPy.

Введение

В настоящее время язык программирования Python [1] предоставляет возможности для реализации численных методов, позволяющих решать прикладные задачи, которые представляют собой сложные математические модели. Одним из основных инструментов для реализации таких методов в рамках Python является научная библиотека SciPy [2]. Этот инструмент содержит встроенные функции для численного решения обыкновенных дифференциальных уравнений (ОДУ) [3], вычисления различных интегралов и специальных функций, обработки сигналов, интерполяции функций.

Постановка задачи

Для решения задачи, связанной с динамикой полёта снаряда в воздухе, была взята математическая модель, основанная на системе дифференциальных уравнений, описывающей движение объекта в пространстве [4]. Эта модель учитывает различные факторы, влияющие на составляющие полета, – аэродинамическое сопротивление, гравитационное притяжение и изменение массы объекта со временем.

Математическая модель полета представлена системой дифференциальных уравнений [4]:

$$\begin{aligned}\frac{d}{dt}x(t) &= v(t) \cdot \cos(\theta(t)) \\ \frac{d}{dt}y(t) &= v(t) \cdot \sin(\theta(t)) \\ \frac{d}{dt}v(t) &= \left(\frac{1}{1/t+1}\right) \left(3 \cdot \exp(-t) - 0.5 \cdot c \cdot p \cdot s \cdot v^2(t)\right) - g \cdot \sin(\theta(t)) + \left(\frac{1}{t+1}\right) \cdot v(t) \\ \frac{d}{dt}\theta(t) &= \frac{-g}{v(t)} \cdot \cos(\theta(t))\end{aligned}$$

где $x(t)$, $y(t)$ – координаты снаряда в момент времени t , $V(t)$ – скорость снаряда, $\theta(t)$ – угол наклона. Параметры: c – коэффициент сопротивления, p – плотность воздуха, s – поперечное сечение снаряда, g – ускорение свободного падения.

Реализация задачи на языке Python

Применение современных численных методов значительно сокращает время для решения задачи. Основное преимущество заключается в возможности использования научной библиотеки SciPy в Python, что позволяет анализировать поведение описанной динамической системы при различных значениях параметров. По результатам вычислений строятся графики зависимостей между величинами, входящими в систему дифференциальных уравнений. Это необходимо для подбора параметров системы, обеспечивающих желаемые свойства в практических приложениях.

Программа, предназначенная для решения описанной задачи, состоит из нескольких алгоритмических блоков. Ниже описаны их функции и особенности применения в том же порядке, в котором они представлены в программе:

1. Подключение модуля `integrate` из библиотеки SciPy для численного решения системы ОДУ. Подключение библиотеки NumPy для работы с массивами и модуля `matplotlib.pyplot` для визуализации данных.

2. Определение функции, описывающей систему обыкновенных дифференциальных уравнений (ОДУ).

3. Определение начальных условий: $x(0) = y(0) = 0$ (м), $v(0) = v_0$, $\theta(0) = \theta_0$ и параметров задачи: $c = 0.2$, $p = 1.29$ (кг/м), $s = 0.25$ (м²), $g = 9.81$ м/с².

4. Использование функции `solve_ivp` из модуля `integrate` библиотеки SciPy для численного решения системы ОДУ [5]:

```
scipy.integrate.solve_ivp(fun, t_span, y0, method='RK45', t_eval=None, dense_output=False,
                          events=None, vectorized=False, args=None, **options),
```

здесь `fun` – функция, описывающая систему ОДУ, принимает текущее время и текущее состояние системы, возвращает производные состояний; `t_span` – кортеж из двух элементов, определяющий начальное и конечное время интегрирования; `y0` – начальное состояние системы, массив значений, соответствующих состояниям системы в начальный момент времени; `method` – метод решения системы дифференциальных уравнений, по умолчанию используется метод RK45, но в нашем случае использован метод BDF [6]; в параметре `t_eval` можно указать точки, в которых требуется найти решение. Остальные параметры устанавливаются по умолчанию.

5. На основании численного решения задачи выполняется построение графиков.

Визуализация решения задачи

На рис. 1 представлены графики траектории полета при разных значениях угла взлета: 30, 45, 60 градусов, начальная скорость $v(0) = 50$ (м/с).

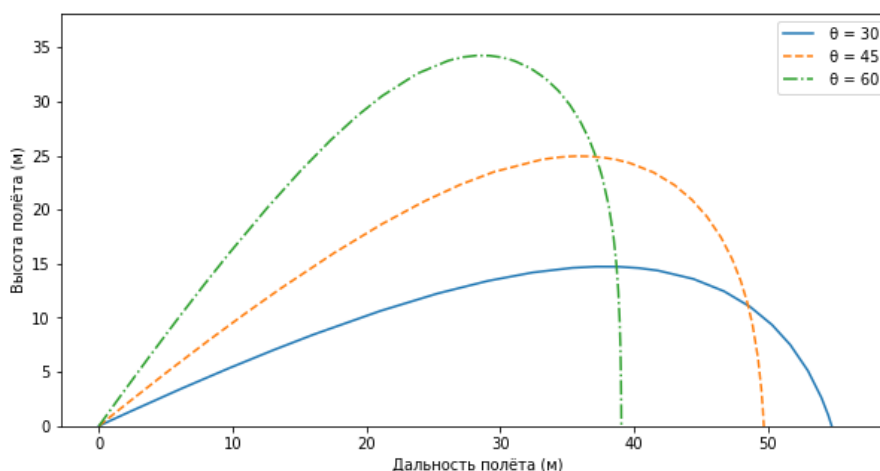


Рис. 1. Траектория снаряда для разных углов взлета.

На основании представленной информации можно сделать вывод, что чем больше угол запуска, тем выше летит снаряд, но на меньшую дальность.

На рис. 2 показана траектория движения при различных значениях начальной скорости – 30, 100, 130 м/с и при начальном угле запуска 45 градусов. На этом рисунке видно, что начальная скорость влияет на высоту и дальность полёта

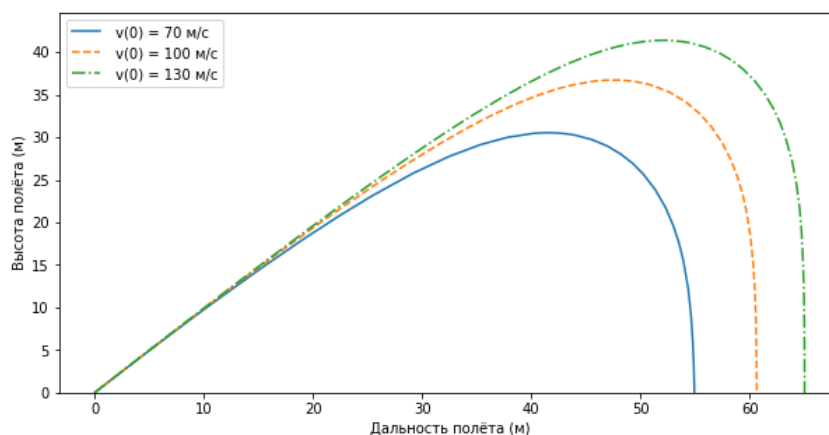


Рис. 2. Траектория снаряда с разной начальной скоростью

Вывод

В ходе нашего исследования изучена динамика полёта снаряда, используя современные численные методы и язык программирования Python. Создана компьютерная модель, которая помогает лучше понять, как различные факторы, такие как аэродинамическое сопротивление и начальные параметры полёта, влияют на траекторию полёта.

Путём анализа установлено, что выбор начального угла и скорости полёта имеет значительное значение для дальности и высоты полёта. Эти выводы имеют важное значение для проектирования, поскольку они помогают понять, какие параметры необходимо учитывать для достижения оптимальных результатов.

Библиографические ссылки

1. Severance C. Python for Everybody: Exploring Data in Python 3. Amazon Digital Services LLC, 2016. 242 p.
2. Берхман К. Р. Основы Python для Data Science. Спб.: Питер, 2023. 272 с.: ил. (Серия «Библиотека программиста»).
3. Нежелская, Л.А. Линейные дифференциальные уравнения и системы линейных уравнений : учеб. Пособие. Томск : Издательство Томского государственного университета, 2023. 142 с.
4. Golub G.H., Ortega J.M. Scientific Computing and Differential Equations: An Introduction to Numerical Methods. Academic Press, 1992. 334 p.
5. `scipy.integrate.solve_ivp` [Electronic resource]. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html (date of access: 25.03.2024).
6. Ascher U. M.; Petzold L. R. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM, Philadelphia, 1998. 331 p.