ПРИКЛАДНЫЕ АСПЕКТЫ МАТЕМАТИКИ И ИНФОРМАТИКИ

DOI: 10.24888/2500-1957-2020-4-46-51

УДК 004.423.22

СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ ПОСТРОЕНИЯ ГРАФИКА ТРАЕКТОРИИ ДВИЖЕНИЯ ГРУЗА С ПРУЖИНОЙ ПО ГОРИЗОНТАЛЬНОЙ ПОВЕРХНОСТИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Ольга Борисовна Гладких
к.ф.-м.н., доцент
og1972@rambler.ru
г. Елец
Ирина Ивановна Васильева
ст. преподаватель
irinavsl@yandex.ru
г. Елец
Максим Олегович Мельников
студент
melnikov.maxx@yandex.ru
г. Елец

Елецкий государственный университет им. И. А. Бунина

Аннотация. Рассмотрено применение метода Рунге–Кутты для решения дифференциальных уравнений. Предложена реализация метода Рунге–Кутты второго порядка на языке программирования Python. Создано приложение с графическим интерфейсом на Python для построения графика траектории движения груза с пружиной по горизонтальной поверхности с учетом трения. Начальные условия для расчетов получены от пользователя посредством диалога с GUI.

Ключевые слова: дифференциальные уравнения, метод Рунге-Кутты, Python, математическое моделирование.

Различные аспекты математического моделирования сопровождают деятельность исследователей с момента зарождения точных наук. В ходе развития современного общества интерес ученых прикован к сложным экономическим, техническим и математическим системам, уже не поддающимся изучению, основанному исключительно на теоретических методах.

Непосредственный эксперимент над подобного рода системами может быть долгим, дорогим, опасным или невозможным. В таких ситуация на помощь исследователям приходит математическое моделирование. Специализированные математические пакеты и языки программирования обеспечивают возможность и удобство проведения тех или иных сложных расчетов.

При описании моделей многих технических, физических систем и устройств используются дифференциальные уравнения. Дифференциальные уравнения также нашли применение в экономике, биологии, химии, экологии и других научных сферах.

Классическая запись однородного дифференциального уравнения имеет вид:

$$F\left(x, y, y', \ldots, y^{(n)}\right) = 0 \tag{1}$$

где x – аргумент, n – наивысший порядок производной, также является порядком дифференциального уравнения.

Решением дифференциального уравнения называется всякая функция $y = \varphi(x)$, которая при подстановке в исходное дифференциальное уравнение обращает его в верное равенство. Частное решение можно получить из общего, задав произвольным постоянным определенные начальные значения. Например, частное решение уравнения первого порядка имеет вид: $y = \varphi(x, \mathrm{C})$, где C – некоторая произвольная постоянная.

Способ задания дополнительных условий определяет два вида задач:

1. Условия заданы в единственной точке – задача Коши.
2. Условия заданы в нескольких разных точках – краевая задача.

Например, задача Коши для системы ОДУ может выглядеть следующим образом:

$$\frac{du}{dt} = F(t, u), t > 0, \qquad u(0) = u_0 \tag{2}$$

В такой задаче требуется по заранее известному решению в точке t=0 найти решения при других значениях t.

На практике исследователи часто сталкиваются с задачами, имеющими начальные данные. Для приближенного решения такого рода задач можно использовать метод Рунге–Кутта. Он является одним из самых распространенных методов, применяемых для решения обыкновенных дифференциальных уравнений.

Неявный метод Рунге–Кутты имеет вид:

$$y_{n+1} = y_n + h\sum_{i=1}^{8} b_i k_i, \qquad k_i = f(x_n + c_i h, y_n + h\sum_{i=1}^{8} a_{ij} k_i), i = 1, \ldots, s, \tag{3}$$

где b_i, a_{ij}, c_i – коэффициенты, h – величина шага сетки по x, s – количество этапов, на которых происходит вычисление нового значения.

Наиболее простым вариантом метода Рунге–Кутты является видоизмененный метод Эйлера с пересчетом:

$$y_{n+1} = y_n + h\frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}. \tag{4}$$

Схема расчета методом Рунге–Кутты второго порядка описывается следующими рекуррентными формулами:

$$\begin{gathered} \mathrm{k}_1 = \mathrm{hf}(\mathrm{x}_\mathrm{n}; \mathrm{y}_\mathrm{n}), \\ \mathrm{k}_2 = \mathrm{hf}(\mathrm{x}_\mathrm{n} + \mathrm{h}; \mathrm{y}_\mathrm{n} + \mathrm{k}_1), \\ \Delta\mathrm{y}_\mathrm{n} = \frac{1}{2}(\mathrm{k}_1 + \mathrm{k}_2), \\ \mathrm{y}_{\mathrm{n}+1} = \mathrm{y}_\mathrm{n} + \Delta\mathrm{y}_\mathrm{n}. \end{gathered} \tag{5}$$

Соответственно, при реализации алгоритма в цикле изначально вычисляются k_1, k_2 , а после значение в новом расчетном узле.

Для решения задачи расчета и последующего построения графика траектории движения груза с пружиной по горизонтальной поверхности с учетом трения было разработано GUI-приложения на языке Python3.7 в среде IDLE Python. Схематичный чертеж задачи представлен на рисунке 1. Приложение посредством диалога с пользователем получает следующие начальные значения: стартовую позицию, скорость, массу, жесткость пружины, трение и время.

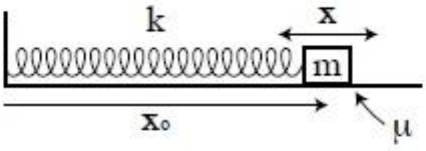

Рис. 1. Схематичный чертеж решаемой задачи

В качестве расчетного метода использовался метод Рунге–Кутты второго порядка точности, реализованный в виде отдельной пользовательской функции rk2 (рис. 2).

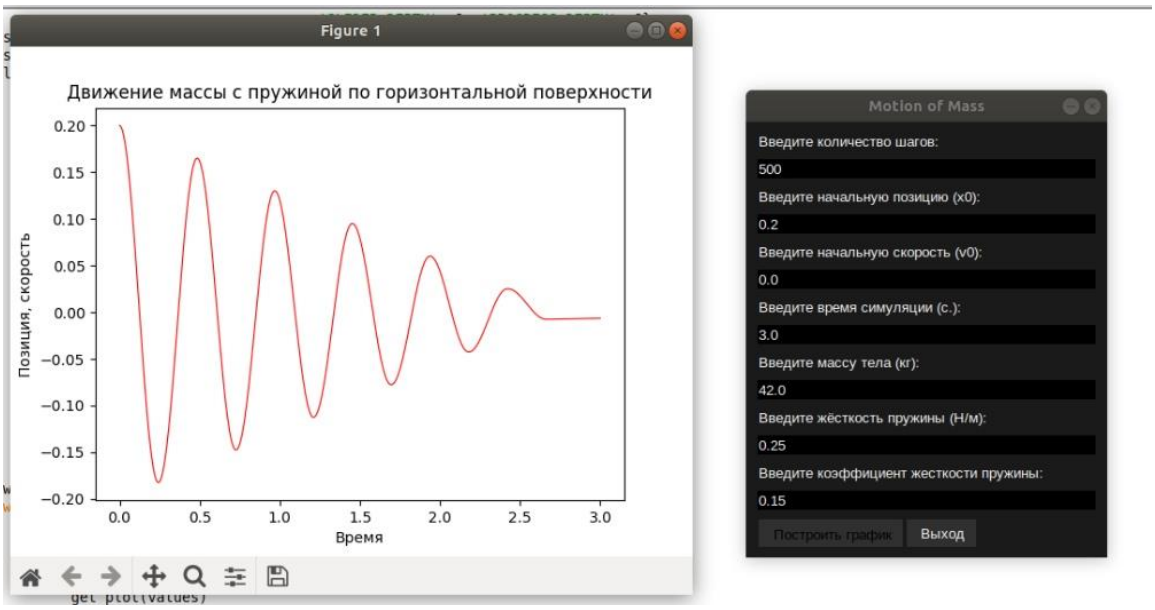

Рис. 2. Интерфейс и результат работы созданного приложения

Простая структура, лаконичность синтаксиса и большое количество дополнительных модулей делают язык Python подходящим инструментом для математического моделирования и проведения разного рода вычислений.

Разработанное приложение задействует следующие модули:

1. Numpy для работы с матрицами и массивами (объекты *np.array* и *np.zeros*);
2. для визуализации графика траектории движения груза – библиотека matplotlib. Создана пользовательская функция get_plot, реализующая настройку внешнего вида и отрисовку графика по входным данным;
3. графический интерфейс, реализующий диалог с пользователем для ввода начальных значений, использует модуль PySimpleGUI. В частности, был использован объект графического окна sg.

Первоначально создается numpy-матрица для хранения результата вычислений. Каждая строка из двух элементов хранит состояние системы в определенный момент времени и в момент времени через *dt*. Первый элемент каждой строки – позиция, второй – скорость. Затем задаются начальные значения, введенные пользователем, и совершается несколько итерация для расчета данных, в последствии используемых при построении графика траектории. Все вычисления происходят в пользовательских функций rk2 и spring_mass. Функция spring_mass разбивает решаемое дифференциальное уравнение второго порядка

$$\frac{dx^2}{dt} = -\frac{k}{m} \cdot x \mp \mu \cdot g, \quad (6)$$

на два дифференциальных уравнения первого порядка:

$$\begin{gathered} \frac{dx}{dt} = v, \\ \frac{dv}{dt} = -\frac{k}{m} \cdot x \mp \mu \cdot g. \end{gathered} \quad (7)$$

Затем в цикле производится расчет и полученные данные отправляются в функцию get_plot для дальнейшей визуализации.

Программная реализация расчетов с помощью пользовательских функций на языке Python:

```
import PySimpleGUI as sg
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
def rk2 (y, time, dt, derivs, k, m, gravity, mu) :
    k0 = dt * derivs(y, time, k, m, gravity, mu)
    k1 = dt * derivs(y + k0, time + dt, k, m, gravity, mu)
    y_next = y + 0.5 * (k0 + k1)
    return y_next
def spring_mass(state, time, k, m, gravity, mu):
    g0 = state[1]
    g1 = (-k / m * state[0] - gravity * mu) if g0 > 0 else (-k / m *
state[0] + gravity * mu)
    return np.array([g0, g1])
```

Листинг 1. Программа для построения графика движения массы и пружины по горизонтальной поверхности с трением

Разработанное приложение предоставляет пользователю простой и интуитивно понятный графический интерфейс. Программа помогает абстрагироваться от технической реализации расчетов и сосредоточиться на быстрой визуализации результатов. Это позволяет использовать данный продукт в образовательных целях, в качестве демонстрационного средства, помогающего закрепить полученные теоритические знания.

Таким образом, языки программирования, в частности язык Python, являются удобным инструментом для моделирования и проведения разного рода математических расчетов. Python предоставляет множество модулей и библиотек, ускоряющих процесс проведения исследований.

В результате разработано приложение для построения графика траектории движения груза на пружине по горизонтальной поверхности с учетом трения. Программа имеет простой и понятный интерфейс. Кроме того, приложение легко модифицировать и расширять. Например, посредством добавления новых пользовательских функций, реализующих другие численные методы, интеграции сторонних библиотек, ускоряющих процесс проведения вычисление и прочее.

Список литературы

1. Алексеев Е. Р., Чеснокова О. В. Решение задач вычислительной математики в пакетах Matchad 12, Matlab 7, Maple 9. М.: НТ Пресс, 2006.
2. Бахвалов Н. С., Жидков Н. П., Кобельников Г. М. Численные методы. М.: Лаборатория базовых знаний, 2001.

3. Васильева И. И. Моделирование физического процесса в висмуте с помощью языка программирования PYTHON // Материалы международной научно-практической конференции «Системы управления, технические системы: устойчивость, стабилизация, пути и методы исследования». Елец: ЕГУ им. И. А. Бунина. 2019. С. 134-138.
4. Звонарев С. В. Основы математического моделирования: учебное пособие. ЕКБ.: Урал, 2019.
5. Маккини У. Python и анализ данных. М.: ДМК Пресс, 2015.

CREATION OF AN APPLICATION FOR PLOTTING THE TRAJECTORY OF A LOAD WITH A SPRING ON A HORIZONTAL SURFACE IN THE PYTHON PROGRAMMING LANGUAGE

O. B. Gladkikh
Ph.D., associate professor
og1972@rambler.ru
Yelets
I. I. Vasilyeva
Senior Lecturer
irinavsl@yandex.ru
Yelets
M. O. Melnikov
student
melnikov.maxx@yandex.ru
Yelets

Bunin Yelets State University

Abstract. The application of the Runge – Kutta method for solving differential equations is considered. An implementation of the second order Runge – Kutta method in the Python programming language is proposed. An application with a graphical interface in Python was created to plot the trajectory of a load with a spring on a horizontal surface, taking into account friction. The initial conditions for calculations are obtained from the user through a dialogue with the GUI.

Keywords: differential equations, Runge-Kutta method, Python, mathematical modeling.

References

1. Alekseev, E. R., Chesnokova, O. V. (2006). Solution of problems of computational mathematics in the packages Matchad 12, Matlab 7, Maple 9 [*Resheniye zadach vychislitel'noy matematiki v paketakh Matchad 12, Matlab 7, Maple 9*]. Moscow.: NT Press.
2. Bakhvalov, N. S., Zhidkov, N. P., Kobelnikov, G. M. (2001). Numerical methods [*Chislennye metody*]. Moscow: Laboratory of Basic Knowledge.
3. McKinney, W. (2015). Python and Data Analysis [*Python i analiz dannyh*]. Moscow: DMK Press.
4. Vasilyeva, I. I. (2019) Modeling the physical process in bismuth using the PYTHON programming language [*Modelirovanie fizicheskogo processa v vismute s pomoshch'yu yazyka programmirovaniya PYTHON*]. *Proceedings of international scientific-practical*

conference Control systems, technical systems: stability, stabilization, ways and methods of research. 134-138. Yelets.

5. Zvonarev, S. V. (2019). Fundamentals of mathematical modeling: a tutorial [*Osnovy matematicheskogo modelirovaniya: uchebnoe posobie*]. Ekaterinburg: Ural.

DOI: 10.24888/2500-1957-2020-4-51-56

УДК 517.983.23

ПРИЛОЖЕНИЕ ТЕОРИИ РИССА-ШАУДЕРА К ИССЛЕДОВАНИЮ РАЗРЕШИМОСТИ УРАВНЕНИЙ ТИПА РОМАНОВСКОГО С ЧАСТНЫМИ ИНТЕГРАЛАМИ

Ирина Адольфовна Елецких
к.ф.-м.н., доцент
yelttskikh.irina@yandex.ru
г. Елец

Елецкий государственный университет им. И. А. Бунина

Аннотация. До середины сороковых годов интересы специалистов по функциональному анализу были сфокусированы почти исключительно на изучение нормированных пространств. Впервые аксиомы таких пространств появились в работе Ф. Рисса о компактных операторах в пространстве $C[a;b]$ в 1918 году, но первое абстрактное изложение теории компактных операторов содержится в диссертации Стефана Банаха, опубликованной в 1932 году в Варшаве. С 50-х годов XX века возрастает прикладная направленность функционального анализа для решения различных задач математики, физики и других прикладных наук. В статье приведены понятия компактного множества и вполне непрерывного оператора, рассмотрена теория Рисса-Шаудера линейных уравнений второго порядка, приводятся теоремы, составляющие содержание этой теории. В качестве приложения теории Рисса-Шаудера приводятся условия существования решений уравнений типа Романовского с частными интегралами в пространстве непрерывных функций. Так как разрешимость операторных уравнений зависит от условий компактности, то в статье приводится пример на установление компактности интегрального оператора.

Ключевые слова: компактное множество, вполне непрерывный оператор, линейные уравнения второго порядка, теория Рисса-Шаудера.

Компактные операторы играют важную роль в различных приложениях. Они допускают достаточно детальное описание, поэтому построенная теория таких операторов позволяет значительно облегчить процесс решения интегральных и дифференциальных уравнений. В качестве определения компактного оператора возьмем сформулированное в учебнике В. С. Владимирова «Обобщенные функции в математической физике».

Пусть $\mathbb{E}$ и $\mathbb{F}$ – банаховы пространства на поле $\mathbb{F}$ действительных или комплексных чисел. Будем обозначать через S_r=$\mathrm{S}_r(0)$ шар радиуса r>0, а через S=$S_1(0)$ единичный шар с центром в нуле [1].

Определение 1. *Линейный оператор* A:$\mathbb{E}\to\mathbb{F}$ *в банаховых пространствах* $\mathbb{E}$ и $\mathbb{F}$ *называется компактным, если образ* $A(S)\subset\mathbb{F}$ *единичного шара* $S\subset\mathbb{E}$ *является предкомпактным множеством в* $\mathbb{F}$.