МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Нижегородский государственный университет
им. Н.И. Лобачевского»
Национальный исследовательский университет**

**Институт информационных технологий, математики и механики
Кафедра алгебры, геометрии и дискретной математики**

ЛАБОРАТОРНАЯ РАБОТА
«Численное решение задачи Коши для ОДУ 2 порядка»

Выполнил: студент группы 381706-02
Окмянский Андрей Владимирович

____________ Подпись

Руководитель:
Эгамов Альберт Исмаилович
____________ Подпись

Нижний Новгород

2020

Оглавление

Введение

Обыкновенные дифференциальные уравнения (ОДУ) широко используются для математического моделирования процессов и явлений в различных областях науки и техники. Переходные процессы в радиотехнике, кинетика химических реакций, динамика биологических популяций, движение космических объектов, модели экономического развития исследуются с помощью ОДУ.

Дифференциальное уравнение — уравнение, в которое входят производные функции, и может входить сама функция, независимая переменная и параметры. Порядок входящих в уравнение производных может быть различен (формально он ничем не ограничен). Производные, функции, независимые переменные и параметры могут входить в уравнение в различных комбинациях или могут отсутствовать вовсе, кроме хотя бы одной производной. Не любое уравнение, содержащее производные неизвестной функции, является дифференциальным уравнением. Например, $\boldsymbol{f'(x)} = \boldsymbol{f(f(x))}$ не является дифференциальным уравнением.

Дифференциальное уравнение порядка выше первого можно преобразовать в систему уравнений первого порядка, в которой число уравнений равно порядку исходного дифференциального уравнения.

Наиболее часто встречаются дифференциальные уравнения вида:

$$\boldsymbol{y^{(n)}=f(x,y,y',y'',\ldots,y^{(n-1)})},$$

в которых старшая производная $\boldsymbol{y^{(n)}}$ выражается в виде функции от переменных x,y и производных $\boldsymbol{y^{(i)}}$ порядков меньше $\boldsymbol{n}$. Такие дифференциальные уравнения называются нормальными или разрешенными относительно производной.

Решением (интегралом) дифференциального уравнения порядка n называется функция ***y(x)***, имеющая на некотором интервале (a, b) производные ***y'(x), y''(x), …, $y^{(n)}$(x)*** до порядка n включительно и удовлетворяющая этому уравнению.

Начальным условием для написанного выше уравнения называется условие

$$\boldsymbol{y(x_0)=y_0,\ y'(x_0)=y_0^{(1)},\ y''(x_0)=y_0^{(2)},\ \ldots,\ y^{(n-1)}(x_0)=y_0^{(n-1)}},$$

где $\boldsymbol{x_0}$ – некоторое фиксированное значение независимой переменной, а $\boldsymbol{y}_0$ и $\boldsymbol{y_0^{(i)}}$ – соответственно, фиксированные значения функции y и всех её производных до порядка $\boldsymbol{n-1}$ включительно.

Дифференциальное уравнение вместе с начальным условием называется начальной задачей или задачей Коши:

$$\begin{cases} \boldsymbol{y^{(n)}=f(x,y,y',y'',\ldots,y^{(n-1)})} \\ \boldsymbol{y(x_0)=y_0,\ y'(x_0)=y_0^{(1)},\ y''(x_0)=y_0^{(2)},\ldots,y^{(n-1)}(x_0)=y_0^{(n-1)}}. \end{cases}$$

В работе будет рассмотрен наиболее распространённый метод Рунге-Кутты 4-го порядка при вычислениях с постоянным шагом интегрирования.

Метод Рунге-Кутты 4-го порядка

Методы Рунге-Кутты — большой класс численных методов решения задачи Коши для обыкновенных дифференциальных уравнений и их систем.

Желание повысить вычислительную эффективность привело к появлению различных вычислительных версий методов Рунге-Кутты. В таких версиях стремились получить формулы из семейства методов Рунге-Кутты, которые бы использовали одни и те же значения функции – правой части уравнения – и определяли бы разные конкретные методы одного порядка (или смежных порядков, например, четвертого и пятого); при этом, чтобы по разности результатов подсчета приближенных значений решения по выведенным близким формулам (с одним и тем же шагом $\boldsymbol{h}$) можно было судить о точности одного из них.

Итак, в работе был рассмотрен метод Рунге-Кутты 4-го порядка, который является самым распространенным среди семейства методов.

Методы Рунге-Кутта обладают следующими свойствами:

1. Эти методы являются одноступенчатыми: чтобы найти $\boldsymbol{y_{m+1}}$, нужна информация о предыдущей точке $\boldsymbol{x_m}, \boldsymbol{y_m}$.

2. Они согласуются с рядом Тейлора вплоть до членов порядка $\boldsymbol{h_p}$, где степень $\boldsymbol{p}$ различна для различных методов и называется порядковым номером или порядком метода.

3. Они не требуют вычисления производных от f (x,y), а требуют вычисления самой функции. Метод Рунге-Кутты используют для расчета стандартных моделей достаточно часто, так как при небольшом объеме вычислений он обладает точностью метода $O^4(h)$.

Рассмотрим Задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка. $\boldsymbol{y'}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}(\boldsymbol{x_0})=\boldsymbol{y_0}$.

Тогда приближенное значение в последующих точках вычисляется по формуле:

$$\boldsymbol{y_{i+1}}=\boldsymbol{y_i}+\mathbf{16}(\boldsymbol{k_1}+2\boldsymbol{k_2}+2\boldsymbol{k_3}+\boldsymbol{k_4}),$$

$$\boldsymbol{k_1}=\boldsymbol{hf}(\boldsymbol{x_i},\boldsymbol{y_i}),$$

$$\boldsymbol{k_2}=\boldsymbol{hf}(\boldsymbol{x_i}+\boldsymbol{h}/2,\boldsymbol{yi}+\boldsymbol{k_1}/2),$$

$$\boldsymbol{k_3}=\boldsymbol{hf}(\boldsymbol{x_i}+\boldsymbol{h}/2,\boldsymbol{yi}+\boldsymbol{k_2}/2),$$

$$\boldsymbol{k_4}=\boldsymbol{hf}(\boldsymbol{x_i}+\boldsymbol{h},\boldsymbol{y_i}+\boldsymbol{k_3}),$$

$$\boldsymbol{h}=(\boldsymbol{x_k}-\boldsymbol{x_1})/\boldsymbol{n}.$$

Где ***i={1, 2, …, k}***, а $\boldsymbol{n}$ есть количество участков, на которые будет разбит фазовый портрет для подсчета коэффициентов.

Расчет на ***i***-ом шаге методом Рунге-Кутты (рис.1).

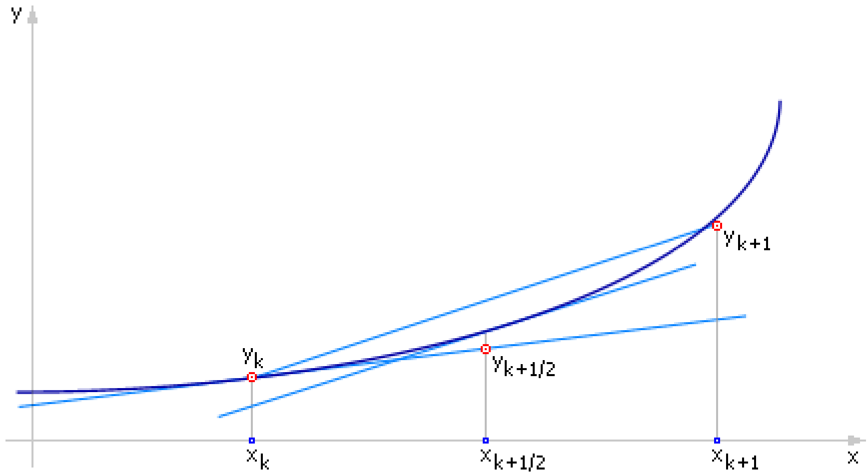

рис. 1

Описание математической модели

Уравнение малых колебаний маятника с трением (коэффициент трения k, $k > 0$) можно записать в виде (рис. 2):

$$\ddot{x} = -x - k\dot{x}$$

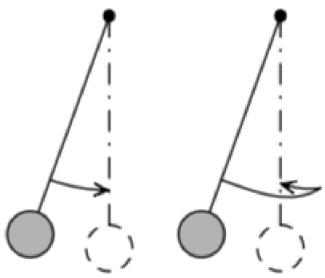

Рис. 2

Так же это уравнение можно привести к системе двух автономных дифференциальных уравнений с помощью замены:

$$x_1 = x, \quad x_2 = \dot{x}$$

Таким образом, сделанная замена приводит начальное уравнение к автономной системе:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = -x_1 - kx_2 \end{cases}$$

с матрицей:

$$\begin{Vmatrix} 0 & 1 \\ -1 & -k \end{Vmatrix}$$

Характеристическое уравнение

$$\begin{vmatrix} -\lambda & 1 \\ -1 & -k-\lambda \end{vmatrix} = 0,$$

$\lambda^2 + k\lambda + 1 = 0$, его корни $\lambda_{1,2} = \frac{-k \pm \sqrt{k^2-4}}{2}$ различные и вещественные, если $k > 2$ (случай сильного трения).

Структура программы

Программа принимает на вход параметр уравнения $\ddot{x} = -x - k\dot{x}$ коэффициен k, $\dot{x}(0)$, $x(0)$, шаг h.

Программа состоит из файла pendulum.py, в нем реализован главный метод RungeKutt(dx,dVx,h) и отрисовка фазового портрета дифференциального уравнения.

Руководство пользователя

Работа с программой осуществляется следующим образом:

1. При запуске пользователю будет предложено ввести параметры дифференциального уравнения $\ddot{x} = -x - k\dot{x}$, задать шаг. Изначально, при открытии программы, начальные параметры уравнения уже присутствуют на своих местах (рис. 3).

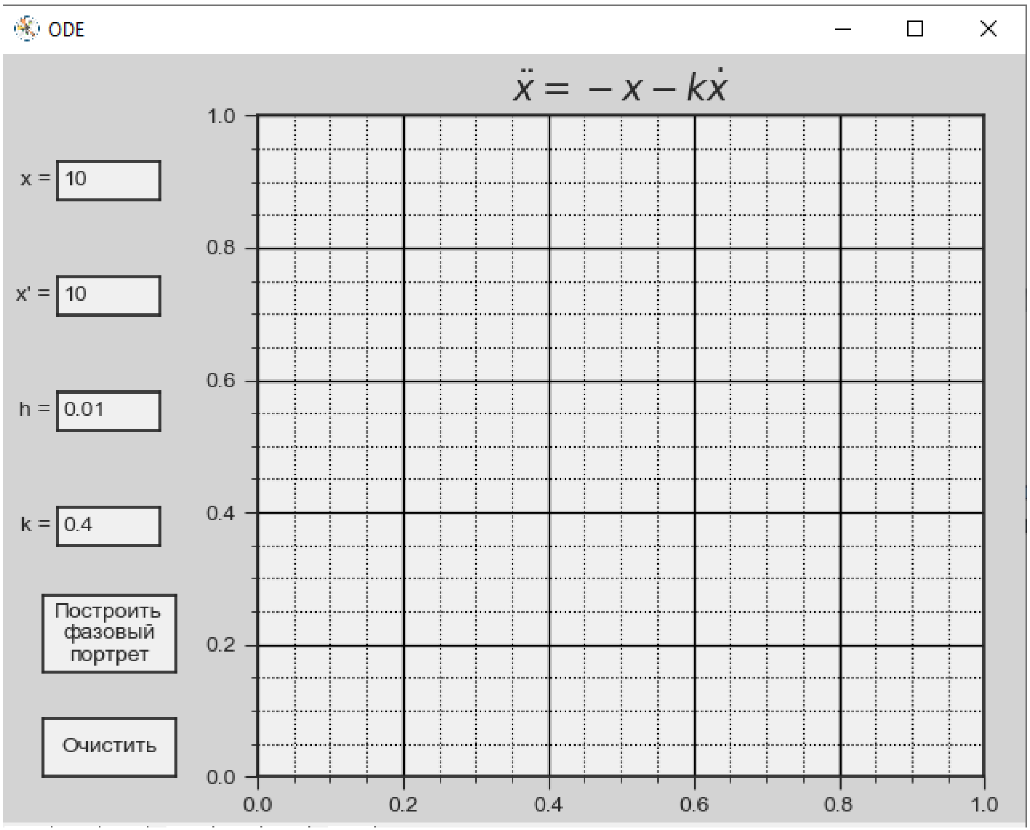

рис. 3

2. При нажатии на кнопку “Построить фазовый портрет”, на появившейся координатной плоскости будет изображен получившийся фазовый портрет (рис. 4).

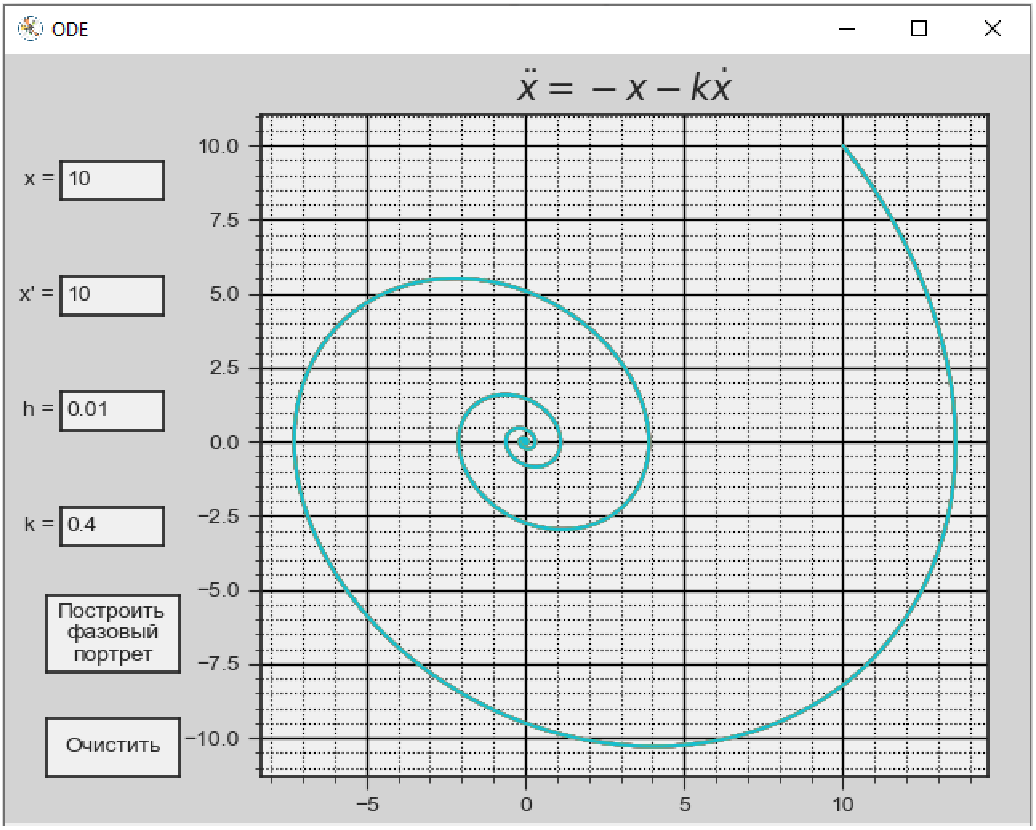

рис. 4

3. Если вводить разные параметры, при нажатии на кнопку “Построить фазовый портрет”, первое и последующие уравнения не исчезнут, а новые появятся на графике (рис. 5).

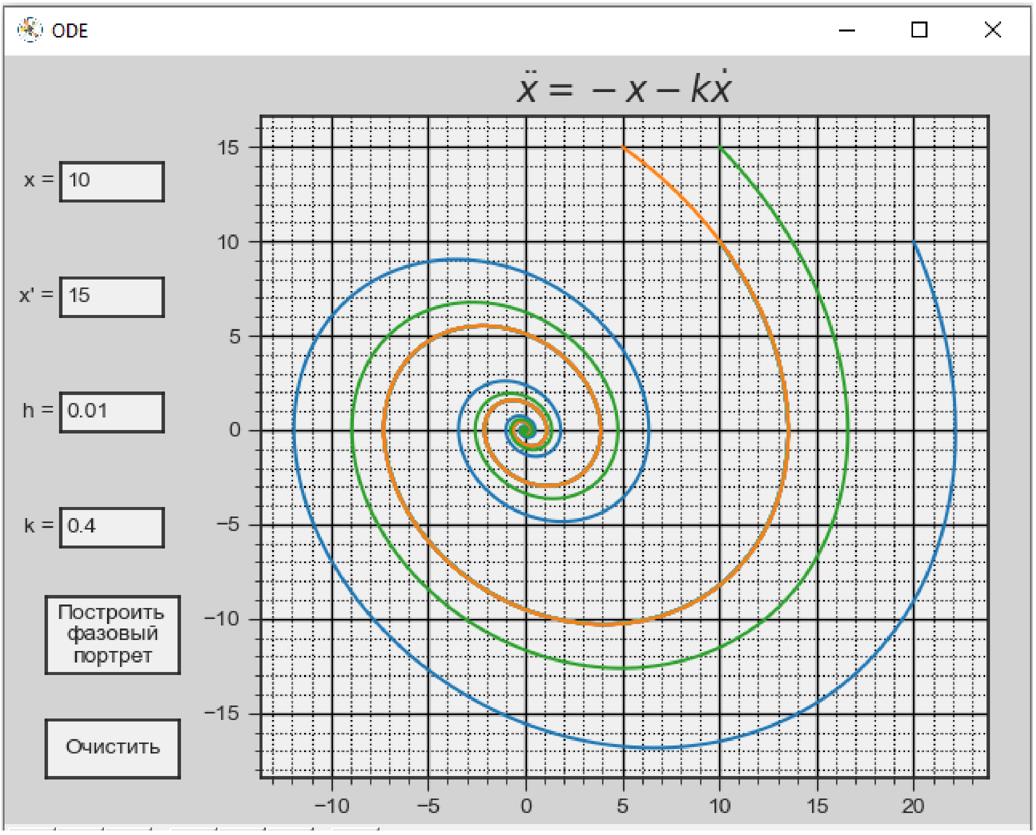

рис. 5

4. Чтобы очистить координатную плоскость, необходимо нажать на кнопку “Очистить”. После этого можно заново строить фазовый портрет уравнения.
5. Если пользователь ввел неверные данные в доступные поля, откроется окно с ошибкой. Пользователь должен ввести корректные данные (рис. 6)

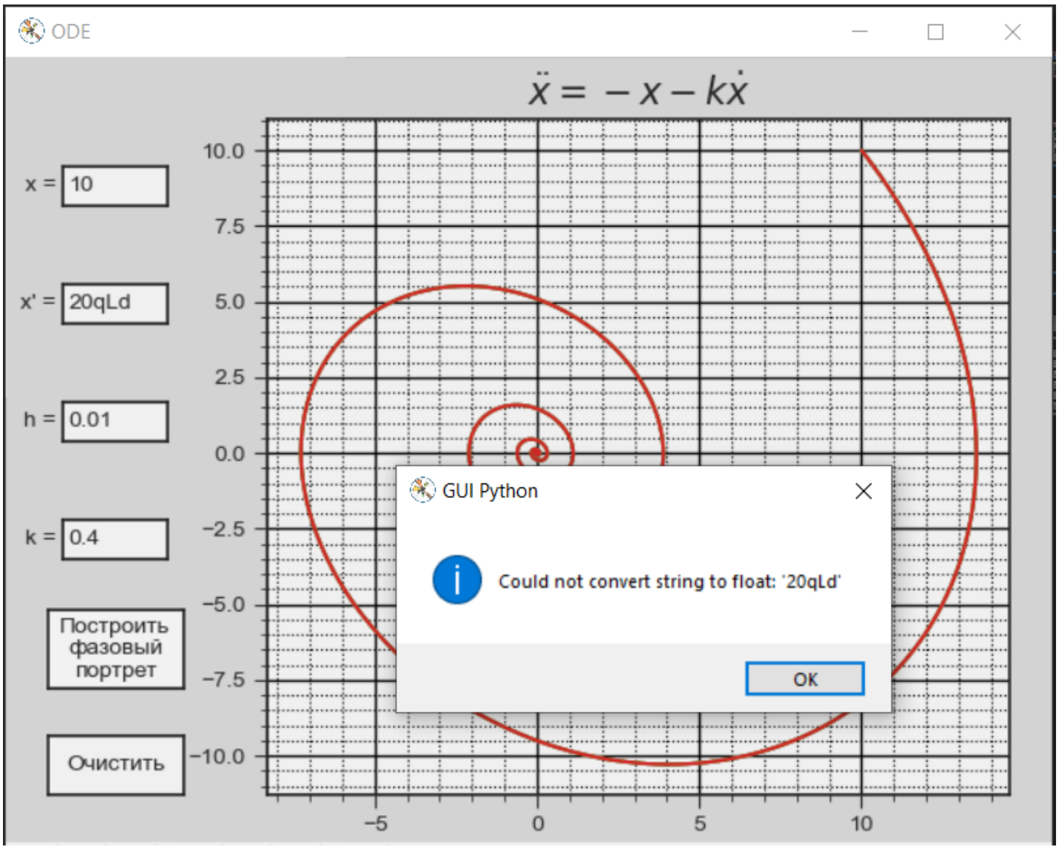

Рис. 6

Руководство программиста

Программа написана на языке Python, версия 3.7. Для вывода графика используется библиотека matplotlib, для вывода сообщений библиотека tkinter.

Ниже представлен листинг основных функций программы:

```python
def RungeKutt(dx,dVx,h):
    x.append(dx)
    y.append(dVx)
    L = 0   # вспомогательная переменная
    i = 0
    while(L < n):   # n = const = 300
        k1X2 = h * Function(x[i], y[i])
        k2X2 = h * Function(x[i] + h / 2, y[i] + k1X2 / 2)
        k3X2 = h * Function(x[i] + h / 2, y[i] + k2X2 / 2)
        k4X2 = h * Function(x[i] + h, y[i] + k3X2)

        k1X1 = h * y[i]
        k2X1 = h * (y[i] + k1X1 / 2)
        k3X1 = h * (y[i] + k2X1 / 2)
        k4X1 = h * (y[i] + k3X1)

        xt = x[i] + (k1X1 + 2 * k2X1 + 2 * k3X1 + k4X1) / 6
        yt = y[i] + (k1X2 + 2 * k2X2 + 2 * k3X2 + k4X2) / 6
        x.append(xt)
        y.append(yt)
        L += h
        i += 1

def Function(dx, dVx):
    return float(- dx - k*dVx)
```

Заключение

В ходе выполнения лабораторной работы был изучен и реализован метод Рунге-Кутта, построение фазовых траекторий. Были решены и реализованы все поставленные задачи и цели, а именно:

Были созданы программные средства, поддерживающие ввод параметров заданного дифференциального уравнения, а так же отрисовка фазового портрета. Была изучена модель колебаний маятника с трением.

Список используемой литературы

1. Метод Рунге-Кутта 4-го порядка: http://stratum.ac.ru/education/textbooks/modelir/lection15.html
2. Самарский А.А. “Введение в численный методы”, Издательство «Лань», 2005. – 228с.
3. Пиголкина Т.С. “Автономные системы. Фазовые траектории. Элементы теории устойчивости”, МФТИ, 2013. – 40с.