Межгосударственное образовательное учреждение

высшего образования

«Белорусско-Российский университет»

Кафедра «Высшая математика»

Курсовой проект

по дисциплине «Численные методы математической физики»

на тему «Решение задачи Коши для обыкновенных дифференциальных уравнений методом Рунге-Кутта 4-го порядка точности»

Выполнил: студент гр. ПМР-211

Азарёнок Афанасий Романович

Руководитель: канд. физ.-мат. наук,

доцент

Маковецкий Илья Иванович

Могилев 2023г.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ…...………………………………………………………………...…3

Основная часть…………………………………………………………………….4

1.Метод Рунге-Кутта 4-го порядка………………………………………..5

2.Реализация метода Рунге-Кутта 4-го порядка ………………………….7

3. Примеры использования метода Рунге-Кутта 4-го порядка ………....10

ЗАКЛЮЧЕНИЕ………………………………………………….……………………….14

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ…………………………….15

**ВВЕДЕНИЕ**

В вычислительной практике часто приходится иметь дело с задачами с начальными данными для системы дифференциальных уравнений. Для приближенного решения таких задач традиционно широко используются методы Рунге-Кутта, связанные с вычислением правой части системы уравнений в некоторых промежуточных точках.

*Цель курсового проекта:*

Изучение теорических основ семейства численных методов, а также моделей техники и физики, порождающих данные задачи, разработка программного продукта, демонстрирующего применение метода к модельной задаче.

*Задачи курсовой работы:*

1. Ознакомиться с методом Рунге-Кутта 4-го порядка;
2. Разработать программный продукт, демонстрирующий применение метода к модельной задаче;
3. Проверить работу метода и убедиться в его работоспособности на различных примерах.

*Объектом* исследования являются обыкновенные дифференциальные уравнения.

*Предметом* исследования является метод решения обыкновенных дифференциальных уравнений методом Рунге-Кутта 4-го порядка.

Исследование проводиться *методом* *анализа* и исследованием первоисточников.

Курсовая работа представлена введением, основной частью, включающую в себя три раздела, заключением и списком использованной литературы.

Во введении сформулированы актуальность, объект, предмет, цель и задачи курсовой работы.

В основной части раскрывается:

1. Ознакомление с методом решения обыкновенных дифференциальных уравнений методом Рунге-Кутта 4-го порядка;
2. Разработка метода Рунге-Кутта 4-го порядка;
3. Проверка работа метода с использованием дифференциального уравнения свободных затухающих колебаний в качестве модели, а также остальные примеры.

**Основная часть**

**1. Метод Рунге-Кутта 4-го порядка**

Рассматривается задача Коши для системы обыкновенных дифференциальных уравнений:



 (1.1)

С использованием векторных обозначений задачу (1.1) можем переписать как задачу Коши для одного уравнения:

 (1.2)

В задаче Коши по известному решению в точке  необходимо найти из системы уравнений (1.2) решение при других .

Таким образом полученную систему можно решать с помощью различных методов, но мы рассмотрим методику решения Рунге-Кутта 4-го порядка. При построении численных алгоритмов будем считать, что решение этой дифференциальной задачи существует, оно единственно и обладает необходимыми свойствами гладкости.

При численном решении задачи (1.2) будем использовать равномерную, для простоты, сетку по переменной  с шагом :



Приближенное решение задачи (1.2) в точке  обозначим . Метод сходится в точке , если  при . Метод имеет p-й порядок точности, если  при .

Простейшая разностная схема для приближенного решения задачи (1.2) имеет вид:

 (1.3)

При  имеем явный метод Эйлера и в этом случае разностная схема аппроксимирует уравнение из задачи (1.2) с первым порядком. Симметричная схема () имеет второй порядок аппроксимации. Подобная схема относится к классу неявных – для определения приближенного решения на новом слое нужно решать нелинейную задачу. Явные схемы второго и более порядков аппроксимации удобно строить, ориентируясь на метод предиктор-корректор. На этапе предиктора используется явная схема:



А на этапе корректора (уточнения) – схема:



В одношаговых методах Рунге-Кутта идеи предиктора-корректора реализуются наиболее полно. Этот метод записывается в общем виде:

 (1.3)

Формула (1.3) основана на  вычислениях функции  и называется -стадийной. Если  при  имеем явный метод Рунге-Кутта. Если  при  и , то  определяется неявно из уравнения:



О таком методе Рунге-Кутта говорят, как о диагонально-неявном. Параметры , ,  определяют вариант метода Рунге-Кутта. Используется следующее представление метода (таблица Бутчера):

 (1.4)

Одним из наиболее распространенных является явный метод Рунге-Кутта четвертого порядка:

 (1.5)

В компактном представлении (1.4) этого метода имеем:



**2. Реализация метода Рунге-Кутта 4-го порядка**

Напишем программу для численного решения задачи Коши для системы обыкновенных дифференциальных уравнений явным методом Рунге-Кутта четвёртого порядка.

Поставим задачу для решения нашей задачи с готовым ответом для сравнения:

 (2.1)

Для начала определим функцию increment которая будет находить величину, на которую будет изменяться функция на каждом шаге.

def increment(f, x, y, tau):

    k0 = tau \* f(x,y)

    k1 = tau \* f(x + tau/2., y + k0/2.)

    k2 = tau \* f(x + tau/2., y + k1/2.)

    k3 = tau \* f(x + tau, y + k2)

    return (k0 + k1\*2. + k2\*2. + k3) / 6

Далее реализуем метод Рунге-Кутта 4-го порядка (rungeKutta) который будет возвращать два массива: один является шагом функции, а другой значение функции на этом шаге которое мы находим с помощью инкрементации которую мы реализовали прежде.

import numpy as np

def rungeKutta(f, x0, y0, tEnd, tau):

    x = []

    y = []

    x.append(x0)

    y.append(y0)

    while x0 < tEnd:

        tau = min(tau, tEnd - x0)

        y0 = y0 + increment(f, x0, y0, tau)

        x0 = x0 + tau

        x.append(x0)

        y.append(y0)

    return np.array(x), np.array(y)

Далее нужно представить нашу функцию в надлежащем виде. При сближенном решении модельной задачи Коши для уравнения второго порядка (2.1) сначала переходим от одного уравнения второго порядка к системе из двух уравнений:



В данном виде несложно понять, как воспользоваться краевыми условиями и отсюда мы реализуем нашу функцию следующим образом:

def f(x, y):

    f = np.zeros((2), "float")

    f[0] = y[1]

    f[1] = -math.sin(y[0])

    return f

y0 = [1., 0.,]

Далее приведем остальные параметры, которые нам не обходимы для метода, такие как начало и конец области определения и шаг в этой же области:

t0 = 0

tEnd = 4\*np.pi

tau = 0.25

Теперь применим наш метод Рунге-Кутта и получим два массива которые в свою очередь выразим в виде графиков и рассмотрим их подробнее:

x, y = rungeKutta(f, t0, y0, tEnd, tau)

plt.plot(x, y[:, 0], color='black', label='u')

plt.plot(x, y[:, 1], color='black', ls='--', label='u\'')

plt.xlabel("t (0 < t < 4pi)")

plt.ylabel("u")

plt.legend()

plt.show()

Полученный результат выведен ниже:

|  |
| --- |
|  |
| Рисунок 1 – Графики полученных функций (u и u’) при |

В рассматриваемой задаче (колебания маятника) при малых  имеем  и уравнение имеет период равный . Нелинейность проявляется в частности, в увеличении периода.

**3. Примеры использования метода Рунге-Кутта 4-го порядка**

Так как на одном примере сложно проверить работоспособность метода, особенно учитывая сложное уравнение, взятое в примере, поэтому рассмотрим примеры использования метода Рунге-Кутта 4-го порядка от простого к сложному.

Пример 1. 

Реализуем функцию, введем необходимые данные, а также выведем наш результат на графике:

def f(x, y):

    return x\*y/2

x0 = 0

xEnd = 1

y0 = 1

tau = 0.25

x, y = rungeKutta(f, x0, y0, xEnd, tau)

plt.plot(x, y, color='black', label='y')

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.show()

|  |
| --- |
|  |
| Рисунок 2 – График функции из примера 1 |

Сравним нашу функцию с функцией которая является аналитически выведенной ():

def f(x):

    return math.exp((x\*\*2)/4)

plt.plot(x, y, color='black', label='y')

X = np.arange(0, 1, 0.01)

plt.plot(X, tuple(map(f, X)), color='black', ls='--', label='real y')

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.show()

|  |
| --- |
|  |
| Рисунок 3 – Сравнение полученного графика и аналитического ответа при |

Пример 2. 

Данное уравнение имеет вид обыкновенного дифференциального уравнения свободных затухающих колебаний, где  – колеблющаяся величина описывающая физический процесс, а  – коэффициент затухания.

Представим уравнение второго порядка в виде системы из уравнений первого порядка воспользовавшись заменой , тогда получим следующую замену:

 (3.1)

Теперь воспользуемся краевыми условиями и с их помощью них составить нашу функцию в виде:

def f(x, y):

    delta = 0.5

    omega = 2

    f = np.zeros((2), 'float')

    f[0] = y[1]

    f[1] = -2\*delta\*f[0] - omega\*\*2\*y[0]

    return f

Далее введем остаточные данные (взяв диапазон от 0 до 10), применим метод Рунге-Кутты и изобразим наш график функций:

|  |
| --- |
|  |
| Рисунок 4 – Графики полученных функций (S и S’) при |

Теперь сравним наш график с графиком функции найденным аналитически:

def S(x):

    return (1/15)\*math.exp(-x/2)\*(math.sqrt(15)\*math.sin(math.sqrt(15)\*x/2)+

                                15\*math.cos(math.sqrt(15)\*x/2))

plt.plot(x, y[:, 0], color='black', label='S')

X = np.arange(0, 10, 0.01)

Y = tuple(map(S, X))

plt.plot(X, Y, color="black", ls='--', label='real S')

plt.xlabel("t (0 < t < 10)")

plt.ylabel("S")

plt.legend()

plt.show()

|  |
| --- |
|  |
| Рисунок 5 – Сравнение полученного графика и аналитического ответа при |

**ЗАКЛЮЧЕНИЕ**

По итогу курсового проекта был изучен метод Рунге-Кутта 4-го порядка, а также рассмотрен этот метод на нескольких моделях в числе которых есть дифференциальное уравнение свободных затухающих колебаний.

Также были сравнены полученные с помощью метода функции с функциями полученные аналитическим методом. Было видно, что метод является весьма точным даже при большом интервале тау, помимо этого в данном курсовом проекте не учитывается возможность аппроксимировать результат использования метода Рунге-Кутты, что подтверждает его точность. Также данный метод легко реализуем в программном коде что делает его достаточно популярным при решении обыкновенных дифференциальных уравнений. Также данный метод в рамках является достаточно обширным, но проблема остается за представлением уравнения в виде функции в коде, так как уравнения могут быть разных порядков или используя две разные переменные, что делает метод достаточно сложно автоматизируемым.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Вабищевич, П. Н. Численные методы. Вычислительный практикум. Практическое применение численных методов при использовании алгоритмического языка PYTHON / П. Н. Вабищевич. – 4-е изд., стер. – М. : ЛЕНАНД, 2021. – 250с.
2. Численные методы в математическом моделировании: учеб. пособие / Н.П. Савенкова, О.Г. Проворова, А.Ю. Мокин. – 2-е изд., испр. и доп. – М. : ИНФРА-М, 2019. – 176 с.
3. Численные методы: учебное пособие/ авт.-сост. А.С. Шевченко. – Барнаул: Изд-во Алт. ун-та, 2016. – 325 с.