**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

**Вычислительная математика**

**Лабораторная работа №6**

**Вариант 2**

Группа: P3266

Выполнила:

Бореева В. Ю.  
Тихонов И. В.

Проверил:

Машина Е. А.

Г. Санкт-Петербург

2024

# Цель работы

Решить задачу Коши для обыкновенных

дифференциальных уравнений численными методами.

# 2 Программная часть задачи:

**from** Equations **import** calculate

**def** adams(equation, x0, lower, upper, yarr, farr, h, e):

xarr = [x0]

i = lower

**while** i < upper + h:

**if** i != lower:

xarr.append(round(i, 6))

i += h

**return** \_\_adams(equation, xarr, yarr, farr, h)

**def** \_\_adams(equation, xarr, yarr, farr, h):

**for** i **in** range(3, len(xarr)):

f1 = farr[i] - farr[i - 1]

f2 = farr[i] - 2 \* farr[i - 1] + farr[i - 2]

f3 = farr[i] - 3 \* farr[i - 1] + 3 \* farr[i - 2] - farr[i - 3]

yarr.append(round(yarr[i] + h \* farr[i] + f1 \* (h \*\* 2) / 2 + f2 \* 5 \* (h \*\* 3) / 12 + f3 \* 3 \* (h \*\* 4) / 8,6))

**if** i != len(xarr) - 1:

farr.append(calculate(equation, xarr[i + 1], yarr[i + 1]))

**return** xarr, yarr, farr

**import** math

**def** calculate(equation, x, y):

**if** equation == 1:

**return** round(y + (1 + x) \* (y \*\* 2), 6)

**elif** equation == 2:

**return** round(3 \* (x \*\* 2) \* y + (x \*\* 2) \* (math.e \*\* (x \*\* 3)), 6)

**elif** equation == 3:

**return** round(-y \* math.cos(x) + math.sin(x) \* math.cos(x), 6)

**def** runge(y, h, p, e):

**return** (y \*\* h - y \*\* (h / 2)) / (2 \*\* p - 1) <= e

**from** Equations **import** calculate, runge

**def** euler(equation, x0, y0, lower, upper, h, e):

xarr = [x0]

yarr = [y0]

i = lower

**while** i < upper + h:

**if** i != lower:

xarr.append(round(i, 6))

i += h

**if** e **is** **None**:

**return** \_\_euler(equation, xarr, yarr, h)

**else**:

**while** **True**:

solution = \_\_euler(equation, xarr, yarr, h)

**if** runge(solution[1], h, 1, e):

**return** solution

**else**:

h /= 2

xarr = [x0]

**while** i < upper + h:

**if** i != lower:

xarr.append(round(i, 6))

i += h

**def** \_\_euler(equation, xarr, yarr, h):

farr = []

**for** i **in** range(len(xarr) - 1):

farr.append(calculate(equation, xarr[i], yarr[i]))

yarr.append(round(yarr[i] + h \* farr[i], 6))

**return** xarr, yarr, farr

**from** Euler **import** euler

**from** RungeKutt **import** runge\_kutt

**from** Adams **import** adams

y0, x0, h, lower, upper, e = 0, 0, 0, 0, 0, 0

equation = int(input(

"Выберите задачу Коши:\n1. y' = y + (1 + x)y^2; (y(1) = -1; h = 0.1; интервал: [1; 1.5])\n2. y' = 3x^2y + "

"x^2e^(x^3); (y(0) = 0; h = 0.1; интервал: [0; 1])\n3. y' = -ycos(x) + sin(x)cos(x); (y(0) = 1; h= 0.2; "

"интервал: [0; 10])\n> "))

**if** equation < 1 **or** equation > 3:

**raise** ValueError("Неизвестная задача.")

datamode = int(input("Выберите метод ввода данных:\n1. Вручную\n2. Из файла\n> "))

**if** datamode == 1:

y0 = int(input("Введите y0: "))

x0 = int(input("Введите x0: "))

h = int(input("Введите h: "))

lower = int(input("Введите нижнюю границу интервала: "))

upper = int(input("Введите верхнюю границу интервала: "))

e = int(input("Введите точность: "))

**elif** datamode == 2:

**if** equation == 1:

y0 = -1

x0 = 1

h = 0.1

lower = 1

upper = 1.5

e = **None**

**elif** equation == 2:

y0 = 0

x0 = 0

h = 0.1

lower = 0

upper = 1

e = **None**

**elif** equation == 3:

y0 = 1

x0 = 0

h = 0.2

lower = 0

upper = 10

e = **None**

**else**:

**raise** ValueError("Неизвестный режим ввода данных.")

solution = euler(equation, x0, y0, lower, upper, h, e)

print(f"Метод Эйлера:\nx: {solution[0]}\ny: {solution[1]}\nf(x,y): {solution[2]}")

farr = solution[2]

solution = runge\_kutt(equation, x0, y0, lower, upper, h, e)

print(f"Метод Рунге-Кутта:\nx: {solution[0]}\ny: {solution[1]}")

**if** len(solution[1]) < 4:

**raise** ValueError("Использование метода Адамса невозможно.")

**else**:

solution = adams(equation, x0, lower, upper, solution[1][0:4], farr, h, e)

print(f"Метод Адамса:\nx: {solution[0]}\ny: {solution[1]}")

**from** Equations **import** calculate, runge

**def** runge\_kutt(equation, x0, y0, lower, upper, h, e):

xarr = [x0]

yarr = [y0]

i = lower

**while** i < upper + h:

**if** i != lower:

xarr.append(round(i, 6))

i += h

**if** e **is** **None**:

**return** \_\_runge\_kutt(equation, xarr, yarr, h)

**else**:

**while** **True**:

solution = \_\_runge\_kutt(equation, xarr, yarr, h)

**if** runge(solution[1], h, 4, e):

**return** solution

**else**:

h /= 2

xarr = [x0]

**while** i < upper + h:

**if** i != lower:

xarr.append(round(i, 6))

i += h

**def** \_\_runge\_kutt(equation, xarr, yarr, h):

**for** i **in** range(len(xarr) - 1):

k1 = h \* calculate(equation, xarr[i], yarr[i])

k2 = h \* calculate(equation, xarr[i] + h / 2, yarr[i] + k1 / 2)

k3 = h \* calculate(equation, xarr[i] + h / 2, yarr[i] + k2 / 2)

k4 = h \* calculate(equation, xarr[i] + h, yarr[i] + k3)

yarr.append(round(yarr[i] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6, 6))

**return** xarr, yarr

ВЫВОД

В ходе выполнения лабораторной работы мы решили задачу Коши для обыкновенных

дифференциальных уравнений численными методами.