

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

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №6
по дисциплине
‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №18

Выполнил:
Студент группы Р3213
Хафизов Булат Ленарович
Преподаватель:
Мальшева Татьяна
Алексеевна



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2022

Цель работы

Изучить численные методы дифференцирования и реализовать два из них средствами программирования. Понять их сходства и различия.

Листинг программы

```
import numpy as np
import matplotlib.pyplot as plt
from math import exp

def runge_method(f, a, b, y0, h):
    dots = [(a, y0)]
    n = int((b - a) / h)
    for i in range(0, n):
        k1 = h * f(dots[i][0], dots[i][1])
        k2 = h * f(dots[i][0] + h / 2, dots[i][1] + k1 / 2)
        k3 = h * f(dots[i][0] + h / 2, dots[i][1] + k2 / 2)
        k4 = h * f(dots[i][0] + h, dots[i][1] + k3)
        dots.append((dots[i][0] + h, dots[i][1] + (k1 + 2 * k2 + 2 * k3 + k4)
/ 6))
    return dots

def adams_method(f, a, b, y0, h):
    n = int((b - a) / h)
    b0 = min(b, a + 3 * h)
    dots = runge_method(f, a, b0, y0, h)
    for i in range(4, n + 1):
        df = f(dots[i - 1][0], dots[i - 1][1]) - f(dots[i - 2][0], dots[i -
2][1])
        d2f = f(dots[i - 1][0], dots[i - 1][1]) - 2 * f(dots[i - 2][0],
dots[i - 2][1]) + \
            f(dots[i - 3][0], dots[i - 3][1])
        d3f = f(dots[i - 1][0], dots[i - 1][1]) - 3 * f(dots[i - 2][0],
dots[i - 2][1]) + \
            3 * f(dots[i - 3][0], dots[i - 3][1]) - f(dots[i - 4][0], dots[i
- 4][1])
        dots.append((dots[i - 1][0] + h,
                    dots[i - 1][1] + h * f(dots[i - 1][0], dots[i - 1][1]) +
                    (h ** 2) * df / 2 + 5 * (h ** 3) * d2f / 12 + 3 * (h **
4) * d3f / 8))
    return dots

def acc(solve_prev, solve_next):
    mx = 0
    for i in range(len(solve_next)):
        mx = max(mx, abs((solve_prev[i // 2][1] - solve_next[i][1]) / (
2 ** 3
        )))
    return mx

def plot(x, y, acc_x, acc_y):
    plt.gcf().canvas.manager.set_window_title("График")
    ax = plt.gca()
    ax.spines['left'].set_position('zero')
    ax.spines['bottom'].set_position('zero')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.plot(1, 0, marker=">", ms=5, color='k',
```

```

        transform=ax.get_yaxis_transform(), clip_on=False)
ax.plot(0, 1, marker="^", ms=5, color='k',
        transform=ax.get_xaxis_transform(), clip_on=False)
plt.plot(x, y, label="y(x)")
plt.plot(acc_x, acc_y, label="acc_y(x)")
plt.legend()
plt.show()

def getTask(task):
    if task == '1':
        return lambda x, y: y + (1 + x) * (y ** 2), \
            lambda x: -1 / x, \
            1, \
            1.5, \
            -1
    elif task == '2':
        return lambda x, y: (x ** 2) - 2 * y, \
            lambda x: 0.75 * exp(-2 * x) + 0.5 * (x ** 2) - 0.5 * x +
0.25, \
            0, \
            1, \
            1
    else:
        return None

def getdata_input():
    data = {}
    print("Выберите метод дифференцирования.")
    print(" 1 - Метод Рунге-Кутты 4-го порядка")
    print(" 2 - Метод Адамса")
    while True:
        method = input("Метод дифференцирования: ")
        if method == '1' or method == '2':
            break
        print("Метода нет в списке!")
    data['method'] = method
    print("Выберите задачу.")
    print(" 1 - y' = y + (1 + x)y^2\n      на [1; 1,5] при y(1) = -1")
    print(" 2 - y' = x^2 - 2y\n      на [0; 1] при y(0) = 1")
    while True:
        task = input("Задача: ")
        func, acc_func, a, b, y0 = getTask(task)
        if func is not None:
            break
        print("Функции нет в списке!")
    data['f'] = func
    data['acc_f'] = acc_func
    data['a'] = a
    data['b'] = b
    data['y0'] = y0
    print("Введите шаг точек.")
    while True:
        h = float(input("Шаг точек: "))
        if h > 0:
            break
        print("Шаг точек должен быть положительным числом.")
    data['h'] = h
    return data

def main():
    print("Лабораторная работа №6")

```

```

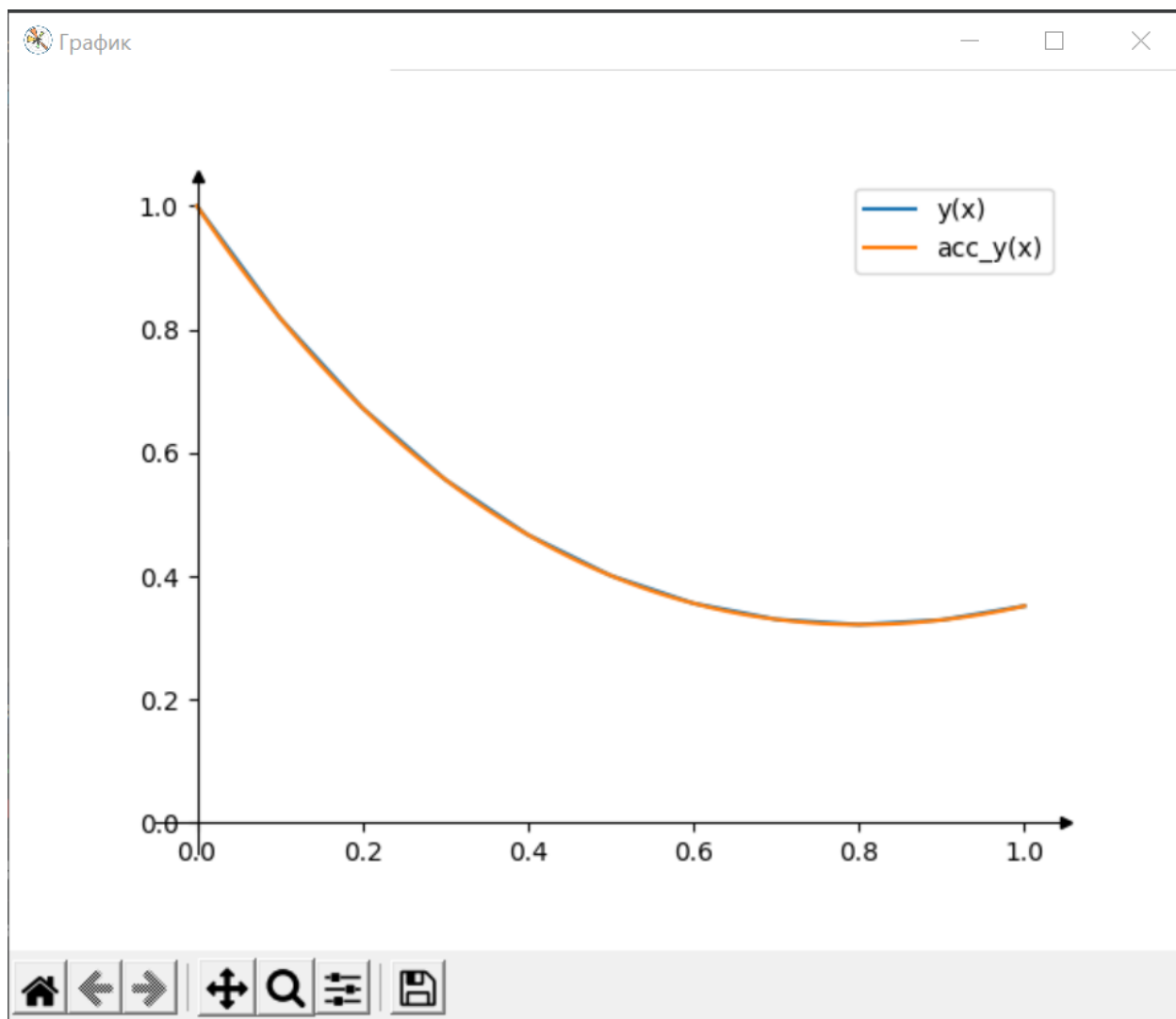
print("Вариант №18")
print("Численное дифференцирование")
data = getdata_input()
if data['method'] == '1':
    h_next = data['h'] / 2
    solve_prev = runge_method(data['f'], data['a'], data['b'],
data['y0'], data['h'])
    solve_next = runge_method(data['f'], data['a'], data['b'],
data['y0'], h_next)
    while acc(solve_prev, solve_next) > 0.01:
        solve_prev = solve_next
        h_prev = h_next
        h_next = h_prev / 2
        solve_next = runge_method(data['f'], data['a'], data['b'],
data['y0'], h_next)
    answer = solve_next
elif data['method'] == '2':
    h_next = data['h'] / 2
    solve_prev = adams_method(data['f'], data['a'], data['b'],
data['y0'], data['h'])
    solve_next = adams_method(data['f'], data['a'], data['b'],
data['y0'], h_next)
    while acc(solve_prev, solve_next) > 0.01:
        solve_prev = solve_next
        h_prev = h_next
        h_next = h_prev / 2
        solve_next = adams_method(data['f'], data['a'], data['b'],
data['y0'], h_next)
    answer = solve_next
else:
    answer = None
if answer is None:
    print("Во время вычисления произошла ошибка!")
else:
    x = np.array([dot[0] for dot in answer])
    y = np.array([dot[1] for dot in answer])
    acc_x = np.linspace(np.min(x), np.max(x), 100)
    acc_y = [data['acc_f'](i) for i in acc_x]
    plot(x, y, acc_x, acc_y)

    print("Результаты вычисления:")
    print("%12s%12s%12s" % ("x", "y", "acc_y"))
    for i in range(len(answer)):
        print("%12.4f%12.6f%12.6f" % (answer[i][0], answer[i][1],
data['acc_f'](answer[i][0])))

main()

```

Результаты выполнения программы



Лабораторная работа №6

Вариант №18

Численное дифференцирование

Выберите метод дифференцирования.

1 – Метод Рунге-Кутты 4-го порядка

2 – Метод Адамса

Метод дифференцирования: 1

Выберите задачу.

1 – $y' = y + (1 + x)y^2$

на $[1; 1,5]$ при $y(1) = -1$

2 – $y' = x^2 - 2y$

на $[0; 1]$ при $y(0) = 1$

Задача: 2

Введите шаг точек.

Шаг точек: 0.1

Результаты вычисления:

x	y	асс_y
0.0000	1.0000	1.0000
0.1000	0.8191	0.8190
0.2000	0.6727	0.6727
0.3000	0.5566	0.5566
0.4000	0.4670	0.4670
0.5000	0.4009	0.4009
0.6000	0.3559	0.3559
0.7000	0.3300	0.3299
0.8000	0.3214	0.3214
0.9000	0.3290	0.3290
1.0000	0.3515	0.3515

Вывод

В результате выполнения данной лабораторной работой я познакомился с численными методами дифференцирования и реализовал метод Рунге-Кутты 4-го порядка и метод Адамса на языке программирования Python, закрепив знания. Также я повторил виды дифференциальных уравнений, определение задачи Коши и некоторые методы их решения.