

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

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по дисциплине Вычислительная Математика
№6 «ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ»**

Вариант 12

Преподаватель: Машина Екатерина Алексеевна

Выполнил: Печкуров Данила Алексеевич

Группа: P3208

Санкт-Петербург, 2024г

Цель работы:

решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

Листинг программы:

```
def euler(self, x0, y0, h, n):
    x_values = [x0]
    y_values = [y0]
    x = x0
    y = y0

    for i in range(n):
        y += h * self.f(x, y)
        x += h
        x_values.append(x)
        y_values.append(y)

    return x_values, y_values

def runge_kutta_4(self, x0, y0, h, n):
    x_values = [x0]
    y_values = [y0]
    x = x0
    y = y0

    for i in range(n):
        k1 = h * self.f(x, y)
        k2 = h * self.f(x + 0.5 * h, y + 0.5 * k1)
        k3 = h * self.f(x + 0.5 * h, y + 0.5 * k2)
        k4 = h * self.f(x + h, y + k3)

        y += (k1 + 2 * k2 + 2 * k3 + k4) / 6
        x += h
        x_values.append(x)
        y_values.append(y)

    return x_values, y_values

def milne(self, x0, y0, h, n):
    x_values = [x0]
    y_values = [y0]
    x = x0
    y = y0

    for i in range(3):
        k1 = h * self.f(x, y)
        k2 = h * self.f(x + 0.5 * h, y + 0.5 * k1)
        k3 = h * self.f(x + 0.5 * h, y + 0.5 * k2)
        k4 = h * self.f(x + h, y + k3)
        y += (k1 + 4 * k2 + k3) * h / 3
```

```

        x += h
        x_values.append(x)
        y_values.append(y)

    for i in range(3, n):
        y_next = y_values[-1] + h / 24 * (55 * self.f(x_values[-1],
y_values[-1]) -
                                           59 * self.f(x_values[-2],
y_values[-2]) +
                                           37 * self.f(x_values[-3],
y_values[-3]) -
                                           9 * self.f(x_values[-4],
y_values[-4]))
        x += h
        x_values.append(x)
        y_values.append(y_next)

    return x_values, y_values

```

Пример и результаты работы программы:

Выберите уравнение:

1. $y' = y + (1+x)y^2$

2. $y' = y + 2x$

3. $y' = x^2 - y$

Введите номер выбранного уравнения: 1

Введите начальное значение x: 0

Введите начальное значение y: -1

Введите шаг h: 0.1

Введите количество шагов n: 5

Метод Эйлера:

i	xi	yi	Inaccuracy
0	0.0000	-1.0000	0.0000
1	0.1000	-1.0000	0.0909
2	0.2000	-0.9900	0.1567
3	0.3000	-0.9714	0.2022
4	0.4000	-0.9459	0.2316
5	0.5000	-0.9152	0.2485

Метод Рунге-Кутты 4-го порядка:

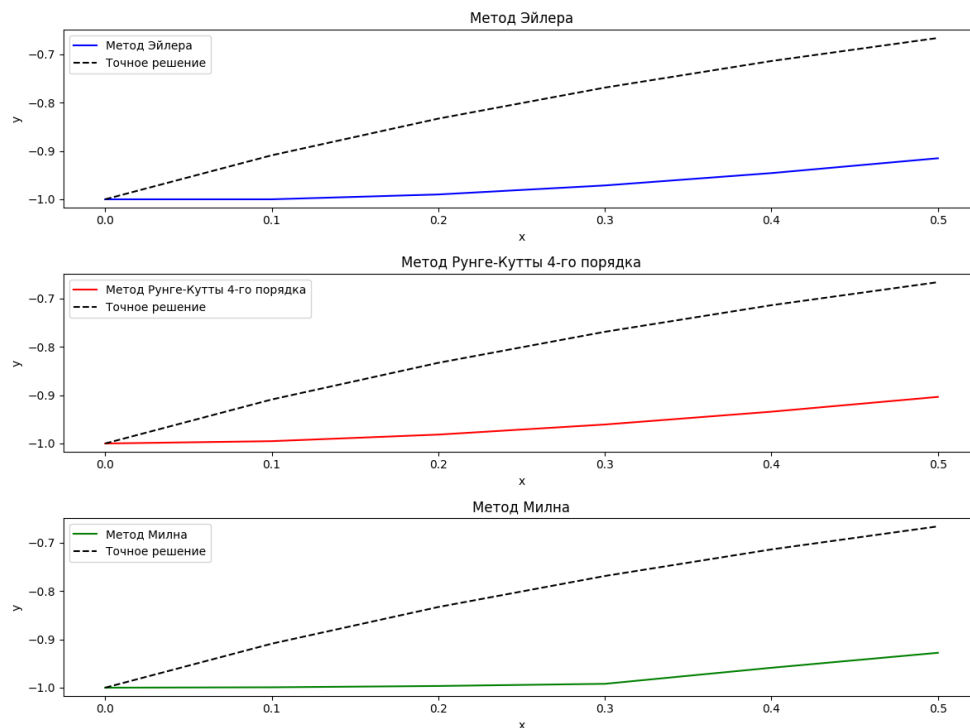
i	xi	yi	Inaccuracy
0	0.0000	-1.0000	0.0000
1	0.1000	-0.9952	0.0861
2	0.2000	-0.9816	0.1483
3	0.3000	-0.9608	0.1916
4	0.4000	-0.9343	0.2200
5	0.5000	-0.9037	0.2371

Метод Милна:

i	xi	yi	Inaccuracy
0	0.0000	-1.0000	0.0000

1	0.1000	-0.9992	0.0901
2	0.2000	-0.9965	0.1631
3	0.3000	-0.9920	0.2228
4	0.4000	-0.9589	0.2446
5	0.5000	-0.9279	0.2612

Process finished with exit code 0



Вывод:

В ходе лабораторной работы я познакомился с численным решением обыкновенных дифференциальных уравнений.

