

Лабораторная работа №3

Задание. Написать программу, которая находит решение задачи Коши указанными в варианте методами с шагом $h=10^{-3}$. Вывести графики полученных решений, а также модуль разности решений в крайней правой точке интервала, на котором задана задача.

Вариант 1.
$$\begin{cases} u' = uv + x - 1 \\ v' = \sqrt{u^2 + 1} - v \end{cases}, \quad u(1) = 1.5, \quad v(1) = 0.75, \quad x \in [1; 2], \quad \text{явный метод Эйлера}$$

(1), метод Рунге-Кутты 2-го порядка (аналог формулы средних прямоугольников) (5).

$$u' = 4u + 6v + e^x$$

Методы, используемые в лабораторной.

Явный метод Эйлера:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad y_0 = u_0. \quad (1)$$

Метод Рунге-Кутты 2-го порядка (аналог формулы средних прямоугольников):

$$\begin{aligned} y_{i+1} &= y_i + h\varphi_1, \\ \varphi_0 &= f(x_i, y_i), \\ \varphi_1 &= f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}\varphi_0\right), \\ y_0 &= u_0. \end{aligned} \quad (5)$$

Код программы:

```
import numpy as np
import matplotlib.pyplot as plt

# Определение функций для системы уравнений
def f(x, y):
    U, V = y
    return np.array([U * V + x - 1, np.sqrt(U ** 2 + 1) - V])

def display(x_values, y_values, title):
    u_values = y_values[:, 0]
    v_values = y_values[:, 1]
    plt.figure(figsize=(10, 6))
    plt.plot(x_values, u_values, label='u(x)')
    plt.plot(x_values, v_values, label='v(x)')
    plt.title(title)
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()
    plt.grid(True)
    plt.show()

# Явный метод Эйлера
def euler_method(f, h, y0, x_values):
    y_values = np.zeros((len(x_values), len(y0)))
    y_values[0] = y0
    for i in range(1, len(x_values)):
        xi = x_values[i]
        yi = y_values[i-1]
```

```

        yi_temp = yi + h*f(xi, yi)
        y_values[i] = yi_temp

    return x_values, y_values

# Метод Рунге-Кутты 2-го порядка
def runge_kutta_2nd_order(f, h, y0, x_values):
    y_values = np.zeros((len(x_values), len(y0)))
    y_values[0] = y0
    for i in range(1, len(x_values)):
        xi = x_values[i]
        yi = y_values[i - 1]
        phi0 = f(xi, yi)
        phi1 = f(xi + h/2, yi + (h/2)*phi0)
        yi_temp = yi + h * phi1
        y_values[i] = yi_temp

    return x_values, y_values

# Начальные условия
startData = np.array([1.5, 0.75])
h = 1e-3
x_range = [1, 2]
N = int((x_range[1] - x_range[0]) / h)
x_values = np.linspace(x_range[0], x_range[1], N + 1)

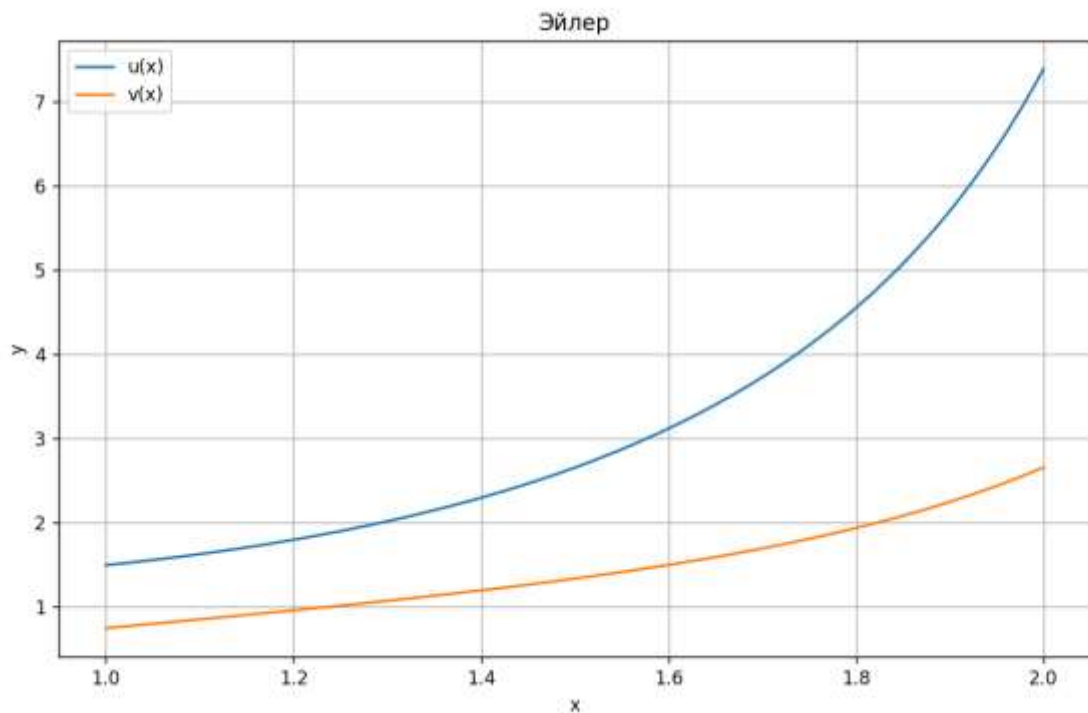
# Вычисление решений методом Эйлера
x_euler, y_euler = euler_method(f, h, startData, x_values)
# Вычисление решений методом Рунге-Кутты 2-го порядка
x_rk2, y_rk2 = runge_kutta_2nd_order(f, h, startData, x_values)

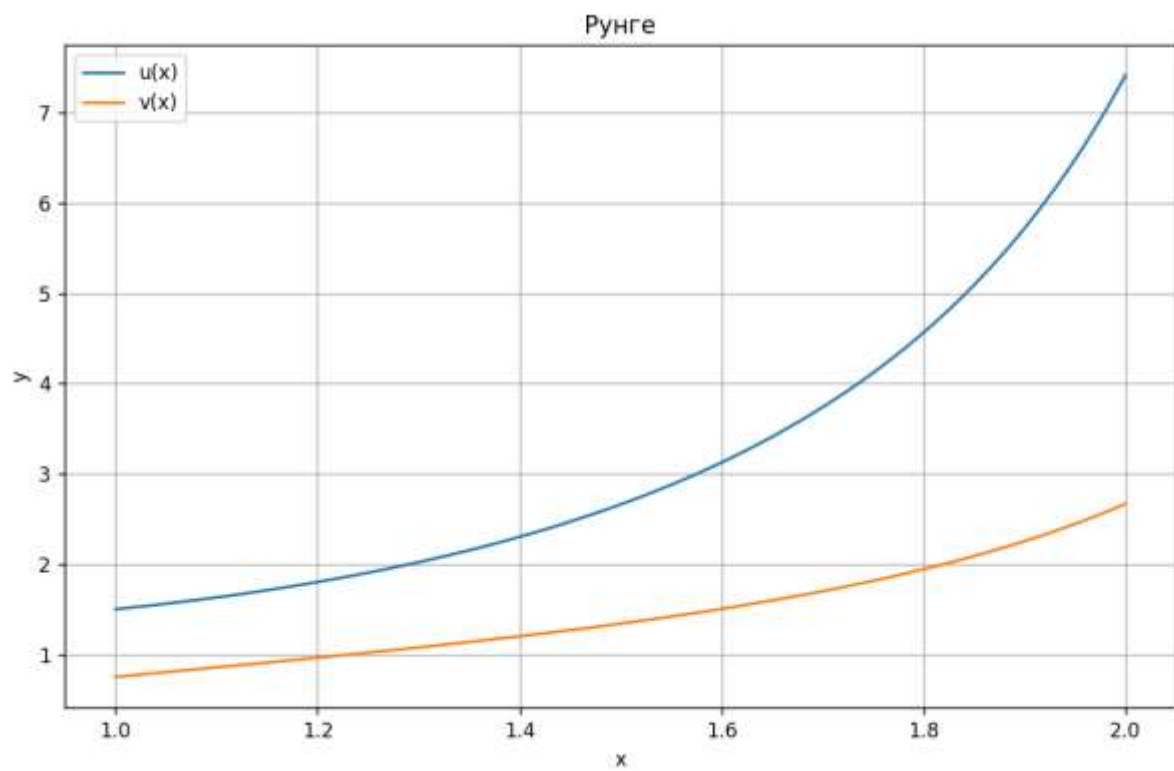
# Построение графиков
display(x_euler, y_euler, "Эйлер")
display(x_rk2, y_rk2, "Рунге")

y1 = y_euler[-1]
y2 = y_rk2[-1]
print("Модуль разности решений в крайней правой точке интервала: ", abs(y1-y2))

```

Результат выполнения:





```
C:\Python\Programming\01-030\030-01\venv\Scripts\python.exe C:\Python\Programming\01-030\030-01\venv\Scripts\python.exe 0.02479495 0.00531616
```

Модуль разности решений в крайней правой точке интервала: [0.02479495 0.00531616]