МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальная научно-образовательная корпорация ИТМО»

ФАКУЛЬТЕТ ПиикТ

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине
«Вычислительная математика»
Вариант №3

Выполнил:
Студент группы P3219
Билобрам Денис Андреевич
Преподаватель:
Бострикова Дарья Константиновна

Санкт-Петербург, 2024

1. Цель лабораторной работы:

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

2. Описание алгоритма решения задачи

Для численного решения обыкновенных дифференциальных уравнений (ОДУ) реализованы одношаговые методы (метод Эйлера, усовершенствованный метод Эйлера, метод Рунге-Кутта 4-го порядка) и многошаговые методы (методы Адамса и Милна). Пользователь выбирает ОДУ и вводит исходные данные: начальные условия, интервал дифференцирования, шаг h, точность є. Программа решает задачу Коши и строит таблицу приближенных значений, оценивая точность методов.

3. Рабочие формулы используемых методов

Метод Эйлера

$$y_{n+1} = y_n + hf(x_n, y_n)$$

Метод Рунге-Кутта 4-го порядка

$$k_1 = f(x_n, y_n)$$
$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$
$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$
$$k_4 = f(x_n + h, y_n + hk_3)$$
$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Метод Милна

$$y_{n+1} = y_{n-3} + \frac{4h}{3}\left(2f(x_n, y_n) - f(x_{n-1}, y_{n-1}) + 2f(x_{n-2}, y_{n-2})\right)$$

4. Листинг программы

```python
# Метод Эйлера
def euler_method(f, y0, x0, xn, h):
    n = int((xn - x0) / h)
    x = np.linspace(x0, xn, n + 1)
    y = np.zeros(n + 1)
    y[0] = y0
    for i in range(n):
        y[i + 1] = y[i] + h * f(x[i], y[i])
    return x, y

# Метод Рунге-Кутты 4-го порядка
def runge_kutta_4th_method(f, y0, x0, xn, h):
    n = int((xn - x0) / h)
    x = np.linspace(x0, xn, n + 1)
    y = np.zeros(n + 1)
    y[0] = y0
    for i in range(n):
        k1 = h * f(x[i], y[i])
        k2 = h * f(x[i] + h / 2, y[i] + k1 / 2)
        k3 = h * f(x[i] + h / 2, y[i] + k2 / 2)
        k4 = h * f(x[i] + h, y[i] + k3)
        y[i + 1] = y[i] + (k1 + 2 * k2 + 2 * k3 + k4) / 6
    return x, y

# Метод Милна
def milne_method(f, y0, x0, xn, h):
    def rk4_initial_values(f, y0, x0, h, steps=3):
        x = [x0 + i * h for i in range(steps + 1)]
        y = [y0]
        for i in range(steps):
            k1 = h * f(x[i], y[i])
            k2 = h * f(x[i] + h / 2, y[i] + k1 / 2)
            k3 = h * f(x[i] + h / 2, y[i] + k2 / 2)
            k4 = h * f(x[i] + h, y[i] + k3)
            y.append(y[i] + (k1 + 2 * k2 + 2 * k3 + k4) / 6)
        return np.array(x), np.array(y)

    x_rk, y_rk = rk4_initial_values(f, y0, x0, h)
    n = int((xn - x0) / h)
    x = np.linspace(x0, xn, n + 1)
    y = np.zeros(n + 1)

    y[:4] = y_rk[:4]
    x[:4] = x_rk[:4]

    for i in range(3, n):
        y_pred = y[i-3] + 4 * h * (2*f(x[i-2], y[i-2]) - f(x[i-1], y[i-1]) + 2*f(x[i], y[i])) / 3
        y_corr = y[i-1] + h * (f(x[i-1], y[i-1]) + 4 * f(x[i], y[i]) + f(x[i+1], y_pred)) / 3
        y[i+1] = y_corr

    return x, y
```

5. Результаты выполнения программы

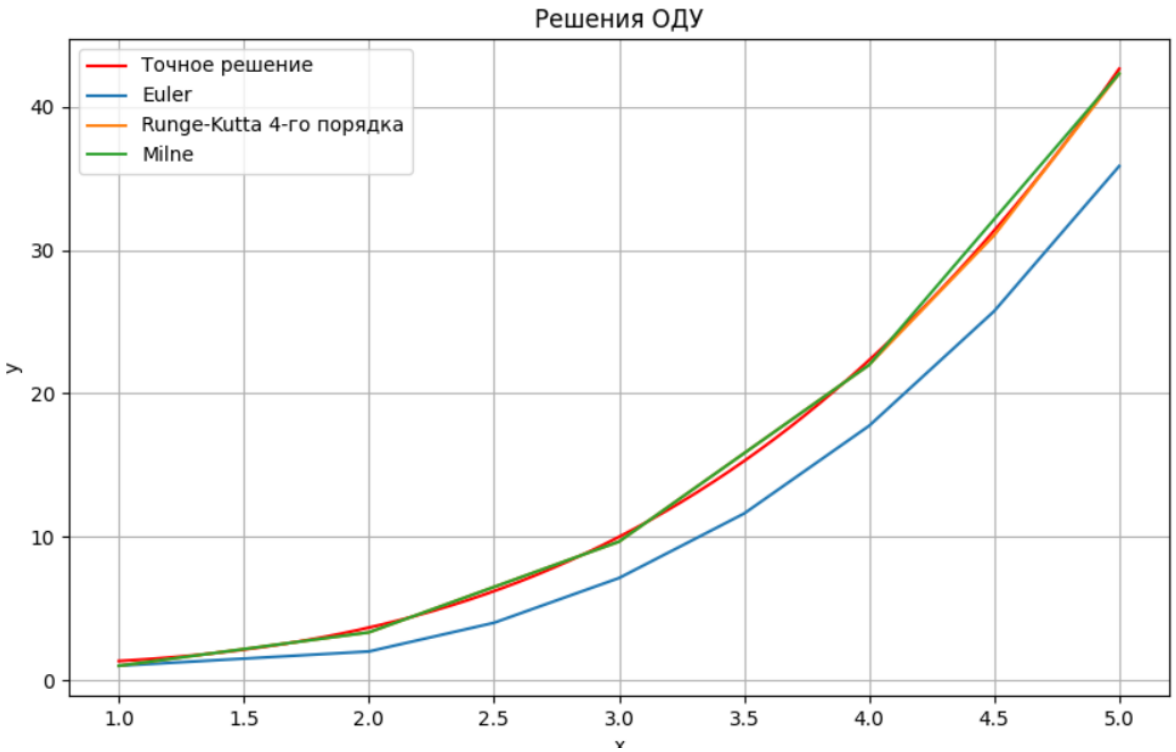

Таблица результатов

Метод: Euler

x	y	Погрешность
1.00000	1.00000	0.00000
2.00000	2.00000	0.50000
2.50000	4.00000	1.37500
3.00000	7.12500	2.50000
3.50000	11.62500	3.87500
4.00000	17.75000	5.50000
4.50000	25.75000	7.37500
5.00000	35.87500	9.50000

Метод: Runge-Kutta 4-го порядка

x	y	Погрешность
1.00000	1.00000	0.00000
2.00000	3.33333	0.10278
3.00000	9.66667	0.42222
4.00000	22.00000	1.07500
4.50000	31.04167	1.42500
5.00000	42.33333	1.82500

Метод: Milne

x	y
1.00000	1.00000
2.00000	3.33333
3.00000	9.66667
4.00000	22.00000
5.00000	42.33333

Максимальная погрешность: 0.33333

6. Выводы

В ходе работы были изучены и реализованы различные численные методы решения обыкновенных дифференциальных уравнений, включая методы Эйлера, Рунге-Кутта и Милна. Были исследованы их точность и применимость для решения задачи Коши.

Метод Эйлера:
Преимущества: Простота реализации и вычислительная эффективность.
Недостатки: Низкая точность и необходимость малого шага для получения приемлемых результатов, что увеличивает число вычислений.

Метод Рунге-Кутта 4-го порядка:
Преимущества: Высокая точность при относительно большом шаге, устойчивость и универсальность.
Недостатки: Более сложная реализация и большее число вычислений на каждом шаге по сравнению с методом Эйлера.
Метод Милна:

Преимущества: Высокая точность при меньшем числе вычислений на шаг, эффективен при работе с гладкими функциями.
Недостатки: Сложность начального запуска, требующего использования других методов (например, метода Рунге-Кутта) для вычисления первых нескольких значений, а также снижение точности и устойчивости при наличии резких изменений в решении.

Можно сделать вывод, что для задач, требующих высокой точности и допускающих сложные вычисления, предпочтительно использовать метод Рунге-Кутта. Метод Эйлера подходит для простых и быстрых оценок с малыми шагами, а метод Милна эффективен для задач с гладкими решениями при наличии хороших начальных значений.