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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа **№6**

**«**Численное решение обыкновенных

Дифференциальных уравнений**»**

по дисциплине «Вычислительная математика**»**

Вариант: **18**

**Преподаватель:**

**Выполнил:**

Шаматульский Роман Константинович

**Группа:** Р3212

Санкт-Петербург, 2025 г.

Оглавление

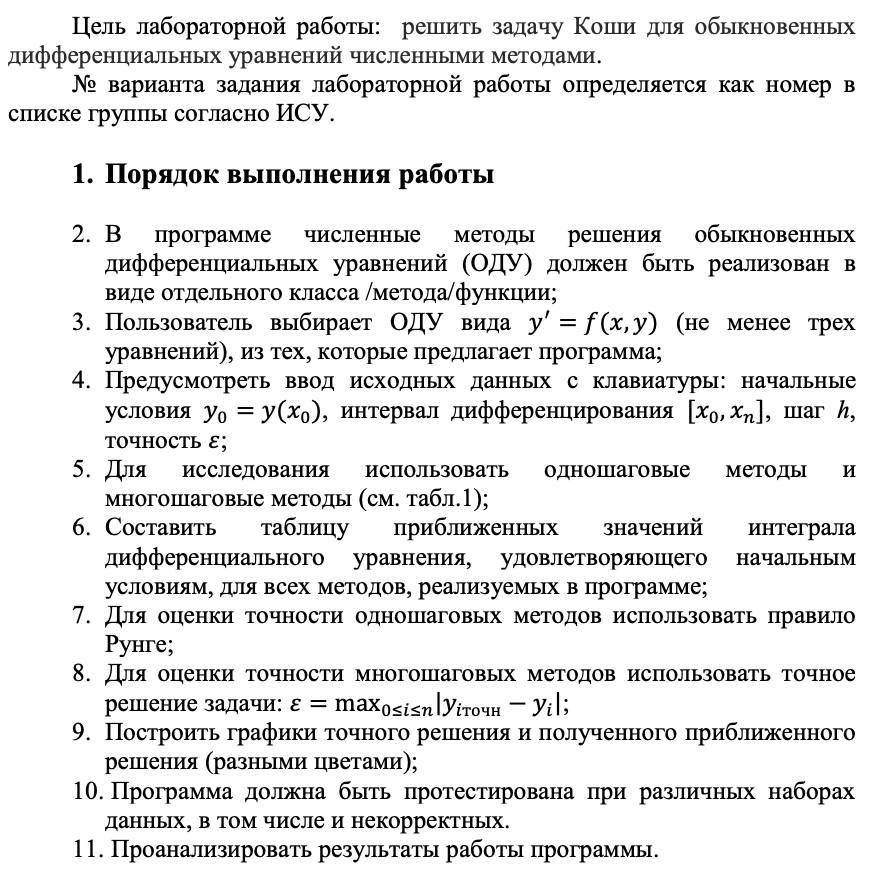
[Текст задания 3](#_Toc196928063)

[Вычислительная часть 4](#_Toc196928064)

[Программная часть 6](#_Toc196928065)

[Вывод 10](#_Toc196928066)

# Текст задания



# Описание алгоритма решения

Сначала выбираем уравнение и вводим начальные данные правую границу ​ и шаг h. Затем последовательно идём по x от до ​: зная в узле производную вычисляем приближённое значение ​ в точке по формуле выбранного метода. Повторяем, пока не достигнем , сохраняем все пары и, при необходимости, сравниваем с точным решением для оценки погрешности.

**Формулы методов:**

1. Эйлер:
2. Рунге-Кутта (4 п.):
3. Милна:

Для :

# Листинг программы

import math  
import matplotlib.pyplot as plt  
  
def eq1\_f(x, y):  
 return x + y  
  
def eq1\_exact(x, x0, y0):  
 return (y0 + 1) \* math.exp(x - x0) - x - 1 + x0  
  
def eq2\_f(x, y):  
 return y \* x \* x  
  
def eq2\_exact(x, x0, y0):  
 return y0 \* math.exp((x\*\*3 - x0\*\*3) / 3)  
  
def eq3\_f(x, y):  
 return y \* x  
  
def eq3\_exact(x, x0, y0):  
 return y0 \* math.exp((x\*\*2 - x0\*\*2) / 2)  
  
def euler(f, x0, y0, xn, h):  
 xs = []  
 ys = []  
 x = x0  
 y = y0  
 while x <= xn + 1e-12:  
 xs.append(x)  
 ys.append(y)  
 y = y + h \* f(x, y)  
 x = x + h  
 return xs, ys  
  
def rk4(f, x0, y0, xn, h):  
 xs = []  
 ys = []  
 x = x0  
 y = y0  
 while x <= xn + 1e-12:  
 xs.append(x)  
 ys.append(y)  
 k1 = f(x, y)  
 k2 = f(x + h/2, y + h \* k1 / 2)  
 k3 = f(x + h/2, y + h \* k2 / 2)  
 k4 = f(x + h, y + h \* k3)  
 y = y + h \* (k1 + 2\*k2 + 2\*k3 + k4) / 6  
 x = x + h  
 return xs, ys  
  
def milne(f, x0, y0, xn, h):  
 xs\_rk, ys\_rk = rk4(f, x0, y0, x0 + 3\*h, h)  
 xs = xs\_rk[:]  
 ys = ys\_rk[:]  
 fs = [f(xs[i], ys[i]) for i in range(4)]  
 i = 3  
 while xs[i] + h <= xn + 1e-12:  
 x\_next = xs[i] + h  
 y\_pred = ys[i-3] + (4\*h/3) \* (2\*fs[i-2] - fs[i-1] + 2\*fs[i])  
 f\_pred = f(x\_next, y\_pred)  
 y\_corr = ys[i-1] + (h/3) \* (fs[i-1] + 4\*fs[i] + f\_pred)  
 xs.append(x\_next)  
 ys.append(y\_corr)  
 fs.append(f(x\_next, y\_corr))  
 i += 1  
 return xs, ys  
  
def runge\_error(method, f, x0, y0, xn, h, p):  
 \_, y\_h = method(f, x0, y0, xn, h)  
 \_, y\_h2 = method(f, x0, y0, xn, h/2)  
 y\_end\_h = y\_h[-1]  
 y\_end\_h2 = y\_h2[-1]  
 return abs(y\_end\_h2 - y\_end\_h) / (2\*\*p - 1)  
  
def main():  
 print("Выберите уравнение:")  
 print("1) y' = x + y")  
 print("2) y' = y \* x^2")  
 print("3) y' = y \* x")  
 choice = int(input())  
 if choice == 1:  
 f = eq1\_f  
 exact = eq1\_exact  
 elif choice == 2:  
 f = eq2\_f  
 exact = eq2\_exact  
 else:  
 f = eq3\_f  
 exact = eq3\_exact  
 x0 = float(input("x0 = "))  
 y0 = float(input("y0 = "))  
 xn = float(input("xn = "))  
 h = float(input("h = "))  
 eps = float(input("epsilon = "))  
 xs\_e, ys\_e = euler(f, x0, y0, xn, h)  
 xs\_rk, ys\_rk = rk4(f, x0, y0, xn, h)  
 xs\_m, ys\_m = milne(f, x0, y0, xn, h)  
 print("\n i\t x\t Euler\t\t RK4\t\t Milne\t\t Exact")  
 N = len(xs\_e)  
 for i in range(N):  
 x\_val = xs\_e[i]  
 y\_ex = exact(x\_val, x0, y0)  
 y\_e = ys\_e[i] if i < len(ys\_e) else float('nan')  
 y\_r = ys\_rk[i] if i < len(ys\_rk) else float('nan')  
 y\_m = ys\_m[i] if i < len(ys\_m) else float('nan')  
 print(f"{i:2d}\t{x\_val:8.4f}\t{y\_e:10.6f}\t{y\_r:10.6f}\t{y\_m:10.6f}\t{y\_ex:10.6f}")  
 err\_euler = runge\_error(euler, f, x0, y0, xn, h, 1)  
 err\_rk4 = runge\_error(rk4, f, x0, y0, xn, h, 4)  
 max\_err\_m = 0.0  
 for i, xm in enumerate(xs\_m):  
 ye = exact(xm, x0, y0)  
 ym = ys\_m[i]  
 max\_err\_m = max(max\_err\_m, abs(ye - ym))  
 print(f"\nОценка погрешности Эйлера (правило Рунге): {err\_euler:.6e}")  
 print(f"Оценка погрешности RK4 (правило Рунге): {err\_rk4:.6e}")  
 print(f"Максимальная погрешность Милна: {max\_err\_m:.6e}")  
 xs\_exact = [x0 + i\*h/10 for i in range(int((xn - x0)/(h/10)) + 1)]  
 ys\_exact = [exact(x, x0, y0) for x in xs\_exact]  
 plt.plot(xs\_exact, ys\_exact, label="Exact", color="black")  
 plt.plot(xs\_e, ys\_e, label="Euler", linestyle="--")  
 plt.plot(xs\_rk, ys\_rk, label="RK4", linestyle="-.")  
 plt.plot(xs\_m, ys\_m, label="Milne", linestyle=":")  
 plt.legend()  
 plt.xlabel("x")  
 plt.ylabel("y")  
 plt.title("Numerical vs Exact Solution")  
 plt.show()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Результат выполнения**

Выберите уравнение:

1) y' = x + y

2) y' = y \* x^2

3) y' = y \* x

1

x0 = 0

y0 = 1

xn = 0.5

h = 0.05

epsilon = 1e-06

i x Euler RK4 Milne Exact

0 0.0000 1.000000 1.000000 1.000000 1.000000

1 0.0500 1.050000 1.052542 1.052542 1.052542

2 0.1000 1.105000 1.110342 1.110342 1.110342

3 0.1500 1.165250 1.173668 1.173668 1.173668

4 0.2000 1.231012 1.242805 1.242806 1.242806

5 0.2500 1.302563 1.318051 1.318051 1.318051

6 0.3000 1.380191 1.399718 1.399718 1.399718

7 0.3500 1.464201 1.488135 1.488135 1.488135

8 0.4000 1.554911 1.583649 1.583649 1.583649

9 0.4500 1.652656 1.686624 1.686624 1.686624

10 0.5000 1.757789 1.797442 1.797443 1.797443

Оценка погрешности Эйлера (правило Рунге): 1.944363e-02

Оценка погрешности RK4 (правило Рунге): 5.140813e-09

Максимальная погрешность Милна: 1.741323e-08

Выберите уравнение:

1) y' = x + y

2) y' = y \* x^2

3) y' = y \* x

1

x0 = 0

y0 = 0

xn = 1

h = 0.1

epsilon = 0.01

i x Euler RK4 Milne Exact

0 0.0000 0.000000 0.000000 0.000000 0.000000

1 0.1000 0.000000 0.005171 0.005171 0.005171

2 0.2000 0.010000 0.021403 0.021403 0.021403

3 0.3000 0.031000 0.049858 0.049858 0.049859

4 0.4000 0.064100 0.091824 0.091824 0.091825

5 0.5000 0.110510 0.148721 0.148721 0.148721

6 0.6000 0.171561 0.222118 0.222119 0.222119

7 0.7000 0.248717 0.313752 0.313752 0.313753

8 0.8000 0.343589 0.425540 0.425541 0.425541

9 0.9000 0.457948 0.559601 0.559603 0.559603

10 1.0000 0.593742 0.718280 0.718282 0.718282

Оценка погрешности Эйлера (правило Рунге): 5.955525e-02

Оценка погрешности RK4 (правило Рунге): 1.299014e-07

Максимальная погрешность Милна: 3.796089e-07

# Вывод

В лабораторной работе были реализованы три метода решения дифференциальных уравнений: метод Милна, Эйлера и Рунге-Кутта. Полученные мной навыки и знания помогут мне в дальнейшем изучении математики.