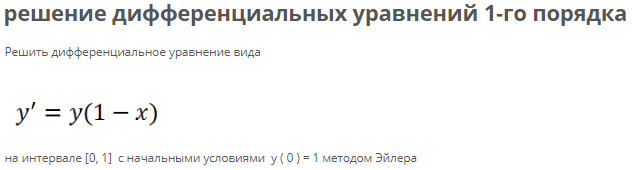
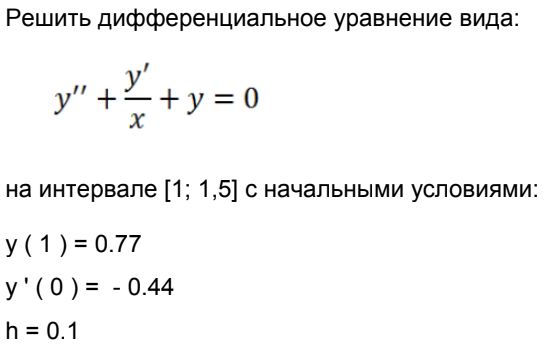
# Лабораторная работа № 3. Численные методы решения дифференциальных уравнений

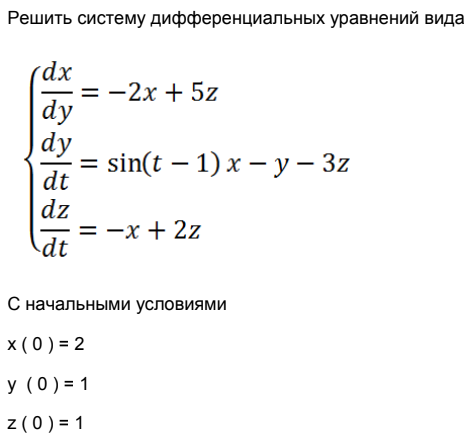
## Инструменты

Использовал интерпретатор, встроенный в Python.

## Задача



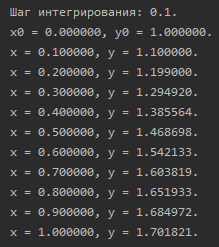




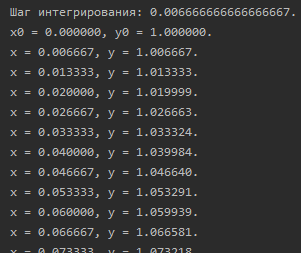
## Код программы

import math  
  
number\_of\_splits = [10, 50, 150]  
  
  
def menu(): # Меню  
 print("\nЧисленные методы решения дифференциальных уравнений\n1. Метод Эйлера\n2. Метод Рунге-Кутта\n"  
 "3. ДУ 2-го порядка\n4. Система ДУ\n5. Выйти из программы")  
 user\_input = int(input())  
 if user\_input == 1:  
 euler\_method()  
 elif user\_input == 2:  
 runge\_kutta\_method()  
 elif user\_input == 3:  
 second\_order\_de()  
 elif user\_input == 4:  
 system\_of\_de()  
 elif user\_input == 5:  
 pass  
 else:  
 print("Введите число от 1 до 5.")  
 menu()  
  
  
def euler\_method(): # Решение дифференциального уравнения методом Эйлера  
 def f(x, y):  
 return y \* (1 - x)  
  
 a = 0  
 b = 1  
 for n in number\_of\_splits:  
 h = (b - a) / n  
 xi = a # Интервал [0; 1]  
 yi = 1 # Начальное условие y(0) = 1  
 print(f"\nШаг интегрирования: {h}.")  
 print(f"x0 = {xi:.6f}, y0 = {yi:.6f}.")  
 for \_ in range(n):  
 yi += h \* f(xi, yi)  
 xi += h  
 print(f"x = {xi:.6f}, y = {yi:.6f}.")  
 menu()  
  
  
def runge\_kutta\_method(): # Решение дифференциального уравнения методом Рунге-Кутта  
 def f(x, y):  
 return y \* (1 - x)  
  
 a = 0  
 b = 1  
 for n in number\_of\_splits:  
 h = (b - a) / n  
 xi = a  
 yi = 1  
 print(f"\nШаг интегрирования: {h}.")  
 print(f"x0 = {xi:.6f}, y0 = {yi:.6f}.")  
 for \_ in range(n):  
 k1 = h \* f(xi, yi)  
 k2 = h \* f(xi + h / 2, yi + k1 / 2)  
 k3 = h \* f(xi + h / 2, yi + k2 / 2)  
 k4 = h \* f(xi + h, yi + k3)  
 yi += ((k1 + 2 \* k2 + 2 \* k3 + k4) / 6)  
 xi += h  
 print(f"x = {xi:.6f}, y = {yi:.6f}.")  
 menu()  
  
  
def second\_order\_de(): # Решение дифференциального уравнения второго порядка  
 h = 0.1  
  
 def y1(y, z):  
 return y + h \* z  
  
 def z1(x, y, z):  
 return z + h \* (-(z / x + y))  
  
 xi = 1  
 yi = 0.77  
 zi = -0.44 # y'  
 print(f"x0 = {xi:.6f}, y0 = {yi:.6f}, z0 = {zi:.6f}.")  
 while xi <= 1.5:  
 zi = z1(xi, yi, zi)  
 yi = y1(yi, zi)  
 xi += h  
 print(f"x = {xi:.6f}, y = {yi:.6f}, z = {zi:.6f}.")  
 menu()  
  
  
def system\_of\_de(): # Решение системы дифференциальных уравнений  
 def dx\_dy(x, z):  
 return -2 \* x + 5 \* z  
  
 def dy\_dt(t, x, y, z):  
 return math.sin(t - 1) \* x - y - 3 \* z  
  
 def dz\_dt(x, z):  
 return -x + 2 \* z  
  
 h = 0.001  
 b = 0.3  
 ti = 0  
 xi = 2  
 yi = 1  
 zi = 1  
 print(f"t0 = {ti:.6f}, x0 = {xi:.6f}, y0 = {yi:.6f}, z0 = {zi:.6f}.")  
 while ti < b:  
 yi += h \* dy\_dt(ti, xi, yi, zi)  
 temporary = xi # Текущий x  
 xi += h \* dx\_dy(xi, zi)  
 zi += h \* dz\_dt(temporary, yi) # Текущий x  
 print(f"t = {ti:.6f}, x = {xi:.6f}, y = {yi:.6f}, z = {zi:.6f}.")  
 ti += h  
 menu()  
  
  
menu()

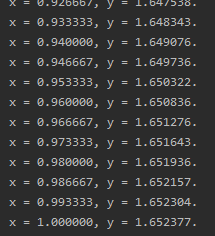
## Результаты



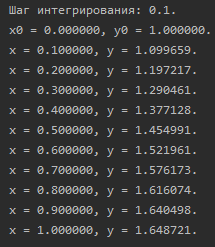
Метод Эйлера, 10 разбиений



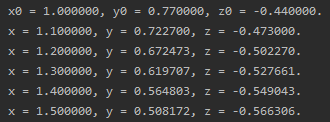
Метод Эйлера, 150 разбиений (начало)



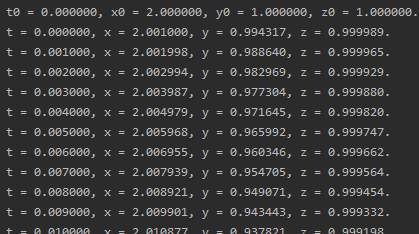
Метод Эйлера, 150 разбиений (конец)



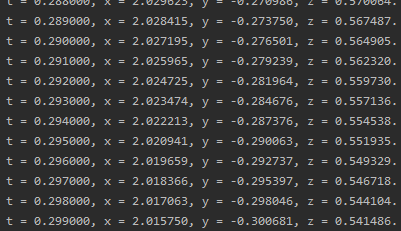
Метод Рунге-Кутта, 10 разбиений



ДУ 2-го порядка



Система ДУ (начало)



Система ДУ (конец)

## Анализ результатов

Точность вычисления зависит от разных факторов, таких как метод и шаг вычисления. Используя метод Рунге-Кутта, мы получим более точный результат, так как вычисления связаны с усредненной производной.