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

Факультет программной инженерии и компьютерных технологий

Лабораторная работа 3

**«Численное интегрирование»**

по дисциплине **«**Вычислительная математика»

Вариант 13

Выполнила:

Павличенко Софья Алексеевна, Р3215

Проверила:   
Малышева Татьяна Алексеевна

Санкт-Петербург 2025г.

Оглавление

[Цель 2](#_Toc193379022)

[Часть 1: Вычислительная реализация задачи 3](#_Toc193379023)

[Часть 2: Программная реализация задачи. 6](#_Toc193379024)

[Листинг программы 6](#_Toc193379025)

[Результат работы программы 9](#_Toc193379026)

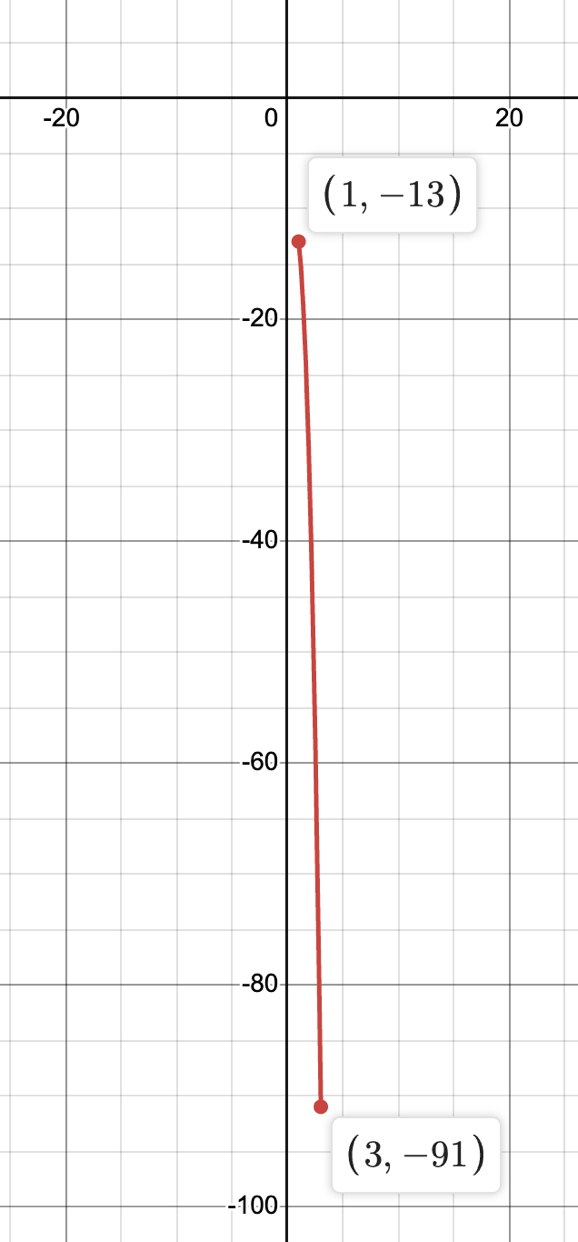
[Выводы 10](#_Toc193379027)

# Цель

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

# Часть 1: Вычислительная реализация задачи

1. **Точно**



1. **По формуле Ньютона-Котеса** при

**По формуле средних прямоугольников** при

**По формуле трапеций** при

**По формуле Симпсона** при

1. Сравним результаты по каждому методу с точным значением интеграла на интервале(
2. По формуле **Ньютона-Котеса** при :
3. По формуле **средних прямоугольников** при :
4. По формуле **трапеций** при :
5. По формуле **Симпсона** при :
6. Определим относительную погрешность вычислений для каждого метода:

1. По формуле **Ньютона-Котеса** при :

2. По формуле **средних прямоугольников** при :

3. По формуле **трапеций** при :

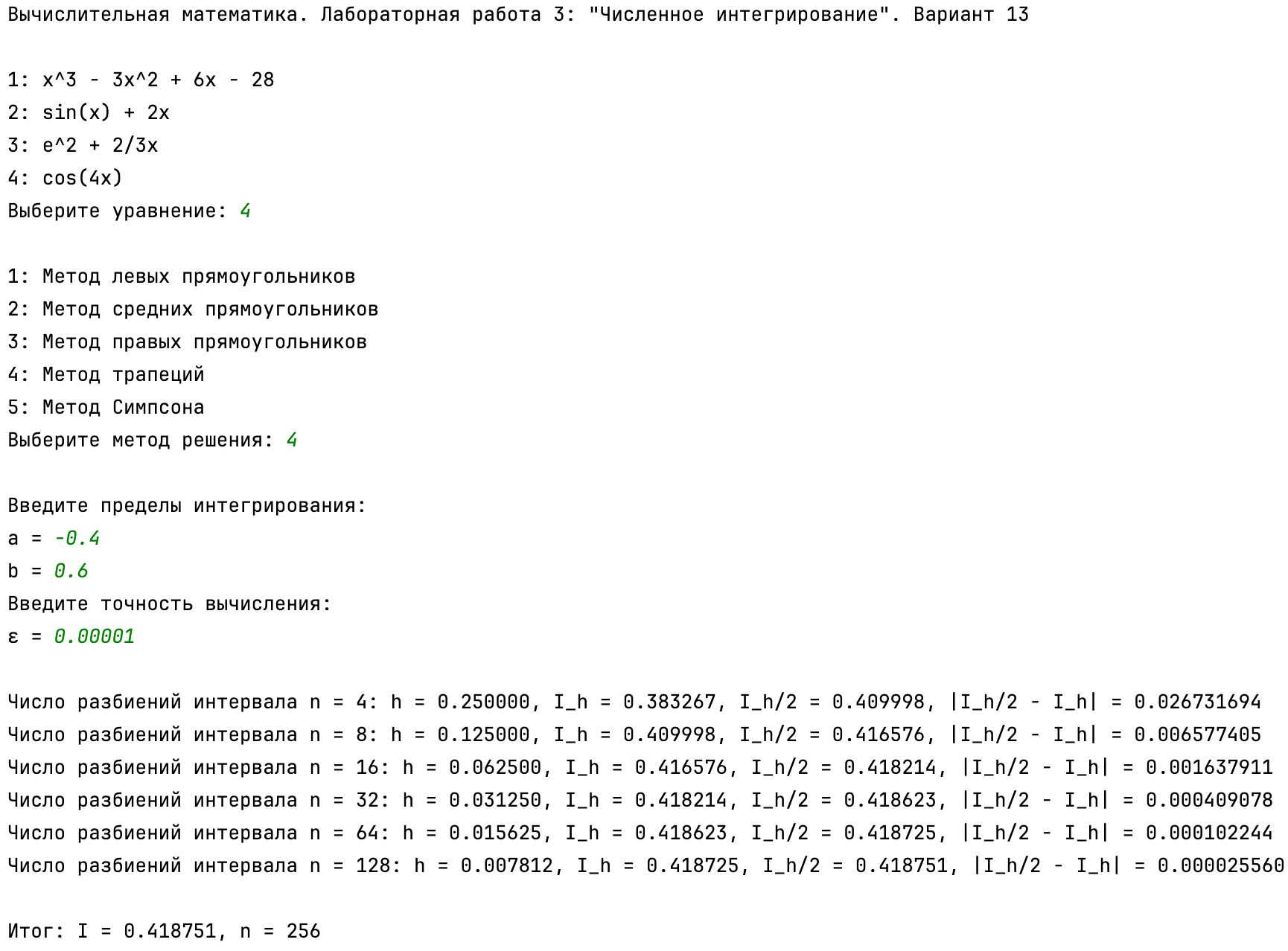
4. По формуле **Симпсона** при :

# Часть 2: Программная реализация задачи.

## Листинг программы

import math  
  
  
def left\_rectangle\_method(n):  
 h = (b - a) / n  
 I0 = h \* sum(f(a + i\*h) for i in range(0, n))  
 I1 = h/2 \* sum(f(a + i\*h/2) for i in range(0, 2\*n))  
 print(f"Число разбиений интервала n = {n}: h = {h:.6f}, I\_h = {I0:.6f}, I\_h/2 = {I1:.6f}, |I\_h/2 - I\_h| = {abs(I1 - I0):.9f}")  
  
 if abs(I1 - I0) < eps:  
 return I1, 2\*n  
 return left\_rectangle\_method(n \* 2)  
  
  
def medium\_rectangle\_method(n):  
 h = (b - a) / n  
 I0 = h \* sum((f(a + i\*h) + f(a + (i-1)\*h)) / 2 for i in range(1, n + 1))  
 I1 = h/2 \* sum((f(a + i\*h/2) + f(a + (i-1)\*h/2)) / 2 for i in range(1, 2\*n + 1))  
 print(f"Число разбиений интервала n = {n}: h = {h:.6f}, I\_h = {I0:.6f}, I\_h/2 = {I1:.6f}, |I\_h/2 - I\_h| = {abs(I1 - I0):.9f}")  
  
 if abs(I1 - I0)/3 < eps:  
 return I1, 2\*n  
 return right\_rectangle\_method(n \* 2)  
  
  
def right\_rectangle\_method(n):  
 h = (b - a) / n  
 I0 = h \* sum(f(a + i\*h) for i in range(1, n + 1))  
 I1 = h/2 \* sum(f(a + i\*h/2) for i in range(1, 2\*n + 1))  
 print(f"Число разбиений интервала n = {n}: h = {h:.6f}, I\_h = {I0:.6f}, I\_h/2 = {I1:.6f}, |I\_h/2 - I\_h| = {abs(I1 - I0):.9f}")  
  
 if abs(I1 - I0) < eps:  
 return I1, 2\*n  
 return right\_rectangle\_method(n \* 2)  
  
  
def trapezoid\_method(n):  
 h = (b - a) / n  
 I0 = h \* ((f(a) + f(b)) / 2 + sum(f(a + i\*h) for i in range(1, n)))  
 I1 = h/2 \* ((f(a) + f(b)) / 2 + sum(f(a + i\*h/2) for i in range(1, 2\*n)))  
 print(f"Число разбиений интервала n = {n}: h = {h:.6f}, I\_h = {I0:.6f}, I\_h/2 = {I1:.6f}, |I\_h/2 - I\_h| = {abs(I1 - I0):.9f}")  
  
 if abs(I1 - I0)/3 < eps:  
 return I1, 2\*n  
 return trapezoid\_method(n \* 2)  
  
  
def Simpson\_method(n):  
 h = (b - a) / n  
 I0 = h/3 \* (f(a) + 4\*sum(f(a + i\*h) for i in range(1, n, 2)) + 2\*sum(f(a + i\*h) for i in range(2, n - 1, 2)) + f(b))  
 I1 = h/6 \* (f(a) + 4\*sum(f(a + i\*h/2) for i in range(1, 2\*n, 2)) + 2\*sum(f(a + i\*h/2) for i in range(2, 2\*n - 1, 2)) + f(b))  
 print(f"Число разбиений интервала n = {n}: h = {h:.6f}, I\_h = {I0:.6f}, I\_h/2 = {I1:.6f}, |I\_h/2 - I\_h| = {abs(I1 - I0):.9f}")  
  
 if abs(I1 - I0)/15 < eps:  
 return I1, 2\*n  
 return Simpson\_method(n \* 2)  
  
  
print('Вычислительная математика. Лабораторная работа 3: "Численное интегрирование". Вариант 13\n')  
  
*# --- Исходные данные ---*n = 4  
  
  
*# --- Функции и их интегралы ---*equations = {  
 '1': lambda x: x\*\*3 - 3\*x\*\*2 + 6\*x - 28,  
 '2': lambda x: math.sin(x) + 2\*x,  
 '3': lambda x: math.e\*\*2 + 2/3\*x,  
 '4': lambda x: math.cos(4\*x)  
}  
  
integrals = {  
 '1': lambda x: x\*\*4/4 - x\*\*3 + 3\*x\*\*2 - 28\*x,  
 '2': lambda x: -math.cos(x) + x\*\*2,  
 '3': lambda x: math.e\*\*2\*x + 1/3\*x\*\*2,  
 '4': lambda x: math.sin(4\*x)/4  
}  
  
  
*# --- Ввод данных ---*print('1: x^3 - 3x^2 + 6x - 28\n2: sin(x) + 2x\n3: e^2 + 2/3x\n4: cos(4x)')  
equation\_number = input('Выберите уравнение: ')  
  
while equation\_number not in {'1', '2', '3', '4'}:  
 equation\_number = input('Выберите первое (1), второе (2), третье (3) или четвёртое (4) уравнение: ')  
  
f = equations[equation\_number]  
  
print('\n1: Метод левых прямоугольников\n2: Метод средних прямоугольников\n3: Метод правых прямоугольников')  
print('4: Метод трапеций\n5: Метод Симпсона')  
method\_number = input('Выберите метод решения: ')  
while method\_number not in {'1', '2', '3', '4', '5'}:  
 method\_number = input('Выберите метод левых прямоугольников (1), метод средних прямоугольников (2), '  
 'метод правых прямоугольников (3), '  
 'метод трапеций (4) или метод Симпсона (5): ')  
  
  
print("\nВведите пределы интегрирования:")  
a = float(input('a = '))  
b = float(input('b = '))  
while a >= b:  
 print('Правая границы должна быть больше левой. Пожалуйста, повторите ввод.')  
 b = float(input('b = '))  
print("Введите точность вычисления:")  
eps = float(input('ε = '))  
print()  
  
I = integrals[equation\_number](b) - integrals[equation\_number](a)  
  
  
*# --- Интегрирование ---*if method\_number == '1':  
 I, n = left\_rectangle\_method(n)  
elif method\_number == '2':  
 I, n = medium\_rectangle\_method(n)  
elif method\_number == '3':  
 I, n = right\_rectangle\_method(n)  
elif method\_number == '4':  
 I, n = trapezoid\_method(n)  
elif method\_number == '5':  
 I, n = Simpson\_method(n)  
  
print(f"\nИтог: I = {I:.6f}, n = {n}")

## Результат работы программы



# Выводы

В ходе работы я изучила численные методы интегрирования. Вручную использовала формулы Ньютона-Котеса, средних прямоугольников, трапеции. На Python программно реализовала метод прямоугольников (в 3-х модификациях: левых, средних и правых), метод трапеций и Симпсона.

Результаты показали, что методы эффективны при правильном выборе числа разбиений интервала интегрирования.