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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа **№3**

**«Численное интегрирование»**

по дисциплине «Вычислительная математика**»**

Вариант: **18**

**Преподаватель:**

**Выполнил:**

Шаматульский Роман Константинович

**Группа:** Р3212

Санкт-Петербург, 2025 г.

Оглавление

[Текст задания 3](#_Toc193344039)

[Вычисление интеграла 4](#_Toc193344040)

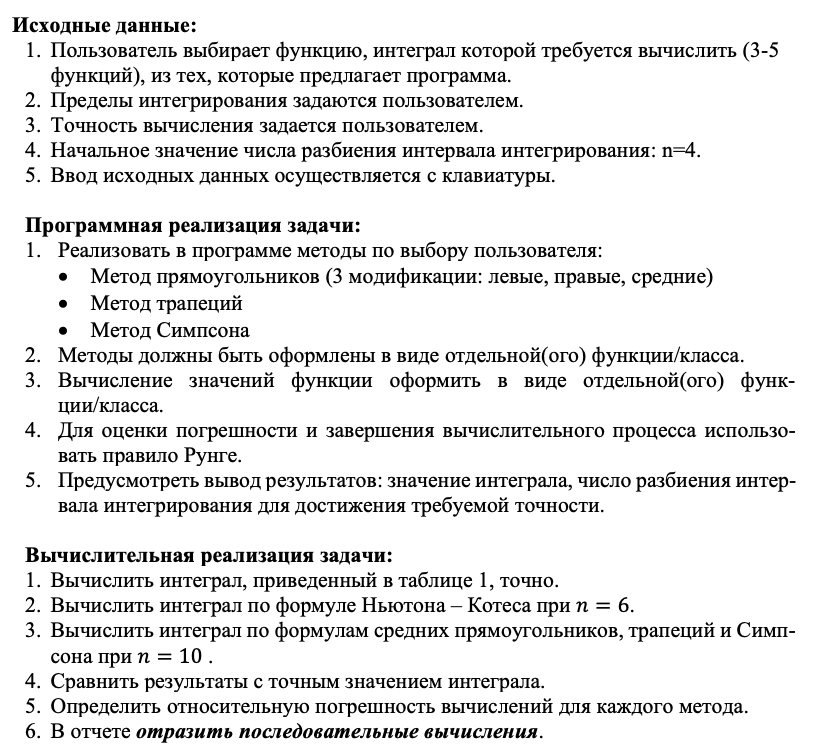
[Рабочие формулы методов 5](#_Toc193344041)

[Листинг программы 6](#_Toc193344042)

[Результат работы программы 9](#_Toc193344043)

[Вывод 10](#_Toc193344044)

# Текст задания



# Вычисление интеграла

**Точное вычисление:**

**Формула Ньютона–Котеса 6-го порядка (на 7 равноотстоящих точках) для отрезка [a,b] имеет вид:**

Шаг h: и точки для i = 0…6.

840 – общий знаменатель для всех коэффициентов.

**Метод средних прямоугольников, трапеций и метод Симпсона:**

Теперь делим отрезок [2,4] на n=10 равных частей. Шаг h = (4 – 2) / 10 = 0.2. Точки разбиения

**Метод средних прямоугольников:**

Для каждого из 10 подотрезков берём **середину**:

Складываем f(середина) × h. Сумма

**Метод трапеции:**

Подставив, получим:

**Метод Симпсона**

Подставив, получим:

**Итого:**

|  |  |  |
| --- | --- | --- |
| **Метод** | **Результат** | **Ошибка** |
| Точный | -47.3333 | 0 |
| Ньютона-Котеса | -47.3333 | 0 |
| Средних прямоугольников | -47.36 | 0.0267 |
| Трапеций | -47.536 | 0.2027 |
| Симпсона | -47.6747 | 0.3417 |

# Рабочие формулы методов

* Левых прямоугольников
* Средних прямоугольников
* Правых прямоугольников
* Трапеций
* Симпсона

# Листинг программы

import math  
import sys  
  
  
def f1(x):  
 return x\*\*2  
  
def f2(x):  
 return math.sin(x)  
  
def f3(x):  
 return math.exp(x)  
  
def f4(x):  
 return 1.0 / x  
  
def f5(x):  
 return x\*\*3 + 2\*x  
  
functions = [f1, f2, f3, f4, f5]  
  
def rectangle\_method(f, a, b, n, mode="left"):  
 h = (b - a) / n  
 total = 0.0  
  
 if mode == "left":  
 for i in range(n):  
 x = a + i \* h  
 total += f(x)  
  
 elif mode == "right":  
 for i in range(1, n+1):  
 x = a + i \* h  
 total += f(x)  
  
 elif mode == "middle":  
 for i in range(n):  
 x = a + (i + 0.5) \* h  
 total += f(x)  
 else:  
 raise ValueError("Неизвестный режим прямоугольников.")  
  
 return total \* h  
  
def trapezoid\_method(f, a, b, n):  
 h = (b - a) / n  
 total = 0.5 \* (f(a) + f(b))  
 for i in range(1, n):  
 x = a + i \* h  
 total += f(x)  
 return total \* h  
  
def simpson\_method(f, a, b, n):  
 if n % 2 != 0:  
 n += 1  
  
 h = (b - a) / n  
 total = f(a) + f(b)  
 odd\_sum = 0.0  
 even\_sum = 0.0  
 for k in range(1, n):  
 x = a + k \* h  
 if k % 2 == 0:  
 even\_sum += f(x)  
 else:  
 odd\_sum += f(x)  
 total += 4 \* odd\_sum + 2 \* even\_sum  
 return total \* h / 3.0  
  
def integrate\_with\_precision(f, a, b, eps, method\_name="rectangle", mode="left", n\_init=4):  
 if method\_name == "rectangle":  
 method\_func = lambda f,a,b,n: rectangle\_method(f,a,b,n,mode=mode)  
 p = 2 if mode == "middle" else 1  
 if p == 2:  
 runge\_factor = 1.0/3.0  
 else:  
 runge\_factor = 1.0 # Если p=1  
  
 elif method\_name == "trapezoid":  
 method\_func = trapezoid\_method  
 runge\_factor = 1.0/3.0  
  
 elif method\_name == "simpson":  
 method\_func = simpson\_method  
 runge\_factor = 1.0/15.0  
  
 else:  
 raise ValueError("Неизвестный метод: " + method\_name)  
  
 n = n\_init  
 I\_old = method\_func(f, a, b, n)  
 while True:  
 n \*= 2  
 I\_new = method\_func(f, a, b, n)  
 # оценка погрешности по правилу Рунге:  
 error\_est = abs(I\_new - I\_old) \* runge\_factor  
  
 if error\_est < eps:  
 return I\_new, n  
 I\_old = I\_new  
  
def main():  
 print("Выберите функцию для интегрирования:")  
 for idx, func in enumerate(functions, start=1):  
 print(f"{idx}) {func.\_\_doc\_\_}")  
  
 choice = int(input("Введите номер функции (1..5): "))  
 if choice < 1 or choice > len(functions):  
 print("Некорректный выбор функции. Завершение.")  
 return  
 f = functions[choice - 1]  
  
 a = float(input("Введите левую границу интегрирования a = "))  
 b = float(input("Введите правую границу интегрирования b = "))  
 if a == b:  
 print("Границы интегрирования совпадают, интеграл будет 0.")  
 return  
  
 eps = float(input("Введите требуемую точность (например, 1e-6): "))  
  
 n\_init = int(input("Введите начальное число разбиений (n >= 2), по умолчанию 4: ") or 4)  
 if n\_init < 2:  
 n\_init = 4  
  
 print("\nВыберите метод интегрирования:")  
 print("1) Прямоугольники (левые)")  
 print("2) Прямоугольники (правые)")  
 print("3) Прямоугольники (средние)")  
 print("4) Трапеции")  
 print("5) Симпсон")  
  
 method\_choice = int(input("Номер метода: "))  
  
 if method\_choice == 1:  
 method\_name = "rectangle"  
 mode = "left"  
 elif method\_choice == 2:  
 method\_name = "rectangle"  
 mode = "right"  
 elif method\_choice == 3:  
 method\_name = "rectangle"  
 mode = "middle"  
 elif method\_choice == 4:  
 method\_name = "trapezoid"  
 mode = ""  
 elif method\_choice == 5:  
 method\_name = "simpson"  
 mode = ""  
 else:  
 print("Некорректный выбор метода. Завершение.", file=sys.stderr)  
 return  
  
 I, N = integrate\_with\_precision(f, min(a, b), max(a, b), eps, method\_name, mode, n\_init)  
  
 # вывод рез-тов  
 print(f"\nРезультаты:")  
 print(f"Метод: {method\_name}, вариант = {mode if mode else '—'}")  
 print(f"Число разбиений, достигнутое для заданной точности: {N}")  
 print(f"Приблизительное значение интеграла: {I:.8f}")  
 print(f"Точность (заданная): {eps}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# Результат работы программы

Выберите функцию для интегрирования:

1) f(x) = x^2

2) f(x) = sin(x)

3) f(x) = e^x

4) f(x) = 1/x (при x ≠ 0)

5) f(x) = x^3 + 2x

Введите номер функции (1..5): 1

Введите левую границу интегрирования a = 50

Введите правую границу интегрирования b = 800

Введите требуемую точность (например, 1e-6): 0.001

Введите начальное число разбиений (n >= 2), по умолчанию 4: 25

Выберите метод интегрирования:

1) Прямоугольники (левые)

2) Прямоугольники (правые)

3) Прямоугольники (средние)

4) Трапеции

5) Симпсон

Номер метода: 5

Результаты:

Метод: simpson, вариант = —

Число разбиений, достигнутое для заданной точности: 50

Приблизительное значение интеграла: 170625000.00000000

Точность (заданная): 0.001

# Вывод

В ходе данной лабораторной работы я познакомился с такими методами вычисления интегралов как: метод прямоугольников (левых, правых и средних), метод трапеций, метод Симпсона. Кроме того, реализовал решение интегралов с помощью этих методов на python. Полученные мной знания помогут мне в дальнейшем изучении математики.