Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования «Санкт-Петербургский
национальный исследовательский университет информационных технологий,
механики и оптики»

Факультет программной инженерии и компьютерной техники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ 3
ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ
ВАРИАНТ 12

Студент: Пышкин Никита Сергеевич, P3213

Преподаватель:

Санкт Петербург 2025

Содержание

Цель лабораторной работы

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Порядок выполнения лабораторной работы

Исходные данные:

1. Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.

2. Пределы интегрирования задаются пользователем.

3. Точность вычисления задается пользователем.

4. Начальное значение числа разбиения интервала интегрирования: n=4.

5. Ввод исходных данных осуществляется с клавиатуры.

Программная реализация задачи:

1. Реализовать в программе методы по выбору пользователя:

• Метод прямоугольников (3 модификации: левые, правые, средние)

• Метод трапеций

• Метод Симпсона

2. Методы должны быть оформлены в виде отдельной(ого) функции/класса.

3. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.

4. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.

5. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

Вычислительная реализация задачи:

1. Вычислить интеграл, приведенный в таблице 1, точно.

2. Вычислить интеграл по формуле Ньютона – Котеса при $n = 6$.

3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при $n = 10$.

4. Сравнить результаты с точным значением интеграла.

5. Определить относительную погрешность вычислений для каждого метода.

6. В отчете отразить последовательные вычисления

Рабочие формулы используемых методов

Ньютона-Котеса:

Рабочая формула: $\int_a^b f(x)dx \approx \int_a^b L_n(x)dx = \sum_{i=0}^{n} f(x_i)c_n^i$

Метод прямоугольников:

Рабочая формула: $\int_a^b f(x)dx \approx S_n = \sum_{i=0}^{n} f(\xi_i)\Delta x_i$

Вычисление дельты: $h_i = h = \frac{b-a}{n} = const$

Рабочая формула для средних: $\int_a^b f(x)dx \approx \sum_{i=0}^{n} h_i f\left(x_{i-\frac{1}{2}}\right)$, где $x_{i-\frac{1}{2}} = \frac{x_{i-1}+x_i}{2} = x_{i-1} + \frac{h_i}{2}$

Метод трапеций:

Рабочая формула: $\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1} + y_i)$

Рабочая формула при h = const: $\frac{h}{2}(y_0 + y_n + 2\sum_{i=1}^{n-1} y_i)$

Метод Симпсона:

Рабочая формула: $\int_a^b f(x) = \frac{h}{3}[y_0 + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2}) + y_n]$

Вычисление заданного интеграла

1) Точное вычисление

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = \left(\frac{x^4}{4} + \frac{2x^3}{3} - \frac{3x^2}{2} - 12x\right)\Big|_1^2 = -\frac{97}{12} = -8.08(3)$$

2) По формуле Ньютона-Котеса при n=6

Пусть $d = \frac{b-a}{6}$

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = f(a)c_6^0 + f(a+d)c_6^1 + f(a+2d)c_6^2 + f(a+3d)c_6^3 + f(a+4d)c_6^4 + f(a+5d)c_6^5 + f(b)c_6^6$$

Подставим коэффициенты из таблицы:

$$f(a)\frac{41(b-a)}{840} + f\left(a + \frac{b-a}{6}\right)\frac{216(b-a)}{840} + f\left(a + \frac{b-a}{3}\right)\frac{27(b-a)}{840} + f\left(a + \frac{b-a}{2}\right)\frac{272(b-a)}{840} + f\left(a + \frac{2(b-a)}{3}\right)\frac{27(b-a)}{840} + f\left(a + \frac{5(b-a)}{6}\right)\frac{216(b-a)}{840} + f(b)\frac{41(b-a)}{840}$$

Подставим наши значения a и b:

$$f(1)\frac{41(2-1)}{840} + f\left(1 + \frac{2-1}{6}\right)\frac{216(2-1)}{840} + f\left(1 + \frac{2-1}{3}\right)\frac{27(2-1)}{840} + f\left(1 + \frac{2-1}{2}\right)\frac{272(2-1)}{840} + f\left(1 + \frac{2(2-1)}{3}\right)\frac{27(2-1)}{840} + f\left(1 + \frac{5(2-1)}{6}\right)\frac{216(2-1)}{840} + f(2)\frac{41(2-1)}{840}$$

Сократим:

$$f(1)\frac{41}{840} + f\left(\frac{7}{6}\right)\frac{216}{840} + f\left(\frac{4}{3}\right)\frac{27}{840} + f\left(\frac{3}{2}\right)\frac{272}{840} + f\left(\frac{5}{3}\right)\frac{27}{840} + f\left(\frac{11}{6}\right)\frac{216}{840} + f(2)\frac{41}{840}$$

Посчитаем:

$$-12*\frac{41}{840}+(-11.1898148)\frac{216}{840}+(-10.074)\frac{27}{840}+(-8.625)\frac{272}{840}+(-6.814)\frac{27}{840}+(-4.615740)\frac{216}{840}+(-2)\frac{41}{840}=-8.08(3)$$

3) По формуле среднего прямоугольника при n = 10

$$h_i=h=\frac{b-a}{n}=\frac{2-1}{10}=0.1$$

Составим табличку значений:

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
y_i	-12	11.549	-10.992	-10.323	-9.536	-8.625	-7.584	-6.407	-5.088	-3.621	-2
$x_{i-1/2}$		1.05	1.15	1.25	1.35	1.45	1.55	1.65	1.75	1.85	1.95
$y_{i-1/2}$		-11.787375	-11.284125	-10.671875	-9.944625	-9.096375	-8.121125	-7.012875	-5.765625	-4.373375	-2.830125

Вычислим результат:

$$\sum_{i=0}^{n}h_i f\left(x_{i-\frac{1}{2}}\right)=0.1\sum_{i=0}^{n}f\left(x_{i-\frac{1}{2}}\right)=0.1\sum_{i=0}^{n}y_{i-\frac{1}{2}}=-8.08875$$

4) По формуле трапеций при n = 10

$$h_i=h=\frac{b-a}{n}=\frac{2-1}{10}=0.1$$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
y_i	-12	-11.549	-10.992	-10.323	-9.536	-8.625	-7.584	-6.407	-5.088	-3.621	-2

$$\frac{h}{2}\left(y_0+y_n+2\sum_{i=1}^{n-1}y_i\right)=\frac{0.1}{2}\big(-12-2+2*(-73.725)\big)=-8.0725$$

5) По формуле Симпсона при n = 10

$$h=\frac{b-a}{n}=\frac{2-1}{10}=0.1$$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
y_i	-12	-11.549	-10.992	-10.323	-9.536	-8.625	-7.584	-6.407	-5.088	-3.621	-2

$$\int_a^b f(x) = \frac{0.1}{3}\left[-12 + 4\big(-11.549 + (-10.323) + (-8.625) + (-6.407) + (-3.621)\big) + 2\big(-10.992 + (-9.536) + (-7.584) + (-5.088)\big) - 2\right] = -8.08(3)$$

Сравнение результатов

Результат точных вычисление: $\mathbf{-8.08(3)}$

Метод Ньютона-Котеса: $\mathbf{-8.08(3)}$ – не отличается от точных вычислений

Метод средних прямоугольников: $\mathbf{-8.08875}$ – отличается от точных вычислений на ≈ -0.005416

Метод трапеций: $\mathbf{-8.0725}$ – отличается от точных вычислений на ≈ 0.01083

Метод Симпсона: $\mathbf{-8.08(3)}$ – не отличается от точных вычислений

Расчет относительной погрешности

Метод Ньютона-Котеса: погрешности нет

Метод средних прямоугольников: $\Delta = \frac{|-8.08(3)-(-8.08875)|}{|-8.08(3)|} \approx 0.067\%$

Метод трапеций: $\Delta = \frac{|-8.08(3)-(-8.0725)|}{|-8.08(3)|} \approx 0.13\%$

Метод Симпсона: погрешности нет

Листинг программы

abstract_solve_integral_method.py:

```
from typing import Tuple, Callable

from abc import ABC, abstractmethod


class AbstractSolveIntegralMethod(ABC):

    @classmethod

    @abstractmethod

    def solve(cls, integral_func: Callable[[float], float], a: float, b: float, e: float, max_iter_count: int = 10**6) -> Tuple[int, int]:

        pass


    @classmethod

    def _check_calc_is_end(cls, first_value: float, second_value: float, k: int, e: float) -> bool:
```

```
        return abs(first_value - second_value) / (2**k + 1) < e
```

rectangle_method.py:

```
from enum import IntEnum
from typing import Tuple, Callable


from .core import AbstractSolveIntegralMethod


class RectangleMethodType(IntEnum):
    LEFT = 0
    MIDDLE = 1
    RIGHT = 2


class RectangleMethod(AbstractSolveIntegralMethod):
    @classmethod
    def solve(cls, integral_func: Callable[[float], float], a: float, b: float, e: float, max_iter_count: int = 10**6, m_type: RectangleMethodType = RectangleMethodType.MIDDLE) -> Tuple[int, int]:
        results, n = [], 4
        while len(results) < 2 or not cls._check_calc_is_end(results[-2], results[-1], 2, e):
            if n >= max_iter_count:
                raise ValueError(f"Не удалось найти решение с удовлетворяющей точностью за {max_iter_count} разбиений")

            h = (b - a) / n
            match m_type:
                case RectangleMethodType.LEFT:
                    start = a
                case RectangleMethodType.MIDDLE:
                    start = a + h / 2
                case RectangleMethodType.RIGHT:
                    start = a + h

            s = 0
```

```
            for i in range(n):
                s += integral_func(start + h * i)

            results.append(s * h)
            n += 1

        return results[-1], n
```

simpsons_method.py:

```
from typing import Tuple, Callable

from .core import AbstractSolveIntegralMethod

class SimpsonsMethod(AbstractSolveIntegralMethod):
    @classmethod
    def solve(cls, integral_func: Callable[[float], float], a: float, 
b: float, e: float, max_iter_count: int = 10**6) -> Tuple[int, int]:
        results, n = [], 4
        while len(results) < 2 or not cls._check_calc_is_end(results[-
2], results[-1], 4, e):
            if n >= max_iter_count:
                raise ValueError(f"Не удалось найти решение с 
удовлетворяющей точностью за {max_iter_count} разбиений")

            h = (b - a) / n

            s = 0
            for i in range(n):
                result = integral_func(a + h * i)
                if i % 2 == 0 and i != 0 and i != n - 1:
                    result *= 4
                elif i % 2 != 0 and i != 0 and i != n - 1:
                    result *= 2

                s += result
```

```
            results.append(s * (h / 3))
            n += 2

        return results[-1], n
```

trapezoid_method.py:

```
from typing import Tuple, Callable


from .core import AbstractSolveIntegralMethod


class TrapezoidMethod(AbstractSolveIntegralMethod):
    @classmethod
    def solve(cls, integral_func: Callable[[float], float], a: float,
b: float, e: float, max_iter_count: int = 10**6) -> Tuple[int, int]:
        results, n = [], 4
        while len(results) < 2 or not cls._check_calc_is_end(results[-
2], results[-1], 2, e):
            if n >= max_iter_count:
                raise ValueError(f"Не удалось найти решение с
удовлетворяющей точностью за {max_iter_count} разбиений")

            h = (b - a) / n

            s = 0
            for i in range(n):
                result = integral_func(a + h * i)
                if i != 0 and i != n - 1:
                    result *= 2

                s += result

            results.append(s * (h / 2))
            n += 1

        return results[-1], n
```

Результаты выполнения программы при различных исходных данных

Пример 1:

```
Выберите функцию:
1. x**2 - 0.5
2. x**3 - 4*x + 1
3. 2**x - 4
4. x**3 + 2*x**2 - 3*x - 12
Введите номер функции: 4
Выберите метод:
1. Метод прямоугольников
2. Метод трапеций
3. Метод Симпсонов
Введите номер метода: 3
Введите пределы интегрирования a b: 1 2
Введите точность вычисления: 0.0001
Результат: -8.014143886878387 (найден за 90 разбиений)
```

Пример 2:

```
Выберите функцию:
1. x**2 - 0.5
2. x**3 - 4*x + 1
3. 2**x - 4
4. x**3 + 2*x**2 - 3*x - 12
Введите номер функции: 2
Выберите метод:
1. Метод прямоугольников
2. Метод трапеций
3. Метод Симпсонов
Введите номер метода: 1
Выберите тип метода:
1. Метод левых прямоугольников
2. Метод средних прямоугольников
3. Метод правых прямоугольников
```

```
Введите номер метода: 3
Введите пределы интегрирования a b: 0.00001
Некорректный ввод, проверьте соответствие формату.
Введите пределы интегрирования a b: 1 5
Введите точность вычисления: 0.00001
Результат: 112.1038683431954 (найден за 2081 разбиений)
```

Заключение

В ходе лабораторной работы я изучил разные численные методы, их погрешности и реализовал их на языке Python.