Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 2

по курсу «Численные методы»

Студент группы ИУ9-62Б Нащёкин Н.Д.

Преподаватель: Домрачева А.Б.

Москва, 2025

1 Задача

- Найти $\int_a^b f(x)\,dx$ по формуле трапеций и по формуле Симпсона с погрешностью $\varepsilon = 0.001$. Сравнить требуемое число разбиений для обоих методов.

2 Основная теория

Метод средних прямоугольников

Пусть функция $f(x)$ непрерывна на отрезке $[a, b]$. Разобьём отрезок на n равных отрезков точками $a = x_0, x_1, ..., x_n = b$. Тогда фигура под графиком функции $f(x)$ разобьётся на криволинейные трапеции на полученных отрезках. Пусть $h = x_i - x_{i-1} = \frac{b-a}{n}, i = 1, ..., n$ – постоянный шаг разбиения.

Для начала рассмотрим так называемые методы левых и правых прямоугольников. На каждом частичном отрезке возьмём левую границу и посчитаем площадь прямоугольника со сторонами $f(x_{i-1})$ и $(x_i - x_{i-1}) = h$. Тогда $I_{li} = hf(x_{i-1})$ – площади левых прямоугольников, $i = 1, ..., n$. В таком случае $I_l = h\sum_{i=1}^{n} f(x_{i-1})$ – приближённое методом левых прямоугольников значение искомого интеграла.

Аналогично, взяв вместо левых границ частичных отрезков правые, получим $I_{ri} = hf(x_i)$ – площади правых прямоугольников, $I_r = h\sum_{i=1}^{n} f(x_i)$ – приближённое значение интеграла.

Недостаток этих методов в том, что они имеют первый порядок точности. Существует метод средних (центральных) прямоугольников, имеющий второй порядок точности. Для его реализации вместо левой или правой границы частичного отрезка берётся средняя точка: $I_{si} = hf(x_i - \frac{h}{2})$. В таком случае приближённое значение интеграла: $I_s = h\sum_{i=1}^{n} f(x_i - \frac{h}{2})$.

Выведем погрешность метода средних прямоугольников. Приближённая формула для площади одного сегмента:

$$\int_{x_{i-1}}^{x_i} f(x)\,dx \approx f\left(x_{i-1} + \frac{h}{2}\right)h,$$

где $h = x_i - x_{i-1}$.

Погрешность на одном сегменте:

$$\delta_i = \int_{x_{i-1}}^{x_i} f(x)\,dx - f\left(x_{i-1} + \frac{h}{2}\right)h.$$

Разложим функцию $f(x)$ в ряд Тейлора в точке $x_{i-1} + \frac{h}{2}$:

$$f(x) = f\left(x_{i-1} + \frac{h}{2}\right) + f'\left(x_{i-1} + \frac{h}{2}\right)\left(x - \left(x_{i-1} + \frac{h}{2}\right)\right) + \frac{f''(\varepsilon_i)}{2}\left(x - \left(x_{i-1} + \frac{h}{2}\right)\right)^2,$$

где $\varepsilon_i \in [x_{i-1}, x_i]$.

Подставляя в выражение для δ_i, получаем:

$$\delta_i = \int_{x_{i-1}}^{x_i} \left(f(x) - f\left(x_{i-1} + \frac{h}{2}\right)\right) dx = \frac{f''(\varepsilon_i)h^3}{24},$$

Абсолютная погрешность на одном отрезке:

$$|\delta_i| \leq \max_{x \in [x_{i-1}, x_i]} |f''(x)| \frac{h^3}{24}.$$

Для всей фигуры (всего n равных сегментов, $h = \frac{b-a}{n}$):

$$|\delta| \leq \max_{x \in [a,b]} |f''(x)| \frac{(b-a)}{24} h^2 = O(h^2).$$

Метод трапеций

Теперь в предыдущих обозначениях рассмотрим похожий метод приближённого нахождения определённого интеграла. Как и ранее, фигура под графиком функции $f(x)$ разбита на множество криволинейных трапеций. Тогда площадь каждой криволинейной трапеции примерно равна

$$\frac{f(x_{i-1}) + f(x_i)}{2} h.$$

Суммируя полученные площади, получим примерное значение интеграла $I = \int_a^b f(x)\, dx$:

$$I \approx \frac{1}{2} h(f(x_0) + f(x_1) + f(x_1) + f(x_2) + f(x_2) + \ldots + f(x_{n-1}) + f(x_{n-1}) + f(x_n)).$$

Заметим, что в полученной формуле все $f(x_i)$, кроме $f(x_0) = f(a)$ и $f(x_n) = f(b)$, встречаются дважды. Учитывая это, получим:

$$I \approx h(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)).$$

Метод трапеций имеет второй порядок точности, как и метод средних прямоугольников. Аналогично нахождению погрешности для метода средних прямоугольников, можно найти погрешность для метода трапеций:

$$|\delta| \leq \max_{x\in[a,b]} |f''(x)| \frac{(b-a)}{12} h^2 = O(h^2).$$

Как видно, для метода трапеций оценка погрешности в два раза хуже.

Метод Симпсона

В предыдущих обозначениях также верна приближённая формула:

$$I \approx \frac{h}{3}(f(x_0)+f(x_n)+4(f(x_1)+f(x_3)+...+f(x_{n-1})+2(f(x_2)+f(x_4)+...+f(x_{n-2})))).$$

Здесь число отрезков n должно быть чётным.

Получим оценку погрешности для метода Симпсона:

$$\int_{x_{i-1}}^{x_{i+1}} f(x)\,dx \approx \frac{h}{3}\left(f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\right),$$

где $h = \frac{x_{i+1} - x_{i-1}}{2}$ — половина ширины параболического интервала.

Для функции $f \in C^4[a, b]$ погрешность на таком двойном отрезке можно выразить как:

$$\delta_i = -\frac{h^5}{90} f^{(4)}(\xi_i), \quad \xi_i \in [x_{i-1}, x_{i+1}].$$

Если отрезок $[a, b]$ разбит на n равных частей (где n чётное), и $h = \frac{b-a}{n}$, то суммарная погрешность метода Симпсона на всём отрезке оценивается как:

$$\left|\int_a^b f(x)\,dx - I\right| \le \max_{x\in[a,b]} \left|f^{(4)}(x)\right| \frac{(b-a)^5}{180n^4},$$

где I — приближённое значение интеграла, полученное по формуле Симпсона. Учитывая, что $h = \frac{b-a}{n}$, получим

$$|\delta| \le \max_{x\in[a,b]} \left|f^{(4)}(x)\right| \frac{(b-a)}{180} h^4 = O(h^4).$$

Правило Рунге

Для оценки погрешности на практике часто применяют правило Рунге: необходимо найти приближённое значение S с шагом h и с шагом $2h$, а затем посчитать $\Delta S(h) = \frac{S(h)-S(2h)}{2^p-1}$, где $p = 2$ для метода трапеций и метода прямоугольников и $p = 4$ для метода Симпсона. Теперь, когда при вычислениях с последовательным увеличением n достигается неравенство $|\Delta S(h)| \le \varepsilon$, где ε – некоторая заданная абсолютная погрешность, вычисления прекращаются.

3 Практическая реализация

Листинг 1 — реализация программы

```
import math
# Интеграл = 8.0366775 - Вычислено калькулятором


def f(x: float) -> float:
    return math.log(2 * x)


eps = 0.001


def trap_method(a: float, b: float, n: int) -> float:
    h = (b - a) / n
    summ = 0
    for i in range(1, n, 1):
        summ += f(a + i * h)
    summ += (f(a) + f(b)) / 2
    return summ * h


def simpson_method(a: float, b: float, n: int) -> float:
    h = (b - a) / n
    sum1 = 0
    sum2 = 0
    for i in range(1, n, 2):
        sum1 += f(a + i * h)

    for i in range(2, n, 2):
        sum2 += f(a + i * h)

    summ = 4*sum1 + 2*sum2 + f(a) + f(b)
    return summ * h / 3


def average_rectangle(a: float, b: float, n: int) -> float:
    h = (b - a) / n
    summ = 0
    for i in range(0, n, 1):
        summ += f(a + i * h + h / 2)
    return summ * h


def runge_rule(S_h, S_2h, p):
    return (S_h - S_2h) / (2 ** p - 1)
```

```python
def main():
    a = 0.5
    b = 2 * math.e

    n_t = 2
    S_2h = trap_method(a, b, n_t)
    S_h = trap_method(a, b, 2 * n_t)
    S_delta = runge_rule(S_h, S_2h, p=2)
    while abs(S_delta) > eps:
        n_t *= 2
        S_2h = trap_method(a, b, n_t)
        S_h = trap_method(a, b, 2 * n_t)
        S_delta = runge_rule(S_h, S_2h, p=2)

    n_s = 2
    S_2h_s = simpson_method(a, b, n_s)
    S_h_s = simpson_method(a, b, 2 * n_s)
    S_delta_s = runge_rule(S_h_s, S_2h_s, p=4)
    while abs(S_delta_s) > eps:
        n_s *= 2
        S_2h_s = simpson_method(a, b, n_s)
        S_h_s = simpson_method(a, b, 2 * n_s)
        S_delta_s = runge_rule(S_h_s, S_2h_s, p=4)

    n_s_t = 2
    S_2h_s_t = average_rectangle(a, b, n_s_t)
    S_h_s_t = average_rectangle(a, b, 2 * n_s_t)
    S_delta_s_t = runge_rule(S_h_s_t, S_2h_s_t, p=2)
    while abs(S_delta_s_t) > eps:
        n_s_t *= 2
        S_2h_s_t = average_rectangle(a, b, n_s_t)
        S_h_s_t = average_rectangle(a, b, 2 * n_s_t)
        S_delta_s_t = runge_rule(S_h_s_t, S_2h_s_t, p=2)

    print(f'      Метод трапеций     Метод Симпсона   Метод сред. прямоугольников')
    print(f'n          {n_t}                {n_s}                    {n_s_t}')
    print(f'I*         {S_h:.6f}           {S_h_s:.6f}           {S_h_s_t:.6f}')
    print(f'R          {S_delta:.6f}           {S_delta_s:.6f}           {S_delta_s_t:.6f}')
    print(f'I* + R     {S_h + S_delta:.6f}           {S_h_s + S_delta_s:.6f}           {S_h_s_t + S_delta_s_t:.6f}')


if __name__ == "__main__":
    main()
```

Мой вариант – 22. По условию необходимо найти приближённое значение интеграла $\int_{0.5}^{2e} ln(2x)\, dx$. Все три метода вычислений реализованы в виде отдельных функций. Также для удобства была выделена функция для вычисления $\Delta S(h)$ по правилу Рунге. На рисунке 1 представлен результат работы программы. Здесь n – число разбиений, $I*$ – вычисленное приближённое значение интеграла, R – последняя $\Delta S(h)$ (уточнение по Ричардсону).

	Метод трапеций	Метод Симпсона	Метод сред. прямоугольников
n	32	8	32
I*	8.035778	8.036107	8.037127
R	0.000897	0.000336	-0.000447
I* + R	8.036674	8.036443	8.036680

Рис. 1 — Результат выполнения программы

4 Вывод

В данной лабораторной работе были реализованы метод трапеций, метод средних прямоугольников и метод Симпсона для приближённого вычисления значения определённого интеграла. Из выведенных программой результатов видно, что метод Симпсона позволяет найти значение интеграла с заданной погрешностью с гораздо меньшим числом разбиений: для метода трапеций и средних прямоугольников это значение равно 32, а для метода Симпсона – 8. Кроме того, оба метода корректно вычисляют приближённое значение заданного интеграла: значение, полученное на калькуляторе, равно 8.0366775.