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»

«Приближенное вычисление определенного интеграла»

Студент группы ИУ9-62Б Волохов А. В.

Преподаватель Домрачева А. Б.

Москва 2025

1 Задача

Необходимо:

1) Реализовать численное интегрирование функции

$$f(x) = \frac{1}{4}x \cdot e^{\frac{x^2}{2}}$$

по трем методам: трапеций, центральных прямоугольников и Симпсона.

2) Осуществить оценку погрешности каждого метода с использованием правила Рунге.

3) Провести сравнение численного результата с аналитическим значением интеграла.

4) Вывести количество разбиений, необходимое для достижения заданной точности.

2 Основная теория

Для приближённого вычисления определённого интеграла

$$I = \int_a^b f(x)\,dx$$

применяются различные квадратурные формулы.

Метод трапеций. Разобьём отрезок $[a, b]$ на n равных частей шириной $h = \frac{b-a}{n}$. Значения функции $f(x)$ в узлах обозначим как $f(x_0), f(x_1), \ldots, f(x_n)$. Тогда формула трапеций имеет вид:

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

Погрешность метода оценивается как:

$$R \approx -\frac{(b-a)}{12}h^2 f''(\xi), \quad \xi \in [a, b]$$

Метод прямоугольников (средних). Пусть $x_i = a + ih$, где $h = \frac{b-a}{n}$. Центр каждого подотрезка — $x_{i-0.5}$. Тогда:

$$I \approx h \sum_{i=1}^{n} f(x_{i-0.5})$$

Погрешность метода:

$$R \approx -\frac{(b-a)}{24} h^2 f''(\xi)$$

Метод Симпсона. Применим разбиение n, кратное 2. Тогда формула имеет вид:

$$I \approx \frac{h}{3} \left(f(a) + f(b) + 4 \sum_{i=1,\text{ нечет}}^{n-1} f(x_i) + 2 \sum_{i=2,\text{ чет}}^{n-2} f(x_i) \right)$$

Погрешность метода:

$$R \leq \frac{(b-a)}{2880} h^4 \max_{x \in [a,b]} |f^{(4)}(x)|$$

Правило Рунге. Если I_h — приближенное значение интеграла при шаге h, а $I_{h/2}$ — при шаге $h/2$, и метод имеет порядок точности p, то:

$$R = \frac{I_{h/2} - I_h}{2^p - 1}$$

$$I \approx I_{h/2} + R$$

Это позволяет адаптивно уточнять значение интеграла до достижения заданной точности ε.

3 Практическая реализация

Код представлен в Листинге 1.

Листинг 1 - lab2.py

```
import math


def f(x):
    return 0.25 * x * math.exp((x ** 2) / 2)


def trapezoidal_rule(f, a, b, n):
    h = (b - a) / n
    s = 0.5 * (f(a) + f(b))
    for i in range(1, n):
        s += f(a + i * h)
    return h * s


def simpson_rule(f, a, b, n):
    if n % 2 != 0:
        raise ValueError("Error")
    h = (b - a) / n
    s = f(a) + f(b)
    s_odd = 0.0
    s_even = 0.0

    for k in range(1, n):
        xk = a + k * h
        if k % 2 == 1:
            s_odd += f(xk)
        else:
            s_even += f(xk)

    return (h / 3) * (s + 4 * s_odd + 2 * s_even)


def midpoint_rule(f, a, b, n):
    h = (b - a) / n
    s = 0.0
    for i in range(n):
        mid = a + (i + 0.5) * h
        s += f(mid)
    return h * s
```

```
def runge_integration(method, f, a, b, eps, p, start_n=2):
    n = start_n
    I_n = method(f, a, b, n)

    while True:
        I_2n = method(f, a, b, 2 * n)
        R = (I_2n - I_n) / (2 ** p - 1)

        if abs(R) < eps:
            I_extrap = I_2n + R
            return I_2n, R, I_extrap, 2 * n

        n *= 2
        I_n = I_2n


if __name__ == "__main__":
    a = 0
    b = 2
    eps = 0.001

    integral = (math.exp(2) - 1) / 4

    integral_star_x2_trap, R_trap, I_extrap_trap, n_trap = runge_integration(
        trapezoidal_rule, f, a, b, eps, p=2)
    integral_star_x2_simpson, R_simpson, I_extrap_simpson, n_simpson =
        runge_integration(simpson_rule, f, a, b, eps, p=4)
    integral_star_x2_midpoint, R_midpoint, I_extrap_midpoint, n_midpoint =
        runge_integration(midpoint_rule, f, a, b, eps, p=2)

    print("Eps: ", eps)
    print("I: ", integral)
    print("{:<13}{:<25}{:<25}{}".format("", "Trapezoid:", "Simpson:", "
        Central rectangle:"))
    print("{:<13}{:<25}{:<25}{}".format("n:", str(n_trap), str(n_simpson),
        str(n_midpoint)))
    print("{:<13}{:<25}{:<25}{}".format("I^*_h/2:", str(integral_star_x2_trap
        ), str(integral_star_x2_simpson), str(integral_star_x2_midpoint)))
    print("{:<13}{:<25}{:<25}{}".format("R:", str(R_trap), str(R_simpson),
        str(R_midpoint)))
    print("{:<13}{:<25}{:<25}{}".format("I^*_h/2 + R:", str(I_extrap_trap),
        str(I_extrap_simpson), str(I_extrap_midpoint)))
```

В результате работы программы получился следующий вывод:

```
Eps:  0.001
I:  1.5972640247326626
             Метод трапеций:          Метод Симпсона:          Метод центр.прямоуг.:
n:           64                       16                       64
I^*_h/2:     1.5979952291462132       1.5973684817844545       1.5968984615945867
R:           -0.0007307881040742684   -9.783391759308203e-05   0.0003651989201990761
I^*_h/2 + R: 1.5972644410421388       1.5972706478668615       1.5972636605147859
```

4 Заключение

В ходе лабораторной работы были реализованы три метода численного интегрирования: трапеций, центральных прямоугольников и Симпсона. Для каждого метода проведена оценка погрешности с использованием правила Рунге.

Наиболее точный результат при заданной точности $\varepsilon = 0.001$ показал метод Симпсона, что подтверждается теоретической оценкой его погрешности $O(h^4)$. Также выявлено, что для достижения аналогичной точности метод трапеций и метод прямоугольников требуют большее число разбиений n.