ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ
(Вариант 9)

Выполнил студент 3 курса ПМиИ
Кондратьев Виталий

***Цель занятия*:**

изучение различных методов вычисления определенных интегралов, практическое интегрирование функций на ЭВМ.

Задания к работе.

1. Вычислить приближенно с заданной точностью интеграл $I = \int_a^b f(x)dx$ по формулам прямоугольников, трапеций и Симпсона. Величину шага определить с помощью двойного пересчета.

2. Определить относительную погрешность вычислений каждого метода по формуле: $\delta = \left| \frac{I - I_h}{I} \right| \cdot 100\,\%$, где I – точное значение интеграла; I_h – приближенное.

3. Составить таблицу в которой указать значение интеграла, полученное с заданной точностью, величину последнего шага интегрирования, количество точек разбиения, относительную погрешность метода.

Метод прямоугольников

Левых:

$$I = h\sum_{i=0}^{n-1} f(x_i), \quad \text{где } h = \frac{b-a}{n}$$

Правых

$$I = h\sum_{i=1}^{n} f(x_i), \quad \text{где } h = \frac{b-a}{n}$$

Погрешность абсолютная

$$\Delta = max\left|\frac{f'(x)}{2}\right|(b-a)h$$

Средних:

$$I = h\sum_{i=1}^{n} f\left(x_{i-1} + \frac{h}{2}\right), \quad \text{где } h = \frac{b-a}{n}$$

Оценка погрешности

$$\Delta = \boldsymbol{max}\left|\frac{\boldsymbol{f''(x)}}{\boldsymbol{24}}\right|(\boldsymbol{b}-\boldsymbol{a})\boldsymbol{h^2}$$

Метод трапеций

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

$$\Delta = max\left|\frac{f''(x)}{12}\right|(b-a)h^2$$

Метод Симпсона

$$I = \frac{h}{3}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_{2i}) + 4\sum_{i=1}^{n} f(x_{2i-1}) + f(x_n)\right]$$

Оценка погрешности

$$\Delta = \frac{b-a}{2880} h^4 max\left|f^{(4)}(x)\right|$$

Условие

№	Подынтегральная функция $f(x)$	Заданная точность	Интервал $[a, b]$	Первообразная функции $F(x)$
9	$x\ln x$	10^{-4}	[2; 6]	$\frac{x^2}{2}\ln x - \frac{x^2}{4}$

Ход выполнения:

<u>Метод левых прямоугольников</u>. Вычисления производим по формуле

$$I = h\sum_{i=0}^{n-1} f(x_i), \quad \text{где } h = \frac{b-a}{n}$$

Во всех методах проводим вычисления для одной итерации, сравниваем абсолютную разность вычисленного нового и предыдущего значения $|I_k - I_{k-1}|$ с заданной точностью $\varepsilon = 10^{-4}$, если модуль разности больше заданной точности, удваиваем n, иначе сохраняем ответ. Все результаты вычислений приведены в итоговой таблице.

<u>Метод правых прямоугольников</u>. Вычисления производим по формуле

$$I = h\sum_{i=1}^{n} f(x_i), \text{где } h = \frac{b-a}{n}$$

<u>Метод средних прямоугольников</u>. Вычисления производим по формуле

$$I = h\sum_{i=1}^{n} f\left(x_{i-1} + \frac{h}{2}\right), \quad \text{где } h = \frac{b-a}{n}$$

<u>Метод трапеций</u>. Вычисления производим по формуле

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

<u>Метод Симпсона.</u> Вычисления производим по формуле

$$I = \frac{h}{3}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_{2i}) + 4\sum_{i=1}^{n} f(x_{2i-1}) + f(x_n)\right]$$

Итоговая таблица

Точное решение 22.8654

Метод решения	Значение интеграла	Величина последнего шага интегрирования	Количество точек разбиения	Относительная погрешность
1. Метод левых прямоугольников	22.8648	0.0001220703125	32768	0.0025 %
2. Метод правых прямоугольников	22.8659	0.0001220703125	32768	0.0025 %
3. Метод средних Прямоугольников	22.8652	0.0625	64	0.0008 %
4. Метод трапеций	22.8655	0.03125	128	0.0004 %
5. Метод Симпсона	22.8654	0.5	8	0.0003 %

Вывод: Метод Симпсона оказался наиболее выгодным для решения этой задачи, точность выше всех остальных и разбиение всего на 8 точек – сошелся быстро. Код программы и ее вывод - в приложении.

ПРИЛОЖЕНИЕ

```
"""
Лабораторная работа №3
Студент ОНК «ИВТ» ВШ КНиИИ направления ПМиИ 3 курса
Кондратьев Виталий
Вариант 9
"""
import math


def rectangle_left_integral(a, b, f, eps):
    n = 2
    ans = 0
    while True:
        h = (b - a) / n
        x = a
        prev_approximation = ans
        ans = 0
        for i in range(n):
            ans += f(x) * h
            x += h

        error = abs(ans - prev_approximation)
        if error < eps:
            return ans, h, n

        n *= 2
```

```python
def rectangle_right_integral(a, b, f, eps):
    n = 2
    ans = 0
    while True:
        h = (b - a) / n
        x = a + h
        prev_approximation = ans
        ans = 0
        for i in range(n):
            ans += f(x) * h
            x += h

        error = abs(ans - prev_approximation)
        if error < eps:
            return ans, h, n

        n *= 2


def rectangle_mid_integral(a, b, f, eps):
    n = 2
    ans = 0
    while True:
        h = (b - a) / n
        x = a + h / 2
        prev_approximation = ans
        ans = 0
        for i in range(n):
            ans += f(x) * h
            x += h

        error = abs(ans - prev_approximation)
        if error < eps:
            return ans, h, n

        n *= 2


def trapezoidal_integral(a, b, f, eps):
    n = 2
    ans = 0
    while True:
        h = (b - a) / n
        x = a
        prev_approximation = ans
        ans = (f(a) + f(b)) / 2
        for i in range(1, n):
            x += h
            ans += f(x)
        ans *= h

        error = abs(ans - prev_approximation)
        if error < eps:
            return ans, h, n

        n *= 2


def simpson_integral(a, b, f, eps):
    n = 1
    ans = 0
    while True:
```

```
        h = (b - a) / n
        x = a
        prev_approximation = ans
        ans = (f(a) + f(b))
        for i in range(1, n):
            x += h
            if i % 2 == 0:
                ans += 2 * f(x)
            else:
                ans += 4 * f(x)
        ans *= h / 3

        error = abs(ans - prev_approximation)
        if error < eps:
            return ans, h, n

        n *= 2


def f(x):
    return x * math.log(x)


def F(x):
    return (x**2/2) * math.log(x) - x**2/4

def print_results(exact_value, calculation_results):
    calculated_value = calculation_results[0]
    h = calculation_results[1]
    n = calculation_results[2]

    relative_error = abs(100 * (exact_value - calculated_value) /
exact_value)
    print(
        f"Ответ: {calculated_value:.4f} \nИтоговый n: {n} \nПоследний шаг:
{h} \nОтносительная погрешность: {relative_error:.4f}\n")


if __name__ == "__main__":
    a = 2
    b = 6
    eps = 10e-4
    exact_solution = F(b) - F(a)
    print(f"Точное решение: {exact_solution:.4f}")

    print("Метод левых прямоугольников:")
    rli = rectangle_left_integral(a, b, f, eps)
    print_results(exact_solution, rli)

    print("Метод правых прямоугольников:")
    rri = rectangle_right_integral(a, b, f, eps)
    print_results(exact_solution, rri)

    print("Метод средних прямоугольников:")
    rmi = rectangle_mid_integral(a, b, f, eps)
    print_results(exact_solution, rmi)

    print("Метод трапеций:")
    ti = trapezoidal_integral(a, b, f, eps)
    print_results(exact_solution, ti)

    print("Метод Симпсона:")
    si = simpson_integral(a, b, f, eps)
    print_results(exact_solution, si)
```

Вывод программы:

```
Точное решение: 22.8654
Метод левых прямоугольников:
Ответ: 22.8648
Итоговый n: 32768
Последний шаг: 0.0001220703125
Относительная погрешность: 0.0025

Метод правых прямоугольников:
Ответ: 22.8659
Итоговый n: 32768
Последний шаг: 0.0001220703125
Относительная погрешность: 0.0025

Метод средних прямоугольников:
Ответ: 22.8652
Итоговый n: 64
Последний шаг: 0.0625
Относительная погрешность: 0.0008

Метод трапеций:
Ответ: 22.8655
Итоговый n: 128
Последний шаг: 0.03125
Относительная погрешность: 0.0004

Метод Симпсона:
Ответ: 22.8654
Итоговый n: 8
Последний шаг: 0.5
Относительная погрешность: 0.0003
```