Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа № 3: Численное интегрирование

Вычислительная математика

Вариант №4

Студент

Дубинин Артём Сергеевич

группа P3215

Преподаватель

Малышева Татьяна Алексеевна

Санкт - Петербург 2025 год

Вычислительная реализация задачи

$$\int_{-3}^{-1} (-2x^3 - 4x^2 + 8x - 4)\,dx$$

1. Вычислим интеграл точно, с помощью формулы Ньютона-Лейбница

$$\int_{-3}^{-1} (-2x^3 - 4x^2 + 8x - 4)\,dx$$

$$-\frac{2x^4}{4} - \frac{4x^3}{3} + \frac{8x^2}{2} - 4x + C = -\frac{x^4}{2} - \frac{4}{3}x^3 + 4x^2 - 4x + C$$

$$\left(-\frac{x^4}{2} - \frac{4}{3}x^3 + 4x^2 - 4x\right)\Big|_{-3}^{-1}$$

$$-\frac{-1^4}{2} - \frac{4}{3}(-1)^3 + 4 + 4 = 8.833$$

$$-\frac{-3^4}{2} - \frac{4}{3}(-3)^3 + 4 * 9 + 4 * 3 = 43.5$$

$$8.833 - 43.5 = -34.67$$

2. Метод Ньютона – Котеса при $n = 6$.

$$\int_a^b f(x)\,dx \approx \sum_{j=1}^{N} c_j f(x_j) = \frac{n \cdot h}{C_n} \sum_{j=1}^{N} \sum_{i=0}^{n} c_{in} f(x_i)$$

$$h = \frac{b-a}{n} \qquad C_n = \sum_{i=0}^{n} c_n^i$$

$$h = \frac{-1+3}{6} = \frac{1}{3}$$

i	*0*	*1*	*2*	*3*	*4*	*5*	*6*
x_i	*-3*	$-\frac{8}{3}$	$-\frac{7}{3}$	-2	$-\frac{5}{3}$	$-\frac{4}{3}$	*-1*
y_i	*-10*	$-\frac{428}{27}$	$-\frac{514}{27}$	-20	$\frac{-518}{27}$	$\frac{-460}{27}$	*-14*

$$I_{cotes} = \frac{n \cdot h}{C_n} \sum_{i=0}^{n} C_n^i f(x)$$

$$I_{cotes} = \frac{6 \cdot \frac{1}{3}}{840}\left(41 \cdot (-10) + 216 \cdot \left(-\frac{428}{27}\right) + 27 \cdot \left(-\frac{514}{27}\right) + 272 \cdot (-20) + 27 \cdot \left(\frac{-518}{27}\right) + \right.$$
$$\left. + 216 \cdot \left(\frac{460}{27}\right) + 41 \cdot -14\right) = -34.67$$

Погрешность = 0

3. Метод средних прямоугольников

$$\int_{-3}^{-1} (-2x^3 - 4x^2 + 8x - 4)dx$$

$$n = 10 \qquad h = \frac{b-a}{n} = 0.2$$

$$x_{i-1/2} = \frac{x_{i-1} + x_i}{2}$$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	-3	-2.8	-2.6	-2.4	-2.2	-2	-1.8	-1.6	-1.4	-1.2	-1
y_i	-10	-13.856	-16.688	-18.592	-19.663	-20	-19.696	-18.848	-17.552	-15.904	-14
$x_{i-1/2}$		-2.9	-2.7	-2.5	-2.3	-2.1	-1.9	-1.7	-1.5	-1.3	-1.1
$y_{i-1/2}$		-11.928	-15.272	-17.64	-19.1275	-19.832	-19.848	-19.272	-18.2	-16.728	-14.952

$$I_{\text{сред}} = h \cdot \sum_{i=1}^{n} y_{i-\frac{1}{2}} = 0.2 \cdot (-172.7995) = -34.5599$$

Погрешность в вычислении интеграла составляет :

$$\Delta I_{\text{средн}} = I - I_{\text{средн}} = -34.67 - (-34.5599) = -0.1101$$

4. Метод трапеций

$$I_{\text{трап}} = \int_{-3}^{-1} (-2x^3 - 4x^2 + 8x - 4)dx = h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right)$$

i	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
x_i	*-3*	*-2.8*	*-2.6*	*-2.4*	*-2.2*	*-2*	*-1.8*	*-1.6*	*-1.4*	*-1.2*	*-1*
y_i	*-10*	*-13.856*	*-16.688*	*-18.592*	*-19.663*	*-20*	*-19.696*	*-18.848*	*-17.552*	*-15.904*	*-14*

$$I_{\text{трап}} = 0.2 \cdot \left(\frac{-10 - 14}{2} + (-184.799)\right) = -39.35978$$

Погрешность в вычислении интеграла составляет :

$$\Delta I_{\text{трап}} = |I - I_{\text{трап}}| = |-34.67 - (-39.35978)| = 4.68978$$

5. Метод Симпсона

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

i	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
x_i	*-3*	*-2.8*	*-2.6*	*-2.4*	*-2.2*	*-2*	*-1.8*	*-1.6*	*-1.4*	*-1.2*	*-1*
y_i	*-10*	*-13.856*	*-16.688*	*-18.592*	*-19.663*	*-20*	*-19.696*	*-18.848*	*-17.552*	*-15.904*	*-14*

$$I = \frac{0.2}{3}(-10 + 4(-87.199) + 2(-73.599) - 14)) = -34.66626$$

Погрешность в вычислении интеграла составляет :

$$\Delta I_{\text{симп}} = |I - I_{\text{симп}}| = |-34.67 - (-34.66626)| = 0{,}0004$$

Программная реализация задачи

```
import math

def left_rectangles(f, a, b, n):
    h = (b - a) / n
    integral = 0.0
    for i in range(n):
        x = a + i * h
        integral += f(x)
    return integral * h

def right_rectangles(f, a, b, n):
    h = (b - a) / n
    integral = 0.0
    for i in range(1, n + 1):
        x = a + i * h
        integral += f(x)
    return integral * h

def middle_rectangles(f, a, b, n):
    h = (b - a) / n
    integral = 0.0
    for i in range(n):
        x = a + (i + 0.5) * h
        integral += f(x)
    return integral * h

def trapezoid(f, a, b, n):
    h = (b - a) / n
    integral = (f(a) + f(b)) / 2
    for i in range(1, n):
        x = a + i * h
        integral += f(x)
    return integral * h

def simpson(f, a, b, n):
    if n % 2 != 0:
        n += 1  # Метод Симпсона требует чётное n
    h = (b - a) / n
    integral = f(a) + f(b)
    for i in range(1, n):
        x = a + i * h
        if i % 2 == 0:
            integral += 2 * f(x)
        else:
            integral += 4 * f(x)
    return integral * h / 3

def calculate_integral():
    print("Выберите функцию для интегрирования:")
    print("1. x^2")
```

```
    print("2. sin(x)")
    print("3. e^x")
    print("4. 1/x")
    print("5. sqrt(x)")

    choice = int(input("Введите номер функции (1-5): "))

    if choice == 1:
        f = lambda x: x ** 2
        func_name = "x^2"
    elif choice == 2:
        f = lambda x: math.sin(x)
        func_name = "sin(x)"
    elif choice == 3:
        f = lambda x: math.exp(x)
        func_name = "e^x"
    elif choice == 4:
        f = lambda x: 1 / x
        func_name = "1/x"
    elif choice == 5:
        f = lambda x: math.sqrt(x)
        func_name = "sqrt(x)"
    else:
        print("Неверный выбор функции")
        return

    a = float(input("Введите нижний предел интегрирования a: "))
    b = float(input("Введите верхний предел интегрирования b: "))

    if choice == 4 and (a == 0 and b == 0):
        print("Ошибка: для функции 1/x пределы должны быть отличны от 0.")
        return
    if choice == 5 and (a < 0 or b < 0):
        print("Ошибка: для функции sqrt(x) пределы не могут быть отрицательными.")
        return
    if a >= b:
        print("Ошибка: верхний предел должен быть больше нижнего.")
        return

    epsilon = float(input("Введите требуемую точность (например, 0.001): "))

    print("\nВыберите метод интегрирования:")
    print("1. Левые прямоугольники")
    print("2. Правые прямоугольники")
    print("3. Средние прямоугольники")
    print("4. Метод трапеций")
    print("5. Метод Симпсона")

    method_choice = int(input("Введите номер метода (1-5): "))

    methods = {
        1: ("Левые прямоугольники", left_rectangles),
```

```
    2: ("Правые прямоугольники", right_rectangles),
    3: ("Средние прямоугольники", middle_rectangles),
    4: ("Метод трапеций", trapezoid),
    5: ("Метод Симпсона", simpson)
}

if method_choice not in methods:
    print("Неверный выбор метода")
    return

method_name, method_func = methods[method_choice]

print("\nРезультаты вычислений:")
print("Функция: ", func_name)
print("Интервал: [", a, ",", b, "]")
print("Метод: ", method_name)
print("Точность: ", epsilon)
print("\nИтерационный процесс:")

# Порядок точности для каждого метода
p = {
    1: 1,  # Левые прямоугольники
    2: 1,  # Правые прямоугольники
    3: 2,  # Средние прямоугольники
    4: 2,  # Метод трапеций
    5: 4   # Метод Симпсона
}[method_choice]

n = 4

while True:
    I_n = method_func(f, a, b, n)
    I_2n = method_func(f, a, b, 2 * n)
    runge_error = abs(I_2n - I_n) / (2 ** p - 1)

    print(f"\nРазбиение n = {n}")
    print(f"{method_name} (n): {I_n:.6f}")
    print(f"{method_name} (2n): {I_2n:.6f}")
    print(f"Оценка погрешности по правилу Рунге: {runge_error:.6f}")

    if runge_error < epsilon:
        integral = I_2n
        break

    n *= 2

print("\nИтоговые результаты:")
print(f"Достигнутая точность: {runge_error:.6f}")
print(f"Число разбиений для достижения точности: {n * 2}")
print(f"Значение интеграла методом {method_name}: {integral:.6f}")

print("\nТаблица значений для последнего разбиения:")
```

```
    h = (b - a) / (n * 2)
    if method_choice == 3:  # Средние прямоугольники
        print("i\tx_i\t\ty_i\t\tx_{i-1/2}\t\ty_{i-1/2}")
        for i in range(n * 2 + 1):
            x_i = a + i * h
            y_i = f(x_i)
            if i > 0:
                x_mid = a + (i - 0.5) * h
                y_mid = f(x_mid)
            else:
                x_mid = ""
                y_mid = ""
            print(f"{i}\t{x_i:.6f}\t{y_i:.6f}\t{x_mid if i > 0 else ":<12}\t{y_mid if i > 0 else
":<12}")
    elif method_choice == 5:  # Симпсон
        print("i\tx_i\t\ty_i\tКоэффициент")
        for i in range(n * 2 + 1):
            x_i = a + i * h
            y_i = f(x_i)
            if i == 0 or i == n * 2:
                coeff = 1
            elif i % 2 == 1:
                coeff = 4
            else:
                coeff = 2
            print(f"{i}\t{x_i:.6f}\t{y_i:.6f}\t{coeff}")
    else:
        print("i\tx_i\t\ty_i")
        for i in range(n * 2 + 1):
            x_i = a + i * h
            y_i = f(x_i)
            print(f"{i}\t{x_i:.6f}\t{y_i:.6f}")

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