Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

«Численное интегрирование»

по дисциплине «Вычислительная математика»

Вариант: **9**

Преподаватель:
Машина Екатерина Алексеевна

Выполнил:
Пронкин Алексей Дмитриевич
Группа: P3208

Санкт-Петербург, 2025 г.

Цель работы

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

1. Обязательная часть

1.1. Программаная реализация

main.py

```
# ---------------------------------------
# Функции, интегралы которых вычисляем
# ---------------------------------------
def f(x):
    """
    3(x)^2
    """
    return 3 * x ** 2


def f2(x):
    """
    3(x)^3+4x
    """
    return 3 * x ** 3 + 4 * x


def f3(x):
    """
    3(x)^2+4x+5
    """
    return 3 * x ** 2 + 4 * x + 5


# ---------------------------------------
# Методы вычисления интегралов
# ---------------------------------------

# ---------------------------------------
# прямоугольники
# ---------------------------------------
def left_rectangle_method(func, a, b, n):
    """
    Вычисление интеграла методом левых прямоугольников.

    Параметры:
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      n: число разбиений.

    Возвращает:
      Приближённое значение интеграла.
    """
    h = (b - a) / n
    total = 0.0
    for i in range(n):
```

```
        x_i = a + i * h  # левая точка подинтервала
        total += func(x_i)
    return total * h


def right_rectangle_method(func, a, b, n):
    """
    Вычисление интеграла методом правых прямоугольников.

    Параметры:
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      n: число разбиений.

    Возвращает:
      Приближённое значение интеграла.
    """
    h = (b - a) / n
    total = 0.0
    for i in range(1, n + 1):
        x_i = a + i * h  # правая точка подинтервала
        total += func(x_i)
    return total * h


def middle_rectangle_method(func, a, b, n):
    """
    Вычисление интеграла методом центральных прямоугольников.

    Параметры:
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      n: число разбиений.

    Возвращает:
      Приближённое значение интеграла.
    """
    h = (b - a) / n
    total = 0.0
    for i in range(n):
        x_i = a + i * h + h / 2  # центральная точка подинтервала
        total += func(x_i)
    return total * h


# ---------------------------------------
# трапеция
# ---------------------------------------
def trapezoid_method(func, a, b, n):
    """
    Вычисление интеграла методом трапеций.

    Параметры:
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      n: число разбиений.

    Возвращает:
      Приближённое значение интеграла.
```

```
    """
    h = (b - a) / n
    total = (func(a) + func(b)) / 2.0
    for i in range(1, n):
        x_i = a + i * h  # левая точка подинтервала
        total += func(x_i)
    return total * h


# ---------------------------------------
# симпсон
# ---------------------------------------
def simpson_method(func, a, b, n):
    """
    Вычисление интеграла методом Симпсона.

    Параметры:
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      n: число разбиений.

    Возвращает:
      Приближённое значение интеграла.
    """
    h = (b - a) / n
    total = func(a) + func(b)
    for i in range(1, n, 2):
        x_i = a + i * h  # левая точка подинтервала
        total += 4 * func(x_i)
    for i in range(2, n, 2):
        x_i = a + i * h
        total += 2 * func(x_i)
    return total * h / 3


# ---------------------------------------
# Вычисление интеграла с правилом Рунге
# ---------------------------------------
def integrate_with_runge(method, func, a, b, eps, order=1, initial_n=4):
    """
    Вычисление интеграла с использованием заданного метода и правила Рунге для
    оценки погрешности.

    Параметры:
      method: функция для вычисления интеграла (например, left_rectangle_method или
       right_rectangle_method),
      func: интегрируемая функция,
      a, b: пределы интегрирования,
      eps: требуемая точность,
      order: порядок метода (для прямоугольников order = 1),
      initial_n: начальное число разбиений (по умолчанию 4).

    Возвращает:
      Кортеж (интегральное приближение, итоговое число разбиений).
    """
    n = initial_n
    I_n = method(func, a, b, n)
    n *= 2
    I_2n = method(func, a, b, n)
```

```python

    # Правило Рунге: |I(2n) - I(n)| / (2^order - 1) < eps
    while abs(I_2n - I_n) / (2 ** order - 1) > eps:
        I_n = I_2n
        n *= 2
        I_2n = method(func, a, b, n)

    return I_2n, n


def main():
    valid_choices = {'1', '2', '3'}

    function_choice = None
    function_to_compute = None
    print("Вычисление интеграла функции")
    print("Выберите функцию, которую хотите проинтегрировать:")
    print("1 - 3x^2")
    print("2 - 3x^3 + 4x")
    print("3 - 3x^2 + 4x + 5")

    while function_choice not in valid_choices:
        function_choice = input("Введите номер функции: ").strip()
        if function_choice not in valid_choices:
            print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")

    if function_choice == '1':
        function_to_compute = f
    elif function_choice == '2':
        function_to_compute = f2
    elif function_choice == '3':
        function_to_compute = f3


    print("Выберите метод вычисления интеграла:")
    print("1 - Метод прямоугольников")
    print("2 - Метод трапеций")
    print("3 - Метод Симпсона")

    choice = None

    while choice not in valid_choices:
        choice = input("Введите номер метода: ").strip()
        if choice not in valid_choices:
            print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")

    if choice == "1":
        print("Какой прямоугольник будем использовать для вычислений?")
        print("1 - Левый")
        print("2 - Средний")
        print("3 - Правый")

        choice2 = None

        while choice2 not in valid_choices:
            choice2 = input("Введите номер метода: ").strip()
            if choice2 not in valid_choices:
                print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")

```

```python
        if choice2 == "1":
            integration_method = left_rectangle_method
            order = 1  # порядок метода для прямоугольников (левый)
        elif choice2 == "2":
            integration_method = middle_rectangle_method
            order = 2  # порядок метода для прямоугольников (средний)
        elif choice2 == "3":
            integration_method = right_rectangle_method
            order = 1  # порядок метода для прямоугольников (правый)
    elif choice == "2":
        integration_method = trapezoid_method
        order = 2
    elif choice == "3":
        integration_method = simpson_method
        order = 4
    else:
        print("Некорректный выбор метода.")
        return

    try:
        a = float(input("Введите нижний предел интегрирования (a): "))
        b = float(input("Введите верхний предел интегрирования (b): "))
        eps = float(input("Введите требуемую точность (eps): "))
    except ValueError:
        print("Ошибка: необходимо вводить числовые значения.")
        return

    result, subdivisions = integrate_with_runge(
        integration_method, function_to_compute, a, b, eps, initial_n=4, order=order
    )

    print("\nРезультаты вычисления:")
    print(f"Приближённое значение интеграла: {result}")
    print(f"Число разбиений для достижения требуемой точности: {subdivisions}")


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

1.2. Вычислительная реализация

$$\int_1^2 \left(2x^3 - 3x^2 + 5x - 9\right) dx$$

1. Точное аналитическое вычисление интеграла

$$\int_1^2 \left(2x^3 - 3x^2 + 5x - 9\right) dx = \frac{x^4}{2} - x^3 + \frac{5x^2}{2} - 9x\Big|_1^2 = \left(\frac{2^4}{2} - 2^3 + \frac{5 \cdot 2^2}{2} - 9 \cdot 2\right) - \left(\frac{1^4}{2} - 1^3 + \frac{5 \cdot 1^2}{2} - 9 \cdot 1\right)$$

$$= -8 - (-7) = -1$$

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

При разбиении отрезка $[1, 2]$ на $n = 6$ равных промежутков получаем шаг

$$h = \frac{2-1}{6} = \frac{1}{6}.$$

Это приводит к 7 узлам:

$$\begin{array}{ll} x_0 = 1, & x_1 = 1 + \frac{1}{6} = \frac{7}{6}, \\ x_2 = 1 + \frac{2}{6} = \frac{4}{3}, & x_3 = 1 + \frac{3}{6} = 1.5, \\ x_4 = 1 + \frac{4}{6} = \frac{5}{3}, & x_5 = 1 + \frac{5}{6} = \frac{11}{6}, \\ x_6 = 2. & \end{array}$$

Из справочных таблиц для закрытой формулы Ньютона–Котеса при $n = 6$ имеют вид:

$$I_{\text{NC}} = \frac{h}{140}\Big[41f(x_0) + 216f(x_1) + 27f(x_2) + 272f(x_3) + 27f(x_4) + 216f(x_5) + 41f(x_6)\Big].$$

Вычислим $f(x) = 2x^3 - 3x^2 + 5x - 9$ в узловых точках:

- $x_0 = 1$:
$$f(1) = 2 - 3 + 5 - 9 = -5.$$
- $x_1 = \frac{7}{6}$:
$$f\left(\frac{7}{6}\right) = 2\left(\frac{7}{6}\right)^3 - 3\left(\frac{7}{6}\right)^2 + 5\left(\frac{7}{6}\right) - 9.$$
  Вычисляя подробнее (приводя к общему знаменателю), получим:
$$f\left(\frac{7}{6}\right) = -\frac{110}{27} \quad (\approx -4.0741).$$
- $x_2 = \frac{4}{3}$:
$$f\left(\frac{4}{3}\right) = -\frac{79}{27} \quad (\approx -2.9259).$$
- $x_3 = 1.5$:
$$f(1.5) = 2(3.375) - 3(2.25) + 7.5 - 9 = 6.75 - 6.75 + 7.5 - 9 = -1.5.$$
- $x_4 = \frac{5}{3}$:
$$f\left(\frac{5}{3}\right) = \frac{7}{27} \quad (\approx 0.2593).$$
- $x_5 = \frac{11}{6}$:
$$f\left(\frac{11}{6}\right) = \frac{65}{27} \quad (\approx 2.4074).$$
- $x_6 = 2$:
$$f(2) = 16 - 12 + 10 - 9 = 5.$$

Подстановка в формулу

Подставляем все значения в формулу:

$$I_{\text{NC}} = \frac{1/6}{140}\Big[41 \cdot (-5) + 216 \cdot \left(-\frac{110}{27}\right) + 27 \cdot \left(-\frac{79}{27}\right)$$
$$+ 272 \cdot (-1.5) + 27 \cdot \frac{7}{27} + 216 \cdot \frac{65}{27} + 41 \cdot 5\Big].$$

Выполним поэлементное умножение: - $41f(1) = 41\cdot(-5) = -205$, - $216f(7/6) = 216\cdot\left(-\frac{110}{27}\right) = -880$ (так как $216/27 = 8$, и $8 \cdot 110 = 880$), - $27f(4/3) = 27 \cdot \left(-\frac{79}{27}\right) = -79$, - $272f(1.5) = 272 \cdot (-1.5) = -408$, - $27f(5/3) = 27 \cdot \frac{7}{27} = 7$, - $216f(11/6) = 216 \cdot \frac{65}{27} = 520$ (так как $216/27 = 8$ и $8 \cdot 65 = 520$), - $41f(2) = 41 \cdot 5 = 205$.

Суммируем:

$$-205 - 880 - 79 - 408 + 7 + 520 + 205 = -840.$$

Таким образом,

$$I_{\text{NC}} = \frac{1}{840} \cdot (-840) = -1.$$

Вывод: по формуле Ньютона–Котеса при $n = 6$ интеграл вычисляется точно и равен -1.

3. Численные методы при $n = 10$

Для всех методов интервал $[1, 2]$ делится на $n = 10$ равных частей, поэтому шаг равен

$$h = \frac{2-1}{10} = 0.1.$$

3.1. Метод средних прямоугольников (центральных точек)

При методе средних прямоугольников вычисляют значение функции в серединах каждого из 10 интервалов. Середины подынтервалов:

$$x_i^* = 1 + \left(i - \tfrac{1}{2}\right) h, \quad i = 1, \ldots, 10,$$

то есть

$$1.05,\ 1.15,\ 1.25,\ 1.35,\ 1.45,\ 1.55,\ 1.65,\ 1.75,\ 1.85,\ 1.95.$$

Приближённое значение интеграла:

$$I_{\text{mid}} = h \sum_{i=1}^{10} f(x_i^*).$$

Вычислим значения функций:

x_i^*	$f(x_i^*) \approx$
1.05	-4.74225
1.15	-4.17575
1.25	-3.53125
1.35	-2.79675
1.45	-1.96025
1.55	-1.00975
1.65	0.06575
1.75	1.28125
1.85	2.64775
1.95	4.17225

Сумма:

$$\sum_{i=1}^{10} f(x_i^*) \approx -10.049.$$

Тогда

$$I_{\text{mid}} = 0.1 \cdot (-10.049) \approx -1.0049.$$

Относительная погрешность:

$$\varepsilon_{\text{mid}} = \frac{|I_{\text{mid}} - (-1)|}{|-1|} = \frac{0.0049}{1} \approx 0.49\%.$$

3.2. Метод трапеций

Для метода трапеций используем узловые точки $x_i = 1 + 0.1 \cdot i,\ i = 0, 1, \ldots, 10$. Значения функции в узлах:

x_i	$f(x_i) \approx$
1.0	-5.0000
1.1	-4.4680
1.2	-3.8640
1.3	-3.1760
1.4	-2.3920
1.5	-1.5000
1.6	-0.4880
1.7	0.6560
1.8	1.9440
1.9	3.3880
2.0	5.0000

Формула метода трапеций:

$$I_{\text{trap}} = \frac{h}{2}\Big[f(x_0) + f(x_{10}) + 2\sum_{i=1}^{9} f(x_i)\Big].$$

Сначала суммируем значения для $i = 1, \ldots, 9$:

$$\sum_{i=1}^{9} f(x_i) \approx -4.468 - 3.864 - 3.176 - 2.392 - 1.500 - 0.488 + 0.656 + 1.944 + 3.388 = -9.900.$$

Подставляем:

$$I_{\text{trap}} = \frac{0.1}{2}\Big[(-5) + 5 + 2(-9.900)\Big] = \ 0.05\,(0 - 19.8) = -0.99.$$

Относительная погрешность:

$$\varepsilon_{\text{trap}} = \frac{|-0.99 - (-1)|}{1} = 0.01 \quad (1\%).$$

3.3. Метод Симпсона

Формула Симпсона при чётном $n = 10$:

$$I_{\text{Simpson}} = \frac{h}{3}\Big[f(x_0) + f(x_{10}) + 4 \sum_{\substack{i=1 \\ i\ \text{нечётное}}}^{9} f(x_i) + 2 \sum_{\substack{i=2 \\ i\ \text{чётное}}}^{8} f(x_i)\Big].$$

Используем те же значения $f(x_i)$ из таблицы выше.
Сумма по нечётным индексам (x_1, x_3, x_5, x_7, x_9):

$$S_{\text{odd}} = -4.468 + (-3.176) + (-1.500) + 0.656 + 3.388 = -5.100.$$

Сумма по чётным индексам (x_2, x_4, x_6, x_8):

$$S_{\text{even}} = -3.864 + (-2.392) + (-0.488) + 1.944 = -4.800.$$

Подставляем:

$$I_{\text{Simpson}} = \frac{0.1}{3}\Big[(-5) + 5 + 4(-5.100) + 2(-4.800)\Big] = \frac{0.1}{3}\Big[0 - 20.4 - 9.6\Big] = \frac{0.1}{3}(-30) = -1.$$

Относительная погрешность:

$$\varepsilon_{\text{Simpson}} = 0 \quad (0\%).$$

4. Сравнение результатов и относительные погрешности

Метод	**Результат**	**Абсолютная ошибка**	**Относительн**
Точное значение	-1	–	
Ньютона–Котеса ($n = 6$)	-1	0	
Метод средних прямоугольников ($n = 10$)	$-1{,}0049$	$\lvert -1{,}0049 - (-1)\rvert \approx 0{,}0049$	
Метод трапеций ($n = 10$)	$-0{,}99$	$\lvert -0{,}99 - (-1)\rvert \approx 0{,}01$	
Метод Симпсона ($n = 10$)	-1	0	

5. Вывод

1. **Точное значение интеграла**: -1.
2. **Формула Ньютона–Котеса** при $n = 6$ (7 узлов) даёт точный результат: -1.
3. **Метод средних прямоугольников** при $n = 10$ даёт значение ≈ -1.0049 (относительная погрешность $\approx 0.49\%$).
4. **Метод трапеций** при $n = 10$ даёт значение ≈ -0.99 (относительная погрешность $\approx 1\%$).
5. **Метод Симпсона** при $n = 10$ вычисляет интеграл точно: -1.

Таким образом, как видно из сравнения, для данного многочлена (степени 3) методы, обладающие достаточной точностью (Ньютон–Котеса и Симпсона), дают абсолютно точное значение, в то время как методы средних прямоугольников и трапеций имеют незначительную погрешность.

9. Расчёт погрешностей измерений

Аппроксимируем зависимость $U(I)$ прямой

$$U_i = \varepsilon - r\, I_i$$

методом наименьших квадратов по $n = 15$ экспериментам.

Обозначения:

$$\bar{I} = \frac{1}{n}\sum_{i=1}^{n} I_i = \frac{14.37 + 12.90 + \cdots + 4.88}{15} \approx 8.026\ \text{мА},$$

$$S = \sum_{i=1}^{n}\big[\,U_i - (\varepsilon - r\, I_i)\big]^2 \approx 0.086646\ \text{В}^2,$$

$$D = \sum_{i=1}^{n}(I_i - \bar{I})^2 \approx 126.012\ (\text{мА})^2.$$

Стандартные ошибки параметров (1 σ):

$$s_r = \sqrt{\frac{S}{(n-2)\,D}} = \sqrt{\frac{0.086646}{13 \cdot 126.012}} \approx 0.007273\ \Omega,$$

$$s_\varepsilon = \sqrt{\frac{S}{n-2}\Big(\frac{1}{n} + \frac{\bar{I}^2}{D}\Big)} = \sqrt{\frac{0.086646}{13}\Big(\frac{1}{15} + \frac{8.026^2}{126.012}\Big)} \approx 0.06206\ \text{В}.$$

Для доверительного интервала ≈95% (коэф. $t \approx 2$) получаем **абсолютные** погрешности:

$$\Delta r = 2\, s_r \approx 2 \cdot 0.007273 = 0.01455\ \Omega, \qquad \Delta\varepsilon = 2\, s_\varepsilon \approx 2 \cdot 0.06206 = 0.12412\ \text{В}.$$

Относительные погрешности:

$$\delta_r = \frac{\Delta r}{|r|}\times 100\% = \frac{0.01455}{0.68099}\times 100\% \approx 2.14\%, \qquad \delta_\varepsilon = \frac{\Delta\varepsilon}{\varepsilon}\times 100\% = \frac{0.12412}{10.88566}\times 100\% \approx 1.14\%.$$

11. Окончательные результаты

Доверительные интервалы (≈95%) для параметров источника:

$$r = (0{,}680995 \pm 0{,}014545)\ \Omega, \quad \delta_r = \frac{0{,}014545}{0{,}680995} \cdot 100\% \approx 2{,}14\%;$$

$$\mathcal{E} = (10{,}885665 \pm 0{,}124120)\ \text{В}, \quad \delta_{\mathcal{E}} = \frac{0{,}124120}{10{,}885665} \cdot 100\% \approx 1{,}14\%.$$

Ток максимальной полезной мощности:

$$I^*_{\text{теор}} = \frac{\mathcal{E}}{2\,r} = \frac{10{,}885665}{2 \cdot 0{,}680995} \approx 7{,}9925\ \text{мА},$$

$$I^*_{\text{эксп}} = 8{,}3100\ \text{мА} \quad \Big(P^{\text{эксп}}_{R,\max} = 44{,}0430\ \text{мВт}\Big),$$

$$\Delta I^* = \left|I^*_{\text{эксп}} - I^*_{\text{теор}}\right| = |8{,}3100 - 7{,}9925| = 0{,}3175\ \text{мА},$$

$$P^{\text{теор}}_{R,\max} = \frac{\mathcal{E}^2}{4\,r} = \frac{(10{,}885665)^2}{4 \cdot 0{,}680995} \approx 43{,}5017\ \text{мВт},$$

$$\Delta P_{R,\max} = \big|44{,}0430 - 43{,}5017\big| = 0{,}5413\ \text{мВт}.$$

Режим согласования нагрузки и источника:

$$R_{\text{согл}} = \frac{P^{\text{теор}}_{R,\max}}{\left(I^*_{\text{теор}}\right)^2} = 0{,}680995\ \Omega \approx r.$$

Проверка тока при $\eta = 0{,}5$:

$$I_{\eta=0.5} = 7{,}9925\ \text{мА} \quad \Big(\text{из графика аппроксимации}\Big),$$

$$\Delta I_{\eta=0.5} = \left|I^*_{\text{теор}} - I_{\eta=0.5}\right| = |7{,}9925 - 7{,}9925| = 0{,}0000\ \text{мА}.$$