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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа по дисциплине Вычислительная Математика №3 «Численное интегрирование»

Вариант 9

Преподаватель: Машина Е. А.

Выполнил: Камянецький Н.В.

Группа: Р3208

Санкт-Петербург, 2024г

Цель работы:

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

Рабочие формулы методов

Методы прямоугольников

$$\int_a^b f(x)dx = h\sum_{i=1}^{n} y_{i-1} - \text{Левые}$$

$$\int_a^b f(x)dx = h\sum_{i=1}^{n} y_{i} - \text{Правые}$$

$$\int_a^b f(x)dx = h\sum_{i=1}^{n} f(\frac{x_i + x_{i-1}}{2}) - \text{Средние}$$

Метод трапеций

$$\int_a^b f(x)dx = h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right)$$

Метод Симпсона

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

Вычисление заданного интеграла

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

$$\int_1^2 (2x^3 - 3x^2 + 5x - 9)\,dx = -1$$

Ньютон лейбн

$$\frac{2x^4}{4} - \frac{3x^3}{3} + \frac{5x^2}{2} - 9x \Big|_1^2$$

Ньютон котес $(n=6)$

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

$$x_i = 1 + i \cdot \frac{1}{6} \qquad i = 0, 1, 2, 3, 4, 5, 6$$

$$x_0 = 1 \quad x_1 = \frac{7}{6} \quad x_2 = \frac{4}{3} \quad x_3 = \frac{3}{2} \quad x_4 = \frac{5}{3} \quad x_5 = \frac{11}{6} \quad x_6 = 2$$

$$\frac{1}{840}\left(41(f(1)+f(2)) + 216\left(f\left(\tfrac{7}{6}\right) + f\left(\tfrac{11}{6}\right)\right) + 27\cdot\left(f\left(\tfrac{4}{3}\right) + f\left(\tfrac{5}{3}\right)\right) + 272 f\left(\tfrac{3}{2}\right)\right)$$
$= 1$

По формуле средних прямоугольников

$$I_{ср} = h \sum_{i=1}^{n} y_{i-1} \qquad h = \frac{b-a}{n} = 0.1$$

≈ -1

i	1	2	3	4	5	6	7	8	9	10
$x_{i-\frac{1}{2}}$	1.05	1.15	1.25	1.35	1.45	1.55	1.65	1.75	1.85	1.95
$y_{i-\frac{1}{2}}$ ~~-4.74 -4.17~~	-4.75	-4.17	-3.53	-2.79	-1.96	-1.01	0.07	1.28	2.65	4.17

Трапеции

$$I_{трап} = h\cdot\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right) \approx -0.99$$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
y_i	-5	-4.47	-3.86	-3.17	-2.39	-1.5	-0.46	0.65	1.94	3.38	5

Симпсона

Таблица трапеции идентична

$$I = \frac{h}{3}\left(-5 + 4(y_1 + y_3 \ldots^{\text{нечёт}}) + 2(y_{i - \text{чёт}}) + 5\right) = -1$$

Листинг программы:

```python
import numpy as np

def init_choice():
    print("Выберите функцию для интегрирования:")
    print("1. x^2 - 4x + 4")
    print("2. sin(x)")
    print("3. e^x")
    choice = int(input("Введите номер функции: "))
    equations = {1: equation1, 2: equation2, 3: equation3}
    return equations.get(choice, None)

def equation1(x):
    return x**2 - 4*x + 4

def equation2(x):
    return np.sin(x)

def equation3(x):
    return np.exp(x)

def chose_rec_method():
    return input("Введите название для метода прямоугольников
[left/middle/right]: ")

# Метод прямоугольников (левые, правые, средние)
def rectangle_method(a, b, n, method):
    h = (b - a) / n
    integral = 0
    for i in range(n):
        if method == "left":
            x = a + i * h
        elif method == "right":
            x = a + (i + 1) * h
        elif method == "middle":
            x = a + (i + 0.5) * h
        integral += func(x)
    return integral * h

# Метод трапеций
def trapezoidal_method(a, b, n):
    h = (b - a) / n
    integral = (func(a) + func(b)) / 2
    for i in range(1, n):
        integral += func(a + i * h)
    return integral * h

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

# Функция для оценки погрешности метода
def runge_rule(prev_integral, integral, order):
    return abs(integral - prev_integral) / (2 ** order - 1)

# Функция для вычисления интеграла с заданной точностью
def compute_integral_with_precision(func, a, b, precision, method):
```

```python
    n = 4
    prev_integral = 0
    if method == "rectangle":
        choice = chose_rec_method()
    while True:
        if method == "rectangle":
            integral = rectangle_method(a, b, n, choice)
            order = 2
        elif method == "trapezoidal":
            integral = trapezoidal_method(a, b, n)
            order = 2
        elif method == "simpson":
            integral = simpson_method(a, b, n)
            order = 4
        if prev_integral != 0:
            error = runge_rule(prev_integral, integral, order)
            if error < precision:
                return integral, n
        prev_integral = integral
        n *= 2

# Основная программа
if __name__ == "__main__":
    # Ввод функции
    func = init_choice()

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

    # Ввод требуемой точности
    precision = float(input("Введите требуемую точность: "))

    # Выбор метода
    print("Выберите метод интегрирования:")
    print("1. Метод прямоугольников")
    print("2. Метод трапеций")
    print("3. Метод Симпсона")
    method_choice = int(input("Введите номер метода: "))

    methods = {
        1: "rectangle",
        2: "trapezoidal",
        3: "simpson"
    }
    method = methods[method_choice]

    # Вычисление интеграла с заданной точностью
    integral, n = compute_integral_with_precision(func, a, b, precision,
method)
    print(f"Приближенное значение интеграла: {integral}")
    print(f"Число разбиений интервала: {n}")
```

Вывод

В ходе выполнения работы удалось найти приближенное значение определенного интеграла с требуемой точностью методами Ньютона-Котеса, Симпсона, трапеций и прямоугольников, а также реализовать их при помощи программы