

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3 по "Вычислительной математике"

Численное интегрирование

Вариант 4

Выполнил: Дьяконов Михаил Павлович  
Группа: Р3211  
Преподаватель: Малышева Татьяна Алексеевна

## Цель работы

Реализовать 3 метода численного интегрирования. Предусмотреть вывод результатов вычислений.

## Описание методов

Метод прямоугольников:

$$\int_a^b f(x) dx h_1 y_0 + h_2 y_1 + \dots + h_n y_{n-1} = \sum_{i=1}^n h_i y_{i-1} - \text{левые прямоугольники}$$

$$\int_a^b f(x) dx h_1 y_1 + h_2 y_2 + \dots + h_n y_n = \sum_{i=1}^n h_i y_i - \text{правые прямоугольники}$$

При  $h_i = h = \frac{b-a}{n} = \text{const}$ :

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

$$\int_a^b f(x) dx = h \sum_{i=1}^n y_i$$

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

Метод трапеций:

$$\int_a^b f(x) dx = \frac{1}{2} \sum_{i=1}^n h_i (y_{i-1} + y_i)$$

Метод Симпсона:

$$\int_a^b f(x) dx = \frac{h}{3} [(y_0 + 4(y_1 + y_3 + \dots + y_{n-1})) + y_n]$$

## Листинг программы

RectangleMethod.kt:

---

```
class RectangleMethod : IntegralCalculationMethod {

    override fun calculate(input: UserInput, splitNumber: Int): Double {
        val h: Double = (input.interval.finish - input.interval.start) / splitNumber
        val rectangleSum: Double = calculateRectangleSum(splitNumber, h, input)
        return h * rectangleSum
    }

    private fun calculateRectangleSum(n: Int, h: Double, input: UserInput): Double {
        var sum: Double = 0.0
        var x: Double = when (input.method) {
            Method.RECTANGLE_LEFT -> input.interval.start
            Method.RECTANGLE_RIGHT -> input.interval.start + h
            Method.RECTANGLE_MIDDLE -> input.interval.start + h / 2
            else -> throw IllegalArgumentException()
        }
        for (i in 0 until n) {
            sum += input.selectedFunction.func(x)
            x += h
        }
        return sum
    }
}
```

---

TrapeziumMethod.kt:

---

```
class TrapeziumMethod : IntegralCalculationMethod {
```

---

```

override fun calculate(input: UserInput, splitNumber: Int): Double {
    val h: Double = (input.interval.finish - input.interval.start) / splitNumber
    val trapezoidSum: Double = calculateTrapezoidSum(splitNumber, h, input)
    return h / 2 * trapezoidSum
}

private fun calculateTrapezoidSum(n: Int, h: Double, input: UserInput): Double {
    var sum: Double = 0.0;
    var x: Double = input.interval.start
    for (i in 0..n) {
        sum += if(i == 0 || i == n) input.selectedFunction.func(x)
               else 2 * input.selectedFunction.func(x)
        x += h
    }
    return sum
}
}

```

---

SimpsonMethod.kt:

```

class SimpsonMethod : IntegralCalculationMethod {

    override fun calculate(input: UserInput, splitNumber: Int): Double {
        val h: Double = (input.interval.finish - input.interval.start) / splitNumber
        val simpsonSum: Double = calculateSimpsonSum(splitNumber, h, input)
        return h / 3 * simpsonSum
    }

    private fun calculateSimpsonSum(n: Int, h: Double, input: UserInput): Double {
        var sum: Double = 0.0
        var x: Double = input.interval.start
        for (i in 0..n) {
            sum += if (i == 0 || i == n) input.selectedFunction.func(x)
                   else if (i % 2 == 1) 4 * input.selectedFunction.func(x)
                   else 2 * input.selectedFunction.func(x)
            x += h
        }
        return sum
    }
}

```

---

IntegralService.kt:

```

class IntegralService {

    fun calculateIntegralWithAccuracy(input: UserInput): CalculationResult {
        val calculationMethod: IntegralCalculationMethod = when (input.method) {
            Method.RECTANGLE_MIDDLE, Method.RECTANGLE_LEFT, Method.RECTANGLE_RIGHT -> RectangleMethod()
            Method.TRAPEZIUM -> TrapeziumMethod()
            Method.SIMPSON -> SimpsonMethod()
        }

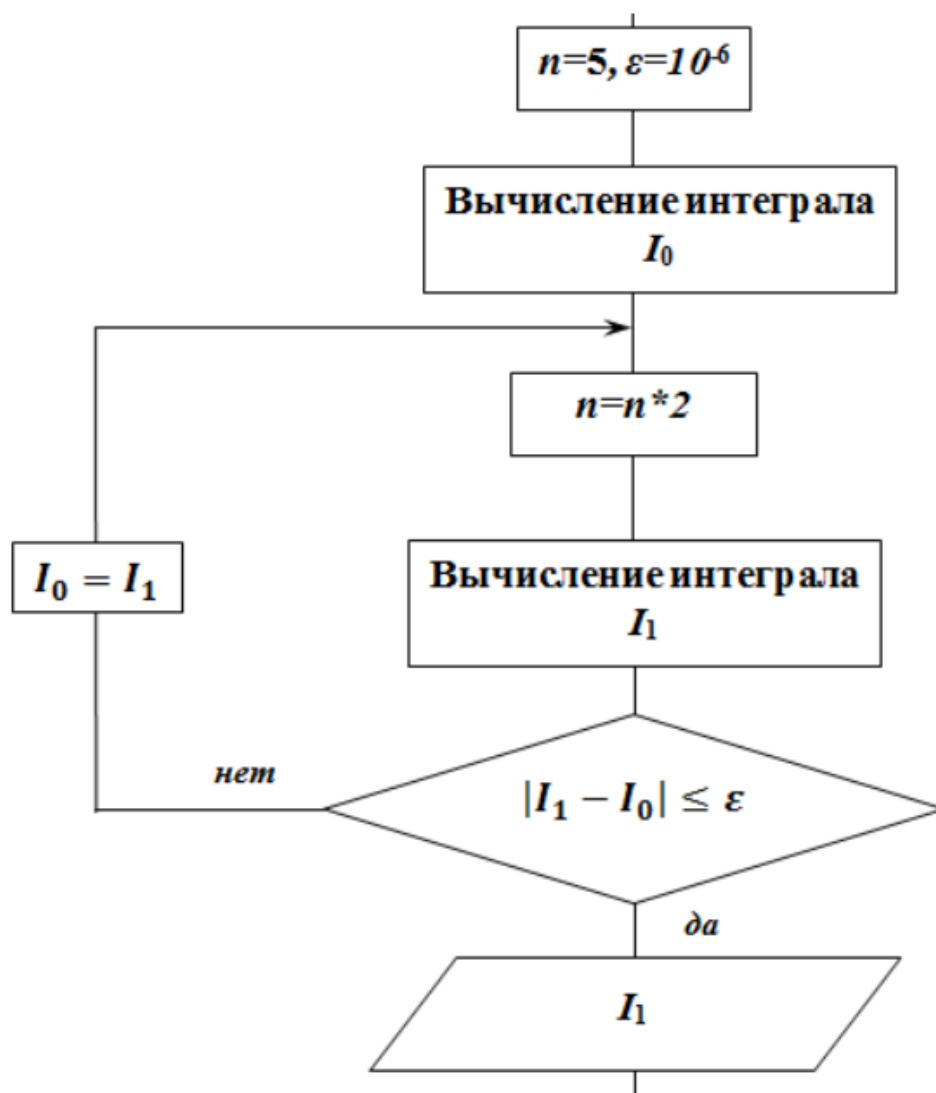
        var splitNumber: Int = SPLIT_NUMBER
        var i0: Double; var i1: Double = calculationMethod.calculate(input, splitNumber)
        do{
            i0 = i1
            i1 = calculationMethod.calculate(input, splitNumber*2)
            splitNumber*=2
        } while (abs(i0-i1) > input.accuracy)

        return CalculationResult(i1, splitNumber, abs(i0-i1))
    }
}

```

---

Алгоритм вычисления интеграла:



## Примеры

Доступные методы:

- 1) Метод левых прямоугольников
- 2) Метод правых прямоугольников
- 3) Метод средних прямоугольников
- 4) Метод трапеций
- 5) Метод Симпсона

Введите номер нужного метода: 3

Доступные функции:

- 1)  $-2x^3 - 4x^2 + 8x - 4$
- 2)  $x^2 + 4x - 17$
- 3)  $\cos(2x)/(x^2 + 1)$
- 4)  $\sqrt{x^4/7 + 3}$

Введите номер нужной функции: 1

Введите пределы интегрирования: -3 -1

Введите точность: 0.01

Значение интеграла: -34.667969

Число разбиений: 64

Погрешность: 0.003906

Доступные методы:

- 1) Метод левых прямоугольников
- 2) Метод правых прямоугольников
- 3) Метод средних прямоугольников
- 4) Метод трапеций
- 5) Метод Симпсона

Введите номер нужного метода: 5

Доступные функции:

- 1)  $-2x^3 - 4x^2 + 8x - 4$
- 2)  $x^2 + 4x - 17$
- 3)  $\cos(2x)/(x^2 + 1)$
- 4)  $\sqrt{x^4/7 + 3}$

Введите номер нужной функции: 1

Введите пределы интегрирования: -3 -1

Введите точность: 0.01

Значение интеграла: -34.666667

Число разбиений: 20

Погрешность: 0.0

## Вывод

В результате выполнения данной лабораторной работы я узнал какие есть численные методы интегрирования. Также узнал о преимуществах и недостатках каждого из них и реализовал их.