Санкт-Петербургский национальный исследовательский университет ИТМО

Вычислительная математика

Лабораторная работа №3
Численное интегрирование

Автор:

Ненов Владислав Александрович

Вариант 5

Группа №P32082

Преподаватель:

Екатерина Алексеевна Машина

Санкт-Петербург
2023

Цель работы

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

Вычислительная часть

Задание

1. Вычислить интеграл, приведенный в таблице 1, точно.
2. Вычислить интеграл по формуле Ньютона – Котеса при n = 5
3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при n = 10.
4. Сравнить результаты с точным значением интеграла.
5. Определить относительную погрешность вычислений для каждого метода.
6. В отчете отразить последовательные вычисления.

Выполнение

$$\int_2^4 (-2x^3 - 3x^2 + x + 5)dx$$

Точное вычисление

$$\int_2^4 (-\ 2x^3 - 3x^2 + x + 5)dx = -\ 164 + 4 =-\ 160$$

Вычисления по формуле Ньютона-Котеса (n=5)

$h = 2 / 5 = 0.4$

x_i	2	2.4	2.8	3.2	3.6	4
y_i	-28.62	-37.53	-59.62	-88.05	-123.59	-167

$$\frac{19(b-a)}{288}y_0 + \frac{75(b-a)}{288}y_1 + \frac{50(b-a)}{288}y_2 + \frac{50(b-a)}{288}y_3 + \frac{75(b-a)}{288}y_4 + \frac{19(b-a)}{288}y_5 \approx -\ 161.002$$

$\Delta I = 0.626\%$

Вычисления по формуле средних прямоугольников (n=10)

i	0	1	2	3	4	5	6	7	8	9
x_i	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
y_i	-21	-28.62	-37.53	-47.83	-59.62	-73	-88.05	-104.89	-123.59	-144.26
$x_{i+1/2}$	2.1	2.3	2.5	2.7	2.9	3.1	3.3	3.5	3.7	3.9
$y_{i+1/2}$	-24.65	-32.9	-42.5	-53.53	-66.10	-80.31	-96.24	-114.0	-133.67	-155.37

$\sum y_{i+1/2} * h = \; -159.86$

$\Delta I = 0.0875\%$

Вычисления по формуле трапеции (n=10)

i	1	2	3	4	5	6	7	8	9
x_i	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
y_i	-28.61	-37.52	-47.83	-59.62	-73	-88.05	-104.88	-123.59	-144.26

$(\sum y_i * 2 + a + b) * h/2 = (-707.4 * 2 + 2 + 4) * 0.1 = \; -140.88$

$\Delta I = 11.95\%$

Вычисления по формуле Симпсона (n=10)

i	0	1	2	3	4	5	6	7	8	9
x_i	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
y_i	-21	-28.61	-37.52	-47.83	-59.62	-73	-88.05	-104.88	-123.59	-144.26

$0.2/3 \; * (-21 + 4(-28.61 - 47.83 - 73 - 104.88) + \; 2(-37.52 - 59.62 - 88.05 - 123.59) - 144.26) = \; -160.00$

$\Delta I \approx 0\%$

Программная реализация

На языке Kotlin

Класс, от которого наследуются все методы решения

```kotlin
abstract class IntegralSolveMethod(val leftBound: Double, val rightBound: Double, private
val accuracy: Double) {
   protected abstract val k: Int

   abstract fun calculate(func: Function, parts: Int) : Double

   fun solve(func: Function) : IntegralSolveResult{
      var n = START_N_PARTS
      do {
         val result2 = calculate(func, n)
         val result1 = calculate(func, n*2)
         n *= 2
      } while ((result2 - result1)/(2.0.pow(k) - 1) > accuracy)
      return IntegralSolveResult(calculate(func, n/2), n/2)
   }
}
```

Метод средних прямоугольников

```kotlin
class MiddleRectSolveMethod(leftBound: Double, rightBound: Double, accuracy:
Double)
    : IntegralSolveMethod(leftBound, rightBound, accuracy)
{
    override val k: Int = 2

    override fun calculate(func: Function, parts: Int): Double {
        val h: Double = (rightBound-leftBound) / parts
        var result = 0.0
        for (i in 0 until parts) {
            result += func.calculate(leftBound+i*h+h/2)
        }
        return result*h
    }
}
```

Метод трапеций

```
class TrapezeSolveMethod(leftBound: Double, rightBound: Double, accuracy: Double)
   : IntegralSolveMethod(leftBound, rightBound, accuracy)
{
   override val k: Int = 2

   override fun calculate(func: Function, parts: Int): Double {
      val h: Double = (rightBound-leftBound) / parts
      var sum = 0.0
      for (i in 1 until parts) {
         sum += func.calculate(leftBound+i*h)
      }
      return (2*sum+leftBound+rightBound) * h/2
   }
}
```

Метод Симпсона

```
class SimpsonSolveMethod(leftBound: Double, rightBound: Double, accuracy: Double)
   : IntegralSolveMethod(leftBound, rightBound, accuracy)
{
   override val k: Int = 4

   override fun calculate(func: Function, parts: Int): Double {
      val h: Double = (rightBound-leftBound) / parts
      var sum = func.calculate(leftBound) + func.calculate(rightBound)
      for (i in 1 until parts step 2) {
         sum += 4*func.calculate(leftBound+h*i)
      }
      for (i in 2 until parts step 2) {
         sum += 2*func.calculate(leftBound+h*i)
      }
      return sum * h/3
   }
}
```