Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №2

по курсу «Численные методы»

«Приближенное вычисление определенного интеграла»

Студент:

Группа: ИУ9-61Б

Преподаватель: Домрачева А.Б.

Москва 2025

1 Постановка задачи

Дано: Интеграл I

$$\int_a^b f(x)dx$$

где $f(x)$ – подынтегральная функция, непрерывная на отрезке $[a, b]$.

Найти: Значение интеграла

$$I^* \approx I$$

При заданной точности $\varepsilon < 0.001$.

Индивидуальный вариант (№4): $y = f(x)$ задана функцией: $y = 2x * \cos(\frac{x}{2})$ на отрезке $[0, \pi]$.

$$\int_0^\pi 2x * \cos(\frac{x}{2})dx = 4.5663706$$

2 Основные теоретические сведения

2.1 Метод центральных прямоугольников

Метод заключается в вычислении площади под графиком подынтегральной функции с помощью суммирования площадей прямоугольников. Ширина прямоугольника определяется шагом разбиения, то есть расстоянием между узлами интегрирования, высота определяется значением подынтегральной функции в узле интегрирования.

Пусть требуется определить значение интеграла функции $f(x)$ на отрезке $[a, b]$. Тогда отрезок разбивается на n равных отрезков длиной $h = \frac{b-a}{n}$. Получаем разбиение данного отрезка точками:

$$x_{i-0.5} = a + (i - 0.5)h \quad i = \overline{1, n}$$

Тогда приближенное значение интеграла на всем отрезке будет равно:

$$I^* = h\sum_{i=1}^{n} f(x_{i-0.5}) = h\sum_{i=1}^{n} f(a+(i-0.5)h)$$

Абсолютная погрешность приближения, полученного методом центральных прямоугольников, оценивается с помощью формулы

$$O(h^2) \le \frac{(b-a)\cdot M_2}{24},$$

где

$$M_2 = \max_{x\in[a,b]} |f''(x)|.$$

2.2 Метод трапеций

Метод заключается в вычислении площади под графиком подынтегральной функции с помощью суммирования площадей трапеций. Высота трапеции определяется шагом разбиения, то есть расстоянием между узлами интегрирования, основания трапеции определяются значениями подынтегральной функции в узлах интегрирования.

Пусть требуется определить значение интеграла функции $f(x)$ на отрезке $[a,b]$. Тогда отрезок разбивается на n равных отрезков длиной $h = \frac{b-a}{n}$. Получаем разбиение данного отрезка точками:

$$x_i = a + ih \quad i = \overline{1,n}$$

Тогда приближенное значение интеграла на всем отрезке будет равно:

$$I^* = h\left(\frac{f(a)+f(x_1)}{2} + \frac{f(x_1)+f(x_2)}{2} + \cdots \right.$$
$$\left. + \frac{f(x_{n-1})+f(b)}{2}\right) = h\left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right)$$

Абсолютная погрешность приближения, полученного методом трапеций, оценивается с помощью формулы

$$O(h^2) \leq \frac{(b-a)\cdot M_2}{12},$$

где

$$M_2 = \max_{x\in[a,b]} |f''(x)|.$$

2.3 Метод Симпсона

Метод заключается в приближении функции на отрезке $[a,b]$ интерполяционным многочленом второй степени функции $P_2(x)$:

$$P_2(x) = f_{i-0.5} + \frac{f_i - f_{i-1}}{h}(x_i - x_{i-0.5}) + \frac{f_i - 2f_{i-0.5} + f_{i-1}}{h^2/2}(x_i - x_{i-0.5})^2$$

Тогда приближенное значение интеграла на всем отрезке будет равно:

$$I^* = \frac{h}{6}\left(f(a) + f(b) + 4\sum_{i=1}^{n} f(x_{i-0.5}) + 2\sum_{i=1}^{n-1} f(x_i)\right)$$

Абсолютная погрешность приближения, полученного методом Симпсона, оценивается с помощью формулы

$$O(h^4) \leq \frac{(b-a)\cdot M_4}{2880},$$

где

$$M_4 = \max_{x\in[a,b]} |f^{IV}(x)|.$$

2.4 Уточнение значения интеграла по Ричардсону

$I \approx I_h^* + O(h^k)$, где k - порядок точности метода, I_h^* - приближенное значение интеграла, вычисленного с помощью метода с шагом h.

Для метода средних прямоугольников и метода трапеций $k = 2$.

Для метода Симпсона $k = 4$.

$$O(h^k) \approx ch^k, \text{ где } c - \text{некоторая константа}, h - \text{шаг}.$$

Будем считать, что вычисления проводятся без вычислительной погрешности, тогда можно записать строгое равенство $I = I_h^* + ch^k$ для шага h

$$I = I_h^* + c\left(\frac{h}{2}\right)^k \text{ для шага } \frac{h}{2}$$

Из равенств можно получить уточненное значение интеграла:

$$I = I_h^* + \frac{I_{h/2}^* - I_h^*}{2^k - 1}$$

Где значение R – уточнение по Ричардсону:

$$R = \frac{I_{h/2}^* - I_h^*}{2^k - 1}$$

Данная величина используется для компенсации методологической погрешности численных методов интегрирования. Для построения процедуры приближенного вычисления интеграла с заданной точностью ε, используется правило Рунге:

$$|R| < \varepsilon$$

3 Реализация

```
package main

import (
  "fmt"
  "math"
)

func f(x float64) float64 {
  // Вариант 4
  return 2 * x * math.Cos(x/2)
}

// Генерация точек на отрезке [a, b] с шагом h
func generatePoints(a, b float64, n int) []float64 {
```

```go
	h := (b - a) / float64(n)
	points := make([]float64, n+1)
	for i := 0; i <= n; i++ {
		points[i] = a + float64(i)*h
	}
	return points
}

// Метод прямоугольников
func rectangleMethod(a, b float64, n int) float64 {
	points := generatePoints(a, b, n)
	h := (b - a) / float64(n)
	sum := 0.0

	for i := 0; i < n; i++ {
		x := points[i] + h/2
		sum += f(x)
	}

	return sum * h
}

// Метод трапеций
func trapezoidMethod(a, b float64, n int) float64 {
	points := generatePoints(a, b, n)
	h := (b - a) / float64(n)
	sum := (f(a) + f(b)) / 2

	for i := 1; i < n; i++ {
		sum += f(points[i])
	}

	return sum * h
}

// Метод Симпсона
func simpsonMethod(a, b float64, n int) float64 {
	if n%2 != 0 {
		n++
	}

	points := generatePoints(a, b, n)
	h := (b - a) / float64(n)
	sum := f(a) + f(b)

	for i := 1; i < n; i++ {
```

```
    x := points[i]
    if i%2 == 0 {
      sum += 2 * f(x)
    } else {
      sum += 4 * f(x)
    }
  }

  return sum * h / 3
}

func applyMethod(method func(a, b float64, n int) float64, a, b float64, n int,
   epsilon float64, p int) (int, float64, float64) {
  methodResult := method(a, b, n)
  methodResult2 := method(a, b, 2*n)
  methodError := math.Abs(methodResult-methodResult2) / (math.Pow(2, float64(p))
    - 1)

  for methodError > epsilon {
    n *= 2
    methodResult = method(a, b, n)
    methodResult2 = method(a, b, 2*n)
    methodError = math.Abs(methodResult-methodResult2) / (math.Pow(2, float64(p)
   ) - 1)
  }
  return n, methodResult, methodError
}

func main() {
  a := 0.0
  b := math.Pi
  epsilon := 0.001
  pSimpson := 4
  pRect := 2
  pTrap := 2

  // Вычисление методом прямоугольников
  nRect := 10
  var rectResult, rectError float64
  nRect, rectResult, rectError = applyMethod(rectangleMethod, a, b, nRect,
   epsilon, pRect)

  // Вычисление методом трапеций
  nTrap := 10
  var trapResult, trapError float64
```

```
    nTrap, trapResult, trapError = applyMethod(trapezoidMethod, a, b, nTrap,
      epsilon, pTrap)

    // Метод Симпсона
    nSimpson := 10
    var simpsonResult, simpsonError float64
    nSimpson, simpsonResult, simpsonError = applyMethod(simpsonMethod, a, b,
      nSimpson, epsilon, pSimpson)

    // Вывод результатов
    fmt.Printf("\n%-20s%-16s%-16s%-16s\n", "Параметр", "rect", "trapec", "Simpson")
    fmt.Println("-----------------------------------------------------------------")
    fmt.Printf("%-20s%-16d%-16d%-16d\n", "n", nRect, nTrap, nSimpson)
    fmt.Printf("%-20s%-16.5f%-16.5f%-16.5f\n", "I(n)", rectResult, trapResult,
      simpsonResult)
    fmt.Printf("%-20s%-16.6f%-16.6f%-16.6f\n", "R", rectError, trapError,
      simpsonError)
    fmt.Printf("%-20s%-16.5f%-16.5f%-16.5f\n", "I(n)+R", rectResult+rectError,
      trapResult+trapError, simpsonResult+simpsonError)
}
```

Листинг 1: Методы вычисления определенного интеграла

4 Результаты

Для заданной функции $f(x)$ были вычислены приближенные значения определенных интегралов с помощью различных методов. Также для каждого метода было вычислено уточнение значения интеграла по Ричардсону. Результаты работы программы представлены в таблице 1.

Таблица 1: Сравнение методов численного интегрирования

Параметр	Прямоугольники	Трапеции	Симпсон
Число разбиений, n	32	64	4
Значение интеграла, $I(n)$	4,56844	4,56534	4,57133
Погрешность, R	0,000516	0,000258	0,000310
Уточненное значение, $I(n) + R$	4,56895	4,56560	4,57164

5 Вывод

По итогам лабораторной работы можно сделать следующие выводы:

- Метод Симпсона продемонстрировал наибольшую эффективность, достигая требуемой точности при минимальном количестве разбиений благодаря высокому порядку сходимости

- Методы прямоугольников и трапеций показали сопоставимую точность, но потребовали значительно большего числа разбиений для её достижения

- Применение уточнения по Ричардсону позволило повысить точность вычислений для всех рассматриваемых методов

Таким образом, для интегрирования гладких функций наиболее предпочтительным является метод Симпсона, как обеспечивающий оптимальное сочетание точности и вычислительной эффективности. Полученные результаты подтверждают теоретические положения о скорости сходимости различных методов численного интегрирования.