Университет ИТМО, факультет ПИиКТ

Лабораторная работа №2 по

“Вычислительная математика”

Вариант: метод прямоугольников

Выполнил: Яремко Р. О.

Группа: P3210

Преподаватель: Перл О.В.

Санкт-Петербург

2020г.

Задание

Пользователь выбирает функцию, интеграл которой он хочет вычислить (3-5 функций), из тех, которые предлагает программа.

В численный метод должен быть передан параметр-агрегат на подпрограмму вычисления значения функции в точке x.

Пользователь задает пределы интегрирования и точность.

Если нижний предел интегрирования >= верхнего предела - интеграл должен считаться корректно!

В результате должны получить:

- значение интеграла
- количество разбиений, на которое пришлось разбить
- полученную погрешность

Для оценки погрешности использовать оценку Рунге.

Код программы

```
package main

import(
  "math"
  "fmt"
  "errors"
  "os"
)

func square(x float64) float64 {
  return x*x
}

func cube(x float64) float64 {
  return x*x*x
}

func calculateError(a,b,e float64) (int, float64, error) {
  if e == 0.0 {
          return 0, 0, errors.New("Деление на 0 или символы")
  }
  floatN := (b-a)/math.Pow(e, 0.25)
```

```
  n := int(math.Ceil(floatN))
  newerr := math.Pow(((b-a)/float64(n)),4)
  return n, newerr, nil
}

func integrate(a,b,side float64, n int, f func(float64) float64) (float64, float64) {
  h := (b-a)/float64(n)
  dh := (b-a)/float64(n*2)
  var result float64 = 0
  var dresult float64 = 0
  for i := 0; i < n; i++ {
        result += f(a+h*(float64(i)+side))
  }
  result *= h
  for i := 0; i < 2*n; i++ {
        dresult += f(a+dh*(float64(i)+side))
  }
  dresult *= dh
  runge := (math.Abs(result-dresult))/2
  return result, runge
}

func main(){
  var negative bool = false
  var a, b, e, sideInt float64
  var mathFunc, side string
  var f func(float64) float64
  fmt.Println("Выберите функцию (default: square)")
  fmt.Println("square: x^2")
  fmt.Println("cube: x^3")
  fmt.Println("cosinus: cos(x)")
  fmt.Println("sinus: sin(x)")
  fmt.Print("Ваш выбор: ")
  fmt.Scan(&mathFunc)
  switch mathFunc {
  case "cube":
        f = cube
  case "square":
        f = square
  case "cosinus":
        f = math.Cos
  case "sinus":
        f = math.Sin
  default:
        f = square
  }
  fmt.Println("Выберите сторону (default: center)")
  fmt.Println("l: left")
```

```
  fmt.Println("r: right")
  fmt.Println("c: center")
  fmt.Print("Ваш выбор: ")
  fmt.Scan(&side)
  switch side {
  case "l":
        sideInt = 0
  case "c":
        sideInt = 0.5
  case "r":
        sideInt = 1
  default:
        sideInt = 0.5
  }
  fmt.Print("Введите нижний и верхний порог: ")
  fmt.Scan(&a, &b)
  if b < a {
        negative = true
        c := b
        b = a
        a = c
  }
  fmt.Print("Введите погрешность: ")
  fmt.Scan(&e)
  n, newerr, err := calculateError(a,b,e)
  if err != nil {
        fmt.Println(err)
        os.Exit(1)
  }
  fmt.Println()
  result, runge := integrate(a,b,sideInt,n,f)
  if math.IsNaN(result) {
        result, newerr, runge = 0,0,0
  }
  if negative {
        result = -result
  }
  fmt.Printf("Результат: %f\n", result)
  fmt.Printf("Количество разбиений: %d\n", n)
  fmt.Printf("Погрешность: %f\n", newerr)
  fmt.Printf("Условие Рунге: %f ", runge)
  if runge < math.E {
        fmt.Println("< e")
  } else {
        fmt.Println("> e")
  }
}
```

Тестовые данные

- Тест 1

  go run riemannSum.go
  Выберите функцию (default: square)
  square: x^2
  cube: x^3
  cosinus: cos(x)
  sinus: sin(x)
  Ваш выбор: cosinus
  Выберите сторону (default: center)
  l: left
  r: right
  c: center
  Ваш выбор: c
  Введите нижний и верхний порог: 1 10
  Введите погрешность: 0.000000001

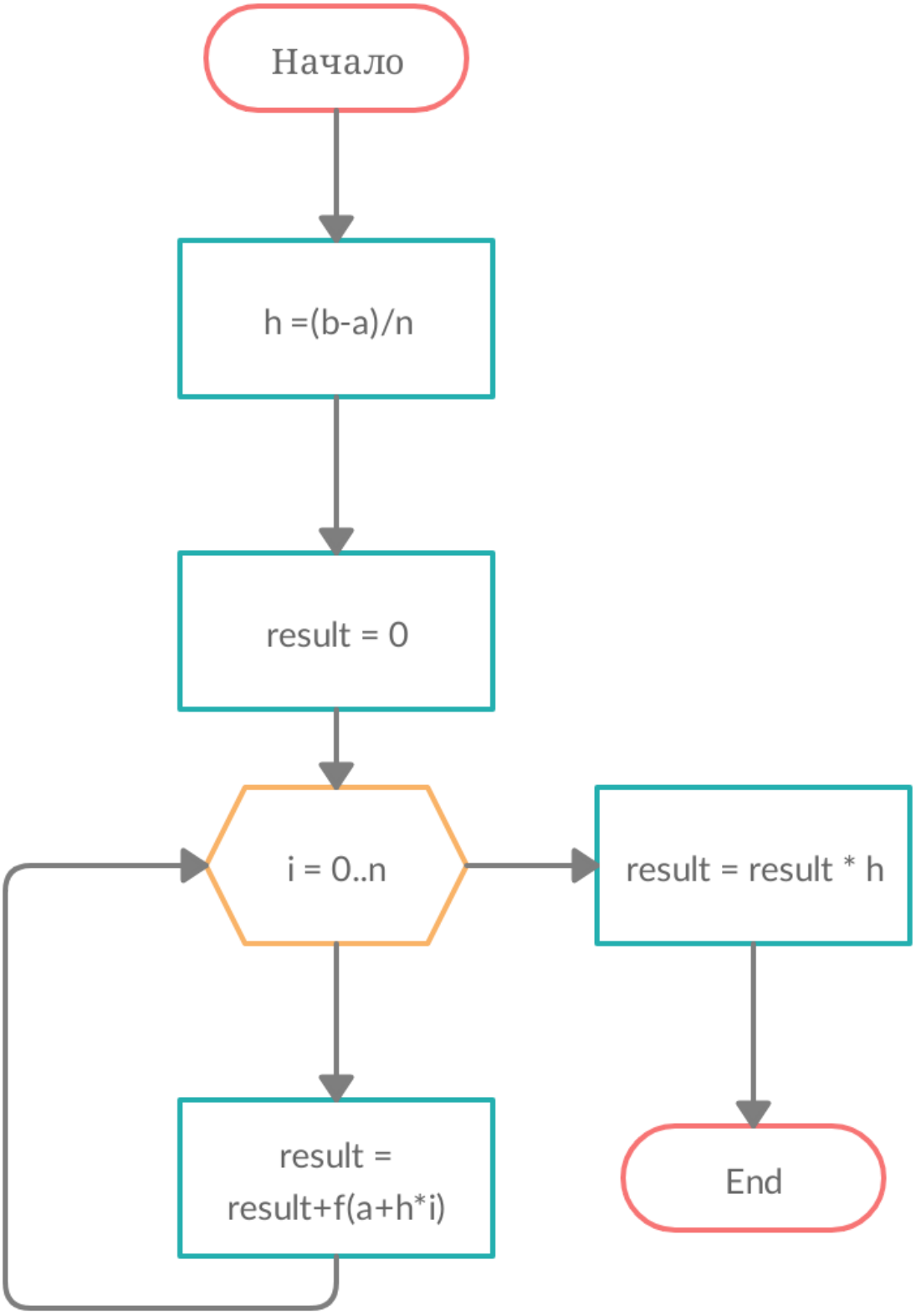
  Результат: -1.385494
  Количество разбиений: 1601
  Погрешность: 0.000000
  Условие Рунге: 0.000001 < e

- Тест 2

  go run riemannSum.go
  Выберите функцию (default: square)
  square: x^2
  cube: x^3
  cosinus: cos(x)
  sinus: sin(x)
  Ваш выбор: cube
  Выберите сторону (default: center)
  l: left
  r: right
  c: center
  Ваш выбор: c
  Введите нижний и верхний порог: 0 10
  Введите погрешность: 0.00001

  Результат: 2499.960548
  Количество разбиений: 178
  Погрешность: 0.000010
  Условие Рунге: 0.014795 < e

Блок-схема


Начало
h =(b-a)/n
result = 0
i = 0..n
result = result * h
result = result+f(a+h*i)
End

Вывод

Преимуществом данного метода является простота его реализации. Минусом является его невысокая точность, чтобы добиться более высокой точности, надо увеличивать количество разбиений начиная от пары тысяч.

У метода есть 3 способа реализации, а именно: левый метод, правый метод и средний метод. Названия говорят сами за себя, в зависимости от выбранного метода построение “прямоугольников” будет заканчиваться при достижении графика функции соответствующей стороны. Стоит отметить, что, исходя из лабораторной работы, самым точным из методов является средний