

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р. Е. АЛЕКСЕЕВА

Кафедра «Прикладная математика»

Отчет

Лабораторная работа №2
по дисциплине «Численные Методы»

Тема: «Численное интегрирование»

Студент

(Подпись) Шохов М.Е.
(Фамилия, И., О.)

17-ПМ 31.10.19
(Группа) (Дата сдачи)

Проверила

(Подпись) Талалушкина Л.В.
(Фамилия, И., О.)

Отчет защищен «__» _____ 2019г.

с оценкой _____

Нижний Новгород, 2019

Оглавление

1.	Цель работы.....	3
2.	Пояснение к заданию.....	4
3.	Описание алгоритма.....	6
4.	Реализация на языке C++.....	7
5.	Реализация на языке Python	11
6.	Реализация на языке Fortran	14
7.	Реализация на языке Matlab	19
8.	Результат выполнения программ.....	22
9.	Вывод.....	25

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		2
№.		Ф.И.О.	Подп.	Дата		

1. Цель работы

Изучить и программно реализовать методы численного интегрирования: левых прямоугольников, правых прямоугольников, средних прямоугольников, трапеций, Симпсона, на языках высокого уровня (C, Python, Matlab, Fortran), а так же сравнить скорость вычисления этих методов.

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		3

2. Пояснение к заданию

1. Метод прямоугольников — метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на многочлен нулевой степени, то есть константу, на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота — значением подынтегральной функции в этих узлах. Алгебраический порядок точности равен 0. (Для формулы средних прямоугольников равен 1).

Если отрезок $[a, b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по:

1. **Формуле левых прямоугольников:** $\int_a^b f(x) dx \approx f(a)(b - a).$
2. **Формуле правых прямоугольников:** $\int_a^b f(x) dx \approx f(b)(b - a).$
3. **Формуле прямоугольников (средних):** $\int_a^b f(x) dx \approx f\left(\frac{a + b}{2}\right)(b - a).$

В случае разбиения отрезка интегрирования на n элементарных отрезков приведённые выше формулы применяются на каждом из этих элементарных отрезков между двумя соседними узлами. В результате, получаются составные квадратурные формулы:

1. Для **левых** прямоугольников: $\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} f(x_i)(x_{i+1} - x_i).$
2. Для **правых** прямоугольников: $\int_a^b f(x) dx \approx \sum_{i=1}^n f(x_i)(x_i - x_{i-1}).$
3. Для **средних** прямоугольников:
$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right)(x_{i+1} - x_i) = \sum_{i=1}^n f\left(\frac{x_{i-1} + x_i}{2}\right)(x_i - x_{i-1}).$$

2. Метод трапеций — метод численного интегрирования функции одной переменной, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию. Площадь под графиком функции аппроксимируется прямоугольными трапециями. Алгебраический порядок точности равен 1.

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		4

Если отрезок $[a, b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по формуле:

$$\int_a^b f(x) dx = \frac{f(a) + f(b)}{2}(b - a) + E(f), \quad E(f) = -\frac{f''(\xi)}{12}(b - a)^3.$$

Это простое применение формулы для площади трапеции — произведение полусуммы оснований, которыми в данном случае являются значения функции в крайних точках отрезка, на высоту (длину отрезка интегрирования).

Если отрезок $[a, b]$ разбивается узлами интегрирования и на каждом из элементарных отрезков применяется формула трапеций, то суммирование даст составную формулу трапеций:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2}(x_{i+1} - x_i)$$

$$x_j = a + jh, h = (b - a)/N, N\text{—четное}$$

3. Формула Симпсона относится к приёмам численного интегрирования. Получила название в честь британского математика Томаса Симпсона (1710—1761).

Суть метода заключается в приближении подынтегральной функции на отрезке $[a, b]$ интерполяционным многочленом второй степени $p_2(x)$, то есть приближение графика функции на отрезке параболой. Метод Симпсона имеет порядок погрешности 4 и алгебраический порядок точности 3.

Формулой Симпсона называется интеграл от интерполяционного многочлена второй степени на отрезке $[a, b]$:

$$\int_a^b f(x) dx \approx \int_a^b p_2(x) dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Где $f(a)$, $f((a+b)/2)$ и $f(b)$ — значения функции в соответствующих точках (на концах отрезка и в его середине).

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		5

3. Описание алгоритма

Основные функции:

- `func()` – возвращает значение функции в точке x ,
- `simpson()` – выполняет численное интегрирование методом Симпсона для заданного отрезка, разбитого на n узлов, и возвращает значение интеграла,
- `trapeze()` - выполняет численное интегрирование методом трапеций для заданного отрезка, разбитого на n узлов, и возвращает значение интеграла,
- `leftRect()` - выполняет численное интегрирование методом левых прямоугольников для заданного отрезка, разбитого на n узлов, и возвращает значение интеграла,
- `rightRect()` - выполняет численное интегрирование методом правых прямоугольников для заданного отрезка, разбитого на n узлов, и возвращает значение интеграла,
- `centralRect()` - выполняет численное интегрирование методом средних прямоугольников для заданного отрезка, разбитого на n узлов, и возвращает значение интеграла
- `checkForCondition()` – выполняет вызов одного из методов численного интегрирования и осуществляет проверку на точность,
- `calculateIntegral()` – выполняет вызов функции `checkForCondition()` для каждого из методов численного интегрирования.

В функции `main()` задаются границы отрезка интегрирования и две точности: $eps1 = 0.0001$, $eps2 = 0.00001$. После этого осуществляется вызов функций `calculateIntegral()` для заданных точностей.

Основные элементы:

- a – левая граница отрезка интегрирования,
- b – правая граница отрезка интегрирования,
- $eps1$ – точность, соответствующая 0.0001 ,
- $eps2$ – точность, соответствующая 0.00001 ,
- n – число узлов в отрезке интегрирования,
- $res1$ – значение интеграла на текущей итерации для текущего n ,
- $res2$ – значение интеграла на предыдущей итерации для предыдущего n ,
- `type` – текущий метод интегрирования.

Вывод программы: границы отрезка интегрирования, текущая точность, численное значение каждого метода интегрирования и n для которого в рамках метода выполнилось условие точности. Такой вывод осуществляется два раза.

Исходная функция		
№		Отрезок
15	$y(x) = \frac{1}{e^x + e^{-x}}$	[0;1]

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		6

4. Реализация на языке C++

Реализация на языке C++

```
#include <iostream>
#include <cmath>

using namespace std;

double func(double point)
{
    return ( 1/( exp(point) + exp(-point) ));
}

double simpson(int n,double a,double b)
{
    double h = (b - a)/n;
    double f0 = func(a);
    double fn = func(b);
    double firstSum = 0;
    double secondSum = 0;
    int m = int(n/2);

    double* xTmp;
    xTmp = new double[n+1];

    for (int i = 0; i < n+1; i++)
        xTmp[i] = (a + h*i);

    for (int i = 1; i < m; i++)
        firstSum += func(xTmp[2*i]);

    for (int i = 1; i < m+1; i++)
        secondSum += func(xTmp[2*i-1]);

    delete [] xTmp;
    return (h/3)*(f0 + 2*firstSum + 4*secondSum + fn);
}

double trapeze(int n,double a,double b)
{
    double h = (b - a)/n;
    double f0 = func(a);
    double fn = func(b);
    double sum = 0;

    double* xTmp;
    xTmp = new double[n+1];

    for (int i = 0; i < n+1; i++)
        xTmp[i] = (a + h*i);

    for (int i = 1; i < n; i++)
        sum += func(xTmp[i]);

    delete [] xTmp;
    return (h/2)*(f0 + 2*sum + fn);
}
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		7

Реализация на языке C++

```
double leftRect(int n,double a,double b)
{
    double h = (b - a)/n;
    double sum = 0;

    double* xTmp;
    xTmp = new double[n+1];

    for (int i = 0; i < n+1; i++)
        xTmp[i] = (a + h*i);

    for (int i = 0; i < n; i++)
        sum += func(xTmp[i]);

    delete [] xTmp;
    return h * sum;
}

double rightRect(int n,double a,double b)
{
    double h = (b - a)/n;
    double sum = 0;

    double* xTmp;
    xTmp = new double[n+1];

    for (int i = 0; i < n+1; i++)
        xTmp[i] = (a + h*i);

    for (int i = 1; i < n+1; i++)
        sum += func(xTmp[i]);

    delete [] xTmp;
    return h * sum;
}

double centralRect(int n,double a,double b)
{
    double f0 = func(a)/2;
    double fn = func(b)/2;
    double h = (b - a)/n;
    double sum = 0;

    double* xTmp;
    xTmp = new double[n+1];

    for (int i = 0; i < n+1; i++)
        xTmp[i] = (a + h*i);

    for (int i = 1; i < n; i++)
        sum += func(xTmp[i]);

    delete [] xTmp;
    return h * (f0 + sum + fn);
}
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		8
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке C++

```
double checkForCondition(string type,double a,double b, double eps)
{
    int n = 2;
    double res2 = 0;
    double res1 = 0;
    bool tmp = true;

    while (tmp)
    {
        if (type == "simpson")
            res1 = simpson(n,a,b);
        else if (type == "trapeze")
            res1 = trapeze(n,a,b);
        else if (type == "leftRect")
            res1 = leftRect(n,a,b);
        else if (type == "rightRect")
            res1 = rightRect(n,a,b);
        else if (type == "centralRect")
            res1 = centralRect(n,a,b);

        double diff = abs(res1 - res2);

        res2 = res1;

        type == "simpson" ? n += 2 : n += 1;
//        if (type == "simpson")
//            n += 2;
//        else
//            n += 1;

        if (eps*15 >= diff && type == "simpson")
        {
            cout << "n = " << n-2 << endl;
            tmp = false;
        } else if (eps*3 >= diff && type != "simpson"){
            cout << "n = " << n-1 << endl;
            tmp = false;
        }
    }
    return res1;
}

void calculateIntegral(double a,double b, double eps)
{
    cout << "Simpson method: " << checkForCondition("simpson",a,b,eps) << endl;
    cout << "Trapeze method: " << checkForCondition("trapeze",a,b,eps) << endl;
    cout << "Left rectangular method: " << checkForCondition("leftRect",a,b,eps) << endl;
    cout << "Right rectangular method: " << checkForCondition("rightRect",a,b,eps) << endl;
    cout << "Central rectangular method: " << checkForCondition("centralRect",a,b,eps) << endl;
    cout << endl;
}
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		9
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке C++

```
int main()
{
    double a,b,eps1,eps2;
    a = 0.0;
    b = 1.0;
    eps1 = 0.0001;
    eps2 = 0.00001;

    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    cout << "eps = " << eps1 << endl;
    calculateIntegral(a,b,eps1);
    cout << endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    cout << "eps = " << eps2 << endl;
    calculateIntegral(a,b,eps2);
}
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		10

5. Реализация на языке Python

Реализация на языке Python

```
import numpy as np
import math
from sys import argv

def func(point) :
    return ( 1/( math.exp(point) + math.exp(-point) ))

def simpson(*args) :
    n,a,b = args
    h = (b - a)/n
    f0 = func(a)
    fn = func(b)
    firstSum = 0
    secondSum = 0
    m = int(n/2)
    xTmp = []
    for i in range(0, n+1) :
        xTmp.append(a + h*i)

    for i in range(1,m) :
        firstSum += func(xTmp[2*i])

    for i in range(1,m+1) :
        secondSum += func(xTmp[2*i-1])

    return (h/3)*(f0 + 2*firstSum + 4*secondSum + fn)

def trapeze(*args) :
    n,a,b = args
    h = (b - a)/n
    f0 = func(a)
    fn = func(b)
    sum = 0
    xTmp = []
    for i in range(0, n+1) :
        xTmp.append(a + h*i)

    for i in range(1,n) :
        sum += func(xTmp[i])

    return (h/2)*(f0 + 2*sum + fn)

def leftRect(*args) :
    n,a,b = args
    h = (b - a)/n
    sum = 0
    xTmp = []
    for i in range(0, n+1) :
        xTmp.append(a + h*i)

    for i in range(0,n) :
        sum += func(xTmp[i])

    return h * sum
```

Реализация на языке Python

```
def rightRect(*args) :
    n,a,b = args
    h = (b - a)/n
    sum = 0

    xTmp = []
    for i in range(0, n+1) :
        xTmp.append(a + h*i)

    for i in range(1,n+1) :
        sum += func(xTmp[i])

    return h * sum

def centralRect(*args) :
    n,a,b = args
    f0 = func(a)/2
    fn = func(b)/2
    h = (b - a)/n
    sum = 0

    xTmp = []
    for i in range(0, n+1) :
        xTmp.append(a + h*i)

    for i in range(1,n) :
        sum += func(xTmp[i])

    return h * (f0 + sum + fn)

def checkForCondition(*args) :
    type,a,b,eps = args

    n = 2
    res2 = 0
    res1 = 0
    tmp = 1
    while tmp == 1 :
        if type == "simpson" :
            res1 = simpson(n,a,b)
        elif type == "trapeze" :
            res1 = trapeze(n,a,b)
        elif type == "leftRect" :
            res1 = leftRect(n,a,b)
        elif type == "rightRect" :
            res1 = rightRect(n,a,b)
        elif type == "centralRect" :
            res1 = centralRect(n,a,b)

        diff = abs(res1 - res2)

        res2 = res1
        n = n + 2 if type == "simpson" else n + 1
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		12

Реализация на языке Python

```

    if eps*15 >= diff and type == "simpson" :
        print(f"n = {n-2}",end=" ")
        tmp = 0
    elif eps*3 >= diff and type != "simpson" :
        print(f"n = {n-1}",end=" ")
        tmp = 0

    return res1

def calculateIntegral(*args) :
    a,b,eps = args

    print(f"Simpson method: {checkForCondition('simpson',a,b,eps)}")
    print(f"Trapeze method: {checkForCondition('trapeze',a,b,eps)}")
    print(f"Left rectangular method:
{checkForCondition('leftRect',a,b,eps)}")
    print(f"Right rectangular method:
{checkForCondition('rightRect',a,b,eps)}")
    print(f"Central rectangular method:
{checkForCondition('centralRect',a,b,eps)}")
    print("")

if __name__ == "__main__" :
    a = 0.0
    b = 1.0
    eps1 = 0.0001
    eps2 = 0.00001

    print(f"a = {a}",f"b = {b}",f"eps = {eps1}",sep="\n")
    calculateIntegral(a,b,eps1)
    print("\n")
    print(f"a = {a}",f"b = {b}",f"eps = {eps2}",sep="\n")
    calculateIntegral(a,b,eps2)

```

6. Реализация на языке Fortran

Реализация на языке Fortran

```

real function func(point)
  real :: point;
  func = ( 1/( EXP(point) + EXP(-point) ));
end function func

module integral
  contains
  real function simpson(n, a, b)
    integer n;
    real b, a, buff;
    real, dimension (:), allocatable :: xTmp;
    real h, secondSum, firstSum
    firstSum = 0.0;
    secondSum = 0.0;
    buff=0;
    m = n/2;
    h = (b - a)/n;

    allocate ( xTmp(n) );
    do i = 0, n
      xTmp(i) = (a + h*(i));
    end do

    do i = 1, m-1
      firstSum = firstSum + func(xTmp(2*(i)));
    end do

    do i = 1, m
      secondSum = secondSum + func(xTmp(2*(i)-1));
    end do

    buff=(h/3)*(func(xTmp(0)) + 2*firstSum + 4*secondSum +
func(xTmp(n)));
    deallocate (xTmp);
    simpson = buff;
  end function simpson

  real function trapeze(n,a,b)
    integer n;
    real a,b;
    real, dimension (:), allocatable :: xTmp;
    real h, msum
    msum = 0

    h = (b - a)/n;

    allocate( xTmp(n) );

    do i = 0, n
      xTmp(i) = (a + h*(i));
    end do
  
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		14

Реализация на языке Fortran

```

do i = 1, n-1
    msum = msum + func(xTmp(i));
end do
trapeze = (h/2)*(func(xTmp(0)) + 2*msum + func(xTmp(n)));
deallocate(xTmp);
end function trapeze

real function leftRect(n,a,b)
integer n;
real a,b,h,msum;

real, dimension (:), allocatable :: xTmp;
msum = 0.0;

h = (b - a)/n;
allocate( xTmp(n) );

do i = 0, n
    xTmp(i) = (a + h*(i));
end do

do i = 0, n-1
    msum = msum + func(xTmp(i));
end do

deallocate(xTmp)
leftRect = h * msum;
end function leftRect

real function rightRect(n,a,b)
integer n;
real a,b,h,msum;

real, dimension (:), allocatable :: xTmp;
msum = 0.0;

h = (b - a)/n;
allocate( xTmp(n) );

do i = 0, n
    xTmp(i) = (a + h*(i));
end do

do i = 1, n
    msum = msum + func(xTmp(i));
end do

deallocate(xTmp);
rightRect = h * msum;
end function rightRect

```

Реализация на языке Fortran

```

real function centralRect(n,a,b)
  integer n;
  real b, a, h, msum, buff;

  real, dimension (:), allocatable :: xTmp;
  msum = 0.0;

  h = (b - a)/n;

  allocate( xTmp(n) );

  do i = 0, n
    xTmp(i) = (a + h*(i));
  end do

  do i = 1, n-1
    msum = msum + func(xTmp(i));
  end do
  buff=h * (func(xTmp(0))/2 + msum + func(xTmp(n))/2);
  deallocate(xTmp);
  centralRect = buff
end function centralRect

real function checkForCondition(mtype, a, b, eps)
  integer mtype, n, tmp;
  real b, a, eps, res1, res2, diff;
  n=1
  res1=0
  res2=0
  diff=0
  tmp=1

  do while (tmp == 1)
    !do while (eps < diff)
      if (mtype == 1) then
        n=n+1
        res1 = simpson(n,a,b);
        diff = ABS(res2 - res1);
        if ((eps*15) >= diff) then
          print *,n
          tmp=0;
        end if
      end if

      if (mtype == 2) then
        res1 = trapeze(n,a,b);
        diff = ABS(res2 - res1);
        if (eps*3 >= diff) then
          print *,n
          tmp=0;
        end if
      end if
    end if
  end if

```


Реализация на языке Fortran

```

        if (mtype == 3) then
            res1 = leftRect(n,a,b);
            diff = ABS(res2 - res1);
            if (eps*3 >= diff) then
                print *,n
                tmp=0;
            end if
        end if

        if (mtype == 4) then
            res1 = rightRect(n,a,b);
            diff = ABS(res2 - res1);
            if (eps*3 >= diff) then
                print *,n
                tmp=0;
            end if
        end if

        if (mtype == 5) then
            res1 = centralRect(n,a,b);
            diff = ABS(res2 - res1);
            if (eps*3 >= diff) then
                print *,n
                tmp=0;
            end if
        end if

        res2 = res1;
        n = n + 1;
    end do

    checkForCondition = res1;
end function checkForCondition

subroutine calculateIntegral(a, b, eps)
    real a, b, eps
    print *, "Simpson method: n =", checkForCondition(1,a,b,eps);
    print *, "Trapeze method: n = ", checkForCondition(2,a,b,eps);
    print *, "Left rectangular method: n = ",
checkForCondition(3,a,b,eps);
    print *, "Right rectangular method: n = ",
checkForCondition(4,a,b,eps);
    print *, "Central rectangular method: n = ",
checkForCondition(5,a,b,eps);

    end subroutine calculateIntegral
end module integral

```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		17

Реализация на языке Fortran

```

program main
  use integral;
  real a,b,eps, eps1;

  a = 0.0
  b = 1.0
  eps = 0.0001
  eps1 = 0.00001

  print *, "a =", a
  print *, "b =", b
  print *, "eps =", 0.0001
  call calculateIntegral(a,b,eps);
  print *, " "
  print *, "a =", a
  print *, "b =", b
  print *, "eps =", 0.00001
  call calculateIntegral(a,b,eps1);
end

```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		18

7. Реализация на языке Matlab

Реализация на языке Matlab

```
function main()
    a = 0.0;
    b = 1.0;
    eps1 = 0.0001;
    eps2 = 0.00001;

    fprintf("a = %d \n", a);
    fprintf("b = %d \n", b);
    fprintf("eps = %d \n", eps1);
    calculateIntegral(a,b,eps1);
    fprintf("\n");
    fprintf("a = %d \n", a);
    fprintf("b = %d \n", b);
    fprintf("eps = %d \n", eps2);
    calculateIntegral(a,b,eps2);
end

function val = func(point)
    val = ( 1/( exp(point) + exp(-point) ) );
end

function val = simpson(n,a,b)
    h = (b - a)/n;
    f0 = func(a);
    fn = func(b);
    firstSum = 0.0;
    secondSum = 0.0;
    m = n/2;

    xTmp = [n+1];
    for i = 1: n+1
        xTmp(i) = ( a + h*(i-1) );
    end

    for i = 2: m
        firstSum = firstSum + func(xTmp(2*(i-1)+1) );
    end

    for i = 2: m+1
        secondSum = secondSum + func(xTmp(2*(i-1)-1+1) );
    end

    val = (h/3)*(f0 + 2*firstSum + 4*secondSum + fn);
end

function val = trapeze(n,a,b)
    h = (b - a)/n;
    f0 = func(a);
    fn = func(b);
    sum = 0;

    xTmp = [n+1];
    for i = 1: n+1
        xTmp(i) = ( a + h*(i-1) );
    end
end
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		19

Реализация на языке Matlab

```

    for i = 2: n
        sum = sum + func(xTmp(i));
    end

    val = (h/2)*(f0 + 2*sum + fn);
end

function val = leftRect(n,a,b)
    h = (b - a)/n;
    sum = 0;

    xTmp = [n+1];
    for i = 1: n+1
        xTmp(i) = ( a + h*(i-1) );
    end

    for i = 1: n
        sum = sum + func(xTmp(i));
    end

    val = h * sum;
end

function val = rightRect(n,a,b)
    h = (b - a)/n;
    sum = 0;

    xTmp = [n+1];
    for i = 1: n+1
        xTmp(i) = ( a + h*(i-1) );
    end

    for i = 2: n+1
        sum = sum + func(xTmp(i));
    end

    val = h * sum;
end

function val = centralRect(n,a,b)
    f0 = func(a)/2;
    fn = func(b)/2;
    h = (b - a)/n;
    sum = 0;

    xTmp = [n+1];
    for i = 1: n+1
        xTmp(i) = ( a + h*(i-1) );
    end

    for i = 2: n
        sum = sum + func(xTmp(i));
    end

    val = h * (f0 + sum + fn);
end

```

Реализация на языке Matlab

```
function val = checkForCondition(type,a,b,eps)
    n = 2;
    res2 = 0;
    res1 = 0;
    tmp = 1;

    while tmp == 1
        if type == "simpson"
            res1 = simpson(n,a,b);
        elseif type == "trapeze"
            res1 = trapeze(n,a,b);
        elseif type == "leftRect"
            res1 = leftRect(n,a,b);
        elseif type == "rightRect"
            res1 = rightRect(n,a,b);
        elseif type == "centralRect"
            res1 = centralRect(n,a,b);
        end

        diff = abs(res1 - res2);

        res2 = res1;
        if (type == "simpson"); n = n + 2; else; n = n + 1; end

        if (eps*15 >= diff && type == "simpson")
            fprintf("n = %d", n-2);
            tmp = 0;
        elseif (eps*3 >= diff && type ~= "simpson")
            fprintf("n = %d", n-1);
            tmp = 0;
        end
    end
    val = res1;
end

function calculateIntegral(a,b,eps)
    fprintf(" Simpson method: %f \n", checkForCondition('simpson',a,b,eps));
    fprintf(" Trapeze method: %f \n", checkForCondition('trapeze',a,b,eps));
    fprintf(" Left rectangular method: %f \n",
    checkForCondition('leftRect',a,b,eps));
    fprintf(" Right rectangular method: %f \n",
    checkForCondition('rightRect',a,b,eps));
    fprintf(" Central rectangular method: %f \n",
    checkForCondition('centralRect',a,b,eps));
end
```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		21

8. Результат выполнения программ

Результат программы на языке C++

```

Терминал
a = 0
b = 1
eps = 0.0001
Simpson method: n = 4
0.432893
Trapeze method: n = 6
0.432313
Left rectangular method: n = 18
0.437709
Right rectangular method: n = 18
0.427933
Central rectangular method: n = 6
0.432313

a = 0
b = 1
eps = 1e-05
Simpson method: n = 4
0.432893
Trapeze method: n = 12
0.432742
Left rectangular method: n = 55
0.434478
Right rectangular method: n = 55
0.431278
Central rectangular method: n = 12
0.432742

Для закрытия данного окна нажмите <ВВОД>...

```

Результат программы на языке Python

```

maxwell@maxwell-HP-Laptop-15-bw0xx: ~/numerical method/LR2
maxwell@maxwell-HP-Laptop-15-bw0xx:~/numerical method/LR2$ python3 int.py
a = 0.0
b = 1.0
eps = 0.0001
n = 4 Simpson method: 0.43289271889974784
n = 6 Trapeze method: 0.43231309676413165
n = 18 Left rectangular method: 0.4377094002509576
n = 18 Right rectangular method: 0.4279331300749545
n = 6 Central rectangular method: 0.43231309676413165

a = 0.0
b = 1.0
eps = 1e-05
n = 4 Simpson method: 0.43289271889974784
n = 12 Trapeze method: 0.4327419056421383
n = 55 Left rectangular method: 0.4344776965951805
n = 55 Right rectangular method: 0.4312781899921249
n = 12 Central rectangular method: 0.4327419056421383

maxwell@maxwell-HP-Laptop-15-bw0xx:~/numerical method/LR2$ 

```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		22

Результат программы на языке Fortran

```

                                integral
a = 0.00000000
b = 1.00000000
eps = 9.99999975E-05
Simpson method: n =          4
0.432892740
Trapeze method: n =          6
0.432313085
Left rectangular method: n =      18
0.437709361
Right rectangular method: n =      18
0.427933097
Central rectangular method: n =      6
0.432313085

a = 0.00000000
b = 1.00000000
eps = 9.99999975E-06
Simpson method: n =          4
0.432892740
Trapeze method: n =         12
0.432741880
Left rectangular method: n =      55
0.434477627
Right rectangular method: n =      55
0.431278169
Central rectangular method: n =     12
0.432741880

Process returned 0 (0x0)   execution time : 0.005 s
Press ENTER to continue.

```

Результат программы на языке Matlab

```

Command Window
>>
>> main
a = 0
b = 1
eps = 1.000000e-04
n = 4 Simpson method: 0.432893
n = 6 Trapeze method: 0.432313
n = 18 Left rectangular method: 0.437709
n = 18 Right rectangular method: 0.427933
n = 6 Central rectangular method: 0.432313

a = 0
b = 1
eps = 1.000000e-05
n = 4 Simpson method: 0.432893
n = 12 Trapeze method: 0.432742
n = 55 Left rectangular method: 0.434478
n = 55 Right rectangular method: 0.431278
n = 12 Central rectangular method: 0.432742
fx >>

```

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		23

	Методы	C++	Python	Fortran	Matlab
eps = 0.0001	Симпсона	0.432893	0.43289271889974784	0.432892740	0.432893
	Трапеции	0.432313	0.43231309676413165	0.432313085	0.432313
	Левых прямоугольников	0.437709	0.4377094002509576	0.437709361	0.437709
	Правых прямоугольников	0.427933	0.4279331300749545	0.427933097	0.427933
	Средних прямоугольников	0.432313	0.43231309676413165	0.432313085	0.432313
eps = 0.00001	Симпсона	0.432893	0.43289271889974784	0.432892740	0.432893
	Трапеции	0.432742	0.4327419056421383	0.432741880	0.432742
	Левых прямоугольников	0.434478	0.4344776965951805	0.434477627	0.434478
	Правых прямоугольников	0.431278	0.4312781899921249	0.431278169	0.431278
	Средних прямоугольников	0.432742	0.4327419056421383	0.432741880	0.432742

9. Вывод

В ходе выполнения лабораторной работы №2 были изучены и программно реализованы методы интегрирования: левых прямоугольников, правых прямоугольников, средних прямоугольников, трапеций и Симпсона на языках высокого уровня (C++, Python, Fortran, Matlab). Так же было установлено, что наивысшая скорость вычисления у метода Симпсона, так как для достижения нужной точности было необходимо меньшее число узлов.

1	Вып.	Шохов М.Е.		4.12.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		4.12.19		
№.		Ф.И.О.	Подп.	Дата		25