МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Численное интегрирование

Лабораторная работа №5

по дисциплине «Вычислительная математика»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

Проверил Резак Е.В.

Хабаровск – 2020г.

**Задание**

Вычислить с заданной точностью ε.

Реализовать:

1) метод прямоугольников;

2) метод трапеций;

3) метод Симпсона.

Проанализировать эффективность методов для различных функций f(x) (линейная, квадратичная, другая – по варианту).

Заданная точность вычисления интеграла обеспечивается методом двойного пересчета.

Вариант 16:



**Листинг**

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Threading;

using System.Globalization;

using System.Windows.Forms.VisualStyles;

namespace Lab4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

void RefreshForm()

{

panel1.Invalidate();

}

void ClearForm()

{

Main.dots.Clear();

Main.lines.Clear();

Main.curves.Clear();

Main.couples = new double[1, 1];

Main.numOfCouples = 0;

Graphic.scale = 100;

label1.Text = "Импортируйте пары значений";

RefreshForm();

}

double RectangleMethod(double a, double b, double h)

{

double result = 0;

double N = (b - a) / h;

for(double i = 0; i < N; i++)

{

result += h \* Main.CountFunc(a + (i + 0.5) \* h);

}

return result;

}

double TrapMethod(double a, double b, double h)

{

double result = 0;

double N = (b - a) / h;

double X1, X2 = a;

for (double i = 1; i <= N; i++)

{

X1 = X2;

X2 = a + i \* h;

result += h \* (Main.CountFunc(X1) + Main.CountFunc(X2)) / 2;

}

return result;

}

double SimpsonMethod(double a, double b, double h)

{

double res = 0;

res += Main.CountFunc(a) + Main.CountFunc(b);

double sum = 0;

for(int i = 2; i < (b - a) / h; i += 2)

{

double x = a + h \* i;

sum += Main.CountFunc(x);

}

res += 2 \* sum;

sum = 0;

for(int i = 1; i < (b - a) / h; i += 2)

{

double x = a + h \* i;

sum += Main.CountFunc(x);

}

res += 4 \* sum;

res \*= h / 3;

return res;

}

private void importButton\_Click(object sender, EventArgs e)

{

string fileText = "";

Form2 importForm = new Form2();

importForm.Owner = this;

importForm.ShowDialog();

if (importForm.DialogResult == DialogResult.OK)

{

StreamReader file;

try

{

file = new StreamReader(Main.sourcePath);

fileText = file.ReadToEnd();

}

catch

{

MessageBox.Show("Введён неверный путь к файлу!");

return;

}

file.Close();

Main.sourcePath = "";

CultureInfo temp\_culture = Thread.CurrentThread.CurrentCulture;

Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture("en-US");

try

{

ClearForm();

string[] separators = {"\n", " ", "\t"};

string[] splittedText = fileText.Split(separators, StringSplitOptions.RemoveEmptyEntries);

//string text = "";

Main.numOfCouples = int.Parse(splittedText[0]);

Main.couples = new double[Main.numOfCouples, 2];

for (int i = 0; i < Main.numOfCouples; i++)

{

for (int j = 0; j < 2; j++)

{

//MessageBox.Show("+" + splittedText[1 + i \* 2 + j] + "+");

Main.couples[i, j] = double.Parse(splittedText[1 + i \* 2 + j]);

//text += Main.couples[i, j].ToString() + " ";

}

//text += '\n';

}

//MessageBox.Show(text);

}

catch

{

MessageBox.Show("Файл не прочитан");

}

Thread.CurrentThread.CurrentCulture = temp\_culture;

RefreshForm();

Graphic.GetDelta();

Graphic.ImportCouples();

}

}

private void showCouples\_Click(object sender, EventArgs e)

{

string text = Main.numOfCouples.ToString() + " пар:\n";

for (int i = 0; i < Main.numOfCouples; i++)

{

for (int j = 0; j < 2; j++)

{

text += Main.couples[i, j].ToString() + " ";

}

text += '\n';

}

MessageBox.Show(text);

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

Graphics graphics = panel1.CreateGraphics();

Pen pen = new Pen(Color.Black, 1);

graphics.DrawLine(pen, new Point(0, 250), new Point(500, 250));

graphics.DrawLine(pen, new Point(250, 0), new Point(250, 500));

int x = 250;

int y = 250;

while(x > 0)

{

x -= Graphic.scale;

graphics.DrawLine(pen, new Point(x, 240), new Point(x, 260));

}

x = 250;

while (x < 500)

{

x += Graphic.scale;

graphics.DrawLine(pen, new Point(x, 240), new Point(x, 260));

}

while (y > 0)

{

y -= Graphic.scale;

graphics.DrawLine(pen, new Point(240, y), new Point(260, y));

}

y = 250;

while (y < 500)

{

y += Graphic.scale;

graphics.DrawLine(pen, new Point(240, y), new Point(260, y));

}

foreach (Line i in Main.lines) // Отрисовка линий

{

graphics.DrawLine(new Pen(Color.Red, 2), new Point(i.firstPoint.x, i.firstPoint.y), new Point(i.secondPoint.x, i.secondPoint.y));

}

foreach (Dot i in Main.dots) // Отрисовка точек

{

graphics.DrawRectangle(new Pen(Color.Green, 2), i.x - 1, i.y - 1, 3, 3);

}

foreach (Curve i in Main.curves) // Отрисовка кривых

{

graphics.DrawCurve(new Pen(Color.Red, 1), i.ConvertToPoints());

}

}

private void refreshButton\_Click(object sender, EventArgs e)

{

RefreshForm();

}

private void clearButton\_Click(object sender, EventArgs e)

{

ClearForm();

}

private void shouDotsButton\_Click(object sender, EventArgs e)

{

string result = "";

int i = 1;

foreach(Dot x in Main.dots)

{

result += i.ToString() + ": " + x.x.ToString() + "/" + x.y.ToString() + "\n";

i++;

}

MessageBox.Show(result);

}

private void countButton\_Click(object sender, EventArgs e)

{

try

{

label1.Text = Main.CountFunc(double.Parse(textBox1.Text)).ToString();

}

catch

{

MessageBox.Show("Введено неверное число!");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void rectangleMethodButton\_Click(object sender, EventArgs e)

{

double a, b, h;

try

{

a = double.Parse(textBoxA.Text);

b = double.Parse(textBoxB.Text);

h = double.Parse(textBoxN.Text);

label1.Text = "S: " + RectangleMethod(a, b, h).ToString();

}

catch

{

MessageBox.Show("Введены неверные значения!");

}

}

private void button2\_Click(object sender, EventArgs e)

{

Main.couples = new double[9, 2];

Main.numOfCouples = 9;

int k = 0;

for(double i = -1.5; i < 3; i += 0.5)

{

Main.couples[k, 0] = i;

Main.couples[k, 1] = Main.CountFunc(i);

k++;

}

Graphic.GetDelta();

Graphic.ImportCouples();

Curve curve = new Curve();

foreach(Dot i in Main.dots)

{

curve.dots.Add(i);

}

Main.curves.Add(curve);

RefreshForm();

}

private void integralButton\_Click(object sender, EventArgs e)

{

double a, b, h;

try

{

a = double.Parse(textBoxA.Text);

b = double.Parse(textBoxB.Text);

h = double.Parse(textBoxN.Text);

label1.Text = "S: " + TrapMethod(a, b, h).ToString();

}

catch

{

MessageBox.Show("Введены неверные значения!");

}

}

private void simpsonButton\_Click(object sender, EventArgs e)

{

double a, b, h;

try

{

a = double.Parse(textBoxA.Text);

b = double.Parse(textBoxB.Text);

h = double.Parse(textBoxN.Text);

label1.Text = "S: " + SimpsonMethod(a, b, h).ToString();

}

catch

{

MessageBox.Show("Введены неверные значения!");

}

}

}

}

**Main.cs**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab4

{

static class Main

{

public static string sourcePath;

public static int numOfCouples = 0;

public static double[,] couples; // Пары значений couples[\*, 0] - x и couples[\*, 1] - y

public static double convertCoef = 1;

public static List<Dot> dots = new List<Dot>();

public static List<Line> lines = new List<Line>();

public static List<Curve> curves = new List<Curve>();

public static double GetMinX()

{

double res = couples[0, 0];

for(int i = 1; i < numOfCouples; i++)

{

if (couples[i, 0] < res)

res = couples[i, 0];

}

return res;

}

public static double GetMaxX()

{

double res = couples[0, 0];

for (int i = 1; i < numOfCouples; i++)

{

if (couples[i, 0] > res)

res = couples[i, 0];

}

return res;

}

public static int Factorial(int k)

{

if (k == 0)

return 1;

else if (k == 1)

return 1;

else if (k > 0)

{

int result = 1;

for (int i = 1; i <= k; i++)

{

result \*= i;

}

return result;

}

else

return 0;

}

public static double CountFunc(double x)

{

double result = Math.Sin(Math.Pow(x + 3, 2) / 2) / 2 + Math.Log(x + 2) / 2 - 1;

return result;

}

}

class Dot

{

public int x, y;

public Dot(int \_x, int \_y)

{

x = \_x;

y = \_y;

}

public Dot(double \_x, double \_y)

{

\_x = \_x \* Graphic.scale;

if (\_x > 0)

{

\_x = Math.Abs(\_x) + Graphic.SCREEN\_HALF;

}

else

{

\_x += Graphic.SCREEN\_HALF;

}

\_y = \_y \* Graphic.scale;

if (\_y > 0)

{

\_y = Graphic.SCREEN\_HALF - \_y;

}

else

{

\_y = Math.Abs(\_y) + Graphic.SCREEN\_HALF;

}

x = (int)\_x;

y = (int)\_y;

}

}

class Line

{

public Dot firstPoint, secondPoint;

public Line(Dot \_firstPoint, Dot \_secondPoint)

{

firstPoint = \_firstPoint;

secondPoint = \_secondPoint;

}

public Line(int \_x1, int \_y1, int \_x2, int \_y2)

{

firstPoint = new Dot(\_x1, \_y1);

secondPoint = new Dot(\_x2, \_y2);

}

}

class Curve

{

public List<Dot> dots = new List<Dot>();

public Point[] ConvertToPoints()

{

Point[] points = new Point[dots.Count];

for(int i = 0; i < dots.Count; i++)

{

points[i] = new Point(dots[i].x, dots[i].y);

}

return points;

}

}

static class Graphic

{

public static int scale = 100;

public const int SCREEN\_HALF = 250;

public static void GetDelta()

{

double max = 0;

for(int i = 0; i < Main.numOfCouples; i++)

{

for(int j = 0; j < 2; j++)

{

if (Math.Abs(Main.couples[i, j]) > max)

max = Math.Abs(Main.couples[i, j]);

}

}

max = (int)max + 1;

max \*= 2;

scale = (int)(500 / max);

}

public static void ImportCouples()

{

for (int i = 0; i < Main.numOfCouples; i++)

{

Main.dots.Add(new Dot(Main.couples[i, 0], Main.couples[i, 1]));

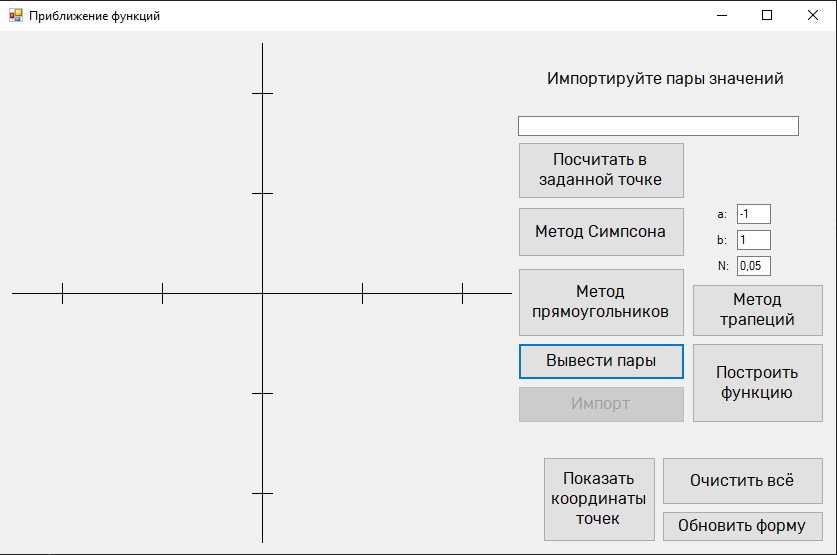
}

}

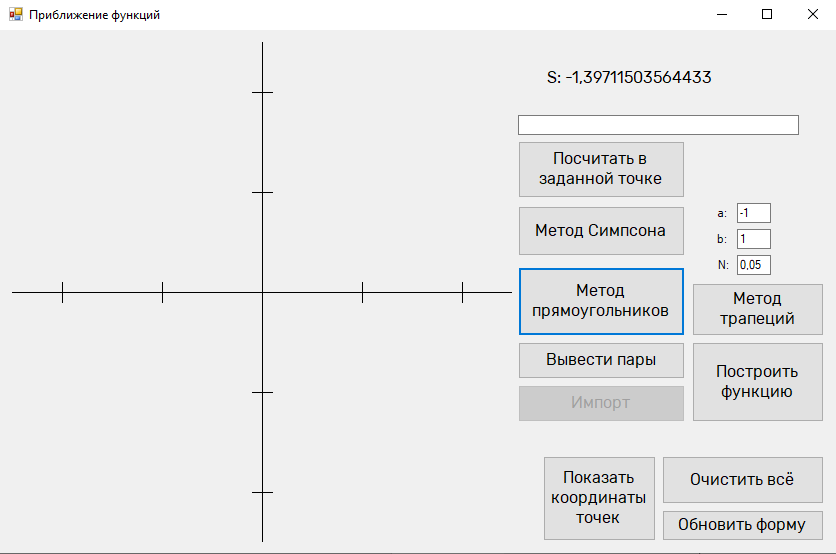
}

}

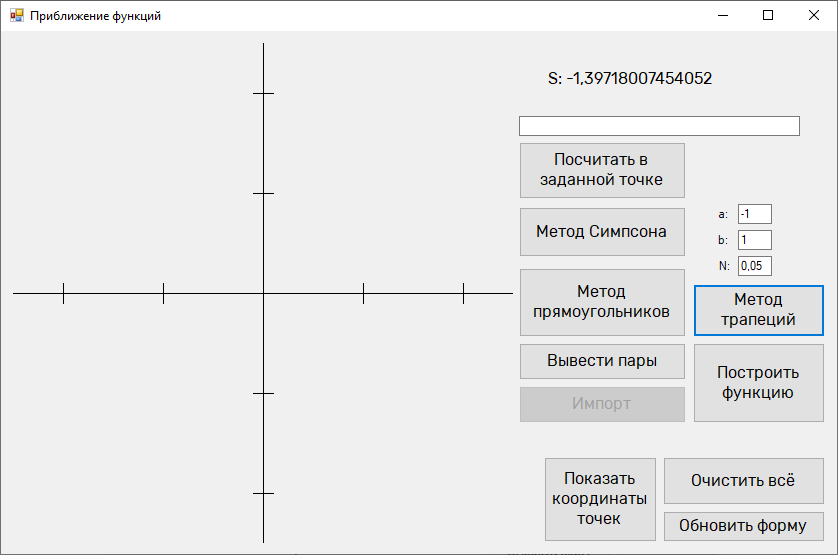
**Вывод программы**

****

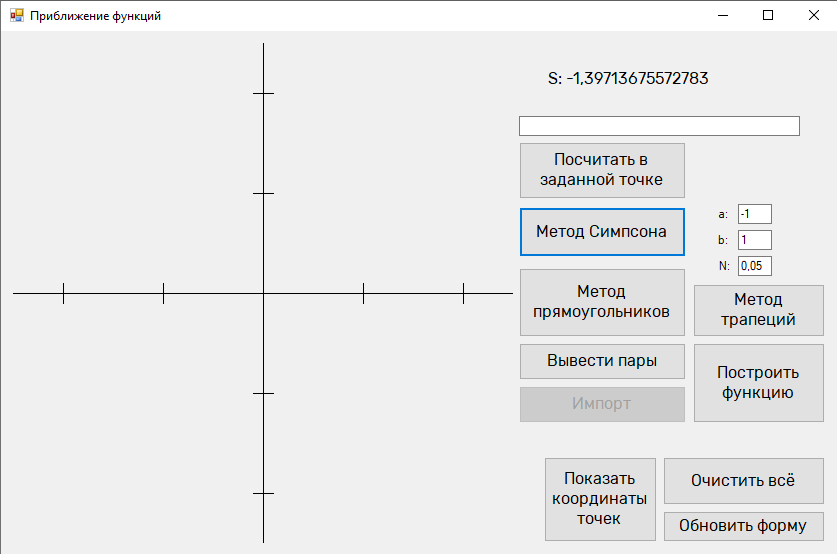
Окно программы



Метод прямоугольников



Метод трапеций



Метод Симпсона

**Вывод**

В ходе данной лабораторной работы были изучены различные методы численного интегрирования. На основе теоретических данных была написана программа, результаты которой совпали с приближенным значением интеграла заданной функции f(x).