Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б. Н. Ельцина»

Уральский энергетический институт

Кафедра «Автоматизированные электрические системы»

**Курсовая работа**

**по дисциплине «Технология программирования»**

**Вариант №29**

Выполнил студент: Ханзафарова А.У

Группа: ЭН-360001

Преподаватель: Семененко С.И.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Екатеринбург. 2018

Оглавление

[Изложение варианта задания 3](#_Toc470214462)

[Постановка задачи в математическом виде 3](#_Toc470214463)

[Схема типов 5](#_Toc470214464)

[Текст программы 5](#_Toc470214465)

Эскиз полученного окна..............................................................................................................9

[Блок-схемы 1](#_Toc470214466)1

[Вывод ............................................................................................................................................1](#_Toc470214468)6

[Список использованной литературы ..1](#_Toc470214469)7

Изложение варианта задания

Для функции f(x)= 10∙x-x2/4 на интервале [-1, 2] рассчитать определённый интеграл приближённым и точным методом. Для приближённого вычисления определённого интеграла использовать метод Симпсона. Результаты, полученные для изменяющейся от 89 до 890 с шагом 100 кратности интегрирования, представить в табличном виде. Расчет выполняется с учетом остаточного члена. Построить график абсолютной погрешности.

Постановка задачи в математическом виде

В методе парабол (Симпсона) определение площади под графиком интегрируемой функции основано на замене двух смежных фрагментов графика участком параболы второго порядка (параболической трапецией). Прежде всего, для этого весь интервал интегрирования должен быть разделен на четное количество отрезков общим количеством n = 2 · m.

Уравнение параболы с осью, параллельной оси Oy, имеет вид:. Коэффициенты A, B и C однозначно определяются из условия, что парабола проходит через три заданные точки. Аналогичные параболы строятся и для других пар отрезков. Сумма площадей параболических трапеций даёт приближённое значение интеграла. При этом площадь одной параболической трапеции равна:

где и - крайние ординаты, а - ордината кривой в середине отрезка. Доказательство:

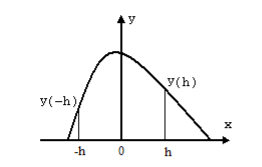


Рис. 1. Параболическая трапеция.

Пусть система координат расположена так, как показано на рисунке 31. Коэффициенты в уравнении параболы определяются из следующих уравнений:

Не вызывает сомнений , что

С другой стороны, площадь параболической трапеции можно определить с помощью определённого интеграла:

Как видно, площадь параболической трапеции совпала с ранее полученной суммой с кратностью h/3. Следовательно:

После этого можно записать формулу для приближенного вычисления интеграла методом парабол:

Суммируя левые и правые части, получим слева искомый интеграл, справа его приближённое значение:

или

Остаточный член для метода Симпсона находится по формуле:

∙ᶓ

где

Схема типов

Схема типов представлена в таблице №1

Таблица №1

|  |  |
| --- | --- |
| Класс Integral | |
| Поля | |
| А | Пределы интегрирования |
| В |
| М | Кратность интегрирования |
| Методы | |
| Integral(double a, double b, int m) | Конструктор |
| fx(double x) | Функция в точке |
| dfx(double x) | Четвертая производная функции в точке |
| Fx(double x) | Первообразная в точке |
| ToString() | Преобразование в строку |
| Свойства | |
| public int m | Кратность интегрирования |
| public double h | Шаг интегрирования |
| public double ИнтЛейбниц | Интеграл по формуле Лейбница |
| public doubleИнтСимпсона | Интеграл по методу парабол (Симпсона) |
|  |  |

Текст программы.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace курсовая\_работа

{

public partial class Form1 : Form

{

Integral I;

public Graphics Area;

public Form1()

{

InitializeComponent();

this.Text = "Численноеинтегрированиефункции 10\*x-x^2/4";

dataGridView1.RowCount = 10;

}

class Integral

{

double A;

double B;

int M;

public Integral(double a, double b, int m)

{

A = a <b ?a : b;

B = a <b ?b : a;

M = m;

}

public int m

{

get { return M; }

set { M = value; }

}

public double fx(double x)

{

return 10 \* x - (Math.Pow(x, 2) / 4);

}

public double Fx(double x)

{

return 5 \* Math.Pow(x, 2) - Math.Pow(x, 3) / 12;

}

public double f4x(double x){ return 0; }

public double h {get { return (B - A) / (2 \* M); }}

public double ИнтЛейбниц {get {return Fx(B) - Fx(A); }}

public double ИнтСимпсон

{

get

{

double sum1 = 0;

for (double i = A + 2 \* h; i< B; i = i + 2 \* h) sum1 += fx(i);

sum1 \*= 2;

double sum2 = 0;

for (double i = A + h; i< B; i = i + 2 \* h) sum2 += fx(i);

sum2 \*= 4;

double sum = sum1 + sum2 + fx(A) + fx(B);

sum \*= h / 3;

double fmax = f4x(A);

for (double i=A; i<B; i=i+h )

{

if (f4x(i) > fmax) fmax=f4x(i);

}

double p = Math.Pow((B - A), 5) / (180 \* Math.Pow(2\*m, 4))\*fmax;

sum += p;

return sum;

}

}

public override string ToString()

{

return String.Format("A={0:f2} B={1:f2} M={2,4}", A, B, M);

}

}

private void button1\_Click(object sender, EventArgs e)

{

double A = Convert.ToDouble(textBox1.Text);

double B = Convert.ToDouble(textBox2.Text);

int M = Convert.ToInt32(textBox3.Text);

I = new Integral(A, B, M);

this.label1.Visible = false;

this.label2.Visible = false;

this.label3.Visible = false;

this.label4.Visible = false;

this.textBox1.Visible = false;

this.textBox2.Visible = false;

this.textBox3.Visible = false;

this.button1.Visible = false;

this.button2.Visible = true;

this.Text += I;

}

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

{

int M0 = Convert.ToInt32(textBox3.Text);

for (int i = 0; i< 10; i++)

{

I.m = (int)((i + 1) / 10.0 \* M0);

dataGridView1.Rows[i].Cells[0].Value = I.m;

dataGridView1.Rows[i].Cells[1].Value = I.ИнтСимпсон;

dataGridView1.Rows[i].Cells[2].Value = I.ИнтЛейбниц;

dataGridView1.Rows[i].Cells[3].Value = I.ИнтЛейбниц- I.ИнтСимпсон;

}

dataGridView1.Visible = true;

button2.Visible = false;

this.button4.Visible = true;

}

private void button4\_Click(object sender, EventArgs e)

{

Area = panel2.CreateGraphics();

int w = panel2.ClientSize.Width;

int h = panel2.ClientSize.Height;

Pen P = new Pen(Color.Brown, 2);

Area.DrawLine(P, 0, 0, w, 0);

Area.DrawLine(P, 0, 0, 0, h);

Area.DrawLine(P, 0, h, w, h);

Area.DrawLine(P, w, 0, w, h);

P = new Pen(Color.DarkMagenta, 3);

Area.DrawLine(P, 10, 10, 10, h - 10);

Area.DrawLine(P, 10, (h - 20)/2, w - 10, (h - 20)/2);

Area.DrawString("A",

new Font("Corier new", 12),

new SolidBrush(Color.DarkRed),

new Point(20, 10));

Area.DrawString("m",

new Font("Corier new", 12),

new SolidBrush(Color.DarkRed),

new Point(w - 20, h/2-40));

int M0 = Convert.ToInt32(textBox3.Text);

float[] p = new float[11];

float[] d = new float[11];

for (int i = 0; i<= 10; i++)

{

I.m = (int)((i + 1) / 10.0 \* M0);

p[i] = I.m;

d[i] = (float)(I.ИнтЛейбниц - I.ИнтСимпсон);

}

P = new Pen(Color.DarkGreen, 3);

float dmax = Math.Abs(d[0]);

foreach (float i in d) if (dmax< Math.Abs(i)) dmax = Math.Abs(i);

float pmax = Math.Abs(p[0]);

foreach (float i in p) if (pmax< Math.Abs(i)) pmax = Math.Abs(i);

float My = ((h-20)/2 ) / dmax;

float Mx = (w - 20)/pmax;

float ix, iy, ix1 = 10 + p[0]\*Mx , iy1 = (h - 20)/2 - d[0] \*My;

for (int i = 0; i< 10; i++)

{

ix = ix1;

iy = iy1;

ix1 = 10 +p[i + 1]\*Mx ;

iy1 = (h - 20)/2 -( d[i + 1] \* My);

Area.DrawLine(P, ix, iy, ix1, iy1);

}

}

}

}

Эскиз окна с полученными результатами

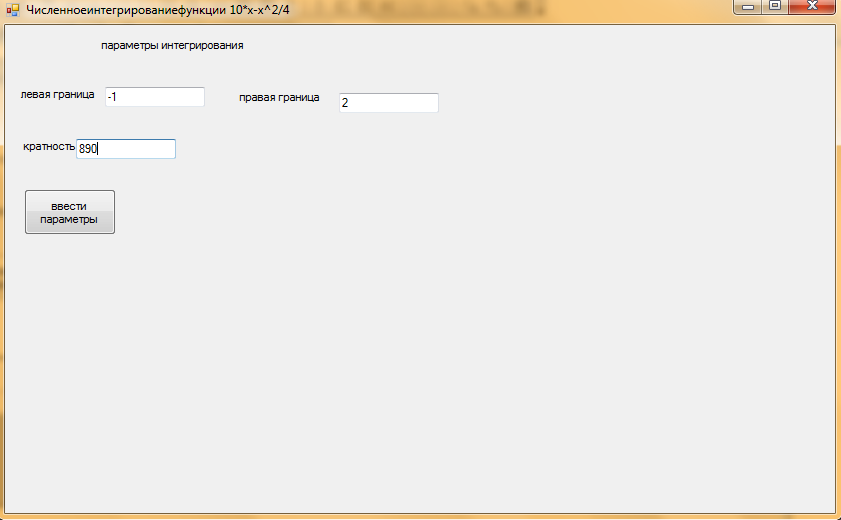


Рис 1. Ввод данных на форме.

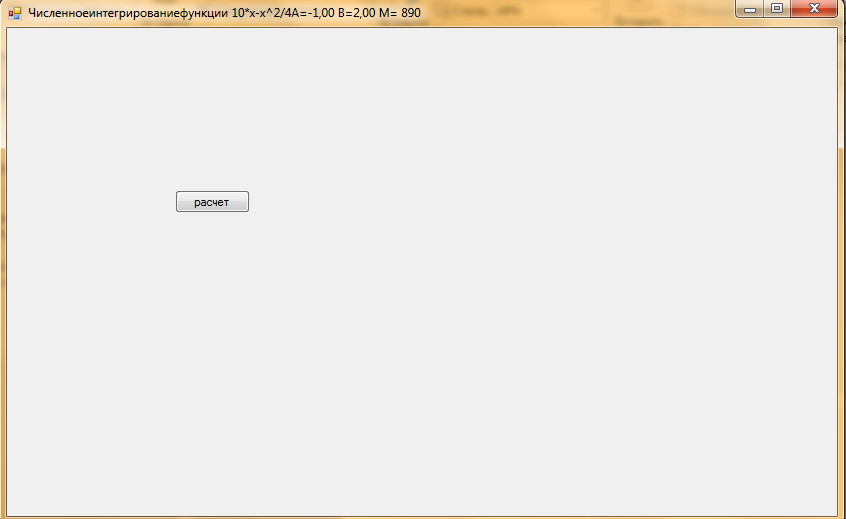


Рис 2. Расчет интеграла.

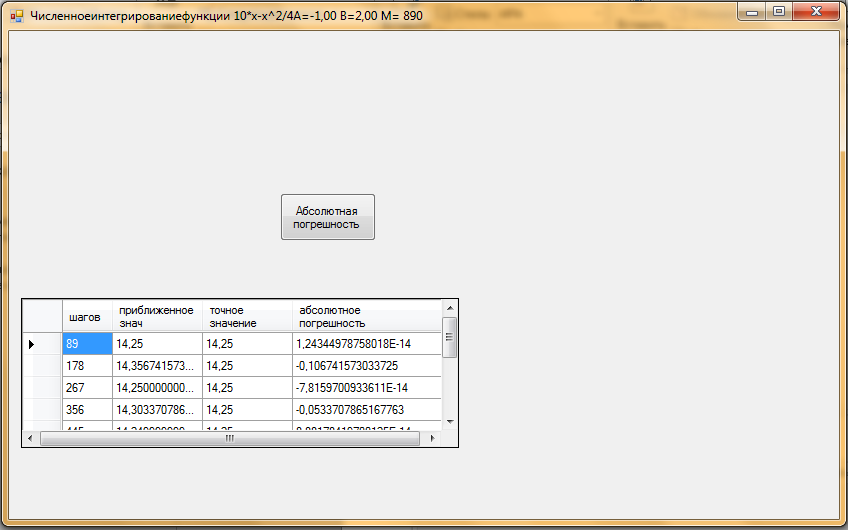


Рис 3. Результаты расчета.

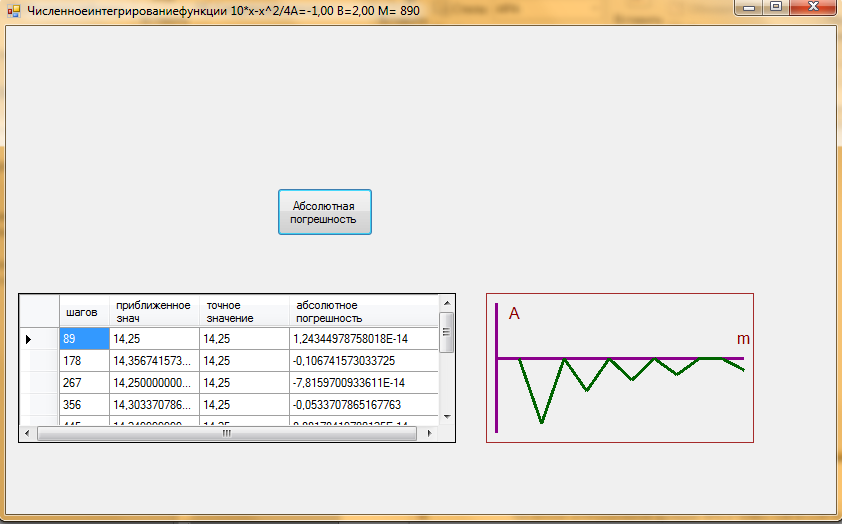


Рис 4. Построение графика абсолютной погрешности.

sum1 = 0

i = A+2\*h

i<B

sum1 += fx(i)

i=i+2\*h

sum1 \*= 2

sum2 = 0

i = A+h

i<B

sum2 += fx(i)

i=i+2\*h

sum1 \*= 4

sum = sum1 + sum2 + fx(A) + fx(B)

sum1 \*= h / 3

i<B

i=A

i=i+h

f4x(i) >fmax

fmax=f4x(i)

fmax = f4x(A)

p = Math.Pow((B - A), 5) / (180 \* Math.Pow(2\*m, 4))\*fmax

sum += p

Рис 5. Блок-схема вычисления интеграла методом Симпсона.

Редактирование первой строки окна

Преобразование textBox1 в число A

label1 становится невидимой

label2 становится невидимой

label3 становится невидимой

Преобразование textBox2 в число B

Преобразование textBox3 в число M

Создание объекта класса Integral

label4 становится видимой

textBox1 становится видимой

textBox2 становится видимой

textBox3 становится видимой

Button1 становится видимой

button2 становится видимой

Рис 6. Блок-схема обработчика нажатия на кнопку button1.

i<10

Преобразование textBox3 в число M0

i = 0

I.m = (int)((i + 1) / 10.0 \* M0);Задание шага интегрирования

Вывод шага интегрирования

Вывод результата приближенным методом

Вывод результата точным методом

i++

Таблица становится видимой

Кнопка 2 становится невидимой

Кнопка 4 становится видимой

Вывод абсолютной погрешности

Рис 7. Блок-схема обработчика нажатия на кнопку button2.

Область рисования – panel2

Задание параметров пера

h= высоте panel2

w = ширина panel2

Рисование правой границы

Рисование верхней границы изображения

Рисование левой границы

Рисование нижней границы

Задание новых параметров пера

Рисование вертикальной оси

Рисование горизонтальной оси

подпись вертикальной оси

подпись горизонтальной оси

Задание массива х

Задание массива f(x)

i = 0

i<= 10

I.m = (int)((i + 1) / 10.0 \* M0);

p[i] = I.m

i++

Задание новых параметров пера

Преобразование textBox3 в число M0

d[i] = (float)(I.ИнтЛейбниц - I.ИнтСимпсон)

Рис 8. Блок-схема обработчика нажатия на кнопку button4.

Продолжение стр. 15

Задание масштаба по оси ОУ

Задание масштаба по оси ОХ

Задание ix1

Задание iy1

i = 0

i<10

i ++

ix = ix1;

iy = iy1;

ix1 = 10 + x[i + 1] \* Mx;

iy1 = h - 10 - fx[i + 1] \* My;;

Рисование отрезка по точкам (ix,iy) (ix1,iy1)

Для iв массиве d

dmax = Math.Abs(d[0])

dmax<Math.Abs(i)

dmax = Math.Abs(i)

Для iв массиве p

pmax = Math.Abs(p[0])

pmax<Math.Abs(i)

pmax = Math.Abs(i)

конец

Продолжение рис 8.

# Выводы

В данной курсовой работе были изучены методы расчета определенных интегралов в среде программирования C# . Значения, полученные при вычислении определенного интеграла заданной функции методом Симпсона (парабол) оказались довольно близкими по сравнению с точным значением, вычисленным методом Ньютона-Лейбница, что показывает достаточно точную аппроксимацию функции в данном методе расчета. Для уменьшения погрешности вычисления, необходимо увеличивать кратность интегрирования. Также для увеличения точности был введен остаточный член. Как видно использование программ позволяет достаточно точно решать математические задачи приближенными методами.

# Список использованной литературы

1. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ С#: Методические указания / О.М.Котов. Екатеринбург: УрФУ, 2016. 41 с.
2. Язык C#: краткое описание и введение в технологии программирования: учебное пособие / О. М. Котов. - Екатеринбург: Изд-во Урал. 2014. 208 с.