Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий Кафедра прикладной математики

Отчет защищен с оценкой Преподаватель Проскурин А. В.

« » 2022 г.

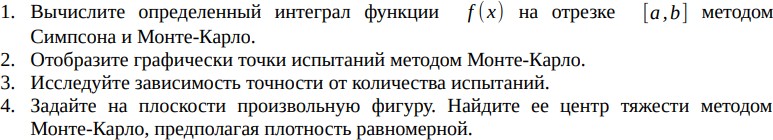
Отчет Лабораторная работа №7

Интегрирование методом Монте-Карло

по дисциплине «Вычислительные алгоритмы»

Студент группы ПИ 92 Д. А. Савиков, Ю. К. Гранкин

Задание

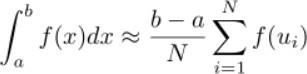


Алгоритм

Метод Симпсона Формула Симпсона:

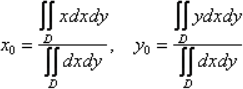


Метод Монте-Карло



N – количество испытаний, u – очередное случайное число принадлежащее отрезку [a,b] Поиск центра тяжести прямоугольного треугольника

Координаты центра тяжести находятся по таким формулам:



Кратные интегралы находятся по формуле:



n – количество испытаний, ni – случайная точка принадлежащая параллелепипеду содержащему треугольник.

В эту сумму входят только те ni которые принадлежат области треугольника.

Так как во всех четырех двойных интегралах J и 1/n одинаковые, то их можно сразу сократить.

В итоге получается, что x0 равно отношению суммы координаты x всех случайных точек, которые принадлежат треугольнику и количества таких точек, а y0 – также, только в числителе сумма координаты y всех случайных точек, принадлежащих треугольнику.

Текст программы 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;

namespace AlgLab7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Поиск f(x)

public double Funk(string f, double x)

{

switch (f) //В зависимости от f выбирается функция

{

case "f(x)=x^2": return x \* x;

case "f(x)=|x|": return Math.Abs(x);

case "f(x)=cos(x)": return Math.Cos(x);

case "f(x)=e^(-x\*x)": return Math.Exp(-x \* x);

}

return 0;

}

шаг

//Поиск интеграла методом Симпсона с постоянным шагом

//f - подинтегральная функция, N - количество сегментов на отрезке [a,b], h -

public double Simp(double a, double b, string f, double h, int N)

{

double res, x; res = 0;

x = a;

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

{

x += h;

res += Funk(f, x - h) + 4 \* Funk(f, x - h / 2) + Funk(f, x);

}

return res \* h / 6;

}

//Поиск интеграла методом Монте-Карло

//f - подинтегральная функция, N - количество испытаний public double Monte(double a, double b, string f, int N)

{

Random rand = new Random(); double y, x, sum; chart1.Series[0].Points.Clear();

chart1.ChartAreas[0].AxisX.Minimum = a; chart1.ChartAreas[0].AxisX.Maximum = b; sum = 0;

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

{

x = rand.NextDouble() \* (b - a) + a; y = Funk(f, x);

sum += y; chart1.Series[0].Points.AddXY(x, y);

}

sum = sum \* (b - a) / N; return sum;

}

//Кнопка поиска интеграла

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

{

string file, funk; double a, b, sim, mon; int nMonk, nSimp;

file = textBox1.Text; if (File.Exists(file))

{

try //Считывание информации из файла

{

интегрирования

StreamReader stream = new StreamReader(file);

funk = stream.ReadLine(); //строка в которой записана функция stream.ReadLine();

a = Convert.ToDouble(stream.ReadLine()); //Границы отрезка

b = Convert.ToDouble(stream.ReadLine()); stream.ReadLine();

nMonk = Convert.ToInt32(stream.ReadLine()); //Кол-во испытаний

для метода Монте-Карло

stream.ReadLine();

nSimp = Convert.ToInt32(stream.ReadLine()); //Кол-во сегментов на которые надо поделить [a,b] для метода Симпсона

stream.Close();

}

catch

{

MessageBox.Show("Данные введены не правильно"); return;

}

}

else

{

MessageBox.Show("Файл не найден"); return;

}

sim = Simp(a, b, funk, (b - a) / nSimp, nSimp); mon = Monte(a, b, funk, nMonk);

StreamWriter str = File.AppendText(file);

str.WriteLine(); //Результаты записываются в тот же файл str.WriteLine("Симпсон: " + sim);

str.WriteLine("Монте-Карло: " + mon); str.Close();

}

//Поиск центра тяжести

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

{

double ax, bx, ay, by, x, y, upx, upy, dow; ax = 1; //Границы для x и y

ay = 1;

bx = 6;

by = 4; chart1.Series[1].Points.Clear(); chart1.ChartAreas[0].AxisX.Minimum = 0;

chart1.ChartAreas[0].AxisX.Maximum = bx + 1; chart1.ChartAreas[0].AxisX.MajorGrid.Interval = 1;

chart1.ChartAreas[0].AxisY.Minimum = 0; chart1.ChartAreas[0].AxisY.Maximum = by + 1; chart1.ChartAreas[0].AxisY.MajorGrid.Interval = 1; chart1.Series[1].Points.AddXY(1, 1); //Построение прямоугольного

треугольника по координатам (1,1), (6,4), (6,1)

chart1.Series[1].Points.AddXY(6, 4);

chart1.Series[1].Points.AddXY(6, 1);

chart1.Series[1].Points.AddXY(1, 1); Random rand = new Random();

upx = upy = dow = 0; //upx - числитель в формуле поиска x центра тяжести, upy - тоже самое но для y, dow - знаменатель для обоих формул

for (int i = 0; i < 900000; i++) //900000 испытаний

{

y = rand.NextDouble() \* (by - ay) + ay; x = rand.NextDouble() \* (bx - ax) + ax;

if (x >= (y - 0.4) / 0.6) //Проверка того входит ли очередная точка в область треугольника

{

upy += y; upx += x; dow++;

}

}

x = upx / dow; y = upy / dow;

chart1.Series[0].Points.Clear();

chart1.Series[0].Points.AddXY(x, y); //Построение точки центра тяжести и вывод результата

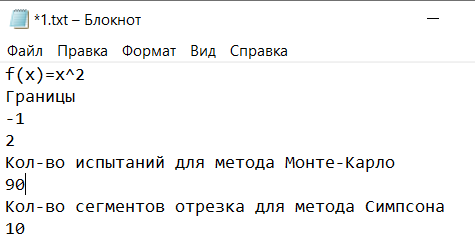
MessageBox.Show("(" + x + " ; " + y + ")");

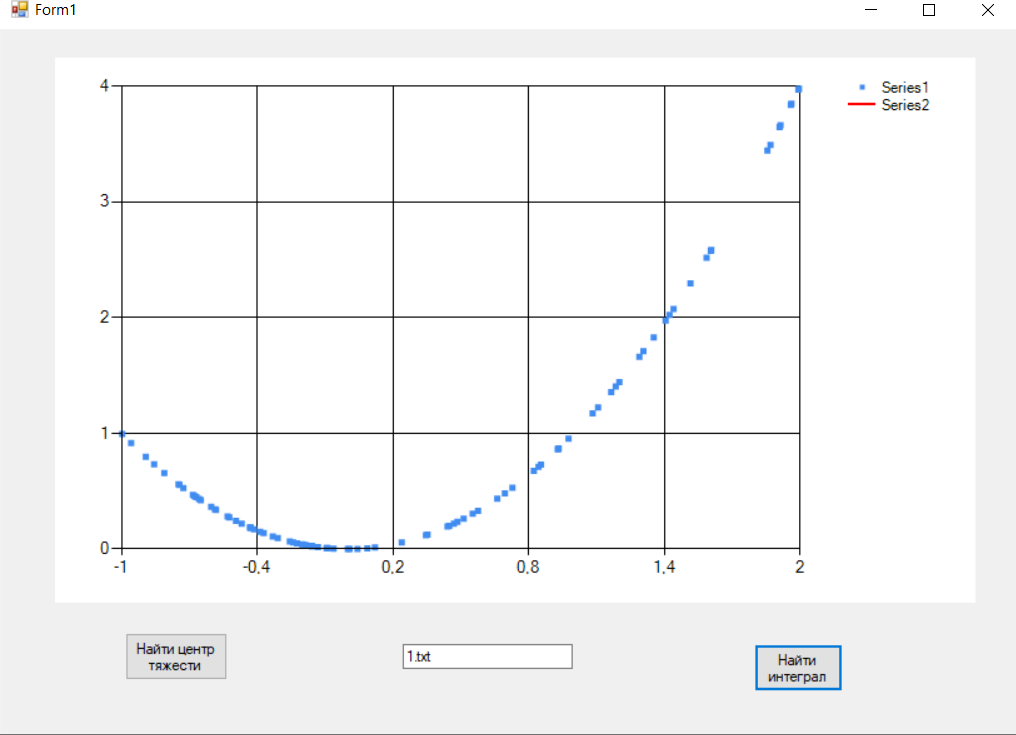
}

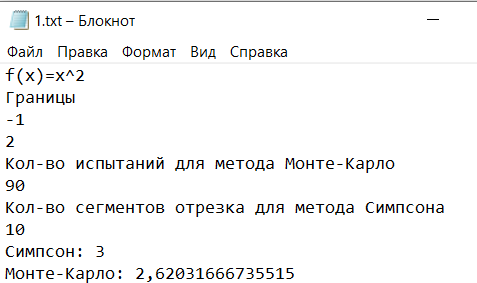
}

}

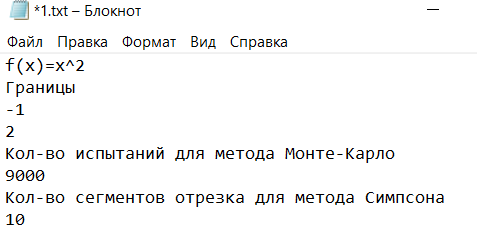
Тест программы 1)

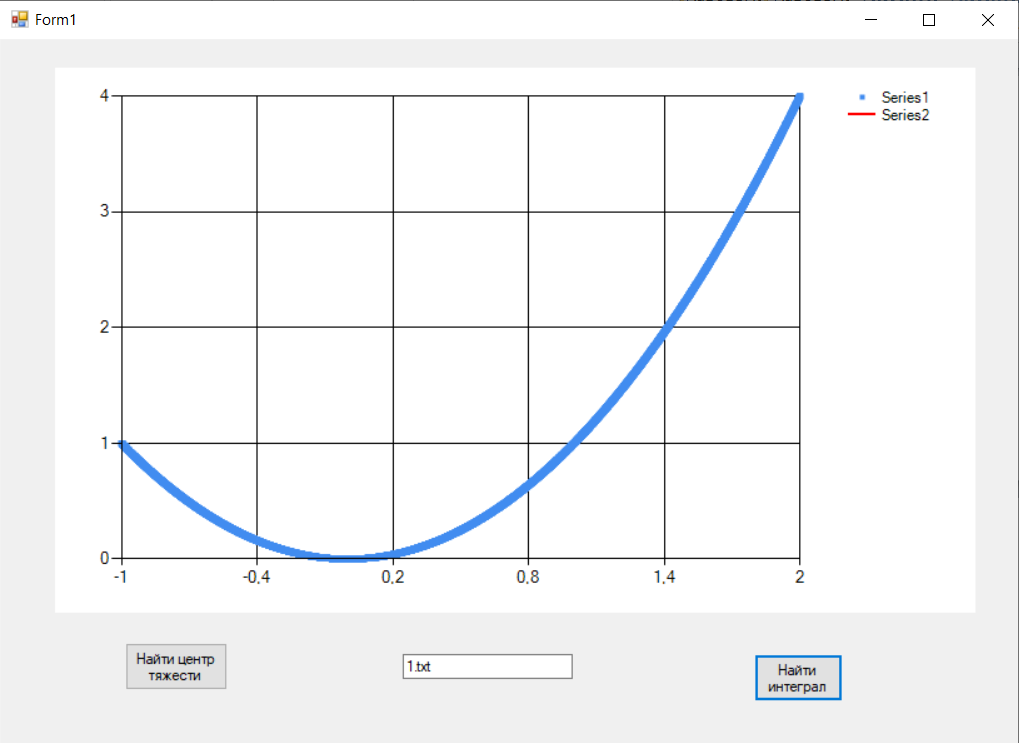


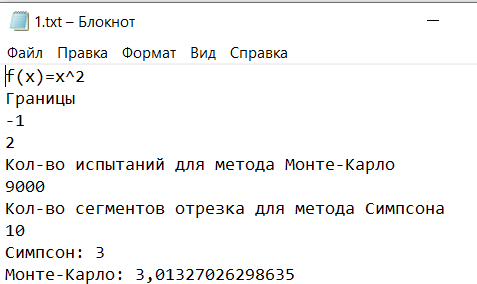




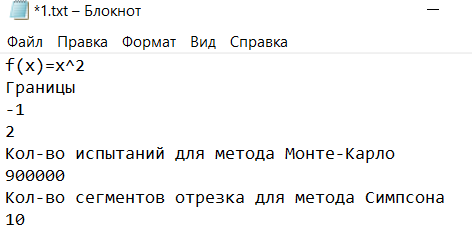
Увеличим кол-во испытаний

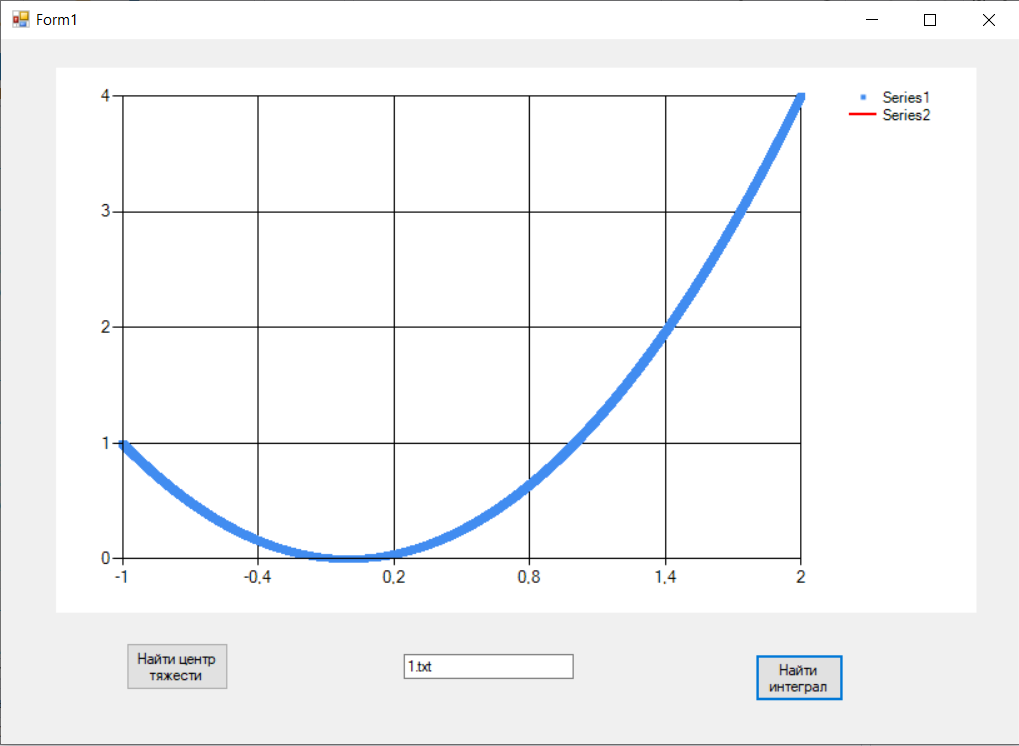


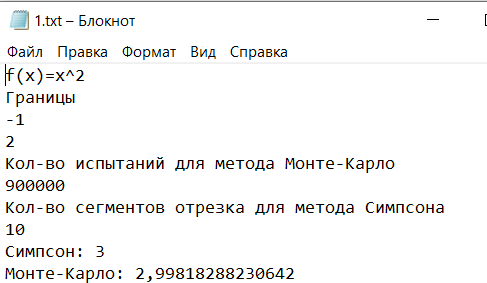




Еще раз увеличим кол-во испытаний

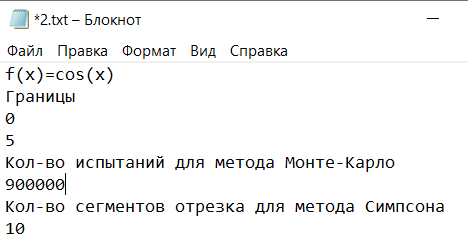


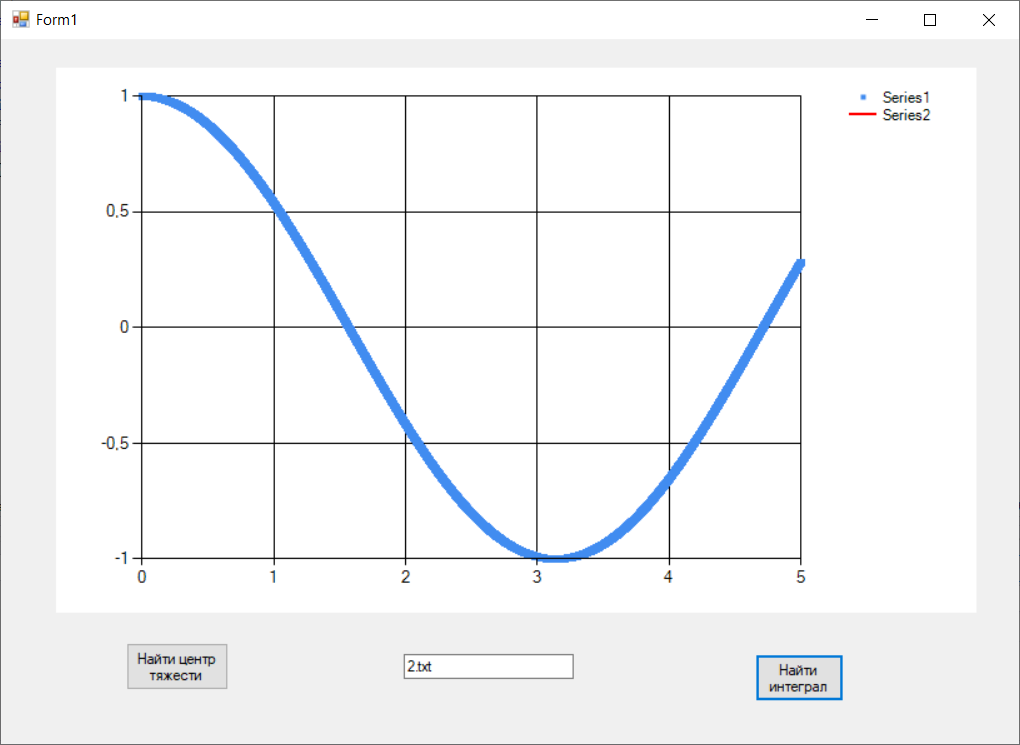


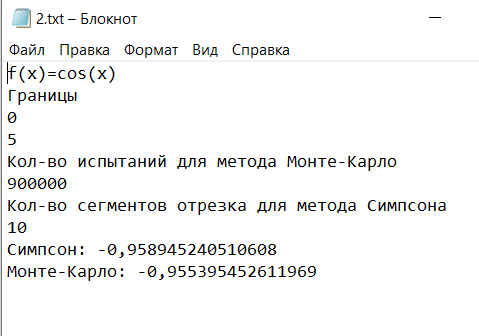


С увеличением количества испытаний точность результата также увеличивается, но для точных результатов нужно очень много испытаний (даже 900000 испытаний дали результат с погрешностью примерно 0,002).

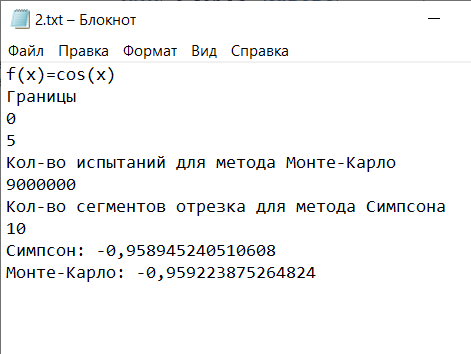
2)

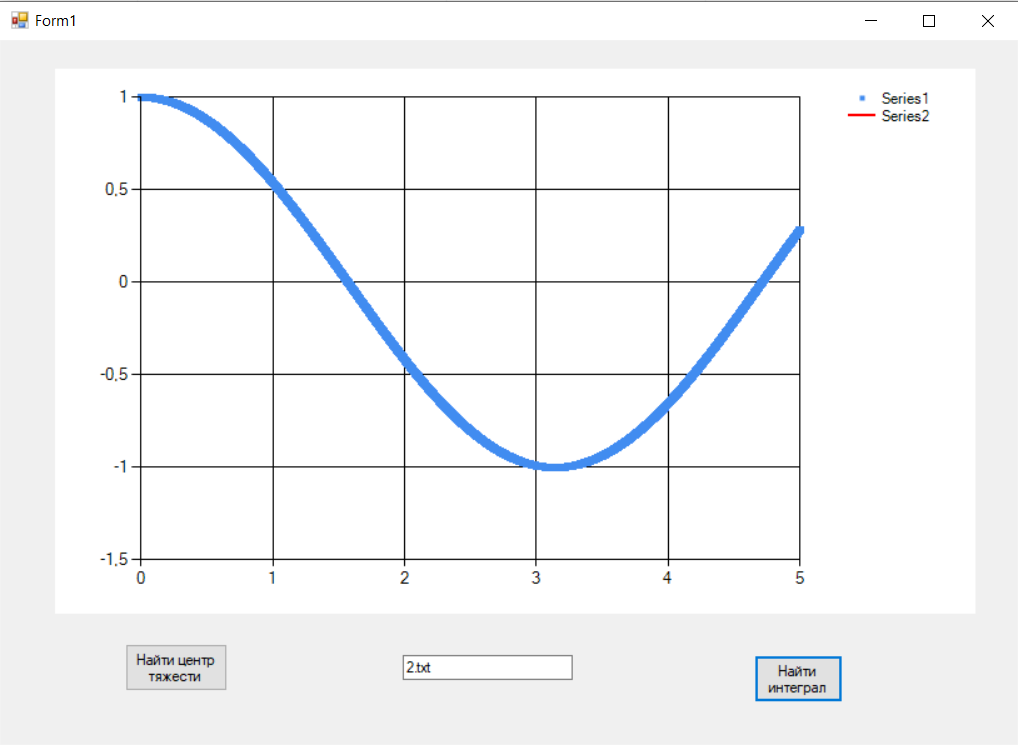




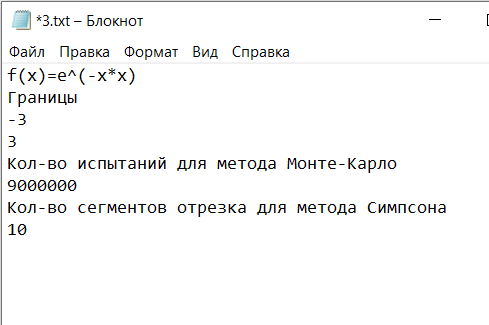


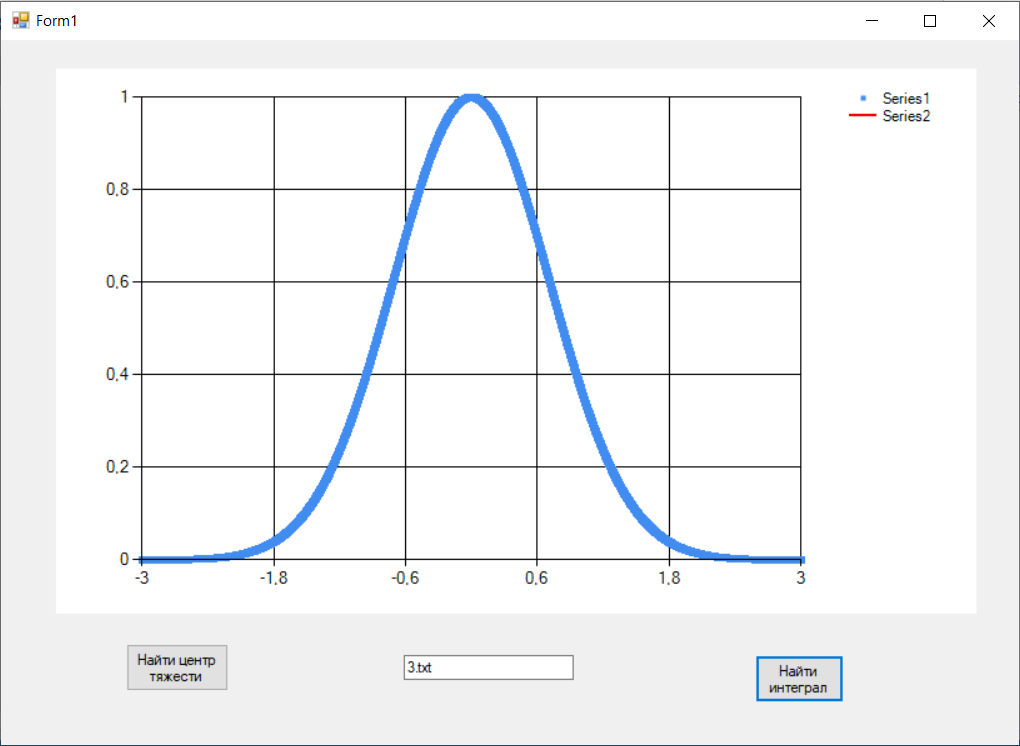
Увеличим кол-во испытаний

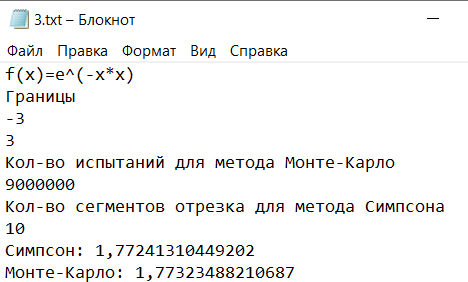




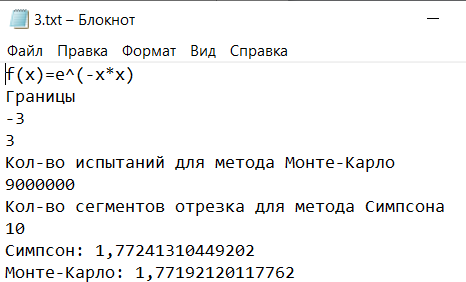
3)





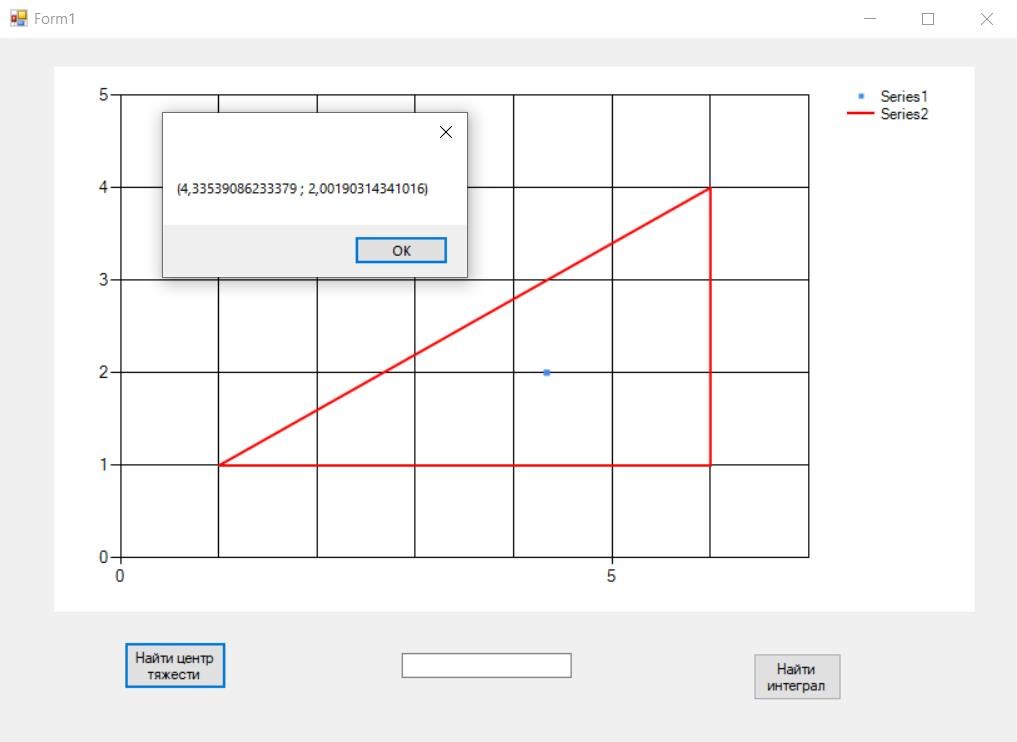


Другой тест с таким же количеством испытаний



Результаты отличаются, потому что они зависят от выбранных случайно точек.

4)Центр тяжести прямоугольного треугольника



Вывод: результат полученный методом Монте-Карло, помимо фактора случайности, зависит также от количества испытаний. Для более-менее высокой точности требуется очень много испытаний. Для того чтобы найти определенный интеграл с точность e и доверительной вероятность y нужно с помощью функции Лапласа найти x удовлетворяющий Ф(x)=y и проделать такое кол-во испытаний, чтобы выполнялось

условие x\*(D/n)^1/2 < e (D –дисперсия случайной величины определяемой в испытаниях, n – кол-во испытаний).