

Министерство образования и науки Российской Федерации
Федеральное государственное образовательное учреждение
высшего профессионального образования

«Алтайский Государственный Технический университет им. И.
И. Ползунова»

Факультет информационных технологий
Кафедра прикладной математики

Отчёт защищён с оценкой _____

«__» 2022 г.

_____ Проскурин А. В.

Отчет
по лабораторной работе №7
«Вычисление интегралов методом Монте-Карло»
по дисциплине «Вычислительные алгоритмы»

Студенты группы ПИ-91: Винокуров А.Ю. , Бедарев А.С.

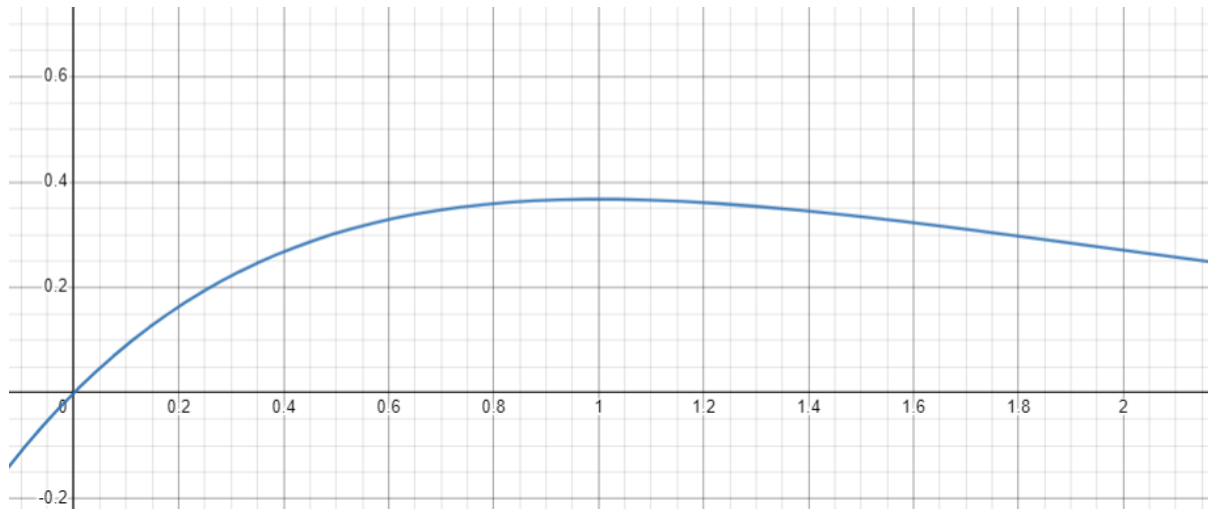
Преподаватель, к. ф. м. н. Проскурин А. В.

Барнаул 2022

Вычисление интегралов методом Монте-Карло

1. Вычислите определенный интеграл функции $f(x)$ на отрезке $[a, b]$ методом Симпсона и Монте-Карло.
2. Отобразите графически точки испытаний методом Монте-Карло.
3. Исследуйте зависимость точности от количества испытаний.
4. Задайте на плоскости произвольную фигуру. Найдите ее центр тяжести методом Монте-Карло, предполагая плотность равномерной.

Функция $y = x * e^{-x}$ на отрезке $[0; 2]$



Первое значение - вычисление интеграла по Симпсону

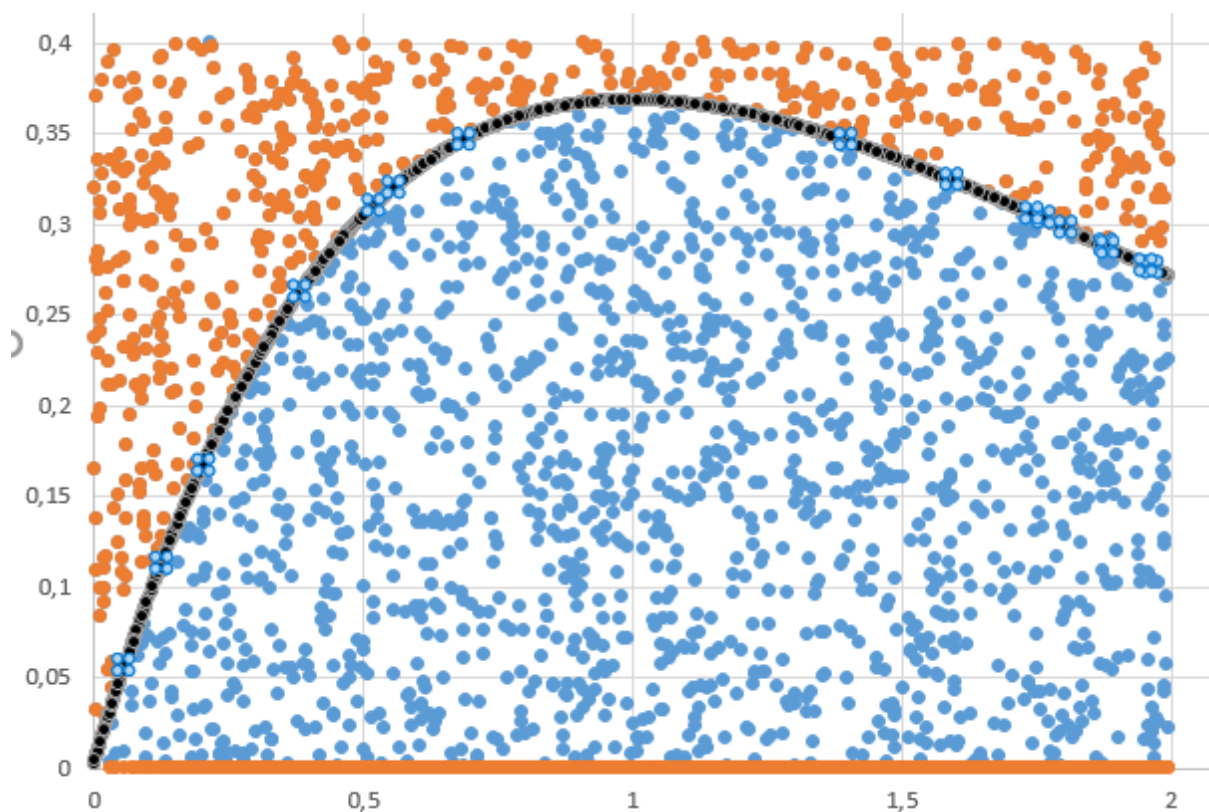
Второе - по монте-карло

C:\Windows\system32\cmd.exe

```
Integral Vaolue = 0,593045288188659
Monte-Carlo Vaolue = 0,592359476174256
Number or Points = 20000
Delta = 0,00578886615021446
Для продолжения нажмите любую клавишу . . .
```

Далее я занес случайные значения, генерируемые программой, внес в Excel, и построил график для наглядности

- Синие точки - попавшие
- Оранжевые - не попавшие



Метод Монте-Карло довольно неточный при вычислении обычного интеграла, поэтому для нормальной точности требуется брать как можно больше точек

```
Integral Vaolue = 0,593045288188659
Monte-Carlo Vaolue = 0,00367879441171442
Number or Points = 200
Delta = -0,589366493776945
Для продолжения нажмите любую клавишу
```

C:\Windows\system32\cmd.exe

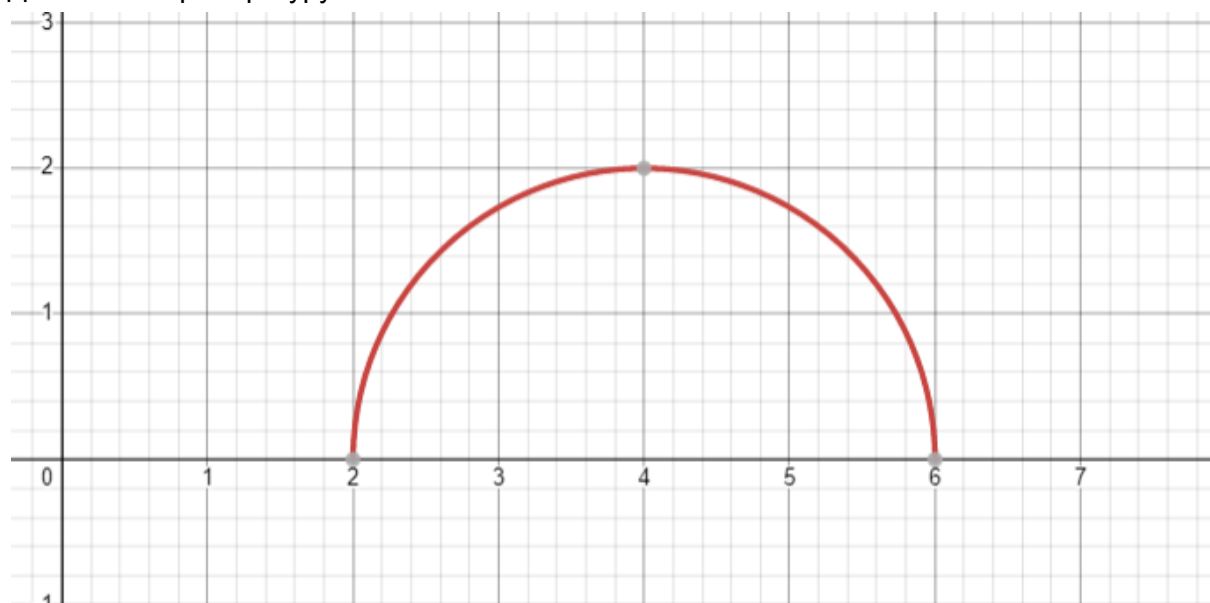
```
Integral Vaolue = 0,593045288188659
Monte-Carlo Vaolue = 0,65556116416751
Number or Points = 2000
Delta = 0,0625158759788509
Для продолжения нажмите любую клавишу . . .
```

C:\Windows\system32\cmd.exe


```
Integral Vaolue = 0,593045288188659
Monte-Carlo Vaolue = 0,592359476174256
Number or Points = 20000
Delta = 0,00578886615021446
Для продолжения нажмите любую клавишу . . .
```

Таким образом чем больше выборка, тем более точный результат получается

Далее я выбрал фигуру



1

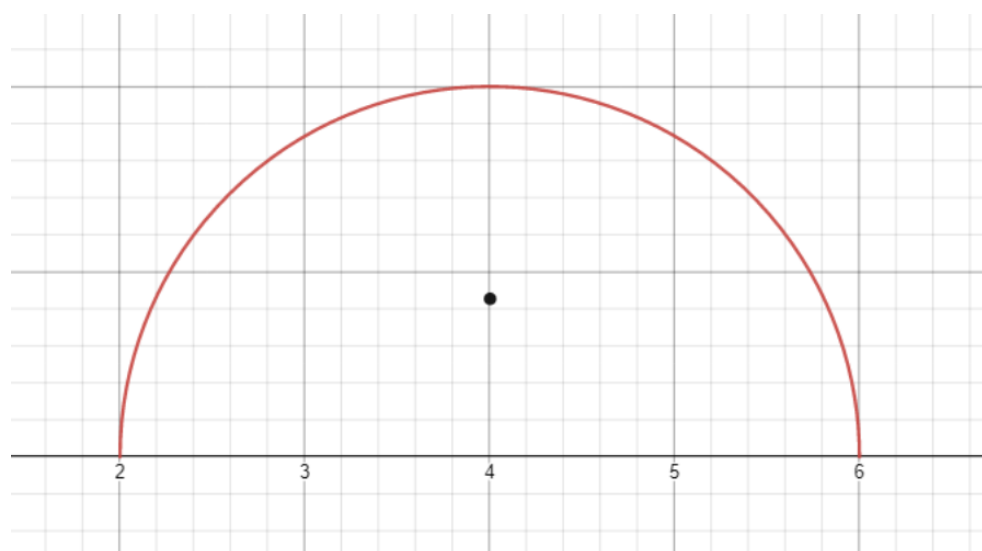

$$y = \sqrt{4 - (x - 4)^2}$$

2

Это полукруг

Вот такие координаты центра тяжести получились

```
Integral Value = 0,50888895047  
X Centre = 4,00187334099243  
Y Centre = 0,853585817106453
```



Y центр = $4 * R/3\pi = 0,8488263631567$

X центр = 4

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Algorithms_L7
{
    public class Program
    {
        static double N = 20000;
        static double func(List<double> X, List<double> Y, List<double> Space, double h) {

            double buff = Space.First();
            double maxY = 0;

            while (buff<=Space.Last())
            {
                X.Add(buff);
                Y.Add(buff * Math.Exp(-buff));

                if (maxY < buff * Math.Exp(-buff))
                {
                    maxY = buff * Math.Exp(-buff);
                }

                buff += h;
            }
            return maxY;
        }

        static bool MoreOrLess(double x, double y)
        {
            double result = Math.Pow(x,2) + Math.Pow(x, 2);

            double yValue = x * Math.Exp(-x);
            if (yValue >= y) return true;
            else return false;
        }

        static bool KrugMoreOrLess(double x, double y)
        {
            double yValue = Math.Sqrt(4 - Math.Pow(x - 4, 2));
            if (yValue >= y) return true;
        }
    }
}

```

```

else return false;

}

static double MonteCarlo( List<double> xSpace, List<double> ySpace)
{
    FileStream Xfstream = new FileStream("X.txt", FileMode.OpenOrCreate);
    StreamWriter Xsw = new StreamWriter(Xfstream);

    FileStream Yfstream = new FileStream("Y.txt", FileMode.OpenOrCreate);
    StreamWriter Ysw = new StreamWriter(Yfstream);

    List<double> Exclusive = new List<double>();

    int counter = 0;

    double xCentre = 0;
    double yCentre = 0;

    int Time = DateTime.Now.Millisecond;
    for (int i = 0; i < N; i++)
    {
        Random xRandom;
        Random yRandom;

        do
        {
            xRandom = new Random(DateTime.Now.Millisecond);
            yRandom = new Random(DateTime.Now.Millisecond);

        } while (Exclusive.Contains((xSpace.Last() - xSpace.First()) * xRandom.NextDouble()));

        Exclusive.Add(((xSpace.Last() - xSpace.First()) * xRandom.NextDouble()));

        double xPoint = (xSpace.Last() - xSpace.First()) * xRandom.NextDouble();
        double yPoint = (ySpace.Last() - ySpace.First()) * yRandom.NextDouble();

        Xsw.WriteLine(xPoint.ToString());
        Ysw.WriteLine(yPoint.ToString());

        if (MoreOrLess(xPoint, yPoint))
        {
            counter++;
            xCentre += xPoint;

```

```
yCentre += yPoint;
```

```
    }  
}
```

```
double Square = (Convert.ToDouble(counter) / N) * (xSpace.Last() - xSpace.First()) * (ySpace.Last() -  
ySpace.First());
```

```
Xfstream.Close();
```

```
Yfstream.Close();
```

```
xCentre /= counter;
```

```
yCentre /= counter;
```

```
Console.WriteLine("X Centre = " + xCentre);
```

```
Console.WriteLine("Y Centre = " + yCentre);
```

```
return Square;
```

```
}
```

```
static double Simpson(List<double> X, List<double> Y, List<double> Space, double h)
```

```
{
```

```
double result = 0;
```

```
int first = 0;
```

```
int last = Y.Count() - 1;
```

```
result += Y.First() + Y.Last();
```

```
result += (4 * Y.Where((item, index) => index % 2 == 1 && index != first).Sum());
```

```
result += (2 * Y.Where((item, index) => (index % 2 == 0) && index != last).Sum());
```

```
return result * h / 3;
```

```
}
```

```
static void Main(string[] args)
```

```
{
```

```
double n = 4;
```

```
List<double> valuesX = new List<double>();
```

```
List<double> valuesY = new List<double>();
```

```
List<double> xSpace = new List<double>() { 0, 2 };
```

```
double h = (xSpace.LastOrDefault() - xSpace.FirstOrDefault()) / n;
```

```
List<double> ySpace = new List<double>() { 0, 2};
```

```
func(valuesX, valuesY, xSpace, h);
```

```
double SimpsonSquare = Simpson(valuesX, valuesY, xSpace, h);
```

```
Console.WriteLine("Integral Vaolue = " + SimpsonSquare);
```

```
double MonteCarloSquare = MonteCarlo(xSpace, ySpace);
```

```
Console.WriteLine("Monte-Carlo Vaolue = " + MonteCarloSquare);
```

```
Console.WriteLine("Number or Points = " + Convert.ToInt32(N));
```

```
Console.WriteLine("Delta = " + Convert.ToDouble( MonteCarloSquare - SimpsonSquare));
```

```
}
```

```
}
```

```
}
```