БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**курса «Имитационное и статистическое моделирование»**

**Вариант 11**

Выполнил студент:

Жданович Павел Александрович

4 курс, 2 группа

Минск, 2018

**Постановка задачи**

Вычислить интеграл

по методу Монте-Карло, используя простейщий метод и метод симметризации подынтегральной функции.

**Краткая теория метода Монте-Карло**

Рассмотрим задачу приближенного вычисления интеграла

где - подмножество из  При n=1 имеем определенный интеграл вида

Заметим, что схема вычислений как многомерных, так и одномерных интегралов, абсолютно аналогична.

Пусть η ‑ произвольная случайная величина с плотностью распределения вероятностей 

Предполагается только, что существуют моменты случайных величин, встречающиеся ниже. Рассмотрим случайную величину, являющуюся функциональным преобразованием случайной величины η.

Можно показать, что 

Случайная величина, используемая для вычисления интеграла по методу Монте-Карло:

Случайная величина, используемая для вычисления интеграла по методу симметризации подынтегральной функции:

**Результат**

Приближенное значение интеграла, полученное аналитическими методами :

Результаты работы программы:

Real value 1.3463879568

Simple method

N result error

10 9.853087E-001 3.610793E-001

20 7.495824E-001 5.968056E-001

50 1.215735E+000 1.306532E-001

100 1.314977E+000 3.141131E-002

200 1.288277E+000 5.811057E-002

500 1.380077E+000 3.368931E-002

1000 1.446044E+000 9.965605E-002

2000 1.332139E+000 1.424902E-002

5000 1.351490E+000 5.102536E-003

10000 1.353453E+000 7.065195E-003

20000 1.334185E+000 1.220317E-002

50000 1.351180E+000 4.791623E-003

100000 1.344270E+000 2.117978E-003

200000 1.350807E+000 4.419137E-003

500000 1.346373E+000 1.485761E-005

1000000 1.349743E+000 3.355074E-003

2000000 1.346505E+000 1.172938E-004

5000000 1.346162E+000 2.263855E-004

Symmetrization method

N result error

10 1.356171E+000 9.782587E-003

20 1.530133E+000 1.837451E-001

50 1.247165E+000 9.922326E-002

100 1.358932E+000 1.254446E-002

200 1.403895E+000 5.750721E-002

500 1.390796E+000 4.440792E-002

1000 1.322999E+000 2.338865E-002

2000 1.364142E+000 1.775388E-002

5000 1.339926E+000 6.461784E-003

10000 1.344433E+000 1.955364E-003

20000 1.347394E+000 1.006256E-003

50000 1.345294E+000 1.094242E-003

100000 1.346046E+000 3.418636E-004

200000 1.345345E+000 1.043129E-003

500000 1.345938E+000 4.501515E-004

1000000 1.346882E+000 4.943130E-004

2000000 1.346327E+000 6.075571E-005

5000000 1.346425E+000 3.665767E-005

**Примечание.** Код демонстрационной программы смотреть в приложении 1. Полный код доступен на https://github.com/SWATOPLUS/MathModeling

Как видно из результатов, симметризация дает большую точность при одинаковом количестве используемых БСВ.

**Приложение 1. Код демонстрационной программы на языке C#**

using System;

using System.Globalization;

using MathModeling.MonteCarlo;

namespace Lab3MonteCarlo

{

class Program

{

static Program()

{

CultureInfo.DefaultThreadCurrentCulture = new CultureInfo("en-us");

CultureInfo.DefaultThreadCurrentUICulture = new CultureInfo("en-us");

}

private static void Main()

{

var realValue = 1.3463879568;

var repeats = 1;

var sampleSizes = new[]

{

10, 20, 50,

100, 200, 500,

1000, 2000, 5000,

10000, 20000, 50000,

100000, 200000, 500000,

1000000, 2000000, 5000000

};

var calculator = new MonteCarloIntegralCalculator(-1, 3, x => Math.Exp(-x \* x) \* Math.Cos(x));

Console.WriteLine($"Real value {realValue}");

Console.WriteLine("Simple method");

Console.WriteLine("N\t result\terror");

foreach (var sampleSize in sampleSizes)

{

var integral = calculator.CalculateIntegral(sampleSize);

var error = Math.Abs(integral - realValue);

Console.WriteLine($"{sampleSize.ToString().PadRight(10)}{integral:E}\t{error:E}");

}

Console.WriteLine("Symmetrization method");

Console.WriteLine("N\t result\terror");

foreach (var sampleSize in sampleSizes)

{

var integral = calculator.CalculateIntegralWithSymmetrization(sampleSize);

var error = Math.Abs(integral - realValue);

Console.WriteLine($"{sampleSize.ToString().PadRight(10)}{integral:E}\t{error:E}");

}

}

}

}