МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа 3

Выполнил:
Анищенко Арсений
4 курс 3 группа
Преподаватель:
Кирлица Валерий Петрович

Минск

2019

Условие

Вариант 9

$$I = \int_{0}^{9} \frac{cos^{2} x}{1 + x^2} dx$$

Вычислить используя простейший метод Монте-Карло и метод симметризации подынтегральной функции.

Ход работы

Теория

Базовую случайную величину можем смоделировать используя генератор Макларена — Марсальи, основанного на мультипликативном конгруэнтном методе:

$$\alpha_i = \alpha_i{}' \ / M$$

$$\alpha_i{}' = \beta\alpha\_i - 1' \ mod M$$

$$i = 1, 2, ...$$

$$\alpha_0{}' = \beta = 65539$$

$$M = 2147483648$$

Методом обратного преобразования получим равномерно распределенную случайную величину:

$$R(a, b) = y(b - a) + a$$

Обычный алгоритм Монте-Карло интегрирования

Предположим, требуется вычислить определённый интеграл

$$\int_{a}^{b} f(x)\, dx$$

Рассмотрим случайную величину u, равномерно распределённую на отрезке интегрирования [a,b]. Тогда f(u) также будет случайной величиной, причём её математическое ожидание выражается как

$$\mathbb{E}f(u) = \int_a^b f(x)\varphi(x)\,dx,$$

но математическое ожидание случайной величины f(u) можно легко оценить, смоделировав эту случайную величину и посчитав выборочное среднее.
Итак, бросаем N точек, равномерно распределённых на [a,b] для каждой точки ui вычисляем f(ui). Затем вычисляем выборочное среднее:

$$\frac{1}{N}\sum_{i=1}^{N} f(u_i)$$

В итоге получаем оценку интеграла:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(u_i).$$

Точность оценки зависит только от количества точек N.
Этот метод имеет и геометрическую интерпретацию. Он очень похож на описанный выше детерминистический метод, с той разницей, что вместо равномерного разделения области интегрирования на маленькие интервалы и суммирования площадей получившихся «столбиков» мы забрасываем область интегрирования случайными точками, на каждой из которых строим такой же «столбик», определяя его ширину как $\frac{b-a}{N}$ и суммируем их площади.

Для вычисления интеграла используем простой метод Монте-Карло. Генерируем равномерно распределенную случайную величину на промежутке интегрирования (0, 9). И находим значение подынтегральной функции в данной точке.
Также используем метод симметризации подынтегральной функции. Для этого для смоделированной случайной величины вычисляем значение симметричной подынтегральной функции:

$$\frac{1}{2}\left(\frac{\cos^2 x}{1+x^2} + \frac{\cos^2(9-x)}{1+(9-x)^2}\right)$$

Интеграл: $\int_0^9 \frac{cos^2 x}{1+x^2}dx$

$$Ef(u) = \int_0^9 \frac{cos^2 x}{(1+x^2)(9-0)}dx$$

$$\int_0^9 \frac{\cos^2 x}{1+x^2}dx = 9Ef(u) = 9E(\frac{\cos^2 u}{1+u^2})$$

$$\int_0^9 \frac{\cos^2 x}{1+x^2}dx \approx 9/N \sum_{i=1}^{N}(\frac{\cos^2 u_i}{1+u_i^2})$$

Реализация

```
/// <summary>
/// MacLaren-Marsaglia generator for base random variable
/// </summary>
3 references
public class MGenerator
{
    private double _alpha;
    private double _beta;
    private double _m;

    1 reference
    public MGenerator(int seed = 65539)
    {
        _alpha = seed;
        _beta = 65539;
        _m = 2147483648;
    }

    1 reference
    public double NextRand()
    {
        _alpha = (_alpha * _beta) % _m;

        return _alpha / _m;
    }
}
```

Класс реализующий генерацию базовой случайной величины методом Макларена — Марсальи.

```
/// <summary>
/// Uniform distributed random generator
/// </summary>
2 references
public class UniformGenerator
{
    private readonly MGenerator _mGenerator;

    3 references
    public double A { get; private set; }
    2 references
    public double B { get; private set; }

    1 reference
    public UniformGenerator(double a, double b, int seed = 65539)
    {
        _mGenerator = new MGenerator(seed);

        A = a;
        B = b;
    }

    1 reference
    public double NextRand()
    {
        var baseVariable = _mGenerator.NextRand();

        return baseVariable * (B - A) + A;
    }
}
```

Класс реализующий генерацию св с равномерным распределением.

```
0 references
class Program
{
    private const double A = 0;
    private const double B = 9;

    private delegate double Func(double x);

    3 references
    static double Function(double x)
    {
        return Math.Pow(Math.Cos(x), 2) / (1 + Math.Pow(x, 2));
    }

    1 reference
    static double SymmetryFunction(double x)
    {
        return (Function(x) + Function(A + B - x)) / 2;
    }

    2 references
    static double DoMontecarlo(Func func, int samples = 10000)
    {
        var uniformGenerator = new UniformGenerator(A, B);
        var ans = 0.0;

        for (int i = 0; i < samples; ++i)
        {
            ans += func(uniformGenerator.NextRand());
        }

        return ans / samples * (B - A);
    }

    0 references
    static void Main(string[] args)
    {
        Console.WriteLine($"Function = {DoMontecarlo(Function)}");
        Console.WriteLine($"Symmetry function = {DoMontecarlo(SymmetryFunction)}");

        Console.ReadKey();
    }
}
```

Тестовый класс, реализующий простой метод Монте-Карло и метод симметризации подынтегральной функции.

Результат

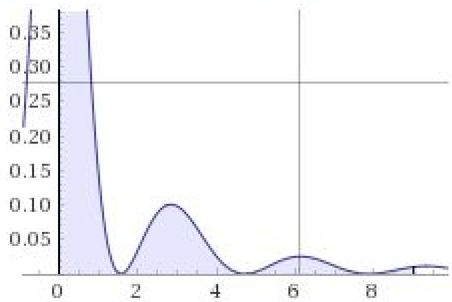
Function = 0,826125212539704
Symmetry function = 0,828573743552735

Полученный при помощи wolframalpha:

Definite integral:

$$\int_0^9 \frac{\cos^2(x)}{1+x^2}\,dx = 0.833897$$

Visual representation of the integral:



Enlarge | Data | Customize