Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тверской государственный университет»

Факультет прикладной математики и кибернетики
Направление «Фундаментальная информатика и информационные
технологии»
Кафедра математической статистики и системного анализа

Курсовая работа

по дисциплине
«Теория вероятностей и математическая статистика»

Направление: 02.03.02 – «Фундаментальная информатика и информационные
технологии»

Вариант №19

Выполнила:
Студентка 26 группы
Демьянова Виктория Вадимовна
Научный руководитель: к.ф.-м.н.
Захарова Ирина Владимировна

Тверь 2016 г.

Оглавление

Описание метода статистического моделирования (метод Монте Карло)

Метод Монте-Карло— общее название группы численных методов, основанных на получении большого числа реализаций стохастического (случайного) процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи

Интегрирование методом Монте-Карло. Геометрический алгоритм Монте-Карло.

Предположим, необходимо взять интеграл от некоторой функции.
Воспользуемся неформальным геометрическим описанием интеграла и будем понимать его как площадь под графиком этой функции.
Для определентия этой площади можно воспользоваться одним из обычных численных методов интегрирования: разбить отрезок на под отрезки, подсчитать площадь под графиком функции на каждом из них и сложить.
Для определения площади под графиком функции можно использовать следующий стохастический алгоритм:

1. Ограничим функцию прямоугольником (n-мерным параллелепипедом в случае многих измерений), площадь которого S{par} можно легко вычислить; любая сторона прямоугольника содержит хотя бы 1 точку графика функции, но не пересекает его;
2. «Набросаем» в этот прямоугольник (параллелепипед) некоторое количество точек (N штук), координаты которых будем выбирать случайным образом;
3. Определим число точек (K штук), которые попадут под график функции;
4. Площадь области, ограниченной функцией и осями координат, S даётся выражением $S = S_{par}\dfrac{K}{N}$

Постановка задачи

Написать программу, рассчитывающую площадь фигуры.
На входе: число наблюдений – n + дополнительные параметры.
На выходе: площадь фигуры и относительная погрешность.
<u>Вариант 19</u> Фигурой является круг радиуса R.

Входной параметр также: $R > 0$.

Замечания

Замечание 1: Если случайная величина ξ имеет равномерное распределение на отрезке[0,1], то случайная величина η =a+(b-a) ξ имеет равномерное распределение на отрезке [a,b].

Замечание 11: Пусть необходимо найти площадь некоторой области D

Для этого:

1. Строим прямоугольник, со сторонами [a,b] и [c,d], содержащий область D

2. Рассмотрим случайные величины ξ и η, где случайная величина ξ имеет равномерное распределение на отрезке [a,b]; случайная величина η имеет равномерное распределение на отрезке [c,d]. Введём случайную величину

$$\varphi = \begin{cases} 1, \text{если} \quad (\xi, \eta) \in D \\ 0, \text{иначе} \end{cases}$$

Вычислим математическое ожидание случайной величины φ

$$M\varphi = 1 \cdot P((\xi,\eta) \in D) + 0 \cdot P((\xi,\eta) \notin D) = P((\xi,\eta) \in D) = \frac{S_D}{S_{abcd}} \ ,$$

где S_D, S_{abcd} – площади области D и прямоугольника abcd, соответственно. При вычислении вероятности мы воспользовались геометрическим определением вероятности, поскольку случайный вектор имеет равномерное распределение в области D. Следовательно, $S_D = S_{abcd} * M_\varphi$

Пусть (x_1,y_1), … ,(x_n,y_n) – полученные(«наблюдённые») значения случайного вектора (ξ, η). Таким образом мы имеем ***n*** измерений случайной величины φ:$z_1 \ldots z_n$, где

$$z_i = \begin{cases} 1, \text{если} \quad (x_i, y_i) \in D \\ 0, \text{иначе} \end{cases}$$

i=1,…,n. Отсюда получаем

$$S_D \approx S_{abcd} * \frac{\sum_{i=1}^{n} z_i}{n}$$

Код программы

```
#include <iostream>
#include <math.h>
#include <locale.h>
using namespace std;
int R;
double a;
double b;
double c;
double d;
//Функция, которая проверяет принадлежит ли точка (x,y) области D
bool in_round(double x, double y)
{
	if (x*x + y*y <= R * R)
	{
		return true;
	}
	return false;
}
//функции, которые генерируют случайную величину с равномерным распределением на
[0,1] и переводят её в случайную величину на отрезке [a,b]
double randomx() { return (double(rand()) / double(RAND_MAX))*(b - a) + a; }
double randomy() { return (double(rand()) / double(RAND_MAX))*(d - c) + c; }
//Подсчёт площади D
double square(int n)
{
	double answer = 0.0;
	for (int i = 0; i < n; i++)
	{
		if (in_round(randomx(), randomy()))answer += 1.0;
	}
	return answer*4*R*R / double(n);
//Площадь abcd = 4 * R * R

}

int main()
{
	cin >> R;
	double const pi = 3.1415926535;
	a = -R;
	b = R;
	c = -R;
	d = R;
	setlocale(0, "");
	int n;
	cin >> n;
	double res = square(n);
	cout<< "При числе наблюдений n=" << n << " Площадь фигуры= " << res << "
Погрешность= " << abs(pi*R*R - res) << endl;
	return 0;
}
```

Результаты работы программы

При радиусе R = 1 и числе наблюдений n = 7. Площадь фигуры = 3,42857
Погрешность = 0,286979

При радиусе R = 1 и числе наблюдений n = 20. Площадь фигуры = 3,2
Погрешность = 0,0584073

При радиусе R = 1 и числе наблюдений n = 50. Площадь фигуры = 3,36
Погрешность = 0,218407

При радиусе R = 4 и числе наблюдений n = 100. Площадь фигуры = 52,48
Погрешность = 2,21452

При радиусе R = 4 и числе наблюдений n = 1000. Площадь фигуры = 51,328
Погрешность = 1,06252