

## Метод Монте-Карло ЛАБОРАТОРНАЯ РАБОТА

<http://www.kimrt.ru>

Онлайн-курс

[Суперкомпьютерные технологии в задачах моделирования](#)

Методы Монте-Карло – группа методов, основанных на генерации относительно большого числа случайных чисел и, как правило, на проведении большого числа вычислительных испытаний с дальнейшим усреднением результата. Эти методы используются для решения задач в различных областях физики, химии, математики, экономики, оптимизации, теории управления и других [1]. Название метода происходит от района Монте-Карло в княжестве Монако, известного своими казино. Ведь именно рулетка является одним из самых широко известных генераторов случайных чисел.

Рассмотрим простой пример, связанный с приближенным вычислением числа  $\pi$ . В этом примере в квадрат со стороной  $a = 2$  вписан круг с радиусом  $R = 0.5a = 1$ . Пусть центр круга соответствует декартовым координатам  $x = 0$  и  $y = 0$ . Если в квадрат случайным образом забросить  $N$  точек, то при условии, что  $N \rightarrow \infty$ , получим соотношение  $N_c / N \approx S_c / S \approx \pi R^2 / a^2 \approx 0.25 \pi$ . Здесь  $N_c$  – число точек, попавших в круг,  $S_c$  – площадь круга и  $S$  – площадь квадрата. Таким образом, из этого соотношения можно выразить  $\pi$  или  $S_c$  для приближенного расчета. Для простоты можно рассмотреть четверть квадрата и четверть круга в области  $\{x=[0..1], y=[0..1]\}$ . Условие для определения попадания точки в круг следующее,

$$r = \sqrt{x^2 + y^2} \leq R,$$

где  $r$  – радиальная координата в полярной системе координат. Алгоритм расчета числа  $\pi$  методом Монте-Карло представлен в листинге 1.

### Листинг 1. Алгоритм расчета числа $\pi$

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <math.h>
#include <random>
#include <ctime>
#define a 2.0 //сторона квадрата
using namespace std;

int main() {
```

<http://www.kimrt.ru>

```
cout << "Program calculates PI by Monte Carlo
method.\n";
//общее количество точек
long int NumPoint = 0;
cout << "Enter number of points: ";
cin >> NumPoint;
//проверка ввода
if ((cin.good()==false) || (NumPoint < 100)) {
    cout<<"\nWrong number or it's less than 100 or
too big!"
        <<endl;
    cout << "Program is finished." << endl;
    system("pause");
    return 1;
}
clock_t jobTime = clock();//время работы
//число точек внутри четверти круга
long int NumPointInCirc = 0;
double R2 = 0.25 * a * a;//квадрат радиуса круга
std::mt19937 gen(time(0));//генератор случайных чисел
std::uniform_real_distribution<> rnd(0, 1);
for (long int i=0; i < NumPoint; i++){
    double x = rnd(gen);
    double y = rnd(gen);
    if ((x*x + y*y)<= R2) NumPointInCirc++;
}
double pi = 4.0*NumPointInCirc/NumPoint;
cout << "Calculated PI= " << pi << endl;
cout << "Absolute error= " << abs(pi-M_PI)/M_PI <<
endl;
//время в тактах микропроцессора
jobTime = clock() - jobTime;
cout<<"Execution time:
"<<(double)jobTime/CLOCKS_PER_SEC
    << " seconds" << endl;
system("pause");
return 0;
}
```

### Задание

Напишите программу приближенного определения объема шара с радиусом  $R = 1$ , вписанного в куб со стороной  $a = 2$ , методом Монте-Карло. Программа запрашивает количество случайных чисел  $N$  и выводит на экран

приближенный ответ и погрешность. При каком значении  $N$  можно получить погрешность менее 5%?

[http://www.kimrt.ru/index/course\\_stm/0-24](http://www.kimrt.ru/index/course_stm/0-24)

*Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.*

#### **Источники**

1. Monte Carlo method. [https://en.wikipedia.org/wiki/Monte\\_Carlo\\_method](https://en.wikipedia.org/wiki/Monte_Carlo_method)