Муниципальное образовательное учреждение
«Средняя общеобразовательная школа №51 г.Твери»

Случайные числа
Метод Монте-Карло
или
метод статистических испытаний

учитель информатики и ИКТ
Цирулева Татьяна Владимировна

Тверь
2016

Содержание

1. Введение

Название метод Монте-Карло обозначает обширную группу численных методов и алгоритмов, основанных на использовании больших последовательностей случайных вещественных или целых чисел, генерируемых компьютером. В настоящее время метод Монте-Карло занимает важное место в вычислительной математике и информатике и бурно развивается как в теоретическом направлении, так и в области вычислительных алгоритмов. Он используется для решения задач в различных областях естественных наук, математики, экономики и статистики.

Математическое обоснование метода Монте-Карло требует весьма серьезной подготовки в области высшей математики, в частности, в области теории вероятностей и математической статистики. Однако идея метода интуитивно очень понятна, она может быть объяснена на бытовом языке и проиллюстрирована на простых, но вполне содержательных примерах. Изучение темы позволит в средней школе активно использовать метод Монте-Карло для решения простейших задач на домашнем компьютере. Изучение идеи метода также естественно привлекает внимание и возбуждает интерес к важному вопросу о соотношении случайного и закономерного в природе, в частности, к вопросу о том, как случайное помогает выявить закономерное.

Источники, на основе которых изучается тема, являются или изначально полностью популярными [1,2], или [3,4] содержат главы (указанные в библиографическом описании), в основном не требующие знания высшей математики. Популярные источники могут быть использованы также для первоначального, но более детального изучения метода Монте-Карло, а работы [3,4] – для заинтересованных и продвинутых учащихся. В приложении приведен листинг программ на языке Pascal для

простейших рассмотренных примеров – вычисления числа π и вычисления определенного интеграла.

2. История и сущность метода Монте-Карло

Применение метода Монте-Карло без использования быстродействующих компьютеров весьма ограничено и, по сути, бесперспективно. Поэтому серьезное развитие данного метода началось только около шестидесяти лет назад, когда производство вычислительных машин приобрело промышленное значение. Первоначально метод Монте-Карло возник и использовался главным образом для решения задач нейтронной физики, связанных с созданием атомной бомбы, где традиционные численные методы оказались малопригодными. Создателями метода считают американских математиков Дж. Неймана и С. Улама, первая работа которых вышла в 1949г. В Советском Союзе первые статьи о методе Монте-Карло были опубликованы в 1955 г. Почти сразу его влияние распространилось на широкий круг задач статистической физики, очень разных по своему содержанию [2,3]. В настоящее время имеется более трех тысяч работ, в которых рассматриваются математические основы метода и его приложения к конкретным задачам. Название методу дал тот факт, что случайные числа можно получать с помощью рулетки, что на самом деле и происходит в казино города Монте-Карло (столица княжества Монако).

Последовательность чисел является случайной, если между составляющими ее числами нет зависимостей: в математике эти зависимости называются корреляциями. Полное рассмотрение вопроса о случайности достаточно сложно, но можно дать рассматриваемому понятию случайности следующее простое практическое объяснение.

Пусть, например, компьютер генерирует последовательность N чисел на промежутке $[0,1)$; во всех современных языках программирования высокого уровня есть датчики (операторы) таких случайных чисел

(например, random в Pascal, rand() в C, Random.NextDouble() в C#). Тогда – при больших N – последовательность считается случайной, если в любом интервале длиной $d < 1$ из указанного промежутка содержится примерно одинаковое число $n(d)$ членов последовательности. Более точно это значит, что хотя числа $n(d)/N$ слегка различны для разных интервалов одной и той же длины d, но их разность для любой пары таких интервалов стремится к нулю при увеличении N. Это рассуждение без труда можно распространить на любые промежутки вещественных чисел, а также плоские фигуры, трехмерные и многомерные области.

Суть метода Монте-Карло в общем можно сформулировать следующим образом. В методе строится последовательность случайных чисел или наборов случайных чисел. На каждом шаге появления элементов последовательности проверяется ряд условий, которые позволяют или не позволяют соотнести это число (набор чисел) последовательности с некоторым событием; в теории вероятностей это событие называется благоприятным исходом. В связи с этим метод Монте-Карло часто называют методом статистических испытаний или статистическим моделированием, подчеркивая, что применение метода не предполагает изначально знание математических связей и позволяет получить их на основе многократного наблюдения (компьютерной генерации) случайных событий в представленной модели. На последнем этапе вычисляется отношение числа благоприятных исходов к общему числу шагов N и полученное значение связывается (непосредственно или как значение некоторой функции) со значением некоторой величины в рассматриваемой (физической, математической, экономической) задаче. Это слишком общее объяснение иллюстрируется конкретными примерами в разделах 4 – 6.

Особенностью метода Монте-Карло является то, что добиться высокой точности на таком пути невозможно. Поэтому обычно считается, что метод Монте-Карло эффективен при решении тех задач, в которых результат нужен

с небольшой точностью, например, в пределах 0,01 (а может быть и 10^{-5}, как при вычислении числа π в разделе 4).

3. Генераторы случайных чисел

Генераторы случайных чисел разделяются на алгоритмические и аппаратные, а также смешанные, которые сочетают в себе методы и алгоритмического, и аппаратного получения последовательностей случайных чисел. Хорошее свойство алгоритмических генераторов заключается в быстроте их работы, а недостаток в том, что получается последовательность «не совсем случайных» чисел, поэтому эти числа называют псевдослучайными, а алгоритмические генераторы часто называют генераторами псевдослучайных чисел. Для аппаратных генераторов ситуация в точности обратная: они очень медленные, но «случайность» чисел последовательности хорошая.

Алгоритмические методы заключаются в последовательных вычислениях случайных чисел с помощью системы математических формул, т.е. конкретного численного алгоритма. К наиболее популярным методам, генерирующим натуральные числа, относится, например, метод чисел Фибоначчи с запаздыванием, разработанный еще в 1958 г. В этом алгоритме каждое следующее число вычисляется через два предыдущих по формуле

$$X(n) = \big(X(n-24) + X(n-55)\big) mod(16), \qquad n = 56{,}57, \dots, N,$$

где символ $mod(16)$ означает, что берется остаток от деления на 16 суммы чисел в круглых скобках, а первые 55 чисел выбираются специальным образом.

Однако еще в середине двадцатого века А.Н. Колмогоров доказал, что невозможно построить алгоритм, который генерирует случайную последовательность натуральных чисел. Поскольку любой компьютер работает с дискретными последовательностями чисел, то отсюда следует, что

никакую последовательность вещественных (и любых других) случайных чисел невозможно построить алгоритмически. У любого алгоритма существует рабочий период N_{max}: для $N > N_{max}$ числа последовательности начинают, в некотором точном смысле, существенно повторяться. Для рассмотренного выше алгоритма рабочий период не превышает нескольких миллионов.

Под аппаратным генератором случайных чисел обычно понимается некоторое устройство и схема генерации, связанные с компьютером и способные по специальной команде (оператор обращения к генератору случайных чисел) выдавать с машинной точностью случайное число из некоторого непрерывного интервала вещественных чисел или из некоторого дискретного множества. В принципе, простейшим аппаратным генератором случайных чисел от 1 до 6 является игральная кость, у которой рабочий период бесконечен. В настоящее время имеется много способов чисто компьютерной аппаратной генерации случайных чисел. Например, устройством может быть процессор, а схемой генерации – отслеживание случайных колебаний температуры процессора в установившемся режиме работы.

4. Вычисление числа π методом Монте-Карло

При вычислении числа π методом Монте-Карло круг диаметра 1 впишем в квадрат со стороной 1, так что центр круга расположен в точке (0.5,0.5), а уравнение имеет вид

$$(x - 0{,}5) \cdot (x - 0{,}5) - (y - 0{,}5) \cdot (y - 0{,}5) < 0{,}25. \qquad (1)$$

Программа в цикле посредством команд $x := random; y := random;$ генерирует последовательность N случайных точек (x, y), расположенных внутри квадрата. Условием благоприятного исхода, о котором шла речь в общем описании метода Монте-Карло в разделе 2, является попадание точки

в круг. Если число попаданий в круг равно Nc, то число Nc/N будет приблизительно равно отношению площадей:

$$\frac{Nc}{N} = \frac{S\text{круг}}{S\text{квадр}}.$$

В нашем случае $S\text{круг} = \pi/4$, $S\text{квадр} = 1$, поэтому $\pi = 4 * Nc/N$ (Рис. 1). При $N = 100000$ находим π = 3.14452.

Наилучший результат π = 3.1415 достигается при $N = 10000000$.

Листинг программы приведен в Приложении А.

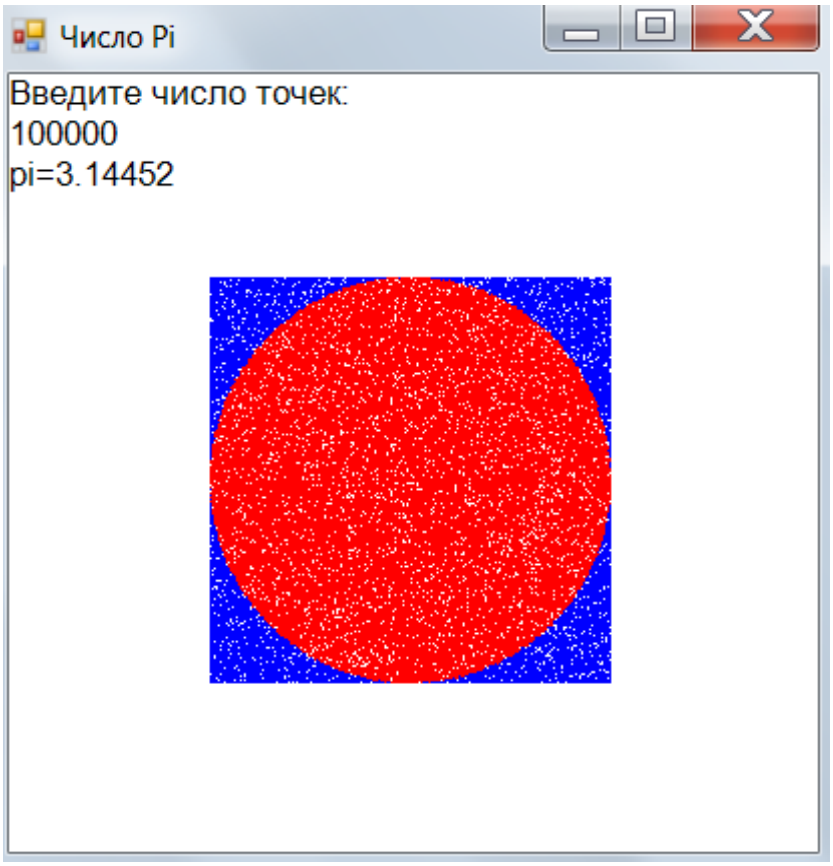

Рис. 1. Вычисление числа π. Здесь $N = 100000$.

5. Вычисление определенных интегралов

Рассмотрим применение метода Монте-Карло к вычислению определенных интегралов. Фактически вычисление определенного интеграла

$$\int_a^b f(x)dx$$

сводится к вычислению площади фигуры, ограниченной осью x, вертикальными прямыми $x = a,\ x = b,$ а также графиком функции (см. Рис.2). Для этого построим прямоугольник, ограниченный прямыми

$$x = a, \qquad x = b, \qquad y = 0, \qquad y = max(f(x), a < x < b).$$

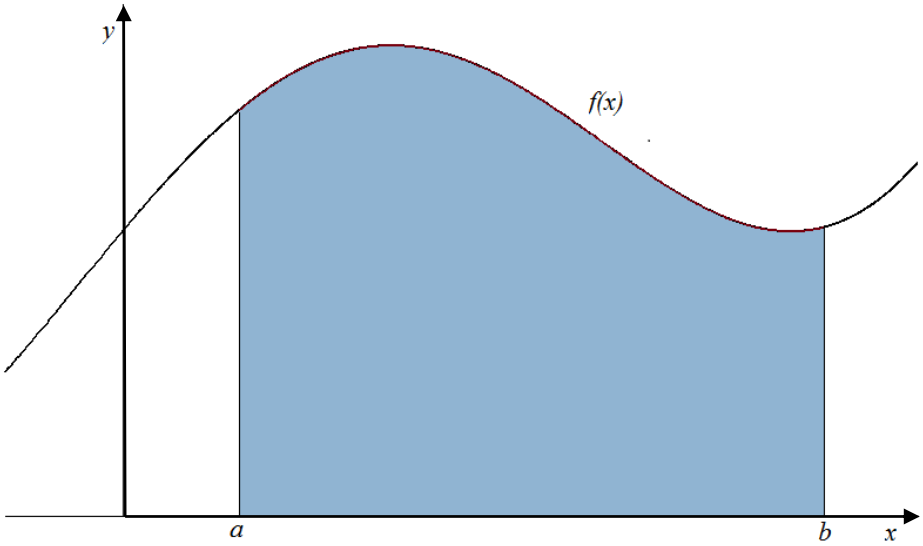

Рис.2. Значение интеграла равно площади фигуры под графиком функции.

Программа в цикле посредством команд

$x := random * (b - a) + a;\quad y := random * max(f(x), a < x < b);$

генерирует последовательность N случайных точек (x, y), расположенных внутри прямоугольника. Если число попаданий в фигуру под графиком функции равно Nc, то число Nc/N будет приблизительно равно отношению площадей фигуры (значение интеграла) и прямоугольника.

Для иллюстрации метода в Приложении Б вычисляется интеграл

$$\int_0^2 (x^2 - x + 1)dx.$$

Здесь $a = 0, b = 2, max(f(x)) = 3$. Точное значение интеграла равно 2.66(6), а при $N = 10000$ случайных испытаний его значение, вычисленное методом Монте-Карло равно 2.664 (см. Рис. 3)

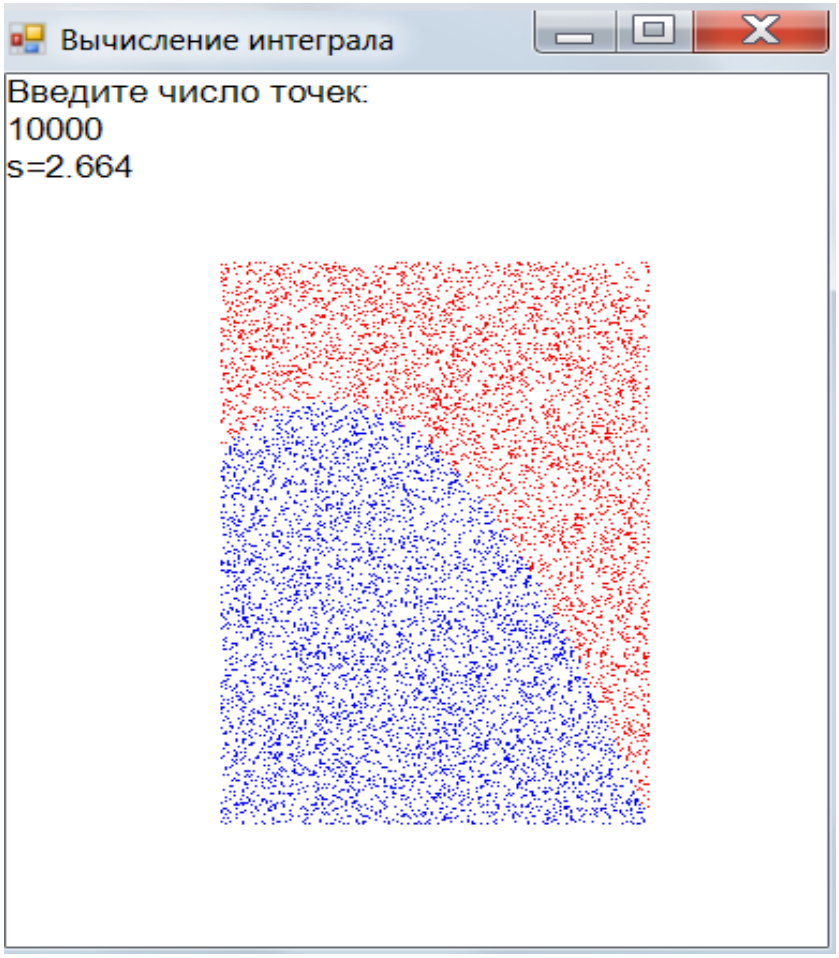

Рис. 3. Значение интеграла равно отношению Nc/N числа благоприятных исходов к общему числу испытаний, умноженному на площадь прямоугольника.

6. Заключение

Изучение метода Монте Крло на уроке в популярной форме позволяет рассмотреть вопросы компьютерной генерации последовательностей равномерно распределенных случайных чисел и примеры применения метода Монте-Карло (метода статистических испытаний) для решения детерминированных задач. На самом деле, как доказывается в цитируемых источниках [3,4], метод Монте-Карло позволяет моделировать любой процесс, на протекание которого влияют случайные факторы. Таким образом, можно говорить о методе Монте-Карло как об универсальном методе решения стохастических задач из различных областей естествознания и экономики. Общим недостатком метода является невысокая точность получаемых результатов.

Рассмотренные примеры легко распространить на другие классы задач, такие как вычисление площадей и многомерных объемов областей сложной формы, решение систем неравенств, а также других задач, допускающих геометрическую формулировку.

7. Список литературы

1. Соболь И.М. Метод Монте-Карло. Популярные лекции по математике. Выпуск 46. М.: Наука, 1985, 78 стр.
2. Паньгина Н.Н., Паньгин А.А. Статистическое моделирование: метод Монте-Карло. Компьютерные инструменты в образовании, № 5, 2002
3. Соболь И.М. Численные методы Монте-Карло. М.: Наука, 1973, 378 стр. (Глава I, Глава VI).
4. Бусленко Н.П. и др. Метод статистических испытаний (метод Монте-Карло). М.: Физматлит, 1962 (Глава I).

9. Приложения

Приложение А: листинг программы для вычисления числа π (Pascal)

```
program Monte_Karlo;
uses graphabc;
var
N,Nc,k,a,b: longint;
    x,y,Pi: real;
begin
  writeln('Введите число точек: ');
  readln(N);
  writeln(N);
  setwindowsize (640,480);
  randomize;   // запускаем генератор случайных чисел
  Nc := 0;
  for k:=1 to N do
    begin
     x:=random; // задаем случайную точку (x,y)
     y:=random; // в квадрате 0<x<1, 0<y<1
     if((x-0.5)*(x-0.5)+(y-0.5)*(y-0.5) <= 0.25)
     then
     begin
      Nc := Nc + 1;
      setwindowtitle ('Число Pi');
      a:=trunc(x*100)+100;
      b:=trunc(y*100)+100;
       SetPixel(a,b,clRed);
     end
     else
     begin
      a:=trunc(x*100)+100;
      b:=trunc(y*100)+100;
      SetPixel(a,b,clBlue);
    end;
    end;
    Pi := 4*Nc/N;
    writeln('pi=',Pi);
    end.
```

Приложение Б: листинг программы для вычисления $\int_0^2 (x^2 - x + 1)dx$

```
program interal;
uses graphabc;
var
  Nc,N,k,a,b:longint;
  s,x,y:real;
begin
  writeln('Введите число точек: ');
  readln(N);
  writeln(N);
  setwindowsize (640,480);
  randomize;   // запускаем генератор случайных чисел
  Nc := 0;
  for k:=1 to N do
    begin
     x:=random*2; // задаем случайную точку (x,y)
     y:=random*3; // в квадрате 0<x<2, 0<y<2
     if(y < x*x - x+1)
     then
     begin
     Nc := Nc + 1;
     setwindowtitle ('Вычисление интеграла');
     a:=trunc(x*100)+100;
     b:=trunc(y*100)+100;
    SetPixel(a,b,clRed);
     end
     else
     begin
     a:=trunc(x*100)+100;
     b:=trunc(y*100)+100;
    SetPixel(a,b,clBlue);
     end;
     end;
  s :=Nc/N*6;
  writeln('s=',s);
end.
```