

5. Использование генератора случайных чисел при моделировании физических систем

Основные свойства генератора случайных чисел

Стандартный генератор случайных чисел осуществляет равновероятную выборку из заданного множества чисел. Например, для задачи типа «орел-решка» нам потребуется генератор, который на каждом шаге выбрасывает одно из двух чисел – ноль или единицу – с одинаковой вероятностью.

В общем случае, генератор случайных чисел применяется к таким физическим задачам, в которых необходимо обеспечить случайную реализацию какого-либо физического параметра, моделирующего вклад сложного и не поддающегося прямому расчету физического явления. К таким задачам относятся задачи о движении частиц и распространении волн в случайно неоднородных средах, задачи статистической физики о взаимодействии подсистемы с модельным термодинамическим ансамблем и т.д. Метод моделирования с помощью генератора случайных чисел (включающий статистическую обработку результатов) называется методом Монте-Карло.

Во всех языках высокого уровня существует стандартный генератор случайных чисел, вызов которого (например, в Паскале) имеет вид $x:=\text{random}$. В других языках может использоваться сочетание rnd . При этом переменной x присваивается числовое значение из диапазона $[0, 1]$, выбранное абсолютно случайно, или, другими словами, появление любого числа из этого диапазона равновероятно. Такой генератор реализует случайную выборку из непрерывной совокупности, а соответствующая ему функция распределения $\varphi(x)$ негауссова ($\varphi(x)$ равна 1 в области значений $0 < x < 1$ и 0 для всех других x).

Если требуется генератор, возвращающий случайное число из дискретного набора, его без проблем можно получить из стандартного генератора, добавив несколько условных операторов, разбивающих множество $[0, 1]$ на равные части. Например, случайный выбор из набора двух чисел $\{-1, 1\}$ для задачи типа «орел-решка» обеспечивает следующая вставка:

```
X:= random;  
if (X < 0.5) then N:= -1 else N:=1};
```

Если случайное число X оказывается меньше 0.5, то переменная N принимает значение (-1) , если $0.5 < X < 1$, то $N=1$. Таким образом, N

является случайным числом с равной вероятностью (равной 0.5), принимающим либо значение (-1), либо значение 1.

Заметим, что во многих языках программирования для таких целей существует дополнительный генератор, в частности в Паскале это функция `rnd(M)`, значением которой является случайное целое число из набора целых чисел $\{0, 1, \dots, M-1\}$.

Прежде чем двигаться дальше, нужно четко определить смысл термина «случайное число». Пусть при очередном вызове `x:=random` появляется число 1. В самом числе 1 нет ничего случайного. Конечно, глядя на программу, мы понимаем, что появление именно этого значения осуществлялось с помощью случайной выборки. Но что, если бы текст программы был недоступен?

Конкретное значение, которое выводится на экран с помощью генератора случайных чисел, называется «реализацией случайного числа». Нам необходимо набрать большое количество этих реализаций. Проанализировав статистические свойства полученной таким образом числовой последовательности, можно сделать вывод о случайном или неслучайном механизме появления рассматриваемых чисел.

Прежде всего, можно вычислить среднее значение и среднеквадратичное отклонение (по формулам среднего арифметического), а также функцию распределения значений этого набора чисел. После чего можно сравнить их с аналогичными параметрами, полученными в теории случайных величин с предполагаемой для данной статистики функцией распределения.

Например, идеальная форма функции распределения для случая `x:=random` описывается формулой $\varphi(x) = \theta(x) - \theta(x-1)$, где $\theta(x)$ – тэта-функция («единичная ступенька»: $\theta(x)=1$ при неотрицательном аргументе x и $\theta(x)=0$ при отрицательном аргументе). Вычисляя средние значения как интеграл по вещественной оси от усредняемой функции, умноженной на функцию распределения $\varphi(x)$, получаем

$$\langle x \rangle = \frac{1}{2}, \quad d^2 = \langle (x - \langle x \rangle)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12} \quad (5.1)$$

В данном случае среднее от любой величины (обозначено угловыми скобками $\langle \dots \rangle$) совпадает с обычным интегралом от этой же величины, взятом на интервале $[0,1]$. После этого можно сравнить найденные теоретические значения с результатами усреднения по набору сгенерированных программой чисел. Для того чтобы вычислить аналогичные средние по набору реализаций, можно дополнить программу следующим набором операторов

```

Mx:=0; Dx:=0;
for i:=1 to N do
begin
x:= random;
Mx:=Mx+x;
Dx:=Dx+x*x
end;
Mx:=Mx/N ;   d:= sqrt((Dx/N)-Mx*Mx);

```

Здесь Mx , d – среднее значение и среднеквадратичное отклонение соответственно, количество реализаций N должно быть большим. Очевидно, что получаемое таким образом Mx не равно в точности $1/2$, так как само также является случайным числом. Если вычислить по этому алгоритму несколько значений Mx при фиксированном N , то окажется, что они флуктуируют вокруг теоретического значения $1/2$. При этом среднее отклонение от $1/2$ будет стремиться к нулю с ростом N по закону $\Delta \sim (1/N)^{1/2}$ (в соответствии с центральной предельной теоремой).

Что можно моделировать с помощью генератора случайных чисел? Например, эксперимент по измерению какой-либо физической величины. Получение экспериментального значения можно моделировать как реализацию случайного числа $x = \langle x \rangle + \Delta x$, где Δx – случайная добавка с выбранными статистическими свойствами. Более сложные примеры рассмотрены в следующих разделах.

Задача о случайных блужданиях

Задача о случайных блужданиях возникает, в частности, при моделировании броуновского движения. Существует несколько формулировок такого рода моделей. Например, задача о случайных блужданиях частицы по двумерной квадратной решетке формулируется следующим образом:

1) частица совершает последовательные шаги по соседним узлам решетки;

2) на каждом шаге она перемещается с занимаемого узла на один из четырех ближайших (эти узлы называются ближайшими соседями);

3) направление перехода выбирается абсолютно случайно.

Таким образом, на каждом шаге частица сдвигается на одну постоянную решетки: или вправо, или влево, или вверх, или вниз от занимаемого узла, причем выбор любого направление перемещения равновероятен. При моделировании методом Монте-Карло выбор одного из направлений осуществляется с помощью генератора

случайных чисел. Полученная таким образом траектория называется случайной реализацией. После этого такие характеристики, как среднеквадратическое смещение, плотность вероятности и т.д., вычисляются с помощью усреднения по большому числу реализаций.

Можно также организовать и движение с переменной длиной шага в любом направлении, что более похоже на броуновское движение, однако оказывается, что статистические свойства этих моделей являются одинаковыми при большом числе шагов, при условии, что средняя длина случайного шага и его дисперсия конечны.

Рассмотрим пример, демонстрирующий моделирование движения броуновской частицы как случайные блуждания точечного объекта.

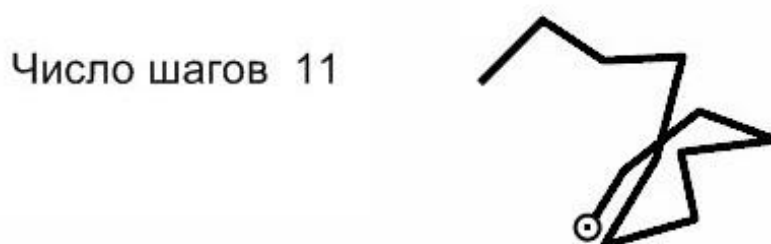


Рис. 5.1. Моделирование траектории броуновского движения с помощью генератора случайных чисел. Начальная точка обозначена кружком

Основная часть программы для построения траектории случайных блужданий на рис. 5.1 состоит из цикла, на каждом шаге которого координата частицы изменяется, получая случайное приращение, то есть частица делает случайный шаг.

1. For i:=1 to N do begin
2. dx:= a*(random - 0.5);
3. dy:= a*(random - 0.5);
4. x:= x+dx;
5. y:= y+dy;
6. xx:= x0+round(x); yy:= y0-round(y);
7. LineTo(xx,yy);
8. end;

В данной программе реализовано случайное блуждание с переменной длиной шага. Средняя длина шага пропорциональна переменной a , проекции декартовых компонент для вектора смещения на один шаг обозначены dx, dy . Строчки 2 – 5 обеспечивают изменение координаты частицы за один шаг.

В результате получается изображение траектории в виде ломаной линии. Экранные координаты точек траектории (вершин ломаной) в программе обозначены переменными $xх, уу$, которые связаны с физическими координатами частицы формулами

$$xх:= x0+round(x); \quad уу:= y0-round(y);$$

Начальное положение частицы на экране соответствует координате $x0, y0$.

Перейдем к рассмотрению основных принципов моделирования броуновского движения в рамках метода Монте-Карло. В простейшем варианте этот метод сводится к вычислению средних значений интересующих нас величин по большому набору случайных реализаций (экземпляров траекторий), каждая из которых приводит к одному значению величины. Эти реализации и создаются генератором случайных чисел.

Вычислим среднее значение квадрата смещение частицы $\langle (r_N)^2 \rangle$, совершившей заданное число шагов N . Под шагом понимается изменение положения частицы за малый интервал времени Δt . Теоретическая формула для среднеквадратического смещения имеет вид

$$\langle \vec{r}_N^2 \rangle = dh^2 N = dh^2 \frac{t}{\Delta t}, \quad (5.2)$$

где d – размерность пространства, h – среднее значение проекции шага на любое направление. В данном случае (в рамках метода Монте-Карло) $\langle \vec{r}_N^2 \rangle$ определяется как среднее значение квадратов смещений \vec{r}_N^2 для большого ансамбля реализаций случайных траекторий, созданных с помощью генератора случайных чисел. Соответствующая формула имеет вид

$$\langle r_N^2 \rangle = \frac{1}{M} \sum_{j=1}^M r_{N,j}^2. \quad (5.3)$$

В формуле (5.3) величина M и есть число реализаций.

В качестве альтернативного варианта рассмотрим блуждания по квадратной решетке, эта задача является статистически эквивалентной исходной. В этом случае частица на каждом шаге с равной вероятностью смещается по одному из четырех возможных направлений, посещая различные вершины решетки. Для блужданий по квадратной решетке ($d=2, h=1$) результат моделирования представлен на рис. 5.2.

Число шагов, $N=100$

Число реализаций, $M=500$

Вычислить

Результат: **203.904**

Теория: **200**

Рис. 5.2. Вычисление среднеквадратического смещения для случайных блужданий по квадратной решетке как среднего по реализациям

Полученное в результате моделирования значение приблизительно соответствует предсказаниям теории, причем точность приближения возрастает с увеличением числа реализаций. При многократном запуске программы получаются различные числа, флуктуирующие вокруг теоретического значения.

Рассмотрим подробно алгоритм моделирования, записанный в виде фрагмента соответствующей программы (без операторов ввода и вывода)

```
1. r:=0;
2. for j:=1 to M do
3. begin
4. x:=0; y:=0;
5.   for i:=1 to N do
6.     begin
7.       if (random>0.5) then dx:=1 else dx:=-1;
8.       if (random>0.5) then dy:=1 else dy:=-1;
9.       x:=x+dx; y:=y+dy;
10.    end;
11. r:=r+(x*x+y*y)/M;
12. end;
```

Построение одного экземпляра случайной траектории осуществляется в цикле в строках 4 – 10, причем начальная точка для всех экземпляров одинакова (строка 4). Проекция смещения по горизонтальной и вертикальной оси dx , dy являются независимыми случайными числами, с равной вероятностью принимающими значения ± 1 . Внешний цикл по переменной j (строки 1 – 12) обеспечивает многократное построение экземпляров случайной траектории и вычисление среднего квадрата смещения (строка 11). По завершение этого цикла переменная r является средним квадратом смещения броуновской частицы, вычисленным методом Монте-Карло.

Задача протекания

Рассмотрим частную формулировку задачи протекания (англ. percolation) – задачу об определении порога протекания в двухфазной среде. Физическим примером может служить композитный материал, составленный из перемешанных случайным образом проводящих и непроводящих гранул. Порог протекания в этом случае – минимальная доля проводящих гранул, при которой образец в целом остается проводником.

Эквивалентная задача на регулярной решетке получается, если потребовать, чтобы каждый узел такой решетки мог быть занят металлическим шаром с вероятностью n , или диэлектрическим с вероятностью $(1-n)$. Электроды помещаются на противоположащих сторонах решетки. Величина n соответствует относительной доле проводящей фазы в упомянутом выше случайном композите и равна отношению числа металлических шаров к полному числу шаров. Диаметр шаров равен постоянной решетки, так что электрическая проводимость обеспечивается при контакте металлических шаров, оказавшихся ближайшими соседями (см. рис. 5.3).

Очевидно, что по мере увеличения числа диэлектрических шаров сопротивление решетки увеличивается, а при некотором значении n_p оно становится бесконечным. Это происходит, когда исчезает последняя цепочка из металлических шаров, связывающих электроды. Величина n_p является случайной и зависит от числа узлов решетки N . Однако, при неограниченном увеличении N , ее среднее значение стремится к определенному пределу, который также обозначается n_p . Он получил название **порога протекания** (percolation threshold). В рассматриваемой модели для плоской квадратной решетки значение порога протекания приблизительно равно 0,5927.

Альтернативный вариант задачи заключается в разрывании случайным образом металлических связей, первоначально соединяющих всех ближайших соседей. В этом случае для плоской квадратной решетки значение порога известно точно: $n_p = 1/2$.

Введем понятие кластера как отдельной группы соприкасающихся между собой металлических шаров. Сопротивление кластера, измеренное между любыми его точками, является конечным. При малом количестве металлических шаров все кластеры невелики и изолированы. С ростом n (доли металлической фазы) отдельные кластеры сливаются и их средний размер увеличивается. В точке n_p (на пороге протекания) возникает «бесконечный» кластер, двигаясь по которому можно пройти с одной

границы сетки до противоположающей границы (рис. 5.3). Поэтому при $n = n_p$ (при движении от меньших значений n к большим) появляется отличная от нуля проводимость. С ростом n бесконечный кластер, постепенно присоединяя конечные кластеры, из очень редкого (разреженного) становится все более мощным.

Образование бесконечного кластера есть фазовый переход второго рода, параметром порядка которого является мощность бесконечного кластера P_∞ , равная относительной доле металлических шаров, принадлежащих бесконечному кластеру. Ниже порога протекания эта величина равна нулю, выше порога она монотонно возрастает с ростом n до значения равного 1 (все шары металлические). Критическое поведение (особое поведение вблизи точки фазового перехода) для мощности бесконечного кластера при $(n - n_p) \approx n_p$ и $n > n_p$, выражается в виде формулы (типичной для критического поведения)

$$P_\infty(n) = P_0 \cdot \left(\frac{n - n_p}{n_p} \right)^\beta = P_0 \cdot \tau^\beta, \quad (5.4)$$

где $\tau = \frac{n - n_p}{n_p}$, $\beta > 0$. Аналогичным образом ведут себя в окрестности порога протекания средний размер конечного кластера $L(\tau)$:

$$L(\tau) = L_0 \cdot |\tau|^{-\nu}, \quad (5.5)$$

и проводимость $Z(\tau)$ при $\tau > 0$:

$$Z(\tau) = Z_0 \cdot \tau^t, \quad (5.6)$$

причем $\nu > 0$, $t > 0$.

В формулах (5.4) – (5.6), по аналогии с фазовым переходом второго рода в магнетиках, введена безразмерная критическая переменная τ , характеризующая близость системы к критической точке;

τ стремится к нулю при приближении к точке фазового перехода. Показатели степени β , ν , t называются критическими индексами.

Для перколяционных фазовых переходов характерно свойство универсальности – значения критических индексов (и другие параметры критического поведения) зависят только от размерности пространства, но не зависят от типа решетки и от того, решеточную или континуальную модель мы рассматриваем. В частности, в двумерном случае $\beta=5/36$, $\nu=4/3$, в трехмерном – $\beta=0.40$, $\nu=0.88$.

Рассмотрим простую задачу об определении порога протекания: найти минимальное значение вероятности n , при которой возникает связное множество шаров одного сорта, имеющее хотя бы по одному представителю на противоположащих гранях. Наличие такого множества следует определять непосредственно по изображению текущей реализации случайной решетке (см. рис. 5.3). Значение доли металлической фазы (закрашенных шаров) на рис. 5.3 чуть выше порога протекания и можно непосредственно убедиться в том, что в системе действительно существует бесконечный кластер. Он соединяет левый нижний угол с правым верхним.

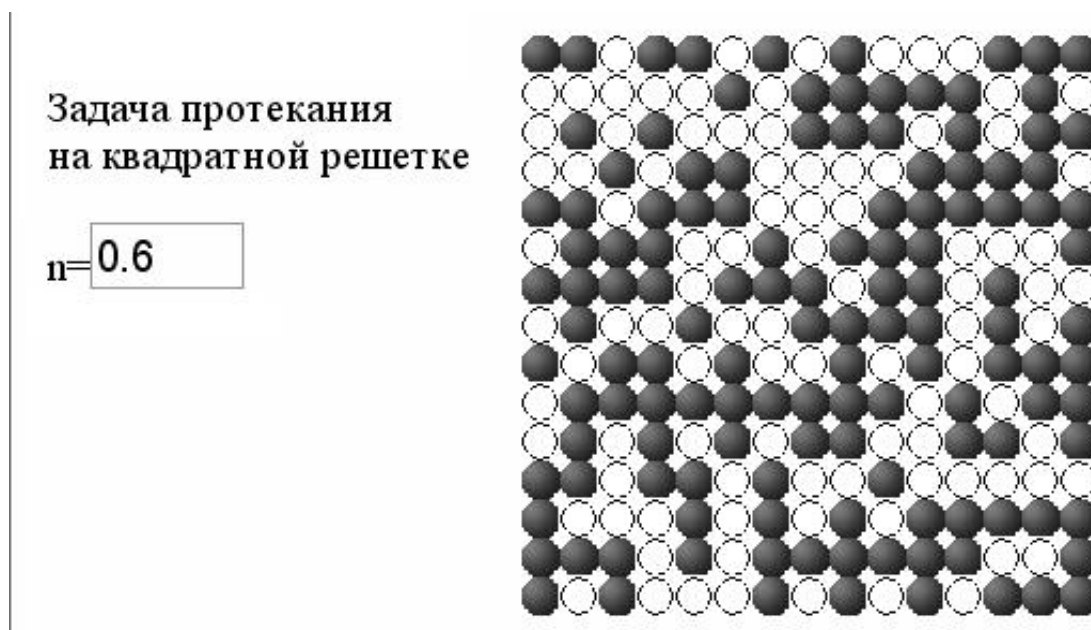


Рис. 5.3. Реализация случайного размещения шаров на квадратной решетке (15 на 15), демонстрирующая появление бесконечного кластера

Однако в системе конечного размера небольшое превышение доли фазы над пороговым значением не гарантирует обязательное существование бесконечного кластера. В этом можно убедиться, многократно перезапуская программу (это приводит к новой случайной реализации размещения шаров).

В соответствующей программе на узлах решетки (независимо друг от друга) должны располагаться случайным образом шары первого или второго сорта. Каждый узел такой решетки может быть занят заштрихованным (металлическим) шаром с вероятностью n , или прозрачным (пустым), с вероятностью $1-n$. Величина n соответствует относительной доле заштрихованных шаров. Диаметр шаров равен постоянной решетки, так что связность обеспечивается при контакте

шаров, оказавшихся ближайшими соседями. Соответствующий код основной части программы можно записать в виде:

```
1. BEGIN
2. Initgraph(gm,gd,"");
3. M:=20; a:=5; n:=0.5;
4. randomize;
5. for i:=1 to M do begin
6. for j:=1 to M do begin
7. x:=100+2*a*j; y:=100+2*a*i;
8. circle(x,y,a);
9. z:=random;
10. if (z<n) then FillEllipse(x,y,a,a);
11. end;end;
12. readln;
13. closegraph; END.
```

В программе точки решетки обозначаются как элементы квадратной матрицы с индексами (i,j) , а декартовы координаты этих точек определяются в строке 7. Постоянная решетки равна $2a$, соответственно радиус кругов на рис. 5.3 равен a . Изначально, вся решетка заполняется «пустыми» окружностями – строка 8.

С помощью генератора случайных чисел для каждой точки решетки с номером (i,j) определяется некоторое случайное число $z:=\text{random}$. Если это число меньше заданной вероятности появления металлического шара n , то в данной точке (i,j) рисуется заштрихованный круг, если нет – остается «пустая» окружность (см. строку 10). Для построения кругов использована процедура `FillEllipse`, которая строит закрашенный заданным цветом эллипс (по умолчанию, цвет – белый).

Таким образом, появление шарика определенного типа в произвольной ячейке является, во-первых, случайным, а, во-вторых, происходит независимо от содержимого всех остальных ячеек.

Следует отметить, что программа, определяющая по данной реализации существует или нет бесконечный кластер, является гораздо более сложной.