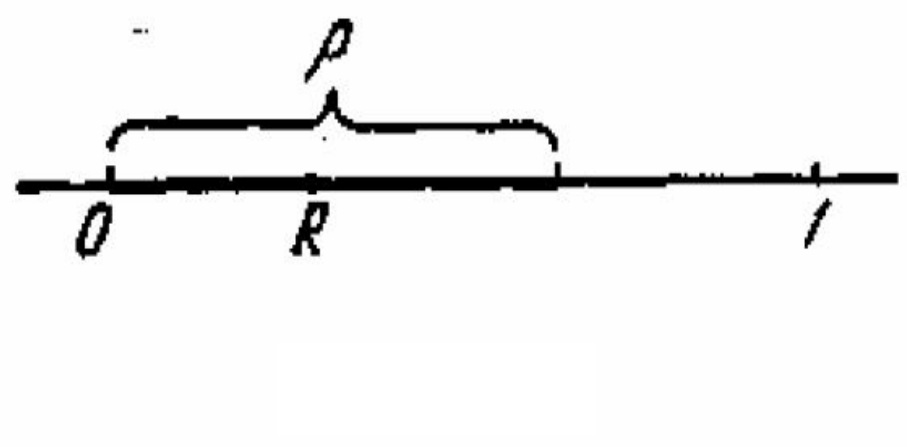
Условимся называть «единичным жребием» любой опыт со случайным исходом, который отвечает на один из следующих вопросов:

1. Произошло или нет событие A?
2. Какое из событий $A_1, A_2, ..., A_к$ произошло?
3. Какое значение приняла случайная величина X?
4. Какую совокупность значений приняла система случайных величин $X_1, X_2, ..., X_к$?

Любая реализация случайного явления методом Монте-Карло строится из цепочки единичных жребиев, перемежающихся с обычными расчетами. Ими учитывается влияние исхода жребия на дальнейший ход событий (в частности, на условия, в которых будет разыгран следующий жребий).

Единичный жребий может быть разыгран разными способами, но есть один стандартный механизм, с помощью которого можно осуществить любую разновидность жребия. А именно, для каждой из них достаточно уметь получать случайное число R, все значения которого от 0 до 1 равновероятны[1]. Условимся кратко называть величину *R*—«случайное число от 0 до 1». Покажем, что с помощью такого числа можно разыграть любой из четырех видов единичного жребия.

1. Произошло или нет событие *A?* Чтобы ответить на этот вопрос, надо знать вероятность p события A. Разыграем случайное число R от 0 до 1, и если оно оказалось меньше p, как показано на рис. будем считать, что событие произошло, а если больше p — не произошло.



А как быть, если число R оказалось в точности равным p? Вероятностью такого совпадения можно пренебречь. А уж если оно случилось, можно поступать как угодно: или всякое «равно» считать за «больше», или за «меньше», или попеременно за то и другое — от этого результат моделирования практически не зависит.

2. Какое из нескольких событий появилось? Пусть события A_1, A_2, ..., $A_к$ несовместны и образуют пол ную группу. Тогда сумма **их** вероятностей p_1, p_2, ..., $p_к$ равна единице. Разделим интервал (0, 1) на k участков длиной p_1, p_2, ..., $p_к$ На какой из участков попало число R — то событие и появилось.

3. Какое значение приняла случайная величина X?

Если случайная величина X дискретна, т. е. имеет значения x_1, x_2, ..., $x_к$ с вероятностями p_1, p_2, ..., $p_к$ то, очевидно, случай сводится к предыдущему. Теперь рассмотрим случай, когда случайная величина непрерывна и имеет заданную плотность вероятности $f(x)$. Чтобы разыграть ее значение, достаточно осуществить следующую процедуру: перейти от плотности вероятности $f(x)$ к функция распределения $F(x)$ по формуле

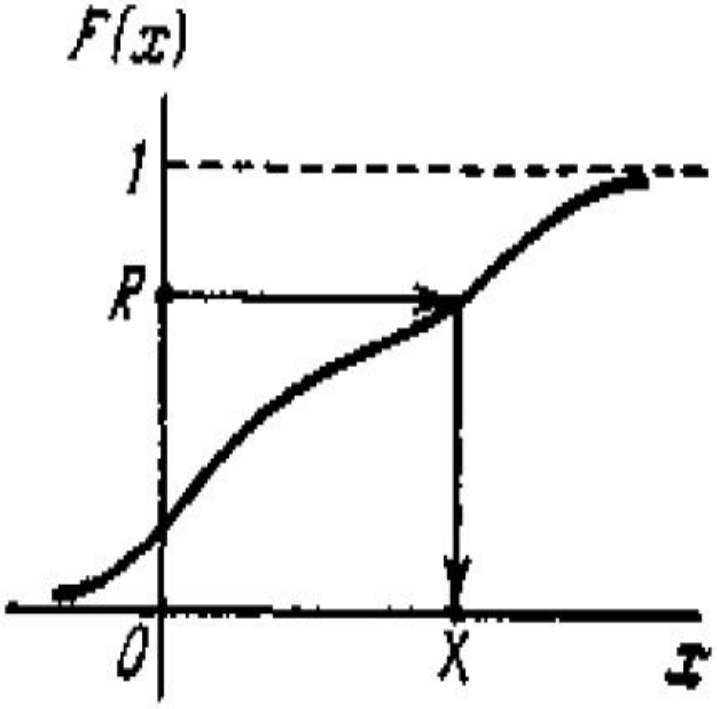
$$F(x) = \int_{-\infty}^{x} f(x)\,dx,$$

затем найти для функции F обратную ей функцию Ψ. Затем разыграть случайное число R от 0 до 1 и взять от него эту обратную функцию:

$$X = \Psi(R).$$

Можно доказать (мы этого делать не будем), что полученное значение X имеет как раз нужное нам распределение $f(x)$.

Графически процедура розыгрыша значения X показана на рис. Разыгрывается число R от 0 до 1 и для него ищется такое значение X, при котором $F(X) = = R$ (это показано стрелками на рис.



На практике часто приходится разыгрывать значение случайной величины, имеющей нормальное распределение. Для нее, как для любой непрерывной случайной величины, правило розыгрыша $X = \Psi(R)$. остается справедливым, но можно поступать и иначе (проще).

Известно, что (согласно центральной предельной теореме теории вероятностей) при сложении достаточно большого числа независимых случайных величин с одинаковыми распределениями получается случайная величина, имеющая приближенно нормальное распределение. На практике, чтобы получить нормальное распределение, достаточно сложить шесть экземпляров случайного числа от 0 до 1. Сумма этих шести чисел

$$Z = R_1 + R_2 + \ldots + R_6$$

имеет распределение, настолько близкое к нормальному, что в большинстве практических задач им можно заменить нормальное. Для того чтобы математическое ожидание и среднее квадратическое отклонение этого нормального распределения были равны заданным m_x, σ_x, нужно подвергнуть величину Z линейному преобразованию и вычислить

$$X = \sigma_X \sqrt{2}(Z - 3) + m_X.$$

4. Какую совокупность значений **приняли случайные величины** $X_1, X_2, \ldots, X_k$? Если случайные величины независимы, то достаточно k раз повторить процедуру, описанную в предыдущем пункте. Если же они зависимы, то разыгрывать каждую последующую нужно на основе ее условного закона распределения при условии, что все предыдущие приняли те значения, которые дал розыгрыш (более подробно останавливаться на этом случае мы не будем).

Таким образом, мы рассмотрели все четыре варианта единичного жребия и убедились, что все они сводятся к розыгрышу (одно- или многократному) случайного числа R от 0 до 1.

Возникает вопрос: а как же разыгрывается это число R? Существует целый ряд разновидностей так называемых «датчиков случайных чисел», решающих эту задачу. Остановимся вкратце на некоторых из них.

Самый простой из датчиков случайных чисел — это вращающийся барабан, в котором перемешиваются перенумерованные шарики (или жетоны). Пусть, например, нам надо разыграть случайное число R от 0 до 1 с точностью до 0,001. Заложим в барабан 1000 перенумерованных шариков, приведем его во вращение и после остановки выберем первый попавшийся шарик, прочтем его номер и разделим на 1000.

Можно поступить и немного иначе: вместо 1000 шариков заложить в барабан только 10, с цифрами 0, 1, 2, ..., 9. Вынув один шарик, прочтем первый десятичный знак дроби. Вернем его обратно, снова покрутим барабан и возьмем второй шарик — это будет второй десятичный знак и т. д. Легко доказать (мы этого делать не будем), что полученная таким образом десятичная дробь будет иметь равномерное распределение от 0 до 1. Преимущество этого способа в том, что он никак не связан с числом знаков, с которым мы хотим знать R.

Числа которые вычисляются по какой либо заданной формуле и могут быть использованы вместо случайных чисел при решении задач численным методом, называются **псевдослучайными числами**.

Из сказанного следует, что оказываются тождественными те свойства случайных и псевдослучайных чисел, которые требуются для моделирования широкого круга задач. По отношению к этим задачам разница между физически генерируемыми "случайными" и псевдослучайным числами практически отсутствует.

К преимуществам псевдослучайных чисел можно отнести:

- небольшие затраты машинного времени для их получения;
- возможность повторных воспроизведений последовательности чисел;
- необходимость однократного тестирования алгоритмов вычисления псевдослучайных чисел.

Из последнего утверждения следует, что разрабатываемые датчики случайных чисел должны подвергаться проверке с помощью специальных тестов, которые должны содержать:

- независимость случайных чисел;
- равномерность их распределения
- k - распределение.

- *Равномерность*. ПСП должна удовлетворять статистическим тестам на равномерность распределения.
- *Независимость*. Любая подпоследовательность последовательности $u_0, u_1, \ldots$ должна быть независима. Генераторы ПСП часто используются для моделирования в d-мерном пространстве и последовательность $u_{dn}, u_{dn+1}, \ldots, u_{dn+d-1}$ должна быть равномерно распределена в d-мерном кубе $[0, 1]^d$.
- *Большой период*. Как было указано ранее, симуляции могут потреблять 10^{12} и более случайных чисел. В таком случае период ρ должен превышать 10^{12}. Для многих генераторов существует стойкая корреляция между $u_0, u_1, \ldots$ и $u_m, u_{m+1}, \ldots$, где $m = \rho/2$ (аналогично и для других делителей периода). Поэтому, на практике период должен значительно превосходить число случайных чисел, которое будет использоваться в опыте. Хорошим правилом для определения ρ является $\rho = m^2$, где m – необходимое число случайных чисел[2].

Среди существующих тестов, самыми жесткими признаются тесты, основанные на проверке многомерной однородности, такие как спектральный тест и тест k-распределения[11].

Определение. Говорят, что псевдослучайная последовательность $\mathbf{x}_i$ периода P, состоящая из w-битных целых, имеет k-распределение с v-битной точностью, если она удовлетворяет следующему условию:

Пусть $\mathrm{trunc}_v(\mathbf{x})$ – число, образованное первыми v битами последовательности $\mathbf{x}$, рассмотрим P векторов вида $(\mathrm{trunc}_v(\mathbf{x}_i), \mathrm{trunc}_v(\mathbf{x}_{i+1}),\ldots, \mathrm{trunc}_v(\mathbf{x}_{i+k-1}))$ $(0 \le i < P)$, длиной kv бит. Тогда каждая из 2^{kv} возможных комбинаций битов встречается равное число раз, за исключением комбинации, состоящей полностью из нулей, которая встречается на один раз меньше.

Для каждого $v = 1, 2,\ldots, w$, пусть $k(v)$ – максимальное число, такое, что последовательность является $k(v)$-распределенной с v-битной точностью.

Заметим, что неравенство $2^{k(v)v} - 1 \leq P$ верно, т.к. максимум P комбинаций может встретиться в одном периоде и число всевозможных битовых комбинаций v наиболее значимых бит в $k(v)$ последовательных словах равно $2^{k(v)v}$.

Геометрическая интерпретация. Разделим каждое целое $\mathbf{x}_i$ на 2^w для нормализации в псевдослучайное вещественное число $\mathbf{x}_i$ из интервала [0, 1]. Поместим P точек в k-мерный куб с координатами $(\mathbf{x}_i, \mathbf{x}_{i+1},\ldots, \mathbf{x}_{i+k-1})(i = 0, 1,\ldots, P-1)$ для всего периода. Каждая из осей данного k-мерного куба разделена на 2^v интервалов. Таким образом, мы разделили куб на 2^{kv} малых куба. Последовательность является k-распределенной с v-битной точностью, если каждый малый куб содержит равное число точек, кроме куба в начале координат, который содержит на одну меньше точек. Следовательно, чем выше $k(v)$ для каждого v, тем более многомерным будет распределение с v-битной точностью. Под тестом на k-распределение, мы будем понимать получение значений $k(v)$.

Тест на k-распределение имеет такую криптографическую интерпретацию: Допустим, что последовательность имеет k-распределение с v-битной точностью и все биты начального заполнения истинно случайны. Тогда знание v наиболее значимых бит первых l слов не дает аналитику возможности предугадать v наиболее значимых бит следующего слова, если $l < k$. Это объясняется тем, что каждая битовая комбинация встречается равновероятно в v битах следующего слова, по определению k-распределения.

В настоящее время существует достаточно много датчиков случайных чисел с хорошими характеристиками.
Наибольшее распространение получил алгоритм Д. Леммера, который называют методом вычетов. Последовательность случайных чисел рассчитывается по следующей формуле

$$\alpha_n = m_n / M$$

$$m_{n+1} = g m_n \pmod{M}$$

где - целые числа и M взаимно просто с g(наибольший общий делитель для M и g равен 1).
Удовлетворительная последовательность псевдослучайных чисел получается, например, при

$$M = 2^{24}, g = 5^{17}, m_0 = 1$$

Период такой последовательности равен 2^{40}.

Моделирование пуассоновской случайной величины

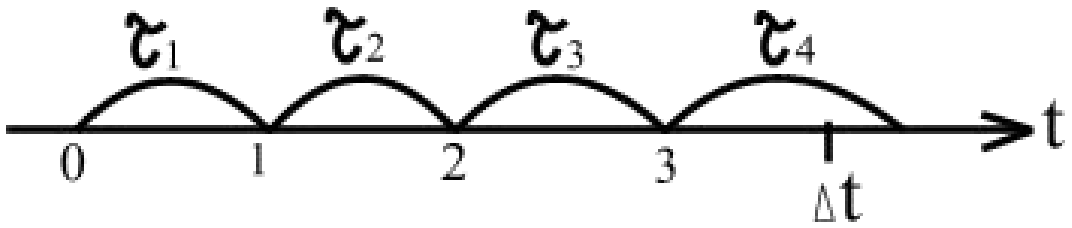
$$P(N = n, \Delta t) = \frac{e^{-\lambda \Delta t} (\lambda \Delta t)^n}{n!}$$

вероятность числа событий на интервале времени Δt

Поток простейший , интервалы между соседними событиями имеют экспоненциальную плотность вероятности

$$f(\tau) = \lambda e^{-\lambda \tau}$$

Моделирование: Моделируем значения случайной величины $\tau_1 \tau_2 \ \tau_n$ и последовательно до тех пор пока не выйдем за пределы отрезка Δt . Число точек на интервале Δt и есть значение случайной величины N.



Моделирование гауссовской случайной величины

Плотность вероятности

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x-m_x)^2}{2\sigma_x^2}}$$

Определим функцию Лапласса

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{z^2}{2}} dz$$

Используя метод обратных функций можно показать, что значение случайной величины X рассчитывается по формуле

$$x = \sigma_x \Phi^{-1}(\alpha - 0.5) + m_x$$

Однако этот алгоритм не используют из-за его большой трудоемкости.

Обычно применяют следующий алгоритм, определяющий сразу два значения гауссовской случайной величины x_1, x_2

$$x_1 = \sqrt{-2\ln \alpha_1}\,\omega_1, \qquad x_2 = \sqrt{-2\ln \alpha_1}\,\omega_2,$$

ω_1, ω_2 координаты изотропного вектора ω на плоскости.

Это означает, что точка $\overline{\omega}\,/\,|\,\omega\,|$

распределена равномерно на окружности $|\,\overline{\omega}\,| = 1$

Моделирование ω_1, ω_2

1 $\gamma_1 = 1 - 2\alpha_1, \quad \gamma_2 = 1 - 2\alpha_2, \quad d^2 = \gamma_1^2 + \gamma_2^2$

2 если $d^2 > 1$, то повторяем 1) и т.д. иначе .

$$\omega_1 = \gamma_1 / d, \quad \omega_2 = \gamma_2 / d$$

Моделирование n – мерной случайной величины

Имеются непрерывные случайные величины $x_1, x_2, \dots, x_n$

совместной плотностью вероятности

$$f(x_1, x_2, \dots, x_n) = f_1(x_1) f_2(x_2 \mid x_1) f(x_3 \mid x_2, x_1) \dots$$

Сначала разыгрывают значение x_1

это значение берется в качестве аргумента условной плотности вероятности

$$f_2(x_2 \mid x_1)$$

разыгрывается значение x случайной величины x_2

x_1, x_2 берутся в качестве значений условной плотности вероятности

$$f_3(x_3 \mid x_2, x_1)$$

И так далее

Тесты diehard — это набор статистических тестов для измерения качества набора случайных чисел. Они были разработаны Джорджем Марсальей (англ.) в течение нескольких лет и впервые опубликованы на CD-ROM, посвящённом случайным числам. Вместе они рассматриваются как один из наиболее строгих существующих наборов тестов (отсюда и название — англ. «die-hard» в качестве прилагательного означает приблизительно «трудноубиваемый» и обычно переводится на русский фразеологизмом «крепкий орешек»).

Дни рождения (Birthday Spacings) — выбираются случайные точки на большом интервале. Расстояния между точками должны быть асимптотически распределены по Пуассону. Название этот тест получил на основе парадокса дней рождения.

Пересекающиеся перестановки (Overlapping Permutations) — анализируются последовательности пяти последовательных случайных чисел. 120 возможных перестановок должны получаться со статистически эквивалентной вероятностью.

Ранги матриц (Ranks of matrices) — выбираются некоторое количество бит из некоторого количества случайных чисел для формирования матрицы над {0,1}, затем определяется ранг матрицы. Считаются ранги.

Обезьяньи тесты (Monkey Tests) — последовательности некоторого количества бит интерпретируются как слова. Считаются пересекающиеся слова в потоке. Количество «слов», которые не появляются, должны удовлетворять известному распределению. Название этот тест получил на основе теоремы о бесконечном количестве обезьян.

Подсчёт единичек (Count the 1's) — считаются единичные биты в каждом из последующих или выбранных байт. Эти счётчики преобразуется в «буквы», и считаются случаи пятибуквенных «слов».

Тест на парковку (Parking Lot Test) — единичные окружности случайно размещаются в квадрате 100×100. Если окружность пересекает уже существующую, попытаться ещё. После 12 000 попыток, количество успешно «припаркованных» окружностей должно быть нормально распределено.

Тест на минимальное расстояние (Minimum Distance Test) — 8000 точек случайно размещаются в квадрате 10 000×10 000, затем находится минимальное расстояние между любыми парами. Квадрат этого расстояния должен быть экспоненциально распределён с некоторой медианой.

Тест случайных сфер (Random Spheres Test) — случайно выбираются 4000 точек в кубе с ребром 1000. В каждой точке помещается сфера, чей радиус является минимальным расстоянием до другой точки. Минимальный объём сферы должен быть экспоненциально распределён с некоторой медианой.

Тест сжатия (The Squeeze Test) — 231 умножается на случайные вещественные числа в диапазоне [0,1) до тех пор, пока не получится 1. Повторяется 100 000 раз. Количество вещественных чисел необходимых для достижения 1 должно быть распределено определённым образом.

Тест пересекающихся сумм (Overlapping Sums Test) — генерируется длинная последовательность вещественных чисел из интервала [0,1). В ней суммируются каждые 100 последовательных чисел. Суммы должны быть нормально распределены с характерными средним и дисперсией.

Тест последовательностей (Runs Test) — генерируется длинная последовательность на [0,1). Подсчитываются восходящие и нисходящие последовательности. Числа должны удовлетворять некоторому распределению.

Тест игры в кости (The Craps Test) — играется 200 000 игр в кости, подсчитываются победы и количество бросков в каждой игре. Каждое число должно удовлетворять некоторому распределению.

Тесты Кнута основаны на статистическом критерии хи^2. Вычисляемое значение статистики хи^2 сравнивается с табличными результатами, и в зависимости от вероятности появления такой статистики делается вывод о ее качестве. Проверка несцепленных серий. Последовательность разбивается на m непересекающихся серий и строится распределение хи^2 для частот появления каждой возможной серии.

Проверка интервалов. Данный тест проверяет равномерность распределения символов в исследуемой последовательности, анализируя длины подпоследовательностей, все элементы которых принадлежат определённому числовому интервалу.

Проверка комбинаций. Последовательность разбивается на подпоследовательности определённой длины, и исследуются серии, состоящие из различных комбинаций чисел.

Тест собирателя купонов. Пусть x1..xn— последовательность длины n и размерности m. Исследуются подпоследовательности определённой длины, содержащие каждое m-разрядное число.

Проверка перестановок. Данный тест проверяет равномерность распределения символов в исследуемой последовательности, анализируя взаимное расположение чисел в подпоследовательностях.

Проверка на монотонность. Служит для определения равномерности исходя из анализа невозрастающих и неубывающих подпоследовательностей.

Проверка корреляции. Данный тест проверяет взаимонезависимость элементов последовательности.

Статистические тесты NIST — пакет статистических тестов, разработанный Лабораторией информационных технологий (англ. Information Technology Laboratory), являющейся главной исследовательской организацией Национального института стандартов и технологий (NIST). В его состав входят 15 статистических тестов, целью которых является определение меры случайности двоичных последовательностей, порождённых либо аппаратными, либо программными генераторами случайных чисел. Эти тесты основаны на различных статистических свойствах, присущих только случайным последовательностям.

Частотный побитовый тест

Суть данного теста заключается в определении соотношения между нулями и единицами во всей двоичной последовательности. Цель — выяснить, действительно ли число нулей и единиц в последовательности приблизительно одинаковы, как это можно было бы предположить в случае истинно случайной бинарной последовательности. Тест оценивает, насколько близка доля единиц к 0,5. Таким образом, число нулей и единиц должно быть примерно одинаковым. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то данная двоичная последовательность не является истинно случайной. В противном случае последовательность носит случайный характер. Стоит отметить, что все последующие тесты проводятся при условии, что пройден данный тест.

Частотный блочный тест

Суть теста — определение доли единиц внутри блока длиной m бит. Цель — выяснить действительно ли частота повторения единиц в блоке длиной m бит приблизительно равна m/2, как можно было бы предположить в случае абсолютно случайной последовательности. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0,01$) двоичная последовательность не носит истинно случайный характер. Если принять $m = 1$, данный тест переходит в тест № 1 (частотный побитовый тест).

Тест на последовательность одинаковых битов

Суть состоит в подсчёте полного числа рядов в исходной последовательности, где под словом «ряд» подразумевается непрерывная подпоследовательность одинаковых битов. Ряд длиной k бит состоит из k абсолютно идентичных битов, начинается и заканчивается с бита, содержащего противоположное значение. Цель данного теста — сделать вывод о том, действительно ли количество рядов, состоящих из единиц и нулей с различными длинами, соответствует их количеству в случайной последовательности. В частности, определяется быстро либо медленно чередуются единицы и нули в исходной последовательности. Если вычисленное в ходе теста значение вероятности $p < 0,01$, то данная двоичная последовательность не является истинно случайной. В противном случае она носит случайный характер.

Тест на самую длинную последовательность единиц в блоке

В данном тесте определяется самый длинный ряд единиц внутри блока длиной m бит. Цель — выяснить действительно ли длина такого ряда соответствует ожиданиям длины самого протяжённого ряда единиц в случае абсолютно случайной последовательности. Если высчитанное в ходе теста значение вероятности $p < 0,01$ полагается, что исходная последовательность не является случайной. В противном случае делается вывод о ее случайности. Следует заметить, что из предположения о примерно одинаковой частоте появления единиц и нулей (тест № 1) следует, что точно такие же результаты данного теста будут получены при рассмотрении самого длинного ряда нулей. Поэтому измерения можно проводить только с единицами.

Тест рангов бинарных матриц

Здесь производится расчёт рангов непересекающихся подматриц, построенных из исходной двоичной последовательности. Целью этого теста является проверка на линейную зависимость подстрок фиксированной длины, составляющих первоначальную последовательность. В случае если вычисленное в ходе теста значение вероятности $p < 0,01$, делается вывод о неслучайном характере входной последовательности бит. В противном случае считаем ее абсолютно случайной. Данный тест так же присутствует в пакете DIEHARD.

Спектральный тест

Суть теста заключается в оценке высоты пиков дискретного преобразования Фурье исходной последовательности. Цель — выявление периодических свойств входной последовательности, например, близко расположенных друг к другу повторяющихся участков. Тем самым это явно демонстрирует отклонения от случайного характера исследуемой последовательности. Идея состоит в том, чтобы число пиков, превышающих пороговое значение в 95 % по амплитуде, было значительно больше 5 %. Если вычисленное в ходе теста значение вероятности $p < 0,01$, то данная двоичная последовательность не является абсолютно случайной. В противном случае она носит случайный характер.

Тест на совпадение неперекрывающихся шаблонов

В данном тесте подсчитывается количество заранее определенных шаблонов, найденных в исходной последовательности. Цель — выявить генераторы случайных или псевдослучайных чисел, формирующие слишком часто заданные непериодические шаблоны. Как и в тесте № 8 на совпадение перекрывающихся шаблонов для поиска конкретных шаблонов длиной m бит используется окно также длиной m бит. Если шаблон не обнаружен, окно смещается на один бит. Если же шаблон найден, окно перемещается на бит, следующий за найденным шаблоном, и поиск продолжается дальше. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0,01$), двоичная последовательность не является абсолютно случайной.

Тест на совпадение перекрывающихся шаблонов

Суть данного теста заключается в подсчете количества заранее определенных шаблонов, найденных в исходной последовательности. Как и в тесте № 7 на совпадение неперекрывающихся шаблонов для поиска конкретных шаблонов длиной m бит используется окно также длиной m бит. Сам поиск производится аналогичным образом. Если шаблон не обнаружен, окно смещается на один бит. Разница между этим тестом и тестом № 7 заключается лишь в том, что если шаблон найден, окно перемещается только на бит вперед, после чего поиск продолжается дальше. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0,01$), двоичная последовательность не является абсолютно случайной.

Универсальный статистический тест Маурера

Здесь определяется число бит между одинаковыми шаблонами в исходной последовательности (мера, имеющая непосредственное отношение к длине сжатой последовательности). Цель теста — выяснить может ли данная последовательность быть значительно сжата без потерь информации. В случае если это возможно сделать, то она не является истинно случайной. В ходе теста вычисляется значение вероятности p. Если $p < 0,01$, то полагается, что исходная последовательность не является случайной. В противном случае делается вывод о её случайности.

Тест на линейную сложность

В основе теста лежит принцип работы линейного регистра сдвига с обратной связью (англ. Linear Feedback Shift Register, LFSR). Цель — выяснить является ли входная последовательность достаточно сложной для того, чтобы считаться абсолютно случайной. Абсолютно случайные последовательности характеризуются длинными линейными регистрами сдвига с обратной связью. Если же такой регистр слишком короткий, то предполагается, что последовательность не является в полной мере случайной. В ходе теста вычисляется значение вероятности p. Если $p < 0{,}01$, то полагается, что исходная последовательность не является случайной. В противном случае делается вывод о её случайности.

Тест на периодичность

Данный тест заключается в подсчете частоты всех возможных перекрываний шаблонов длины m бит на протяжении исходной последовательности битов. Целью является определение действительно ли количество появлений 2m перекрывающихся шаблонов длиной m бит, приблизительно такое же как в случае абсолютно случайной входной последовательности бит. Последняя, как известно, обладает однообразностью, то есть каждый шаблон длиной m бит появляется в последовательности с одинаковой вероятностью. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то данная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер. Стоит отметить, что при m=1 тест на периодичность переходит в частотный побитовый тест (№ 1).

Тест приблизительной энтропии

Как и в тесте на периодичность в данном тесте акцент делается на подсчёте частоты всех возможных перекрываний шаблонов длины m бит на протяжении исходной последовательности битов. Цель теста — сравнить частоты перекрывания двух последовательных блоков исходной последовательности с длинами m и m+1 с частотами перекрывания аналогичных блоков в абсолютно случайной последовательности. Вычисляемое в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0,01$), двоичная последовательность не является абсолютно случайной.

Тест кумулятивных сумм

Тест заключается в максимальном отклонении (от нуля) при произвольном обходе, определяемым кумулятивной суммой заданных (-1, +1) цифр в последовательности. Цель данного теста — определить является ли кумулятивная сумма частичных последовательностей, возникающих во входной последовательности, слишком большой или слишком маленькой по сравнению с ожидаемым поведением такой суммы для абсолютно случайной входной последовательности. Таким образом, кумулятивная сумма может рассматриваться как произвольный обход. Для случайной последовательности отклонения от произвольного обхода должны быть вблизи нуля. Для некоторых типов последовательностей, не являющихся в полной мере случайными подобные отклонения от нуля при произвольном обходе будут достаточно существенными. Если вычисленное в ходе теста значение вероятности $p < 0,01$, то входная двоичная последовательность не является абсолютно случайной. В противном случае она носит случайный

Тест на произвольные отклонения

Суть данного теста заключается в подсчёте числа циклов, имеющих строго k посещений при произвольном обходе кумулятивной суммы. Произвольный обход кумулятивной суммы начинается с частичных сумм после последовательности (0,1), переведённой в соответствующую последовательность (-1, +1). Цикл произвольного обхода состоит из серии шагов единичной длины, совершаемых в случайном порядке. Кроме того такой обход начинается и заканчивается на одном и том же элементе. Цель данного теста — определить отличается ли число посещений определенного состояния внутри цикла от аналогичного числа в случае абсолютно случайной входной последовательности. Фактически данный тест есть набор, состоящий из восьми тестов, проводимых для каждого из восьми состояний цикла: −4, −3, −2, −1 и +1, +2, +3, +4. В каждом таком тесте принимается решение о степени случайности исходной последовательности в соответствии со следующим правилом: если вычисленное в ходе теста значение вероятности $p < 0,01$, то входная двоичная последовательность не является абсолютно случайной. В противном случае она носит случайный характер.