Датчики последовательности псевдослучайных чисел.

Оглавление

1. Генерация случайных чисел

Генерация случайных чисел необходима для решения задач методом статических испытаний (методом Монте-Карло).

Метод Монте-Карло – это численный способ решения математических задач при помощи моделирования случайных величин.

Метод позволяет, как моделировать любой процесс, на протекание которого влияют случайные факторы, так и решать многие не связанные с какими либо случайностями, математические задачи (например, вычисление интеграла), для которых можно придумать вероятностную модель (и даже не одну), позволяющие решать эти задачи.

Т.о. можно говорить о методе Монте-Карло как об универсальном методе решения математических задач. Однако это широкое применение стало возможным только благодаря использованию ЭВМ.

Для реализации метода статических испытаний требуются датчики равномерно распределённых на отрезке [1,0] случайных чисел.

Известны два способа генерации таких чисел:

- физический, например счетчик α-частиц или электронный генератор белого шума
- математический:
    - использование заранее составленной таблицы случайных чисел;
    - вычисление, по какой либо формуле ряда очередных чисел, имитирующих значение случайной величины R. Под словом “имитирующих” понимается, что эти числа удовлетворяют ряду тестов так, как если бы он были значениями этой случайной величины. Такие числа называют псевдослучайными числами.

Достоинства способа генерации случайных чисел вычислением очередного псевдослучайного числа:

- алгоритм (программа) генерирования, как правило, очень коротка;
- скорость генерации имеет порядок скорости работы ЭВМ;
- нужно лишь один раз проверить статическое «качество» псевдослучайной последовательности чисел несколькими специальными статистическими тестами, не противоречат ли те или иные свойства групп чисел гипотезе о том, что эти числа – значения случайной величины с равномерным рас-

пределением. После этого алгоритм генерации псевдослучайных чисел можно безбоязненно использовать –при расчетах методом статических испытаний.

Для генерации последовательности псевдослучайных чисел, имеющих квазиравномерное распределение, широко используют рекуррентные способы вычисления последующего числа R_{i+1} из предыдущего R_i. Это осуществляется с помощью функции преобразования:

$$R_{i+1}=Q(R_i) \qquad (1.1)$$

отображающей множество $\{R_i\}$ на само себя.

Задание начального числа R_0 в формуле (1.1) определяет псевдослучайную последовательность полностью и однозначно. Но её стохастические свойства аналогичны свойствам случайно выбранных значений.

Все приведённые ниже датчики дают последовательность неповторяющихся псевдослучайных чисел базовой длины L, после которой её часть длины l периодически повторяется. Величину L называют длиной апериода, а l - длиной периода последовательности. Т.о. в базовую длину (апериод) входит и период. Базовая длина – это та начальная длина, которую можно использовать в качестве датчика псевдослучайной последовательности.

Разработано большое количество датчиков случайных чисел, использующих различные формулы и алгоритмы преобразования.

Ниже приводятся различные функции преобразования (1.1), которые не требуют прямых действий с регистрами ЭВМ на уровне машинных команд.

2. Оценка качества датчиков случайных чисел

В программном обеспечении практически всех микроЭВМ имеется встроенная функция генерации последовательности псевдослучайных квазиравномерно распределённых чисел. Однако для проведения статического моделирования к генерации случайных чисел предъявляются повышенные требования. Качество результатов такого моделирования напрямую зависит от качества генератора равномерно распределенных случайных чисел, т.к. эти числа являются также источниками (исходными данными) для получения других случайных величин с заданным законом распределения.

К сожалению, идеальных генераторов не существует, а список их известных свойств пополняется перечнем недостатков. Это приводит к риску использования в компьютерном эксперименте плохого генератора. Поэтому перед проведением компьютерного эксперимента необходимо либо оценить качество встроенной в ЭВМ функции генерации случайных чисел, либо выбрать подходящий алгоритм генерации случайных чисел.

Для применения в вычислительной физике генератор должен обладать следующими свойствами:

1. Вычислительной эффективностью – это как можно меньшее время вычисления очередного цикла и объём памяти для работы генератора.
2. Большой длиной L случайной последовательности чисел. Этот период должен включать в себя, по крайней мере, необходимое для статического эксперимента множество случайных чисел. Кроме того, опасность представляет даже приближение к концу L, что может привести к неверным результатам статического эксперимента.

   Критерий достаточной длины псевдослучайной последовательности выбирают из следующих соображений. Метод Монте-Карло заключается в многократном повторении рассчётов выходных параметров модулируемой системы, находящейся под воздействием входных параметров флуктуирующих с заданными законами распределения. Основой реализации метода является генерация случайных чисел с равномерным распределением в интервале [0,1], из которых формируются случайные числа с заданными законами распределения. Далее производится подсчёт вероятности моделируемого события как отношение числа повторов модельных опытов с благополучным исходом к числу общего по-

вторения –опытов при заданных исходных условиях (параметрах) модели.

Для надёжного, в статистическом смысле, вычисления этой вероятности число повторений опыта можно оценить по формуле:

$$N_{\min} = \left[\frac{\Phi^{-1}((1-\beta)/2)}{2\Delta} \right]^2 \qquad (2.1)$$

где Φ^{-1} - функция, обратная функции нормального распределения, β - доверительная вероятность ошибки Δ измерения вероятности.

Следовательно, для того чтобы ошибка не выходила за доверительный интервал $\pm\Delta$ с доверительной вероятностью, например β=0,95 надо, чтобы число повторений опыта было не меньше:

$$N_{\min} = \left[\frac{1.96}{2\Delta} \right]^2 \qquad (2.2)$$

Например, для 10% ошибки (Δ=0,1) получим $N_{\min} = 96$, а для 3% ошибки (Δ=0,03) уже получим $N_{\min} = 1067$.

Для других исходных условий модели новая серия повторений опытов должна проводиться на другой псевдослучайной последовательности. Поэтому либо функция генерации псевдослучайной последовательности должна иметь параметр, изменяющий её (R_0), либо её длина должна быть не менее:

$$L \geq K * N$$

где K- число исходных условий (точек на кривой определяемой методом Монте-Карло), N- число повторений модельного опыта при заданных исходных условиях, L- длина псевдослучайной последовательности.

3. Воспроизводимостью. Как указано выше, желательно иметь параметр, изменяющий генерацию псевдослучайных чисел. Обычно это R_0**.** Поэтому очень важно, чтобы изменение R_0 не портило качества (т.е. статистических параметров) генератора случайных чисел.
4. Хорошими статистическими свойствами. Это наиболее важный показатель качества генератора случайных чисел. Однако его нельзя оценить каким-либо одним критерием или тестом, т.к. не существует необходимых и достаточных критериев случайности конечной последовательности чисел. Самое большее, что можно сказать о псевдослучайной по-

следовательности чисел это то, что она “выглядит” как случайная. Никакой один статистический критерий не является надёжным индикатором точности. По меньшей мере, необходимо использовать несколько тестов, отражающих наиболее важные стороны качества генератора случайных чисел, т.е. степени его приближения к идеальному генератору.

Поэтому, кроме тестирования генератора, чрезвычайно важна проверка его с помощью типовых задач, допускающих независимую оценку результатов аналитическими или численными методами.

Можно сказать, что представление о надёжности псевдослучайных чисел создаётся в процессе их использования с тщательной проверкой результатов всегда, когда это возможно.

3. ТЕСТЫ

Тест№1

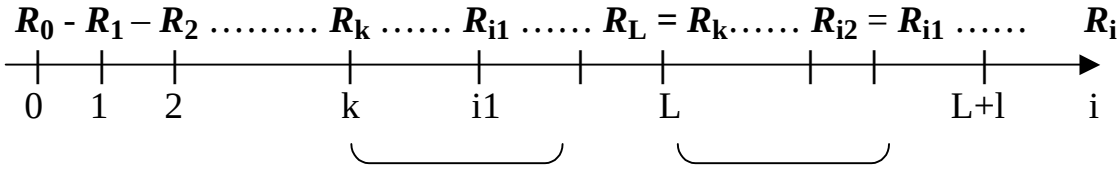
Измерение длин L и l датчика псевдослучайной последовательности чисел.

Т.к. вычисления по формуле (1.1):

$$R_{i+1}=Q(R_i)$$

являются детерминированным процессом, то максимальная длина L_{max} последовательности неповторяющихся чисел определяется разрядностью ЭВМ. Если ЭВМ выводит результаты вычислений, например, 8-ми разрядной десятичной дробью, но производит вычисления в сетке 10-ти разрядной дроби, то длина L_{max} не может быть больше числа разных возможных значений при вычислении 10-ти разрядной дроби, т.е. $L_{max} \le 10^{10}$. Фактическая длина $L<L_{max}$ определяется характером функции генерации (1.1).

Псевдослучайная последовательность начинается с числа R_0. Обозначим порядковый номер последнего из неповторяющихся чисел L-1. Тогда в последовательности $R_0, R_1, R_2, \ldots, R_{L-1}$ все генерируемые числа разные. Этот начальный участок последовательности длины L называют длиной апериодичности (11).



l l

После отрезка апериодичности снова следуют отрезки длины l периодического повторения последовательности чисел. Эти периоды являют собой часть отрезка апериодичности от i=k до i=L-1.

Пусть $R_L = R_k$, тогда периоды l состоят из последовательности $R_k, R_{k+1},\ldots, R_{L-1}$, являющейся частью отрезка [0,L-1]. Таким образом, всегда l<L и тоже определяется свойствами преобразования (1.1).

Начиная с номера i = L проявляется периодичность:

$$R_i = R_{i-l}, \text{ для } i \geq L$$
$$l = L-k$$

Для экспериментального определения длины периодичности l и длины апериодичности L выбираем число i_1 априорно настолько большим, чтобы быть заранее уверенным, что $i_1>k$.

Запускаем программу генерации последовательности $\{R_i\}$ с начального значения R_0 и счётчик номеров i.

При $i=i_1$ величину R_{i1} запоминаем.

С этого шага все последующие числа R_{i1+1}, R_{i1+2} и т.д. сравниваем с R_{i1}. Пусть при $i=i_2$ получено первое совпадение $R_{i2}=R_{i1}$. Тогда длина периода $l=i_2-i_1$.

Длину апериода L находим путём сравнения двух членов последовательности $\{R_i\}$,разнесённых вилкой на интервал l, для чего генерируем одновременно две последовательности $\{R_i\}$ и $\{R_i^{\wedge}\}$, разнесённые на интервал l.

- Шаг i<l, вычисляется только последовательность $\{R_i\}$ с начального значения R_0.
- Шаг i= L, продолжается вычисление $\{R_i\}$ и начинается вычисление $R_i^{\wedge} = R_{l-i}$, т.е. с начального значения R_0.
- На каждом шаге $i \geq l$ величины R_i и $R_i^{\wedge}$ сравниваются. При i= L будет выполняться $R_i = R_i^{\wedge}$.

Детально реализация этой идеи за один проход программы осуществляется следующим образом:

- Запускается счётчик шагов i.
- Запускается программа вычисления $\{R_i\}$. На шаге i=0 выполняется операция: R: = R_0. Далее на всех последующих шагах выполняется операция: R:=Q(R)
- При i= l включается программа вычисления $R_i^{\wedge}$, т.е. выполняются операции: R: = Q(R)
  $R^{\wedge}$: = R_0
- Начиная с шага i= l+1 и далее работают обе программы:

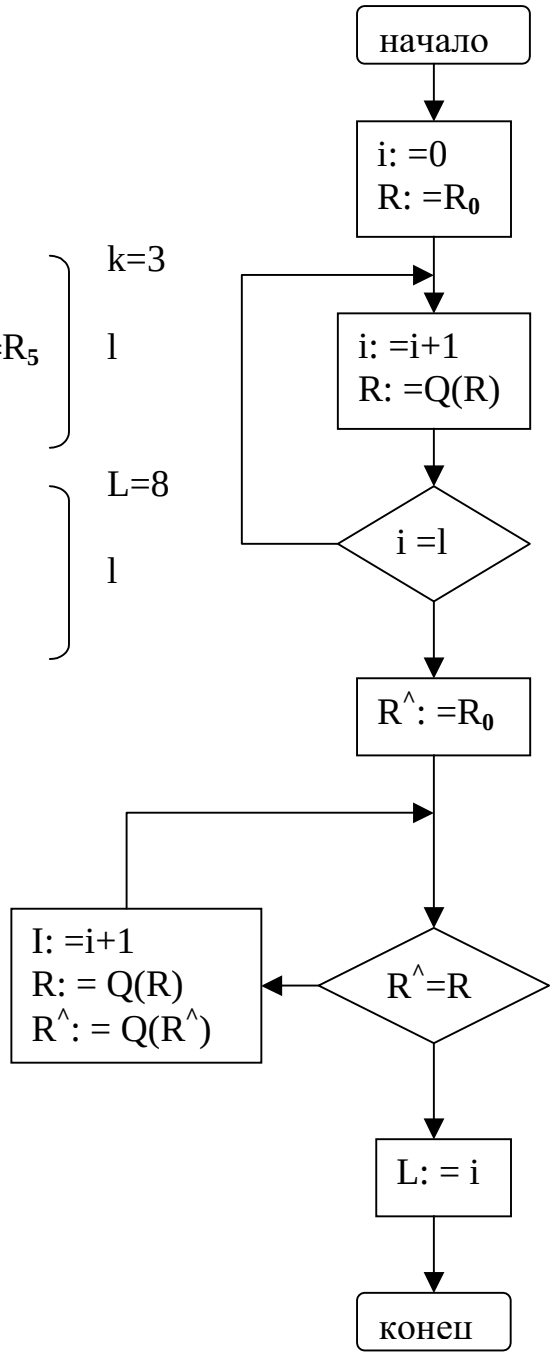
R: = Q(R)
$R^{\wedge}$: = Q($R^{\wedge}$)

- Начиная с шага i= l производится сравнение $R_i^{\wedge}$ с R_i
- При шаге i= L произойдёт совпадение $R_i^{\wedge} = R_i$
- Задача решена.

Например:

i	R		
0	R_0		
1	R_1		
2	R_2		
3	R_3		k=3
4	R_4		
5	R_5 (i=l=5), $R_5^{\wedge}=R_0 \neq R_5$		l
6	R_6 $R_6^{\wedge}=R_1 \neq R_6$		
7	R_7 $R_7^{\wedge}=R_2 \neq R_7$		
8	$R_8 = R_3$ $R_8^{\wedge}=R_3 \neq R_8$		L=8
9	$R_9 = R_4$		
10	$R_{10} = R_5$		l
11	$R_{11} = R_6$		
12	$R_{12} = R_7$		
13	$R_{13} = R_8 = R_3$		
14			

l = L-k=8-3=5



Тесты на равномерность распределения псевдослучайных чисел

Тесты к- равномерности. [2]

Пусть $\alpha_1,\alpha_2,\alpha_3,\ldots$бесконечная последовательность "настоящих" случайных выборочных значений для равномерного распределения на отрезке [0,1].

Разобьём эту последовательность на группы из k величин и обозначим их как k- мерные вектора $\eta_i^{(k)}$.

Пример разбиения на 3-х мерные вектора:

$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ \alpha_8\ \alpha_9\ \alpha_{10}\ \alpha_{11}\ \alpha_{12}\ \ldots$

$(\alpha_1\ \alpha_2\ \alpha_3)\ (\alpha_4\ \alpha_5\ \alpha_6)\ (\alpha_7\ \alpha_8\ \alpha_9)\ (\alpha_{10}\ \alpha_{11}\ \alpha_{12})\ \ldots$

$\eta_1^{(3)}\qquad \eta_2^{(3)}\qquad \eta_3^{(3)}\qquad \eta_4^{(3)}$

Возьмём единичный k- мерный куб. "Единичный"- означает, что грань куба имеет длину, равную единице (т.е. как раз равна отрезку существования случайной величины α_i). "k- мерный" – означает, что куб имеет k граней.

С вероятностью равной единице вектора $\eta_n^{(k)}$ равномерно заполняют единичный k- мерный куб. Это означает, что частота попадания вектора в любую прямоугольную область куба стремится к объёму этой области при $n\to\infty$. Бесконечные последовательности чисел, обладающие такими свойствами, называются k - равномерными.

Ясно, что очень важно проверять k-равномерность псевдослучайных чисел, хотя бы для k=1,2,3,4. Т.к. требуется моделировать случайность, то и проверку следует поводить с помощью статистических критериев, например критерия хи- квадрат (χ^2).

Это реализуется следующим образом. Разобьем каждую грань нашего куба на q частей. Т.е. единичный k-мерный куб будет состоять из $M= q^k$ одинаковых кубиков с объемом $1/q^k$ каждый.

Пусть $m_1, m_2, \ldots, m_M$ количество точек из последовательности $\eta_1^{(k)}, \eta_2^{(k)}, \ldots, \eta_N^{(k)}$ попавших в соответствующие кубики. Очевидно:

$$m_1+m_2+\ldots+m_M = N$$

Известно, что для "настоящей случайной" последовательности с равномерным распределением и независимыми векторами $\eta_i^{(k)}$ величина

$$\sum_{i=1}^{M}(m_i - \frac{N}{M})^2 \Big/ \frac{N}{M}$$

является случайной и имеет χ^2 распределение с (M-1) степенями свободы.

Поэтому в качестве теста на k-равномерность подсчитываем из реализации псевдослучайной последовательности R_i опытную χ^2 величину:

$$\chi^2_{эксп} = \sum_{j=1}^{M}(m_j - \frac{N}{M})^2 / \frac{N}{M} \qquad (A)$$

где m_j – количество k-мерных точек (векторов $\eta_n^{(k)}$, попавших в j-ую ячейку k-мерного куба).

$$\sum_{j=1}^{M} m_j = N$$

N/M - теоретическая частота попадания случайной величины $\eta^{(k)}$ в ячейку k-мерного куба.

1/M - теоретическая вероятность попадания случайной величины $\eta^{(k)}$ в любую ячейку.

Вычисленные значения χ^2 сравниваем с табличными значениями $\chi^2_{табл}(p,\nu)$. Эту величину называют доверительной границей, где:

p – заданный уровень риска (уровень значимости). Его обычно берут в пределах 0,05…0,001.

ν=M-2 – число степеней свободы.

Если окажется, что $\chi^2_{эксп} > \chi^2_{табл}$ (B),

то надо поставить под сомнение гипотезу о том, что наша псевдослучайная последовательность имеет равномерное распределение и независимые вектора $\eta^{(k)}$.

При вычислении по формуле (A) вместо α_i берём R_i.

Выбор N определяется из следующих соображений. С одной стороны следует использовать всю длину L псевдослучайной последовательности тестируемого датчика, т.е. брать N=L/k. С другой стороны, если предполагается использовать последовательность по частям, то надо тестировать k-равномерность всех частей.

Величина q задаёт число M ячеек разбиения куба. Величину M не рекомендуется брать больше $\sqrt{N}$. Откуда можно определить q. Однако, при q>10 уже для k=3 объём вычислений становится нереальным. Поэтому приходится брать небольшие значения q, что несколько снижает качество проверки на равномерность.

Тест №2

2.1 Одномерная размерность

Куб вырождается в отрезок прямой [0,1]. Тест представляется обычной гистограммой. Вектора $\eta_i^{(1)}$ совпадают с R_i.

Алгоритм:

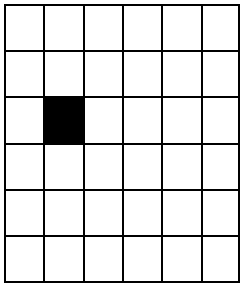
- Разбиваем отрезок [0,1] на q равных частей. M= q, N=L.

$$\underbrace{0 \quad |m_1 \,|\, m_2 \,|\, \ldots \,|\, m_j \,|\ldots\, |m_M| \quad 1}_{q \text{ частей}} \qquad \begin{matrix} R_1 & R_2 & R_3 & \ldots\ldots & R_N \\ \eta_1^{(1)} & \eta_2^{(1)} & \eta_3^{(1)} & \ldots\ldots & \eta_N^{(1)} \end{matrix}$$

- Подсчитываем m_j попаданий чисел R_i в ячейки куба.
- Вычисляем $\chi^2_{эксп}$ по формуле (A).
- Принимаем решение по формуле (B).
- Строим график гистограммы.

2.2 Двумерная размерность

Двумерный куб есть плоскость. Например, при q=6 имеем:



$M= q^{(k)}= 6^2 = 36$

Алгоритм:

- Разобьём псевдослучайную последовательность на группы по два:

  $R_1\ R_2\ \ R_3\ R_4\ \ R_5\ R_6\ \ldots\ldots\ R_{L-1}\ R_L$ N=L/k=L/2

  $\eta_1^{(2)}\quad \eta_2^{(2)}\quad \eta_3^{(2)}\quad \ldots\ldots\quad \eta_N^{(2)}$

- Подсчитываем количества $m_1, m_2, \ldots, m_M$ точек $\eta^{(2)}$ попавших в каждую ячейку плоскости. Длина N подвергаемой тестированию последовательности (части или всей последовательности) должна быть не меньше M^2.

- Вычисляем значение $\chi^2_{эксп}$ и принимаем решение о гипотезе равномерности.
- Вместо двумерной гистограммы представляем график “звёздного неба”, т.е. график расположения N точек $\eta_i^{(2)}$ на плоскости. Если между точками $\eta_i^{(2)}$ существует корреляция, то она визуально проявляется в виде упорядочения структур расположения точек на плоскости. См. также примечание к тексту 2.2.

2.3 Трёхмерная размерность
2.4 Четырёхмерная размерность

Тесты составляются аналогично п. 2.1 и 2.2.

Тест №3
Вычисление коэффициентов неравномерности

Коэффициенты неравномерности распределения генерируемых псевдослучайных чисел вычисляются в процентах по формуле:

$$K_н = \left(\frac{1}{N/M} * \sqrt{\frac{\sum_{j=1}^{M} m_j - \frac{N}{M}}{M}} \right) * 100$$

m_j – количество чисел (точек $\eta_i^{(1)}$) попавших в j-ую ячейку гистограммы.

M – количество ячеек гистограммы.

N/M– среднее ожидаемое количество случайных чисел попадающих в ячейку гистограммы.

Этот тест удобно применять для сравнения датчиков между собой на равномерность гистограммы, т.е. на одномерную k – равномерность. В [4] приведены данные сравнения датчиков по $K_н$ для микрокалькуляторов МК-52 и МК-61. Описание приведённых в таблице датчиков см. ниже.

Номер датчика	6	7	5
$K_н$, %	11,1	7,7	6,9

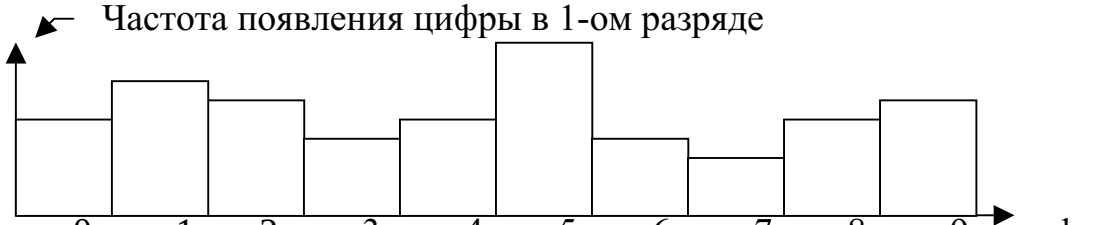
Для датчиков 6 и 7 коэффициенты $K_н$ представлены для N=1000 и M=10. Для датчика 5 N=37, и R_0=0.1234567.

Цифровая размерность

Тест №4

4.1 Одномерная цифровая размерность

Пусть случайные числа представлены 8-разрядной дробью. Берём числа и строим гистограмму для каждого разряда в отдельности. Например, строим гистограмму для первого после запятой разряда. Из N чисел R_i подсчитываем, сколько раз в первом разряде появляется цифра 0, затем – цифра 1и т.д. до цифры 9. Получаем гистограмму для первого разряда.



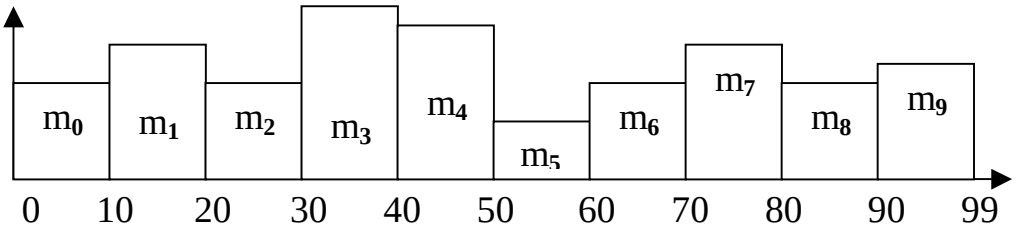
Гипотезу о равномерности распределения цифр в 1-ом разряде проверяем критерием хи-квадрат.

4.2 Двумерная цифровая размерность

Разряды после запятой числа R_i разбиваем на группы по два разряда: $R_i = a_0 \mid a_1\, a_2 \mid a_3\, a_4 \mid a_5\, a_6 \mid a_7\, a_8 \mid$

Для каждой пары разрядов строим гистограмму (всего 4 гистограммы) и оцениваем равномерность гистограммы критерием хи-квадрат.

Гистограмму для пары разрядов строим следующим образом. Пару разрядов рассматриваем как целые числа $\gamma^{(2)}$, принимающие значения от 0 до 99. На оси гистограммы представляем 10 ячеек для каждого десятка чисел $\gamma^{(2)}$:



Строи гистограмму, например, для первой после запятой пары разрядов. Берём N чисел R_i и подсчитываем частоту m_j, j=0,1,…,0 или частость m_j/N попадания чисел $\gamma^{(2)}$ в каждый интервал гистограммы, уславливаясь относить круглые (в смысле десяток) числа к определённому интервалу, так чтобы каждый интервал состоял из 10 чисел. Например, $\gamma^{(2)}=30$ следует относить к интервалу m_3, см. рисунок.

4.4 Трёхмерная и четырёхмерная цифровая размерность

Тестируется аналогично.

Тест №5

Корреляционные свойства последовательности $\{R_i\}$

Вычисляем статические оценки коэффициентов корреляции c(k) для пар вида (R_i, R_{i+k}) и определяем насколько значимо они отличаются от нуля. Заметим, что из (k+1)-мерной равномерности следует, что c(k)=0. Но обратное конечно неверно.

$$c(k)=\frac{\langle R_i * R_{i+k}\rangle-\langle R_i\rangle*\langle R_{i+k}\rangle}{\langle R_i * R_i\rangle-\langle R_i\rangle*\langle R_i\rangle}$$

R_i- i член последовательности $\{R_i\}$.

$<R_i* R_{i+k}>$- для каждого конкретного значения k это сумма всех возможных произведений $R_i* R_{i+k}$ делённая на число произведений.

Если R_i и R_{i+k} не коррелированны, то:

$$\langle R_i * R_{i+k}\rangle=\langle R_i\rangle*\langle R_{i+k}\rangle$$

и, следовательно: c(k)=0.

Тест №6

Проверка решения известной типовой задачи.

Алгоритм:

- Генерируем последовательность $\{R_i\}$, i=0,1,…N-1.
- На каждом шаге генерации вычисляем величину:

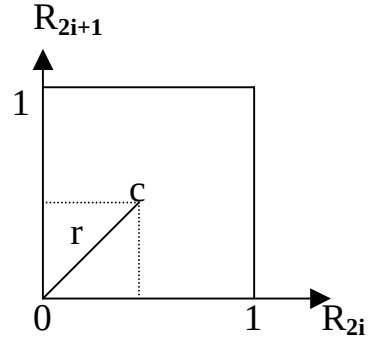
$$r = \sqrt{R_{2i}^2 + R_{2i+1}^2}$$

- Если r<1, то в счётчик событий добавляется единица. Это условие геометрически соответствует тому, точка на плоскости (x, y) с координатами $x=R_{2i+1}$, $y=R_{2i}$ попадёт в круг с радиусом r=1. Т.к. область определения R_i есть отрезок [0,1], то при больших N отношение числа точек (содержимое счётчика c), попавших в четверть круга, к общему числу проанализированных точек N/2, равно отношению соответствующих площадей:

$$\lim_{N\to\infty} \frac{c}{N/2} = \frac{(\text{площадь круга с } r=1)/4}{(\text{площадь квадрата со стороной } 1)} = \frac{\pi * \frac{1^2}{4}}{1^2}$$

Т.е. получаем: $\frac{2c}{N} \to \frac{\pi}{4}$

- По данным счётчика c и заданной величине N вычисляем экспериментальное значение числа π: $\pi_{эксп}=8*c/N$
- Сравнивая $\pi_{эксп}$ с табличным “истинным” значением определяем, сколько верных знаков π можно вычислить с помощью нашего датчика. Очевидно, что следует брать N=L.



Примечание к тесту 2.2.

Интересно строить “звёздное небо” также и парами чисел R_i разнесёнными на k шагов при последовательном изменении i=0,1,… Т.е. парами:

Для k=1	Для k=2
R_0 R_1	R_0 R_2

R_1 R_2	R_1 R_3
R_2 R_3	R_2 R_4
…	…

4.Датчики случайных чисел

Датчик №1: Метод Неймана.

Первый алгоритм для получения псевдослучайных чисел был предложен Дж. Нейманом. Его называют методом середины квадратов.

Пусть задано 4-значное число R_0=0.9876. Возведём его в квадрат. Получим 8-значное число R_0^2=0.97535376. Выберем 4 средние цифры из этого числа и положим R_1=0.5353. Затем снова возведём в квадрат и извлечём из него 4 средние цифры. Получим R_2 и т.д. Этот алгоритм не оправдал себя. Получалось больше чем нужно малых значений R_i.

Однако, представляет интерес исследовать качество этого генератора со смещённой вправо группой выбора цифр у R_i^2:

$$R_{i+1} = \mathrm{INT}(R_i^2 * 10^a) * 10^{-b} - \mathrm{INT}(R_i^2 * 10^{a-b})$$

где a-максимальная значность дроби для данной ЭВМ (например, a=8).

b-количество знаков после запятой в числе R_i (например, 5).

INT (A)- целая часть числа.

Для a=8, b=5, R_0=0.51111111 на ПК ZX-Spectrum получается около 1200 неповторяющихся чисел.

Задание: Исследование следует проводить при варьировании a, b, R_0. Найти при каких величинах a, b, R_0 получается наибольшая длина L последовательности неповторяющихся чисел при “хороших” стохастических параметрах. Установить влияет ли вели-

чина R_0 на качество датчика. Если влияет, то определить область “приемлемых” значений параметра R_0. Представить результаты тестирования оптимального варианта значений a, b, R_0.

Мультипликативные алгоритмы.

Датчик №2: Линейный конгруэнтный генератор Лемера 1951.

$$U_{i+1} = (U_i * M + C) \bmod p$$

где U_i, M, C и p–целые числа.

A mod B- остаток от деления нацело A на B,

$$A \bmod B = A - B * INT(A/B)$$

Генерируемая последовательность имеет повторяющийся цикл не превышающий p чисел.

Максимальный период получается при C≠0, но такой генератор даёт плохие стохастические результаты [3].

При C=0 генераторы называют мультипликативными. Они имеют лучшие стохастические параметры. Формулы их использования называют ещё методом вычетов.

Наиболее популярным для получения псевдослучайных чисел является метод вычетов по такой формуле [2]:

$$U_{i+1} = (U_i * M) \bmod p = U_i * M - p * INT(U_i * M / p)$$
$$R_i = U_i / p$$

где U_i, M, p-целые числа, $0<R_i<1$, $1 \le U_i \le p-1$.

Если выбрать U_0 и M такими, чтобы для $R_0=U_0/p$ получалась несократимая дробь и взять p и M взаимно простыми, тогда все R_i будут несократимыми дробями вида: $R_i= U_i/p$.

Получим наибольшую (но не более p) длину неповторяющейся последовательности чисел. Значения U_0, p и M удобно выбрать из простых чисел.

Задание: Исследовать при каких U_0, p и M длина последовательности неповторяющихся чисел будет не менее 10000 при «хороших» стохастических параметрах. Определить влияет ли величина R_0 при M и p= const на статические характеристики датчика. Если влияет, то определить область допустимых величин U_0. Представить результаты тестирования генератора для оптимальных величин p, M и U_0.

Датчик №3: Модификация Коробова [9].

$$U_{i+1} = (U_i * M) \bmod p$$
$$R_i = U_i / p$$

где p- большое простое число, например 2027, 5087, …

M - целое число, отвечающее условиям:

$$M = p - 3^n,\ M \approx p/2$$

Например, для p=5087 берём n=7. Потому что 3^7=2187, а 3^8=6561 будет уже больше p. Итак: M=5087-2187=2900.

Получаем числа U_i в интервале [1,p-1]=[1,5086] и числа R_i в интервале (0,1).

Задание: Подобрать M и p при которых получаются лучшие статические параметры датчика и наибольшая длина L. Выяснить влияет ли величина R_0 на стохастические характеристики датчика и, если влияет, то определить область допустимых значений R_0. Представить результаты тестирования датчика для оптимальных значений M, p и R_0.

Другие формы вычетов.

Датчик №4:

$$R_{i+1} = FRC(M * R_i),\ R_0 = u/p \qquad (4.1)$$

где FRAC(A)=A-INT(A)- есть дробная часть числа A.

p и M-специально подобранные целые постоянные.

Значение u (u-целое число, u<p) однозначно определяет последовательность. Длина последовательности неповторяющихся чисел, очевидно, не превосходит p.

Смысл формулы состоит в следующем: Разложим R_0 в бесконечную M-ичную дробь, т.е. дробь по степеням M^{-k}:

$$R_0 = a_1 * M^{-1} + a_2 * M^{-2} + ... = 0.a_1a_2a_3... \qquad (4.2)$$

где a_k принимает одно из возможных значений%0, 1, …M-1.

Теперь:

$$R_1 = FRAC(M * R_0) = 0.a_2a_3...$$

$$R_2 = FRAC(M * R_1) = FRAC(M * 0.a_2a_3...) = 0.a_3a_4... = FRAC(M^2 * R_0) \quad (4.3)$$

.....

$$R_i = FRAC(M^i * R_0) = 0.a_{i+1}a_{i+2}...$$

Т.е. операция получения R_i состоит в перенесении запятой в R_0 на i позиций вправо и отбрасывании целой части.

Для того, чтобы дробь (4.2) была бесконечной необходимо, чтобы выражение R_0=p/u являло собой несократимую дробь.

Например, возьмём $p=2^m$, m-целое число. Тогда для того, чтобы u и p были взаимно простыми достаточно выбрать u таким, чтобы в разложении его на простые сомножители среди них не встречался бы множитель 2. Можно выбирать u и p взаимно простыми и другими способами.

Примечание: Многие специалисты [2] исследовали и проверяли псевдослучайную последовательность двоичных чисел V_i:

$$V_i := FRAC(M * V_i) \qquad (4.4)$$
$$V_0 := 2^{-m}$$

m-число двоичных разрядов мантиссы ячейки ЭВМ.

Они брали множитель M вида:

$M = 5^{2q+1}$,q-целое число.

Согласно формуле (4.3) имеем:

$$V_i := FRAC(5^{i*(2q+1)} * V_0) \qquad (4.5)$$

Опишем получение последовательности величин V_i аналогично алгоритму (4.1), (4.2).

Запишем дробь $V_0=2^{-m}$ в системе счисления с основанием 5:

$$V_0 = 2^{-m} = \sum_{k=1}^{\infty} a_k * 5^{-k} = 0.a_1a_2a_3... \qquad (4.6)$$

где a_k принимает одно из значений: 0, 1, 2, 3, 4.

Соотношение (4.5) означает, что запятая в (4.6) переносится на (2q+1)*i позиций вправо и целая часть полученного числа отбрасывается. Отсюда ясно, что при малых 2q+1 величины V_i, V_{i+1} будут заметно зависимыми. Поэтому рекомендуется брать q максимальным из тех, для которых выполняется условие:

$$M = 5^{2q+1} < 2^m \qquad (4.7)$$

Методами теории чисел можно показать, что для указанных параметров M, V_0 длина последовательности будет:

$$L = 2^{m-2}$$

Величина 5^{2q+1} в двоичном виде оканчивается на 01. Поэтому все V_i есть двоичные дроби, последние два разряда которых равны 01.

Вследствии равенства $L=2^{m-2}$ остальные m-2 разряда пробегают все возможные значения (комбинации).

Поэтому в качестве V_0 можно брать любую m-разрядную двоичную дробь такого типа.

Этот метод вычетов реализован программой в машинных кодах на ЭВМ М-220 и БЭСМ-6.

Величины M и m взяты равными соответственно:

$M=5^{15}$ и 5^{17} и m=36 и 40.

Для БЭСМ-6 длина $L\approx 2*10^{11}$.

Задание: Исследовать как влияют величины M, u и p формулы (4.1) на стохастические качества и на длины L и l последовательности псевдослучайных чисел этого датчика. Определить влияет ли R_0 на статистические параметры датчика и, если влияет, то найти допустимую зону задания R_0. Представить результаты тестирования датчика для его оптимальных параметров.

Датчик №5

В [5], [6] предлагается в формуле(4.1) выбирать M и R_0 следующими способами:

Способ 1: $M=\gamma t\pm 3$, t- целое число.

R_0- любое нечётное число, меньшее 1,т.е. дробь вида 0,xxxxxxн, н - нечётная цифра.

В [6] рекомендуется брать M=37. Однако, такой датчик заметно меняет свои статистические характеристики при изменении R_0.

Способ 2: M=16q+11, q-целое число.

R_0 - любое нечётное число меньшее 1. Например, при M=91, R_0=0.11111111 получается $L\approx 10000$.

Задание: Исследовать такими способами выбираемых констант их влияние на стохастические качества датчика и длины l и L последовательности псевдослучайных чисел. Посмотреть также, как влияет выбор чётного R_0. Установить влияет ли R_0 на параметры датчика и, если влияет, то оценить область приемлемого значения R_0. Представить результаты тестирования датчика с подобранными оптимальными параметрами.

Датчик №6 [4], [7]

$$R_{i+1} = FRAC(11 * R_i + \pi)$$

При R_0=0.5 получается около 8000 неповторяющихся чисел.
При R_0=0.002 получается около 9000 неповторяющихся чисел.

Задание: Исследовать влияние выбора R_0 на стохастические качества датчика и длины l и L псевдослучайной последовательности. Можно попробовать заменить константу 11 на другое простое число не большее 100. Представить результаты тестирования датчика для нескольких R_0.

Датчик №7 Датчик со счётчиком [5], [7]

$$\begin{aligned} z_{i+1} &= z_i + 10^{-n} \\ R_{i+1} &= FRAC((R_i / z_{i+1}) + \pi) \end{aligned} \qquad (7.1)$$

где n- константа, целое число, определяемое типом ЭВМ;
z_0, R_0- любые числа, меньшие 1;
Для z_0=0.011 и R_0=0 для программируемого калькулятора получается около $8*10^7$ неповторяющихся чисел.

Задание: исследовать влияние выбора величин n, z_0, R_0 на стохастические параметры датчика и на длины L и l псевдослучайной последовательности чисел. Установить влияет ли величина R_0 на качество датчика и, если влияет, то определить область допустимых величин R_0. Представить результаты тестирования датчика при оптимальных значениях n, z_0, R_0.

Тригонометрические алгоритмы.

Основаны на использовании ошибки вычисления косинуса больших аргументов.

Датчик №8 [7]

$$R_{i+1} = \frac{1}{\pi}\arccos(\cos(10^n * R_i)$$

где n-целое число, определяемое типом ЭВМ. Например, для программируемого микрокалькулятора n=9. Для ПК ZX-Spectrum n=4.

R_0 - любое число меньшее 1.

Задание: исследовать влияние величин n и R_0 на стохастические характеристики датчика и на длины l и L псевдослучайной последовательности. Определить оптимальную, в смысле качества и максимальной длины L, величину n. Установить влияет ли R_0 на качество датчика и, если влияет, то найти допустимые пределы изменения R_0. Представить результаты тестирования датчика при оптимальной величине n.

Датчик №9: Тригонометрический алгоритм со счётчиком [7].

$$R_{i+1} = \frac{1}{\pi}\arccos(\cos(i+100) * R_0)$$

На программируемом микрокалькуляторе получается 10^9 неповторяющихся чисел.

Задание: Установить влияет ли величина R_0 на качество датчика. Представить результаты тестирования датчика для нескольких R_0.

Датчик №10: Алгоритм со счётчиком [5].

$$R_{i+1} = \frac{1}{\pi}\arccos(\cos((10^n - 1) * R_i)$$

где n-целое число, определяемое типом ЭВМ. Для программируемого микрокалькулятора n=8. Получается около 10^8 неповторяющихся чисел. Для ПК ZX-Spectrum n=4.

Задание: Подобрать оптимальную, в смысле хорошего качества и длин l и L, величину n. Установить влияет ли величина R_0 на качество датчика и, если влияет, то найти допустимую область изменения R_0. Представить результаты тестирования датчика для оптимальной величины n и нескольких R_0.

Литература

1. Соболь И.М. Метод Монте-Карло. –М: Наука,1972.
2. Михайлов Г.А. Некоторые вопросы теории методов Монте-Карло. –М.:Наука,1974.
3. Хеерман Д.В. Методы компьютерного эксперимента в теоретической физике. –М: Наука,1990.

4. Стрельянов А.И. Производство вычислений на программируемых микрокалькуляторах МК-52, МК-54, МК-61. –М: Машиностроение,1990.
5. Дьяконов В.П. Справочник по рассчётам на микрокалькуляторах, изд. 3. –М: Наука,1989.
6. Дьяконов В.П. Рассчёт нелинейных и импульсных устройств на программируемых микрокалькуляторах. –М: радио и связь,1984.
7. Епанечников В.А., Цветков А.Н. Справочник по прикладным программам для микрокалькуляторов. –М: Финансы и статистика,1988.
8. Цветков А.Н., Епанечников В.А. Прикладные программы для микроЭВМ «Электроника» Б-3, МК-54, МК-56. –М: Финансы и статистика,1984.
9. Астанин Л.Ю. и др. Применение программируемых калькуляторов для инженерных и научных рассчётов. –М: Энергоиздат,1986.
10. Х. Гулд, Я. Тобочник Компьютерное моделирование в физике, том2. –М: Мир,1990. §11.6 Случайные числа.
11. Полляк Ю.Г. Вероятностное моделирование на ЭВМ. –М: Сов. Радио,1971. Гл.2,стр. 66-112.
12. Дьяконов В.П. Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ. –М: Наука,1987.