



ЛЕКЦИЯ 6.
*« ПСЕВДОСЛУЧАЙНЫЕ
ПОСЛЕДОВАТЕЛЬНОСТИ
И ПРОЦЕДУРЫ ИХ МАШИННОЙ
ГЕНЕРАЦИИ»*

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ И СИСТЕМ

ПЛАН ЗАНЯТИЯ:

- Эталон генератора случайных чисел.
- Непрерывное распределение случайных чисел.
- Дискретное распределение случайных чисел.
- Требования к генератору случайных чисел.
- Основные способы генерации случайных чисел:
 - аппаратный (физический);
 - табличный (файловый);
 - алгоритмический (программный).
- Алгоритмические методы:
 - метод серединных квадратов;
 - метод серединных произведений;
 - метод перемешивания;
 - линейный конгруэнтный метод.
- Приближенные способы преобразования случайных чисел.
- Универсальный способ преобразования случайных чисел.
- Проверка качества работы генератора случайных чисел.

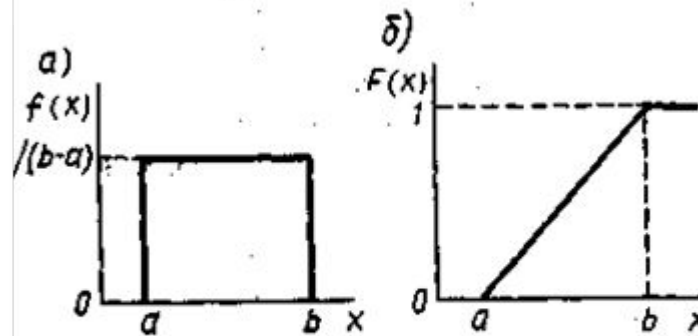
ЭТАЛОН ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ

- При статистическом моделировании систем одним из основных вопросов является учет стохастических воздействий
- Результаты статистического моделирования существенно зависят от качества исходных (базовых) последовательностей случайных чисел

За **эталон генератора** случайных чисел (**ГСЧ**) принят такой генератор, который порождает последовательность случайных чисел с **равномерным законом распределения** в интервале (0; 1).

- Непрерывная случайная величина ξ имеет *равномерное распределение* в интервале (a, b), если ее **функция плотности** (а) и **функция распределения** (б) примет вид :

$$f(x) = \begin{cases} 1/(b-a), & a \leq x \leq b, \\ 0, & x < a, x > b; \end{cases}$$
$$F(x) = \begin{cases} 0, & x < a, \\ (x-a)/(b-a), & a \leq x \leq b, \\ 1, & x > b. \end{cases}$$



- Числовые характеристики случайной величины, принимающей значения x — это математическое ожидание, дисперсия и среднее квадратическое отклонение соответственно:

$$M|\xi| = \int_a^b xf(x) dx = \int_a^b x dx / (b-a) = (a+b)/2;$$

$$D|\xi| = \int_a^b (x - M|\xi|)^2 f(x) dx = (b-a)^2/12;$$

$$\sigma_\xi = +\sqrt{D|\xi|} = (b-a)/(2\sqrt{3}).$$

- При моделировании систем на со случайными числами из интервала $(0, 1)$, где границы интервала соответственно $a=0$ и $b=1$, частным случаем равномерного распределения является функция плотности и функция распределения, соответственно имеющие вид:

$$f(x) = \begin{cases} 1, & 0 \leq x \leq 1, \\ 0, & x < 0, x > 1; \end{cases} \quad F(x) = \begin{cases} 0, & x < 0, \\ x, & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$$

- Такое распределение имеет математическое ожидание $M[\xi] = 1/2$ и дисперсию $D[\xi] = 1/12$.

- Получить такое распределение на цифровой ЭВМ невозможно, так как машина оперирует с ***n*-разрядными числами**. На ЭВМ вместо непрерывной совокупности равномерных случайных чисел интервала (0, 1) используют дискретную последовательность 2^n случайных чисел того же интервала.
- Закон распределения такой дискретной последовательности называют квазиравномерным распределением.**

Случайная величина ξ , имеющая квазиравномерное распределение

в интервале (0, 1), принимает значения $x_i = i/(2^n)$ с вероятностями $p_i = \frac{1}{2^n}$ $i = 0, 2^n - 1$

Математическое ожидание и дисперсия квазиравномерной случайной величины соответственно имеют вид

$$M[\xi] = \sum_{i=0}^{2^n-1} \frac{i}{2^n-1} \frac{1}{2^n} = \frac{1}{(2^n-1)2^n} \sum_{i=0}^{2^n-1} i = \frac{(2^n-1)2^n}{(2^n-1)2^n \cdot 2} = \frac{1}{2}$$

$$D[\xi] = \sum_{i=0}^{2^n-1} \frac{1}{2^n} \left[\frac{i}{2^n-1} - \frac{1}{2} \right]^2 = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \left(\frac{i^2}{(2^n-1)^2} - \frac{i}{2^n-1} + \frac{1}{4} \right) =$$

$$= \frac{1}{2^n} \left(\frac{(2^n-1)2^n(2^{n+1}-1)}{6(2^n-1)^2} - \frac{(2^n-1)2^n}{2(2^n-1)} + \frac{1}{4} \right) = \frac{1}{12} \frac{2^n+1}{2^n-1}$$

ТРЕБОВАНИЯ К ГЕНЕРАТОРУ СЛУЧАЙНЫХ ЧИСЕЛ

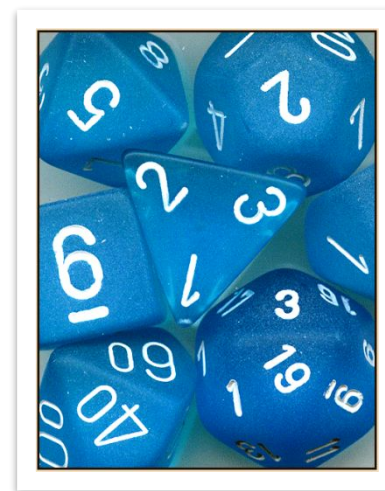
Полученные с помощью идеального генератора псевдослучайные последовательности чисел должны:

- состоять из квазиравномерно распределенных чисел;
- содержать статистически независимые числа;
- быть воспроизводимыми;
- иметь неповторяющиеся числа (серии, последовательности);
- получаться с минимальными затратами машинного времени;
- занимать минимальный объем машинной памяти.



На практике используются три основных способа генерации случайных чисел:

- ▣ **аппаратный** (физический);
- ▣ **табличный** (файловый);
- ▣ **алгоритмический** (программный).



АППАРАТНЫЙ СПОСОБ ГЕНЕРАЦИИ ЧИСЕЛ

- Генерация случайных чисел *вырабатываются специальной электронной приставкой* — **генератором (датчиком)** случайных чисел, — служащей в качестве одного из внешних устройств ЭВМ
- Реализация этого способа генерации не требует дополнительных вычислительных операций ЭВМ по выработке случайных чисел, а необходима только операция обращения к внешнему устройству (датчику).
- В основе лежит какой-либо физический эффект, чаще всего используются:
 - шумы в электронных и полупроводниковых приборах;
 - явления распада радиоактивных элементов;
 - механические устройства;
 - и т. д.

Достоинства способа:

- запас чисел не ограничен;
- расходуется мало операций;
- не занимает место в памяти.

Недостатки:

- требуется периодическая проверка;
- нельзя воспроизводить последовательности;
- используется специальное устройство;
- необходимы меры по обеспечению стабильности.

АППАРАТНЫЙ СПОСОБ ГЕНЕРАЦИИ ЧИСЕЛ

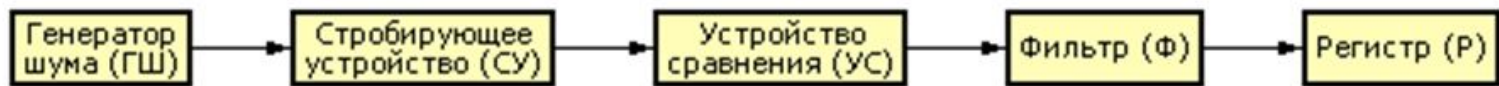
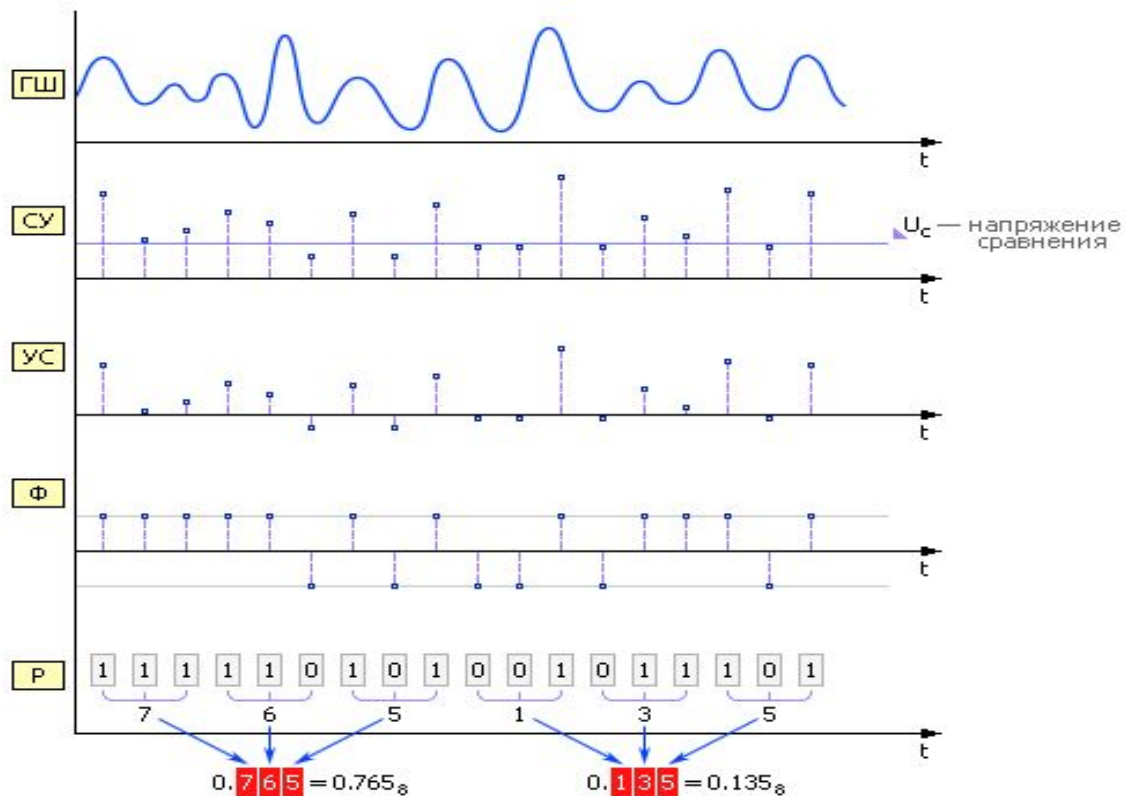


Рис. Схема аппаратного метода генерации случайных чисел



ТАБЛИЧНЫЙ СПОСОБ ГЕНЕРАЦИИ ЧИСЕЛ

- ❑ Случайные числа, представленные в виде таблицы (*содержащей проверенные некоррелированные, то есть никак не зависящие друг от друга, цифры*), помещаются в память ЭВМ.
- ❑ Этот способ получения случайных чисел обычно используют при сравнительно небольшом объеме таблицы и файла чисел.

Пример: Обходя таблицу слева направо сверху вниз, можно получать равномерно распределенные от 0 до 1 случайные числа с нужным числом знаков после запятой (в примере используется для каждого числа по три знака). Так как цифры в таблице не зависят друг от друга, то таблицу можно обходить разными способами, например, сверху вниз, или справа налево, или, скажем, можно выбирать цифры, находящиеся на четных позициях.

| Случайные цифры | Равномерно распределенные от 0 до 1 случайные числа |
|-----------------|---|
| 9 2 9 2 0 4 2 6 | 0.929 |
| 9 5 7 3 4 9 0 3 | 0.204 |
| 5 9 1 6 6 5 7 6 | 0.269 |
| ... | ... |

Достоинства:

- ❑ дает действительно случайные числа;
- ❑ требуется однократная проверка;
- ❑ можно воспроизводить последовательности.

Недостатки:

- ❑ запас чисел ограничен;
- ❑ много места в ОЗУ;
- ❑ необходимо время для обращения к памяти
- ❑ большие трудности порождения и проверки такого рода таблиц

- Способ получения последовательности случайных чисел основанный на формировании случайных чисел в ЭВМ с помощью специальных алгоритмов и реализующих их программ.
- Каждое случайное число вычисляется с помощью соответствующей программы по мере возникновения потребностей при моделировании системы на ЭВМ.
- Числа, генерируемые с помощью этих ГСЧ, всегда являются псевдослучайными (или квазислучайными), то есть каждое последующее сгенерированное число зависит от предыдущего:

$$x_{i+1} = \Phi(x_i),$$

- Последовательности, составленные из таких чисел, образуют петли, то есть обязательно существует цикл, повторяющийся бесконечное число раз. Повторяющиеся циклы называются периодами.

Достоинства:

- высокое быстродействие;
- требуется однократная проверка;
- многократная воспроизводимость последовательности чисел;
- мало места в памяти и нет внешних устройств.

Недостатки:

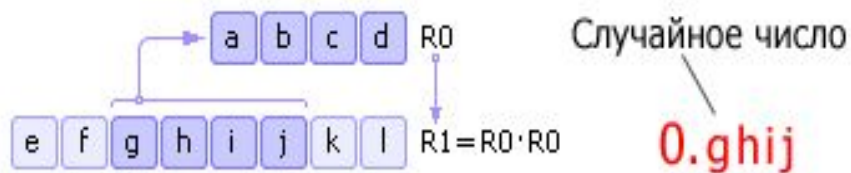
- запас чисел ограничен периодом последовательности;
- числа нельзя в полной мере назвать случайными;
- затраты машинного времени.

Методы получения ГСЧ:

- ✓ метод серединных квадратов;
- ✓ метод серединных произведений;
- ✓ метод перемешивания;
- ✓ линейный конгруэнтный метод.

МЕТОД СЕРЕДИННЫХ КВАДРАТОВ

- Имеется некоторое четырехзначное число R_0 . Это число возводится в квадрат и заносится в R_1 . Далее из R_1 берется середина (четыре средних цифры) — новое случайное число — и записывается в R_0 . Затем процедура повторяется (см. рис.).
- Отметим, что на самом деле в качестве случайного числа необходимо брать не **ghij**, а **0.ghij** — с приписанным слева нулем и десятичной точкой. Этот факт отражен как на рис., так и на последующих подобных рисунках.



Пример: если начальное число $x_0=0,2152$, то $(x_0)^2=0,04631104$, т. е. $x_1=0,6311$, затем $(x_1)^2=0,39828721$, т. е. $x_2=0,8287$, и т.

д.

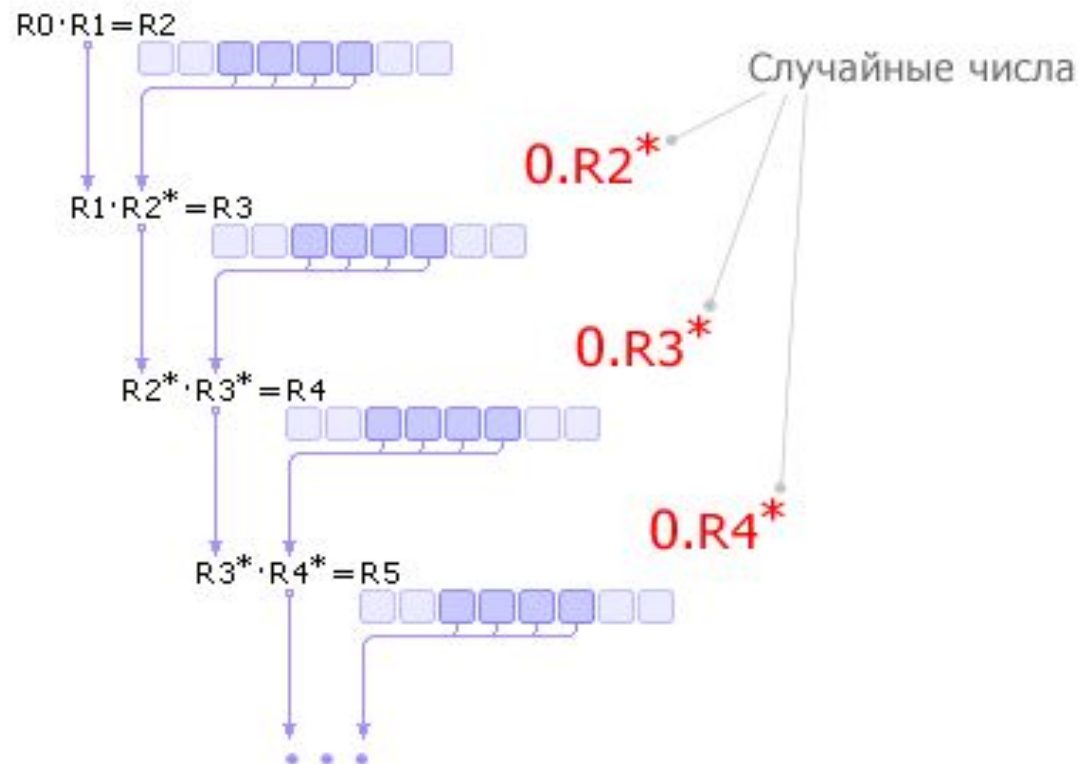
Недостатки метода:

- если на некоторой итерации число R_0 станет равным нулю, то генератор вырождается, поэтому важен правильный выбор начального значения R_0 ;
- генератор будет повторять последовательность через M^n шагов (в лучшем случае), где n — разрядность числа R_0 , M — основание системы счисления;
- повторение последовательности может произойти и раньше, если начальное число будет выбрано неудачно;

Описанный выше способ был предложен Джоном фон Нейманом и относится к 1946 году. Поскольку этот способ оказался ненадежным, от него очень быстро отказались.

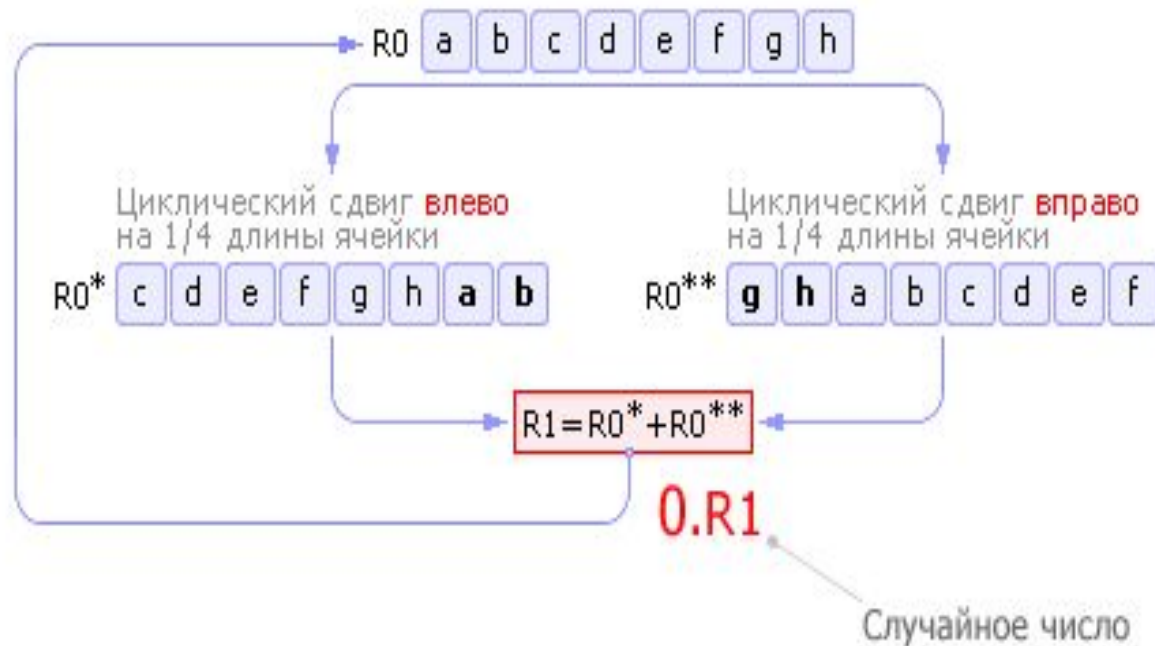
МЕТОД СЕРЕДИННЫХ ПРОИЗВЕДЕНИЙ

- Число R_0 умножается на R_1 , из полученного результата R_2 извлекается середина R_2^* (это очередное случайное число) и умножается на R_1 . По этой схеме вычисляются все последующие случайные числа



МЕТОД ПЕРЕМЕШИВАНИЯ

- В методе перемешивания используются операции циклического сдвига содержимого ячейки влево и вправо. Идея метода состоит в следующем. Пусть в ячейке хранится начальное число $R0$. Циклически сдвигая содержимое ячейки влево на $1/4$ длины ячейки, получаем новое число $R0^*$. Точно так же, циклически сдвигая содержимое ячейки $R0$ вправо на $1/4$ длины ячейки, получаем второе число $R0^{**}$.
- Сумма чисел $R0^*$ и $R0^{**}$ дает новое случайное число $R1$.
- Далее $R1$ заносится в $R0$, и вся последовательность операций повторяется



- Линейный конгруэнтный метод является одной из простейших и наиболее употребительных в настоящее время процедур, имитирующих случайные числа.
- В этом методе используется операция $\text{mod}(x, y)$, возвращающая остаток от деления первого аргумента на второй. Каждое последующее случайное число рассчитывается на основе предыдущего случайного числа по следующей формуле:

$$r_{i+1} = \text{mod}(k \cdot r_i + b, M)$$

где M — модуль ($0 < M$), k — множитель ($0 \leq k < M$), b — приращение ($0 \leq b < M$),
 r_0 — начальное значение ($0 \leq r_0 < M$)

- Последовательность случайных чисел, полученных с помощью данной формулы, называется **линейной конгруэнтной последовательностью**.
- Многие авторы называют линейную конгруэнтную последовательность при $b = 0$ **мультипликативным конгруэнтным** методом, а при $b \neq 0$ — **смешанным конгруэнтным** методом

Для качественного генератора требуется **подобрать подходящие коэффициенты**. Необходимо, чтобы число M было довольно большим, так как период не может иметь больше M элементов. С другой стороны, деление, используемое в этом методе, является довольно медленной операцией, поэтому для двоичной вычислительной машины логичным будет выбор $M = 2^N$, поскольку в этом случае нахождение остатка от деления сводится внутри ЭВМ к двоичной логической операции «AND». Также широко распространен выбор наибольшего простого числа M , меньшего, чем 2^N : в специальной литературе доказывается, что в этом случае младшие разряды получаемого случайного числа $r_i + 1$ ведут себя так же случайно, как и старшие, что положительно сказывается на всей последовательности случайных чисел в целом. В качестве примера можно привести одно из чисел **Мерсенна**, равное $2^{31} - 1$, и таким образом, $M = 2^{31} - 1$.

Одним из требований к линейным конгруэнтным последовательностям является **как можно большая длина периода**. Длина периода зависит от значений M , k и b . Теорема, которая приводится ниже, позволяет определить, возможно ли достижение периода максимальной длины для конкретных значений M , k и b .

Теорема. Линейная конгруэнтная последовательность, определенная числами M , k , b и r_0 , имеет период длиной M тогда и только тогда, когда:

- числа b и M взаимно простые;
- $k - 1$ кратно p для каждого простого p , являющегося делителем M ;
- $k - 1$ кратно 4, если M кратно 4.

Примеры использования линейного конгруэнтного метода для генерации случайных чисел:

Пример 1:

$$M = 2^N, \quad k = 3 + 8 \cdot q \text{ (или } k = 5 + 8 \cdot q), \quad b = 0, \quad r_0 \text{ — нечетно .}$$

Было установлено, что ряд псевдослучайных чисел, генерируемых на основе данных из примера 1, будет повторяться через каждые $M/4$ чисел. Число q задается произвольно перед началом вычислений, однако при этом следует иметь в виду, что ряд производит впечатление случайного при больших k (а значит, и q). Результат можно несколько улучшить, если b нечетно и $k = 1 + 4 \cdot q$ — в этом случае ряд будет повторяться через каждые M чисел. После долгих поисков k исследователи остановились на значениях 69069 и 71365.

Пример 2:

$$M = 2^{31} - 1, \quad k = 1\,220\,703\,125, \quad b = 7, \quad r_0 = 7$$

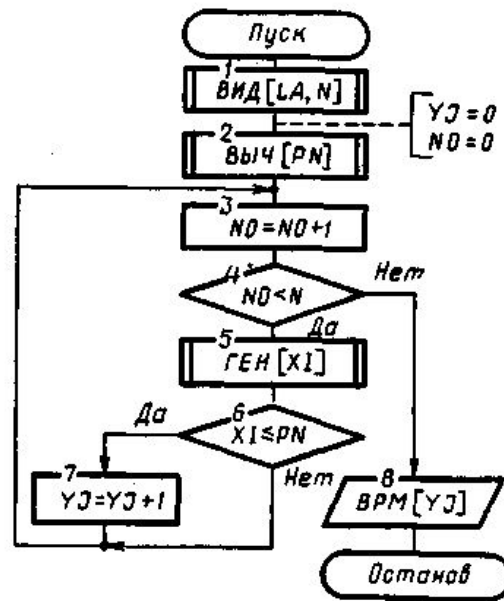
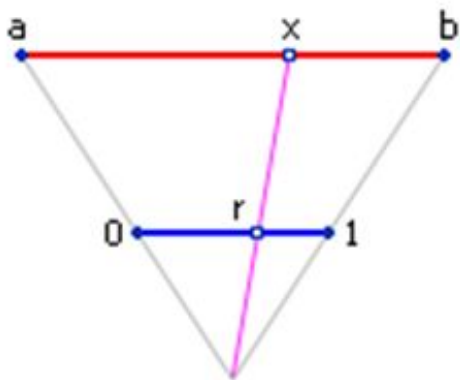
Генератор случайных чисел, использующий данные из примера 2, будет выдавать случайные неповторяющиеся числа с периодом, равным 7 миллионам.

Мультипликативный метод генерации псевдослучайных чисел был предложен Д. Г. Лехмером (D. H. Lehmer) в 1949 году.

ПРЕОБРАЗОВАНИЯ

В практике моделирования систем приближенные **способы преобразования случайных чисел** классифицируются следующим образом:

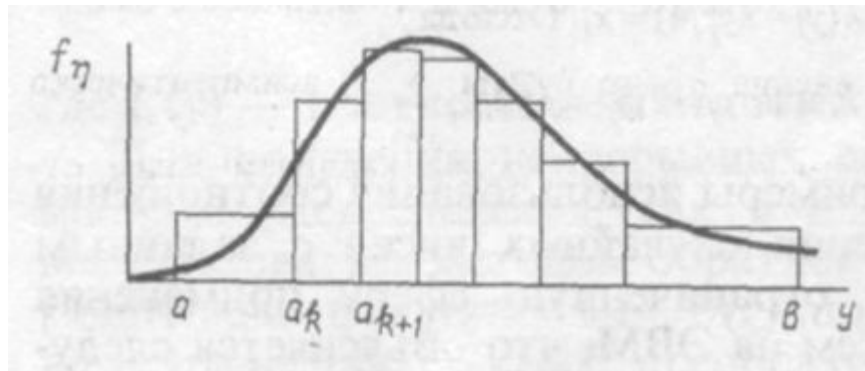
- 1) **универсальные способы**, с помощью которых можно получать случайные числа с законом распределения любого вида;
- 2) **не универсальные способы**, пригодные для получения случайных чисел с конкретным законом распределения.



УНИВЕРСАЛЬНЫЙ СПОСОБ ПРЕОБРАЗОВАНИЯ

Универсальный способ получения случайных чисел, базируется на *кусочной аппроксимации функции плотности*.

Пусть требуется получить последовательность случайных чисел $\{y_i\}$ с функцией плотности $f_n(y)$, возможные значения которой лежат в интервале (a, b) . Представим $f_n(y)$ в виде кусочно-постоянной функции, т. е. разобьем интервал (a, b) на m интервалов.



Будем считать, что функция плотности на каждом интервале постоянна. Тогда случайную величину η можно представить в виде $\eta = a_k + \eta_k$

где a_k — абсцисса левой границы k -го интервала;

η_k — случайная величина, возможные значения которой располагаются равномерно внутри k -го интервала.

УНИВЕРСАЛЬНЫЙ СПОСОБ ПРЕОБРАЗОВАНИЯ

На участке (a_k, a_{k+1}) случайная величина η_k распределена равномерно. Целесообразно разбить (a, b) на интервалы так, чтобы вероятность попадания случайной величины η_k в любой интервал была постоянной и не зависела от номера интервала.

Для вычисления a_k воспользуемся следующим соотношением:

$$\int_{a_k}^{a_{k+1}} f_{\eta}(y) dy = 1/m.$$

Алгоритм машинной реализации этого способа получения случайных чисел сводится к выполнению следующих действий:

- 1) генерируется случайное равномерно распределенное число x_i из интервала $(0, 1)$;
 - 2) с помощью этого числа случайным образом выбирается интервал ;
- генерируется число x_{i+1} и масштабируется с целью приведения его к интервалу (a_k, a_{k+1}) , т. е. домножается на коэффициент $(a_{k+1} - a_k)$
 - вычисляется случайное число $(a_{k+1} - a_k)x_{i+1}$ с требуемым законом распределения.

В пункте 2 целесообразно для этой цели построить таблицу (сформировать массив), в которую предварительно поместить номера интервалов k и значения коэффициента масштабирования, для приведения числа к интервалу (a, b) . Получив из генератора случайное число x_i , с помощью таблицы можно сразу определять абсциссу левой границы a_k и коэффициент масштабирования

Достоинства способа:

При реализации на ЭВМ требуется небольшое количество операций для получения каждого случайного числа, так как операция масштабирования выполняется только один раз перед моделированием

$$(a_{k+1} - a_k)$$

ПРОВЕРКА КАЧЕСТВА РАБОТЫ ГЕНЕРАТОРА

От качества работы ГСЧ зависит качество работы всей системы и точность результатов. Поэтому случайная последовательность, порождаемая ГСЧ, должна удовлетворять целому ряду критериев.

Осуществляемые проверки бывают двух типов:

- проверки на равномерность распределения;
- проверки на статистическую независимость.

Требуется выполнить лабораторную работу...

При выполнении работы следует использовать:

1. Материалы к лабораторной работе (*одноименный файл прилагается*).
2. Задание к лабораторной работе (*одноименный файл прилагается*).

ССЫЛКА НА РЕСУРС



<http://yadi.sk/d/DkzYhU6Q9nGs7>