

**Аналитико-статистическое  
моделирование  
информационных систем**

**Кафедра информационных управляющих систем**

Лекции читает

канд.техн.наук, доцент

Литвинов Владислав Леонидович

## Список литературы:

1. О.И. Кутузов, Т.М. Татарникова

### **МОДЕЛИРОВАНИЕ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ**

<http://dvo.sut.ru/libr/ius/w101kutu/index.htm>

2. *Боев В. Д.*, Моделирование систем. Инструментальные средства GPSS WORLD. Учеб. пособие — СПб.: БХВ-Петербург, 2004. — 368 с.
3. *Боев В. Д., Сыпченко Р. П.* Компьютерное моделирование. Элементы теории и практики. Учеб. пособие — СПб.: Военная академия связи, 2009. — 432 с.
4. *Бражник А. Н.*, Имитационное моделирование: возможности GPSS WORLD — СПб.: Реноме, 2006. — 439 с.

# Тема лекции 1:

Моделирование  
телекоммуникационных систем.  
Основные понятия и определения.

- Технология моделирования сложных технических систем, к классу которых относятся ТКС, опирается на имитационное машинное моделирование. Термин “имитационное моделирование” часто отождествляется с термином “статистическое моделирование”. Статистическое моделирование как метод машинной реализации имитационных моделей систем, подверженных случайным воздействиям, в настоящее время наиболее употребим.

- Статистическое моделирование – это технология, основанная на применении законов математической статистики и способностях современных компьютеров порождать и обрабатывать за короткие промежутки времени огромное количество данных. Тем самым становится возможным строить достаточно точные модели реальных сложных объектов.

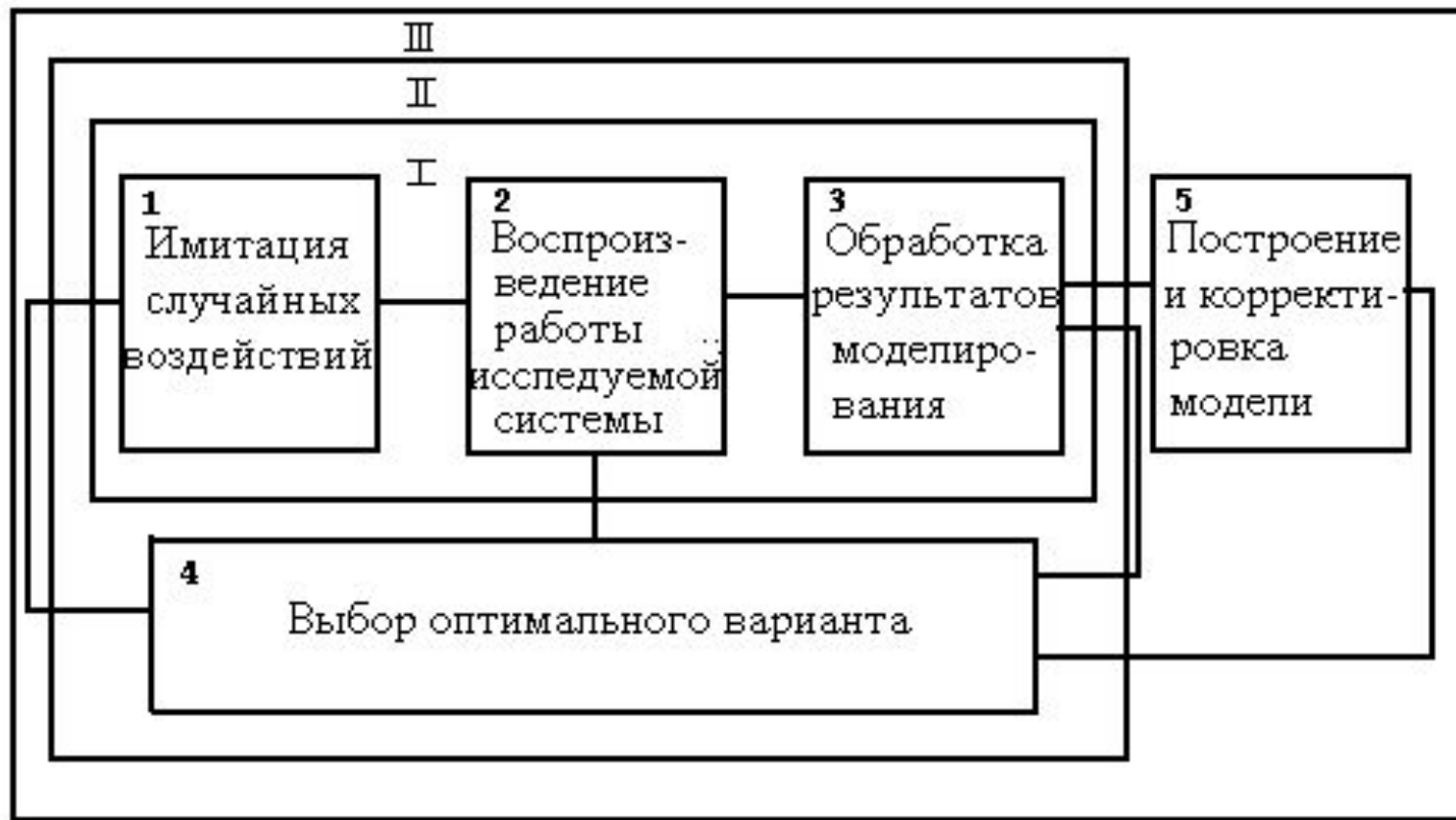


Рис. 1.1 - Процесс моделирования системы

Блоки 1,2,3 представляют операции по исследованию одного варианта модели. Эти операции повторяются при различных реализациях случайных процессов, образуя внутренний цикл моделирования (ЦИКЛ I).

Процедура выбора оптимального варианта моделируемой системы (блок 4) управляет экспериментом путем изменения соответствующим образом вариантов модели. При этом блоки 1,2,3 (внутренний цикл) охватываются цепями обратной связи (ЦИКЛ II).

Связь 3-4-2 отражает адаптацию моделируемой системы.

Связь 3-4-1 может возникнуть, если при оптимизации варьируется не только модель системы, но и модель случайных воздействий, рассматриваемых, например, как описание конфликтующей стороны.

Оценка результатов исследования вариантов модели оказывается типовой операцией (ЦИКЛ I), многократно выполняемой как в динамическом цикле корректировки модели (ЦИКЛ III), так и в цикле оптимизации (ЦИКЛ II): любой метод поиска экстремума основан на сравнении значений оптимизирующего показателя.

Таким образом, доминирующим в схеме (рис.1.1) является статистическое моделирование (ЦИКЛ 1).



В основе статистического моделирования лежит процедура, применяемая для моделирования случайных величин и функций и носящая название метода статистических испытаний (метод Монте-Карло).

Общая схема метода Монте-Карло может быть записана в виде:

$$\theta = \int y(x) p(x) dx \approx \tilde{\theta} = \frac{1}{M} \sum_{i=1}^M y(x_i), \quad x_i \approx p(x).$$

По закону больших чисел среднее арифметическое сходится к математическому ожиданию

- В каждом опыте разыгрывается реализация  $x$  случайной величины  $X$  (в  $i$ -м опыте реализация  $x_i$ ) в соответствии с распределением  $p(x)$  и вычисляется значение функции в виде  $u(x_i)$ . Индекс  $i$  подчеркивает, что для каждой ( $i$ -й) реализации процесса аргументы, составляющие вектор  $X$ , имеют свои случайные значения. Вычисленное очередное значение  $u(x_i)$  добавляется к накапливаемой сумме  $\sum u(x_i)$ .

- На этом заканчивается очередной опыт. После того как проведено  $M$  опытов, вычисляется итоговая оценка в виде правой части выражения (1.1). Опыты повторяются до тех пор, пока дисперсия оценки не снизится до требуемой величины, зависящей от допустимой погрешности и коэффициента доверия.

- Проиллюстрируем суть метода Монте-Карло относительно простым примером. Пусть требуется оценить надежность системы (рис.1.2).
- Система выполняет свою функцию, если работают цепочки блоков: 1,2,5,7; 1,3,5,7; 1,4,6,7. Какие-то блоки могут отказать. Каждый блок характеризуется временем безотказной работы  $t_j$ ,  $j=1\dots7$ .
- Пусть заданы плотности распределения  $p_j(t_j)$ , Какова надежность системы в целом?

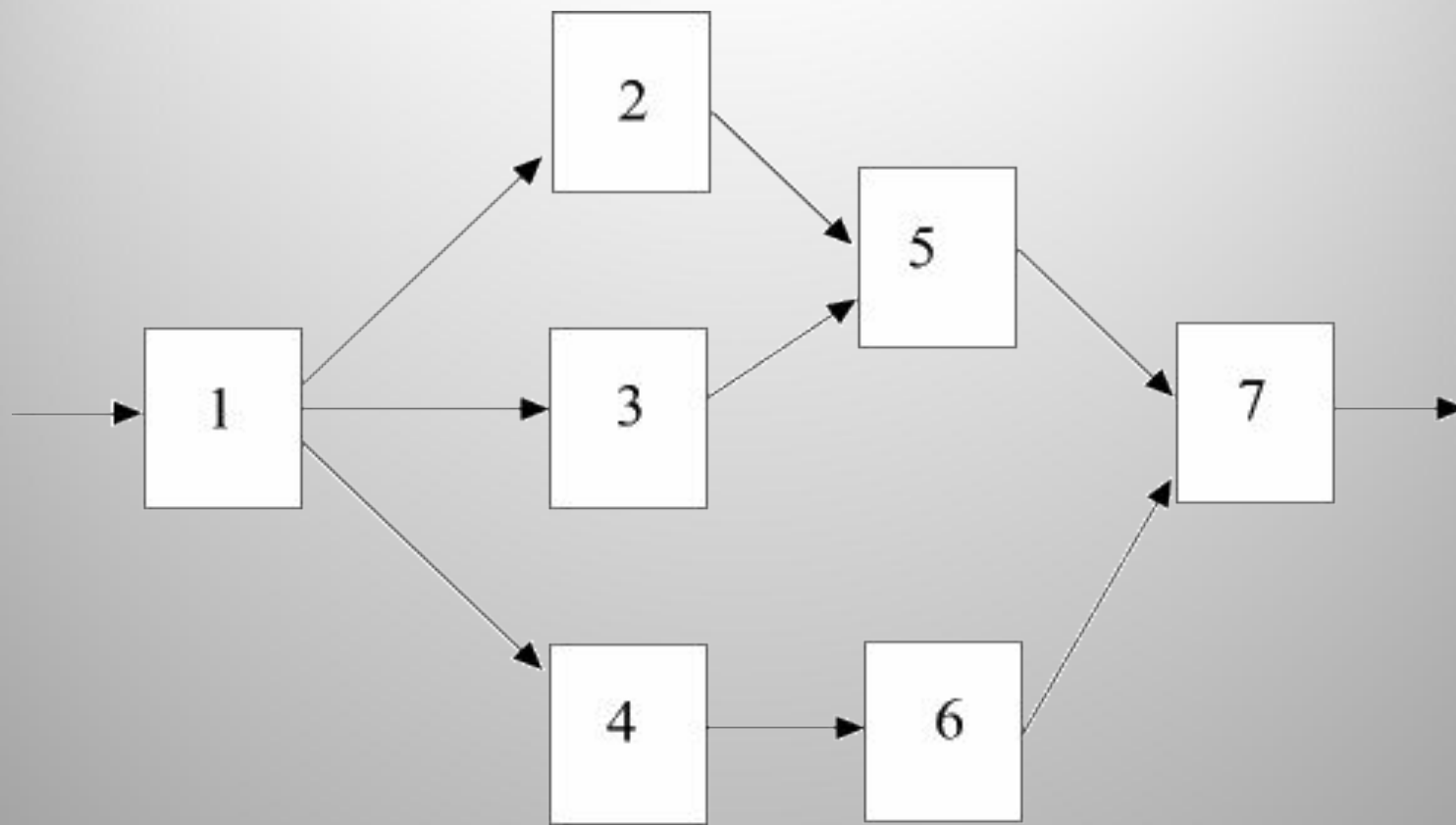


Рис.1.2. Блочная структура системы

Таким образом, испытания реальной системы заменены на испытания математической модели. Каждое испытание сопровождается расчетом. Поэтому имитационное моделирование и называют численным экспериментом на ЭВМ с математической моделью (модель выступает как объект исследования). При реализации испытания возможны и логические операции. И расчетные, и логические операции реализуются на ЭВМ с помощью соответствующих алгоритмов, которые в совокупности и составляют моделирующий алгоритм. Моделирующий алгоритм обеспечивает построение траекторий смены состояний системы во времени, а воспроизведение случайных факторов, определяющих эти состояния, конструируется с использованием заданных законов случайных событий и величин и реализуется с помощью датчиков базовой случайной величины (БСВ).

- Существует множество систем, процессы функционирования в которых могут быть представлены моделями информационных потоков, получившими название систем массового обслуживания (СМО). Это прежде всего процессы в технических системах - телефония, радиосвязь и телекоммуникации, вычислительные машины, системы и вычислительные сети.
- При их анализе наиболее важно определить скорость передачи или обработки информации, оценить пропускную способность, загрузку оборудования и т. д.
- При анализе транспортных систем важнейшими задачами являются определение скорости и объема перевозок, сокращение простоев и др. Процессы жизнедеятельности в биологических системах требуют прежде всего определения благоприятных условий жизни, размножения и развития отдельных особей или популяции (колонии, сообщества) в целом.
- Многие процессы деятельности человека (социальные, экономические, экологические) могут быть представлены моделями типа СМО.
- И даже обучение, представляемое как усваивание знаний и забывание, также может быть описано такими моделями.

Любая подобная система неизбежно испытывает различного рода возмущения, источниками которых могут быть либо внешние воздействия, обусловленные случайными или систематическими изменениями окружающих условий, либо внутренние флуктуации, возникающие в самой системе в результате взаимодействия элементов. При исследовании эти системы представляются в виде стохастических моделей дискретных процессов (СМДП). Несмотря на успешное развитие и применение методов аналитического моделирования СМДП, основным методом исследования таких систем остается имитационное моделирование на ЭВМ с применением специализированных языков программирования.



За всю историю развития вычислительной техники было создано более 300 языков моделирования дискретных процессов. Одним из первых языков описания СМДП, появившихся в начале 60-х годов, был язык блок-диаграмм, предложенный Гордоном, идеи которого оказались настолько плодотворны, что использовались во многих последующих разработках в нашей стране и за рубежом. На основе языка блок-диаграмм в 70-х годах был создан и в последующем адаптирован к ПК широко используемый в настоящее время для моделирования большого класса систем язык и система моделирования GPSS (General Purpose Simulation System - Система моделирования общего назначения).

**GPSS** (англ. *General Purpose Simulation System* —  
общецелевая система моделирования) —  
язык программирования — общецелевая  
система моделирования) — язык  
программирования, используемый для  
имитационного моделирования различных  
систем, в основном систем массового  
обслуживания.

- Система GPSS была разработана сотрудником фирмы [IBM](#) Система GPSS была разработана сотрудником фирмы IBM [Джеффри Гордоном](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в [1961 году](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году. Гордоном были созданы 5 первых версий языка: GPSS (1961), GPSS II (1963), GPSS III (1965), GPSS/360 (1967) и GPSS V (1971). Известный ранее только специалистам, в нашей стране этот программный пакет завоевал популярность после издания в [СССР](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году. Гордоном были созданы 5 первых версий языка: GPSS (1961), GPSS II (1963), GPSS III (1965), GPSS/360 (1967) и GPSS V (1971). Известный ранее только специалистам, в нашей стране этот программный пакет завоевал популярность после издания в СССР в [1980 году](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году. Гордоном были созданы 5 первых версий языка: GPSS (1961), GPSS II (1963), GPSS III (1965), GPSS/360 (1967) и GPSS V (1971). Известный ранее только специалистам, в нашей стране этот программный пакет завоевал популярность после издания в СССР в 1980 году монографии [Т. Дж. Шрайбера](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году. Гордоном были созданы 5 первых версий языка: GPSS (1961), GPSS II (1963), GPSS III (1965), GPSS/360 (1967) и GPSS V (1971). Известный ранее только специалистам, в нашей стране этот программный пакет завоевал популярность после издания в СССР в 1980 году монографии Т. Дж. Шрайбера. В ней была рассмотрена одна из ранних версий языка — GPSS/360, а также основные особенности более мощной версии — GPSS V, поддерживаемой компанией [IBM](#) Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году. Гордоном были созданы 5 первых версий языка: GPSS (1961), GPSS II (1963), GPSS III (1965), GPSS/360 (1967) и GPSS V (1971). Известный ранее только специалистам, в нашей стране этот программный пакет завоевал популярность

- Система GPSS изучается во многих учебных заведениях в России и за рубежом. Широко используется для решения практических задач. Студенческая версия системы бесплатна. Однопользовательская лицензия персональной версии GPSS World 5.2.0.-23000 рублей. (<http://www.elina-computer.ru/>). Динамическим элементом модели является транзакт - абстрактный объект, который перемещается между статическими элементами, воспроизводя различные события реального моделируемого объекта. В процессе работы модели накапливается статистика, автоматически выводимая по завершении процесса моделирования. Статические элементы модели: источники транзактов, устройства, очереди и другие. Их расположение в модели определяется блоками.

- В математических моделях (ММ) сложных объектов , представленных в виде систем массового обслуживания (СМО), фигурируют средства обслуживания, называемые **обслуживающими аппаратами** (ОА), и обслуживаемые заявки, называемые **транзактами**. Так, в модели производственной линии ОА отображают рабочие места, а транзакты - поступающие на обработку детали, материалы, инструмент. Состояние СМО характеризуется состояниями ОА, транзактов и очередей к ОА. Состояние ОА описывается двоичной переменной, которая может принимать значения "занят" или "свободен". Переменная, характеризующая состояние транзакта, может иметь значения "обслуживания" или "ожидания". Состояние очереди характеризуется количеством находящихся в ней транзактов.

- **Имитационная модель** СМО представляет собой алгоритм, отражающий поведение СМО, т.е. отражающий изменения состояния СМО во времени при заданных потоках заявок, поступающих на входы системы. Параметры входных потоков заявок - внешние параметры СМО. Выходными параметрами являются величины, характеризующие свойства системы - качество ее функционирования. Примеры выходных параметров: **производительность** СМО - среднее число заявок, обслуживаемых в единицу времени; **коэффициенты загрузки оборудования** - отношение времен обслуживания к общему времени в каждом ОА; **среднее время обслуживания одной заявки**.

- Основное свойство ОА, учитываемое Основное свойство ОА, учитываемое в модели СМО, - это затраты времени на обслуживание, поэтому внутренними параметрами в модели СМО являются величины, характеризующие это свойство ОА. Обычно время обслуживания рассматривается как случайная величина и в качестве внутренних параметров фигурируют параметры законов распределения этой величины. Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и интенсивностях поступления заявок на входы, при вариациях параметров ОА, при различных дисциплинах обслуживания заявок.
-

- **Дисциплина обслуживания** - правило, по которому заявки поступают из очередей на обслуживание. Величина, характеризующее право на первоочередное обслуживание, называется **приоритетом**. В моделях СМО заявки, приходящие на вход занятого ОА, образуют очереди, отдельные для заявок каждого приоритета. При освобождении ОА на обслуживание принимается заявка из непустой очереди с наиболее высоким приоритетом.



- Основной тип ОА - устройства, именно в них происходит обработка [транзактов](#) с затратами времени. К ОА относятся также **накопители (памяти)**, отображающие средства хранения обрабатываемых деталей в производственных линиях или обрабатываемых данных в вычислительных системах. Накопители характеризуются не временами обслуживания заявок, а емкостью - максимально возможным количеством одновременно [находящихся](#) в накопителе заявок. К элементам имитационных моделей СМО кроме ОА относят также **узлы** и **источники заявок**. Связи ОА между собой реализуют **узлы**, т.е. характеризуют правила, по которым заявки направляются к тому или иному ОА.

## Пример кода для системы GPSS World

- GENERATE (POISSON(1,40)) ; генерация потока транзактов ; Поток транзактов пуассоновский со средним интервалом 40 единиц
- QUEUE mainQ ; Войти в регистратор очереди
- SEIZE F007 ; Попытка занять устройство
- DEPART mainQ ; Покинуть регистратор очереди
- ADVANCE (Normal(1,35,4)) ; Моделирование процесса обслуживания. Время обслуживания ; распределено по нормальному (гауссову) закону ; с математическим ожиданием 35 и среднеквадратическим отклонением 4
- RELEASE F007 ; Освободить устройство
- TERMINATE ; удаление транзакта

# Датчики БСВ

- **Базовой** случайной величиной (БСВ) в статистическом моделировании называют непрерывную случайную величину  $z$ , равномерно распределенную на интервале  $(0,1)$ . Ее плотность распределения вероятностей (**п.р.в.**) имеет вид:
- $f(t) = 1, 0 < t < 1.$  (2.1)
- Математическое ожидание (м.о.) и дисперсия БСВ составляют
- $M(z) = 1/2$  (2.2)
- $D(z) = 1/12$  (2.3)
- соотвественно.

- БСВ моделируется на ЭВМ с помощью *датчиков* БСВ. Датчик БСВ - это устройство или программа, выдающая по запросу одно или несколько независимых значений  $z_1, \dots, z_n$  БСВ.
- Датчики БСВ могут быть трех типов: табличные, физические и программные.
- **Табличный** датчик БСВ - это просто таблица случайных чисел. Основной недостаток такого датчика - ограниченное количество случайных чисел в таблицах. А в статистическом эксперименте часто требуется не ограниченное заранее их количество.
- **Физический** датчик БСВ - это специальное радиоэлектронное устройство в ЭВМ, содержащее источник электронного шума. Шум преобразуется в случайные числа с распределением (1.1). Недостатки физического датчика БСВ: невозможность повторения каких-либо ранее полученных реализаций  $z_1, \dots, z_n$  без их предварительной записи в память ЭВМ, схемная нестабильность и сложность тиражирования датчика.

- **Программный** датчик БСВ обычно вычисляет значения  $z_1, z_2, \dots$ , по какой-либо рекуррентной формуле типа
- $z_i = f(z_n)$ , (2.4)
- при заданном стартовом значении  $z_0$ .
- Заданное значение  $z_0$  полностью определяет всю последовательность реализаций  $z_1, z_2, \dots$ , поэтому  $z$  часто называют *псевдослучайной* величиной. Но ее статистические свойства идентичны свойствам "чисто случайной" последовательности, что и обеспечивает успех статистического моделирования.
- Программный датчик БСВ имеет следующие преимущества: простота создания датчика, простота применения, простота тиражирования, надежность, быстроедействие, высокая точность достижения необходимых статистических свойств, сравнимая с точностью представления вещественных чисел, компактность, повторяемость, когда это нужно, любых последовательностей случайных значений без их предварительного запоминания.
- В дальнейшем мы будем рассматривать только программные датчики БСВ.

- Имея датчик БСВ  $z$ , можно промоделировать любые случайные факторы: непрерывные или дискретные случайные величины (как простые, так и многомерные), случайные события, случайные процессы и поля и т.д. Для этого достаточно соответствующим образом преобразовать последовательность  $z_1, z_2, \dots$ . Поэтому БСВ  $z$  и называют базовой.
- Теоретически в качестве базовой можно было бы взять почти любую случайную величину (**с.в.**). Использование **с.в.**  $z$  с распределением (2.1) обусловлено технологическими соображениями: простотой и экономичностью датчика, простотой преобразования  $z$  в другие случайные факторы, относительной простотой тестирования датчика.

# Метод середины квадрата

- Метод середины квадрата предложен для получения псевдослучайных чисел Д. фон Нейманом в 1946 г. Вот один из вариантов этого метода.
  1. Возьмем произвольное 4-значное число.
  2. Возведем полученное число в квадрат и, если необходимо, добавим к результату слева нули до 8-значного числа.
  3. Возьмем четыре цифры из середины 8-значного в качестве нового случайного 4-значного числа.
  4. Если нужны еще случайные числа, то перейдем к 2.
- Например, если взять в качестве начального числа 1994, то из него получается следующая последовательность псевдослучайных чисел:  
9760 2576 6357 4114 9249 5440 5936 2360 5696 4444 7491 1150 3225 4006 0480  
2304 3084 5110 1121 2566 ...
- Сам по себе метод середины квадрата не получил широкого распространения, так как выдает "больше чем нужно малых значений" . Но открытый в нем принцип используется во многих, если не во всех, более поздних датчиках БСВ. Этот принцип состоит в вырезании нескольких цифр из результата какой-либо операции над числами.

# Мультипликативный конгруэнтный метод

- Так называемый мультипликативный конгруэнтный датчик БСВ задается двумя параметрами: модулем  $m$  и множителем  $k$ . Обычно это достаточно большие целые числа.
- При заданных  $m, k$  числа  $z_1, z_2, \dots$ , вычисляются по рекуррентной формуле:
  - $$A_i = (kA_{i-1}) \bmod m, \quad i = 1, 2, \dots, \quad (2.5)$$
  - $$z_i = A_i / m,$$
- где  $m$  - модуль,  $k$  - множитель,  $A_0$  - начальное значение,  $\bmod$  - операция вычисления остатка от деления  $kA_{i-1}$  на  $m$ .



- Таким образом,  $A_1$  определяется как остаток от деления  $kA_0$  на  $m$ ;
- $A_2$  - как остаток от деления  $kA_1$  на  $m$  и т.д. Поскольку все числа  $A_i$  - это остатки от деления на  $m$ , то  $0 < A_i < m$ .
- Разделив последнее неравенство на  $m$ , видим, что  $0 < A_i/m < 1$ , т. е.
- $0 < z_i < 1$ .
- Из неравенства  $0 < A_i < m$  вытекает также, что датчик (2.5) дает периодическую последовательность  $A_i$ . Действительно, число всех возможных остатков от 0 до  $m - 1$  равно  $m$  и, рано или поздно, на каком-то шаге  $i$  обязательно появится значение  $A_i$ , уже встречавшееся ранее. С этого момента последовательность  $A_i$  "зациклится".
- Длина периода  $T$  будет не больше  $m - 1$ . Например, если встретится остаток  $A_i = 0$ , то далее, согласно (2.5), будет  $A_{i+1} = 0, A_{i+2} = 0, \dots$ , т.е. длина периода  $T = 1$ . Ненулевых же остатков в интервале  $0 < A_i < m$  всего  $m - 1$ , и, если все они войдут в период, будет  $T = m - 1$ . Это имеет место, например, при  $m = 13, k = 7$ ; в этом случае ряд  $A_i$  выглядит так:
  - 1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1, 7, ... .
- $T = m - 1 = 12$

- Поскольку в качестве случайной можно использовать лишь подпоследовательность  $A_i$  внутри одного периода, то параметры датчика выбирают так, чтобы длина периода  $T$  была максимальной. С учетом ограничения  $T < m - 1$  модуль  $m$  берут максимально возможным. Чтобы упростить вычисление остатков по (2.5), для двоичных ЭВМ часто берут  $m = 2^n$ . Рекомендуется также брать достаточно большой множитель  $k$ , причем взаимно простой с  $m$ .
- *Можно найти подробные рекомендации по выбору параметров  $m$ ,  $k$  и начального значения  $A_0$ . Заметим, однако, что в настоящее время не известны правила, которые гарантировали бы высокое качество датчика без его специального статистического тестирования.*

- Датчик (2.5) называют мультипликативно-конгруэнтным потому, что он использует две основные операции - умножение (англ. multiplication) и вычисление остатка (в теории чисел - получение конгруэнтного числа). Можно было бы поэтому перевести его название и как "множительно-остатковый датчик".
- Обратим внимание также и на то, что операция вычисления остатка воплощает здесь упоминавшийся неймановский принцип вытаскивания цифр. Это становится очевидным, если записывать числа в системе счисления с основанием  $m$ . Тогда операция  $X \bmod m$  означает выбор последней цифры из числа  $X$ . Для  $m = 2^n$  операция  $X \bmod m$  означает также выделение последних  $n$  цифр из двоичной записи числа  $X$ .

МОДУЛЬ $m$	МНОЖИТЕЛЬ $k$
$2^{35}$	$5^{16}$
$2^{35}$	$5^{15}$
$2^{31} - 1$ (простое число)	16807
$2^{31} - 1$	63036001
$2^{40}$	$5^{17}$
$2^{36}$	$5^{15}$
$2^{42}$	$5^{17}$

# Тестирование равномерности

- Обозначим равномерное распределение вероятностей на интервале  $(0,1)$  через  $R[0,1]$ . Тогда утверждение, что БСВ  $z$  имеет распределение  $R[0,1]$ , можно кратко записать в виде  $z \sim R[0,1]$ .
- С помощью статистических тестов проверяют два свойства датчика, делающих его точной моделью идеальной БСВ, - это *равномерность* распределения чисел  $z_i$ , выдаваемых датчиком на интервале  $(0,1)$ , и их статистическая *независимость*. При этом числа  $z_i$  рассматривают как реализации некоторой с.в., т.е. как статистическую выборку.
- Достаточно простым методом проверки равномерности распределения является частотный тест. Он основан на законе больших чисел и выполняется по следующему алгоритму.

- 1. Разобьем интервал (0,1) на  $K$  равных отрезков (например,  $K = 10$ ).
- 2. Сгенерируем  $n$  чисел  $z_1, \dots, z_n$  с помощью тестируемого датчика БСВ (например,  $n = 100$ ).
- 3. Подсчитаем, сколько чисел попало в каждый из  $k$  отрезков, т.е. найдем числа попаданий  $n_1, \dots, n_k$ .
- 4. Рассчитаем относительные частоты попаданий в отрезки:
 
$$p_i = \frac{n_i}{n}$$
- 
- 5. Построим гистограмму частот  $p_1, \dots, p_k$  на  $K$  отрезках интервала (0,1).
- 6. Повторим действия (2) - (5) для большего значения  $n$  (например, для  $n = 10\,000$ ).
- 7. Оценим по полученным гистограммам сходимость каждой частоты  $p_i$  к вероятности  $p = 1/K$  того, что БСВ попадет в  $i$ -й отрезок. Согласно закону больших чисел должно быть
 
$$p_i \rightarrow 1/K$$
- 
- 

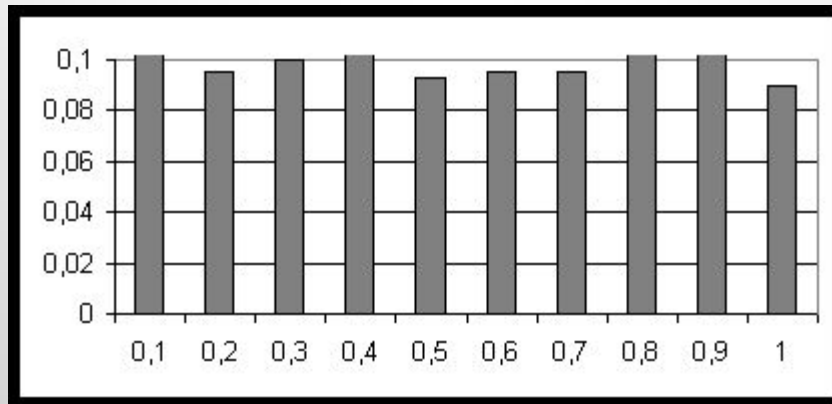
$$(2.6)$$

Это значит, что высоты столбиков во второй гистограмме должны в целом быть ближе к уровню  $1/K$ , чем в первой. Тестирование датчика на равномерность можно совместить с оцениванием м.о. и дисперсии с.в. Оценки  $\hat{M}$  и  $\hat{D}$  для м.о. и дисперсии рассчитываются соответственно по формулам:

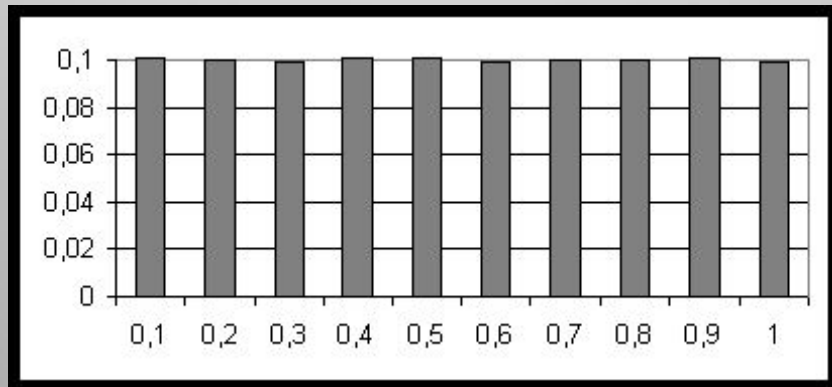
$$\hat{M} = \frac{1}{n} \sum_{i=1}^n z_i, \quad (2.7)$$

$$\hat{D} = \frac{1}{n} \sum_{i=1}^n z_i^2 - (\hat{M})^2. \quad (2.8)$$

С ростом  $n$  оценки должны сходиться по вероятности к точным значениям  $M(z) = 1/2$ ,  $D(z) = 1/12 = 0.08333\dots$



n=1000



n=100000



# Тестирование независимости

Простейшую проверку статистической независимости реализаций  $z_1, z_2, \dots$ , можно осуществить, оценивая корреляцию между числами  $z_i$  и  $z_{i+s}$ , отстоящими друг от друга на шаг  $s > 1$ .

Для вывода формулы, по которой можно рассчитать коэффициент корреляции чисел  $z_i$  и  $z_{i+s}$ , рассмотрим две произвольные с.в.  $x, y$ .

Коэффициент корреляции определяется для них формулой:

$$(2.9) \quad R(x, y) = \frac{M(xy) - M(x)M(y)}{\sqrt{D(x)D(y)}}.$$

- Если известно, что  $x, y \sim R[0,1]$ , то  $M(x) = M(y) = 1/2$  и  $D(x) = D(y) = 1/12$ , то есть (2.9) принимает вид:

- $$R(x,y) = 12 M(xy) - 3. \quad (2.10)$$

- Условимся рассматривать пару чисел  $(z_i, z_{i+s})$  как реализацию пары **с.в.**  $(x,y)$ . Тогда в выборке  $z_1, \dots, z_n$  имеем всего  $n - s$  реализаций этой пары:
- $(z_1, z_{1+s}), (z_2, z_{2+s}), \dots, (z_n, z_{n+s})$ .
- По ним можно рассчитать оценку  $R'$  коэффициента корреляции  $R(x,y)$ , заменяя в (2.10) **м.о.**  $M(xy)$  соответствующим средним арифметическим:

$$R' = 12 \frac{1}{n-s} \left( \sum_{i=1}^{n-s} z_i z_{i+s} \right) - 3.$$

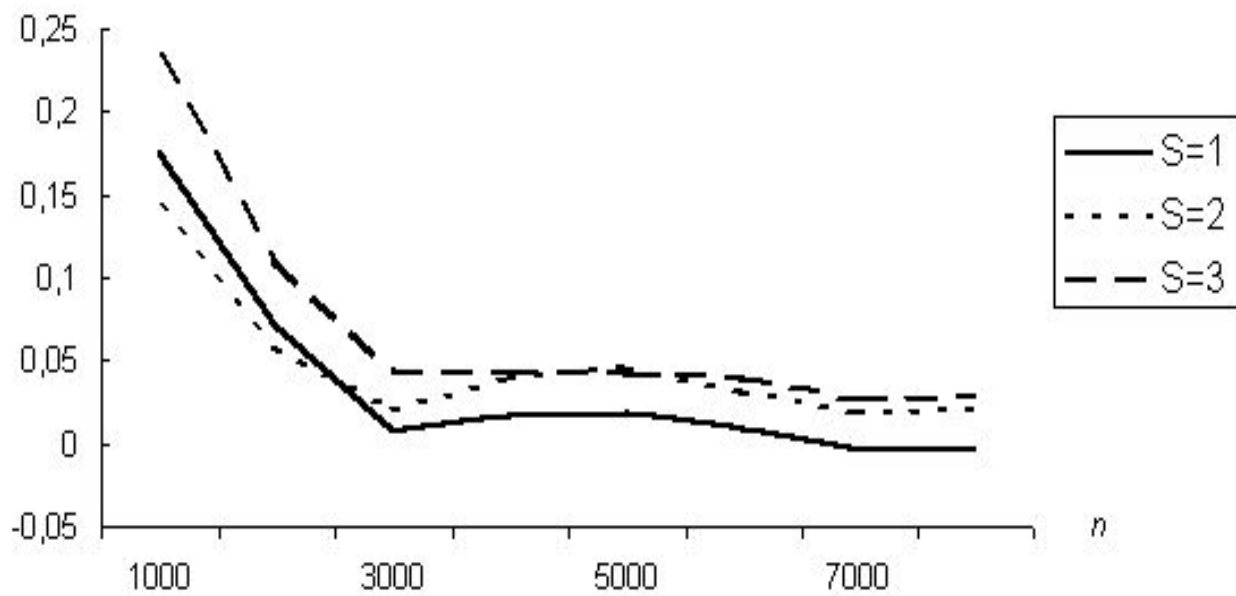
- С ростом  $n$  оценка  $R'$  должна приближаться к нулю, в противном случае датчик БСВ не отвечает требованию независимости.
- Конечно, если  $R'$  сходится к нулю, то это еще не гарантирует наличие независимости, но все же один из тестов оказывается успешно выдержанным. При желании всегда можно продолжить испытания датчика другими методами.

В табл. 2.3 показан пример тестирования датчика БСВ. Как видно из таблицы, оценки коэффициентов корреляции для  $s = 1, 2, 3$  постепенно сходятся к нулю.

На рис. 2.3 изображены соответствующие графики.

Таблица 2.3

$n$	Оценка $R'$ для $s = 1, 2, 3$		
	$s = 1$	$s = 2$	$s = 3$
1000	0.175	0.142	0.232
2000	0.071	0.059	0.111
3000	0.007	0.021	0.043
4000	0.017	0.040	0.043
5000	0.019	0.045	0.042
6000	0.009	0.031	0.039
7000	-0.003	0.019	0.027
8000	-0.003	0.021	0.028



- В настоящее время существуют десятки специальных методов, пригодных для тестирования датчиков БСВ. Однако эта проблема имеет не математическую природу, поэтому она не имеет корректного математического решения .
- Дж. Клейнен отмечает, что "Наилучший в настоящее время, по-видимому, мультипликативный генератор, однако он требует тщательного выбора сомножителя".
- Т.Дж. Шрайбер описал датчик БСВ, встроенный в язык GPSS (в версиях для IBM 360/370).
- Точное описание алгоритма и данные тестирования этого датчика не приводятся, как и в фирменной документации . Наше доверие к датчику может питаться авторитетом фирмы IBM и, в особенности, широким апробированием языка GPSS в прикладных исследованиях и в учебных курсах моделирования.
- Хорошим подтверждением высокого качества датчика можно считать также то, что на GPSS моделируются системы массового обслуживания и результаты сравниваются с точными аналитическими решениями.