ЛЕКЦИЯ 3. ЭЛЕМЕНТЫ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ

Составитель:

д.т.н. Колесникова С.И.

skolesnikova@yandex.ru

СУТЬ МЕТОДА МОНТЕ-КАРЛО

Метод статистического моделирования (Монте-Карло)

1) метод получения с помощью ЭВМ отдельных реализаций реальных (случайных, СВ) данных о процессах, происходящих в моделируемой системе;
2) базируется на использовании случайных чисел, т.е. возможных значений некоторой случайной величины с заданным распределением вероятностей;
3) множество полученных реализаций СВ обрабатываются и классифицируются с использованием методов математической статистики;

Случайный процесс моделируется при помощи генератора случайных величин: повторение значений СВ; на основе полученных случайных данных вычисляются вероятностные характеристики СВ.

История вопроса.

Н. Метрополис, С. Улам. Метод Монте-Карло, 1949.

Дж. Нейман, С. Улам (США, >1940).

В.В. Чавчанидзе, Ю.А. Шрейдер, В.С. Владимиров (1955-56).

ПРИМЕРЫ.

1. Расстояние между двумя множествами с определенными свойствами:

   $N(0,1), N(1,1)$

2. Длительность между наступлениями отказов в технических устройствах.
3. Стенд на проверку средней длительности работы электролампочек.

ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ МЕТОДА МОНТЕ-КАРЛО

Пусть требуется вычислить какую-то неизвестную величину ***m***.

Попытаемся реализовать значения некоторой ее замены в виде эмпирической СВ ξ таким образом, чтобы **M{ξ} =*m*** с известной дисперсией **D{ξ} =*b*²**.

Рассмотрим N реализаций $\xi_1,\dots,\xi_N$ теоретического распределения с характеристиками ξ.

Если N достаточно велико, то согласно ЦПТ распределение суммы

$\rho_N = \sum_{i=1}^{N} \xi_i \sim \mathrm{N}\left(Nm; Nb^2\right)$ будет приблизительно нормальным.

Метод расчета *m* и оценка погрешности метода Монте-Карло

$$P\left(\left|\frac{\rho_N}{N} - m\right| \le k\frac{b}{\sqrt{N}}\right) = P\left(\left|\frac{1}{N}\sum_i \xi_i - m\right| \le k\frac{b}{\sqrt{N}}\right) \to 2\Phi(k) - 1,$$

где Φ(x) — функция распределения стандартного нормального распределения.

Величины ***b*** и ***N*** определяют оценку погрешности метода МК (больше или меньше) в существенной зависимости от правильности выбранной модели для СВ ξ.

$$\Downarrow$$

Важность выбора СВ в методе Монте-Карло: выбрав, правильную случайную величину, можно получить более высокую точность вычислений, при меньшем числе итераций.

<u>Точность вычислений</u> зависит от количества ***N*** СВ $\xi_1,\dots,\xi_N$, включенных в сумму $\sum_{i=1}^{N} \xi_i$: для увеличения точности вычислений в 10 раз нужно увеличить ***N*** в 100 раз.

ПРИМЕРЫ ПРИМЕНЕНИЯ МЕТОДА МОНТЕ-КАРЛО

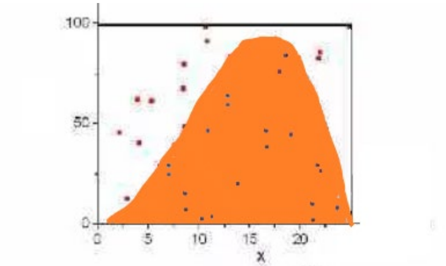
- расчет системы массового обслуживания;
- расчет качества и надежности изделий;
- теория передачи сообщений;
- вычисление определенного интеграла;
- задачи вычислительной математики;
- задачи нейтронной физики и др.

Джон фон Нейман, Станислав Улам и Николас Метрополис (Лос-Аламосская научная лаборатория), разработали алгоритм Метрополиса, также известный как метод Монте-Карло.

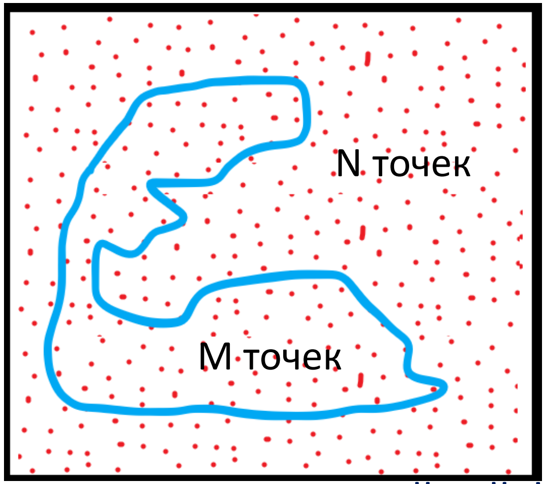
ММК. Базовый метод стохастической оптимизации

Первое полное испытание схемы Теллера — Улама (водородная бомба)

Стохастический подход для аппроксимации многомерных интегралов в уравнениях переноса, возникших в связи с задачей о движении нейтрона в изотропной среде.



$$S_{\text{функции}} = S_{\text{прямоуг.}} \frac{n_{\text{функции}}}{n_{\text{прямоуг.}}}$$



Площадь криволинейной фигуры

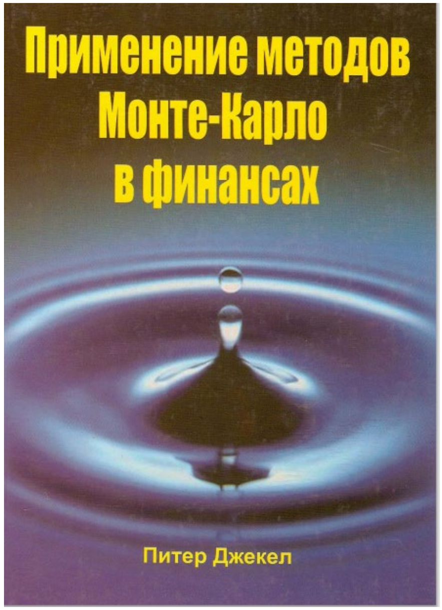
$$\frac{S_{крив.обл.}}{S_{прямоуг.обл.}} \cong \frac{M(N)}{N} \Rightarrow$$

$$S_{крив.обл.} = S_{прямоуг.обл.} \frac{M(N)}{N}$$

$$\frac{M(N)}{N} \cong \lim_{N\to\infty} \frac{M(N)}{N}.$$

1. Метод приближенный.
2. Распределение должно быть равномерным.
3. Чем больше точек, тем точнее.
4. Точность ограничена свойствами датчика ГПСЧ

ПРИМЕРЫ ПРИМЕНЕНИЯ МЕТОДА МОНТЕ-КАРЛО



Пример. В АСУ ТП передаются сигналы от производственного оборудования (объекта управления) к управляющему компьютеру. Длительность передачи сигнала - СВ, распределенная показательно со средним значением 5 мс.
В канале связи возможны помехи. Интервалы между моментами помех - СВ, распределенные показательно. Помехи возникают в среднем 20 раз в секунду.
Если во время передачи сигнала возникает хотя бы одна помеха, то сигнал искажается.

Пример. Поиск минимума функции

Инициализация:

$f_{min}=\infty$, $x_{min}=NaN$ (Not A Number – неопределенность типа (0/0)),

N – очень большое целое число – число генерируемых точек, $i:=0$,

1. Цикл: повторять пока $i < N$

    1.1. x_i := случайная точка

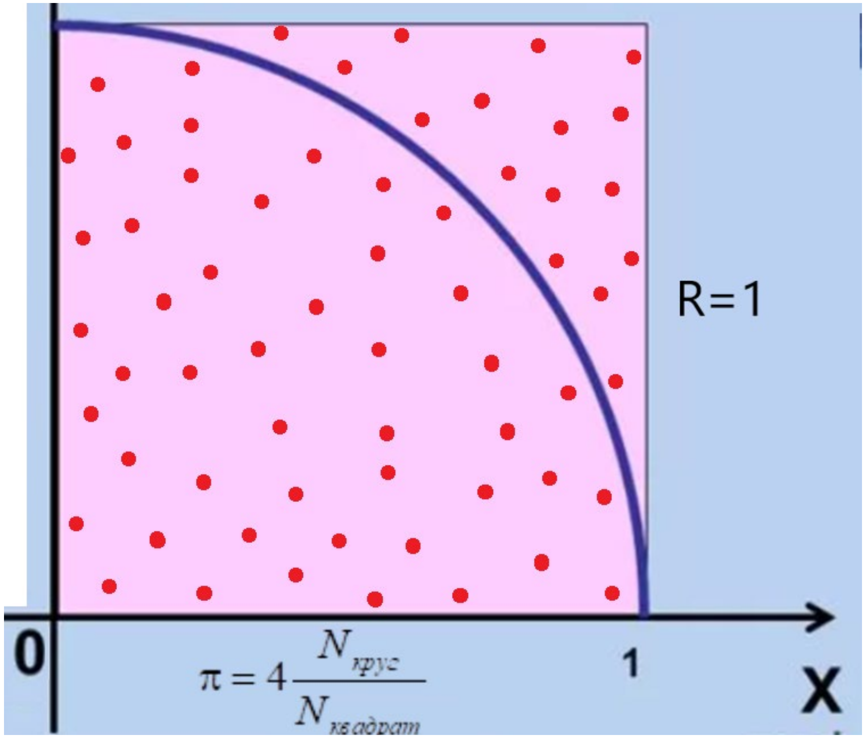
    1.2. Если $f(x_i)<f_{min}$, то $x_{min}=x_i$, $f_{min}=f(x_i)$

    1.3. i:=i+1 и перейти на шаг 1.1.

Если значения функции вычисляются в точках, полученных на основе равномерного распределения области S, наименьшее значение функции сходится к глобальному минимуму с вероятностью 1.

ПРИМЕРЫ ПРИМЕНЕНИЯ МЕТОДА МОНТЕ-КАРЛО

$$\pi = 4\frac{N_{круг}}{N_{квадрат}}$$



Номер испытания	T (точки внутри круга	T (точек всего)	Результат
1	18	31	3,21
2	20	36	3,17
3	**14**	**26**	**3,07**

ПРИМЕРЫ ПРИМЕНЕНИЯ МЕТОДА МОНТЕ-КАРЛО

Система контроля качества продукции состоит из трех приборов. Вероятность безотказной работы каждого из них в течение времени *T* равна 5/6. Приборы выходят из строя независимо друг от друга. При отказе хотя бы одного прибора вся система перестает работать. Найти вероятность $P_{отк}$ того, что система откажет за время *T.*

Аналитическое решение. События A – выход из строя хотя бы одного из трех приборов за время T и $\overline{A}$ – ни один из трех приборов не выйдет из строя за время T, противоположные.

$$P(\overline{A}) = \left(\frac{5}{6}\right)^3 \Rightarrow P_{отк} = P(A) = 1 - P(\overline{A}) = 1 - \left(\frac{5}{6}\right)^3 = 0.429420.$$

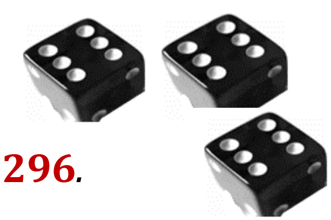
Решение задачи методом Монте-Карло (имитационное моделирование ситуации).

Возможны два подхода:

1) непосредственное проведение натурального эксперимента – наблюдение за работой системы в течение времени *T.* Многократное повторение этого эксперимента может оказаться трудноосуществимым или просто невозможным.

2) имитация другими экспериментами, имеющими с реальной ситуацией одинаковую вероятностную структуру.

Подбрасывание трех игральных костей: если выпадет одно очко, то будем считать, что прибор вышел из строя; если два, три, четыре, пять, шесть очков, то будем считать, что прибор работал безотказно. Вероятность того, что выпадет одно очко, так же как и вероятность выхода прибора из строя, равна 1/6, а вероятность того, что выпадет любое другое число очков, как и вероятность безотказной работы прибора, равна 5/6.



M – отказов системы; *N* – общее число проведенных испытаний.

Вероятность отказа будет равна: $P_{отк} = \frac{M}{N}$. При $N = 50000$ **$P_{отк}$(ММК)= 0.421296**.

ТИПЫ ДАТЧИКОВ БАЗОВЫХ ПСЕВДОСЛУЧАЙНЫХ ВЕЛИЧИН

Датчик случайной величины (СВ) (генератор псевдослучайных (ПСВ) чисел (ГСЧ)) – устройство, позволяющее по запросу получать реализацию a или несколько независимых реализаций $a_1,\dots,a_r$ базовой СВ α.

Существует три типа датчиков: 1) табличные; 2) физические; 3) программные.

Табличный датчик ПСВ – таблица случайных чисел, представляющая собой экспериментально полученную выборку реализаций равномерно распределенной в $[0,1)$ случайной величины. Недостатки табличного датчика ПСВ:

1)нехватка табличных случайных чисел;

2)расход оперативной памяти ЭВМ для хранения таблицы.

Физический датчик ПСВ –специальное радиоэлектронное устройство - приставка к ЭВМ, выходной сигнал которого имитирует ПСВ; монета; игральный кубик; вращающаяся стрелка, ГСЧ на основе радиоактивного распада; ГСЧ на основе шумов со встроенного микрофона в компьютере α. Недостатки физического датчика ПСВ:

1)невозможность повторения некоторой ранее полученной реализации a, т.к. $P\{\alpha = a\} = 0$.

2)схемная нестабильность: необходимость контроля работы датчика при очередном его использовании.

Программный датчик ПСВ – это программа, служащая для имитации на ЭВМ реализаций $a_1, a_2, \dots$базовой СВ; порождает последовательность *псевдослучайных чисел:*

по происхождению эти числа не случайные, а получаются по *известному математическому* закону.

Алгоритмы: линейный конгруэнтный, алгоритм Блюм-Блюма-Шуба, вихрь Мерсенна, метод Фибоначчи с запаздываниями и алгоритм AES.....

ПРИМЕР: ЛИНЕЙНЫЙ КОНГРУЭНТНЫЙ АЛГОРИТМ ГПСЧ (1949, Д. Г. Лемер)

$$X_{n+1} = (aX_n + c) \, mod \, m$$

X_n – текущее случайное число,
X_{n+1} – следующее случайное число,
m – модуль, $m \geq 2$,
a – множитель, $0 \leq a < m$,
c – приращение, $0 \leq c < m$,
X_0 – начальное значение последовательности, $0 \leq X_0 < m$.

Периодичность получаемой последовательности меньше или равна **m**.
Пример параметров:
a = 9219741426499971445, c = 1, m = 2^{63}-1.

Данный метод широко применяется там, где от ГПСЧ не требуется криптографической стойкости.
Самым известные примеры - генератор Парка-Миллера и генератор RANDU.

ПРИМЕР: МЕТОД ФИБОНАЧЧИ С ЗАПАЗДЫВАНИЯМИ (1958, Дж. Митчелом, Д. Мур)

$$X_k = (X_{k-a} + X_{k-b})\ mod\ (2^m)$$

X_k – произвольные целые числа, не всегда являющиеся четными;

a,b – целые положительные числа (так называемые «запаздывания»);

m – четное число.

Числа a и b для данного метода являются «магическими», и их не стоит выбирать случайным образом. При увеличении их значений вырастает размерность пространства, в котором не теряется равномерность получаемых последовательностей случайных чисел. Однако, в связи с этим так же увеличивается объем требуемой памяти.

Существует множество различных пар чисел a и b.

Разновидность алгоритма с параметрами **a = 24, b = 55, m = 64** (Дж. Марсаглия, 1985). При таких параметрах реализованный генератор предоставляет периодичность в 2^{116}-1 чисел.

На основе этого метода было создано множество различных алгоритмов шифрования (Fish, Pike,...).

ЭТАПЫ МОДЕЛИРОВАНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН

1) генерирование N реализаций случайной величины с требуемой функцией распределения;

Особенность: **все методы для получения заданного распределения используют преобразование равномерно распределенной величины**:

- задается СВ, равномерно распределенная в интервале (0,1);
- осуществляется обратное отображение;
- фиксируется новая СВ с заданным распределением.

2) использование полученной СВ в контексте моделируемого объекта;

3) статистическая обработка реализации.

ПРИМЕР. Генерация случайных чисел в C#

Основа	Варианты
//Создание объекта типа «*Random*», для генерации	//Создание объекта для генерации чисел (с
чисел из (0,1)	указанием начального значения)
Random rnd = new Random();	Random rnd = new Random(333);

//Получить очередное (в данном случае - первое)	//Получить случайное число (в диапазоне от
случайное число на основе вызова метода «*Next*»	0 до 10)
объекта «*Random*»	int value = rnd.Next(0, 10);

Сравнительная характеристика реализаций алгоритмов ГПСЧ

Название алгоритма	Линейный конгруэнтный метод	Метод Фибоначчи с запаздываниями	Алгоритм Блюм-Блюма-Шуба	Вихрь Мерсенна	AES	ГСЧ из модуля secrets
Плюсы	порождает статистически «хорошую» последовательность чисел	рекомендуем для использования в статистических алгоритмах и в моделировании	криптостойкий; прост в реализации	большая периодичность; плохо предсказуем; хорошая скорость	криптостойкий; широко распространен; различные области применения в информационной безопасности	криптостойкий; сравнительно быстрый; использует в качестве источника энтропии источник операционной системы
Минусы	предсказуемость; нулевая криптостойкость; малая периодичность	не является криптостойким; относительно большая вычислительная сложность	низкая скорость; линейная вычислительная сложность	не является криптостойким; долго «разогревается»	производительность ниже, чем у простых ГПСЧ	является базовым решением безопасности; в будущем могут возникнуть проблемы с безопасностью ключей
Реализация	на основе исходного кода [8]	на основе исходного кода [8]	на основе алгоритма из [14]	стандартный модуль random языка Python	модуль Fortuna библиотеки PyCrypto	стандартный модуль secrets языка Python
Период	2^{63}-1	2^{116}-1	N/A	2^{19937}-1	N/A	N/A[2]

Для выбора последовательности случайных цифр можно взять дробную часть числа π

(π = 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211 7…)

АККУРАТНОСТЬ ИСПОЛЬЗОВАНИЯ ГЕНЕРАТОРА СВ

<table>
<tr><th>Разные числа</th><th>Одинаковые числа</th></tr>
<tr><td>
<pre>
using System;
namespace MyProgram
{
  class Program
  {
    static void Main(string[]
args)
    {
     Random rnd = new Random();
      for (int i = 0; i < 10;
i++)
      Console.WriteLine("{0,4}",
rnd.Next(-10,10));
      Console.ReadKey();
    }
  }
}
</pre>
</td><td>
<pre>
using System;
namespace MyProgram
{
  class Program
  {
    static void Main(string[]
args)
    {
      for (int i = 0; i < 10;
i++)
      {
        Random rnd = new
Random();

Console.WriteLine("{0,4}",
rnd.Next(-10, 10));
      }
      Console.ReadKey();
    }
  }
}
</pre>
</td></tr>
</table>

!!! При создании нового экземпляра класса Random каждый раз, когда требуется сгенерировать случайное число, можно получить одинаковые генерируемые числа.

9
8
4
8

7
7
7
7

ПРИМЕНЕНИЕ ГЕНЕРАТОРА СВ В МЕТОДЕ ОБРАТНОЙ ФУНКЦИИ ДЛЯ ПОЛУЧЕНИЯ СВ С ЗАДАННЫМ РАСПРЕДЕЛЕНИЕМ

Теорема. (Смирнов Н.В.). Если X удовлетворяет уравнению

$$\int_{-\infty}^{X} f(t)dt = \alpha,\ \alpha \in [0;1],$$

где $\alpha,\ \alpha \in [0;1]$ - СВ, равномерно распределенная на $[0;1]$, то СВ X распределена по закону $F(X) \triangleq \int_{-\infty}^{X} f(t)dt$. Подробно:

$$F(X) \triangleq \int_{-\infty}^{X} f(t)dt \triangleq \int_{-\infty}^{X} F'(t)dt = F(X) = \alpha,\ \alpha \in [0;1] \Rightarrow X = F^{-1}(\alpha)$$

ПРИМЕР 1. Получить СВ с функцией показательного распределения вероятностей.

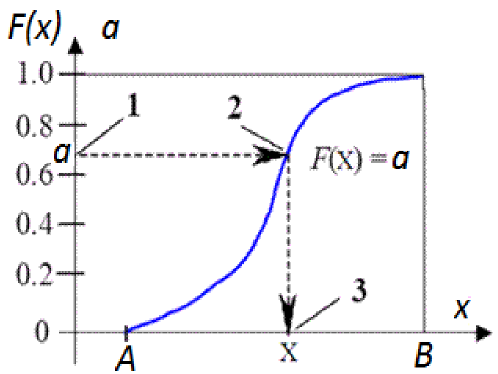
Решение. $f(x)=\lambda e^{-\lambda x} \Rightarrow F(x)=1-e^{-\lambda x}$.

1)Получим случайное число α, равномерно распределенное в интервале от 0 до 1.

2)Полагаем $\alpha = 1 - e^{-\lambda x}$.

3)Выражаем величину x из этого уравнения, получаем: $x = (-1/\lambda)\cdot\ln(1 - \alpha)$.

ПРИМЕР 2. Получить СВ с функцией произвольного распределения вероятностей с ограничением: $X \in [A;B]$



$$X = F^{-1}(\alpha)$$

Пример 3. Получить СВ с функцией равномерного распределения вероятностей на отрезке $[a,b]$.

Решение. $f(x)$ – плотность равномерного распределения $R[a,b]$:

$$f(x)=\begin{cases}\dfrac{1}{b-a}, & \text{если} \quad x\in[a,b],\\ 0, & \text{если} \quad x\notin[a,b].\end{cases}$$

Функция распределения $F(x)=\begin{cases}0, & \text{если} \quad x<a\\ \dfrac{x-a}{b-a}, & \text{если} \quad x\in[a,b],\\ 1, & \text{если} \quad x>b.\end{cases}$

$F(x)=\alpha \Rightarrow x=(b-a)\cdot\alpha+a, \quad \alpha\in[0,1]. \Rightarrow$

Моделирование требуемой СВ с равномерным распределением $R[a,b]$ ξ осуществляется по формуле

$\xi=(b-a)\cdot\alpha+a,$ где $\alpha \sim R[0,1]$.

ЭТАПЫ МОДЕЛИРОВАНИЯ

1. ПОСТАНОВКА ЗАДАЧИ и обоснование КРИТЕРИЯ ОПТИМАЛЬНОСТИ. Для выбора критерия оптимальности необходимо провести сравнение возможных критериев и выбрать соответствующий задачам эксперимента.
2. Разработка структурной математической модели: выбор метода решения поставленной задачи, определяются ограничения и переменные, производится унификация символики, ищутся аналоги постановки задачи.
3. Сбор и обработка информации (базы данных и т.д., предобработка, сокращение выборки, ...).
4. Математическое решение задачи (до возможного уровня).
5. Построение числовой модели (программа расчета на ЭВМ полученных ключевых формул).
6. Анализ решения. Оценка АДЕКВАТНОСТИ полученного решения. Ретроспективные расчеты по модели, сопоставление с имеющимися результатами других исследователей, предыдущими данными, расчетами по другим моделям, экспертными оценками и т.д.
7. Корректировка задачи при установлении неадекватности, Установление областей применимости модели, границ параметров по каждому эндогенному параметру и областей применимости модели по экзогенным параметрам.
8. Написание отчета по исследованию модели, подведение итогов, формулирование выводов и предложений, построение прогнозов развития исследуемого объекта, ВЫЯВЛЕНИЕ СВЯЗЕЙ МЕЖДУ ПАРАМЕТРАМИ МОДЕЛИ И РЕЗУЛЬТАТОМ.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АНАЛИТИЧЕСКИХ И ИМИТАЦИОННЫХ МОДЕЛЕЙ СЛОЖНЫХ ОБЪЕКТОВ

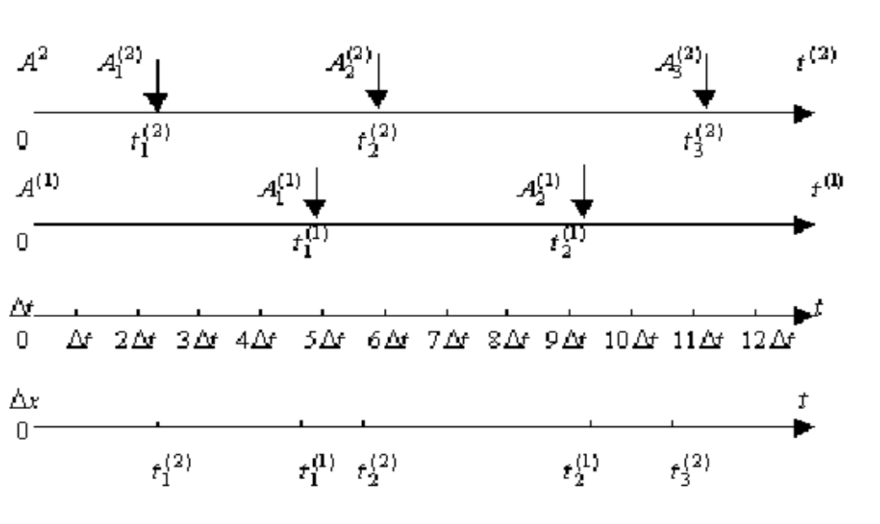
Аналитическая модель (АМ)	*Имитационная модель* (ИМ)
совокупность функциональных соотношений, логических условий, описывающих связи между параметрами, переменными и показателями эффективности системы S	программы (алгоритмы), позволяющие имитировать на ЭВМ поведение отдельных элементов системы S и связей между ними в течение заданного времени моделирования
Условия применения	
- сравнительно простые (как правило, линейные или корректно линеаризуемые) системы; - упрощенные примитивы реальных систем с целью изучения некоторых свойств системы	- широкий класс систем практически любой сложности, аналитические модели которых частично или полностью не определенны; - в случаях, когда практическое построение и/или использование АМ невозможно
Достоинства	
универсальность, высокая степень общности и значимости результатов	- законы функционирования всей системы в целом могут быть неизвестными; *возможности исследования*: - системы на различных уровнях ее детализации; - динамики взаимодействия элементов системы во времени и пространстве; - оценивания характеристик системы в разные моменты времени
Недостатки	
- чувствительность к степени сложности системы; - частичная возможная неадекватность реальной системе	- дороговизна, требует больших временных затрат; - меньшая степень общности по сравнению с АМ: не выявляют общие закономерности функционирования классов систем; - не существует надежных методов оценки адекватности ИМ.

МОДЕЛЬНОЕ И РЕАЛЬНОЕ ВРЕМЯ. ВРЕМЕННАЯ ДИАГРАММА В ИМ

Модельное время t – для обеспечения имитации параллельных или одновременных событий системы S.

Реальное время системы S функционирование которой имитируется.

Машинное время имитации отражает затраты ресурсов времени ЭВМ на организацию имитационного моделирования.

<table>
<tr><th>Состав системы: $S=\{A^{(1)},A^{(2)}\}$</th><th>Временные оси локального модельного времени</th></tr>
<tr><td rowspan="4">
</td><td>t^1 для элемента A^1</td></tr>
<tr><td>t^2 для элемента A^2</td></tr>
<tr><td>принцип «Δt» (приращения времени)
$A_1^{(2)}\sim 3\Delta t, A_1^{(1)}, A_2^{(2)}\sim 6\Delta t, A_2^{(1)}\sim 10\Delta t, A_3^{(2)}\sim 12\Delta t.$</td></tr>
<tr><td>модельного времени по принципу «Δx» (приращения состояния)
$x(t^*+0)=x(t^*)+\Delta x, t^*\in T$</td></tr>
</table>

$x(0), x(\Delta t)=x(2\Delta t)=x(0),\ x(3\Delta t)=x(t_1^{(2)}), x(4\Delta t)=x(3\Delta t), x(5\Delta t)=x(t_1^{(1)}),$

$x(6\Delta t)=x(t_2^{(2)}), x(7\Delta t)=x(8\Delta t)=x(9\Delta t)=x(6\Delta t), x(10\Delta t)=x(t_2^{(1)}),$

$x(11\Delta t)=x(10\Delta t), x(112\Delta t)=x(t_3^{(2)}).$

ОСНОВНЫЕ ПОЛОЖЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Шаг 1. Построение математической модели М.

Шаг 2. Разработка моделирующего алгоритма и построение имитационной модели ИМ.

Шаг 3. Исследование системы S с помощью модели ИМ: проведение имитационных экспериментов, обработка и интерпретация результатов.

Между шагами существует обратная связь, обеспечивающая уточнение, корректировку и учет дополнительной информации при разработке и использовании имитационной модели.

Роберт Шеннон:

1. «Проверка модели – этап чрезвычайно важный, поскольку имитационные модели вызывают впечатление реальности, и как разработчики моделей, так и их пользователи легко проникаются к ним доверием. К сожалению, для случайного наблюдателя, а иногда и для специалиста, искушенного в вопросах моделирования, бывают скрыты исходные предположения, на основе которых строилась данная модель. Поэтому проверка, выполненная без должной тщательности, может привести к катастрофическим последствиям».
2. Время, необходимое на проектирование и работу с моделью:
3. **25% на постановку задачи, 20% на сбор и анализ данных,
   30% на разработку модели,25% на реализацию.**
4. ....интерес представляет не изящная модель и искусное использование математических методов, а реальные проблемы и способы их решения.

ЛИТЕРАТУРА ПО ММК

1. Н. Метрополис, С. Улам. Метод Монте-Карло, 1949.
2. Дж. Нейман, С. Улам (США, >1940).
3. В.В. Чавчанидзе, Ю.А. Шрейдер, В.С. Владимиров (1955-56).
……………………………………………………………………………………………………
4. Есипов А.С., Паньгина Н.Н., Громада М.И. Информатика (задачник). Санкт-Петербург, “Наука и Техника”, 2001
5. Севастьянов Б.А. Вероятностные модели. М., “Наука”, 1992
6. Пайтген Х.-О., Рихтер П.Х. Красота фракталов. М., “Мир”, 1993
7. И.М.Соболь «Метод Монте-Карло», М., 1985
8. /GIS/Learning/Monte-Carlo_2/Page01.htm
9. /~gene/probset/prob13.koi8.html
10. /Exponenta_Ru/educat/systemat/boziev/13.asp.htm
11. /2001/leto/stend/Vynderkind.htm
12. /docs/TViMS/NP/lekziitv/lekziya17.htm

КЛАССИЧЕСКАЯ ЛИТЕРАТУРА ПО ТМО

1. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. М., 1987.
2. Тихоненко О. М. Модели массового обслуживания в информационных системах: Учебное пособие для вузов. - Минск: "Технопринт", 2003. – 326 с.
3. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. Высш.шк., 2000, 480 с.
4. Вентцель Е. С. Исследование операций. М., «Советское радио», 1972, 552 с.
5. Т.Л.Саати. Элементы теории массового обслуживания и ее приложения: Пер. с англ. /Под. ред. И.Н. Коваленко, изд-ие 2. М., 1971.
6. Д.Кениг, Д.Штойян. Методы теории массового обслуживания: Пер. с нем. /Под. ред. Г.П.Климова. М., 1981.
7. Г.И.Ивченко, В.А.Каштанов, И.Н.Коваленко. Теория массового обслуживания. М., 1982.
8. Хинчин А.Е. Работы по математической теории массовогообслуживания / ред.: Б.В. Гнеденко. - 2-е изд., стереотип. - М. : УРСС, 2004. – 235с.
9. Фомин Г.П. Математические методы и модели в коммерческой деятельности : Учебник для вузов / Г. П. Фомин. - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2005. - 614с.

1.