Метод динамики средних

- Часто целью моделирования является определение средних количеств элементов, находящихся в одинаковых состояниях, а не вероятности каждого состояния.
- Метод динамики средних позволяет непосредственно определять математическое ожидание числа элементов сложной системы, находящихся в одинаковых состояниях.
- Метод дает приближенные результаты. Но обладает замечательным свойством: чем больше система имеет элементов и состояний, тем точнее результат математического моделирования.
- Принимаются следующие допущения:
  - в системе протекает случайный марковский процесс;
  - элементы системы однородны – они блуждают по одним и тем же состояниями, т.е. состояния, их число и их вероятности - одинаковые;
  - элементы меняют состояния независимо друг от друга.

Метод динамики средних

Поставим задачу следующего вида:

- N элементов и M состояний.
- Пусть элементы системы блуждают по своим состояниям: k=1…..M.
- Найти среднюю численность k-го состояния (мат. ожидание) и дисперсию k-го состояния.

  $M_k(t)=?$

  $D_k(t)=?$
- Введём в рассмотрение случайную величину $x_k(t)$ – численность элементов, находящихся в k-ом состоянии.
- Тогда $x_1(t)+x_2(t)+\ldots+x_M(t)=N$

Метод динамики средних

$X_k^{(i)}(t) = 1$, если i-ый элемент находится в состоянии k;
$X_k^{(i)}(t) = 0$, в противном случае.

$$x_k(t) = \sum_{i=1}^{N} x_k^{(i)}(t)$$

Т.к. элементы однородны, состояния независимы, то случайная величина $x_k(t)$ имеет биномиальное распределение (распределение Бернулли).

$$M_k(t) = N * P_k(t)$$

$$D_k(t) = N * P_k(t) * (1 - P_k(t))$$

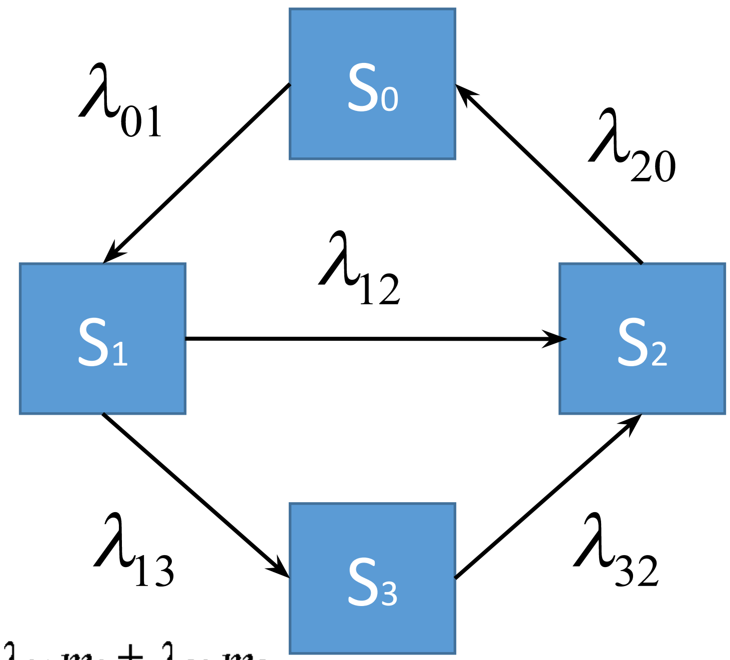
Таким образом, получаем связь между вероятностью k- го состояния элемента в произвольный момент времени с математическим ожиданием численности этих состояний по всем элементам.
Т.е. достаточно определить вероятности $P_k(t)$ для одного элемента.

Метод динамики средних

Алгоритм:

- Описать состояния одного элемента системы.
- Нарисовать граф состояний для одного элемента.
- Получить для каждого состояния его среднюю численность.
- Составить дифференциальные уравнения динамики средних. При этом каждый член в уравнении равен произведению интенсивности данного перехода на среднюю численность состояния до перехода. Исходящий поток имеет знак минус, входящий – плюс.
- Решить систему дифференциальных уравнений относительно средних численностей.
- Вычислить значения дисперсий и средних квадратических отклонений.

Метод динамики средних



$$S_0 : \mu p_1 - \lambda p_0 = 0 \Rightarrow p_1 = \frac{\lambda}{\mu} p_0$$

$$S_0 : \frac{d\, p_0 N}{t} = -\lambda_{01} p_0 N + \lambda_{20} p_2 N \Rightarrow \frac{d\, m_0}{t} = -\lambda_{01} m_0 + \lambda_{20} m_2$$

$$S_1 : \frac{d\, m_1}{t} = -(\lambda_{12} + \lambda_{13}) m_2 + \lambda_{12} m_0$$

$$S_2 : \frac{d\, m_2}{t} = -\lambda_{20} m_2 + \lambda_{12} m_1 + \lambda_{32} m_3$$

$$S_3 : \frac{d\, m_3}{t} = -\lambda_{32} m_3 + \lambda_{13} m_1$$

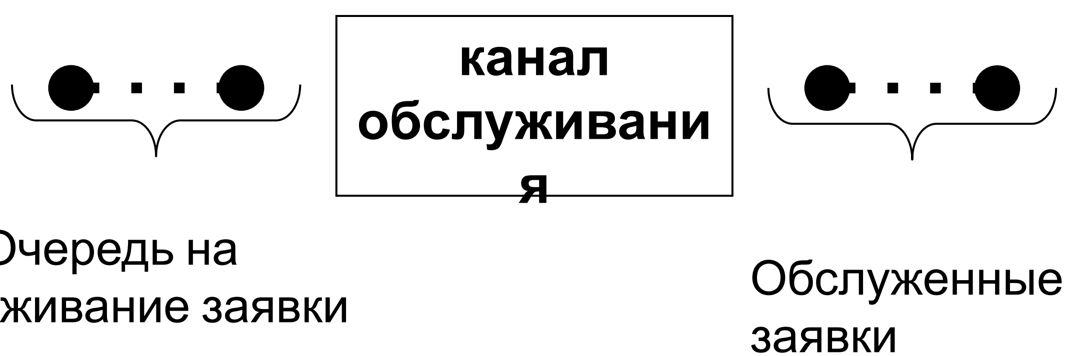
$$\sum_{k=1}^{N} m_k(t) = 1 * N$$

Системы массового обслуживания

СМО. Основные понятия

Под СМО понимают системы, на вход которых подается случайный поток однотипных заявок (событий), которые обрабатывают однотипными каналами или устройствами.

Условно систему массового обслуживания можно представить следующим образом:



СМО. Основные понятия

- Каналом обслуживания называется техническое или какое-либо другое устройство, обеспечивающее выполнение какой-либо заявки.
- В теории массового обслуживания работа канала характеризуется тем временем, которое затрачивается на обслуживание одной заявки.
- В общем случае оно считается случайным и часто время обслуживания в теории массового обслуживания принимают подчиняющимся пуассоновскому закону распределения.
- Тогда на выходе канала обслуживания будет появляться обслуженные заявки с интервалом времени, подчиняющимся пуассоновскому закону распределения.
- Также, если на входе системы существует очередь и каждая заявка обслуживается независимо от других и не покидает систему во время обслуживания, а только после его завершения, то интервалы времени между обслуженными заявками будут также независимы и подчинены пуассоновскому закону распределения.

Структура СМО

Структура системы задается:

- потоком заявок,
- количеством приборов (каналов) в системе,
- длительностями обслуживания,
- числом мест ожидания,
- в сложных системах – алгоритмом прохождения заявки по различным каналам системы.

Структура СМО

- Однородный поток заявок полностью описывается случайными моментами их поступления в СМО, то есть ему соответствует случайный процесс.
- В случае параллельных каналов и обслуживания заявок на любом свободном канале, СМО является многоканальной. При единственном канале – одноканальной.
- Если обслуживание идет последовательно на разных приборах, то выделяют:
  - многофазные СМО, если одновременно на разных приборах могут обслуживаться несколько заявок;
  - многоэтапные СМО, если одновременно возможно обслуживание только на одном приборе.

Дисциплина обслуживания

- **Дисциплина обслуживания** – это порядок распределения заявок между свободными каналами, кроме того это закон образования очереди, закон поведения заявок в очереди и порядок обслуживания заявок.
- Различают следующие варианты порядка обслуживания:
  - прямой;
  - инверсионный;
  - случайный;
  - с разделением времени;
  - с разделением процессора;
  - пакетный
  - и др.

Порядок обслуживания

- **Прямой порядок** обслуживания – обслуживание в порядке поступления – FIFO (First In First Out).
- **Инверсионный или стековый порядок** обслуживания – обслуживание заявки, поступившей в систему последней – LIFO (Last In First Out).
- **Случайный порядок** обслуживания – любая заявка, из имеющихся в очереди, может быть выбрана на обслуживание с равной вероятностью.
- **Режим разделяемого времени** – вся длительность обслуживания разбивается на этапы, после завершания некоторого этапа тем или иным образом выбирается одна из заявок для обслуживания её очередного этапа.

Порядок обслуживания

- **Режим равномерного разделения процессора** – процессор (т.е. канал) может одновременно обслуживать несколько заявок. Но при этом интенсивность (т.е. скорость) обслуживания каждой заявки уменьшается во столько же раз, сколько одновременно заявок обрабатывается процессором.
- **Пакетное обслуживание**. Задается некоторое правило формирования пакетов – например, в пакет, который поступает на обслуживание, включают все заявки, которые находились в системе на момент освобождения канала от предыдущего пакета.

Порядок обслуживания

Кроме того, заявкам, тем или иным способом, может назначаться приоритет. Это обычно требуется делать при наличии разнородных требований в системе.

Приоритетные правила приписывают каждому типу заявок соответствующее значение приоритета. Способы задания могут быть следующие:

- относительный;
- абсолютный;
- чередующийся;
- в порядке возрастания длительности обслуживания;
- в порядке убывания длительности обслуживания
- и др.

Порядок обслуживания

- **Относительный приоритет** – при завершении обслуживания очередной заявки, из очереди следующей выбирается заявка с наивысшим приоритетом.
- **Абсолютный приоритет** – при поступлении в СМО заявки с более высоким приоритетом, чем заявка, которая обслуживается в данный момент, происходит прерывание обслуживания. Поступившая заявка начинает обслуживаться прибором(каналом), а прерванная заявка:
  - может быть возвращенной в очередь с последующим дообслуживанием;
  - может быть возвращенной в очередь и затем быть обслуженой заново;
  - может быть отброшена.

Порядок обслуживания

- **Чередующийся приоритет**. За всеми заявками того же типа, что и обслуживающаяся в данный момент, закрепляется наивысший приоритет. После того, как все такие заявки будут обслужены, из очереди некоторым образом выбирается следующее требование, например по правилу относительных приоритетов.
- При обслуживании в первую очередь требований с **наименьшим временем обслуживания**, можно получить оптимальное обслуживание, а именно, - минимальное среднее время пребывания в системе требований при отсутствии задержек и потерь из-за переориентации и изменения порядка обслуживания. Однако, чаще всего неизвестно заранее, какова будет длительность обслуживания.

Характериситки СМО

Работа СМО оценивается с помощью параметров эффективности:

- среднее число заявок, обслуженных в единицу времени;
- среднее число занятых каналов;
- средняя длина очереди;
- среднее время ожидания в очереди;
- среднее число заявок, получивших отказ в обслуживании и т.д.

Также для анализа могут быть интересны следующие вероятности:

- вероятность застать систему в свободном состоянии;
- вероятность потери заявки из-за занятости системы;
- вероятность ожидания заявкой начала обслуживания в очереди при поступлении в систему;
- вероятность нахождения в систему определенного количества заявок.

Задачи СМО

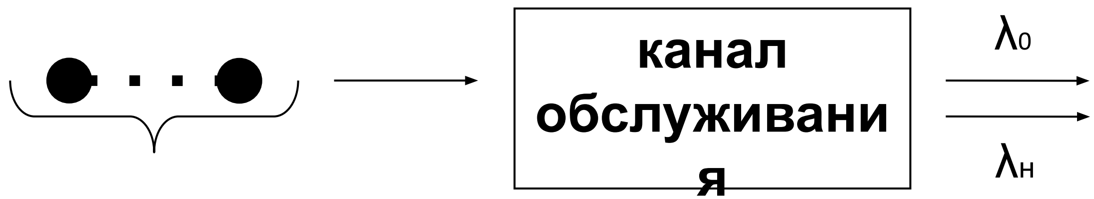
Можно выделить несколько основных классов задач:

- выяснение условий существования стационарных распределений искомых характеристик;
- нахождение значений определенных вероятностей, распределений вероятностей и их числовых характеристик.
- оценка характеристик СМО при помощи реальных наблюдений или имитационного моделирования;
- сравнение и оптимизация СМО по различным критериям за счет подбора значений числовых параметров системы, порядка обслуживания, структуры и т.д.
- исследование асимптотического поведения характеристик СМО при стремлении каких-либо парметров к некоторым граничным значениям.

Замкнутые и разомкнутые СМО

- **Разомкнутые системы**: на вход поступает извне некоторый поток заявок, причем источник заявок в систему не входит и анализу не подвергается. Разомкнутые системы массового обслуживания бывают **с отказами** и **без отказов**: если система свободная, она обслуживается, если нет, то получает отказ на обслуживание.
- В **замкнутой системе** число источников заявок ограничено и интенсивность поступления заявок зависит от состояния источников, обусловленных работой самой системы. При поступлении заявки на обслуживание заявка либо обслуживается, либо, если система занята, уходит из очереди. Очередь может быть **ограниченная** и **не ограниченная.** В замкнутой СМО заявка рано или поздно обслуживается, то есть никогда не получает отказа.

Разомкнутые СМО



На вход поступает поток заявок на обслуживание с интенсивностью λ.

На выходе:

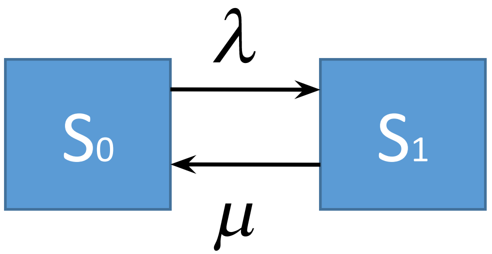
- λ_0 - интенсивность потока обслуженных заявок;
- $\lambda_н$ - интенсивность потока необслуженных заявок.

$$\lambda = \lambda_0 + \lambda_H$$

$$1 = P_0 + P_H$$

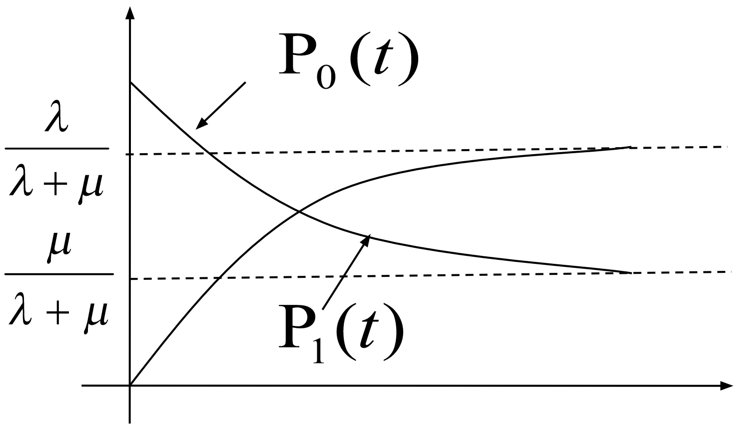
Одноканальная разомкнутая СМО

$$\begin{cases} \dfrac{dP_0}{dt} = -\lambda P_0 + \mu P_1 \\ \dfrac{dP_1}{dt} = -\mu P_1 + \lambda P_0 \end{cases}$$



$$P_0(0) = 1;$$

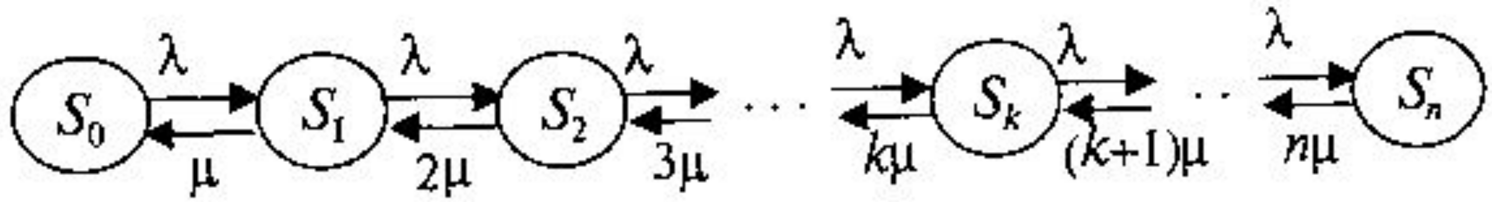
$$P_1(0) = 0;$$



$$t -> \infty : P_{Обсл.} = P_0 = \frac{\mu}{\lambda + \mu}$$

$$P_{Отк.} = 1 - \frac{\mu}{\lambda + \mu} = \frac{\lambda}{\lambda + \mu}$$

Многоканальная СМО с отказами



- все каналы однотипны;
- время обслуживания заявки случайно и образует простейший поток интенсивностью μ;
- поступающий поток заявок интенсивностью λ будем считать простейшим;
- если все каналы заняты, то очередная заявка не будет удовлетворена.

Многоканальная СМО с отказами

Уравнения Колмогорова для финальных вероятностей:

$$S_0 : \mu\, p_1 - \lambda\, p_0 = 0 \Rightarrow p_1 = \frac{\lambda}{\mu} p_0$$

$$S_1 : 2\mu\, p_2 + \lambda\, p_0 - (\lambda + \mu)\, p_1 = 0 \Rightarrow p_2 = \frac{\lambda^2}{2\mu^2} p_0$$

$$S_2 : 3\mu\, p_3 + \lambda\, p_1 - (\lambda + 2\mu)\, p_2 = 0 \Rightarrow p_3 = \frac{\lambda^3}{2*3*\mu^3} p_0$$

…

$$S_{k-1} : k\mu\, p_k + \lambda\, p_{k-2} - (\lambda + (k-1)\mu)\, p_{k-1} = 0 \Rightarrow p_k = \frac{\lambda^k}{k!\,\mu^k} p_0$$

$$\sum_{k=1}^{N} p_k(t) = 1$$

Многоканальная СМО с отказами

- Приведенная интенсивность заявок - среднее число поступивших заявок за среднее время обслуживания одной заявки.

  $\alpha = \lambda/\mu$

  Тогда, решая уравнения, получим формулы Эрланга:

$$p_0 = \left(1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \ldots + \frac{\alpha^n}{n!}\right)^{-1} = \left(\sum_{k=0}^{n} \frac{\alpha^k}{k!}\right)^{-1}$$

$$p_k = \frac{\alpha^k}{k!} p_0$$

Многоканальная СМО с отказами

Если все каналы заняты – заявка получит отказ. Поэтому вероятность отказа:

$$p_{отк} = p_n = \frac{\alpha^n}{n!} p_0$$

Вероятность обслуживания заявки:

$$Q = 1 - p_{отк} = 1 - \frac{\alpha^n}{n!} p_0$$

Абсолютная пропускная способность – среднее число заявок, обслуженных в единицу времени:

$$A = \lambda Q$$

Среднее число загруженных каналов:

$$N_{загр.} = p_1 + 2\,p_2 + \ldots + n\,p_n$$

Коэффициент загрузки одного канала:

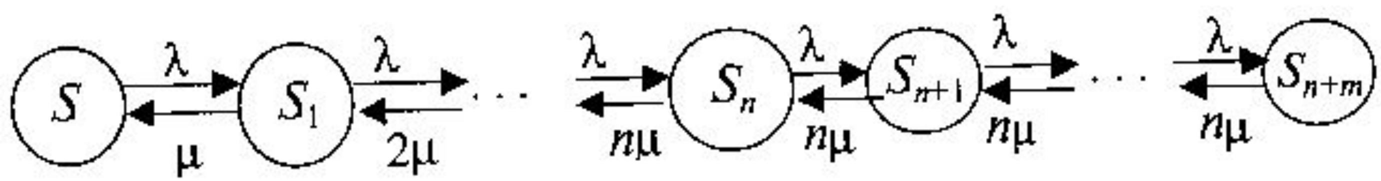
$$K_{загр.} = \frac{N_{загр.}}{n}$$

Многоканальная СМО с бесконечной очередью

Допущения:

- все каналы однотипны;
- время обслуживания заявки случайно и образует простейший поток интенсивностью μ;
- поступающий поток заявок интенсивностью λ будем считать простейшим;
- если все каналы заняты, то очередная заявка ставится в конец очереди;
- система может иметь бесконечное число состояний:
  - S_0 – все каналы свободны;
  - S_1 – занят один канал;
  - S_n – заняты все каналы;
  - S_{n+m} – заняты все каналы и m заявок в очереди;
  - и т.д.

Многоканальная СМО с бесконечной очередью



Преобразуя уравнения Колмогорова для конечных вероятностей аналогично СМО с отказами получим:

$$p_k = \frac{\alpha^k}{k!} p_0, k = 1,2,\ldots,n$$

$$p_k = \frac{\alpha^n}{n!} \left(\frac{\alpha}{n}\right)^{k-n} * p_0, k > n$$

$$p_0 = \left(1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \ldots + \frac{\alpha^n}{n!}\left(1 + \frac{\alpha}{n} + \frac{\alpha^2}{n^2} + \frac{\alpha^3}{n^3} + \ldots\right)\right)^{-1}$$

Многоканальная СМО с бесконечной очередью

Если α>=n, то длина очереди стремится к бесконечности.

$\alpha / n < 1:$

$$p_0 = (1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \ldots + \frac{\alpha^{n+1}}{n!(n-\alpha)})^{-1}$$

Среднее число загруженных каналов:

$$N_{загр.} = \alpha$$

Коэффициент загрузки одного канала:

$$K_{загр.} = \frac{\alpha}{n}$$

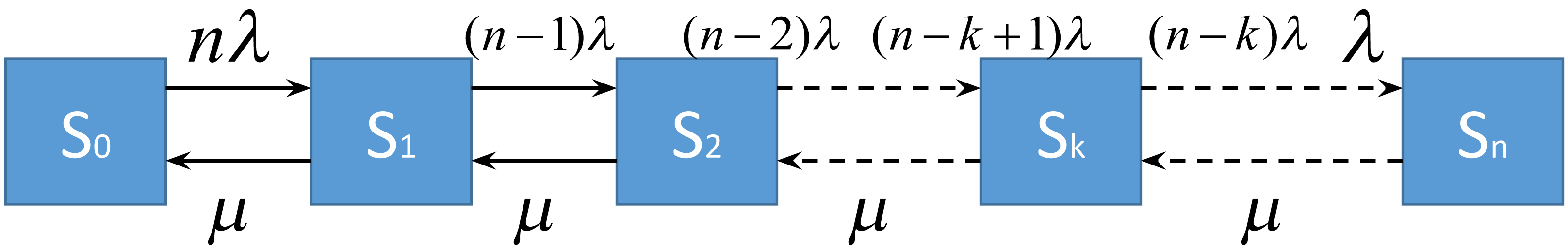
Среднее число заявок в очереди:

$$L_{оч} = \sum_{k=1}^{\infty} k\, p_{n+k} = \frac{\alpha^{n+1} p_0}{n * n!(1-\alpha / n)^2}$$

Среднее время ожидания в очереди:

$$W_{оч} = \frac{L_{оч}}{\lambda}$$

Замкнутая одноканальная СМО



$$p_1 = \frac{n\lambda}{\mu} p_0$$

$$p_2 = \frac{n(n-1)\lambda^2}{\mu^2} p_0$$

$$p_k = \frac{n(n-1)(n-2)*\ldots*(n-k+1)\lambda^n}{\mu^n} p_0 = \frac{\lambda^n n!}{\mu^n (n-k)!} p_0 = \alpha^n p_0 \frac{n!}{(n-k)!}$$

$$p_n = \frac{n(n-1)(n-2)\ldots*1*\lambda^n}{\mu^n} p_0 = \frac{\lambda^n n!}{\mu^n} p_0 = \alpha^n p_0 n!$$

Имитационное моделирование

Имитационное моделирование

Ситуации, в которых рекомендуется применять имитационное моделирование:

- не существует законченной постановки задачи исследования;
- аналитические методы являются математически трудоемкими и трудно реализуемыми;
- одной из задач является наблюдение за поведением отдельных составляющих системы в течение некоторого периода времени;
- процесс или явление невозможно наблюдать в реальной обстановке;

Имитационное моделирование

- необходимо управлять протеканием процессов в системе замедляя или ускоряя их в процессе моделирования (имитации);
- осуществляется подготовка специалистов и освоение новой техники;
- при проверке гипотез и изучении новых ситуаций в сложных системах, о которых мало что известно;
- основная задача моделирования – изучение последовательности событий, возникающих в проектируемой системе, поиск узких мест и оптимизация проекта.

Имитационное моделирование

- Суть имитационного подхода – процесс функционирования сложной системы представляют в виде алгоритма, реализуемого на компьютере.

- Как и при аналитическом подходе, при постановке задачи выделяются гипотезы о функционировании как всей системы, так и отдельных элементов.

- Имитационный подход позволяет максимально использовать всю известную информацию о системе.

Имитационное моделирование

Моделируется не только система (структура), но и время.

Обычно используют один из трех следующих представлений времени:

- **реальное время** моделируемой системы;
- **модельное время**, по которому обеспечивается синхронизация событий в системе;
- **машинное время**, соответствующее затратам ресурсов времени компьютера в процессе моделирования.

Имитационное моделирование

Построение имитационной модели (имитатора) можно разбить на этапы, похожие на этапы построения математической модели вообще, но с некоторыми отличиями.

Этапы построения имитационной модели:

- содержательное описание объекта моделирования;
- концептуальное описание объекта моделирования;
- формальное описание объекта моделирования;

Имитационное моделирование

- описание имитатора;
- программирование и отладка имитационной модели;
- испытание и исследование модели;
- применение модели;
- анализ результатов.

Имитационное моделирование

- **Содержательное описание объекта моделирования**

  Формулируются основные цели моделирования, определяется объект имитации, границы и ограничения модели, выбираются критерии эффективности для сравнения различных вариантов системы

Имитационное моделирование

- **Концептуальное описание объекта моделирования**

  Выдвигаются основные гипотезы и допущения относительно модели. Формулируется упрощенное алгоритмическое описание объекта моделирования. Проводится декомпозиция системы.

  Таким образом имеется:

- описание объекта моделирования;
- параметры и переменные моделирования;
- критерии эффективности;
- список методов обработки и способов представления результатов моделирования.

Имитационное моделирование

- **Формальное описание объекта моделирования**

Формальное представление поведения компонентов системы и их взаимодействия с помощью аппроксимации функциональными ззависимости, алгоритмического описания или смешанного подхода.

Производится предварительная проверка построенного формального описания на соответствие объекту.

Выбираются средства программирования и аппаратное обеспечение.

Имитационное моделирование

- **Описание имитатора**

  Перевод формального описания в описание имитатора. Решаются вопросы о моделировании времени и об основных моментах функционирования имитатора – начальные условия, условия остановки, сбор и сохранение логов и промежуточных данных, планирование.

Имитационное моделирование

- **Программирование и отладка имитационной модели**

  Техническое задание на ПО. Проектирование ПО, кодирование, тестирование и отладка.

  Создание технической документации – руководства, средства помощи и справки и т.д.

Имитационное моделирование

- **Испытание и исследование модели**

  Проверка правильности алгоритма в ходе работы программы-имитатора, проверка адекватности. При необходимости – калибровка модели.

  Оценка точности и устойчивости результатов, чувствительности критериев эффективности.

Имитационное моделирование

- **Применение модели**

    Составление плана экспериментов – максимум информации при минимальных временных и вычислительных затратах.

    Проведение экспериментов и сбор результатов.

- **Анализ результатов**

    Анализ результатов с целью поддержки принятия решений, выработки рекомендаций по проектированию или модификации системы, или другого использования полученной информации.

    Применение различных средств представление результатов – графики, диаграммы, средства компьютерной графики.

Имитационное моделирование

- Дискретно- событийное моделирование (discrete-event simulation (DES))
- Агентное моделирование (Agent-based modeling)
- Системная динамика (System dynamics)

Имитационное моделирование

- При построении имитационной модели необходимо использовать языки программирования. Часто, имитационные модели сложных технологических систем разрабатывают с использованием универсальных языков программирования (C, C++, Java, C# и т.д.).
- Альтернативой является использование методов основанных на специализированных языках и системах имитационного моделирования GPSS, SIMAN, SLAM, SIMSCRIPT, SIMULA, GASP.

Имитационное моделирование

Специализированные языки моделирования описывают поведение системы от события к событию, означающим начало или окончание технологической операции.

Основные преимущества использования специализированных языков компьютерной имитации:

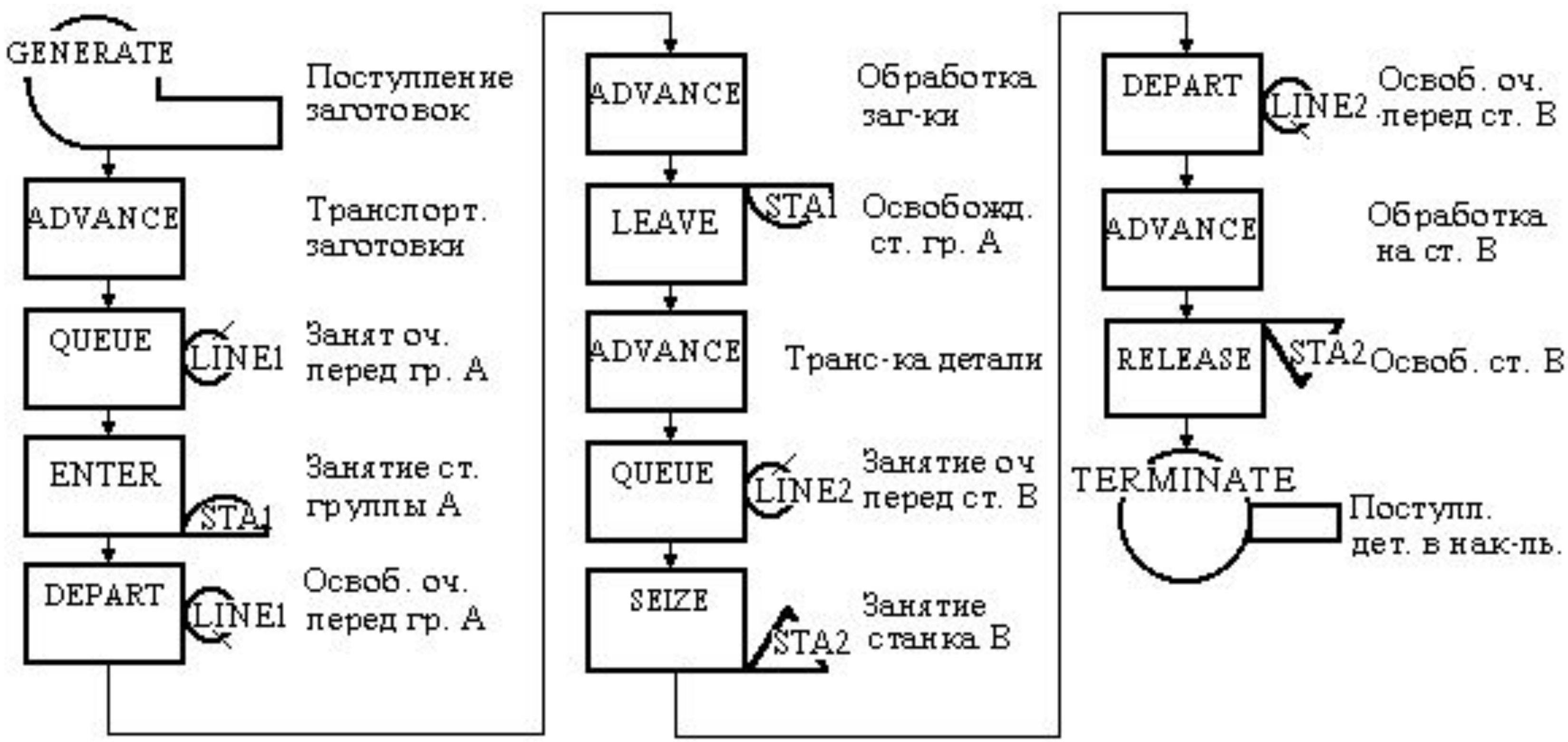
- упрощение и ускорение создания имитационных моделей;
- автоматическое формирование определенных типов данных, необходимых в процессе имитационного моделирования;
- без специального на то указания пользователя собирается множество статистических данных, описывающих поведение модели;
- возможность конструирования сложных имитационных моделей пользователям, не являющимися профессиональными программистами;

Имитационное моделирование

- программы имитационных моделей на специализированных языках моделирования близки к описаниям моделируемых систем на естественном языке (например, при моделировании СМО последовательность обработки заявки будет иметь вид: занять прибор (команда SEIZE), задержать на время обслуживания (команда ADVANCE), освободить прибор (команда RELEASE));
- направленность на моделирование систем без получения аналитических закономерностей процессов, так как техпроцесс отображается не системой уравнений, а взаимодействием отдельных динамических элементов во времени и пространстве.

СМО

- Многие специализированные системы и языки имитационного моделирования используют в качестве основы Системы Массового Обслуживания.



СМО

Пример имитатора одноканальной СМО с отказами.

Имеется источник заявок, который характеризуется:

- общим числом заявок за всё время наблюдения N_{max};
- числом заявок N(t), сгенирированных к моменту времени t;
- законом распределения интервалов времени Δt между появлением заявок;
- временем появления текущей заявки t_n.

СМО

Канал обработки характеризуется следующими параметрами:

- состоянием $S(t)$ в момент времени t – свободен или занят;
- числом N_w обслуженных заявок;
- суммарным временем T_w нахождения в занятом состоянии;
- производительностью Δw, т.е. законом распределения времен выполнения заявок определенного типа;
- временем W_n окончания обслуживания n-й заявки.

СМО

Параметры эффективности.

- Вероятность отказа можно оценить как частость отказов.

$$P_{отк}(t) = W_{отк}(t) = (N(t)-N_w(t))/N(t)$$

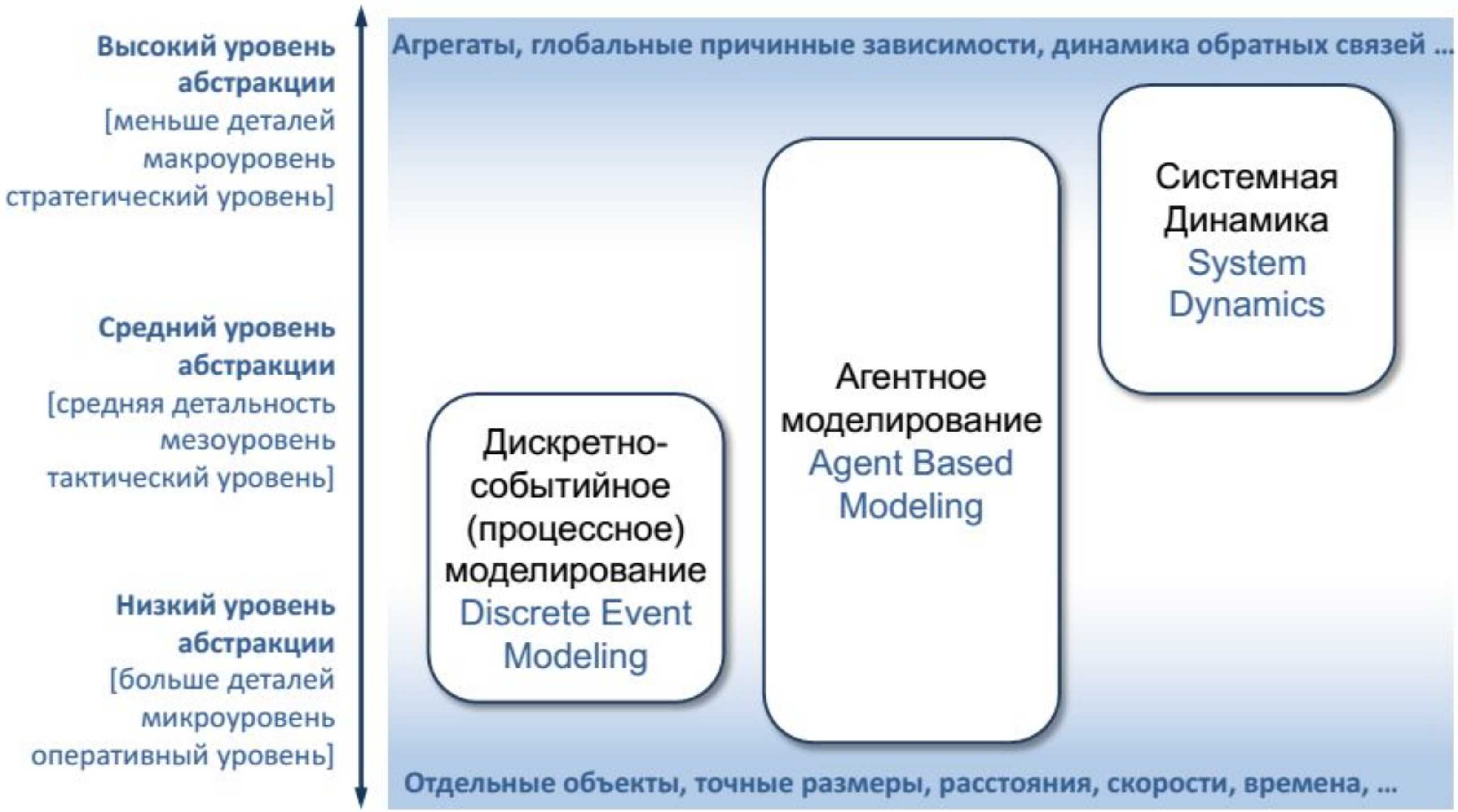
- Коэффициент загрузки канала:

$$K_з(t) = T_w(t)/t$$

- Задается число N_{max} заявок, которые должны быть сгенерированы и статистические параметры этих заявок, имитирующие случайный процесс поступления в реальной системе.
- Чем больше число заявок, тем больше точность оценок.
- Также задаются статистические характеристики процесса обработки заявок, имитирующие случайный процесс обработки в реальной системе.
- С увеличением времени наблюдения за системой, т.е. числа заявок, параметры эффективности будут стремится к предельным значениям.

Имитационное моделирование

Имитационное моделирование



Системная динамика

- Этот подход был разработан и предложен Джеем Форрестером в конце 1950х как “исследование информационных обратных связей в промышленной деятельности с целью показать как организационная структура, усиления (в политиках) и задержки (в принятии решений и действиях) взаимодействуют, влияя на успешность предприятия”.
- Приложения системной динамики также включают социальные, урбанистические, экологические системы.

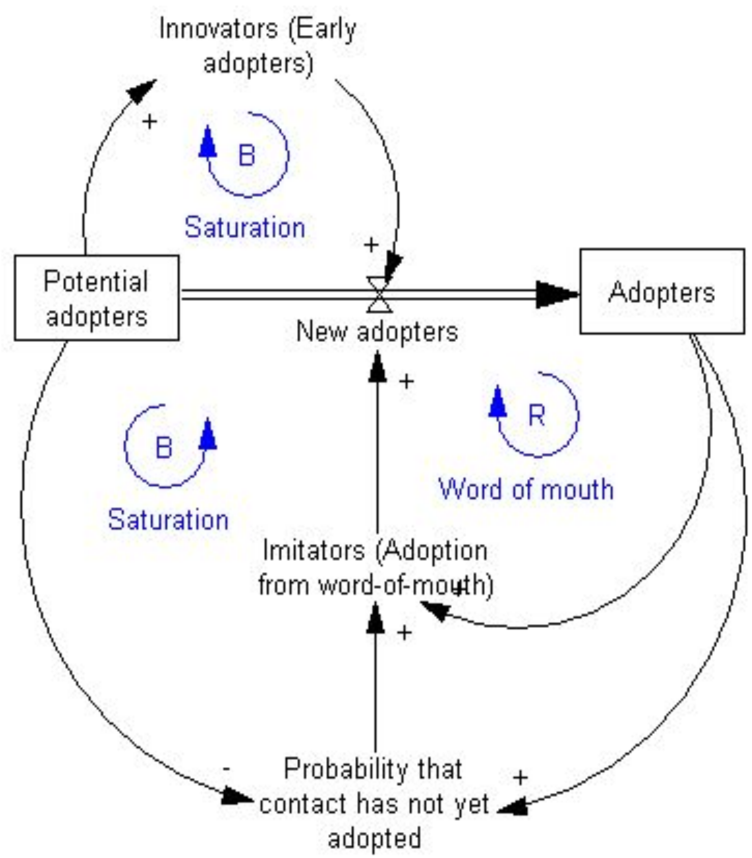
Системная динамика

Основными компонентами модели системной динамики являются:

- накопители (и их уровни), stocks, (например, материальных объектов, знаний, людей, денег);
- потоки, flows, характеризующие изменение накопителей;
- связи, через которые потоки и накопители влияют друг на друга;
- вспомогательные переменные, для задания влияний параметров.

Системная динамика

- СД абстрагируется от отдельных объектов и событий и предполагает “агрегатный” взгляд на процессы, концентрируясь на политиках, этими процессами управляющих.
- Моделирование с помощью методологии СД – это представление структуры и поведения системы как множества взаимодействующих положительных и отрицательных обратных связей и задержек.
- Причинные связи показывают как процесс А влияет на процесс Б. При этом процесс Б тоже, возможно, влияет на процесс А, но через длинную цепочку причинно-следственных связей.
- Только изучение динамики всей системы со всеми ее связями может привести к корректному пониманию процессов развития системы.



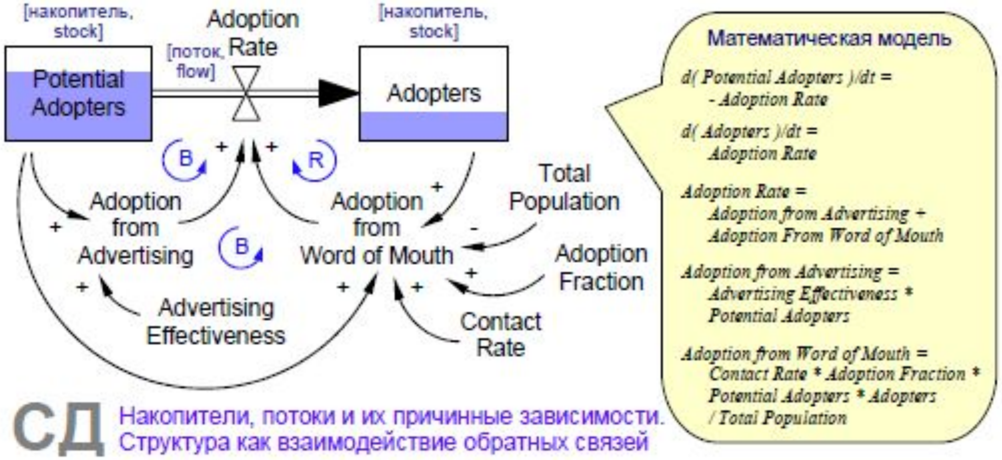
Системная динамика

Классическая модель распространения нового продукта Франка Басса:

- Потенциальные клиенты (*Potential Adopters)* становятся клиентами (*Adopters)* со скоростью распространения *(Adoption Rate),* которая зависит от рекламы и “устной рекламы”, т.е. общения клиентов с не-клиентами.
- Влияние рекламы моделируется следующим образом:
  - Некий постоянный процент потенциальных клиентов (*Potential Adopters * Advertising Effectiveness).*
  - Все контактируют со всеми. Количество контактов человека в единицу времени обозначается как *Contact Rate.*
  - Если клиент общался с потенциальным клиентом, последний становится клиентом с вероятностью *Adoption Fraction.*
  - Таким образом, в единицу времени все клиенты обратят

  *Adopters * Contact Rate * Adoption Fraction * [Potential Adopters / (Potential Adopters + Adopters)]* потенциальных клиентов в клиентов.

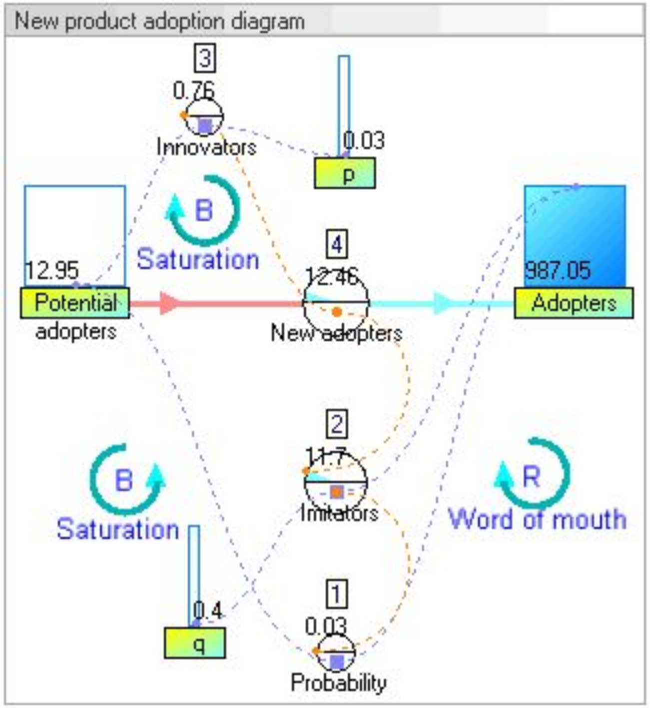
Системная динамика



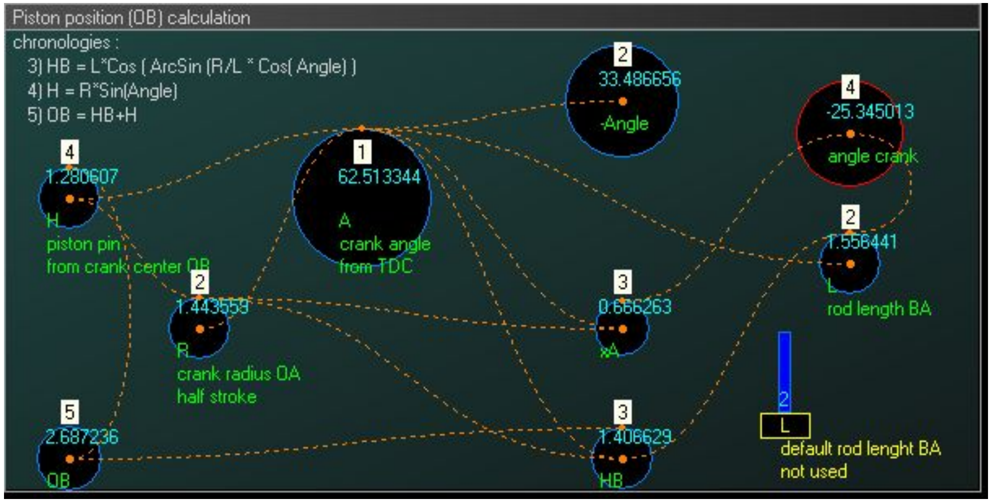
Модель распространения нового продукта Франка Басса

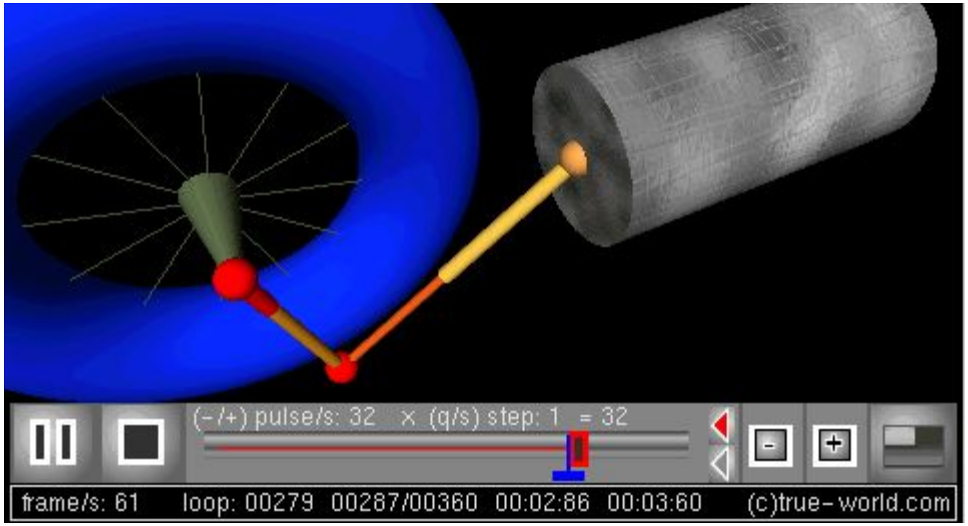
- `d(`Потенциальные клиенты`)/dt = -`Продажи
- `d(`Клиенты`)/dt =` Продажи
- Продажи = Продажи из-за рекламы + Продажи из-за устной рекламы
- Продажи из-за рекламы = Эффективность рекламы * Потенциальные клиенты
- Продажи из-за устной рекламы = Частота контактов * Эффективность устной рекламы * Потенциальные клиенты * Клиенты / (Потенциальные клиенты + Клиенты)

Системная динамика



Системная динамика





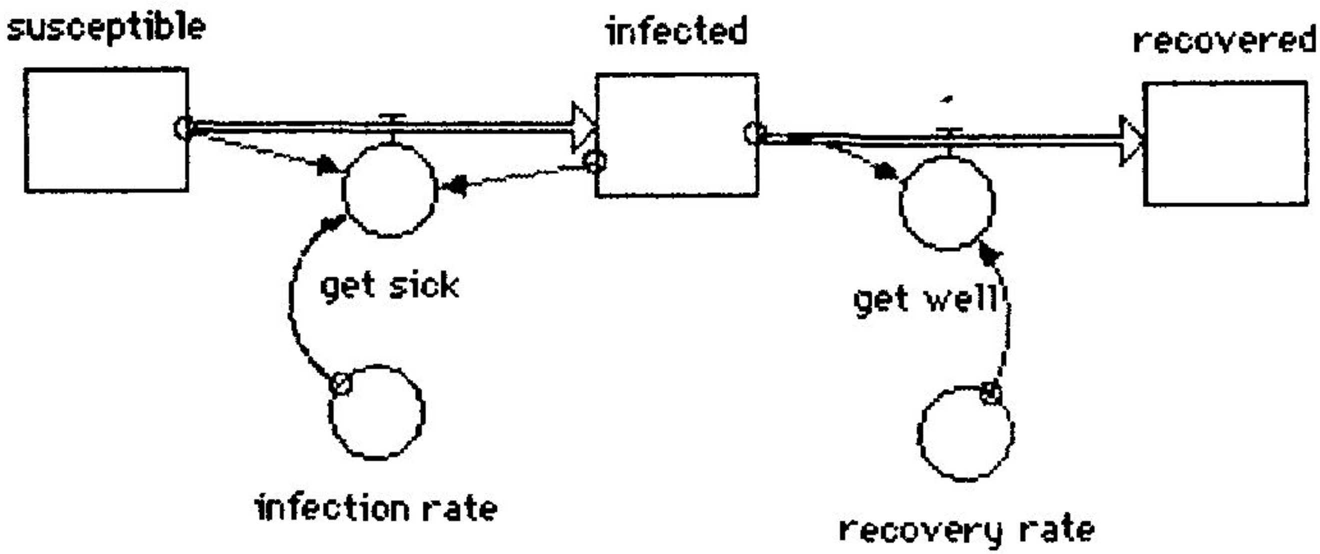
Системная динамика

Математически, системно-динамическая модель – это система дифференциальных уравнений.

Имитационное моделирование с использованием системной динамики можно представить как последовательность шагов:

- определить границ задачи (и системы);
- выявить основные накопители и потоки, которые меняют уровни этих накопителей;
- выявить источники информации, которые влияют на потоки;
- выявить основные обратные связи;
- нарисовать диаграмму, связывающую накопители, потоки и источники информации;
- записать уравнения, которые определяют потоки;
- оценить параметры и начальные условия (экспертной оценкой, статистической оценкой, используя внешние источники информации);
- провести моделирование.

Системная динамика




```
d(susceptible)/dt  = - get_sick;
d(infected)/dt  = get_sick - get_well;
d(recovered)/dt  = get_well;
get_sick = f(infected, susceptible, infection_rate);
get_well = f(infected, recovery_rate).
```

```
get_sick = infected * susceptible *infection_rate
get_well = infected * recovery_rate
```

```
susceptible(t+Δt) = susceptible(t) - get_sick * Δt
```

Агентное моделирование

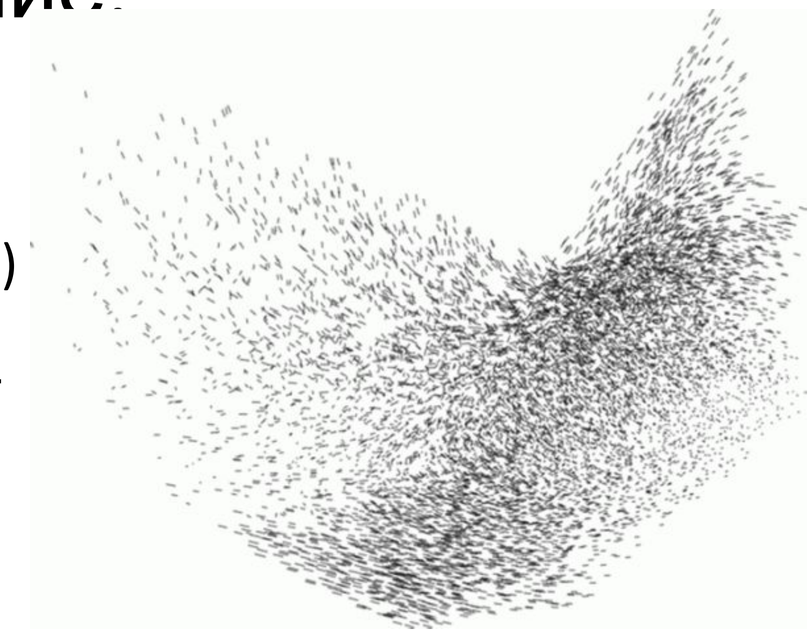
- Исследование поведения **децентрализованных** агентов, которое определяет поведение системы в целом. Моделирование снизу вверх.
- В основе модели лежит набор основных элементов, из взаимодействия которых рождается обобщенное поведение системы.
- «Возникающее» поведение (emergent behavior) представляет собой результат взаимодействия элементов системы. Соответственно, в рамках данного подхода к моделированию возникает необходимость корректно отобразить механизм поведения и взаимодействия элементов системы – т.н. «агентов».
- Агент представляет собой «некую сущность, которая обладает активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, может взаимодействовать с окружением и другими агентами, а также может изменяться»

Агентное моделирование

По одному из распространенных представлений агент должен обладать следующими характеристиками :

- **Агент является «идентифицируемым»**, т.е. представляет собой конечного индивидуума с набором определенных характеристик и правил, определяющих его поведение и правила принятия решений. Агент автономен и может независимо действовать и принимать решения по взаимодействию с другими агентами.
- **Агент находится в определенной среде, позволяющей ему взаимодействовать с другими агентами**. Агент может коммуницировать с другими (контактировать при определенных условиях и отвечать на контакт).
- **Агент имеет определенную цель**, влияющую на его поведение.
- **Агент гибок и обладает способностью самообучения** с течением времени на основе собственного опыта. В ряде случаев агент может даже изменять правила поведения на основе полученного опыта.

Агентное моделирование.



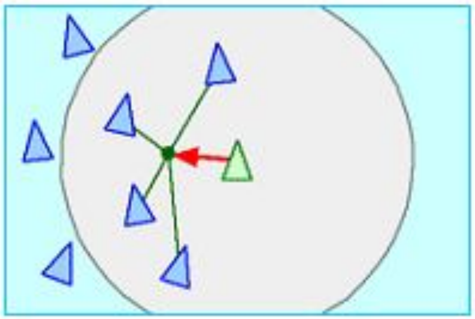
Допущения

- Поведение (его ключевые аспекты) агентов может быть описано.
- Механизмы взаимодействия могут быть описаны.
- Система может быть построена снизу вверх.

Примеры

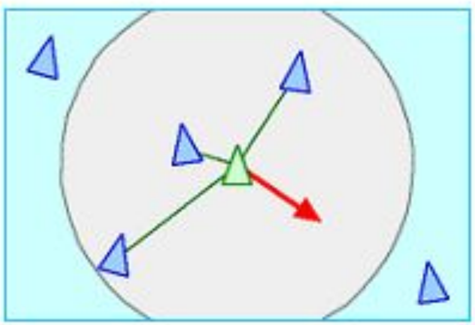
- Люди, группы, организации.
- Другие живые организмы с социальной структурой.
- Механизмы, роботы и их системы.

Агентное моделирование. Пример правил

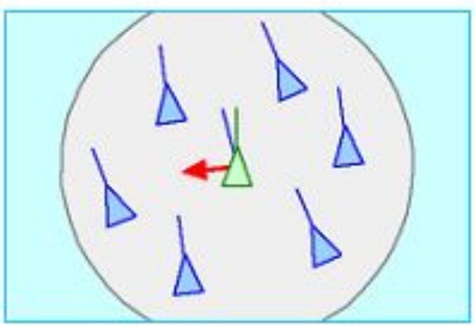


Простое моделирование стаи.

•Сплоченность – двигаться к средней позиции ближайших соседей.



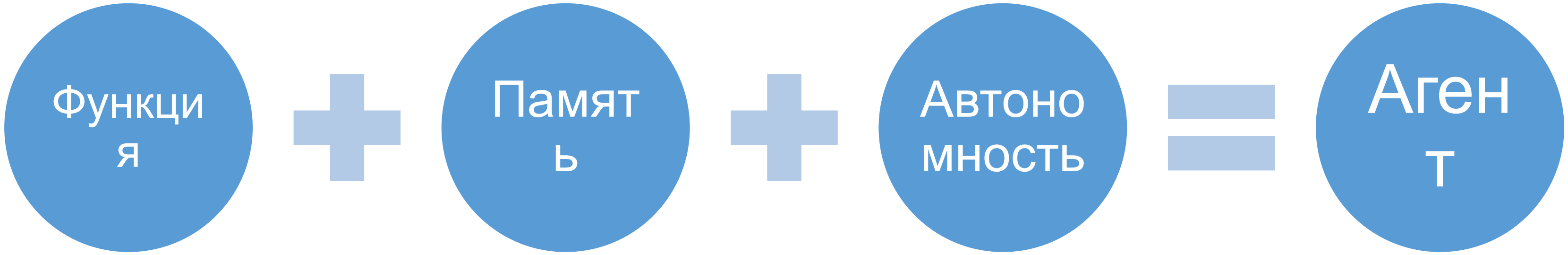
•Разделение – двигаться, чтобы избежать давки.



•Ориентация - двигаться в среднем направлении ближайших соседей

http://www.red3d.com/cwr/boids/

Агентное моделирование

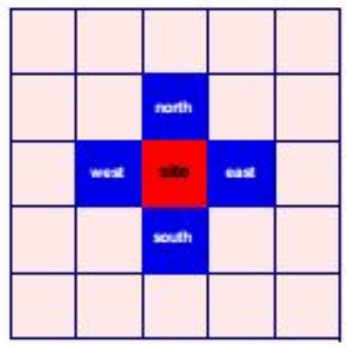


Агентное моделирование

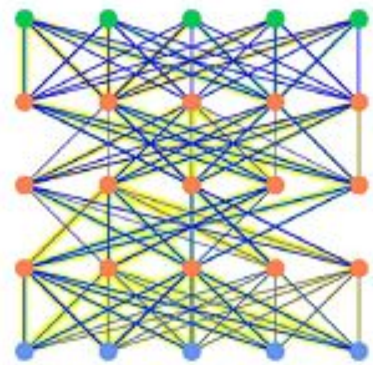
- Нет централизованного управления:
  - работой всей системы;
  - моделированием системы;
  - изменением состояний модели/системы.
- Агенты соединяются с соседними объектами по средствам топологии взаимодействия.
- Существует много различных видов топологий.
- Топологии могут быть статическими и динамическими.

Агентное моделирование. Топологии взаимодействия

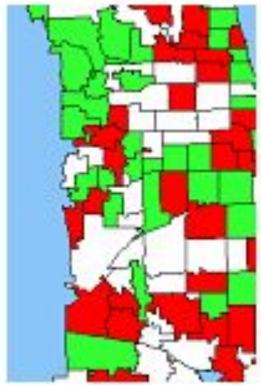
Евклидово пространство (2д, 3д)



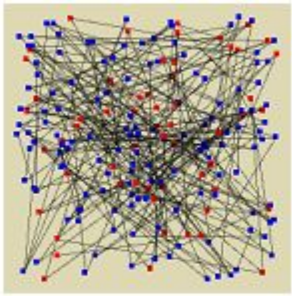
Дискретная сетка

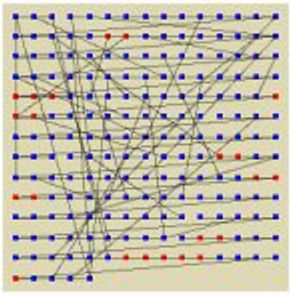


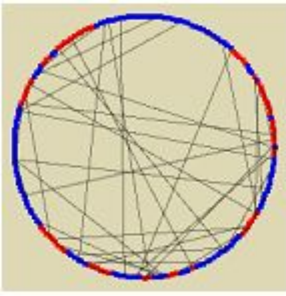
Сетевая структура

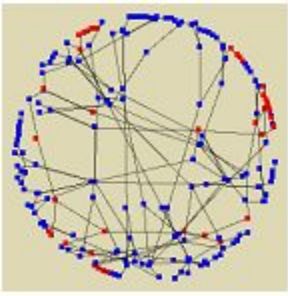


ГИС – геологическая информационная система



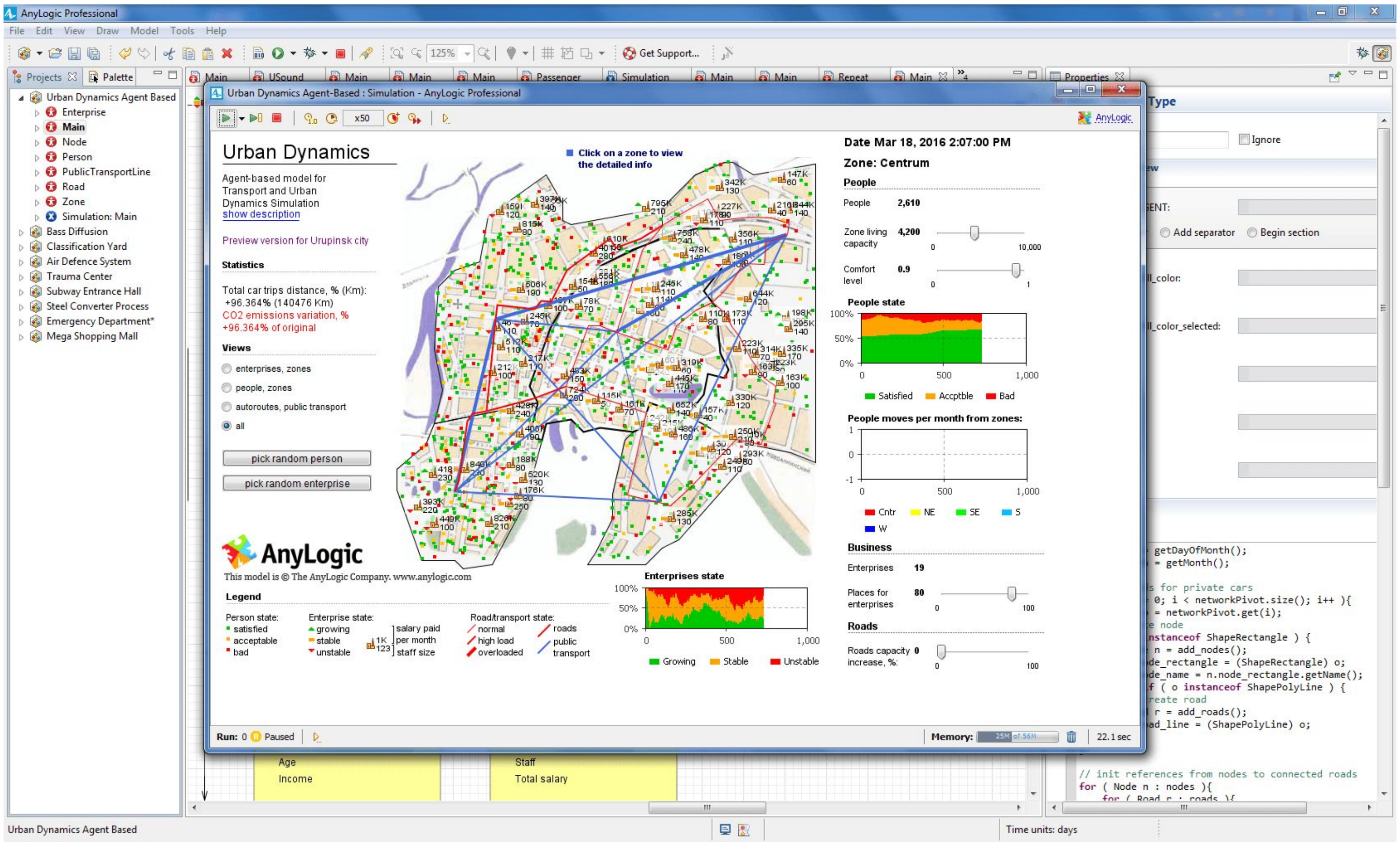






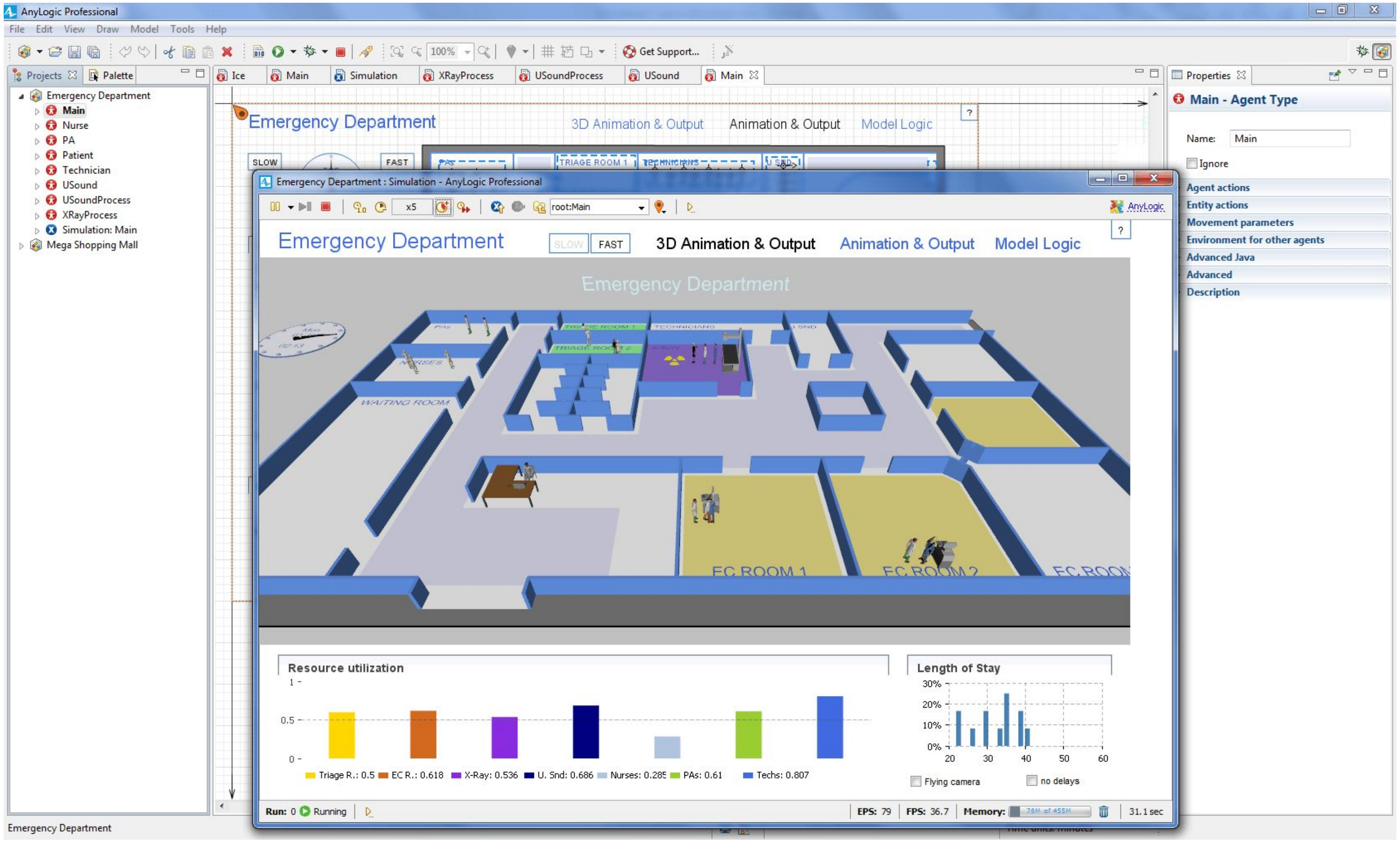
…

Агентное моделирование



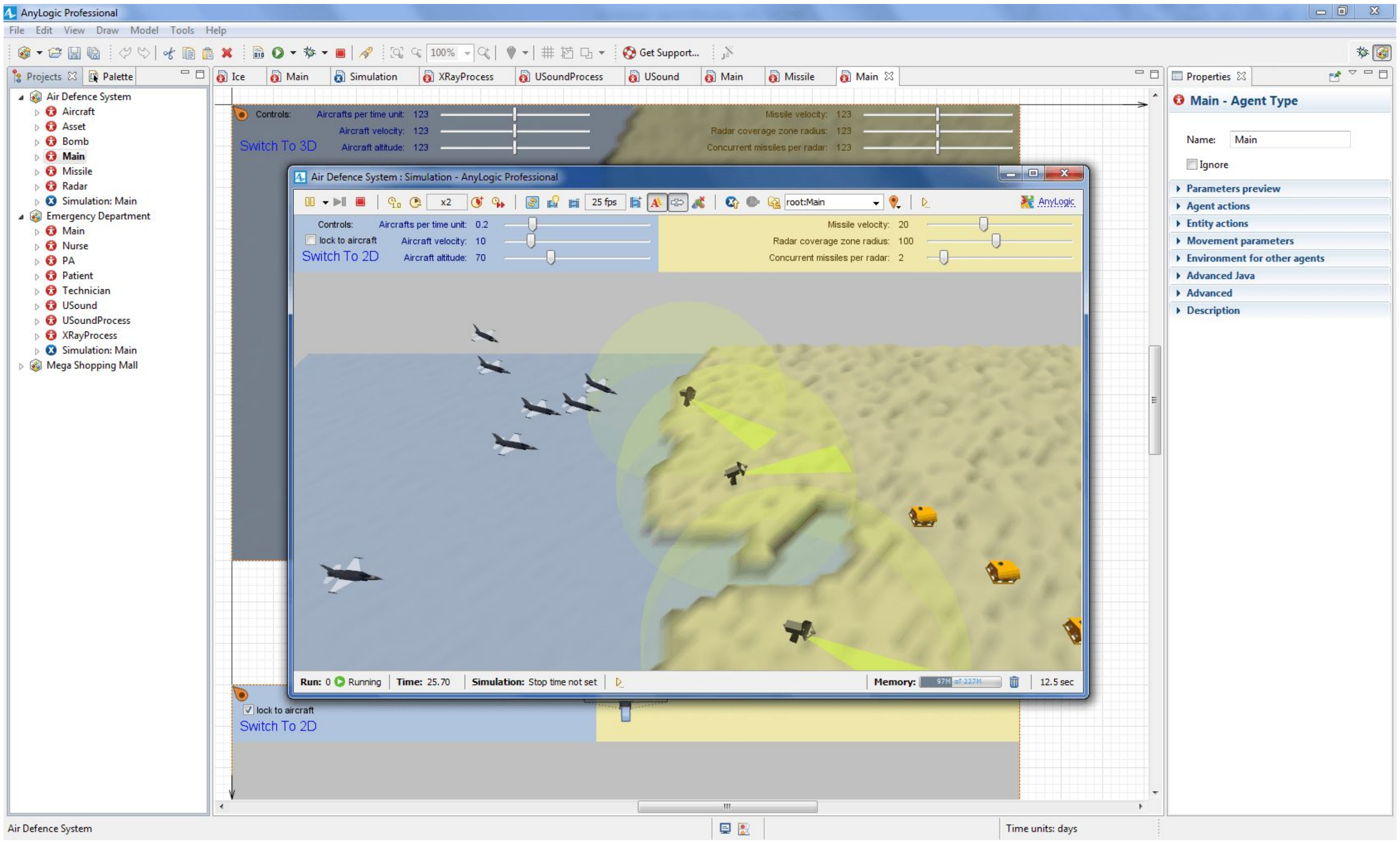
http://www.anylogic.ru/screentshots

Агентное моделирование



http://www.anylogic.ru/screentshots

Агентное моделирование



http://www.anylogic.ru/screentshots

Когда использовать агентное моделирование?

- Существует естественное представление компонентов системы как агентов:
  - есть решения и поведения, которые могут быть определены дискретно (с ограничениями);
  - адаптация и изменение поведения агентов являются важными;
  - необходимо обучение и динамическое стратегическое поведение агентов;
  - важны динамические связи между агентами, связи могут формироваться и исчезать;
  - образование структур из агентов, адаптация и обучение важны на уровне структур;
  - есть пространственная составляющая для поведения и взаимодействия агентов.
- Может потребоваться масштабирование вверх до произвольного уровня.
- Структурные изменения процесса – результат моделирования, а не входные данные.

Метод Монте-Карло

Метод Монте-Карло

$$\int \dots \int_{\Omega} g(x_1, x_2, \dots, x_n) dx_1$$

Интеграл заменяется математическим ожиданием функции g(ξ) случайной величины ξ, равномерно распределенной в произвольной области Ω n-мерного пространства, причем для приближенной оценки математического ожидания используется усреднение по достаточно большой выборке значений функции $g(\xi_k)$.

Метод Монте-Карло

Принципиальную основу метода Монте-Карло составляет усиленный закон чисел в форме Колмогорова:

Чтобы среднее арифметическое независимых реализаций случайной величины ξ сходилось с вероятностью единица к ее математическому ожиданию, необходимо и достаточно, чтобы это математическое ожидание существовало.

Метод Монте-Карло

- Усиленный закон больших чисел для зависимых случайных величин – например для стационарной Марковской цепи с последовательностью состояний $\Gamma_{i_1}, \Gamma_{i_2}, \ldots$ и матрицей переходов P также справедлив.
- Если $p = (p_1, p_2, \ldots, p_n)$ - единственное стационарное распределение стохастической матрицы P, то для любого начальной распределения p_0 и произвольной функции состояний $g(\Gamma)$ вероятность:

$$P\left(\lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} g(\Gamma_{ik}) = \sum_{j=1}^{n} p_j * g(\Gamma_j)\right) = 1$$

- Где $i_1 \rightarrow i_2 \rightarrow \ldots$ - последовательность переходов цепи Маркова, M – размер выборки.
- Случайные величины, связанные в цепь Маркова – реализации случайного процесса в дискретные моменты времени.

Метод Монте-Карло

- Выборка по значимости – «в более существенных частях области интегрирования – больше точек».

- Интеграл

$$J = \int \dots \int_{\Omega} g(x_1, x_2, \dots, x_n) f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

- При выполнении условий:

$$f(x_1, x_2, \dots, x_n) \geq 0$$

$$\int \dots \int_{\Omega} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n = 1$$

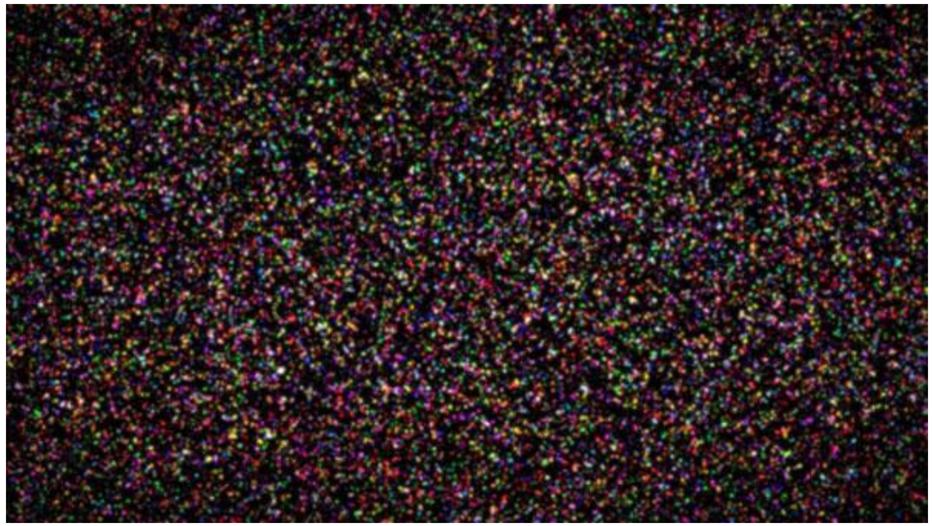
Метод Монте-Карло

- Оценкой интеграла может служить среднее арифметическое:

$$\langle J \rangle = \frac{1}{M} \sum_{k=1}^{M} g({x_1}^k, {x_2}^k, \dots, {x_n}^k),$$

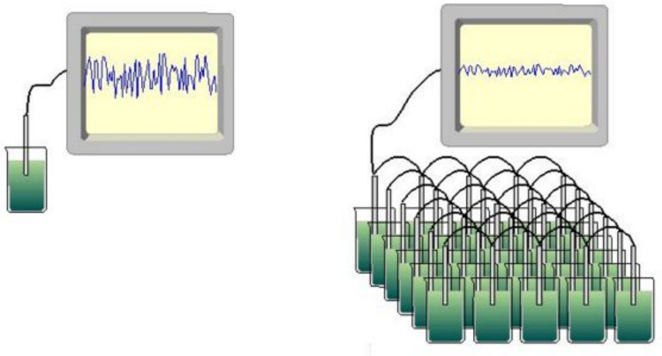
- $\{{x_1}^k, {x_2}^k, \dots, {x_n}^k\}$ – случайные реализации n-мерной точки, имеющей плотность распределения *f*, *M* – размер выборки, т.е. число этих реализаций.
- Если под $g(x_1, x_2, \dots, x_n)$ понимать значение некоторой физической величины в точке фазового пространства. А под функцией $f(x_1, x_2, \dots, x_n)$ - плотность вероятности найти систему в этой точке фазового пространства в соответствующем данной системе статистическом ансамбле, то значение интеграла *J* даст среднее значение наблюдаемой величины *g*.

Статистический ансамбль



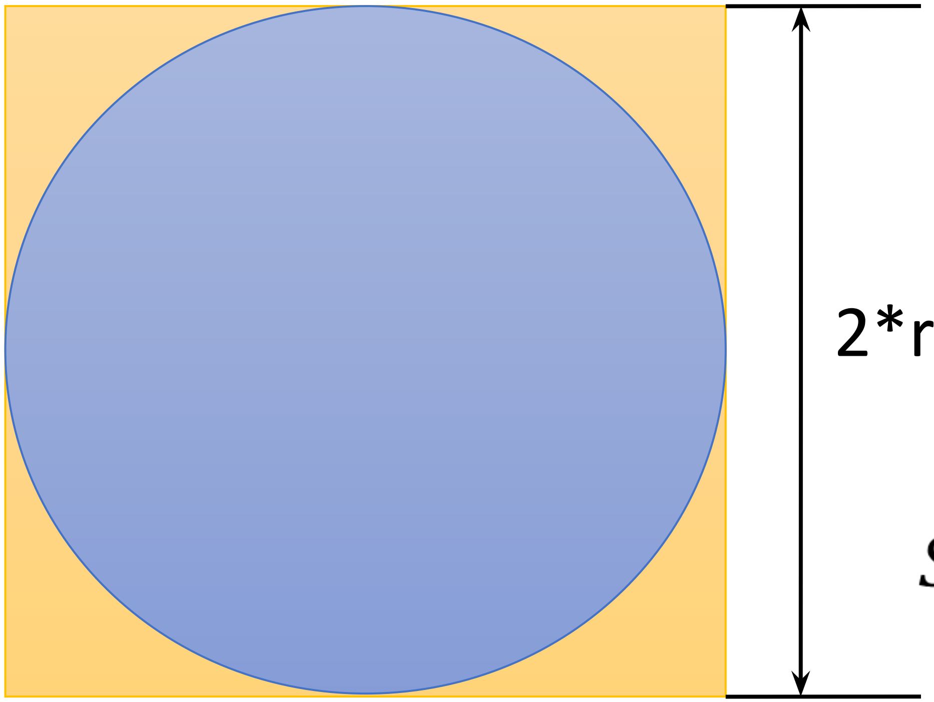
- Литр газа может содержать 10^{22} атомов.
- Чтобы полностью описать такую систему (или точнее её состояние) нужно знать три компоненты скорости и три компоненты положения в пространстве (координаты) для каждого атома. Т.е. $6 * 10^{22}$ вещественных чисел.
- Нас обычно интересуют средние значения свойств - давления, температуры, объема и др., которые не зависят от детального поведения каждого атома. Задача – оценить свойства системы, не зная всей информации о ней.

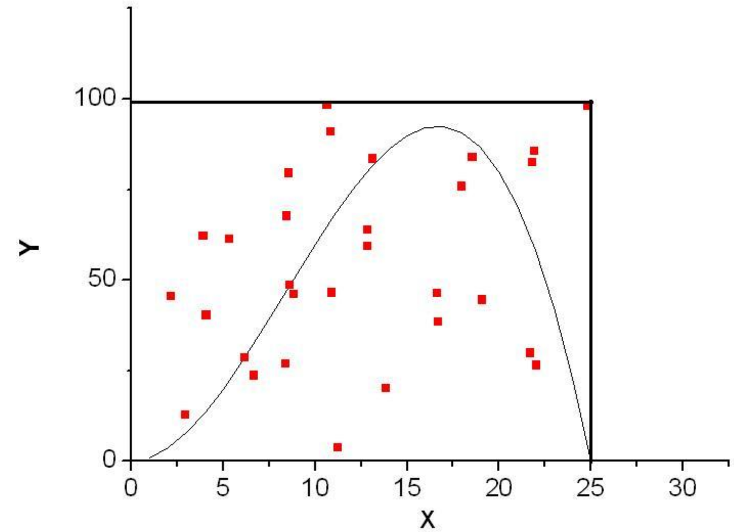
Статистический ансамбль



- Ансамбль – много копий (может даже бесконечно много) некоторой системы, каждая из которых представляет собой состояние, в котором система может находится.
- Т.е. по сути, статистический ансамбль – распределение вероятностей состояния системы.

Метод Монте-Карло





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

$$\frac{n_{\text{круг}}}{n_{\text{квадрат}}} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4} \quad \pi = 4\frac{n_{\text{круг}}}{n_{\text{квадрат}}}$$

Метод Монте-Карло

- Усреднение по ансамблю можно свести к усреднению вдоль марковской цепи.
- Задача состоит в организации такого случайного блуждания в конфигурационном пространстве, при котором выполняются 2 условия:
  - для полученной марковской цепи должно существовать стационарное распределение вероятности;
  - вероятность появления различных состояний должна совпадать с функцией плотности вероятности для данного ансамбля.

Метод Монте-Карло

- Для этого необходимо, чтобы для матрицы переходов P выполнялись 3 условия:
  - сумма вероятностей перехода из данного состояния в любое другое, достижимое за один шаг, равна 1 (стохастичность):

$$\sum_j p_{ij} = 1$$

  - вероятность любого состояния может быть получена как сумма произведений всех других состояний, из которых можно за один шаг перейти в данное, на вероятности одношаговых переходов (стационарность):

$$f_j = \sum_i f_i p_{ij}\,, f_k \geq 0 \quad \forall k, \sum_k f_k = 1$$

  - если i и j – два допустимых состояния, то существует конечное l – число шагов, такое что вероятность перехода из i в j за l шагов отлична от нуля:

$${p_{ij}}^l \neq 0$$

Метод Монте-Карло

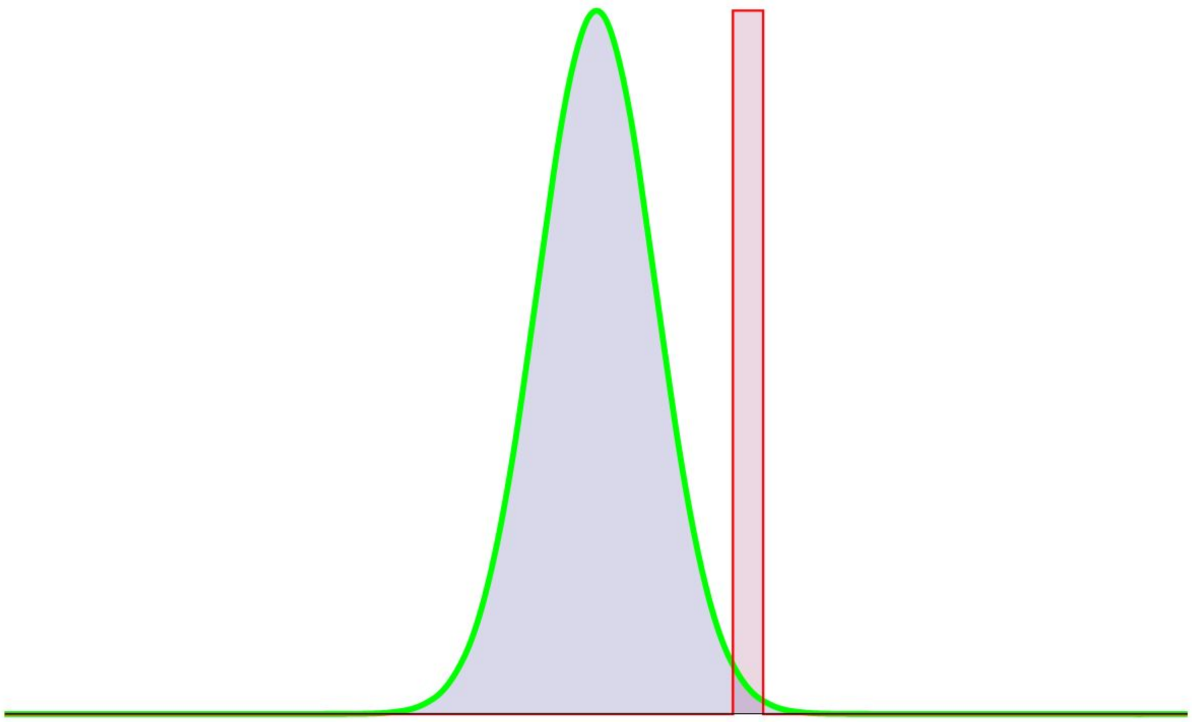
- На практике на матрицу переходов накладывают более сильное условие - *условие детального баланса или микроскопической обратимости.*

$$f_i p_{ij} = f_j p_{ji}$$

- Для двух любых состояний системы, между которыми возможен переход за один элементарный шаг алгоритма, произведения вероятности обнаружить систему в данном состоянии на вероятность перехода во второе состояние должны совпадать.

Метрополис-Гастингс

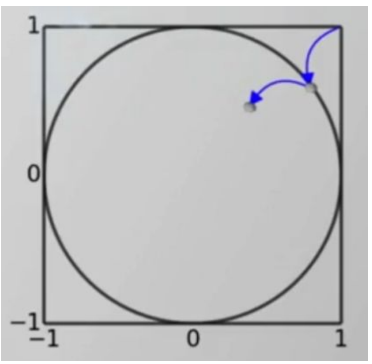
x_0

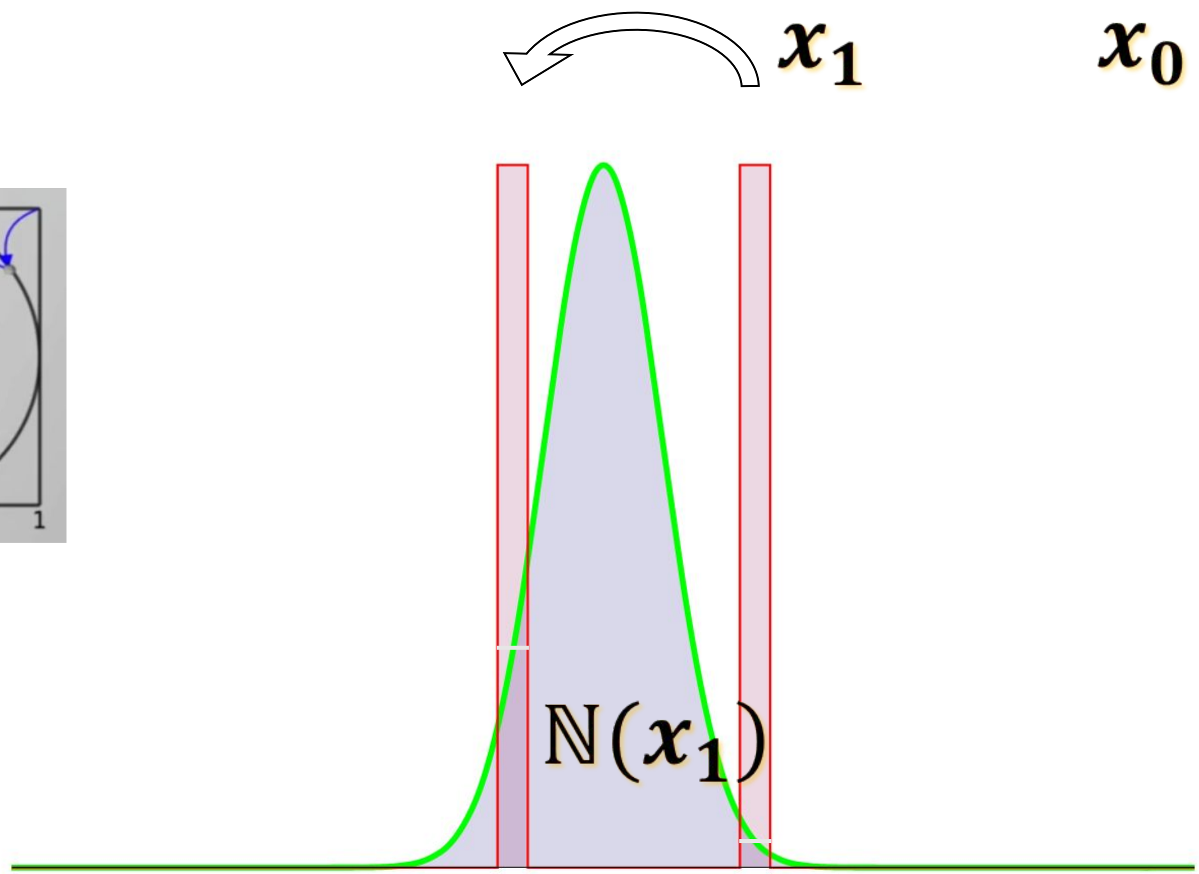


Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс

$$a_{x_0 \to x_1} = \frac{\mathbb{N}(x_1)}{\mathbb{N}(x_0)} > 1$$





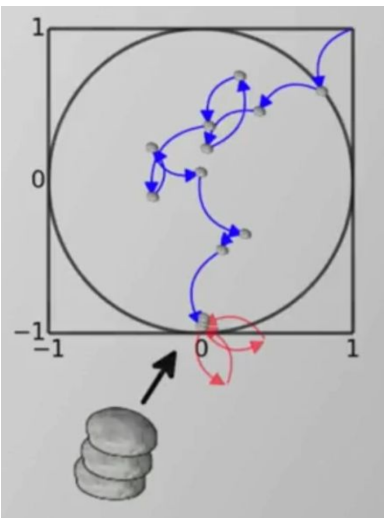
$\mathbb{N}(\boldsymbol{x_0})$

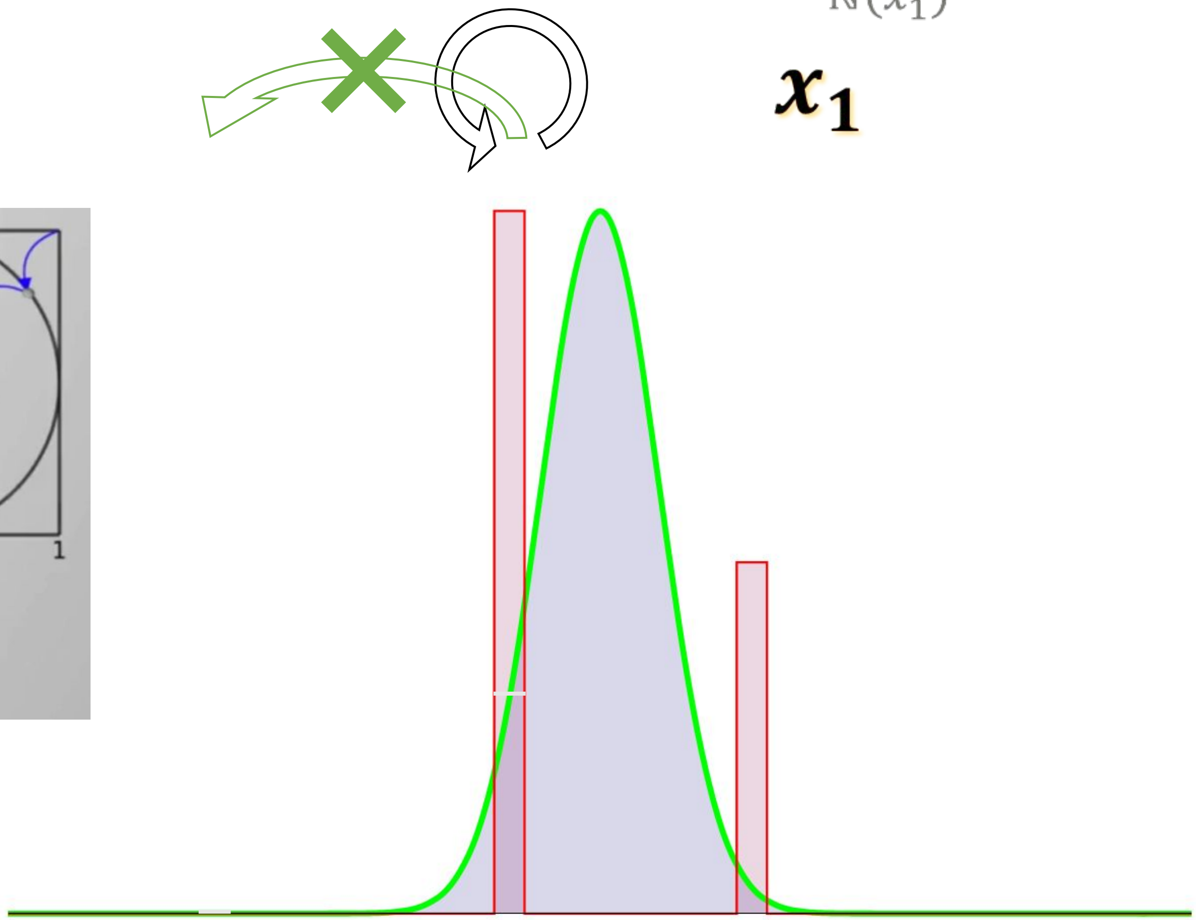
Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс

x_2'

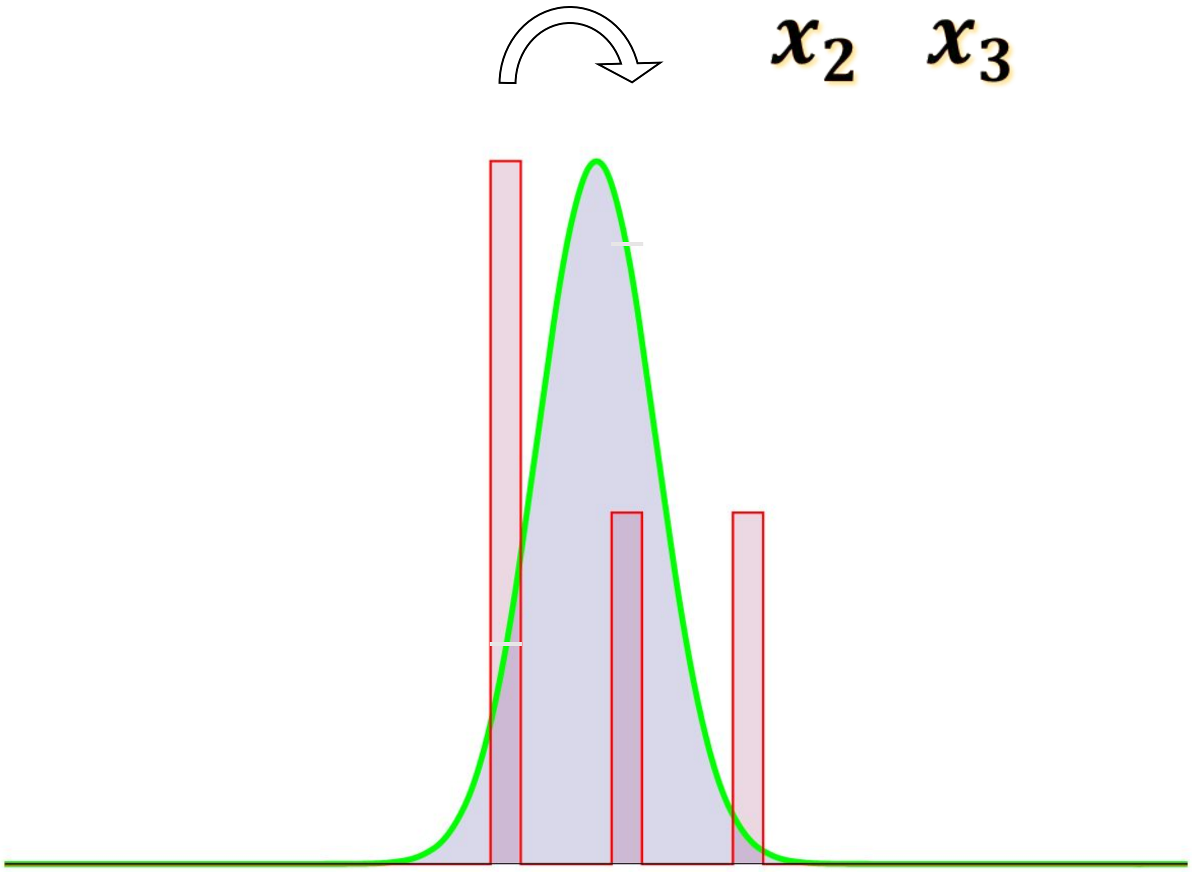
$$a_{x_1 \to x_2} = \frac{\mathbb{N}(x_2)}{\mathbb{N}(x_1)} \ll 1$$





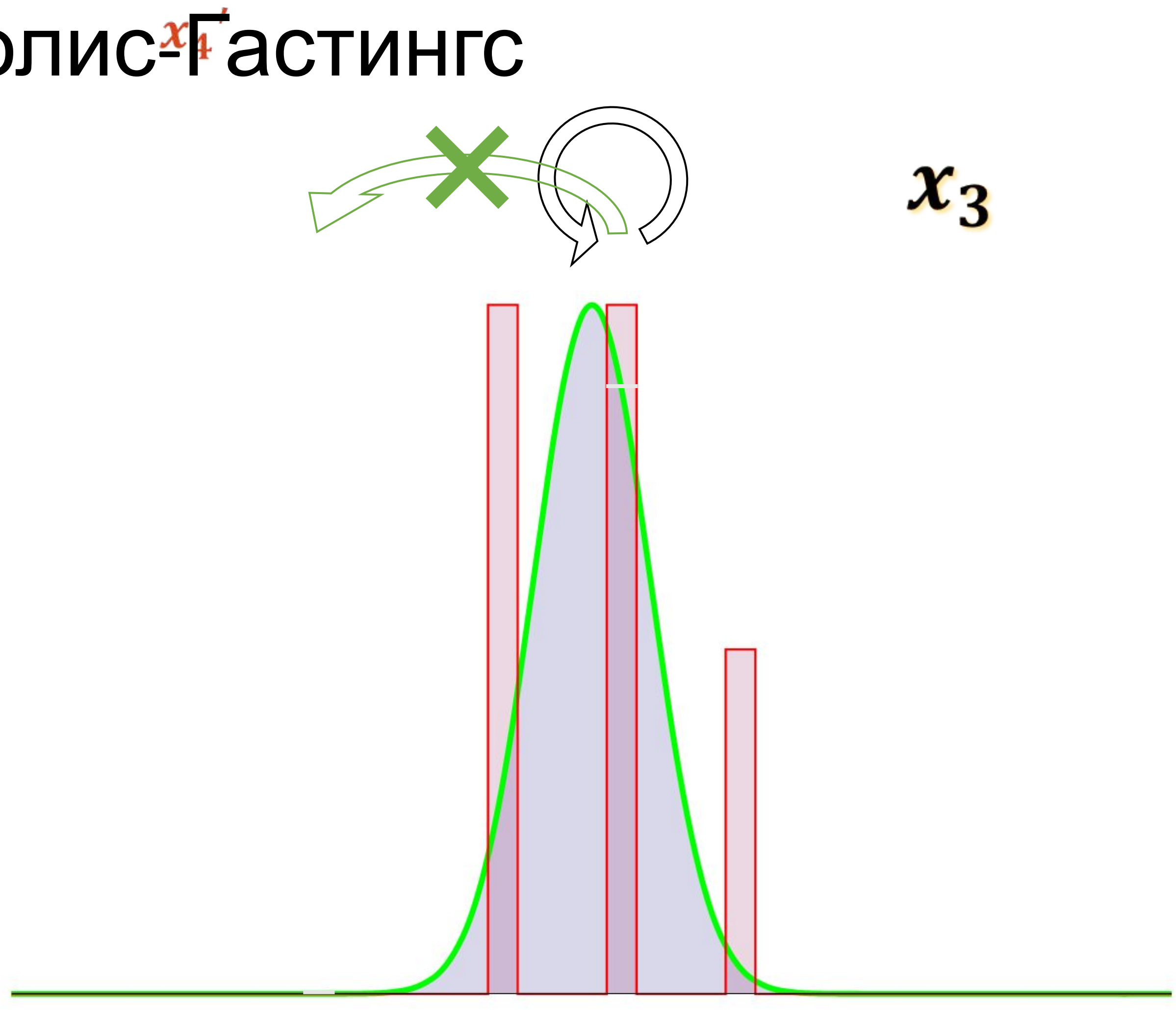
Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс



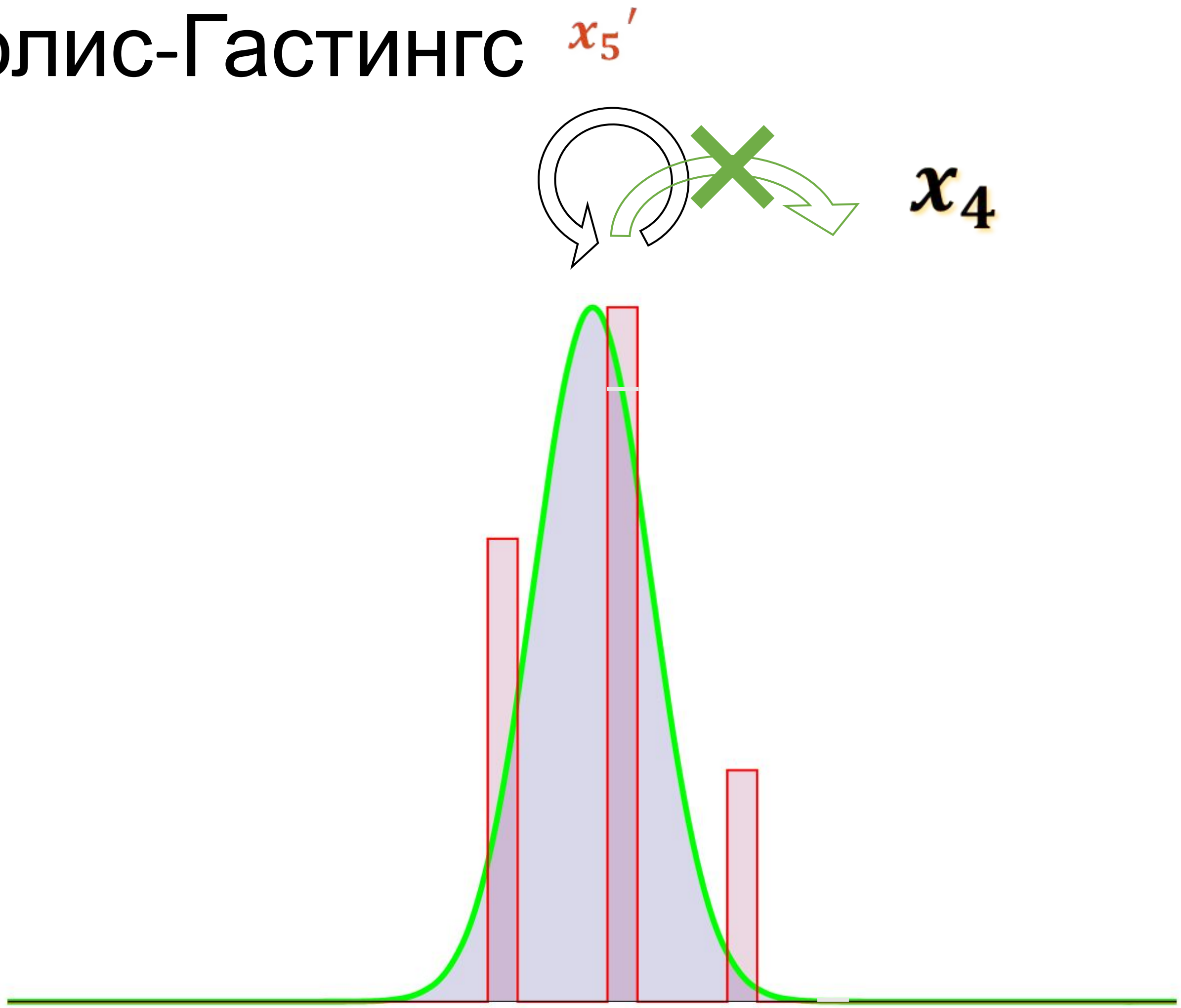
Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс



Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

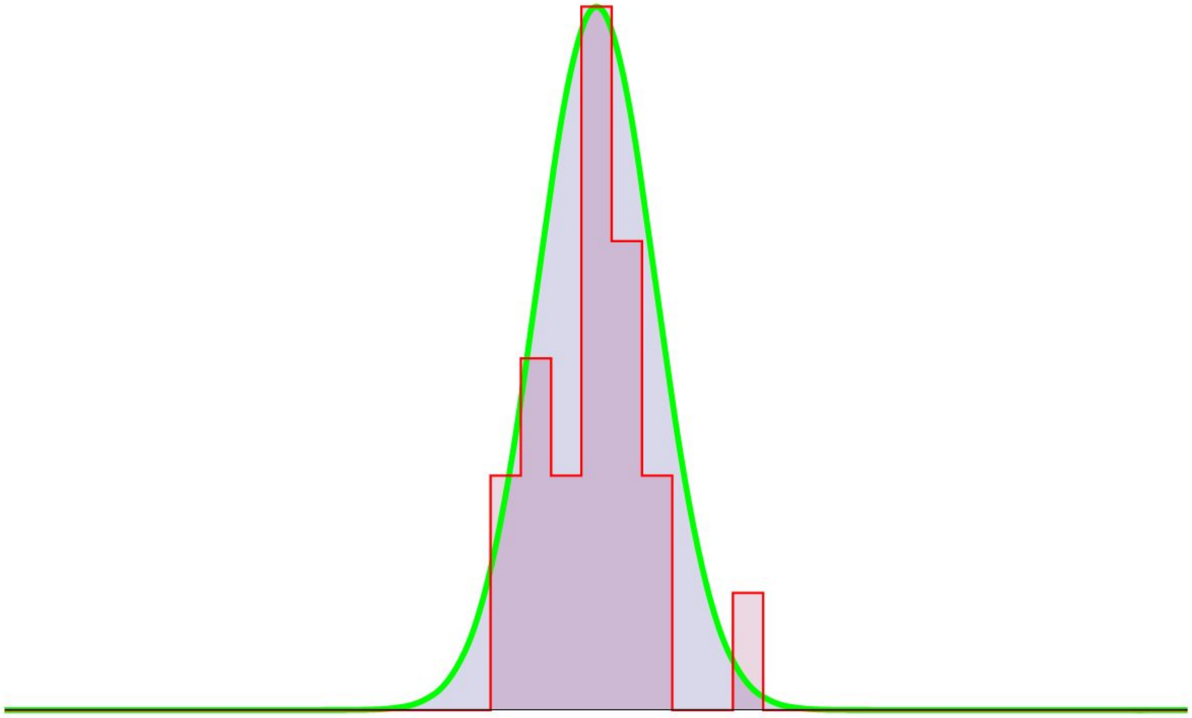
Метрополис-Гастингс



Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс

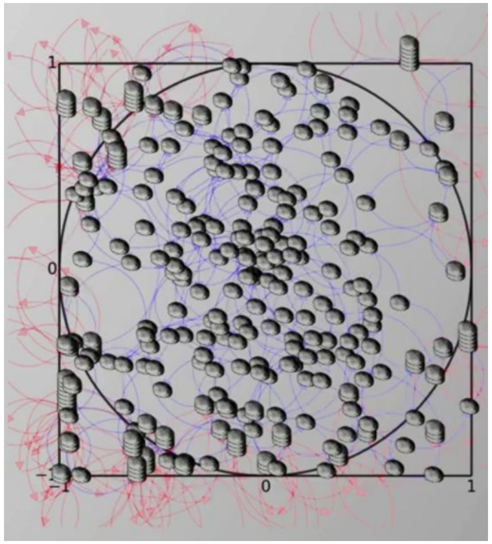
$$n = 20$$

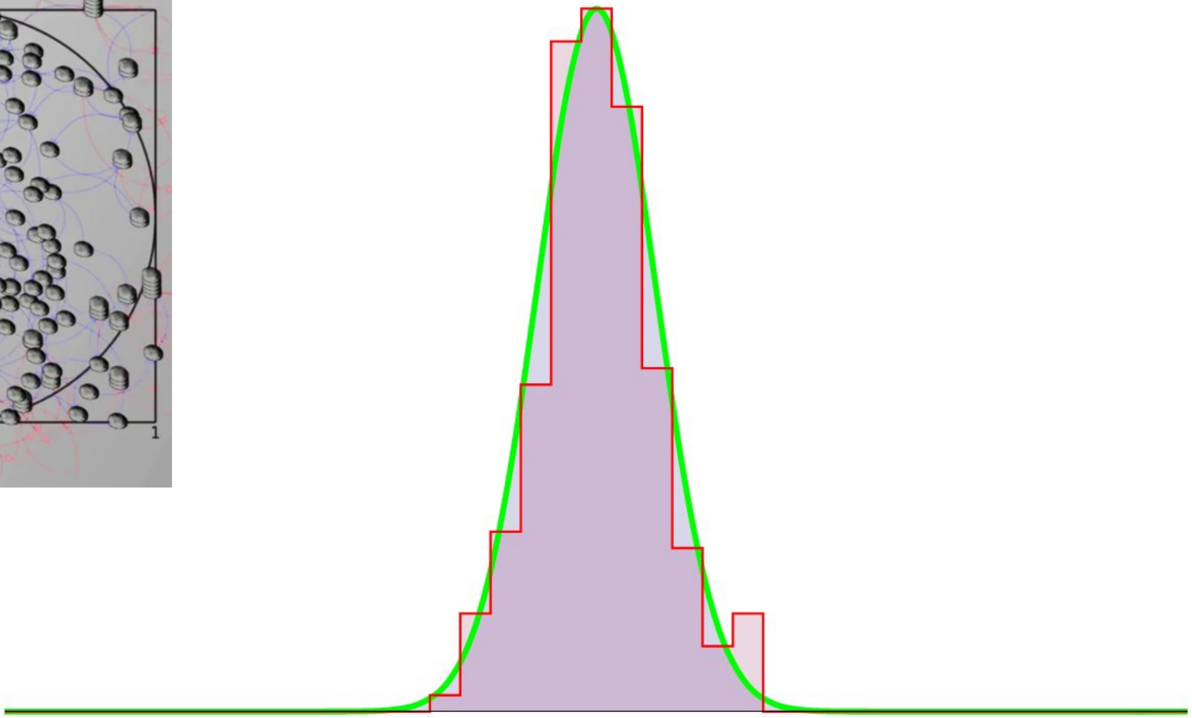


Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс

$$n = 200$$

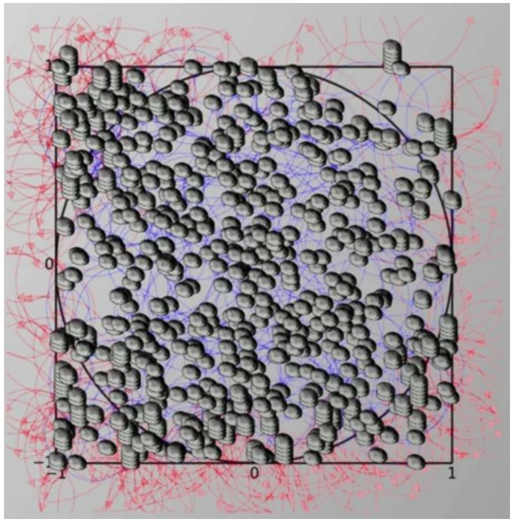


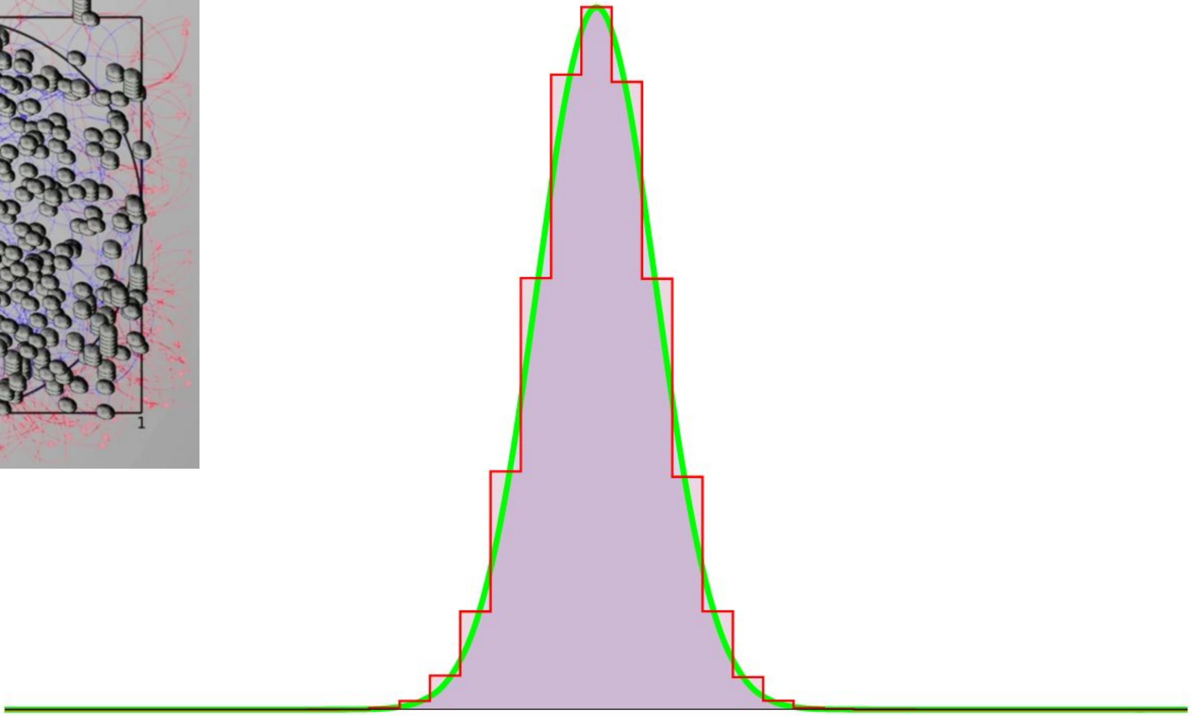


Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метрополис-Гастингс

$$n = 2000$$

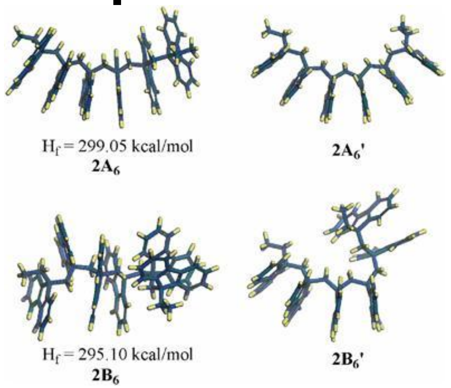


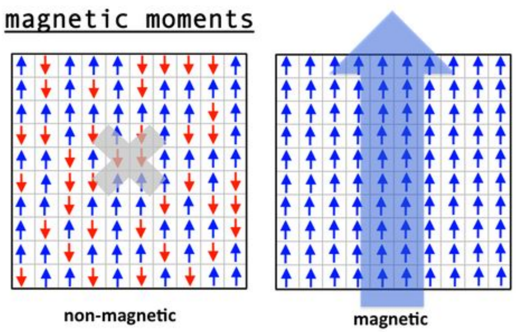


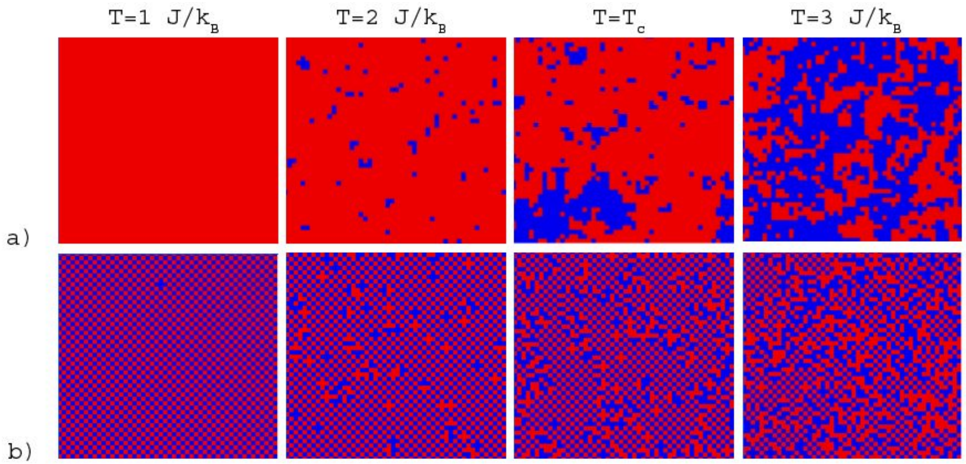
Slide by Anton S. Kaplanyan
Karlsruhe Institute of Technology

Метод Монте-Карло. Примеры

- Вычисление многомерных интегралов.
- Моделирование всех возможных конформаций цепных молекул (полимеров и др.)
- Получение полной фазовой диаграммы системы, моделирование фазовых переходов (классический пример – модель Изинга)







$$H = -J \sum_{\langle ij \rangle} S_i S_j - B \sum_i S_i$$

$$T_c = \frac{2J}{\ln 1 + \sqrt{2}} \approx 2.269J$$

$$P = \begin{cases} 1, \Delta E = \ E_{new} - E \leq 0 \\ e^{-\frac{\Delta E}{k_B T}}, \Delta E > 0 \end{cases}$$

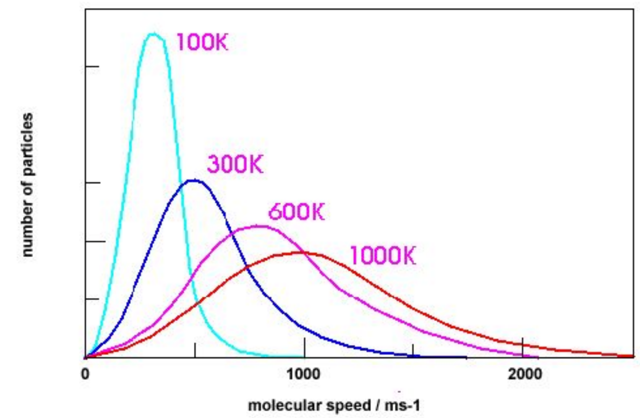
Метод Монте-Карло в естественных науках

- Квантовый метод Монте-Карло
  - Диффузный метод Монте-Карло (DMC)
  - Вариационный метод Монте-Карло (VMC)
- Метод Монте-Карло в статистической физике
  - Вычисление многомерных интегралов для вычисления макроскопических свойств системы. Например для некоторого свойства А:

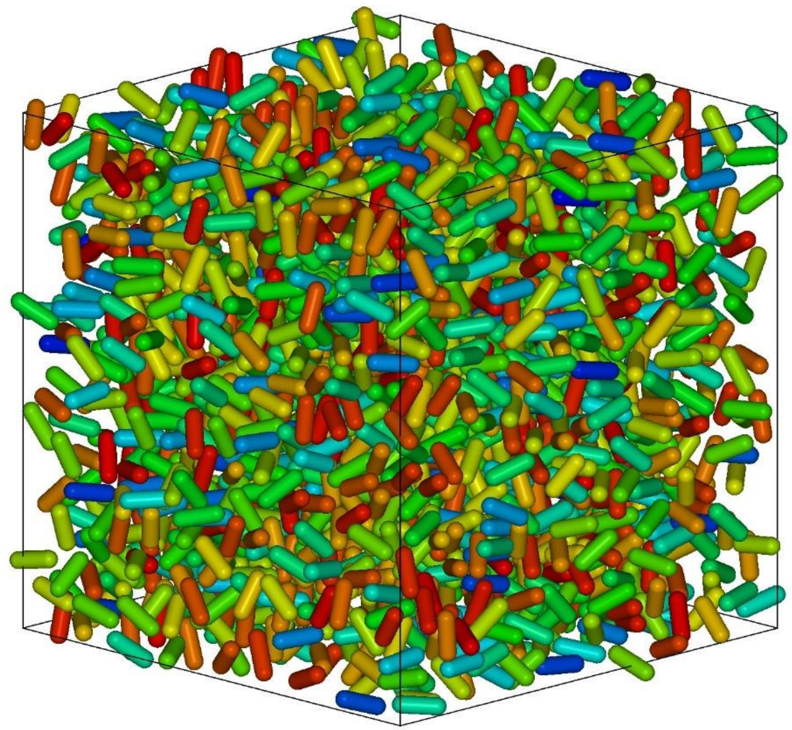
$$< A >= \int_{PS} A_{\vec{r}} \frac{e^{-\beta E_{\vec{r}}}}{Z} d\vec{r} \quad \beta \equiv 1/k_b T \quad Z = \int_{PS} P(\vec{r}) d\vec{r}$$

$\vec{r}$ - вектор всех степеней свободы системы

Метод Монте-Карло в статистической физике

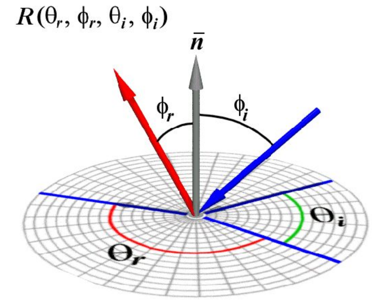


- Вместо того, чтобы моделировать динамическое поведение системы, производится генерация всех возможных состояний в соответствии со статистикой Больцмана.

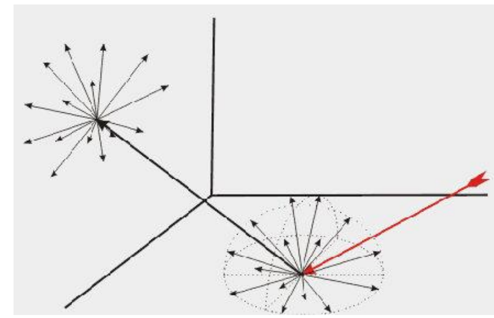


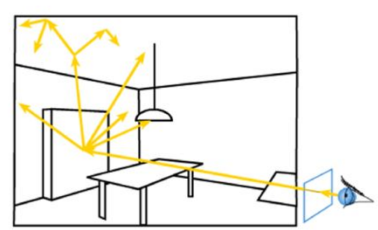
- Для определения нового состояни системы из текущего используется Марковская цепь.
- Каждый проход (итерация) метода переход в новое состояние.

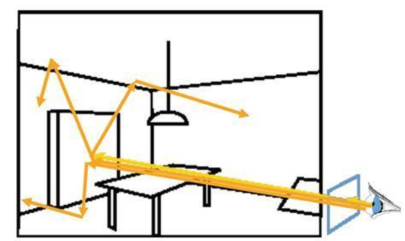
Метод Монте-Карло. Примеры



Метод Монте-Карло лежит в основе большинства алгоритмов фотореалистичного синтеза изображений в компьютерной графики.







- Трассировка пути и её вариации – метод Монте – Карло
- Перенос света Метрополиса и его вариации – алгоритм Метрополиса – Гастингса - Markov Chain Monte-Carlo

$$I(\varphi_r,\theta_r)=\iint_{\varphi_i\,\theta_i} L(\varphi_i,\theta_i)R(\varphi_i,\theta_i,\varphi_r,\theta_r)\cos(n,l_{\varphi_i,\theta_i})\,d\varphi_i\,d\theta_i$$