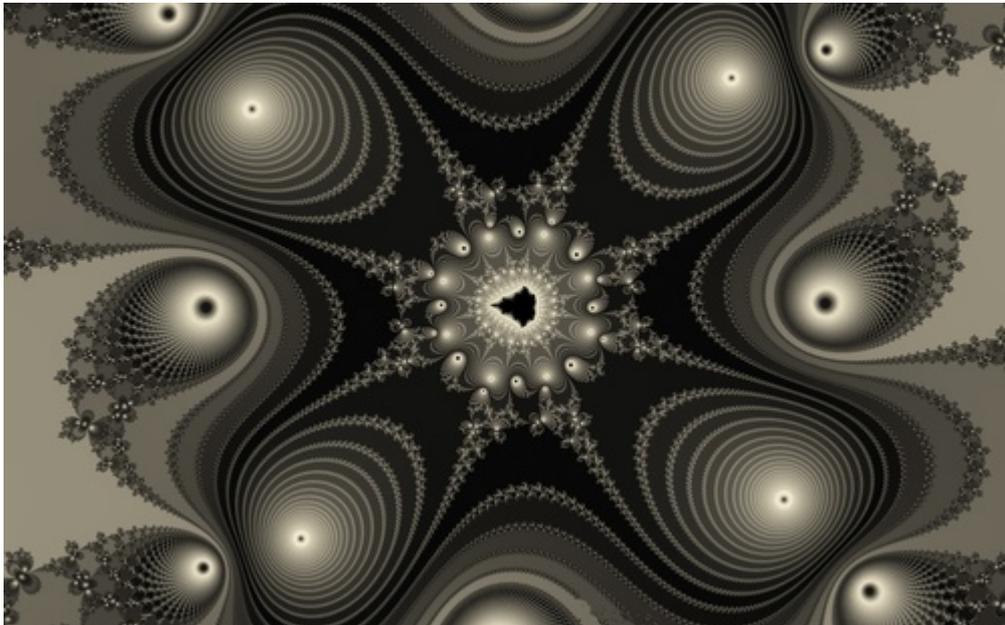


**Н.Л. Леонова**

# **Компьютерное моделирование**

**Курс лекций**



**Санкт-Петербург  
2015**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ  
РАСТИТЕЛЬНЫХ ПОЛИМЕРОВ»**

---

**Н.Л. Леонова**

**КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ**

**КУРС ЛЕКЦИЙ**

**Санкт-Петербург**

**2015**

УДК 681.3(075)  
Л476  
ББК 32.97я7

Леонова Н.Л. Компьютерное моделирование: курс лекций /СПбГТУРП. -  
СПб.,2015. - 88 с.

В работе дается курс лекций по дисциплине «Компьютерное моделирование». Рассмотрены основные понятия курса, этапы построения моделей и виды моделирования, предлагается информация по теории клеточных автоматов, информация о фракталах.

Курс лекций предназначен для студентов, обучающихся по направлению 01.03.02 «Прикладная математика и информатика», но также может быть полезен в процессе изучения дисциплин «Численные методы», «Вычислительная математика».

Рецензенты:

доцент кафедры «Информационно-измерительные системы и технологии»  
Санкт-Петербургского государственного электротехнического университета  
«ЛЭТИ» канд. техн. наук В.С. Коновалова;

доцент кафедры «Прикладная математика и информатика» СПбГТУРП,  
канд. техн. наук. П.Е. Антонюк

Рекомендовано к изданию Редакционно-издательским советом  
университета в качестве курса лекций.

---

Редактор и корректор Н.П. Новикова

Техн. редактор Л.Я.Титова

Темплан 2015 г., поз. 118

---

Подп. к печати 15.12.15. Формат 60x84/16. Бумага тип.№ 1.

Печать офсетная. 5,5 уч.-изд. л.; 5,5 усл. печ. л.

Тираж 50 экз. Изд.№118. Цена «С». Заказ

---

Ризограф Санкт-Петербургского государственного технологического  
университета растительных полимеров, 198095, СПб., ул. Ивана Черных, 4.

© Леонова Н.Л., 2015

© Санкт-Петербургский государственный  
технологический университет  
растительных полимеров, 2015

# ЛЕКЦИЯ 1

## ВВЕДЕНИЕ

### 1.1. Понятия модели и моделирования. Классификация моделей

Слово «модель» произошло от латинского слова «modulus», означает «мера», «образец». Его первоначальное значение было связано со строительным искусством, и почти во всех европейских языках оно употреблялось для обозначения образа или вещи, сходной в каком-то отношении с другой вещью. Например, перед строительством здания, сооружения делали его уменьшенную копию для обсуждения, улучшения, утверждения проекта.

Моделирование в научных исследованиях стало применяться еще в глубокой древности и постепенно захватывало все новые области научных знаний: техническое конструирование, строительство и архитектуру, астрономию, физику, химию, биологию и, наконец, общественные науки. Большие успехи и признание практически во всех отраслях современной науки принес методу моделирования XX век. Однако методология моделирования долгое время развивалась отдельными науками независимо друг от друга. Отсутствовала единая система понятий, единая терминология. Лишь постепенно стала осознаваться роль моделирования как универсального метода научного познания.

Термин «модель» широко используется в различных сферах человеческой деятельности и имеет множество смысловых значений. Мы будем рассматривать только такие модели, которые являются инструментами получения знаний.

**Модель** - это такой материальный или мысленно представляемый, т.е. информационный, объект, который в процессе исследования замещает объект-оригинал, обладая его существенными информационными свойствами (качественно-логическими и количественно-математическими), т.е.

характером отношений между элементами изучаемого объекта и его отношений к другим объектам физической реальности, так, что изучение модели дает новые знания об объекте-оригинале. Более строго, по сути модель представляет собой вид информационной системы, копирующей целевые системы (информационные, энергетические, вещественные), и предназначенной для изучения свойств последних. По форме модель может быть воплощена на любом физическом носителе: вещественном изделии, компьютерной программе.

**Моделирование** - процесс построения, изучения и применения моделей. Оно тесно связано с такими категориями, как абстракция, аналогия, гипотеза и др. Процесс моделирования обязательно включает и построение абстракций, и умозаключения по аналогии, и конструирование научных гипотез. Главная особенность моделирования в том, что это метод познания с помощью объектов-заместителей.

Модель выступает как своеобразный инструмент познания, который исследователь ставит между собой и объектом и с помощью которого изучает интересующий его объект. Именно эта особенность метода моделирования определяет специфические формы использования абстракций, аналогий, гипотез, других категорий и методов познания.

В самом общем случае при построении модели исследователь отбрасывает те характеристики, параметры объекта-оригинала, которые несущественны для изучения объекта. Выбор характеристик объекта-оригинала, которые при этом сохраняются и войдут в модель, определяется целями моделирования. Обычно такой процесс абстрагирования от несущественных параметров объекта называют формализацией. Более точно, **формализация** - это замена реального объекта или процесса его формальным описанием.

Основное требование, предъявляемое к моделям, – это их адекватность реальным процессам или объектам, которые замещает модель.

Практически во всех науках о природе, живой и неживой, об обществе построение и использование моделей является мощным орудием познания. Реальные объекты и процессы бывают столь многогранны и сложны, что лучшим (а иногда и единственным) способом их изучения часто является построение и исследование модели, отображающей лишь какую-то грань реальности и потому многократно более простой, чем эта реальность. Многовековой опыт развития науки доказал на практике плодотворность такого подхода. Более конкретно, необходимость использования метода моделирования определяется тем, что многие объекты (или проблемы, относящиеся к этим объектам) непосредственно исследовать или вовсе невозможно, или же это исследование требует слишком много времени и средств.

В моделировании есть два различных подхода. Это натурное и абстрактное моделирование.

**Натурная модель, физическая модель** - это модель-копия объекта, выполненная из другого материала, в другом масштабе, с отсутствием ряда деталей. Например, это игрушечный кораблик, домик из кубиков, модель самолета, используемая в авиаконструировании и др.

**Абстрактная модель, информационная модель** - это модель, отображающая реальность путем не вещественных, а информационных связей - словесным описанием в свободной форме, описанием, формализованным по каким-то правилам, математическими соотношениями и т.п.

## **1.2. Адекватность модели объекту моделирования**

Модель строится, в частности, для того, чтобы получить дополнительную информацию об объекте моделирования. Подразумевается при этом, что информация, полученная при исследовании модели, может быть с той или иной степенью достоверности перенесена на объект. Необходимое условие для перехода от исследования объекта к исследованию модели и дальнейшего

перенесения результатов на объект исследования — адекватность модели и объекта.

**Адекватность** предполагает воспроизведение моделью с необходимой полнотой всех характеристик объекта, существенных для цели моделирования.

Нельзя говорить об абсолютной адекватности, при которой модель по всем параметрам соответствует оригиналу, особенно если строится модель природных или социальных явлений и процессов (неконструктивных объектов). В этом случае оценка степени сходства может опираться в основном на оценку отличия от оригинала.

Понятие адекватности основывается на строгих в математическом отношении понятиях изоморфизма и гомоморфизма, но не совпадает с ними. Например, объект может быть изоморфен модели по структуре, в то время как целью моделирования является изучение его поведения.

Две системы называются **изоморфными**, если между ними существует **взаимно-однозначное соответствие**: каждому элементу одной системы соответствует элемент другой и наоборот.

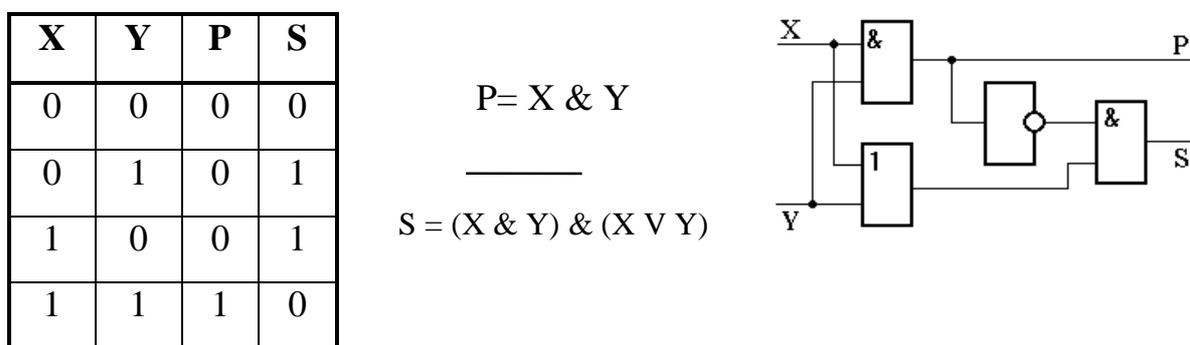
Две системы называются **гомоморфными**, если между ними существует **однозначное соответствие**: каждому элементу одной системы соответствует элемент другой, но не наоборот.

В общем случае обеспечение изоморфизма модели и объекта исследования по всем аспектам моделирования может быть не только трудновыполнимым, но и излишним, поскольку сложность модели при этом может оказаться настолько значительной, что не будет возможным никакое упрощение решаемой задачи.

### ПРИМЕР 1.1

Смоделировать работу такого элементарного устройства компьютера, как полусумматор двоичных кодов, можно с помощью таблицы истинности,

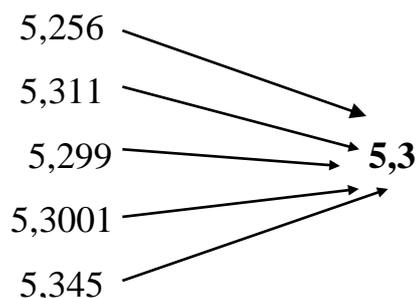
структурных формул или функциональной схемы (рис. 1.1). Все эти модели изоморфны по отношению друг к другу. Различаются лишь языки представления моделей.



*Рис. 1.1. Таблица истинности, структурные формулы и функциональная схема полусумматора*

### ПРИМЕР 1.2

При решении уравнения вы можете получить, например, значение корня, равное 5,256 или 5,311, или 5,299, или 5,3001, или 5,345. При округлении с точностью до одного знака после запятой все они преобразуются к числу 5,3. Восстановить по округленному значению, каким же было исходное число до округления, невозможно. Это пример гомоморфных моделей.



*Рис. 1.2. Округление чисел*

### **1.3. Адекватность конструктивных моделей**

Для установления адекватности в случае конструктивных, в том числе информационных моделей, необходимо сформулировать цель моделирования и уточнить, какой из аспектов изучаемого объекта: внешний вид, структура

или поведение представляет в данном случае интерес. В этом случае проблема адекватности сводится к установлению соответствующего изоморфизма или гомоморфизма.

### ПРИМЕР 1.3

Рассмотрим маятник, состоящий из тяжелого груза, подвешенного на конце нити. Известно, что моделью колебаний этого маятника может служить уравнение

$$x=A\sin(\omega t),$$

где  $x$  — отклонение от положения равновесия.

Адекватна ли эта модель поведению маятника?

Если посмотреть на колебания реального маятника, то можно заметить, что со временем размах колебаний становится все меньше и в конце концов маятник останавливается. Уравнение  $x=A\sin(\omega t)$  не предсказывает такого поведения, т.е. налицо явный неизоморфизм в поведении конструктивного объекта и его модели.

Тем не менее, если ввести следующие ограничения:

- отклонение  $x$  от положения равновесия мало (малые колебания);
- время  $t$  наблюдения за маятником мало,

то приведенное уравнение достаточно хорошо описывает поведение маятника (становится изоморфной), в чем можно убедиться непосредственным экспериментом.

Можно сказать, что при соблюдении вышеназванных условий уравнение  $x=A\sin(\omega t)$  адекватно описывает движение реального маятника.

### **1.4. Адекватность модели объекта в случае недоступности наблюдателю самого объекта**

Если наблюдателю доступны разные модели объекта, но недоступен сам объект, он может сравнить имеющиеся модели и выделить некоторые

"инвариантные" (присутствующие во всех моделях) моменты, которые с большой степенью достоверности можно отнести к самому объекту.

#### ПРИМЕР 1.4

Хорошо известная серия картин Клода Моне представляет собор в г. Руане в различные времена года и различное время суток ("Руанский собор в полдень", "Руанский собор в сумерках" и др.). Несмотря на существенное различие этих образов (моделей) можно сделать определенные достоверные выводы и о самом Руанском соборе.

### **1.5. Установление адекватности в случае существования единственной модели объекта**

Если наблюдателю доступна только одна модель, вопрос о ее адекватности объекту принимается на основе следующих фундаментальных научных положений:

- 1) **непротиворечивость:** невозможна одновременная истинность высказывания (А) и противоречащего ему высказывания (не А);
- 2) **закон достаточного основания:** «...ни одно явление не может оказаться истинным или действительным, ни одно утверждение справедливым, без достаточного основания, почему дело обстоит именно так, а не иначе...» (Г.В.Лейбниц);
- 3) **закон сохранения энергии:** энергия поля + энергия объекта=const;
- 4) **закон сохранения вещества:** вещество никуда не исчезает и ниоткуда не возникает;
- 5) **свойство симметрии:** если какое-либо состояние или процесс встречается в природе, то для него существует обращенное во времени состояние или процесс, который также может реализоваться в природе.

Кроме того, адекватность модели оценивается на основе общих эвристических принципов:

- 1) **принцип простоты**. Был известен еще в древние века. В явном виде его сформулировал философ XVI в. Оккам: “Не плоды рассуждений больше сущности”;
- 2) **принцип эстетики (“правило красоты”)**: из двух во всем остальном одинаковых моделей надо выбирать более красивую;
- 3) **принцип соответствия**: если корректно уточнить адекватную модель или область действия адекватной модели, то в результате получится адекватная модель (впервые отчетливо сформулирован великим датским физиком Н.Бором).

Приведем несколько примеров.

### ПРИМЕР 1.5

Как известно, в классической физике понятие массы является мерой инертности тела, а также мерой его гравитационного взаимодействия.

Можно рассмотреть две теории, в которых:

- понятия инертной и гравитационной массы в принципе могут численно различаться, поскольку это разные понятия;
- выполняется принцип эквивалентности этих масс, предложенный А.Эйнштейном в 1910 г.

Какая из этих теорий более адекватна?

Разумеется, у Эйнштейна были различные аргументы для введения этого принципа. Но принцип простоты подразумевает, что в реальности все должно быть устроено как можно проще. При этом простота вовсе не подразумевает упрощенности.

### ПРИМЕР 1.6

Периодический закон Менделеева в своей первой формулировке утверждал, что свойства химических элементов являются периодической функцией их атомных весов. Ради такой простой формулировки

Д.И.Менделеев пренебрег некоторыми противоречащими ей фактами: тем, что аргон с атомным весом 39,94 стоит в таблице раньше калия, атомный вес которого 39,09, так же как и аналогичными нарушениями закона между элементами 27 и 28, 52 и 53 и еще в четырех местах таблицы. Формулировка, которая содержала бы все эти оговорки, была бы неприемлемо сложной. Это противоречие между точностью и сложностью формулировки было устранено после того, как удалось установить, что свойства элементов являются периодической функцией не атомного веса, а величины заряда ядра. Исключений из этого простого закона в таблице Менделеева нет.

### ПРИМЕР 1.7

Примером, подтверждающим верность "правила красоты", может служить история открытия структуры молекулы дезоксирибонуклеиновой кислоты (ДНК), описанная в книге Д.Д.Уотсона "Двойная спираль". На завершающей стадии исследования была предпринята попытка сделать объемную модель молекулы ДНК. Для этого изготовили металлические шарики для каждого атома этой молекулы (химическая формула ДНК была известна). "Атомы" с положительной валентностью имели штырьки, а с отрицательной - соответствующее количество отверстий. Из этих элементов исследователям удалось собрать модель, в которой были задействованы все атомы и не было ни одного лишнего штырька или отверстия. Все в точности соответствовало составу молекулы. Однако конструкторы модели после некоторого разглядывания ее единодушно решили: нет, эта модель не соответствует действительности. Слишком уж она некрасива!

Модель разобрали и после нескольких попыток собрали новую. Она, так же как и первая, соответствовала химической формуле, но в отличие от нее была красива, изящна, гармонична. Более поздние рентгеновские исследования подтвердили, что натуральная молекула устроена именно так, как вторая, более красивая модель.

### ПРИМЕР 1.8

Вернемся к примеру 1.3. Верно ли то, что уравнение  $x=A\sin(\alpha t)$  будет по-прежнему адекватно описывать малые колебания маятника, если он находится на Луне? Прямое сравнение объекта и модели, а следовательно, установление изоморфизма в данном случае исключено. Тем не менее мы можем с некоторой поправкой – заменой значения константы  $g$ , входящей в формулу подсчета частоты  $\alpha$ , считать, что это уравнение будет адекватно описывать малые колебания маятника и на Луне. Заметим, что при движении маятника в соответствии с приведенным выше уравнением выполняется закон сохранения энергии, в чем можно убедиться прямым подсчетом.

Устанавливая адекватность модели в новых условиях, мы в данном случае руководствовались общенаучным принципом соответствия — если корректно уточнить адекватную модель (в данном случае с соблюдением закона сохранения энергии), то в результате получится адекватная модель.

### ЗАМЕЧАНИЕ

Адекватность - весьма тонкое и многогранное понятие. Иногда проще установить неадекватность модели, т.е. ее несоответствие некоторым общим научным принципам. Например, если в модели одновременно допустимы утверждения  $A$  и  $\neg A$ , то эта модель заведомо неадекватна. Сам же принцип познания через отрицание называется *апофатическим принципом*. Апофатический принцип позволяет строго фиксировать границы нашего познания, и некоторые философы предсказывают, что этот принцип будет ведущим принципом познания в XXI в.

## ЛЕКЦИЯ 2

### ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ

#### 2.1. Информационные модели

Информационные модели — класс знаковых моделей, описывающих информационные процессы (возникновение, передачу, преобразование и использование информации) в системах самой разнообразной природы.

Граница между вербальными, математическими и информационными моделями может быть проведена весьма условно. Возможно, информационные модели следовало бы считать подклассом математических моделей. Однако в рамках информатики как самостоятельной науки, отдельной от математики, физики, лингвистики и других наук, выделение класса информационных моделей является целесообразным.

Информационные модели можно разбить на классы:

1. По степени формализации:
  - формализованные;
  - частично формализованные;
  - неформализованные.
2. По роли в управлении объектом:
  - регистрирующие;
  - эталонные;
  - прогностические;
  - имитационные;
  - оптимизационные.
3. По фактору учета времени:
  - статические;
  - динамические, которые подразделяются на детерминированные и вероятностные или стохастические.

## ЗАМЕЧАНИЕ

В *детерминированных* системах новое состояние зависит только от времени и текущего состояния системы. Другими словами, если имеются условия, определяющие переход системы в новое состояние, то для детерминированной системы можно однозначно указать, в какое именно состояние она перейдет. Для *стохастической* системы можно указать лишь множество возможных состояний перехода и, в некоторых случаях, — вероятностные характеристики перехода в каждое из этих состояний.

4. По языку кодирования:

- дескриптивные (на естественных или специальных языках описания);
- наглядные, т.е. на языках представления (рисунки, чертежи, графики, диаграммы);
- смешанные (графы, таблицы, схемы, карты, видеофильмы и т.п.).

### **2.2. Основные понятия информационного моделирования**

Начиная с конца XX в. основным подходом к информационному моделированию стал объектный подход. Перечислим основные понятия объектно-ориентированного моделирования.

*Экземпляром* будем называть представление предмета реального мира с помощью некоторого набора его характеристик, существенных для решения данной информационной задачи (служащей контекстом построения информационной модели). Множество экземпляров, имеющих одни и те же характеристики и подчиняющихся одним и тем же правилам, называется *объектом*.

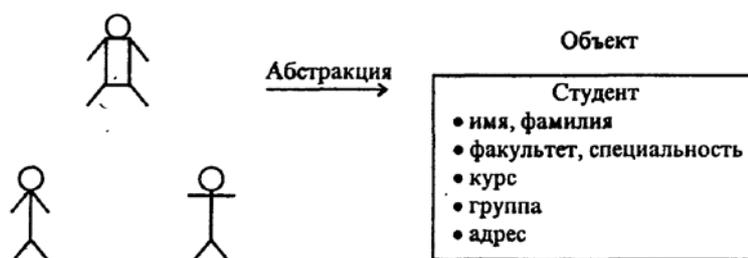


Рис .2.1. Пример абстрагирования при построении информационной модели

Таким образом, *объект* есть абстракция предметов реального мира, объединяемых общими характеристиками и поведением (рис. 2.1).

Информационная модель какой-либо реальной системы состоит из объектов. Каждый объект в модели должен быть обеспечен уникальным и значимым именем (а также идентификатором, служащим ключом для указания этого объекта, связи его с другими объектами модели). Таким образом, *обозначение (наименование) объекта* — это элементарная процедура, лежащая в основе информационного моделирования.

Объект представляет собой один типичный (но неопределенный) экземпляр чего-то в реальном мире и является простейшей информационной моделью. Объекты представляют некие «сущности» предметов реального мира, связанные с решаемой задачей.

*Класс* — это группа объектов, обладающих сходными свойствами, а именно, данными и поведением.

#### ПРИМЕР 2.1

Геометрические фигуры — класс; окружность — объект класса «Геометрические фигуры»; окружность конкретного радиуса — экземпляр класса «Геометрические фигуры».

Большинство объектов, с которыми приходится встречаться, относятся к одной из следующих категорий:

- реальные объекты;
- роли;
- события;
- взаимодействия;
- спецификации.

*Реальный объект* — это абстракция физически существующих предметов. Например, на автомобильном заводе это кузов автомобиля, двигатель, коробка передач; при перевозке грузов это контейнер, средство перевозки.

**Роль** — абстракция цели или назначения человека, части оборудования или учреждения (организации). Например, в университете как в учебном заведении это студент, преподаватель, декан; в университете как в учреждении это приемная комиссия, отдел кадров, бухгалтерия, деканат.

**Событие** — абстракция чего-то случившегося. Например, поступление заявления от абитуриента в приемную комиссию университета, сдача (или несдача) экзамена.

**Взаимодействия** — объекты, получаемые из отношений между другими объектами. Например, сделка, контракт (договор) между двумя сторонами, свидетельство об образовании, выдаваемое учебным заведением его выпускнику.

**Объекты-спецификации** используются для представления правил, стандартов или критериев качества. Например, перечень знаний, умений и навыков выпускника факультета информатики, рецепт проявления фотопленки.

Для каждого объекта должно существовать его **описание** — короткое информационное утверждение, позволяющее установить, является некоторый предмет экземпляром объекта или нет. Например, описание объекта «Абитуриент университета» может быть следующим: человек, имеющий среднее образование, подавший в приемную комиссию документы и заявление о приеме.

Предметы реального мира имеют характеристики (такие, например, как имя, название, регистрационный номер, дата изготовления, вес и т.д.). Каждая отдельная характеристика, общая для всех возможных экземпляров объекта, называется **атрибутом**. Для каждого экземпляра атрибут принимает определенное значение. Так, объект Книга имеет атрибуты Автор, Название, Год издания, Количество страниц.

У каждого объекта должен быть **идентификатор** — множество из одного или более атрибутов, значения которых определяют каждый экземпляр

объекта. Для книги атрибуты Автор и Название совместно образуют идентификатор. В то же время Год издания и Количество страниц идентификаторами быть не могут ни врозь, ни совместно, так как не определяют объект. Объект может иметь и несколько идентификаторов, каждый из которых составлен из одного или нескольких атрибутов. Один из них может быть выбран как привилегированный для соответствующей ситуации.

### ***Способы представления объекта:***

1. ***Графически.*** Объект вместе со своими атрибутами может быть изображен в виде рамки, содержащей имя объекта и имена атрибутов. Атрибуты, которые составляют привилегированный идентификатор объекта, могут быть выделены (например, символом \* слева от имени атрибута):

Книга
* Автор
* Название
. Год издания
. Количество страниц

2. ***В текстовом представлении.*** В эквивалентном текстовом представлении это может иметь следующий вид:  
Книга (Автор. Название. Год издания. Количество страниц).  
Привилегированный идентификатор подчеркивается.
3. ***Таблицей.*** В этой интерпретации каждый экземпляр объекта является строкой в таблице, а значения атрибутов, соответствующих каждому экземпляру, — ячейками строки (табл.2.1).

Таблица 2.1. Таблица как представление информационной модели

Автор	Книга		
	название	год	количество
Грин А. С	Бегущая по волнам	1988	279
Стивенсон Р. Л.	Остров сокровищ	1992	269
Скотт В.	Ричард Львиное Сердце	1993	349
Гончаров И. А.	Обрыв	1986	598

Можно классифицировать атрибуты по принадлежности к одному из трех различных типов:

- описательные;
- указывающие;
- вспомогательные.

**Описательные** атрибуты представляют факты, внутренне присущие каждому экземпляру объекта. Если значение описательного атрибута изменится, то это говорит о том, что некоторая характеристика экземпляра изменилась, но сам экземпляр остался прежним.

**Указательные** атрибуты могут использоваться как идентификаторы (или часть идентификаторов) экземпляра. Если значение указывающих атрибутов изменяется, то это говорит лишь о том, что новое имя дается тому же самому экземпляру.

**Вспомогательные** атрибуты используются для связи экземпляра одного объекта с экземпляром другого объекта.

## ПРИМЕР 2.2

Автомобиль (гос. номер; марка; цвет; владелец).

Атрибут «цвет» является описательным, атрибуты «гос. номер» и «марка» — указательными, атрибут «владелец» — вспомогательным, служащим для связи экземпляра объекта Автомобиль с экземпляром объекта Автолюбитель. Если значение вспомогательного атрибута изменится, это говорит о том, что теперь другие экземпляры объектов связаны между собой.

### 2.3. Связи между объектами

В реальном мире между предметами существуют различные отношения. Если предметы моделируются как объекты, то отношения, которые систематически возникают между различными видами объектов, отражаются в информационных моделях как связи. Каждая связь задается в модели определенным именем. Связь в графической форме представляется как линия между связанными объектами и обозначается идентификатором связи.

Существует три вида связи: один-к-одному, один-ко-многим и многие-ко-многим (рис. 2.2-2.4).

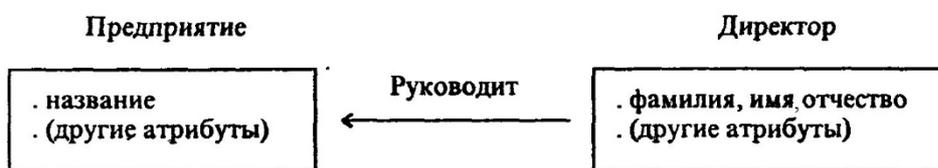


Рис. 2.2. Пример связи «один-к-одному»

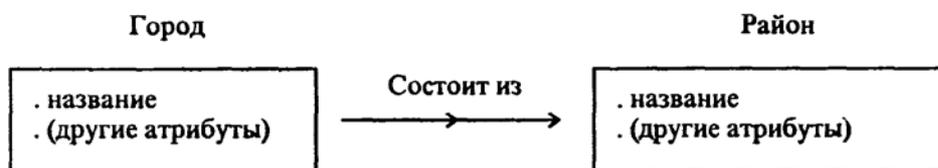


Рис. 2.3. Пример связи «один-ко-многим»

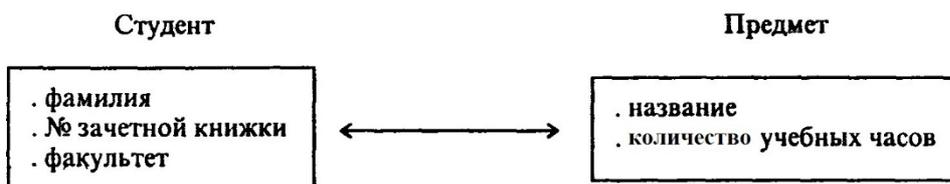


Рис. 2.4. Пример связи «многие-ко-многим»

Помимо множественности, связи могут подразделяться на безусловные и условные. В безусловной связи для участия в ней требуется каждый экземпляр объекта. В условной связи принимают участие не все экземпляры объекта. Связь может быть условной как с одной, так и с обеих сторон.

Все связи в информационной модели требуют описания, которое, как минимум, включает:

- идентификатор связи;
- формулировку сущности связи;
- вид связи (ее множественность и условность);
- способ описания связи с помощью вспомогательных атрибутов объектов.

Дальнейшее развитие представлений информационного моделирования связано с развитием понятия связи и структур, ими образуемых. Самой распространенной информационной моделью является, так называемая, **графовая структура**. Она является основой решения огромного количества задач информационного моделирования.

**Разновидности графовой структуры:** очередь, циклическая структура, деревья, сети, блок-схемы, базы данных (граф, в котором узлами являются таблицы, а дугами — связи), гипертекст (граф, узлами которого являются тексты, а дугами — гиперссылки), карты и схемы (графы, наложенные на изображение).

### ПРИМЕР 2.3

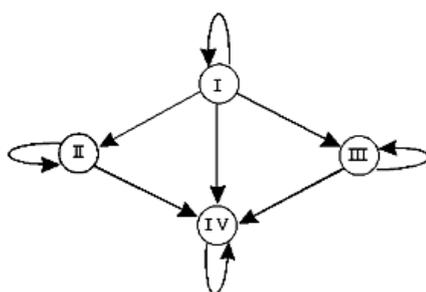


Рис. 2.5. Граф совместимости групп крови



Рис. 2.6. Дерево административной структуры РФ

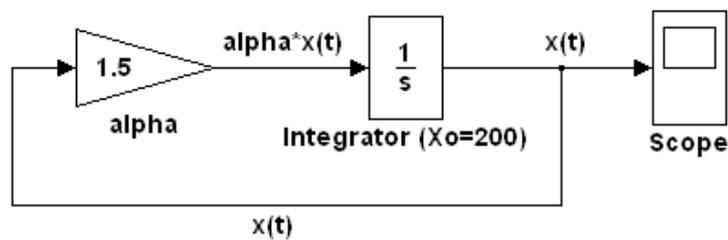


Рис.2.7. S-модель динамики популяций Мальтуса ( $\frac{\partial x}{\partial t} = \alpha x$ )

Разновидностью информационного моделирования является математическое моделирование.

## ЛЕКЦИЯ 3 МАТЕМАТИЧЕСКИЕ МОДЕЛИ

### 3.1. Основные понятия

*Математическая модель* — приближенное описание объекта моделирования, выраженное с помощью математической символики.

Математические модели появились вместе с математикой много веков назад. Огромный толчок развитию математического моделирования придало появление ЭВМ. Применение вычислительных машин позволило

проанализировать и применить на практике многие математические модели, которые раньше не поддавались аналитическому исследованию. Реализованная на компьютере математическая модель называется **компьютерной математической моделью**, а проведение целенаправленных расчетов с помощью компьютерной модели называется **вычислительным экспериментом**.

**Этапы компьютерного математического моделирования** изображены на рис. 3.1.

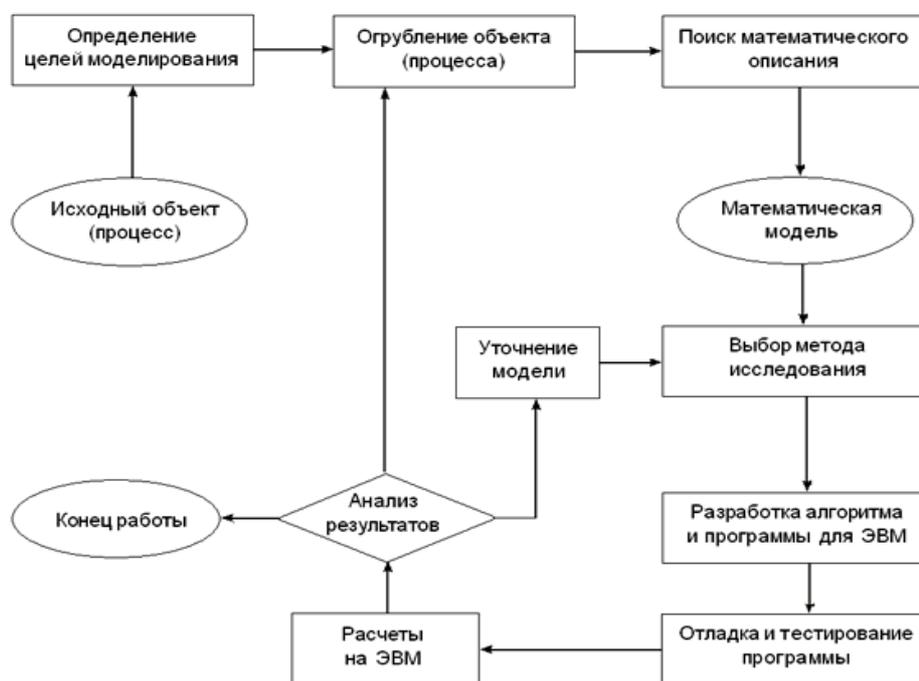


Рис. 3.1. Этапы математического моделирования

Первый этап — *определение целей моделирования, анализ проблемы и постановка задачи*. Цели моделирования могут быть различными.

1. Модель нужна для того, чтобы *понять*, как устроен конкретный объект, каковы его структура, основные свойства, законы развития и взаимодействия с окружающим миром.

### ПРИМЕР 3.1

Пусть объект исследования — взаимодействие потока жидкости или газа с телом, являющимся для этого потока препятствием. Опыт показывает, что сила сопротивления потоку со стороны тела растет с

ростом скорости потока, но при некоторой достаточно высокой скорости эта сила скачком уменьшается с тем, чтобы с дальнейшим увеличением скорости снова возрасти. Что же вызвало уменьшение силы сопротивления? Математическое моделирование позволяет получить четкий ответ: в момент скачкообразного уменьшения сопротивления вихри, образующиеся в потоке жидкости или газа позади обтекаемого тела, начинают отрываться от него и уноситься потоком.

2. Модель нужна для того, чтобы научиться *управлять* объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях.

### ПРИМЕР 3.2

Какой режим полета самолета выбрать для того, чтобы полет был безопасным и экономически наиболее выгодным? Как составить график выполнения сотен видов работ на строительстве большого объекта, чтобы оно закончилось в максимально короткий срок? Множество таких проблем систематически возникает перед экономистами, конструкторами, учеными.

3. Модель нужна для того, чтобы *прогнозировать* прямые и косвенные последствия реализации заданных способов и форм воздействия на объект.

### ПРИМЕР 3.3

Прогнозирование последствий тех или иных воздействий на объект может быть как относительно простым делом в несложных физических системах, так и чрезвычайно сложным — на грани выполнимости — в системах биолого-экономических, социальных. Если ответить на вопрос об изменении режима распространения тепла в тонком стержне при изменениях в составляющем его сплаве относительно легко, то проследить (предсказать) экологические и климатические последствия строительства крупной ГЭС или социальные последствия изменений налогового

законодательства несравненно труднее. Возможно, и здесь методы математического моделирования будут оказывать в будущем более значительную помощь.

Второй этап - *огрубление модели* - определение входных и выходных параметров модели; разделение входных параметров по степени важности влияния их изменений на выходные. Такой процесс называется ранжированием или разделением по рангам.

Составим список величин, от которых зависит поведение объекта или ход процесса, а также тех величин, которые желательно получить в результате моделирования. Обозначим первые из них (входные) через  $x_1, x_2, \dots, x_n$ ; вторые (выходные) через  $y_1, y_2, \dots, y_n$ . Символически поведение объекта или процесса можно представить в виде

$$y_j = F_j(x_1, x_2, \dots, x_n), j=1, 2, \dots, n,$$

где  $F_j$  — те действия, которые следует произвести над входными параметрами, чтобы получить результаты.

Хотя запись  $F_j(x_1, x_2, \dots, x_n)$  напоминает обозначение функции, мы здесь используем ее в более широком смысле. Лишь в простейших ситуациях здесь  $F$  есть функция в обычном смысле; чаще всего она выражает лишь наличие некоторой связи между входными и выходными параметрами модели.

Входные параметры  $x_i$  могут быть известны «точно», т.е. поддаваться (по крайней мере, в принципе) измерению однозначно и с любой заданной степенью точности — тогда они являются детерминированными величинами. Так, в классической механике, сколь сложной ни была бы моделируемая система, входные параметры детерминированы, и соответственно, детерминирован процесс эволюции такой системы. Однако в природе и обществе гораздо чаще встречаются процессы иного рода, когда значения входных параметров известны лишь с определенной степенью вероятности, т.е. эти параметры являются вероятностными (стохастическими), и, соответственно, случайным является процесс эволюции системы.

Случайный – не значит непредсказуемый. Просто в этой ситуации характер исследования и задаваемых вопросов резко меняется – они приобретают вид «С какой вероятностью...?», «С каким математическим ожиданием...?» и т.п. Примеров случайных процессов не счесть как в науке, так и в обыденной жизни (силы, действующие на летящий самолет в ветреную погоду; переход улицы при большом потоке транспорта и т.д.).

Для стохастической модели выходные параметры могут быть как величинами вероятностными, так и однозначно определяемыми. Например, на перекрестке улиц можно ожидать зеленого сигнала светофора и полминуты, и две минуты (с разной вероятностью), но среднее время ожидания есть величина вполне определенная, и именно она может быть объектом моделирования.

Важнейшим этапом моделирования является разделение входных параметров по степени важности влияния их изменений на выходные. Такой процесс называется ранжированием (разделением по рангам). Чаще всего невозможно, да и не нужно учитывать все факторы, которые могут повлиять на значения интересующих нас величин  $y_j$ . От того, насколько умело выделены важнейшие факторы, зависит успех моделирования, быстрота и эффективность достижения цели. Выделить наиболее значимые факторы и отсеять менее важные может лишь специалист в той предметной области, к которой относится модель.

Отбрасывание менее значимых факторов огрубляет объект моделирования и способствует пониманию его главных свойств и закономерностей. Умело ранжированная модель должна быть адекватна исходному объекту или процессу в отношении целей моделирования. Обычно определить, адекватна ли модель, можно только в процессе экспериментов с ней, анализа результатов первоначального моделирования.

Третий этап - *построение математической модели*. На этом этапе происходит переход от абстрактной формулировки модели к формулировке,

имеющей конкретное математическое представление. Математическая модель — это уравнения, системы уравнений, системы неравенств, дифференциальные уравнения или системы таких уравнений и пр.

Четвертый этап - *выбор метода исследования математической модели.*

Чаще всего здесь используются численные методы, которые хорошо поддаются программированию. Как правило, для решения одной и той же задачи подходит несколько методов, различающихся точностью, устойчивостью и т.д. От верного выбора метода часто зависит успех всего процесса моделирования.

Пятый этап - *разработка алгоритма, составление программы для ЭВМ* — трудно формализуемый процесс. В зависимости от характера задачи и склонностей программиста используются языки программирования FORTRAN, PASCAL, DELPHI, C и др. или специальные пакеты прикладных программ для решения математических задач (MATLAB, MathCAD и т.п.).

Шестой этап - *отладка и тестирование программы.* Работа программы проверяется на тестовой задаче с заранее известным ответом. Это — лишь начало процедуры тестирования, которую трудно описать формально исчерпывающим образом. Обычно тестирование заканчивается тогда, когда пользователь по своим профессиональным признакам сочтет программу верной.

Седьмой этап - *собственно вычислительный эксперимент*, в процессе которого выясняется, соответствует ли модель реальному объекту (процессу). Модель достаточно адекватна реальному процессу, если некоторые характеристики процесса, полученные на ЭВМ, совпадают с экспериментально полученными характеристиками с заданной степенью точности. В случае несоответствия модели реальному процессу возвращаемся к одному из предыдущих этапов.

### 3.2. Классификация математических моделей

В основу классификации математических моделей можно положить различные принципы. Можно классифицировать модели по отраслям наук (математические модели в физике, биологии, социологии и т.д.). Можно классифицировать по применяемому математическому аппарату (модели, основанные на применении обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, стохастических методов, дискретных алгебраических преобразований и т.д.). Наконец, если исходить из общих задач моделирования в разных науках безотносительно к математическому аппарату, наиболее естественна такая классификация:

- дескриптивные (описательные) модели;
- оптимизационные модели;
- многокритериальные модели;
- игровые модели.

Поясним это на примерах.

**Дескриптивные (описательные) модели.** Например, моделирование движения кометы, вторгшейся в Солнечную систему, производится с целью предсказания траектории ее полета, расстояния, на котором она пройдет от Земли, и т.д. В этом случае цели моделирования носят описательный характер, поскольку нет никаких возможностей повлиять на движение кометы, что-то в нем изменить.

**Оптимизационные модели** используются для описания процессов, на которые можно воздействовать, пытаясь добиться достижения заданной цели. В этом случае в модель входят один или несколько параметров, доступных влиянию. Например, меняя тепловой режим в зернохранилище, можно задаться целью подобрать такой режим, чтобы достичь максимальной сохранности зерна, т.е. оптимизировать процесс хранения.

**Многокритериальные модели.** Нередко приходится оптимизировать процесс по нескольким параметрам одновременно, причем цели могут быть

весьма противоречивыми. Например, зная цены на продукты и потребность человека в пище, нужно организовать питание больших групп людей (в армии, детском летнем лагере и др.) физиологически правильно и, одновременно с этим, как можно дешевле. Ясно, что эти цели совсем не совпадают, т.е. при моделировании будет использоваться несколько критериев, между которыми нужно искать баланс.

**Игровые модели** могут иметь отношение не только к компьютерным играм, но и к весьма серьезным вещам. Например, полководец перед сражением при наличии неполной информации о противостоящей армии должен разработать план: в каком порядке вводить в бой те или иные части и т.д., учитывая и возможную реакцию противника. Есть специальный раздел современной математики — теория игр, изучающая методы принятия решений в условиях неполной информации.

### 3.3. Пример построения математической модели

**Этап 1. Определение цели моделирования, анализ проблемы, постановка задачи**

Цель моделирования — понять, как движется Луна в гелиоцентрической системе координат. Для достижения этой цели необходимо найти зависимость изменения со временем радиус-вектора  $R_{\text{ЛС}}(t)$  движения Луны вокруг Солнца.

**Этап 2. Определение входных и выходных параметров модели**

Примем следующие допущения:

- 1) сложное движение Луны вокруг Солнца представим **в виде суммы двух движений**: движения Земли вокруг Солнца и движения Луны вокруг Земли;
- 2) Земля вокруг Солнца и Луна вокруг Земли движутся по **круговым орбитам** (эллиптичностью орбит Земли и Луны пренебрежем) с параметрами:

$R_{ЗС} = 1,496 \cdot 10^8$  км — радиус орбиты движения Земли вокруг Солнца;  
 $T_{ЗС} = 3,156 \cdot 10^7$  с — период движения Земли вокруг Солнца;  
 $R_{ЛЗ} = 3,844 \cdot 10^5$  км — радиус орбиты движения Луны вокруг Земли;  
 $T_{ЛЗ} = 2,36 \cdot 10^6$  с — период движения Луны вокруг Земли.

Таким образом, входными параметрами модели будут  $R_{ЗС}$ ,  $T_{ЗС}$ ,  $R_{ЛЗ}$ ,  $T_{ЛЗ}$ .

### **Этап 3. Построение математической модели**

Рассмотрим движение Луны (точка **Л**) в гелиоцентрической системе отсчета с началом координат в точке **С** и геоцентрической системе отсчета с началом координат в точке **З** (рис. 3.2).

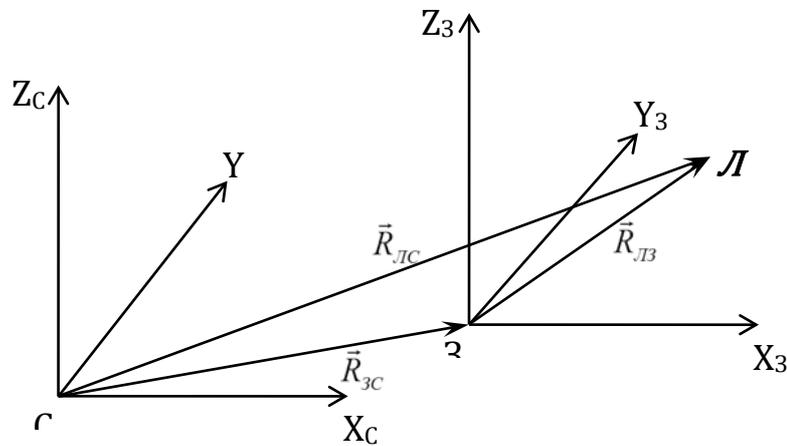


Рис. 3.2. Движение Луны в гелиоцентрической и геоцентрической системах координат

Тогда движение Луны вокруг Солнца можно описать радиус-вектором  $\vec{R}_{LC}(t)$ :

$$\vec{R}_{LC}(t) = \vec{R}_{LZ}(t) + \vec{R}_{ZC}(t). \quad (3.1)$$

При движении материальной точки по окружности радиуса  $R$  с постоянной угловой скоростью  $\omega$  координаты радиус-вектора, проведенного из начала координат к текущему положению точки, меняются по закону

$$\vec{R}(t) = \begin{cases} R \cos(\omega t + \varphi_0) \\ R \sin(\omega t + \varphi_0) \end{cases} = \begin{cases} R \cos(\frac{2\pi}{T} t + \varphi_0), \\ R \sin(\frac{2\pi}{T} t + \varphi_0) \end{cases} \quad (3.2)$$

где  $\varphi_0$  — начальная фаза, которую далее будем считать равной 0.

Тогда движение Луны в гелиоцентрической системе координат будет описываться уравнениями

$$\vec{R}_{ЛС}(t) = \begin{cases} x_{ЛС}(t) \\ y_{ЛС}(t) \end{cases} = \begin{cases} R_{ЛЗ} \cos\left(\frac{2\pi}{T_{ЛЗ}}t\right) + R_{СЗ} \cos\left(\frac{2\pi}{T_{СЗ}}t\right), \\ R_{ЛЗ} \sin\left(\frac{2\pi}{T_{ЛЗ}}t\right) + R_{СЗ} \sin\left(\frac{2\pi}{T_{СЗ}}t\right) \end{cases}. \quad (3.3)$$

**Этап 4. Выбор метода исследования математической модели.**

Решать систему уравнений (3.3) будем численно средствами среды для динамического моделирования Simulink.

**Этап 5. Разработка алгоритма**

S-модель, соответствующая расчетным формулам (3.3), представлена на рис.3.3.

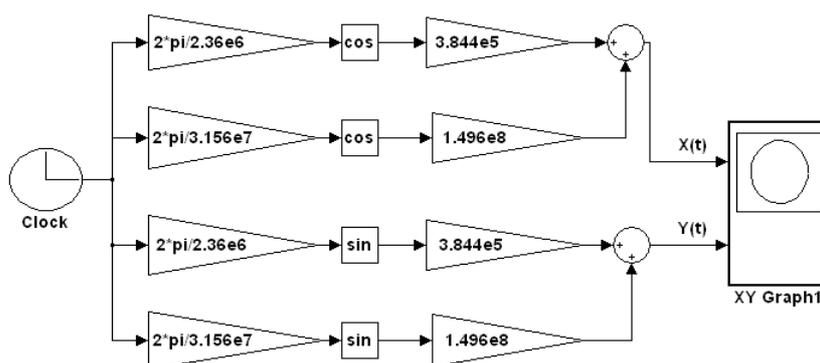


Рис. 3.3. S-модель движения Луны в гелиоцентрической системе координат.

**Этапы 6-7. Отладка программы, проведение модельного эксперимента и анализ результатов**

Зададим интервал моделирования, приблизительно равный  $T_{СЗ}$ , а диапазон значений по осям координат блока *XY-Graph* приблизительно от  $-R_{СЗ}$  до  $+R_{СЗ}$ . Шаг модельного времени выбираем таким образом, чтобы, с одной стороны, график строился с необходимой точностью, а, с другой стороны, достаточно быстро. Результат проведения модельного эксперимента представлен на рис. 3.4.

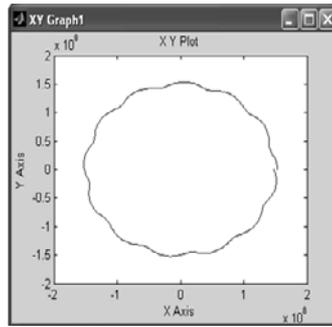


Рис. 3.4. Траектория движения Луны в гелиоцентрической системе координат

Проанализируем полученный результат. Найдем отношение модуля силы притяжения Луны к Солнцу  $F_{ЛС}$  к модулю силы притяжения Луны к Земле  $F_{ЛЗ}$ :

$$\frac{F_{ЛС}}{F_{ЛЗ}} = \frac{G \cdot \frac{m_{Л} \cdot m_{С}}{R_{ЛС}^2}}{G \cdot \frac{m_{Л} \cdot m_{З}}{R_{ЛЗ}^2}} = \frac{m_{С}}{m_{З}} \cdot \frac{R_{ЛЗ}^2}{R_{ЛС}^2} \approx 2,2.$$

Луна притягивается к Солнцу примерно в 2 раза сильнее, чем к Земле. Таким образом, Луна обращается вокруг Солнца, но близкое расположение Земли искривляет траекторию движения Луны (рис.3.5).

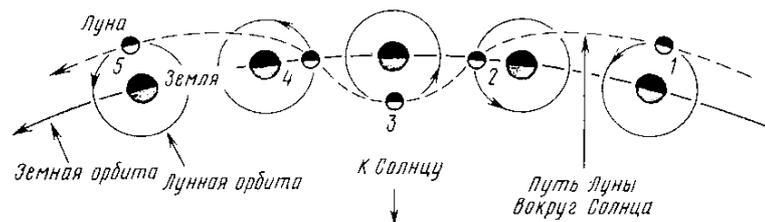


Рис. 3.5. Траектории движения Земли и Луны вокруг Солнца

Анализируя рис. 3.4 и 3.5, убеждаемся в адекватности построенной модели. Теперь можно проводить модельные эксперименты, меняя входные параметры модели.

## ЛЕКЦИЯ 4

### ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

#### 4.1. Понятие имитационного моделирования

Имитационное моделирование (от англ. simulation) – это распространенная разновидность аналогового моделирования, реализуемого с помощью набора математических инструментальных средств, специальных имитирующих компьютерных программ и технологий программирования, позволяющих посредством процессов-аналогов провести целенаправленное исследование структуры и функций реального сложного процесса в памяти компьютера в режиме имитации, выполнить оптимизацию некоторых его параметров.

Имитационной моделью называется специальный программный комплекс, который позволяет имитировать деятельность какого-либо сложного объекта. Он запускает в компьютере параллельные взаимодействующие вычислительные процессы, которые являются по своим временным параметрам (с точностью до масштабов времени и пространства) аналогами исследуемых процессов. В странах, занимающих лидирующее положение в создании новых компьютерных систем и технологий, научное направление (Computer Science) использует именно такую трактовку имитационного моделирования, а в программах магистерской подготовки по данному направлению имеется соответствующая учебная дисциплина.

Следует отметить, что любое моделирование имеет в своей методологической основе элементы имитации реальности с помощью какой-либо символики (математики) или аналогов. Поэтому иногда в российских вузах имитационным моделированием стали называть целенаправленные серии многовариантных расчетов, выполняемых на компьютере с применением экономико-математических моделей и методов. Однако с точки зрения компьютерных технологий, такое моделирование – это обычные

вычисления, выполняемые с помощью расчетных программ или табличного процессора Excel. Математические расчеты (в том числе табличные) можно производить и без ЭВМ: используя калькулятор, логарифмическую линейку, правила арифметических действий и вспомогательные таблицы. Но имитационное моделирование – это чисто компьютерная работа, которую невозможно выполнить подручными средствами. Поэтому часто для этого вида моделирования используется синоним компьютерное моделирование.

Имитационную модель нужно создавать. Для этого необходимо специальное программное обеспечение – система моделирования. Специфика такой системы определяется технологией работы, набором языковых средств, сервисных программ и приемов моделирования.

Имитационная модель должна отражать большое число параметров, логику и закономерности поведения моделируемого объекта во времени (временная динамика) и в пространстве (пространственная динамика). Моделирование объектов экономики связано с понятием финансовой динамики объекта. С точки зрения специалиста (информатика-экономиста, математика-программиста или экономиста-математика), имитационное моделирование контролируемого процесса или управляемого объекта – это высокоуровневая информационная технология, которая обеспечивает два вида действий, выполняемых с помощью компьютера:

- 1) работы по созданию или модификации имитационной модели;
- 2) эксплуатацию имитационной модели и интерпретацию результатов.

Имитационное (компьютерное) моделирование экономических процессов обычно применяется в двух случаях:

- для управления сложным бизнес-процессом, когда имитационная модель управляемого экономического объекта используется в качестве инструментального средства в контуре адаптивной системы управления, создаваемой на основе информационных (компьютерных) технологий;

- при проведении экспериментов с дискретно-непрерывными моделями сложных экономических объектов для получения и отслеживания их динамики в экстренных ситуациях, связанных с рисками, натурное моделирование которых нежелательно или невозможно.

Можно выделить следующие типовые задачи, решаемые средствами имитационного моделирования при управлении экономическими объектами:

- моделирование процессов логистики для определения временных и стоимостных параметров;
- управление процессом реализации инвестиционного проекта на различных этапах его жизненного цикла с учетом возможных рисков и тактики выделения денежных сумм;
- анализ клиринговых процессов в работе сети кредитных организаций (в том числе, применение к процессам взаимозачетов в условиях российской банковской системы);
- прогнозирование финансовых результатов деятельности предприятия на конкретный период времени (с анализом динамики сальдо на счетах);
- бизнес - реинжиниринг несостоятельного предприятия (изменение структуры и ресурсов предприятия-банкрота, после чего с помощью имитационной модели можно сделать прогноз основных финансовых результатов и дать рекомендации о целесообразности того или иного варианта реконструкции, инвестиций или кредитования производственной деятельности);
- анализ адаптивных свойств и живучести компьютерной региональной банковской информационной системы (например, частично вышедшая из строя после катастрофического землетрясения 1995 г. в Японии система электронных расчетов и платежей продемонстрировала высокую живучесть: операции возобновились

через несколько дней);

- оценка параметров надежности и задержек в централизованной экономической информационной системе с коллективным доступом (на примере системы продажи авиабилетов с учетом несовершенства физической организации баз данных и отказов оборудования);
- анализ эксплуатационных параметров распределенной многоуровневой ведомственной информационной управляющей системы с учетом неоднородной структуры, пропускной способности каналов связи и несовершенства физической организации распределенной базы данных в региональных центрах;
- моделирование действий курьерской вертолетной группы в регионе, пострадавшем в результате природной катастрофы или крупной промышленной аварии;
- анализ сетевой модели РБВ.Т для проектов замены и наладки производственного оборудования с учетом возникновения неисправностей;
- анализ работы автотранспортного предприятия, занимающегося коммерческими перевозками грузов, с учетом специфики товарных и денежных потоков в регионе;
- расчет параметров надежности и задержек обработки информации в банковской информационной системе.

Приведенный перечень является неполным и охватывает те примеры использования имитационных моделей, которые описаны в литературе или применялись авторами на практике. Действительная область применения аппарата имитационного моделирования (ИМ) не имеет видимых ограничений. Например, спасение американских астронавтов при возникновении аварийной ситуации на корабле APOLLO стало возможным только благодаря проигрыванию различных вариантов спасения на моделях космического комплекса.

Система ИМ, обеспечивающая создание моделей для решения перечисленных задач, должна обладать следующими свойствами:

- возможностью применения имитационных программ совместно со специальными экономико-математическими моделями и методами, основанными на теории управления;
- инструментальными методами проведения структурного анализа сложного экономического процесса;
- способностью моделирования материальных, денежных и информационных процессов и потоков в рамках единой модели, в общем модельном времени;
- возможностью введения режима постоянного уточнения при получении выходных данных (основных финансовых показателей, временных и пространственных характеристик, параметров рисков и др.) и проведении экстремального эксперимента.

Историческая справка. Имитационное моделирование экономических процессов – разновидность экономико-математического моделирования. Однако этот вид моделирования в значительной степени базируется на компьютерных технологиях. Многие моделирующие системы, идеологически разработанные в 1970-1980-х гг., претерпели эволюцию вместе с компьютерной техникой и операционными системами (эффективно используются в настоящее время на новых компьютерных платформах). Кроме того, в конце 1990-х гг. появились принципиально новые моделирующие системы, концепции которых не могли возникнуть раньше – при использовании ЭВМ и операционных систем 1970-1980-х гг.

#### **4.2. Технология имитационного моделирования**

Имитационное моделирование реализуется посредством набора математических инструментальных средств, специальных компьютерных программ и приемов, позволяющих с помощью компьютера провести

целенаправленное моделирование в режиме имитации структуры и функций сложного процесса, а также оптимизацию некоторых его параметров. Набор программных средств и приемов моделирования определяет специфику системы моделирования – специального программного обеспечения.

В отличие от других видов и способов математического моделирования с применением ЭВМ, имитационное моделирование имеет свою специфику: запуск в компьютере взаимодействующих вычислительных процессов, которые являются по своим временным параметрам – с точностью до масштабов времени и пространства – аналогами исследуемых процессов.

### **4.3. Этапы имитационного моделирования**

Имитационное моделирование как особая информационная технология состоит из следующих основных этапов.

**1. Структурный анализ процессов.** Проводится формализация структуры сложного реального процесса путем разложения его на подпроцессы, выполняющие определенные функции и имеющие взаимные функциональные связи согласно легенде, разработанной рабочей экспертной группой. Выявленные подпроцессы, в свою очередь, могут разделяться на другие функциональные подпроцессы. Структура общего моделируемого процесса может быть представлена в виде графа, имеющего иерархическую многослойную структуру. В результате появляется формализованное изображение имитационной модели в графическом виде.

Экономические процессы содержат подпроцессы, не имеющие физической основы и протекающие виртуально, так как оперируют с информацией, деньгами, логикой, законами и их обработкой, поэтому структурный анализ является эффективным этапом при моделировании экономических процессов.

На этом этапе описываются *экзогенные переменные*, т.е. те переменные, которые задаются вне модели, т.е. известны заранее. Еще описываются параметры - это коэффициенты уравнений модели. Часто их не разделяют.

*Эндогенные переменные* - это те переменные, которые определяются в ходе расчетов по модели и не задаются в ней извне.

**2. Формализованное описание модели.** Графическое изображение модели, функции, выполняемой каждым подпроцессом, условия взаимодействия всех подпроцессов и особенности поведения моделируемого процесса (временная, пространственная, финансовая динамики должны быть описаны на специальном языке одним из способов):

- описание вручную на алгоритмическом языке, т.е. написание программы на языке программирования;
- автоматизированное описание с помощью компьютерного графического конструктора.

### **3. Построение модели:**

- обычно это трансляция и редактирование связей (сборка модели);
- режимы интерпретации и компиляция;
- верификация (калибровка) параметров, работа на тестовых примерах.

**4. Проведение модельного эксперимента.** Проводится оптимизация определенных параметров реального процесса. Этому должен предшествовать процесс, который называется планированием эксперимента.

## **4.4. Виды имитационного моделирования**

I. **Системная динамика** — вид моделирования, где для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере. Такой вид моделирования более всех других видов помогает понять суть происходящего выявления причинно-следственных связей между объектами и явлениями. С помощью системной динамики строят модели бизнес-процессов, развития города, модели производства, динамики популяции, экологии и развития эпидемии. Метод основан Форрестером в 1950-х гг.

II. *Дискретно-событийное моделирование* — подход к моделированию, предлагающий абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы, такие как: «ожидание», «обработка заказа», «движение с грузом», «разгрузка» и др. Дискретно-событийное моделирование наиболее развито и имеет огромную сферу приложений — от логистики и систем массового обслуживания до транспортных и производственных систем. Этот вид моделирования наиболее подходит для моделирования производственных процессов. Основан Джеффри Гордоном в 1960-х гг.

III. *Агентное моделирование* — относительно новое (1990-е - 2000-е гг.) направление в имитационном моделировании, которое используется для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами и законами (как в других видах моделирования), а наоборот, глобальные правила и законы являются результатом индивидуальной активности членов группы. Цель агентных моделей — получить представление об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении ее отдельных активных объектов и взаимодействии этих объектов в системе. Агент — некая сущность, обладающая активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением, а также самостоятельно изменяться.

## ЛЕКЦИЯ 5

### МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ПРОЦЕССОВ

#### 5.1. Введение

В вероятностных моделях смена состояний моделируемой системы определяется случайными величинами.

Событие называется случайным, если оно достоверно непредсказуемо. Случайность пронизывает наш мир и чаще всего играет отрицательную роль в нашей жизни. Однако есть обстоятельства, в которых случайность может оказаться полезной.

Одним из распространенных приближенных методов решения задач вычислительной математики является случайный метод, называемый методом Монте-Карло. Сущность метода заключается в том, что для решения какой-либо математической задачи, связанной с вычислением числа  $I$ , строится некоторая случайная величина  $\xi$ , такая, что математическое ожидание этой случайной величины  $E(\xi)$  является значением искомого решения. Проведя серию вычислительных экспериментов со случайной величиной  $\xi$ , мы можем найти приближенное решение как среднее значение результатов эксперимента.

## 5.2. Метод Монте-Карло

Создателями метода статистических испытаний (метода Монте-Карло) считают американских математиков Д. Неймана и С. Улама. В 1944 г., в связи с работами по созданию атомной бомбы, Нейман предложил широко использовать аппарат теории вероятностей для решения прикладных задач с помощью ЭВМ. Данный метод был назван так в честь города в округе Монако, из-за рулетки, простейшего генератора случайных чисел.

Первоначально метод Монте-Карло использовался главным образом для решения задач нейтронной физики, где традиционные численные методы оказались малопригодными. Далее его влияние распространилось на широкий класс задач статистической физики, очень разных по своему содержанию. К разделам науки, где все в большей мере используется метод Монте-Карло, следует отнести задачи теории массового обслуживания, задачи теории игр и математической экономики, задачи теории передачи сообщений при наличии помех и ряд других.

Метод Монте-Карло (или метод статистических испытаний) можно определить, как **метод моделирования случайных величин с целью вычисления характеристик их распределений**. Суть состоит в том, что результат испытаний зависит от некоторой случайной величины, распределенной по заданному закону. Поэтому результат каждого отдельного испытания носит случайный характер. Как правило, составляется программа для осуществления одного случайного испытания. Проведя серию испытаний, получают **выборку**. Полученные статистические данные обрабатываются и представляются в виде численных оценок интересующих исследователя величин (характеристик системы).

Испытание повторяется  $N$  раз, причем каждый опыт не зависит от остальных, и результаты всех опытов усредняются. Это значит, что число испытаний должно быть достаточно велико, поэтому метод существенно опирается на возможности компьютера.

Теоретической основой метода Монте-Карло являются предельные теоремы теории вероятностей. Они гарантируют высокое качество статистических оценок при весьма большом числе испытаний. Метод статистических испытаний применим для исследования как стохастических, так и детерминированных систем. Практическая реализация метода Монте-Карло невозможна без использования компьютера.

#### Проиллюстрируем метод Монте-Карло.

Пусть требуется определить площадь круга известного диаметра с помощью выборок из значений случайной величины. Впишем круг в квадрат; таким образом, стороны квадрата будут равны диаметру круга. Разобьем далее квадрат на единичные квадраты (каждый площадью 1). Разумеется, можно найти площадь круга подсчетом числа единичных квадратов (или их частей), попавших внутрь круга. Однако наша цель состоит в использовании выборок, поскольку при моделировании информацию можно получать лишь таким

образом. Чтобы объяснить, как это делается, рассмотрим конкретный численный пример.

Пусть круг имеет радиус  $r=5$  см и его центр в точке  $(1; 2)$ . Уравнение окружности будет иметь вид:

$$(x - 1)^2 + (y - 2)^2 = 25.$$

Квадрат определяется его вершинами  $(-4; -3)$ ,  $(6; -3)$ ,  $(-4; 7)$  и  $(6; 7)$ . Любая точка  $(x,y)$  внутри квадрата или на его границе должна удовлетворять неравенствам  $-4 \leq x \leq 6$  и  $-3 \leq y \leq 7$ .

Применение выборок при использовании метода Монте-Карло основано на предположении, что все точки в квадрате могут появляться с одинаковой вероятностью, т.е.  $x$  и  $y$  распределены равномерно с плотностями вероятности

$$f(x) = \begin{cases} 1/10, & \text{при } -4 \leq x \leq 6, \\ 0 & \text{в противном случае;} \end{cases}$$
$$f(y) = \begin{cases} 1/10, & \text{при } -3 \leq y \leq 7, \\ 0 & \text{в противном случае.} \end{cases}$$

Подсчитаем число точек, попавших внутрь круга или на окружность. Предположим, что выборка состоит из  $n$  наблюдений и  $m$  из  $n$  точек попали внутрь круга или на окружность. Тогда  $S_{\text{круга}} = m/n$ ,  $S_{\text{квадрата}} = m/n (10 \times 10) = 100 m/n$ .

Подобный способ оценивания площади круга можно обосновать тем, что в процессе получения выборки любая точка  $(x, y)$  может с одинаковой вероятностью попасть в любое место квадрата. Поэтому отношение  $m/n$  представляет оценку площади круга относительно площади квадрата.

В целях изучения влияния статистической ошибки при моделировании задача решалась для различных значений  $n$ , равных 100, 200, 500, 1000, 2000, 5000 и 10000. Кроме того, при каждом  $n$  было проведено 10 прогонов, в каждом из которых использовались различные последовательности случайных чисел из интервала  $(0,1)$ .

Таблица 5.1. Таблица оценка площади круга

№ прогона	Оценки площади круга ( $r=5$ см) при данном числе испытаний $n$						
	100	200	500	1000	2000	5000	10000
1	78	79,5	77,4	76,2	78,8	78,22	78,77
2	70	77	81	76,2	78,7	78,6	78,23
3	81	79,5	77,2	79	78,15	77,72	78,88
4	70	77	77	79,7	78,7	77,76	78,63
5	79	77	79,4	77	79,45	79	78,21
6	81	76	79,2	78,8	77,65	78,68	78,27
7	77	78	79	77,3	78,4	79,08	79,64
8	78	79,5	80,2	80,2	77,05	78,54	78,27
9	82	76,5	80,4	79,5	79,75	78,34	78,67
10	75	82	75,6	79,8	79	78,22	78,16
<b>Среднее</b>	77,1	78,2	78,64	78,37	78,56	78,42	78,57
<b>Дисперсия</b>	18,3	3,5	3,1	2,4	0,66	0,23	0,22
<b>Точное значение площади = 78,54 см<sup>2</sup></b>							

С ростом числа генерируемых точек (продолжительности прогона модели) оценки площади круга приближаются к точному значению (78,54 см<sup>2</sup>) (таблица 5.1). Сначала оценки колеблются около точного значения, а затем стабилизируются. Это условие обычно достигается после повторения эксперимента достаточное количество раз. Наблюдаемое явление типично для результатов любой имитационной модели. Обычно в большинстве имитационных моделей нас интересуют результаты, полученные в стационарных условиях.

При возрастании  $n$  от 100 до 200 дисперсия резко уменьшается с 18,3 до 3,5. За исключением этого интервала, столь резкого уменьшения дисперсии нигде больше не наблюдается. Кроме того, существует предел, за которым увеличение продолжительности прогона модели уже не дает существенного повышения точности результата, измеряемой дисперсией. Это замечание очень важно, поскольку затраты на эксплуатацию имитационной модели прямо пропорциональны продолжительности прогонов. Поэтому желательно найти компромисс между большой точностью (т.е. малой дисперсией) и небольшими затратами на процедуру получения результатов.

### 5.3. Вычисление интегралов по методу Монте-Карло

Во многих задачах исходные данные носят случайный характер, поэтому для их решения должен применяться статистико-вероятностный подход. На основе таких подходов построен ряд численных методов, которые учитывают случайный характер вычисляемых или измеряемых величин. К ним принадлежит и метод статистических испытаний, называемый также методом Монте-Карло, который применяется к решению некоторых задач вычислительной математики, в том числе и для вычисления интегралов.

Метод Монте-Карло состоит в том, что рассматривается некоторая случайная величина  $\xi$ , математическое ожидание которой равно искомой величине  $x$ :

$$\xi = \frac{\xi_1 + \xi_2 + \dots + \xi_n}{n} \approx x.$$

Проводится серия  $n$  независимых испытаний, в результате которых получается (генерируется) последовательность  $n$  случайных чисел  $\xi_1, \xi_2, \dots, \xi_n$ , и по совокупности этих значений приближенно определяется искомая величина

$$M_\xi = M\left(\frac{1}{n} \sum_{i=1}^n \xi_i\right) = \frac{1}{n} M \sum_{i=1}^n \xi_i = \frac{nx}{n} = x.$$

## Интегрирование одномерных интегралов

Интегрирование одномерных определенных интегралов методом Монте-Карло производится по изложенному ниже алгоритму.

$$I = \int_a^b f(x) dx = \frac{b-a}{n} \sum_{i=1}^n f(x_i),$$

где  $x_i$  - равномерно распределенная случайная величина;  $f(x)$  – подынтегральная функция;  $n$  - количество случайных аргументов  $x_i$ ;  $b$  и  $a$  - верхний и нижний пределы интегрирования. В Маткаде имеется встроенная функция **rnd(x)**, возвращающая равномерно распределенную случайную величину в диапазоне 0 - x. Для решения задач методом Монте-Карло необходимо составить программу с использованием этой функции.

### ПРИМЕР 5.1

Вычислить методом Монте-Карло приведенный ниже интеграл. Приведено решение численным методом и методом Монте-Карло,

$$\int_0^2 \frac{\sqrt[5]{x^3 + 2x + 8}}{\sqrt[3]{x^5 + 3x^4 + 8x + 9}} dx$$

Решение:

Выберем  $N=1000000$

$$\begin{array}{l} W := \\ \text{for } i \in 1..N \\ \quad x \leftarrow \text{rnd}(2) \\ \quad y \leftarrow y + \frac{\sqrt[5]{x^3 + 2x + 8}}{\sqrt[3]{x^5 + 3x^4 + 8x + 9}} \\ \quad W \leftarrow \frac{y \cdot 2}{N} \end{array}$$

Ответ:  $W=1,142$ .

### ПРИМЕР 5.2

Вычислить методом Монте-Карло интеграл

$$y = \int_{-3}^{10} (x^2 + 3x + 8) dx$$

Задача отличается от предыдущей тем, что нижний предел не равен нулю, а функция  $\text{rnd}(x)$  вычисляет случайные числа в пределах 0-x. Поэтому здесь вычисляются два интеграла.

Решение:

Выбираем  $m=1000000$ .

```
y1 := | y2 ← 0
      | y3 ← 0
      | for i ∈ 0.. m
      |   | x1 ← rnd(10)
      |   | y2 ← y2 + (x12 + 3·x1 + 8)
      |   | x2 ← rnd(-3)
      |   | y3 ← y3 + (x22 + 3·x2 + 8)
      |   | i
      | y2 ←  $\frac{10 \cdot y2}{m}$ 
      | y3 ←  $\frac{3 \cdot y3}{m}$ 
      | y1 ← y2 + y3
      | y1
```

Ответ:  $Y1=582,833$ .

## Интегрирование многомерных интегралов

Многомерные интегралы

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_m}^{b_m} f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m$$

вычисляются методом Монте-Карло по алгоритму

$$\frac{(b_1 - a_1)(b_2 - a_2) \dots (b_m - a_m)}{m} \sum_{i=1}^m f(x_1, x_2, \dots, x_m),$$

где  $x_i$  - случайная равномерно распределенная величина;  
 $f(x_1, x_2, \dots, x_m)$  - подынтегральная функция;  $b_i, a_i$  ( $i=1, 2, \dots, m$ ) - верхний и нижний пределы интегрирования.

Многомерные интегралы вычисляются в Маткаде методом Монте-Карло с помощью встроенной функции `runif(m, a, b)`, возвращающей вектор из  $m$  равномерно распределенных случайных чисел в пределах от  $a$  до  $b$ .

### ПРИМЕР 5.3

Вычислить методом Монте-Карло интеграл:

$$\int_0^2 \int_0^2 \int_0^2 (x_1 + x_2 + x_3) dx_1 dx_2 dx_3$$

Данный интеграл имеет одинаковые пределы, что облегчает задачу.

Решение: Выберем  $N=10000$ .

```
w := | y ← 0
      | for i ∈ 1..N
      |   | v ← runif(3, 0, 2)
      |   | y ← y + v_0 + v_1 + v_2
      |   | i
      | w ← 2.2 ·  $\frac{2 \cdot y}{N}$ 
      | w
```

Ответ:  $w=23,915$

### ПРИМЕР 5.4

Вычислить методом Монте-Карло интеграл

$$y = \int_1^5 \int_2^8 x_1 x_2 dx_1 dx_2$$

Решение:

Выберем  $N=100000$ .

```
w := | y ← 0
      | for i ∈ 1..N
      |   | v ← runif(2,0,1)
      |   | x1 ← 4·v0 + 1
      |   | x2 ← 6·v1 + 2
      |   | y ← y + (x1·x2)
      |   | i
      | w ←  $\frac{4 \cdot 6 \cdot y}{N}$ 
      | w
```

Ответ:  $w=362,38$

Так как интегралы имеют различные пределы, мы формируем функцией `runif` вектор  $v$  с диапазоном вычислений от нуля до единицы, а затем преобразуем составляющие этого вектора так, чтобы выдерживались заданные пределы.

Для многомерных интегралов большой размерности вычисление методом Монте-Карло происходит значительно быстрее, чем при использовании встроенной функции интегрирования. Так, интеграл

$$y = \int_1^2 \int_5^{10} \int_2^8 \int_3^7 \int_4^9 \int_1^8 \int_0^5 \int_4^8 \int_6^9 \int_5^7 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} dx_1 dx_2 dx_3 dx_4 dx_5 dx_6 dx_7 dx_8 dx_9 dx_{10}$$

вычисляется на компьютере с тактовой частотой 2 гигагерца встроенной функцией интегрирования более получаса, а методом Монте-Карло при  $N=100000$  менее двух минут.

## Поиск глобального экстремума функции в заданной области методом Монте -Карло

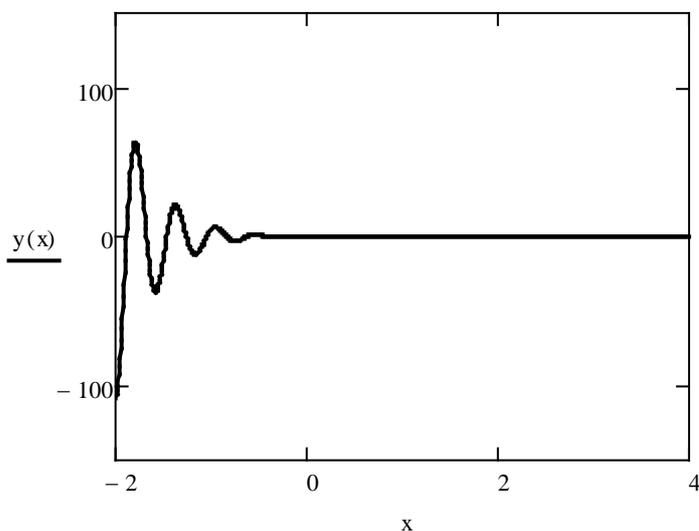
Нелинейные функции в большинстве случаев имеют не один, а несколько экстремумов.

Самый большой максимум или минимум называется глобальным экстремумом, остальные - локальными. При решении задачи, как правило, необходимо определение глобального экстремума. Ниже дается решение такой задачи в Маткаде для заранее заданной области. Вне ее глобальный экстремум может быть другим.

Пусть задана многоэкстремальная функция

$$y = x e^{-2x} \sin(15x).$$

Рассмотрим ее графики при различных пределах изменения аргумента  $x$ :  
На рис. 5.1 видно, что для области изменения аргумента  $-2 \leq x \leq 4$  глобальный экстремум находится в районе  $x=-2$  и равен примерно 75.



*Рис. 5.1. График заданной функции*

Из рис. 5.2 следует, что для другой области изменения аргумента глобальный экстремум находится в районе  $x=0,5$ .

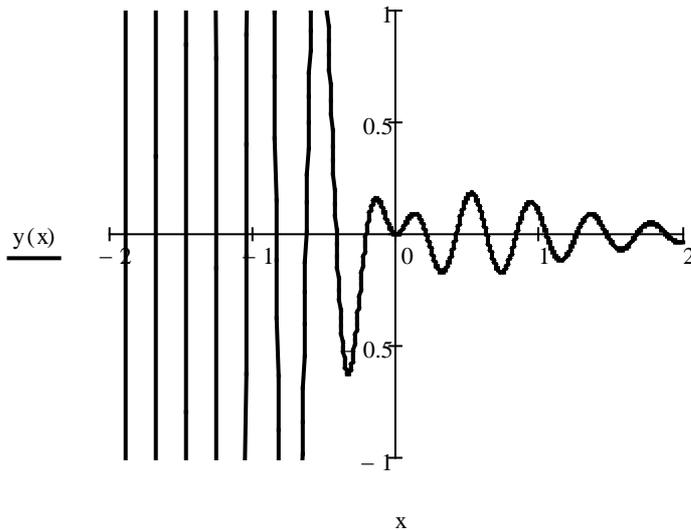


Рис. 5.2. График функции в новом диапазоне  $x$

Найдем глобальный минимум этой функции в пределах изменения аргумента  $0 < x < 3$ , используя метод Монте-Карло. Сформируем два вектора  $X$  и  $Y$ , присвоив их нулевым элементам значение нуль.

Зададимся количеством случайных чисел  $N$ , которые мы будем использовать для вычисления минимума. Чем больше это количество, тем точнее будет результат:

$$X_0 := 0, \quad Y_0 := 0, \quad N := 10000.$$

С помощью функции  $\text{rnd}(x)$  создадим вектор случайных значений элементов  $x_i$ . Функция  $\text{rnd}(x)$  генерирует равномерно распределенные случайные числа в интервале  $0-x$ . На графике видно, что нам достаточен интервал  $0-3$ .

$$i := 1..N, \quad X_i := \text{rnd}(3).$$

Теперь в векторе  $X$  помещено 10000 случайных чисел. Вычислим значения функции от них и поместим их в вектор  $Y$ :

$$Y_i := y(X_i).$$

Величину максимального элемента этого вектора найдем, используя функцию  $\text{max}$ :

$$Y_0 := \text{max}(Y), \quad Y_0 = 0.184 .$$

Мы вычислили (приближенно) максимальное значение заданной функции. Теперь необходимо определить значение аргумента, соответствующее минимальному значению функции. Для этого составим небольшую программку и вычислим по ней ответ:

$$X_0 := \text{for } i \in 1..N$$

$$\left| \begin{array}{l} X_0 \leftarrow X_i \text{ if } Y_0 = Y_i \\ X_0 \end{array} \right. \quad X_0 = 0.5$$

## ЛЕКЦИЯ 6

### ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ

#### 6.1. Основные понятия

Методология генерирования случайных чисел имеет долгую и интересную историю. Самые ранние методы генерирования выполнялись вручную, например: вытягивание жребия, бросание игральных костей, сдача карт или вытаскивание пронумерованных шариков из урны, в которой они предварительно были «тщательно перемешаны». Многие лотереи до сих пор функционируют таким образом. В начале XX в. специалисты по статистике также проявили интерес к случайным числам. Были созданы механизированные устройства для более быстрого генерирования случайных чисел: в конце 1930-х гг. Кендалл и Бабингтон-Смит использовали быстро вращающийся диск для подготовки таблицы, содержащей 100000 случайных однозначных чисел. Позднее были разработаны электрические схемы, основанные на произвольно пульсирующих электронных лампах, которые выдавали до 50 случайных чисел в секунду. Одна из таких схем реализована в электронном индикаторном устройстве для получения случайных чисел (Electronic Random Number Indicator Equipment, ERNIE), использовавшемся Британским почтовым ведомством для выбора победителей в лотерею. Другое электронное устройство применялось компанией Rand Corporation

для генерирования таблицы, содержащей 11000000 случайных цифр. Было изобретено и множество других схем, таких как произвольный выбор чисел из телефонных книг, результатов переписи населения или использование цифр в числе  $\pi$  до 100000 десятичных разрядов. В последние годы также проявляется интерес к созданию и использованию физических устройств для получения случайных чисел, например, Миятаке описывает устройство, основанное на подсчете гамма-излучения.

С распространением компьютеров (и моделирования) все более пристальное внимание стало уделяться методам генерирования, или генераторам случайных чисел, совместимых со способом работы компьютеров. Одна возможность состоит в прямом подключении к компьютеру электронных устройств генерирования случайных чисел, таких как ERNIE. У этого метода есть несколько недостатков, основной заключается в том, что точно воспроизвести ранее сгенерированный поток чисел невозможно.

Поэтому в исследованиях 1940-1950-х гг. ученые обратились к *численным*, или *арифметическим*, способам генерирования «случайных» чисел. Эти методы являются последовательными, каждое новое число определяется одним или несколькими предшествующими числами в соответствии с заданной арифметической формулой. Первый такой арифметический генератор, который был предложен фон Нейманом и Метрополисом в 1940-х гг, известен как *метод серединных квадратов*. Далее приведен пример его использования.

#### ПРИМЕР 6.1

Возведем в квадрат положительное четырехзначное целое число  $Z_0$  и получим целое число из восьми цифр. Если понадобится, добавим нули в левую часть, чтобы в числе было ровно восемь цифр. Возьмем средние четыре цифры из этого восьмизначного числа в качестве следующего четырехзначного числа  $Z_1$ . Поместим десятичную запятую слева от числа  $Z_1$ , чтобы получить первое случайное число с распределением  $U(0,1)$ , а именно

$V_v$ . Затем число  $Z_2$  представим средними четырьмя цифрами  $Z$ , а второе случайное число  $U_2$  получим, разместив десятичную запятую слева от цифр числа  $Z_2$ , и т. д. В табл. 6.1 собраны несколько первых значений для  $Z_0 = 7182$  (первые четыре цифры справа от десятичной запятой в числе  $e$ ).

Таблица 6.1. Метод серединных квадратов

<b>i</b>	<b><math>Z_t</math></b>	<b><math>V_v</math></b>	<b><math>Z_2</math></b>
0	7182	—	51581124
1	5811	0,5811	33767721
2	7677	0,7677	58936329
3	9363	0,9363	87665769
4	6657	0,6657	44315649
5	3156	0,3156	09960336
6	9603	0,9603	92217609
7	2176	0,2176	04734976
8	7349	0,7349	12229009
9	2290	0,2290	05244100
10	2441	0,2441	05958481
11	9584	0,9584	91853056
12	8530	0,853	72760900
13	7609	0,7609	57896881
14	8968	0,8968	80425024
15	4250	0,425	18062500
16	625	0,0625	00390625
17	3906	0,3906	82119844
18	1198	0,1198	01435204
19	4352	0,4352	12390400
20	3904	0,3904	15241216

Может показаться, что метод серединных квадратов обеспечивает хорошую перестановку элементов одного числа для получения следующего и такое случайное правило обеспечивает хороший способ генерирования случайных чисел. В действительности это не совсем так. Одна из серьезных проблем состоит в том, что данный метод имеет стойкую тенденцию возвращать значения, которые стремятся к нулю (продолжите еще немного табл. 6.1 или назначьте  $Z_0 = 1009$ , первые четыре цифры из таблицы Rand Corporation.) Таким образом, нельзя рассчитывать на то, что

хороший генератор случайных чисел можно создать, производя манипуляции с каким-либо одним числом для получения следующего.

Метод серединных квадратов вовсе не является случайным, т.е. непредсказуемым (это наиболее существенный его недостаток). На самом деле, если мы знаем одно число, следующее число является полностью предопределенным, поскольку правило его получения неизменно. Фактически, когда задается  $Z_0$ , предопределяется вся *последовательность* чисел  $Z_t$  и  $U_t$ . Это замечание касается всех арифметических генераторов (далее мы будем рассматривать только данный тип). Поэтому арифметические генераторы именуется *псевдослучайными*.

Типы базовых датчиков:

- физические (любой физической шум) – не используются, так как характеристики нестабильны и реализацию повторить нельзя;
- псевдослучайные – строятся на основе *детерминированного* алгоритма, но полученные результаты неотличимы от случайных.

Псевдослучайные базовые датчики строятся по модели

$$x_{n+1} = f(x_n) \text{ при заданном } x_0.$$

Хороший арифметический генератор случайных чисел должен обладать следующими свойствами.

1. Получаемые числа должны быть равномерно распределены в интервале  $[0, 1]$  и не должны иметь корреляции друг с другом, в противном случае результаты моделирования могут оказаться полностью недействительными.
2. Чтобы генератор можно было использовать на практике, он должен обладать быстродействием и не требовать больших затрат памяти.
3. Генератор должен обеспечивать возможность точно воспроизводить заданный поток случайных чисел. Во-первых, это позволяет упростить отладку компьютерной программы и проверить, правильно ли она работает. Во-вторых, что гораздо важнее, вы

можете использовать *идентичные* случайные числа при моделировании различных систем и выполнить их более точное сравнение.

4. В генераторе должен быть предусмотрен простой способ получения отдельных потоков случайных чисел. Поток - это просто часть последовательности случайных чисел, производимых генератором, очередной поток начинается в том месте, где заканчивается предыдущий. Мы можем рассматривать различные потоки как отдельные и независимые генераторы (при условии, что весь поток, который, как правило, имеет длину 100000 или более чисел, не применяется). Таким образом, пользователь может выделить определенный поток для конкретного источника случайности при моделировании.

Для формирования «истинно» случайных чисел используются аналого-цифровые преобразователи естественных источников случайных шумов (шумы электронных и полупроводниковых устройств, радиоактивный распад и т.п.). Случайные числа, генерируемые на ЭВМ (аппаратно или программно), называются псевдослучайными (ПСЧ). Последовательность псевдослучайных чисел всегда бывает периодической, т.е. в ней можно выделить повторяющиеся фрагменты.

К псевдослучайным числам предъявляются следующие требования:

- 1) псевдослучайные числа должны быть равномерно распределены на интервале  $[0; 1]$ ;
- 2) псевдослучайные числа должны быть независимыми;
- 3) период последовательности псевдослучайных чисел должен быть большим;
- 4) последовательность псевдослучайных чисел должна быть воспроизводимой.

Большинство программных средств содержат встроенные генераторы случайных чисел (ГСЧ). Однако псевдослучайные числа, генерируемые ими, часто не удовлетворяют перечисленным выше требованиям. Поэтому для решения задач компьютерного моделирования используют более надежные ГСЧ.

Генерация произвольного случайного числа состоит из двух этапов:

- 1) генерация случайного числа  $r_i$ , равномерно распределенного в интервале от 0 до 1 (генерация базового СЧ);
- 2) преобразование базовых СЧ  $r_i$  в случайные числа  $x_i$ , которые распределены по заданному закону распределения или в заданном интервале.

## 6.2. Линейные конгруэнтные генераторы

Сегодня очень часто применяются *линейные конгруэнтные генераторы* (ЛКГ), созданные Лемером. В них последовательность целых чисел  $Z_1, Z_2, \dots$  определяется по рекурсивной формуле

$$Z_i = (aZ_{i-1} + c) \bmod m, \quad (6.1)$$

где  $m$  (модуль),  $a$  (множитель),  $c$  (приращение) и  $Z_0$  (начальное число или значение) являются неотрицательными целыми числами. Таким образом, согласно формуле (6.1), для получения  $Z_i$  нужно разделить  $(aZ_{i-1} + c)$  на  $m$  и взять остаток от деления, т.е.  $Z_i$  будет остатком этого деления. Поэтому  $0 \leq Z_i \leq m - 1$ , а чтобы получить искомые случайные числа  $U_i$  (при  $i = 1, 2, \dots$ ) в интервале  $[0, 1]$ , примем  $U_i = Z_i/m$ . Мы концентрируем внимание главным образом на  $Z_i$ , хотя следует иметь в виду и точный характер деления  $Z_i$  на  $m$  в связи с различиями в способах обработки разными компьютерными компиляторами чисел с плавающей запятой. Кроме того, что целые числа  $m, a, c$  и  $Z_0$  должны быть неотрицательными, они должны удовлетворять неравенствам  $0 < m, a < m, c < m$  и  $Z_0 < m$ .

Против использования ЛКГ можно высказать следующие соображения:

**Во-первых**, значения  $Z_{it}$ , определенные по формуле (6.1), вовсе не являются случайными; это касается всех генераторов псевдослучайных чисел. Однако, тщательно подбирая эти четыре параметра, мы стараемся вызвать такое поведение величин  $Z_i$ , при котором соответствующие величины  $U_i$  представляются как независимые и равномерно распределенные случайные величины с распределением  $U(0,1)$ , когда они проверяются с помощью ряда тестов.

**Во-вторых**, при использовании ЛКГ  $Z_i$  могут принимать лишь рациональные значения  $0, 1/m, 2/m, \dots, (m-1)/m$ . В действительности  $Z_i$  могут принимать только значения мантисс этих величин в зависимости от определения констант  $m, a, c$  и  $Z_0$ , а также от характера деления с плавающей запятой на  $m$ . Поэтому у нас нет возможности получить значение  $Z_i$  между, например,  $0,1/m$  и  $0,9/m$ , в то время как  $i$ -я возможность *должна* возникать с вероятностью  $0,8/mc > 0$ . Для модуля  $m$  обычно выбирается очень большое значение, скажем,  $10^9$  или больше, при котором точки в интервале  $[0,1]$ , куда могут попадать значения  $U_t$ , располагаются очень плотно. Для  $m > 10^9$  существует по меньшей мере 1 Млрд возможных значений. Но цикличность неизбежна. Согласно формуле (6.1), как только  $Z_i$  получает значение, которое у нее уже было, сгенерируется та же самая последовательность величин, и этот цикл повторяется бесконечно. Длина цикла называется *периодом* генератора. Для ЛКГ значение  $Z_i$  зависит *только* от предыдущего целого числа  $Z_{i-1}$ , а поскольку  $0 < Z_i < m - 1$ , период генератора не превышает величину  $m$ . Если период действительно равен  $m$ , считается, что ЛКГ имеет *полный период*. Разумеется, если генератор имеет полный период, какое бы ни было выбрано начальное число  $Z_0$  из последовательности  $\{0, 1, \dots, m - 1\}$ , весь цикл будет воспроизведен в некотором порядке. Если же период генератора меньше полного, длина цикла будет зависеть от выбора определенного значения  $Z_0$ , в этом случае мы будем говорить о периоде *начального значения* для этого генератора.

Поскольку в широкомасштабных проектах имитационного моделирования могут использоваться сотни тысяч случайных чисел, очевидно, что для них требуются ЛКГ с длинными периодами. Более того, лучше всего иметь ЛКГ с полным периодом, поскольку тогда у нас есть уверенность, что каждое целое число в интервале от 0 до  $m - 1$  возникнет в каждом цикле только один раз, а это способствует равномерности распределения значений  $U$ . Даже ЛКГ с полным периодом на некоторых участках цикла могут иметь неравномерное распределение. Например, если мы сгенерируем только  $m/2$  последовательных значений  $Z_i$ , могут остаться большие пробелы в последовательности возможных значений  $0, 1, \dots, m - 1$ . Таким образом, полезно знать, как выбирать значения  $m$ ,  $a$  и  $c$  так, чтобы у соответствующего ЛКГ был полный период. Следующая теорема, доказанная Халлом и Добеллом, дает нам определение параметров.

#### Теорема

ЛКГ, определенный по формуле (6.1), имеет полный период тогда и только тогда, когда выполняются три следующих условия:

- а) единственное положительное целое число, на которое без остатка делятся как  $m$ , так и  $c$ , равно 1 ( $c$  является взаимно простым по отношению к  $m$ );
- б) если  $q$  является простым числом (делится только само на себя и на 1), на которое делится  $m$ , то  $a - 1$  делится на  $q$ ;
- в) если  $m$  делится на 4, то  $a - 1$  тоже делится на 4.

Из-за первого условия теоремы 1 ЛКГ работают по-разному с параметром  $c > 0$  (*смешанные* ЛКГ) и с параметром  $c = 0$  (*мультипликативные* ЛКГ).

### 6.3. Смешанные генераторы

При  $c > 0$  выполнение первого условия теоремы 1 возможно, следовательно, мы можем получить полный период  $m$ . Сначала выберем значение  $m$ .

Чтобы получить длинный период и высокую плотность величин  $U_i$  в интервале  $[0, 1]$ , величина  $m$  должна иметь большое значение. Кроме того,

деление на  $m$  для получения остатка в формуле (6.1) — довольно длительная арифметическая операция. Поэтому желательно избежать явного выполнения такого деления. Выбор  $m$ , который является удачным во всех этих отношениях:  $m = 2^b$ , где  $b$  — число битов (двоичных знаков) в слове задействованного компьютера, которые действительно доступны для хранения данных. В частности, во многих компьютерах и компиляторах используются 32-битовые слова, при этом крайний левый бит является знаковым, следовательно,  $b = 31$ . Если  $b$  имеет достаточно большое значение, например,  $b > 31$ , тогда  $m > 2^{31} > 2,1 \cdot 10^9$ . Кроме того, выбирая  $m = 2^b$  мы действительно избегаем явного деления на  $m$  для большинства компьютеров, поскольку можно воспользоваться *переполнением целых чисел*. Наибольшее целое число, которое может быть представлено, равно  $2^b - 1$ , и любая попытка сохранить большее целое число  $W$  (имеющее, например,  $h > b$  двоичных знаков) завершится потерей левых (наиболее значимых)  $h - b$  двоичных знаков целого числа, превысившего допустимый размер. Оставшиеся  $b$  двоичных знаков составляют ровно  $W \bmod 2^b$ .

Как же следует выбирать значения  $a$  и  $c$ , когда  $m = 2^b$ , чтобы получить хороший смешанный ЛКГ? Опыт показывает, что лучше отказаться от использования смешанного ЛКГ вообще. Более простые и понятные мультипликативные ЛКГ зарекомендовали себя не хуже смешанных и применяются гораздо чаще.

#### 6.4. Мультипликативные генераторы

Мультипликативные ЛКГ выгодно применять, так как для них не требуется определять параметр  $c$ , но у них не может быть полного периода, потому что первое условие теоремы 1 для них выполняться не будет (поскольку, например,  $m$  является положительным числом, и как  $m$ , так и  $c=0$  делятся на него без остатка). Можно получить период  $m-1$ , если

осторожно подбирать значения  $m$  и  $a$ . Мультипликативные ЛКГ появились еще до смешанных и исследовались более интенсивно. Большинство ЛКГ, применяемых сегодня, являются мультипликативными, поскольку факт улучшения эффективности, на который возлагались надежды в связи с введением смешанных генераторов, окончательно не доказан.

Как и при использовании смешанных ЛКГ, для вычислений по-прежнему подходит выбор  $m=2^b$ , таким образом мы избегаем явного деления. Однако можно доказать, что в этом случае период составляет самое большее  $2^{b-2}$ , т.е. лишь четвертая часть целых чисел от  $0$  до  $m - 1$  может быть получена в качестве значений переменных  $Z_i$ . Фактически период составляет  $2^{b-2}$ , если значение  $Z_0$  является нечетным, а параметр  $a$  имеет вид  $8k + 3$  или  $8k + 5$  для некоторых значений  $k = 0, 1, \dots$ . Кроме того, нам, как правило, неизвестно, куда попадут эти  $m/4$  целых числа, т.е. между полученными переменными  $Z$  могут быть недопустимо большие промежутки. К тому же, если для параметра  $a$  мы выбираем вид  $2^l + j$  (так что умножение  $Z_{i-1}$  на  $a$  заменяется смещением и прибавлением  $j$ ), можно получить неудовлетворительные статистические свойства. Известен, например, генератор RANDU, который имеет такие значения параметров:  $m = 2^{31}$ ,  $a = 2^{16} + 3 = 65\,539$ ,  $c = 0$  - и, как оказалось, очень неподходящие статистические свойства, поэтому следует избегать его применения. Даже отказаться от выбора  $a=2^l+j$ , используя  $m=2^b$ , - тоже не самый подходящий вариант параметра для мультипликативных ЛКГ, хотя бы потому, что период  $m/4$  короче, и по этой причине возникает вероятность появления пропусков в значениях  $Z$ .

Из-за трудностей, возникавших в связи с выбором  $m=2^b$  в мультипликативных ЛКГ, основное внимание уделялось поиску других способов определения значения  $m$ . Метод, оказавшийся довольно успешным, был представлен Хатчинсоном, который приписывает

авторство идеи Лемеру. Вместо того, чтобы использовать  $m=2^b$ , было предложено определять значение  $m$  как наибольшее простое число, которое меньше  $2^b$ . Например, если  $b=31$ , то наибольшее простое число, которое меньше  $2^{31}$ , соответственно будет составлять  $2^{31} - 1 = 247483647$ . Теперь для простого числа  $m$  можно показать, что период составляет  $m-1$ , если  $a$  - это *первообразный элемент по модулю  $m$* , т.е. наименьшее целое число  $l$ , для которого  $a^l - 1$  делится на  $m$ , составляет  $l = m - 1$ . Если таким образом выбрать значения  $m$  и  $a$ , то можно получить каждое целое число  $1, 2, \dots, m - 1$  один раз в каждом цикле, так что  $Z_0$  может быть любым целым числом от 1 до  $m-1$ , а в результате все равно будет получен период  $m-1$ . Такие генераторы называются *мультипликативными ЛКГ с простым модулем*.

## ЛЕКЦИЯ 7

### КЛЕТОЧНЫЕ АВТОМАТЫ

#### 7.1. Введение

Идея клеточных автоматов появилась в конце 1940-х гг. Она была задумана и сформулирована Джоном фон Нейманом и Конрадом Цусе независимо друг от друга как универсальная вычислительная среда для построения, анализа и сравнения характеристик алгоритмов.

Нейман отмечал, что клеточные автоматы являются дискретными динамическими системами, поведение которых полностью определяется в терминах локальных зависимостей. В значительной степени так же обстоит дело для большого класса непрерывных динамических систем, определенных уравнениями в частных производных. В этом смысле клеточные автоматы в информатике являются аналогом физического понятия «поля».

Клеточный автомат может мыслиться как стилизованный мир.

Пространство представлено равномерной сеткой, каждая ячейка или клетка которой содержит несколько битов данных; время идет вперед дискретными шагами, а законы мира выражаются единственным набором правил, скажем, небольшой справочной таблицей, по которой любая клетка на каждом шаге вычисляет своё новое состояние по состояниям её близких соседей. Таким образом, законы системы являются локальными и повсюду одинаковыми. «Локальный» означает, что для того, чтобы узнать, что произойдет здесь мгновение спустя, достаточно посмотреть на состояние ближайшего окружения: никакое дальноедействие не допускается. «Одинаковость» означает, что законы везде одни и те же: я могу отличить одно место от другого только по форме ландшафта, а не по какой-то разнице в законах.

Следует отметить, что клеточные автоматы – это не просто машины, работающие с разбитым на клетки полем. Область применения клеточных автоматов почти безгранична: от простейших «крестиков-ноликов» до искусственного интеллекта. Тема клеточных автоматов очень актуальна, так как может привести к разгадкам многих вопросов в окружающем мире. Создатель игры «Жизнь» Конуэй считал, что нашу Вселенную можно представить клеточным автоматом, который управляет движением элементарных частиц в соответствии с некоторыми правилами (кстати, Конуэй придумал еще одно применение клеточных автоматов в этой области: представим себе достаточно большое количество «первичного бульона» из хаотически распределенных клеток, если можно ожидать появления из такого хаоса структур, способных самовоспроизводить себя, то это еще одно подтверждение теории зарождения жизни на Земле).

Клеточные автоматы используются для моделирования гидродинамических и газодинамических течений. Уравнения гидродинамики соответствуют математической модели, описывающей поведение решетчатого газа, одного из клеточных автоматов, на

макроуровне. Структуры, возникающие в этих клеточных автоматах, похожи на возмущение поведения поверхности потока жидкости механическим препятствием. Примитивные одномерные клеточные автоматы могут моделировать процесс горения различного характера. В настоящее время теория клеточных автоматов наиболее перспективно прилагается к вопросу о разработке самовосстанавливаемых электронных цепей.

Клеточные автоматы применимы не только в математике и физике, а также в биологии, экономике, социологии, информатике и т. д. С помощью клеточных автоматов успешно решались задачи моделирования течений со свободной границей, распространения тепловых потоков, роста и динамики доменов, роста дендритов, описания движения толпы. Их можно использовать при составлении генетических алгоритмов. Представим себе клеточный автомат, для клеток которого дополнительным условием выживания является выработка некоторой последовательности выходных данных (назовем ее условно реакцией) в ответ на последовательность входных данных (раздражение, являющееся свойством среды), предсказывающее последующее состояние среды. Чтобы такой автомат функционировал, добавляется также механизм случайного изменения правил выработки реакции (мутации) и передачи вновь возникающим клеткам информации о правилах реагирования соседей (наследования). Помимо исследования условий развития моделей живых систем, такой подход позволяет решать и некоторые практические задачи, в частности поиск кратчайшего пути на графе. Структура графа кодируется некоторым образом в хромосомах клеток. Предполагается, что алгоритмы, приобретенные вследствие мутаций и наследования, будут соответствовать решениям задачи.

По своему поведению клеточные автоматы делятся на четыре класса. К первому классу относятся автоматы, приходящие через определенное

время к устойчивому однородному состоянию. Автоматы второго класса через некоторое время после пуска генерируют стационарные или периодические во времени структуры. В автоматах третьего класса по прошествии некоторого времени перестает наблюдаться корреляция процесса с начальными условиями. Наконец, поведение автоматов четвертого класса сильно определяется начальными условиями, и с их помощью можно генерировать весьма различные шаблоны поведения. Такие автоматы являются кандидатами на прототип клеточной вычислительной машины. В частности, с помощью специфических клеточных конфигураций игры «Жизнь», которая как раз и является автоматом четвертого типа, можно построить все дискретные элементы цифрового компьютера.

Отметим еще одно применение клеточных автоматов в информатике – шифрование и сжатие данных.

Клеточные автоматы применимы при реализации эффективной системы распознавания образов. Один из возможных путей ее создания – построение динамической системы, аттракторами которой в ее конфигурационном пространстве были бы типичные картины-образы. Начальные условия всегда окажутся в области притяжения одной из картин, с течением времени система трансформирует начальные параметры, приводя их к наиболее близкой структуре – аттрактору, т.е. произойдет автоматическое распознавание образа. Причем можно создавать обучающиеся системы распознавания, в них законы эволюции имеют состояние программирования. Использовать клеточные автоматы можно и при решении оптимизационных задач. Часто в различных сферах деятельности возникают задачи нахождения оптимального варианта из неограниченного числа возможных. Точного решения, как правило, не требуется, но дискретный компьютер не способен даже приблизительно дать оптимальный результат.

Таким образом, клеточные автоматы нашли и находят широкое применение во многих сферах человеческой деятельности, многие задачи которых стало возможным решить только с помощью компьютера. Рассмотрим несколько примеров клеточных автоматов.

## 7.2. Игра «Жизнь»

Наверное, наиболее известным из них можно считать клеточный автомат под названием игра «Жизнь». Создана игра «Жизнь» была в 1970 г. Джоном Хортоном Конуэем, математиком Кембриджского университета. Возникающие в процессе игры ситуации очень похожи на реальные процессы, происходящие при зарождении, развитии и гибели колонии живых организмов. По этой причине «Жизнь» можно отнести к быстро развивающейся в наши дни категории игр, имитирующих процессы, происходящие в живой природе.

Рассматривается бесконечная плоская решётка квадратных ячеек-клеток. Время в этой игре дискретно ( $t=0, 1, 2, \dots$ ). Клетка может быть живой или мёртвой. Изменение её состояния в следующий момент времени  $t+1$  определяется состоянием её соседей в момент времени  $t$  (соседей у клетки восемь, четверо имеют с ней общие рёбра, а четверо только вершины). Основная идея состоит в том, чтобы, начав с некоего расположения клеток, проследить за эволюцией исходной позиции под действием «генетических законов» Конуэя, которые управляют рождением, гибелью и выживанием клеток. Конуэй тщательно подбирал свои правила и долго проверял их на «практике», добиваясь, чтобы они по возможности удовлетворяли трём условиям:

- Не должно быть ни одной исходной конфигурации, для которой существовало бы простое доказательство возможности неограниченного роста популяции.
- В то же время, должны существовать такие начальные конфигурации,

которые заведомо обладают способностью беспредельно развиваться.

- Должны существовать простые начальные конфигурации, которые в течение значительного промежутка времени растут, претерпевают различные изменения и заканчивают свою эволюцию одним из следующих трёх способов: полностью исчезают (либо из-за перенаселенности, т.е. слишком большой плотности живых клеток, либо наоборот, из-за разреженности клеток, образующих конфигурацию); переходят в устойчивую конфигурацию и перестают изменяться вообще или же, наконец, выходят на колебательный режим, т.е. бесконечный цикл превращений с определенным периодом.

Следствием этих требований явились следующие правила игры «Жизнь»:

1. **Выживание.** Каждая клетка, имеющая две или три соседние живые клетки, выживает и переходит в следующее поколение.
2. **Гибель.** Каждая клетка, у которой больше трёх соседей, погибает из-за перенаселённости. Каждая клетка, вокруг которой свободны все соседние клетки или же занята всего одна клетка, погибает от одиночества.
3. **Рождение.** Если число занятых клеток, с которыми граничит какая-нибудь пустая клетка, в точности равно трём, то на этой клетке происходит рождение нового организма.

Зададимся вопросами: Какие основные типы структур (т.е. конфигураций, определяющих поведение сообществ на больших периодах) могут существовать в такой системе? Каковы здесь законы организации структур? Могут ли они взаимодействовать, и к чему это приводит? Выясним, какие закономерности являются следствиями представленных выше правил.

Первая закономерность – свойство локализации – структуры, разделённые двумя «мёртвыми» (пустыми) клетками, не влияют друг на друга.

Вторая закономерность – система клеток, которую описывает игра «Жизнь», развивается необратимо. В самом деле, конфигурация в момент времени  $t$  полностью определяет будущее (состояние в моменты  $t+1$ ,  $t+2$  и так далее). Но восстановить прошлое системы по её настоящему не удаётся. Картина здесь такая же, как в одномерных отображениях, только прообразов у данной конфигурации может быть бесконечно много. Докажем это утверждение: воспользуемся свойством локализации и расположим вокруг данной конфигурации множество локализованных одиночных клеток или их пар так, чтобы они не влияли на неё и друг на друга. Понятно, что все они исчезнут на следующем шаге, никоим образом не повлияв на будущее системы. Здесь мы можем заметить признаки нелинейных диссипативных структур: эти структуры определяли поведение системы при  $t$ , стремящемся к бесконечности в случае различных начальных данных.

Третья закономерность – как показывают длительные наблюдения за процессом развития колоний, конфигурации, не обладавшие в начале игры симметрией, обнаруживают тенденцию к переходу в симметричные формы. Обретенные свойства симметрии в процессе дальнейшей эволюции не утрачиваются, симметрия конфигурации может лишь обогащаться.

Условимся классифицировать конфигурации клеток по следующим параметрам:

- По количеству клеток в комбинации: единичная клетка, дуплет (2 клетки в комбинации), триплет (3 клетки) и т.д.
- По перспективе развития: развивающиеся (неограниченный рост), стабильные (количество клеток в популяции колеблется около какого-то среднего значения), вымирающие (популяция стабильно уменьшается), периодические (количество клеток принимает несколько фиксированных значений через определенный период).

Теперь рассмотрим типичные структуры, появляющиеся в игре

«Жизнь». Простейшими являются стационарные, т.е. не зависящие от времени структуры (сам Конуэй называет их «любителями спокойной жизни»). Их примеры показаны на рис. 7.1. С помощью этих стационарных структур можно получить множество других. В самом деле, если мы имеем такую структуру, то конфигурация, полученная поворотом на  $90^\circ$ , также будет стационарной. Конфигурации в нижних рядах показывают, как можно достраивать определённые структуры до любых размеров.

Используя свойство локализации, можно строить «большие» стационарные структуры из «малых» - элементарных.

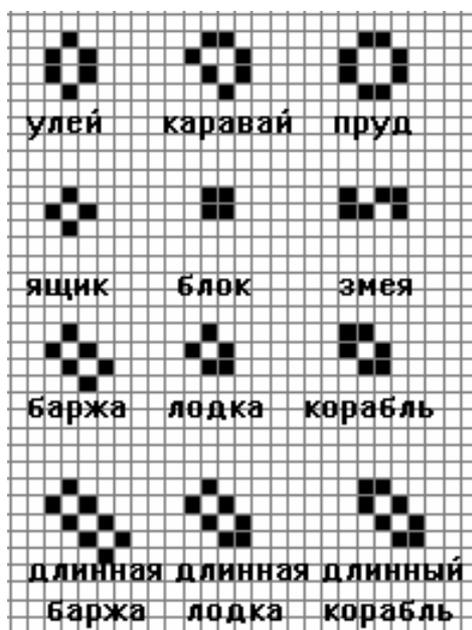
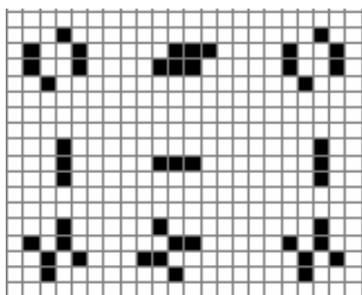


Рис. 7.1. Примеры стационарных структур, реализующихся в игре «Жизнь»

Можно считать, что стационарные структуры повторяют себя на каждом шаге по времени. Но есть и другие конфигурации, повторяющие себя через  $N$  шагов, так называемые  $N$ -циклы (периодические структуры).

Примеры 2-циклов показаны на рис. 7.2. При эволюции различных сообществ часто встречается 2-цикл, показанный во второй строке и называемый «семафором».



*Рис. 7.2. Примеры периодических структур (2-циклы), реализующихся в игре «Жизнь» (жаба, семафор, часы)*

Известно много различных периодических конфигураций. Однако эффективные алгоритмы, позволяющие строить различные конфигурации с данным периодом  $N$ , по-видимому, в настоящее время не созданы.

Эволюция взятых наугад начальных данных часто приводит к возникновению простейших локализованных структур (показанных на рис. 7.1) и семафоров. Однако возможны и более сложные типы эволюции, например, когда сообщество клеток симметрично «достраивается», и возникают циклы большого периода, имеющие сложную форму.

В игре «Жизнь» существуют конфигурации, которые могут передвигаться по плоскости. Одной из них является «планер» (или «глайдер») – конфигурация из 5 клеток (рис. 7.3).



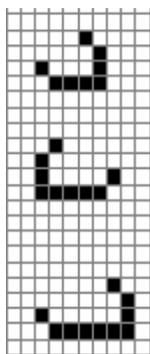
*Рис. 7.3. Планер (глайдер) - перемещающаяся конфигурация из 5 клеток*

После второго хода планер немного сдвигается и отражается относительно диагонали. В результате двух последующих ходов планер «выходит из пике», ложится на прежний курс и сдвигается на одну клетку вправо и одну клетку вниз относительно начальной позиции.

Скорость, с которой шахматный король перемещается по доске в любом направлении, Конуэй называет «скоростью света». Выбор Конуэя пал именно на этот термин из-за того, что в изобретённой им игре большие скорости просто не достигаются. Ни одна конфигурация не воспроизводит

себя достаточно быстро, чтобы двигаться с такой скоростью. Конуэй доказал, что максимальная скорость на диагонали составляет одну четверть скорости света. Поскольку планер переходит сам в себя после четырёх ходов и при этом опускается на одну клетку по диагонали, то говорят, что он скользит по полю со скоростью, равной одной четвертой скорости света. Конуэй также показал, что скорость любой конечной фигуры, перемещающейся по вертикали или по горизонтали на свободные клетки, не может превышать половину скорости света. (Скорость движения равна дроби, в числителе которой стоит число ходов, необходимых для воспроизведения фигуры, а в знаменателе – число клеток, на которое она при этом смещается). Понятно, что в силу симметрии есть планеры, распространяющиеся вдоль любой диагонали квадрата в обоих направлениях.

Впрочем, некоторые конфигурации могут передвигаться не вдоль диагоналей, а по вертикальным и горизонтальным прямым. Таковы, например, три «корабля» показанные на рис. 7.4. (Обратим внимание на то, что далеко не любая конфигурация такого типа будет кораблём). Кстати, планер является кораблём легчайшего веса. Во время движения из кораблей возникают «искры», которые тут же гаснут при дальнейшем движении кораблей.



*Рис. 7.4. Корабли – конфигурации, реализующиеся в игре «Жизнь», способные перемещаться*

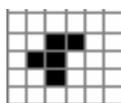
Одиночные корабли без эскорта не могут занимать в длину больше шести клеток, в противном случае на поле начинают появляться различные мелкие фигуры, препятствующие движению корабля. Конуэй обнаружил,

что более длинным кораблям (которые он называл «сверхтяжёлыми») необходим эскорт из двух или большего числа кораблей меньших размеров. Корабли эскорта не дают возникнуть препятствиям на пути сверхтяжёлого корабля. Конуэй вычислил, что корабль длиной в сто клеток требует эскорта, состоящего из тридцати трёх (!) кораблей.

Итак, мы располагаем планерами и различными кораблями. Возникает вопрос, что происходит, когда они сталкиваются между собой или с различными стационарными конфигурациями (стационарами). Столкновения могут быть очень разнообразны в зависимости от курса планера и его фазы при столкновении. Столкновение двух планеров или планера со стационаром может приводить к их «аннигиляции». В столкновении может рождаться целый набор семафоров и стационаров.

Обратим внимание на следующую закономерность. Если конфигурация все время локализована в квадрате размером  $N \times N$ , то она является набором стационаров и циклов, период которых не превышает  $2^N$ . В самом деле, каждая клетка может находиться в одном из двух состояний, а всего клеток в области  $N^2$ , поэтому при  $t > 2^N$  конфигурации начнут повторяться.

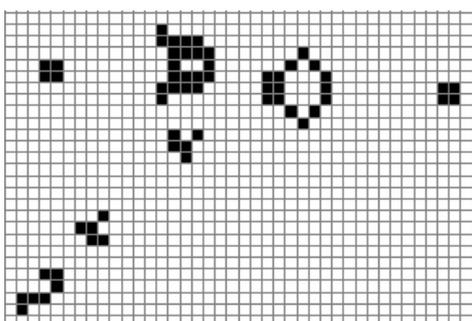
Рассматривая непрерывные среды, можно говорить о резонансном возбуждении – начальных данных, приводящих к более сложной эволюции решений, чем в остальных случаях. В игре «Жизнь» есть аналог такого поведения. Обратим внимание на конфигурацию, показанную на рис. 7.5, называемую « $r$ -пентамино». Возникающие клетки занимают всё большую часть плоскости, рождается несколько планеров, причем это сообщество будет развиваться далее. Ни одна из других конфигураций, состоящих из пяти клеток, не приводит к такому сложному поведению.



*Рис. 7.5. Конфигурация из 5 клеток, демонстрирующая эффект резонансного возбуждения ( $r$ -пентамино)*

Как правило, эволюция взятых наугад конфигураций приводит к появлению наборов стационаров, семафоров, планеров. При этом общее число клеток при  $t$ , стремящемся к бесконечности, оказывается ограниченным (это было заложено в требованиях Конуэя), однако, при некоторых начальных данных эволюция системы может качественно меняться. Такое поведение характерно для ряда биологических систем, в частности, эволюционных процессов. Маловероятное событие может качественно изменить поведение системы, привести к появлению новых видов. Именно поэтому игра «Жизнь» находит применение в экологических моделях, при моделировании морфогенеза, в других биологических задачах.

Чем большую площадь занимает сообщество, тем сложнее оно может себя вести. Поэтому большой интерес вызывают неограниченно растущие в пространстве конфигурации. Одну из них, называемую «катапультой» или «планерным ружьём», предложил в 1970 г. Р. Госпер-младший. Видно, что катапульта через каждые 30 шагов повторяет себя и выпускает планер (рис. 7.6). Планерное ружьё заполняет пространство потоком планеров. Конуэй высказал гипотезу, согласно которой не существует ни одной начальной конфигурации, способной беспредельно расти. Иначе говоря, любая конфигурация, состоящая из конечного числа живых клеток, не может перейти в конфигурацию, в которой число живых клеток превосходило бы некий конечный предел. Но гипотеза оказалась ошибочной! Опровержение – планерное ружьё.



*Рис. 7.6. Планерное ружьё (катапульта) – конфигурация, генерирующая за каждые 30 ходов планер*

Есть ещё более сложные сообщества клеток, которые могут двигаться, оставляя за собой большой набор семафоров и стационаров. Одно из них — «паровоз» (он имеет довольно сложную структуру). Поиск таких конфигураций – довольно трудоёмкий процесс, требующий применения специальных алгоритмов, и он под силу лишь квалифицированным специалистам.

Также были найдены особые конфигурации, которые Джон Тьюки назвал «садами Эдема». «Сады Эдема» не могут возникнуть в ходе работы клеточного автомата, поскольку их не способна породить никакая конфигурация. В силу этого они должны быть заданы с самого начала – в нулевом поколении. Не имея предшественников, они не могут быть самовоспроизводящимися. Алви Р. Смит нашел способ применить теорему Мура к игре Конуэя и показал, что конфигурация по типу «садов Эдема» возможна и в «Жизни». Формулы, выведенные Муром, позволили Смиуту утверждать, что подобная конфигурация всегда может быть заключена в квадрат со стороной в 10 000 000 000 клеток.

Приведённые примеры показывают, что в обсуждаемой дискретной системе существует большое количество различных типов упорядоченности, которые определяют асимптотическое поведение некоторого множества конфигураций (в этом смысле они оказываются эквивалентны аттракторам динамических систем). Однако можно доказать большее – в игре «Жизнь» существуют сколь угодно сложные типы упорядоченности, эта дискретная система оказывается эквивалентна универсальной вычислительной машине.

ЭВМ можно рассматривать как конечный набор простейших логических элементов, осуществляющих операции И, ИЛИ, НЕ, определённым образом соединённых проводами, по которым распространяется набор импульсов, кодирующих последовательность

нулей и единиц. В качестве генератора таких импульсов в игре «Жизнь» выступает планерное ружьё. Наличие планера в потоке естественно интерпретировать как единицу, отсутствие - как ноль. Столкновение планеров, приводящее к их аннигиляции, позволяет построить элемент НЕ, направив два потока под прямым углом (если планер в определённом месте есть в первом потоке, то после столкновения планер в другом потоке на этом месте исчезнет). Более сложным образом конструируются другие элементы.

Для анализа ситуаций, возникающих в игре «Жизнь», применяется компьютер. В программе, моделирующей этот клеточный автомат, используется квадратная матрица, которая и является полем для игры. При смене хода анализируется каждый элемент старой матрицы и строится на её основе новая, которая соответствует конфигурации на следующем шаге эволюции. Для более детального исследования игра Конуэя расширена на несколько популяций, каждая из которых развивается по своим правилам. Правила для каждой популяции выбираются из следующих:

- Условия рождения и смерти. Задаются четыре параметра (параметры можно менять в процессе игры): минимальное и максимальное количество соседей своей популяции, при котором рождается клетка; минимальное и максимальное количество соседей, при котором клетка выживает и переходит в следующее поколение.
- Соседями клетки могут быть любые клетки, находящиеся в квадрате 3x3 с центром в данной клетке.

Еще один клеточный автомат, в котором за кажущейся простотой скрывается очень интересный объект. Идея довольно проста – полем для автомата служит кольцо толщиной в одну клетку. Следующее поколение получается из предыдущего и отображается под всей структурой. Таким образом, мы имеем плоскость, по одной оси которой единственная

пространственная координата, а по другой – время, в результате чего мы можем просмотреть всю эволюцию популяции.

Правила автомата довольно просты – они похожи на правила «Жизни» в одномерном случае.

- если над исследуемой клеткой количество соседей равно 3, клетка «рождается»;
- если над клеткой соседей меньше 2, то она «умирает».

В процессе развития такой популяции получается очень интересный узор – «салфетка Серпинского».

Салфетка Серпинского – фрактал, который строится следующим образом. Правильный треугольник делим средними линиями на четыре равных треугольника и внутренность центрального выбрасываем. С тремя оставшимися делаем то же самое и так до бесконечности. После счётного числа выбрасываний остаётся множество  $S$ , называемое салфеткой Серпинского. На рис. 7.7 представлена салфетка на третьем шаге.

#### Программа на MathCad моделирования салфетки Серпинского

$$V := \begin{pmatrix} 1 \\ 0 \\ \frac{1}{2} + i \cdot \frac{\sqrt[3]{3}}{2} \\ 0 \end{pmatrix}$$

$$U(u,v,w) := \text{augment}(u, \text{augment}(v, w))$$

$$T(x) := \frac{1}{2} \cdot U(x, x+V_1, x+V_2)$$

```

Sp(n):=
  x0 ← V
  for i ∈ 1..n
    xi ← x0
    tmp ← x0
    yi ← T(xi)
    x0 ← yi
  tmp ← augment(tmp, yi)
  tmp

```

```

n := 6
Y := Im(Sp(n))
X := Re(Sp(n))

```

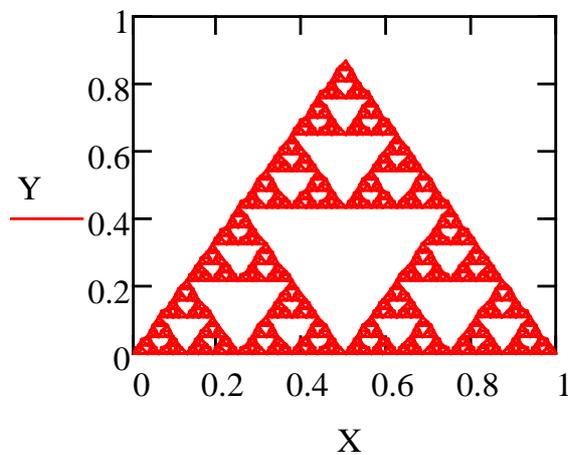


Рис.7.7. Салфетка Серпинского

## ЛЕКЦИЯ 8

### ФРАКТАЛЫ

#### 8.1. Понятие "фрактал"

Понятия **фрактал** и **фрактальная геометрия**, появившиеся в конце 1970-х гг., с середины 1980-х гг. прочно вошли в обиход математиков и программистов. Слово **фрактал** образовано от латинского **fractus** и в переводе означает *состоящий из фрагментов*. Оно было предложено Бенуа Мандельбротом в 1975 г. для обозначения нерегулярных, но самоподобных структур, которыми он занимался. Рождение фрактальной геометрии принято

связывать с выходом в 1977 г. книги Мандельброта «*The Fractal Geometry of Nature*». В его работах использованы научные результаты других ученых, работавших в период 1875-1925 гг. в той же области (Пуанкаре, Фату, Жюлиа, Кантор, Хаусдорф). Но только в наше время удалось объединить их работы в единую систему.

Роль фракталов в машинной графике сегодня достаточно велика. Они приходят на помощь, например, когда требуется с помощью нескольких коэффициентов задать линии и поверхности очень сложной формы. С точки зрения машинной графики, фрактальная геометрия незаменима при генерации искусственных облаков, гор, поверхности моря. Фактически найден способ легкого представления сложных неевклидовых объектов, образы которых весьма похожи на природные.

Одним из основных свойств фракталов является самоподобие. В самом простом случае небольшая часть фрактала содержит информацию о всем фрактале.

Определение фрактала, данное Мандельбротом, звучит так: «*Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому*».

## **8.2. Классификация фракталов**

Чтобы представить все многообразие фракталов, удобно прибегнуть к их общепринятой классификации.

### ***1. Геометрические фракталы***

Фракталы этого класса самые наглядные. В двухмерном случае их получают с помощью некоторой ломаной (или поверхности в трехмерном случае), называемой *генератором*. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную-генератор, в соответствующем масштабе. В результате бесконечного повторения этой процедуры получается геометрический фрактал.

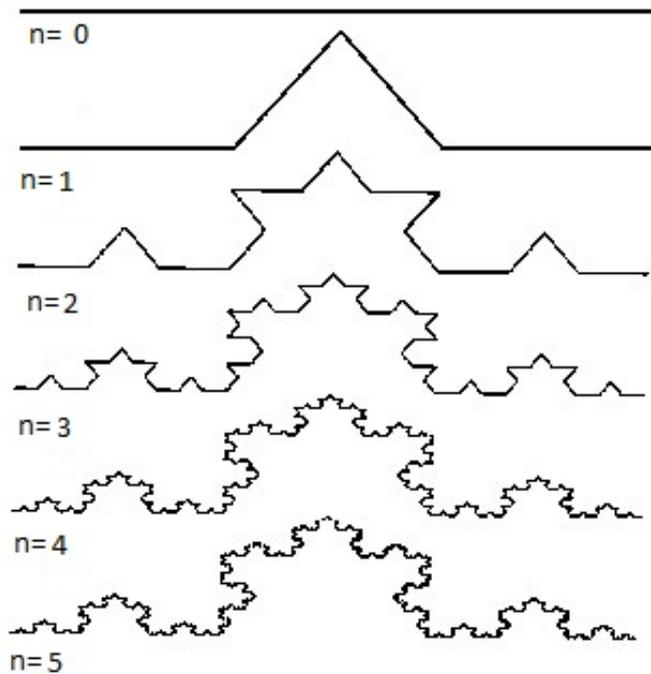


Рис 8.1. Построение триадной кривой Коха

Рассмотрим один из таких фрактальных объектов - триадную кривую Коха. Построение кривой начинается с отрезка единичной длины (рис.8.1) - это 0-е поколение кривой Коха. Далее каждое звено (в нулевом поколении один отрезок) заменяется на *образующий элемент*, обозначенный на рис. 8.1 через  $n=1$ . В результате такой замены получается следующее поколение кривой Коха. В 1-м поколении - это кривая из четырех прямолинейных звеньев, каждое длиной по  $1/3$ . Для получения 3-го поколения проделываются те же действия - каждое звено заменяется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения, все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. Кривая  $n$ -го поколения при любом конечном  $n$  называется *предфракталом*. На рис.8.1 представлены пять поколений кривой. При  $n$ , стремящемся к бесконечности, кривая Коха становится фрактальным объектом.

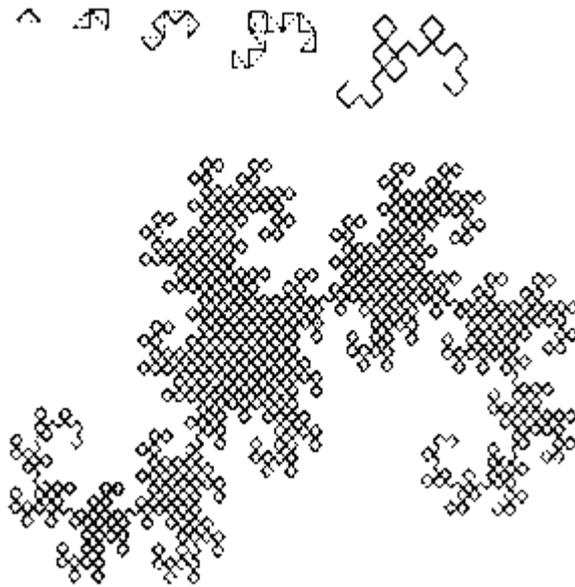


Рис 8.2. Построение дракона Хартера-Хейтуэя

Для получения другого фрактального объекта нужно изменить правила построения. Пусть образующим элементом будут два равных отрезка, соединенных под прямым углом. В нулевом поколении заменим единичный отрезок на этот образующий элемент так, чтобы угол был сверху. Можно сказать, что при такой замене происходит смещение середины звена. При построении следующих поколений выполняется правило: самое первое слева звено заменяется на образующий элемент так, чтобы середина звена смещалась влево от направления движения, а при замене следующих звеньев направления смещения середин отрезков должны чередоваться. На рис.8.2 представлены несколько первых поколений и 11-е поколение кривой, построенной по вышеописанному принципу. Предельная фрактальная кривая (при  $n$ , стремящемся к бесконечности) называется *драконом Хартера-Хейтуэя*.

В машинной графике использование геометрических фракталов необходимо при получении изображений деревьев, кустов, береговой линии. Двухмерные геометрические фракталы используются для создания объемных текстур (рисунка на поверхности объекта) .

## 2. Алгебраические фракталы

Это самая крупная группа фракталов. Получают их с помощью нелинейных процессов в  $n$ -мерных пространствах. Наиболее изучены двумерные процессы. Интерпретируя нелинейный итерационный процесс как дискретную динамическую систему, можно пользоваться терминологией теории этих систем: *фазовый портрет, установившийся процесс, аттрактор* и т.д.

Известно, что нелинейные динамические системы обладают несколькими устойчивыми состояниями. То состояние, в котором оказалась динамическая система после некоторого числа итераций, зависит от ее начального состояния. Поэтому каждое устойчивое состояние (или как говорят - аттрактор) обладает некоторой областью начальных состояний, из которых система обязательно попадет в рассматриваемые конечные состояния. Таким образом, фазовое пространство системы разбивается на *области притяжения* аттракторов. Если фазовым является двумерное пространство то, окрашивая области притяжения различными цветами, можно получить *цветовой фазовый портрет* этой системы (итерационного процесса). Меняя алгоритм выбора цвета, можно получить сложные фрактальные картины с причудливыми многоцветными узорами. Неожиданностью для математиков стала возможность с помощью примитивных алгоритмов порождать очень сложные нетривиальные структуры.

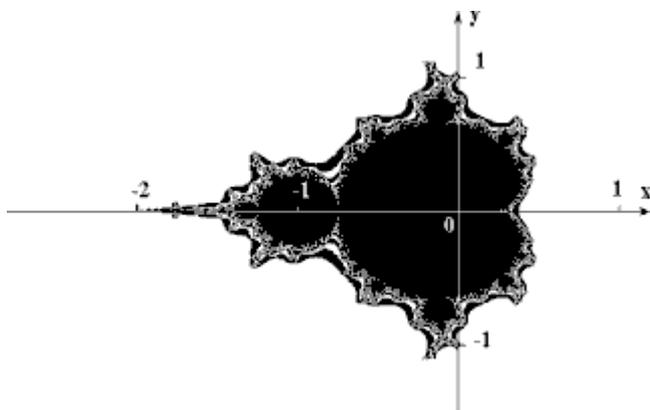


Рис 8.3. Множество Мандельброта

В качестве примера рассмотрим множество Мандельброта (рис.8.3 и 8.4). Алгоритм его построения достаточно прост и основан на простом итеративном выражении:

$$Z[i+1] = Z[i] * Z[i] + C,$$

где  $Z[i]$  и  $C$  - комплексные переменные.

Итерации выполняются для каждой стартовой точки  $C$  прямоугольной или квадратной области - подмножестве комплексной плоскости. Итерационный процесс продолжается до тех пор, пока  $Z[i]$  не выйдет за пределы окружности радиуса 2, центр которой лежит в точке  $(0,0)$  (это означает, что аттрактор динамической системы находится в бесконечности), или после достаточно большого числа итераций (например, 200-500)  $Z[i]$  сойдется к какой-нибудь точке окружности. В зависимости от количества итераций, в течение которых  $Z[i]$  оставалась внутри окружности, можно установить цвет точки  $C$  (если  $Z[i]$  остается внутри окружности в течение достаточно большого количества итераций, итерационный процесс прекращается и эта точка раstra окрашивается в черный цвет).

Вышеописанный алгоритм дает приближение к так называемому множеству Мандельброта. Множеству Мандельброта принадлежат точки, которые в течение *бесконечного* числа итераций не уходят в бесконечность (точки, имеющие черный цвет). Точки, принадлежащие границе множества (именно там возникают сложные структуры), уходят в бесконечность за конечное число итераций, а точки, лежащие за пределами множества, уходят в бесконечность через несколько итераций (белый фон).

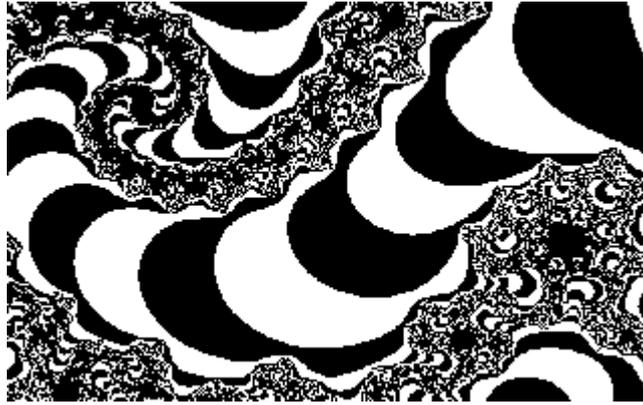


Рис 8.4. Участок границы множества Мандельброта, увеличенный в 200 раз

### **3. Стохастические фракталы**

Еще одним известным классом фракталов являются стохастические фракталы, которые получаются в том случае, если в итерационном процессе случайным образом менять какие-либо его параметры. При этом получаются объекты, очень похожие на природные, - несимметричные деревья, изрезанные береговые линии и т.д. Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.

### **8.3. Системы итерируемых функций**

Метод "Систем Итерируемых Функций" (Iterated Functions System - IFS) появился в середине 1980-х гг. как простое средство получения фрактальных структур.

IFS представляет собой систему функций из некоторого фиксированного класса функций, отображающих одно многомерное множество на другое. Наиболее простая IFS состоит из аффинных преобразований плоскости:

$$X' = A * X + B * Y + C$$

$$Y' = D * X + E * Y + F$$

В 1988 г. известные американские специалисты в теории динамических систем и эргодической теории Барнсли и Слоан предложили некоторые идеи, основанные на соображениях теории динамических систем, для сжатия и хранения графической информации. Они назвали свой метод методом

фрактального сжатия информации. Происхождение названия связано с тем, что геометрические образы, возникающие в этом методе, обычно имеют фрактальную природу в смысле Мандельброта.

На основании этих идей Барнсли и Слоан создали алгоритм, который, по их утверждению, позволит сжимать информацию в 500-1000 раз. Вкратце метод можно описать следующим образом. Изображение кодируется несколькими простыми преобразованиями (в нашем случае аффинными), т.е. коэффициентами этих преобразований (в нашем случае A,B,C,D,E,F).

Например, закодировав какое-то изображение двумя аффинными преобразованиями, мы однозначно определяем его с помощью 12-и коэффициентов. Если теперь задаться какой-либо начальной точкой (например, X=0, Y=0) и запустить итерационный процесс, то мы после первой итерации получим две точки, после второй - четыре, после третьей - восемь и т.д. Через несколько десятков итераций совокупность полученных точек будет описывать закодированное изображение. Но проблема состоит в том, что очень трудно найти коэффициенты IFS, которые кодировали бы произвольное изображение.

Для построения IFS применяют, кроме аффинных, и другие классы простых геометрических преобразований, которые задаются небольшим числом параметров. Например, проективные:

$$X' = (A1 * X + B1 * Y + C1) / (D1 * X + E1 * Y + F1)$$

$$Y' = (A2 * X + B2 * Y + C2) / (D2 * X + E2 * Y + F2)$$

или квадратичные:

$$X' = A1 * X * X + B1 * X * Y + C1 * Y * Y + D1 * X + E1 * Y + F1$$

$$Y' = A2 * X * X + B2 * X * Y + C2 * Y * Y + D2 * X + E2 * Y + F2$$

преобразования на плоскости.

В качестве примера использования IFS для построения фрактальных структур рассмотрим кривую Коха (см. рис.8.1) и дракона Хартера-Хейтуэя (см. рис.8.2). Выделим в этих структурах подобные части и для каждой из них

вычислим коэффициенты аффинного преобразования. В аффинный коллаж будет включено столько аффинных преобразований, сколько существует частей, подобных целому изображению.

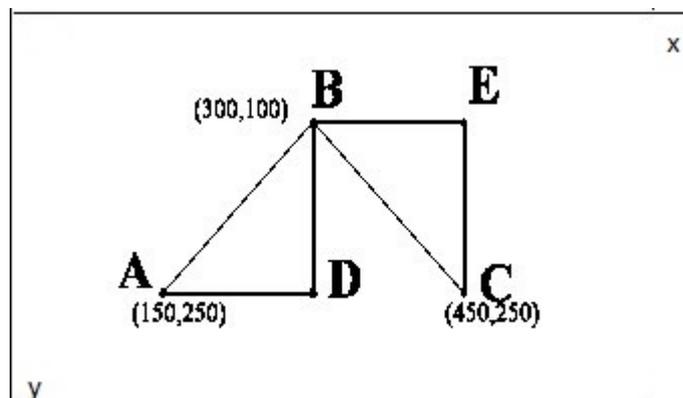


Рис 8.5. Заготовка для построения IFS дракона Хартера-Хейтуэя

Построим IFS для дракона Хартера-Хейтуэя. Для этого расположим первое поколение этого фрактала на сетке координат дисплея 640 x 350 (рис.8.5). Обозначим точки получившейся ломаной **A**, **B**, **C**. По правилам построения, у этого фрактала две части, подобные целому (на рис.8.5 - это ломаные **ADB** и **BEC**). Зная координаты концов этих отрезков, можно вычислить коэффициенты двух аффинных преобразований, переводящих ломаную **ABC** в **ADB** и **BEC**:

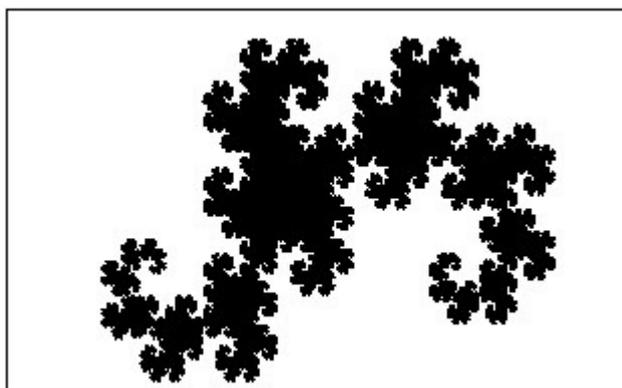
$$X' = -0.5 * X - 0.5 * Y + 490$$

$$Y' = 0.5 * X - 0.5 * Y + 120$$

$$X' = 0.5 * X - 0.5 * Y + 340$$

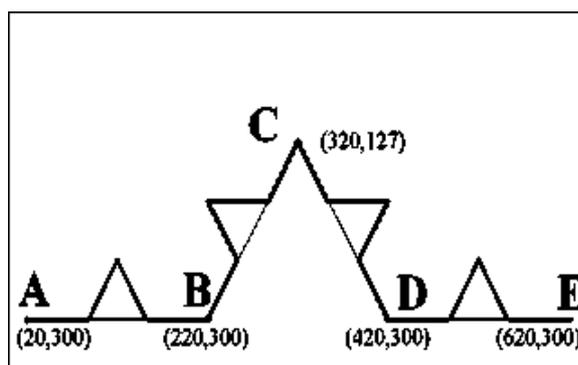
$$Y' = 0.5 * X + 0.5 * Y - 110$$

Задавшись начальной стартовой точкой (например,  $X=0$ ,  $Y=0$ ) и итерационно действуя на нее этой IFS, после десятой итерации на экране получим фрактальную структуру, изображенную на рис.8.6, которая представляет собой дракон Хартера-Хейтуэя. Его кодом (сжатым описанием) является набор коэффициентов двух аффинных преобразований.



*Рис 8.6. Дракон Хартера-Хейтуэя, построенный с помощью IFS  
в прямоугольнике 640x350*

Аналогично можно построить IFS для кривой Коха. Нетрудно видеть, что эта кривая имеет четыре части, подобные целой кривой (см. рис 8.1). Для нахождения IFS опять расположим первое поколение этого фрактала на сетке координат дисплея 640 x 350 (рис.8.7).



*Рис 8.7. Заготовка для построения IFS кривой Коха*

Для ее построения требуется набор аффинных преобразований, состоящий из четырех преобразований:

$$X' = 0.333 * X + 13.333$$

$$Y' = 0.333 * Y + 200$$

$$X' = 0.333 * X + 413.333$$

$$Y' = 0.333 * Y + 200$$

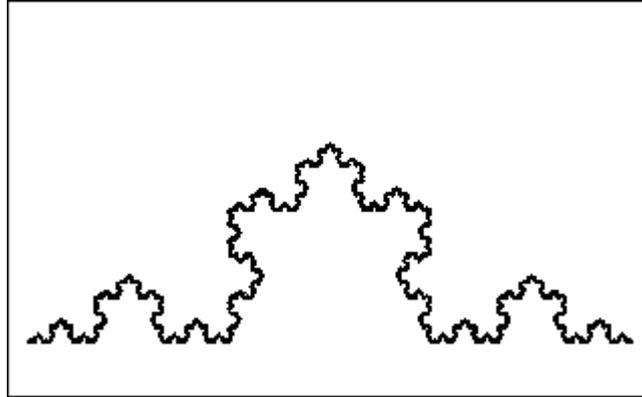
$$X' = 0.167 * X + 0.289 * Y + 130$$

$$Y' = -0.289 * X + 0.167 * Y + 256$$

$$X' = 0.167 * X - 0.289 * Y + 403$$

$$Y' = 0.289 * X + 0.167 * Y + 71$$

Результат применения этого аффинного коллажа после десятой итерации можно увидеть на рис.8.8.



*Рис 8.8. Кривая Коха, построенная с помощью IFS в прямоугольнике 640x350*

Использование IFS для сжатия обычных изображений (например, фотографий) основано на выявлении локального самоподобия, в отличие от фракталов, где наблюдается глобальное самоподобие, и нахождение IFS не слишком сложно (мы сами только что в этом убедились). По алгоритму Барнсли происходит выделение в изображении пар областей, меньшая из которых подобна большей, и сохранение нескольких коэффициентов, кодирующих преобразование, переводящее большую область в меньшую. Требуется, чтобы множество "меньших" областей покрывало все изображение. При этом в файл, кодирующий изображения, будут записаны не только коэффициенты, характеризующие найденные преобразования, но и местоположение и линейные размеры "больших" областей, которые вместе с коэффициентами будут описывать локальное самоподобие кодируемого изображения. Восстанавливающий алгоритм в этом случае должен применять каждое преобразование не ко всему множеству точек, получившихся на предыдущем шаге алгоритма, а к некоторому их подмножеству, принадлежащему области, соответствующей применяемому преобразованию.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- Математическое и имитационное моделирование: метод. указания по выполнению лаб.-практ. цикла работ для студентов направления подготовки 230700.62 (прикладная информатика) в соответствии с ФГОС/ сост. Г. Л. Нохрина; - Екатеринбург: УГЛТУ, 2012. - 28 с.: ил., <http://elar.usfeu.ru/handle/123456789/982>
- Козин Р.Г. Математическое моделирование: учебное пособие. – М.: МИФИ, 2006. – 85 с.
- Белова И.М. Компьютерное моделирование: учебно-методическое пособие для студентов направления «Прикладная математика и информатика». — М.: МГИУ, 2007. — 81 с.
- Астафьев Г.Б., Короновский А.А., Храмов А.Е. Клеточные автоматы: учебно-методическое пособие. - Саратов: ГосУНЦ «Колледж», 2003.- 24 с.

# ОГЛАВЛЕНИЕ

1. ЛЕКЦИЯ 1. ВВЕДЕНИЕ .....	3
1.1. Понятия модели и моделирования. Классификация моделей .....	3
1.2. Адекватность модели объекту моделирования .....	5
1.3. Адекватность конструктивных моделей .....	7
1.4. Адекватность модели объекта в случае недоступности наблюдателю самого объекта .....	8
1.5. Установление адекватности в случае существования единственной модели объекта .....	9
2. ЛЕКЦИЯ 2. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ .....	13
2.1. Информационные модели .....	13
2.2. Основные понятия информационного моделирования.....	14
2.3. Связи между объектами.....	19
3. ЛЕКЦИЯ 3. МАТЕМАТИЧЕСКИЕ МОДЕЛИ.....	21
3.1. Основные понятия.....	21
3.2. Классификация математических моделей .....	27
3.3. Пример построения математической модели.....	28
4. ЛЕКЦИЯ 4. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ .....	32
4.1. Понятие имитационного моделирования.....	32
4.2. Технология имитационного моделирования.....	36
4.3. Этапы имитационного моделирования .....	37
4.4. Виды имитационного моделирования .....	38
5. ЛЕКЦИЯ 5. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ПРОЦЕССОВ .....	39
5.1. Введение .....	39
5.2. Метод Монте-Карло.....	40
5.3. Вычисление интегралов по методу Монте-Карло.....	44
6. ЛЕКЦИЯ 6. ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ .....	51
6.1. Основные понятия .....	51
6.2. Линейные конгруэнтные генераторы.....	56
6.3. Смешанные генераторы .....	58
6.4. Мультипликативные генераторы .....	59
7. ЛЕКЦИЯ 7. КЛЕТОЧНЫЕ АВТОМАТЫ .....	61
7.1. Введение .....	61
7.2. Игра «Жизнь» .....	65
8. ЛЕКЦИЯ 8. ФРАКТАЛЫ .....	76
8.1. Понятие "фрактал".....	76
8.2. Классификация фракталов .....	77
8.3. Системы итерируемых функций .....	82
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	87