

**Ю. Б. Колесов**  
**Ю. Б. Сениченков**

# **МОДЕЛИРОВАНИЕ СИСТЕМ**

## **ДИНАМИЧЕСКИЕ и ГИБРИДНЫЕ СИСТЕМЫ**

Рекомендовано Учебно-методическим объединением  
по университетскому политехническому образованию  
в качестве учебного пособия для студентов высших учебных  
заведений, обучающихся по направлению подготовки  
220100 — "Системный анализ и управление"

Санкт-Петербург

«БХВ-Петербург»

2012

УДК 681.3.06(075.8)  
ББК 32.973.26-018.2я73  
К60

**Колесов Ю. Б., Сениченков Ю. Б.**

К60 Моделирование систем. Динамические и гибридные системы.  
Учебное пособие. — СПб.: БХВ-Петербург, 2012. — 224 с.: ил.

ISBN 5-94157-578-5

Учебное пособие посвящено компьютерному моделированию сложных динамических систем. Рассмотрены основные понятия моделирования — объект, модель, система, математическая модель, компьютерная модель, динамическая система, гибридная система, а также основы языка моделирования, необходимые для понимания приводимых примеров. Подробно рассматриваются две основные математические модели — (классическая) динамическая система и ее обобщение — гибридная система. Обсуждаются проблемы построения иерархических многокомпонентных моделей сложных динамических систем. Материал сопровождается большим количеством примеров.

*Для студентов вычислительных специальностей технических вузов*

УДК 681.3.06(075.8)  
ББК 32.973.26-018.2я73

#### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Татьяны Олоновой</i>
Корректор	<i>Татьяна Кошелева</i>
Оформление обложки	<i>Елена Беляева</i>
Зав. производством	<i>Николай Тверских</i>

Рецензенты:

Евгеньев Г.Б., д.т.н., профессор кафедры «Компьютерные системы автоматизации производств» МГТУ им. Баумана

Ивановский Р.И., д.т.н., профессор кафедры «Распределенные вычисления и компьютерные сети» СПбГПУ

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 18,06.  
Тираж 2000 экз. Заказ №  
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Отпечатано с готовых диапозитивов  
в ОАО "Техническая книга"  
190005, Санкт-Петербург, Измайловский пр., 29

ISBN 5-94157-578-5

© Колесов Ю. Б., Сениченков Ю. Б., 2006  
© Оформление, издательство "БХВ-Петербург", 2006

# Оглавление

<b>Предисловие .....</b>	<b>5</b>
<b>Глава 1. Введение в компьютерное моделирование .....</b>	<b>9</b>
<b>Модель.....</b>	<b>10</b>
Реальный объект и модель.....	10
Виды моделей .....	13
От моделей математических к компьютерным моделям.....	14
Компьютерное моделирование и вычислительный эксперимент.....	19
<b>Автоматизация компьютерного моделирования.....</b>	<b>21</b>
Программные средства моделирования .....	22
Языки моделирования .....	28
Классификация компьютерных моделей .....	31
<b>Пространство, время, поведение .....</b>	<b>35</b>
Пространство состояний.....	35
Время .....	37
Синхронизация.....	39
Объект и его окружение. Изолированные и открытые модели .....	42
Объект и система объектов.....	45
Динамические и статические модели .....	45
Детерминированные и вероятностные модели.....	46
Учет запаздывания.....	47
Непрерывные, дискретные, непрерывно-дискретные модели .....	48
Гибридные системы.....	51
Последовательные и параллельные процессы.....	53
Обобщение понятия состояния .....	54
<b>Основные конструкции языка моделирования.....</b>	<b>55</b>
Описание динамических систем .....	56
Описание гибридных систем .....	57

Состояния .....	58
Переходы .....	60
<b>Глава 2. Изолированные однокомпонентные системы.....</b>	<b>63</b>
Изолированные системы .....	63
Непрерывные модели .....	65
Непрерывно-дискретные модели. Гибридные системы .....	84
Модели, сводящиеся к динамическим и гибридным системам.....	114
Дискретные модели .....	115
Уравнения в частных производных .....	121
Марковские модели .....	124
Дискретные модели. Цепи Маркова .....	125
Непрерывные модели. Непрерывные цепи Маркова .....	128
<b>Глава 3. Компонентные модели .....</b>	<b>131</b>
Композиция параллельных компонентов .....	138
Параллельное объединение непрерывных компонентов .....	141
Ориентированные блоки .....	142
Неориентированные блоки .....	153
Параллельное объединение гибридных компонентов .....	159
Композиция параллельно работающих блоков с контактами.....	171
Пример: синхронная и асинхронная композиция гибридных автоматов....	171
Последовательная композиция компонентов .....	175
<b>Глава 4. Численное моделирование .....</b>	<b>179</b>
Системы линейных алгебраических уравнений.....	181
Проблема собственных значений .....	188
Системы нелинейных алгебраических уравнений.....	194
Системы обыкновенных дифференциальных уравнений .....	196
Системы алгебро-дифференциальных уравнений .....	209
<b>Литература .....</b>	<b>216</b>
Дополнительная литература к главе 1 .....	217
Дополнительная литература к главе 2.....	218
Дополнительная литература к главе 3.....	218
Дополнительная литература к главе 4.....	219
<b>Предметный указатель .....</b>	<b>221</b>

# Предисловие

Тема, обозначенная в заголовке данной книги — моделирование систем — практически безгранична. Поэтому стоит пояснить, о чем говорится в этой книге и о чем в ней *не* говорится. В этой книге *нет* рецептов, как строить математические модели конкретных прикладных систем. Различным моделям физических, биологических, экономических и других систем посвящено множество специальных работ, и мы не будем их пересказывать. В этой книге *не* рассматривается теория динамических систем, равно как и теории гибридных систем уделено ровно столько внимания, сколько необходимо для понимания основных проблем, возникающих при моделировании сложных объектов. Эта книга также *не* является введением в численные методы, используемые в современных пакетах моделирования.

Эта книга посвящена вопросам *компьютерного* моделирования *сложных* динамических систем.

*Сложной динамической системой* (СДС) будем называть систему, обладающую следующими свойствами:

- система содержит много компонентов, состав которых может изменяться во время ее функционирования;
- компоненты имеют различную физическую природу;
- между компонентами имеются как физические, так и информационные связи;
- в общем случае система имеет иерархическую многоуровневую структуру;
- элементарные компоненты могут быть непрерывными, дискретными или непрерывно-дискретными (гибридными).

Для того чтобы провести вычислительный эксперимент с СДС, необходимо уметь решать ряд задач.

Во-первых, нужно правильно построить математическую модель всей системы в целом по математическим моделям отдельных компонентов и связей между ними, в том числе нужно уметь делать это и динамически при изменении состава компонентов модели. Во-вторых, как правило, уравнения, описывающие совокупное поведение системы, допускают только численное решение. Следовательно, нужно уметь преобразовывать математические модели из "внешнего" представления, удобного для человека, к "внутреннему", с которым могут работать современные численные методы. Кроме того, необходимо уметь правильно интерпретировать результаты вычислений и причины трудностей, возникающих при построении численного решения — в терминах исходных моделей. Наконец, в-третьих, необходимо уметь правильно определять моменты возникновения дискретных событий в модели, приводящих к смене поведения, и корректно "склеивать" решения до и после возникновения события. Решение этих трех задач для СДС выходит за границы возможностей человека и требует создания специальных инструментальных средств автоматизации моделирования.

Книга состоит из трех томов. *Том I* посвящен теоретическим основам компьютерного моделирования. В нем вводятся базовые понятия и рассматриваются две математические модели — динамическая и гибридная системы, для которых уже существуют приемлемые для практики решения перечисленных выше задач. В *томе I* также рассмотрены общие вопросы автоматического объединения компонентов, а также современные численные методы, необходимые для моделирования СДС. Все вводимые понятия иллюстрируются простыми примерами, выполненными с помощью пакета моделирования СДС MvStudium ([www.mvstudium.com](http://www.mvstudium.com)), в коллектив разработчиков которого входят авторы данной книги.

*Том II* посвящен компьютерным инструментам автоматизации моделирования СДС. В нем излагаются основные современные подходы к компьютерному моделированию сложных динамических объектов и рассматриваются наиболее распространенные пакеты моделирования. Обсуждаются основные принципы объектно-ориентированного моделирования и предлагается язык моделирования СДС, построенный для базового набора математических моделей, рассмотренных в *томе I*, а также структура и принципы работы инструментальных средств автоматизации моделирования СДС.

*Том III* включает большой набор примеров систем, для которых построены и подробно описаны компьютерные модели с использованием различных подходов и пакетов моделирования.

Сейчас вы держите в руках *том I*.

Эта книга — результат обработки нескольких курсов лекций из цикла "Моделирование", читаемых авторами на кафедре "Распределенные вычисления и компьютерные сети" Санкт-Петербургского государственного политехнического университета. Курсы, подвергшиеся переработке, в учебном плане кафедры объединены под общим названием "Моделирование сложных динамических систем".

*Главы 1 и 2* в основном соответствуют курсу "Введение в моделирование динамических систем". Для данного курса предусмотрено два часа лекций в неделю и два часа лабораторных работ, на которых студенты приобретают навыки работы с имеющимися на кафедре математическими пакетами.

*Глава 3* является составляющей курса "Математическое программное обеспечение. Машинные арифметики, систематизированные коллекции, математические пакеты", а именно — разделом, посвященным программным реализациям численных методов решения обыкновенных дифференциальных уравнений. Курс также сопровождается лабораторными работами, но их цель уже другая — студент должен сам разработать программное обеспечение для решения конкретной математической задачи, или провести вычислительный эксперимент, используя имеющиеся в пакете средства. Именно здесь он впервые сталкивается с проблемой ручной предварительной подготовки уравнений к их численному решению.

Перечисленным курсам предшествует достаточно большой курс "Численный анализ", и поэтому предполагается, что студенты владеют материалом, изложенным в *главе 4*. В данной книге он играет роль справочного раздела.

Цикл "Моделирование сложных динамических систем" заканчивается курсом "Компьютерное моделирование многокомпонентных систем" и научным семинаром. Научный семинар играет особую роль в процессе обучения — студент самостоятельно выбирает один из доступных пакетов моделирования и, выступая в роли разработчика пакета, пытается наилучшим образом представить достоинства "своей" среды моделирования. Для сравнения обычно выбирается одна и та же модель для пакетов одного класса. Параллельно читаемый курс "Компьютерное моделирование многокомпонентных систем" помогает увидеть то общее, что объединяет современные среды моделирования, и что отличает их.

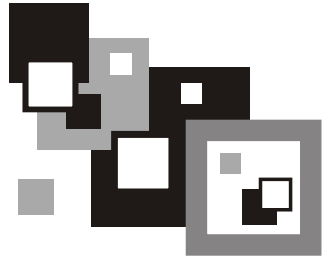
Попытка найти единый подход к описанию современных технологий моделирования сложных динамических систем оказалась трудной задачей. В издаваемых сегодня книгах делается упор на технику разработки моделей в различных средах без пояснения того, что стоит за этой техникой. Обучение моделированию порой сводится к выучиванию языка среды и набора "протопанных" другими тропинок, ведущих к правдоподобному результату.

Хотелось бы уйти от такой практики преподавания моделирования, но насколько это удалось — судить читателю.

Сложность задачи приводила к многократным переделкам материала, и мы признательны редакции издательства "БХВ-Петербург", что она в конце концов сумела остановить процесс внесения изменений в очередной, уже отредактированный и готовый к верстке текст. Особая благодарность Анне Кузьминой, на чью долю выпало чтение и редактирование многочисленных вариантов книги.

*Авторы. 29 ноября 2005 года.*





## Глава 1

# Введение в компьютерное моделирование

В последние годы все чаще говорят о *компьютерных экспериментах* и необходимых для их проведения *компьютерных инструментах*.

Компьютерные эксперименты отличаются от привычных натуральных экспериментов тем, что исследователь экспериментирует не с реальным объектом, а с его компьютерной моделью.

Компьютерное моделирование как метод исследования является естественным развитием математического моделирования. В основе компьютерных моделей, по крайней мере тех, о которых речь пойдет далее, лежат математические модели. Эти модели строятся автоматически по описанию структуры и поведения исследуемой системы, принятому в языке моделирования. Построенные математические модели обычно сводятся к системам уравнений, решение которых редко удается найти в замкнутой форме, и их приходится решать численно, с помощью программных реализаций численных методов. Программная реализация математической модели строится автоматически пакетом моделирования. Наконец при компьютерном моделировании широко используется возможность визуализации как самой модели, так и ее поведения.

Исследуемая модель предстает перед пользователем в виде узнаваемых графических образов, ее параметры можно регулировать и ею можно управлять почти так же, как и в реальной жизни. В компьютерной модели используемая математическая модель, ее программная реализация, системное и математическое программное обеспечение, необходимые для воспроизведения поведения модели, спрятаны за дружелюбным интерфейсом. Все это позволяет создавать и исследовать компьютерные модели специалистам, далеким от прикладной математики и информатики.

Для разработки компьютерных моделей и экспериментирования с ними создано специальное программное обеспечение — универсальные среды моделирования, которые уже практически повсеместно вытеснили ручное исследование сложных математических моделей. Эти инструменты, включающие графические редакторы моделей и виртуальные стенды, постепенно заменяют особый вид программного обеспечения — пакеты прикладных программ, еще недавно использовавшиеся для проведения вычислительного эксперимента. Системы моделирования позволяют создавать виртуальные, или лучше компьютерные, лаборатории, аналогичные реальным, оснащенные моделями измерительных приборов и датчиками сигналов.

Современные инструменты моделирования предлагают исследователям новые технологии математического моделирования и проведения вычислительного эксперимента.

Нашей целью является изучение методов построения компьютерных моделей, возможностей компьютерного моделирования и технологий проведения компьютерных экспериментов.

С тех пор как Ньютон предложил использовать математические модели для изучения, объяснения и предсказания поведения окружающего нас мира, было создано огромное количество различных математических моделей. Мы будем рассматривать только модели, в основе которых лежат динамические и гибридные системы. Такие математические модели назовем *сложными динамическими системами*.

## Модель

Зарождение любой естественной науки начинается с натуральных экспериментов над реальными объектами и системами для выяснения их свойств, обнаружения закономерностей и законов, которым они подчиняются. Однако давно было замечено, что очень много можно узнать о реальном мире, если вместо реальных объектов изучать их упрощенные заменители — модели. В последующем выяснилось, что строить и использовать модели можно не только для познания окружающего нас мира.

## Реальный объект и модель

*Моделью* объекта называется любой другой объект, отдельные свойства которого полностью или частично совпадают со свойствами исходного. Модель заменяет исходный объект, сохраняя только некоторые, существенные свойства объекта. Какие свойства считать существенными, а какие — нет, определяется целями моделирования. С точки зрения существенных свойств

реальный объект и его модель не должны различаться в рамках заданной точности. Построение и использование модели вместо исходного объекта называется *моделированием*.

Модель редко удается построить сразу. Построение модели — трудно формализуемый процесс. Сложно заранее сказать, какие свойства окажутся существенными, а какие — нет. Это можно выяснить только по ходу экспериментов с моделью и сравнения ее свойств со свойствами исходного объекта. Не менее трудно заранее определить область применимости модели, для этого также требуются эксперименты. Процесс построения модели — это итерационный процесс.

Моделирование можно представить себе как последовательность этапов:

1. Выбор объекта и целей моделирования.
2. Разработка критериев соответствия между объектом и моделью. Создание модели.
3. Проверка соответствия модели и объекта.
4. Использование модели для решения поставленных задач.

Построенные модели можно использовать для различных целей.

□ **Познание.** В процессе построения модели познается, как устроен исходный объект. В этом случае модель отражает представление исследователя о существенных факторах, закономерностях и причинно-следственных связях в изучаемом объекте. Правильность наших представлений проверяется с помощью натуральных экспериментов: результаты моделирования сопоставляются с реальными экспериментальными данными. В определенном смысле все научные исследования сводятся к построению моделей природных объектов. В процессе познания существующие теории или модели мира видоизменяются, уточняются границы их применимости. Вместо одних теорий появляются другие, объясняющие более широкий круг явлений. В тоже время старые и новые модели могут прекрасно сосуществовать.

Примером могут служить законы механики Аристотеля и Ньютона. Согласно Аристотелю, равномерное движение порождается постоянно приложенной силой. Согласно Ньютону, постоянно приложенная сила порождает постоянное ускорение. Это две теории — два различных объяснения того, как устроен мир. Законы Аристотеля проще и "подтверждены" повседневной жизнью: практически все мы и не один раз толкали "заглохший" автомобиль до заправки. Законы Ньютона более сложны. Они также позволяют нам объяснить, но уже совсем по-другому, почему нам приходится подталкивать автомобиль, чтобы он двигался при наличии силы трения. Но их возможности существенно больше, с их помощью

можно объяснить природу движения космических тел. Как показывают исследования, многие в повседневной жизни опираются на законы Аристотеля, и ничего страшного с ними не случается, в то время как для научных и инженерных расчетов это недопустимо, и необходимо применять законы Ньютона.

- **Предсказание.** Модель позволяет предсказывать поведение интересующего нас объекта в заданных условиях. Исследуя модели, можно узнать заранее, какими будут реакции объекта на управляющие воздействия. Модель может предсказывать существование у исследуемого объекта оптимальных режимов работы. С ее помощью можно находить значения параметров, при которых они реализуются. С помощью модели можно исследовать особые режимы работы, которые трудно или опасно изучать на реальном объекте. Однако возможности предсказания могут оказаться шире — удачно построенная модель способна предсказать явления, которые мы еще не наблюдали, и тем самым указать новое направление научных исследований.
- **Обучение.** Модель может имитировать реальный объект, т. е. сохранять дополнительно многие его внешние черты. Такие модели или имитаторы используются при обучении. Имитатор пульта управления электростанцией, например, может служить тренажером при подготовке персонала к последующей работе в реальной обстановке. Макеты космических кораблей или военных судов могут служить наглядными пособиями для будущих космонавтов и военных. Модели музеев — компьютерные диски с фотографиями самого музея, его залов, а также его экспонатов используются при дистанционном обучении.

Модели объектов могут изучаться вместо реальных объектов в учебных лабораториях, демонстрироваться на лекциях. В последнее время все большую популярность в обучении как очном, так и дистанционном приобретают компьютерные лаборатории.

- **Конструирование.** Модель может создаваться как прототип нового устройства. Например, спецификация будущего программного комплекса одновременно является его моделью. Интересно отметить, что здесь нарушаются привычные для нас причинно-следственные соотношения: не модель формируется для реального объекта, а реальный объект создается на основании разработанной и исследованной модели. Модели-прототипы используются в основном на ранних этапах создания новых технических систем. Например, всем программистам хорошо известны заглушки, применяемые вместо еще неразработанных модулей в больших программных комплексах. При отладке алгоритмов управления на ранних этапах проектирования повсеместно используются имитаторы объектов управления.

Необходимость замещения или моделирования может быть также обусловлена тем, что реальный эксперимент нецелесообразен, опасен или вообще неосуществим при современном уровне техники. Трудно себе представить, что человечество будет ставить эксперименты с Землей, повышая ее температуру или меняя орбиту, или использовать Солнце как экспериментальный образец при изучении звезд.

Очевидно, что любая модель — это лишь приближение к моделируемому объекту, т. е. обладает только специально отобранными свойствами. Теоретически можно поставить задачу создать модель, максимально приближенную к исходному объекту. Такая постановка важна только для теории познания, когда мы стремимся все более полно познать мир. Для предсказания, конструирования и обучения исчерпывающе полной модели объекта не нужно — достаточно такого приближения, которое решает поставленную задачу. Рукотворные, технические объекты создаются по формальным спецификациям. Реальный объект, вышедший из сборочного цеха, всегда неизмеримо богаче по своим свойствам собственной спецификации или макета.

Для одного объекта одновременно может существовать несколько различных моделей. Эти модели могут отражать различные свойства исходного объекта, существенные для разных применений. Например, при исследованиях аэродинамических свойств будущего самолета в аэродинамической трубе достаточно его геометрического макета, изготовленного в натуральную величину или уменьшенного. В то же время для тренировки обслуживающего персонала или группы спецназа макет самолета должен точно отражать планировку салона, расстановку кресел и даже используемые отделочные материалы. Модели могут также отражать одни и те же свойства с различной степенью точности. Например, при исследовании полета теннисного мяча разумно предположить, что Земля плоская. При исследовании полета снаряда, выпущенного из дальнобойного орудия, уже имеет смысл использовать сферическую модель Земли. Наконец, при расчете траектории межконтинентальной баллистической ракеты необходимо рассматривать Землю как эллипсоид и учитывать гравитационные аномалии.

## Виды моделей

Модели можно условно разделить на две группы: *материальные* и *идеальные*. Первой группе соответствует предметное моделирование, а второй — абстрактное. В данной книге рассматривается абстрактное моделирование, поэтому о предметном моделировании мы скажем всего несколько слов. Основными разновидностями предметного моделирования являются физическое и аналоговое моделирование.

*Физическим* принято называть моделирование (макетирование), при котором реальный объект заменяется его увеличенной или уменьшенной копией,

сохраняющей геометрические пропорции и физические принципы действия. При физическом моделировании используется теория подобия, которая позволяет установить количественные отношения между свойствами модели и реального объекта, границы, в которых они сохраняются и, используя эти соотношения, по зависимостям, обнаруженным в модели, строить зависимости, справедливые для реального объекта. Кроме того, при физическом моделировании может использоваться не одна, а несколько моделей.

*Аналоговое моделирование* основано на замене исходного объекта объектом другой физической природы, поведение которого определяется аналогичными физическими законами. Например, колебания и резонанс в механических системах аналогичны колебаниям и резонансу в электрических цепях. В то же время исследовать колебания тока в электрической цепи, аналогичные угловым колебаниям ракеты, значительно проще и безопаснее, чем экспериментировать с летящей ракетой. Наибольшее распространение аналоговое моделирование получило в эру аналоговых вычислительных машин (АВМ), постепенно вытесненными цифровыми. Недостатками аналогового моделирования являются сложность создания модели и низкая точность результатов.

*Идеальные модели* — это абстрактные образы замещаемых объектов. Различают два типа идеального моделирования: интуитивное и знаковое. *Интуитивное моделирование* используется человеком и другими живыми существами для отражения окружающего мира и предсказания его реакций. Как оно осуществляется, мы не знаем. *Знаковое моделирование* предполагает использование в качестве моделей знаков или символов: схемы, графики, чертежи, тексты на разных языках, включая формальные, математические формулы и теории. Наиболее важным видом знакового моделирования является *математическое моделирование*.

Обязательным участником знакового моделирования является *интерпретатор знаковой модели*. Чертежи, тексты, формулы сами по себе не имеют никакого смысла без того, кто понимает их и использует в своей деятельности. Обычно интерпретатором знаковой модели является человек, понимающий смысл используемых знаков, однако уже сейчас существует множество технических устройств, обладающих схожими возможностями.

## От моделей математических к компьютерным моделям

Из всех возможных математических моделей мы будем рассматривать системы обыкновенных дифференциальных уравнений (динамические системы) и системы дифференциальных уравнений, дополненные системами алгебраических уравнений, с заданными начальными условиями, а также модели, сводящиеся к ним.

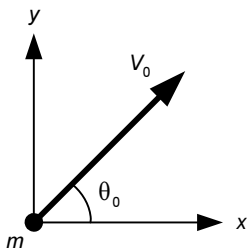
## Второй закон Ньютона

$$m \frac{d^2 x}{dt^2} = F,$$

записанный в форме дифференциального уравнения, описывает *динамические системы*, изучаемые уже в школе.

В школьном курсе физики только упоминается о существовании уравнений в приведенной выше форме. Для решения школьных задач по механике используются решения дифференциальных уравнений, записанные в виде формул с набором параметров. Так как для многих применение математического моделирования на этом и заканчивается, математическое моделирование предстает в далеком от действительности виде: гений пишет уравнения и решает их с помощью карандаша и бумаги, а практики подставляют в готовые формулы нужные значения параметров.

Примером такого "идеалистического" моделирования может служить исследование движения тела, брошенного с начальной скоростью  $V_0$  под углом  $\theta_0$  к горизонту в безвоздушной среде (рис. 1.1).



**Рис. 1.1.** Тело, брошенное под углом к горизонту

Динамика движения задается системой уравнений

$$\begin{cases} \frac{d^2 x}{dt^2} = 0, \\ \frac{d^2 y}{dt^2} = -g, \end{cases} \quad x(0) = 0, y(0) = 0, \quad \left. \frac{dx}{dt} \right|_{t=0} = V_0 \cdot \cos \theta_0, \quad \left. \frac{dy}{dt} \right|_{t=0} = V_0 \cdot \sin \theta_0 \quad (1.1)$$

которая имеет очевидное, выраженное через элементарные функции, решение:

$$\begin{cases} x = V_0 \cdot \cos \theta_0 \cdot t, \\ y = V_0 \cdot \sin \theta_0 \cdot t - \frac{g \cdot t^2}{2}. \end{cases}$$

Из этой модели легко получить в явном виде зависимость дальности точки падения от угла бросания:

$$L = \frac{2 \cdot V_0^2 \cdot \sin \theta_0 \cdot \cos \theta_0}{g}. \quad (1.2)$$

Продифференцировав это выражение по  $\theta_0$  и приравняв полученное выражение нулю, найдем, что угол максимальной дальности является решением уравнения  $\cos^2 \theta_0 - \sin^2 \theta_0 = 0$  и равен  $45^\circ$ . И этот ответ легко проверить на практике.

К сожалению, такие простые модели и их "ручные" способы анализа уже давно ушли в прошлое. Прежде всего, это связано с тем, что подавляющее большинство представляющих практический интерес математических моделей являются слишком сложными и не имеют решений, которые можно представить в замкнутой форме, т. е. в виде явного выражения, в которое входят только функции из заданного множества и операции над ними. Такие решения обычно строятся либо "руками", либо с помощью математических пакетов, умеющих выполнять символьные вычисления, и обычно называются *символьными*.

Если в условия предыдущей задачи добавить учет сопротивления воздуха, то получим систему уравнений:

$$\left\{ \begin{array}{l} \frac{dx}{dt} = V_x, \\ \frac{dV_x}{dt} = -\frac{F \cdot \cos T}{m}, \\ \frac{dy}{dt} = V_y, \\ \frac{dV_y}{dt} = -g - \frac{F \cdot \sin T}{m}, \\ F = C_X \cdot S_m \cdot \frac{\rho \cdot V^2}{2}, \\ V = \sqrt{V_x^2 + V_y^2}, \\ \sin T = \frac{V_y}{V}, \\ \cos T = \frac{V_x}{V}, \end{array} \right. \quad x(0) = 0, \quad y(0) = 0, \quad V_x(0) = V_0 \cdot \cos \theta_0, \quad V_y(0) = V_0 \cdot \sin \theta_0, \quad (1.3)$$

которую необходимо уже решать численно.

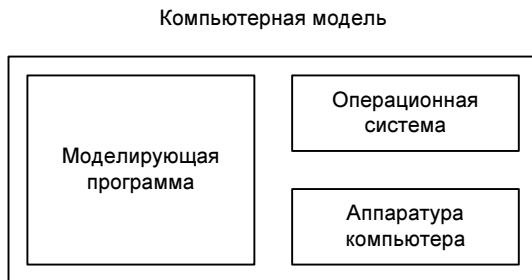


И символьное решение, и численное, конечно же, можно попытаться найти вручную, но обычно сложность сегодняшних задач такова, что найти их решение под силу только мощным вычислительным машинам.

Использование вычислительных машин при математическом моделировании привнесло качественный скачок в эту область. Новую технологию стали описывать в виде знаменитой триады: "математическая модель — численные методы — программная реализация численных методов", а машинные эксперименты над математической моделью называть *вычислительными экспериментами*.

Вычислительный эксперимент проводится над компьютерной моделью, часть которой и описывает приведенная выше триада. Компьютерная модель обладает особыми свойствами по сравнению с математической моделью. Она не является просто записанной на другом языке, языке компьютера, математической моделью.

Компьютерная модель имеет две составляющие — программную и аппаратную (рис. 1.2).



**Рис. 1.2.** Составляющие компьютерной модели

Программная составляющая (моделирующая программа) является абстрактной знаковой моделью специального вида, которая интерпретируется физическим устройством — процессором компьютера и "выполняется". В результате мы наблюдаем некоторый физический процесс, в частности движение образов на экране, которые мы интерпретируем как поведение модели. Под компьютером в данном случае достаточно понимать любое устройство, состоящее из аппаратной и программной частей, способное интерпретировать и выполнять программы. Это может быть и суперкомпьютер, и встроенный микропроцессор с "защитой" в его память программой. Совокупность компьютера и моделирующей программы является уже физическим устройством и, таким образом, компьютерные модели можно считать особым видом физических моделей.

Существование таких особых физических моделей на базе компьютеров позволяет говорить еще об одной стороне компьютерного моделирования. С этой точки зрения компьютерное моделирование обладает уникальным набором привлекательных свойств, к числу которых, прежде всего, относятся почти неограниченная сложность моделей. Удобным также является разделение на "мягкую" часть (программное обеспечение), которую необходимо изменять при переходе к другой модели, и "жесткую" часть (аппаратуру и операционную систему), которая остается неизменной. Компьютерная модель, таким образом, в определенных условиях может выступать в качестве имитатора реального объекта. Ее можно "подключать" к другим реальным объектам точно так, как ее физические прототипы. Это свойство компьютерных моделей позволяет использовать их в составе *полунатурных моделей*, собранных из реальных устройств и имитаторов реальных устройств, используемых при создании и испытаниях сложных технических систем.

Справедливости ради надо заметить, что компьютерные модели могут создаваться не обязательно на основе математических моделей. В их основе могут лежать записанные на языке программирования представления об объекте, существующие в сознании разработчика (пунктирная линия на рис. 1.3). Если назначение такой модели — создание на ее основе математической модели и затем ее реализация на вычислительной машине (сплошная линия на рис. 1.3), то такой подход вполне допустим. Такое моделирование в последнее время называют *имитационным*. Имитационное моделирование в этом смысле — это еще один этап компьютерного моделирования, когда мы первоначально имеем дело с трудно формализуемой моделью.

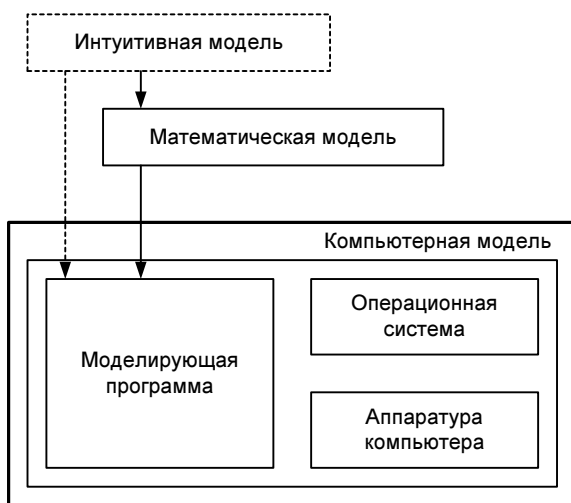


Рис. 1.3. Соотношение моделей

## Компьютерное моделирование и вычислительный эксперимент

Компьютерное моделирование всегда предполагает проведение вычислительного эксперимента. В рассмотренной нами модели, соотношение (1.2) сразу определяет зависимость дальности точки падения от  $\theta_0$ ,  $V_0$  и  $g$  (на рис. 1.4 показана зависимость дальности от угла  $\theta_0$ ). Это соотношение было получено чисто математическими методами, не требующими применения вычислительной техники.

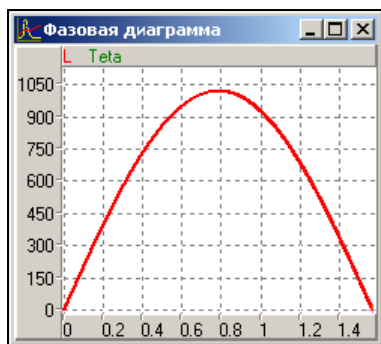


Рис. 1.4. Теоретическая зависимость дальности от угла бросания

Если же вы не имеете такого явного соотношения (как, например, для системы уравнений (1.3)), то вам необходимо несколько раз численно решить систему уравнений, задающую динамику объекта (рис. 1.5), определить точки падения и после этого получить искомую зависимость (рис. 1.6, а).

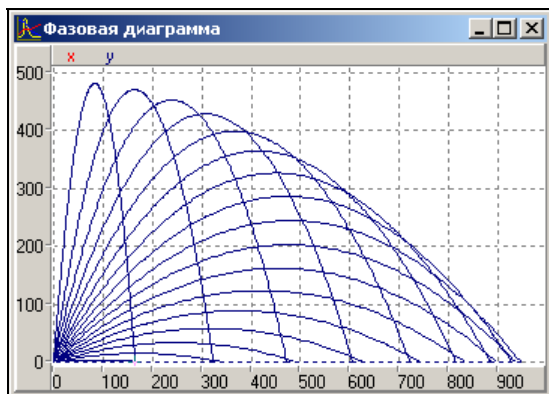
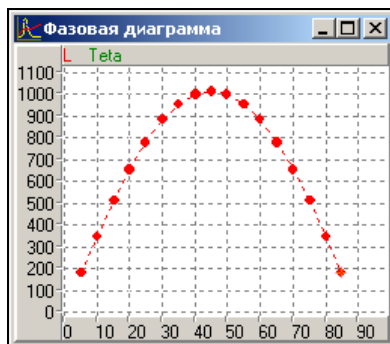
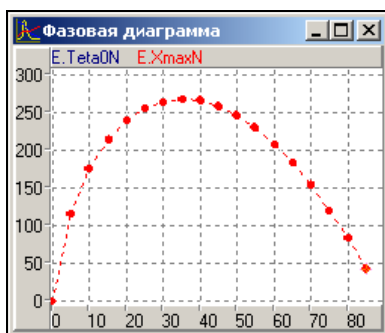


Рис. 1.5. Результаты вычислительного эксперимента



а



б

**Рис. 1.6.** Экспериментальные зависимости дальности полета от угла бросания: *а* — без сопротивления воздуха; *б* — с учетом сопротивления воздуха

Если из соотношения (1.2) сразу следует, что угол максимальной дальности не зависит от  $V_0$  и  $g$ , то для выявления зависимости дальности полета от  $V_0$ ,  $g$  и плотности воздуха  $\rho$  в модели (1.3) вам придется провести несколько серий экспериментов. После обработки результатов можно будет сделать вывод, что при увеличении  $V_0$  и  $\rho$  угол максимальной дальности уменьшается (рис. 1.6, б), а при увеличении  $g$  — увеличивается, приближаясь к  $45^\circ$ .

Таким образом, компьютерное моделирование действительно является особым видом проведения экспериментов с моделью исследуемого объекта.

Однако при проведении компьютерных экспериментов необходимо принимать специальные меры предосторожности, т. к. интересующие нас зависимости получены на основании численных расчетов, в которых могут присутствовать вычислительные погрешности. Кривая на рис. 1.4 также построена на основании точек, рассчитанных по зависимости (1.2) с помощью компьютера, однако

мы уверены, что вычислительные ошибки в этом случае ничтожны. Знание зависимости (1.2) всегда позволяет нам проверить правильность кривой, построенной компьютером. В то же время, если мы будем получать значение  $L$  путем прямого решения системы уравнений (1.1), то не сможем быть уверены, что истинный характер зависимости  $L(\theta_0)$  таков, как показано на рис. 1.6, а, а не такой, как на рис. 1.7.



Рис. 1.7. Пример графика, в который, возможно, вкралась вычислительная ошибка

Приведенная на графике ошибка, конечно же, является всего лишь примером кривой, которая может появиться в результате проводимого эксперимента. В нашем конкретном случае надо очень постараться, чтобы получить такую ошибочную зависимость. Однако при проведении сколько-нибудь сложного эксперимента ошибиться можно на каждом из этапов. Построенная математическая модель может содержать ошибки. Ошибки могут внести и численные методы, и их программные реализации, использующие арифметику машинных чисел. Возможны ошибки даже при такой простой операции, как построение графиков. Иногда требуется приложить много усилий, чтобы убедиться в правильном соответствии результатов компьютерного эксперимента результатам реальных экспериментов, а тем более убедить себя и других в правильности обнаруженной новой зависимости.

## Автоматизация компьютерного моделирования

Необходимость автоматизации процесса моделирования возникла в середине прошлого столетия, когда появилась возможность использовать для моделирования вычислительные машины.

В наиболее простой постановке задача сводится к следующему. Дана математическая модель в виде системы уравнений, требуется построить решение и

изучить его свойства в зависимости от значений параметров, лежащих в заданных пределах. Близкой с точки зрения задач автоматизации является задача поиска значений параметров системы уравнений, доставляющих экстремум заданного функционала качества. Даже если дополнительно известно, что решение или экстремум существует, необходимо проделать большую работу: выбрать численный метод для решения поставленной задачи, написать его программную реализацию и многократно решать систему уравнений с различными значениями параметров. Процесс решения таких задач ускоряется, если существуют библиотеки численных методов. Создание библиотек численных методов для заданных классов уравнений исторически было первым шагом на пути автоматизации моделирования. К концу 70-х годов прошлого столетия были созданы специализированные коллекции численных методов практически для всех областей численного анализа.

Наличие библиотек не спасает от необходимости многократно проводить и обрабатывать результаты отдельных экспериментов для различных входных данных. Дальнейшие шаги на пути автоматизации моделирования были связаны с разработкой систем автоматического проектирования (САПР) и систем автоматизации вычислительного эксперимента — пакетов прикладных программ (ППП). В обоих случаях исходные математические модели (уравнения) считаются заданными. САПР характеризуются большим числом исходных данных, стандартными расчетами и необходимостью оформлять результаты экспериментов в строго регламентированном виде. Именно эти этапы и автоматизируются. ППП предназначены для решения уникальных задач и, как правило, разрабатываются большими коллективами специалистов из различных областей. При создании ППП многократно изменяется исходная математическая модель, строятся новые численные методы и их программные реализации, т. к. исследователи имеют дело с малоизученным и сложным объектом. После внесения каждого изменения приходится заново повторять все эксперименты и сравнивать результаты. Автоматизация этих операций ускоряет процесс исследования. На основании созданных новых моделей уже можно создавать САПР.

Наиболее трудно автоматизируемым этапом является процесс построения модели. В то же время при разработке аппаратуры успешно применяются технологии разработки больших систем из стандартных блоков. Этот подход может быть использован и при построении сложных моделей. Воплощение этой идеи привело к созданию языков моделирования, применяемых в универсальных конструкторах моделей.

## Программные средства моделирования

Для решения уникальных задач, в том числе и задач моделирования, в условиях ограниченности быстродействия или памяти компьютера, для создания моделирующих программ применяются традиционные методы проектирования

программ. Автоматизация, конечно же, имеет место и здесь, но это автоматизация процесса создания программных комплексов, и собственно к моделированию она не относится.

На заре компьютерного моделирования все моделирующие программы были уникальными и писались непосредственно на существовавших в то время языках программирования (Алголе или Фортране). В качестве спецификации будущей программы выступала запись на математическом языке. Эффективность полученного кода повышалась за счет использования языка Ассемблера (написание всей программы или ее наиболее трудоемких частей в машинных командах иногда давало серьезный выигрыш в быстродействии). Подчеркнем, что этот способ создания моделирующих программ может быть использован и сейчас, но в те годы он был единственно возможным.

Для создания даже простой моделирующей программы непосредственно на выбранном процедурном языке требуется много времени. Причина больших временных затрат — существенная разница в уровнях абстракции математической модели и программы для вычислительной машины. Разработчик вынужден держать "в голове" соответствие между описаниями этих различных уровней, чтобы переводить любое изменение модели с математического уровня на уровень программного кода и интерпретировать полученные результаты машинных экспериментов в терминах исходной постановки задачи. Перевод и интерпретация, если они выполняются программистом, являются неиссякаемым источником ошибок. Такой "ручной" перевод математического описания в программный код требует очень высокой квалификации программиста. В то же время, как и всякая ручная работа, выполненная профессионалом, созданная руками моделирующая программа может оказаться очень эффективной. Если она многократно используется с незначительными модификациями для длительных расчетов при большом количестве параметров, то выигрыш в производительности часто перекрывает все трудности ее создания.

Для создания систематизированных коллекций для численных расчетов использовался и используется язык Фортран. Фортран обладает возможностью оптимизировать программу на различных уровнях, в частности на аппаратном. Систематизированные коллекции уникальны, создаются годами, тщательно тестируются. Их применение и особенно модификация требуют специальных знаний как в области численных методов, так и в программировании.

Появление коллекций и библиотек резко расширило возможности моделирования. Если математическая модель представляла собой не очень большую систему уравнений, то для программиста, знакомого с прикладной областью, для которой создавалась модель, с применяемыми численными методами и

соответствующей коллекцией, не составляло труда перевести ее в операторы Фортрана. Правда, найти такого программиста было сложно, и поэтому над программой совместно работали: специалист в прикладной области, математик и программист.

Разработка коллекций оказалась сложной задачей, потребовавшей создания теории машинной арифметики и методов оценки ошибок округления. Впервые было осознано, что ручной счет и машинные вычисления — не одно и то же, и что использование вычислительной техники приводит к появлению качественно новых задач.

Для сложных моделей, а также для моделей исследовательского типа, когда исходные математические модели еще очень "сырые", стали использоваться системы автоматизации моделирования (их часто называют просто системами моделирования или пакетами моделирования). Не случайно первые инструменты автоматизации моделирования возникли для создания моделей систем массового обслуживания, т. е. моделей с переменной структурой и параллельными процессами.

Система автоматизации моделирования позволяет автоматически строить моделирующую программу по математической модели системы и автоматически преобразовывать результаты вычислительных экспериментов на уровень абстракции математической модели. Таким образом, из цикла разработки модели исключаются трудозатраты на "ручное" программирование, а также ошибки программирования (остаются, конечно, ошибки самой системы моделирования, которые для отработанных пакетов довольно немногочисленны). Использование систем автоматизации моделирования позволяет на порядок повысить производительность труда при создании моделей, увеличить их надежность и резко расширить область применения компьютерного моделирования. В настоящее время существуют сотни программных пакетов моделирования.

При использовании системы автоматизации моделирования разработчик формулирует математическую модель исследуемой системы на формальном входном языке моделирования (рис. 1.8). Современные пакеты моделирования, как правило, включают специальные визуальные редакторы, позволяющие вводить описание моделируемой системы в форме, максимально удобной для восприятия человеком. Математические выражения пишутся с использованием многоэтажных дробей, символов интегралов, сумм и производных. Структура и поведение изображаются в виде структурных схем и графов переходов. Эти графические описания автоматически преобразуются в программу модели (жирная стрелка на рис. 1.8), которая вместе с исполняющей системой пакета моделирования и составляет моделирующую программу.



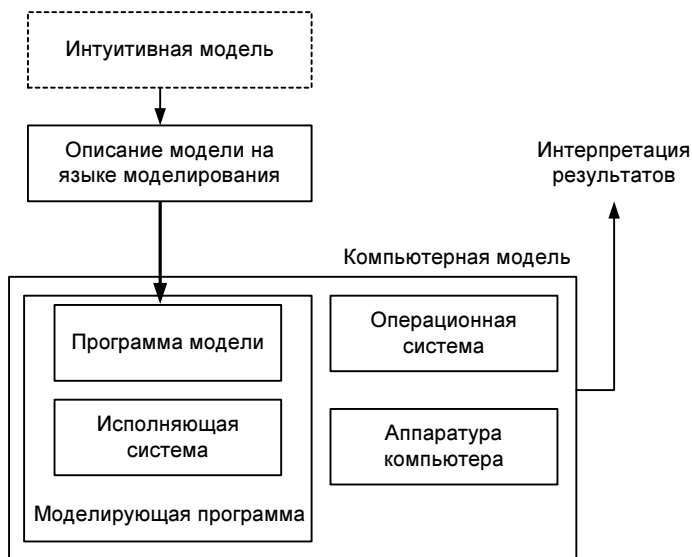


Рис. 1.8. Преобразование данных в системе автоматизации моделирования

"Программа модели" может быть либо действительно совокупностью программных модулей на некотором промежуточном языке программирования, которые после связывания с модулями исполняющей системы пакета моделирования образуют независимую выполняемую программу (такой пакет называется *компилирующим*), либо специальным внутренним представлением описания модели, которое воспринимается и интерпретируется исполняющей системой пакета (такой пакет называется *интерпретирующим*).

Постепенно графические описания унифицировались, систематизировались операции, и стали возникать языки моделирования. Необходимость использования специальных языков моделирования обусловлена несколькими причинами. Во-первых, традиционный математический язык все же ориентирован на человека как интерпретатора и опускает огромное множество "мелочей", совершенно необходимых для полного и однозначного определения описания модели, понятного транслятору пакета моделирования. Во-вторых, для удобства пользователей язык моделирования часто включает в себя семантику определенной области применения (например, языки моделирования систем массового обслуживания, такие как GPSS, включают такие специальные конструкции, как ресурс, очередь, транзакция и т. п.).

На рис. 1.9 представлена схема классификации (конечно, достаточно условная) пакетов моделирования. Специализированные пакеты используют специфические понятия конкретной прикладной области (химической технологии, теплотехники, электротехники и т. д.), решают подчас недоступные

пользователю конкретные системы уравнений, таким образом, имеют узкую область применения. Область применения универсальных пакетов шире, т. к. они ориентированы на определенный класс математических моделей и применимы для любой прикладной области, в которой эти модели используются. Специализированные пакеты очень трудно использовать для моделирования и исследования сложных систем с компонентами различной физической природы, т. к. каждую компоненту приходится изучать автономно с помощью различных пакетов. Полученные специализированные модели чрезвычайно сложно использовать в дальнейшем для построения комплексной модели сложной системы. Поэтому даже при моделировании отдельных подсистем предпочтительно использовать "универсальные" пакеты. "Универсальные" пакеты обычно разделяют на математические пакеты и пакеты компонентного моделирования.

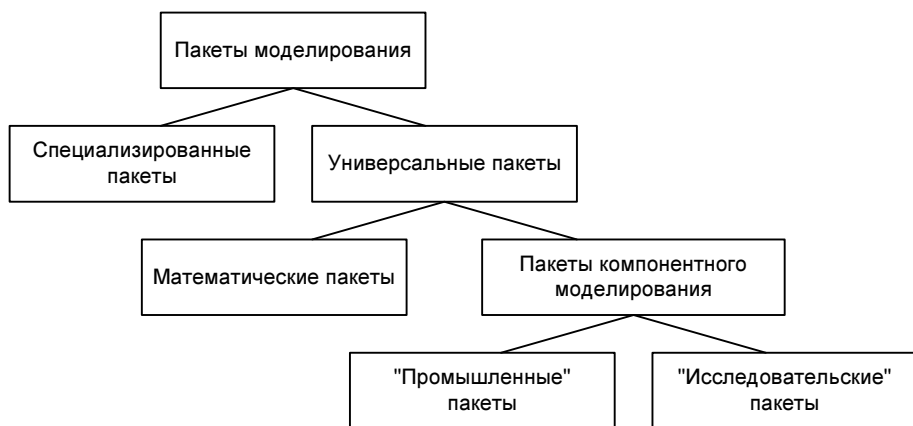


Рис. 1.9. Классификация пакетов моделирования

В математических пакетах (Mathematica, Mathcad, MATLAB, Maple) предполагается, что математическая модель всей моделируемой системы уже построена и ее требуется только исследовать. Такой подход характерен в основном для научных исследований, когда необходимо, прежде всего, убедиться в наличии требуемых свойств у новой модели. Математические пакеты позволяют проводить символьные преобразования модели, находить, если это возможно, решения уравнений в замкнутой форме, а случае неудачи — решать уравнения численно. Для проведения вычислительного эксперимента приходится писать программу на достаточно сложном языке программирования.

Компонентное моделирование предполагает, что описание моделируемой системы строится из компонентов (в т. ч. и готовых библиотечных), а совокупная математическая модель формируется пакетом автоматически. Размерность и

сложность совокупной системы уравнений таковы, что их решение приходится искать численно. Символьные вычисления если и проводятся, то лишь при решении отдельных вспомогательных задач. Компонентное моделирование — это основной способ проектирования технических объектов.

Пакеты компонентного моделирования по способам их применения или технологии моделирования также можно разделить на две группы.

К первой отнесем пакеты, предназначенные для решения сложных промышленных и научно-исследовательских задач большими производственными или научными коллективами. В таких проектах ведущую роль играет организация работ: хорошо налаженное взаимодействие между отдельными группами, быстрый доступ к многочисленным экспериментальным данным и библиотекам программ, тщательное документирование и тестирование, многовариантные расчеты. При этом обычно используются хорошо изученные готовые математические модели, которые лишь модифицируются и приспособляются для решения конкретных задач. Пользователи пакета подразделяются на две категории: разработчики библиотек готовых моделей и обычные пользователи, работа которых сводится к составлению схем из типовых блоков и параметрической настройке блоков. Пакеты первой группы условно назовем *"промышленными"*.

Совсем другая технология характерна для предварительных исследований, выполняемых отдельными учеными или проектировщиками. Библиотеки готовых моделей в этом случае используются весьма ограничено. Исходным материалом служат плохо формализованные "сырые" модели, т. е. модели, чьи свойства еще не вполне осознаны. Это означает, что необходимо уметь организовывать и поддерживать непрерывную обратную связь между исследователем и исследуемой моделью. Несмотря на большие достижения в области автоматического синтеза систем с заданными показателями, на практике разработка новой технической системы — это, прежде всего, просмотр большого числа пробных вариантов. Назовем пакеты второй группы *"исследовательскими"*, подчеркивая этим, что они уступают по количеству уникальных возможностей промышленным, зато более просты для освоения и доступны отдельному исследователю при решении относительно несложных задач из практически любой прикладной области. Под "несложными" будем понимать не простые задачи, а задачи посильные одному разработчику, не являющемуся специалистом в области программирования и вычислений. С "исследовательскими" пакетами тесно связана концепция активного вычислительного эксперимента, предусматривающая:

- визуализацию результатов моделирования не после эксперимента, а во время эксперимента;
- возможность интерактивного вмешательства пользователя в ход вычислительного эксперимента;
- возможность использования 2D- и 3D-анимации, в том числе интерактивной.

Активный вычислительный эксперимент является аналогом активного натурального эксперимента и позволяет исследователю "играть" с моделью, непосредственно видеть ее поведение, воздействовать на нее и наблюдать отклик.

Пакет автоматизации моделирования должен:

- обладать удобным для пользователя интерфейсом, позволяющим создавать математическую модель исследуемой системы с помощью используемого языка моделирования высокого уровня и обеспечивать контроль корректности этой модели;
- автоматически, по созданной математической модели, строить компьютерную модель;
- предоставлять пользователю возможность проводить активный вычислительный эксперимент с компьютерной моделью на том уровне абстракции, который удобен пользователю, и обеспечивать его корректное проведение.

## Языки моделирования

За последние десятилетия языки моделирования качественно изменились.

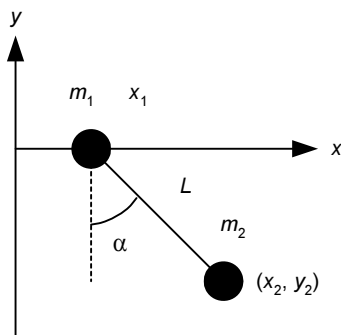
Первые языки моделирования (SLAM, Simula) представляли собой макрорасширения языков программирования (Фортран, Алгол) и с точки зрения технологии моделирования следовали за существовавшими тогда языками программирования.

Качественный скачок произошел с появлением объектно-ориентированных языков моделирования. Исторически они появились раньше объектно-ориентированных языков программирования. Однако и первые языки объектно-ориентированного моделирования, такие как Simula-67 и НЕДИС, по сути, предлагали пользователю язык программирования высокого уровня со специальными библиотеками классов.

В настоящее время считается, что язык моделирования должен быть графическим. Структура модели должна описываться различными типами иерархических блок-схем, схемами, состоящими из "физических" компонентов, а ее поведение — на языке, максимально приближенном к естественному математическому языку, в виде карт состояний, т. е. на языках, удобных человеку, а не машине. Обо всех этих формах мы поговорим подробнее дальше. Помимо чисто технических проблем, связанных с разработкой сложных визуальных редакторов, это требование приводит и к теоретическим проблемам. Повышение уровня абстракции входных языков и появление в них графических конструкций все более удаляет нас от достаточно сложных и жестких форм, присущих языкам программирования, на которых, в конце концов, и создается моделирующая программа.

Рассмотрим возникающие проблемы на примере моделей, поведение которых описывается системой дифференциальных уравнений. Существует много различных методов построения моделей динамических систем, приводящих к различным формам записи уравнений. Уравнения могут быть представлены в форме Коши. Это могут быть уравнения Лагранжа. Уравнения могут быть разрешены и не разрешены относительно старших производных. В случае использования законов сохранения к системе дифференциальных уравнений добавляются алгебраические. Долгие годы считалось, что в языках моделирования нужно разрешать только те формы записи уравнений, которые максимально приближены к формам, принятым в специализированных коллекциях, предназначенных для численного решения уравнений. В то же время пользователю гораздо удобнее записывать исходные физические зависимости в той форме, как они получены. Назовем это естественной формой. Рассмотрим различия этих форм на примере.

Пусть две точечные массы  $m_1$  и  $m_2$  соединены жестким стержнем длины  $L$  и подвешены таким образом, что точка подвеса массы  $m_1$  не закреплена и может двигаться по оси  $x$  (рис. 1.10).



**Рис. 1.10.** Пример физической системы со сложной динамикой

Ниже приведено описание этой системы с помощью уравнений Лагранжа:

$$\begin{cases} (m_1 + m_2) \frac{d^2 x_1}{dt^2} + m_2 L \cos \alpha \frac{d^2 \alpha}{dt^2} = m_2 L \sin \alpha \left( \frac{d\alpha}{dt} \right)^2, \\ \cos \alpha \frac{d^2 x_1}{dt^2} + L \frac{d^2 \alpha}{dt^2} = -g \sin \alpha, \\ x_2 = x_1 + L \sin \alpha, \\ y_2 = -L \cos \alpha. \end{cases}$$

И эту форму в данном случае следует считать естественной или свободной. Однако большинство программ для численного решения дифференциальных уравнений требует приведения данной системы к форме Коши, т. е. к системе уравнений первого порядка, разрешенных относительно первых производных.

Конечно, эта система уравнений может быть преобразована к форме Коши:

$$\left\{ \begin{array}{l} \frac{dx_1}{dt} = V_1, \\ \frac{d\alpha}{dt} = \omega, \\ \frac{d\omega}{dt} = \frac{K4 \cdot K3 + K1 \cdot K6}{K4 \cdot K2 - K5}, \\ \frac{dV_1}{dt} = \frac{K3 \cdot L + K2 \cdot K6}{K5 - K2 \cdot K4}, \\ K1 = m_1 + m_2, \\ K2 = m_2 \cdot L \cdot \cos \alpha, \\ K3 = m_2 \cdot L \cdot \sin \alpha \cdot \omega^2, \\ K4 = \cos \alpha, \\ K5 = K1 \cdot L, \\ K6 = g \cdot \sin \alpha. \end{array} \right.$$

Однако при выполнении даже такого несложного преобразования вручную велик риск допустить ошибки. Кроме того, теряется физический смысл уравнений.

Существование в языке моделирования графических операторов, позволяющих в удобной форме описывать структуру и поведение, ведет к необходимости автоматически приводить введенные пользователем или построенные пакетом уравнения к формам, с которыми могут работать библиотеки программных реализаций численных методов. Свободная форма уравнений приводит к еще одной очень важной задаче — задаче автоматического анализа выполнения условий существования и единственности решения. Современные программные реализации численных методов предполагают, что задачи хорошо поставлены и имеют решение.

Таким образом, современный пакет моделирования должен включать средства автоматического анализа выполнения условий, позволяющих утверждать, что написанная пользователем или созданная автоматически система уравнений имеет единственное решение и алгоритмы преобразования систем уравнений к формам, с которыми могут работать программные реализации численных методов. Такие средства еще только создаются.