

# **ОСНОВЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ. ПОСТРОЕНИЕ И АНАЛИЗ МОДЕЛЕЙ С ПРИМЕРАМИ НА ЯЗЫКЕ MATLAB**

Рассмотрены основные понятия и методы математического моделирования. Дана классификация основных структур математических моделей систем, а также подходов к выбору структуры и параметров модели. Описаны асимптотические и численные методы статического и динамического анализа моделей. Приведены описание и примеры применения программной среды MATLAB, ставшей международным стандартом учебного программного обеспечения в области математического моделирования, а также работающего в этой среде пакета ADAM для анализа дифференциально-алгебраических моделей.

Предназначено для аспирантов, слушателей факультетов повышения квалификации и студентов, изучающих курсы «Основы математического моделирования», «Методы исследования технических систем на ЭВМ», «Основы научных исследований», «Теоретические основы САПР», а также для преподавателей и инженеров, интересующихся вопросами математического моделирования.

## **ОГЛАВЛЕНИЕ**

Предисловие	3
<b>Часть I. ПОСТРОЕНИЕ МОДЕЛЕЙ</b>	
<b>1. Методология математического моделирования. Системный анализ</b>	<b>6</b>
1.1. Понятие системы. Примеры систем	6
1.2. Этапы системного анализа	10
1.3. Сложные системы и декомпозиция	16
1.4. Экспертные оценки	19
<b>2. Выбор структуры математической модели</b>	<b>25</b>
2.1. Классификация моделей	25
2.1.1. Статические и динамические модели	26
2.1.2. Дискретные и непрерывные модели	27
2.2. Модели состояния динамических систем	32
2.2.1. Модели общего вида	32
2.2.2. Линейные модели	33
2.2.3. Дискретизация и континуализация	36
2.3. Детерминированные и стохастические модели	39
2.4. Нечеткие модели	44
2.4.1. Нечеткие множества и лингвистические переменные	44
2.4.2. Нечеткие системы	48
2.4.3. Задачи группировки и упорядочения	52
2.4.4. Нечеткие числа	54
2.4.5. Вероятность или нечеткость?	57
2.5. Принципы выбора модели	58
<b>3. Выборы параметров математической модели</b>	<b>61</b>
3.1. Предварительные преобразования	61

3.1.1. Линейно параметризованные модели	61
3.1.2. Преобразование статических моделей	62
3.1.3. Преобразование динамических моделей	65
3.2. Регрессионный анализ и метод наименьших квадратов	70
3.3. Адаптивные модели и рекуррентные методы	73

## Часть II. АНАЛИЗ МОДЕЛЕЙ

<b>1. Анализ статики математических моделей систем</b>	<b>76</b>
1.1. Аналитические методы	78
1.1.1. Метод возмущений	79
1.1.2. Метод асимптотических разложений	80
1.2. Численные методы	81
1.2.1. Метод простых итераций	85
1.2.2. Метод Ньютона и его модификации	87
1.2.3. Метод продолжения по параметру	92
1.3. Особенности численного решения систем линейных алгебраических уравнений	93
<b>2. Анализ динамики математических моделей систем</b>	<b>96</b>
2.1. Аналитические методы	96
2.1.1. Метод возмущений	96
2.1.2. Метод асимптотических разложений	98
2.2. Численные методы	100
2.2.1. Одношаговые методы решения задачи Коши	104
2.2.2. Многошаговые методы решения задачи Коши	109
2.2.3. Методы решения систем дифференциальных уравнений, неразрешенных относительно производной	113
2.2.4. Методы решения линейных систем дифференциальных уравнений	114
2.2.5. Устойчивость численного интегрирования	118
2.2.6. Выбор шага интегрирования	122
2.2.7. Особенности интегрирования жестких систем	125
2.2.8. Выбор метода интегрирования	130
<b>3. Культура вычислений на ЭВМ</b>	<b>132</b>

## Часть III. РЕШЕНИЕ ЗАДАЧ В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB

<b>1. Базовые возможности системы MATLAB</b>	<b>140</b>
1.1. Начало работы	140
1.2. Действия с матрицами	141
1.2.1. Ввод и простейшие операции	141
1.2.2. Деление матриц	143
1.2.3. Модификация матриц	143
1.3. Условные операторы и циклы	144
1.4. Функции и М-файлы	145
1.4.1. Функциональные М-файлы	145
1.4.2. Скрипт-файлы	147

1.5. Графика	147
<b>2. Решение задач линейной алгебры</b>	<b>149</b>
<b>3. Решение задач численного анализа</b>	<b>152</b>
3.1. Решение систем нелинейных алгебраических уравнений	152
3.2. Решение обыкновенных дифференциальных уравнений	153
<b>4. Решение задач теории систем</b>	<b>155</b>
4.1. Расчет частотных и временных характеристик систем с обратной связью	155
4.2. Преобразование математических моделей	160
4.3. Вычисление матричной экспоненты	161
4.4. Оценка параметров модели	162
<b>5. Решение задач анализа динамики систем при помощи пакета АДАМ</b>	<b>164</b>
Приложение 1. Базовые функции среды MATLAB	168
Приложение 2. Функции тулбокса и «системы управления» (CONTROL SYSTEM TOOLBOX)	177
Заключение	180
Список литературы	183

## ПРЕДИСЛОВИЕ

Математическое моделирование — это методология научной и практической деятельности людей, основанная на построении, исследовании и использовании математических моделей объектов и процессов. Математическим моделированием занимался, в сущности, каждый, кто применял математику на практике, начиная от великих ученых древности и кончая школьником, решающим задачи «на составление уравнений». Однако в самостоятельную научную дисциплину математическое моделирование оформилось лишь в последние несколько десятилетий, в связи с распространением и широким применением ЭВМ. Нельзя отождествлять математическое моделирование с прикладной математикой, имеющей дело, как и любой раздел математики, лишь с математическими объектами. Предмет математического моделирования шире и связан, в первую очередь, с формализацией и интерпретацией формальных результатов. Иными словами, началом и концом процесса математического моделирования является окружающая действительность.

Центральным понятием математического моделирования является математическая модель (ММ) — совокупность математических объектов и отношений, отображающих объекты и отношения некоторой области реального мира (предметной области). Выделяют три этапа математического моделирования: построение ММ (формализация задачи), исследование (анализ модели) и использование (синтез решения). Этап формализации тесно связан с научно-инженерной дисциплиной, именуемой системным анализом [1—3]. На этом этапе решаются так называемые «прямые» задачи: по заданным значениям входов системы определяются ее выходы. Для этапа синтеза, наоборот, характерны «обратные» задачи: определение входов системы по заданным (желаемым) значениям ее выходов. Использование ММ возможно для различных целей: прогнозирования [4], исследования [5], проектирования [6, 7], управления [8, 9].

Сравнение большого числа различных подходов и методов показывает (см. [10—19]), что многие ММ и методики их применения не зависят от конкретной области приложения. Другими словами, одни и те же математические модели и методы могут одинаково применяться в различных областях и для различных целей. Это обстоятельство определяет внутреннее единство предмета математического моделирования и его место в системе подготовки специалиста. Курс «Математическое моделирование» («Применение математических методов для решения инженерных задач») должен опираться на курс высшей математики и предшествовать курсам специальной подготовки.

Однако подобные дисциплины в учебных планах большинства вузов появляются лишь на последних годах обучения или даже в планах постдипломной подготовки (аспирантура) и переподготовки (курсы повышения квалификации). Эта тенденция сохраняется и при переходе на многоступенчатую систему образования. С учетом этого и в силу ограниченности объема пособия выбран обзорный стиль изложения: более подробно даются определения и свойства понятий, а методы исследования (решения задач), как правило, лишь перечисляются со ссылкой на литературу. В то же время важной частью курса являются примеры. Ведь овладеть математическим моделированием — это значит научиться решать задачи, а, как известно, единственный способ научиться решать задачи — это решать их.

Пособие состоит из трех частей. Часть I является исправленным и дополненным вариантом [20] и посвящена вопросам построения математических моделей, выбора ее структуры и параметров. В части II излагаются методы асимптотического и численного анализа статических и динамических режимов непрерывных моделей, описываемых дифференциальными уравнениями. В пособии не затрагиваются вопросы анализа дискретных моделей, с которыми можно ознакомиться, например, по книгам [12, 13, 24, 26, 32], а также вопросы синтеза решения, которым посвящена обширная литература по методам оптимизации и принятия решений (см. например, [1, 6—9, 25—31]).

Третья часть посвящена программной поддержке математического моделирования. Современный подход к этой проблеме состоит в применении так называемых программных сред, предоставляющих пользователю набор функциональных (расчетных) и сервисных возможностей, допускающих рас-

ширение с учетом специфики решаемых задач. Описывается одна из наиболее популярных программных сред MATLAB, разработанная фирмой The MathWorks, Inc. и ставшая фактически международным стандартом учебного программного обеспечения в областях линейной алгебры, теории систем, теории управления, обработки сигналов и ряда других. Приводятся примеры решения задач в среде MATLAB, в том числе с применением пакета анализа дифференциально-алгебраических моделей ADAM. Пакет ADAM является оригинальной разработкой [61, 62] и обладает полной языковой и программной совместимостью с системой MATLAB. Язык пакета ADAM позволяет описывать нелинейные непрерывные и дискретные динамические системы со связями и задания на имитационное моделирование, линеаризацию и дискретизацию систем.

Материал учебного пособия основан на лекциях и практических занятиях, проводившихся авторами в течение ряда лет для студентов, аспирантов и на ФПК преподавателей по информатике Балтийского государственного технического университета. Главы 1, 2 части II написаны Д. Л. Егоренковым; часть I и глава 3 части II — А. Л. Фрадковым; часть III (кроме главы 5) — В. Ю. Харламовым; глава 5 части III — Д. Л. Егоренковым, А. А. Спиридоновым и А. Л. Фрадковым.

В учебном пособии отражены также некоторые результаты НИР, проводившихся по гранту 93—013—16322 Российского фонда фундаментальных исследований и по проекту НДС-10 программы «Нелинейные динамические системы: качественный анализ и управление» Госкомитета Российской Федерации по высшему образованию.

Учебное пособие может быть использовано при подготовке бакалавров, магистров и докторов, а также при переподготовке и повышении квалификации специалистов в курсах «Основы математического моделирования», «Основы научных исследований», «Применение математических методов в инженерных расчетах», «Теоретические основы САПР». Оно может оказаться полезным для инженеров и преподавателей, интересующихся применением математических методов и компьютеров в различных областях науки и технологии.

---

# Часть I

## ПОСТРОЕНИЕ МОДЕЛЕЙ

### 1. МЕТОДОЛОГИЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ. СИСТЕМНЫЙ АНАЛИЗ

#### 1.1. Понятие системы. Примеры систем

Базовым понятием математического моделирования является понятие *системы* [1—3, 21, 22]. Система в широком смысле — эквивалент понятия математической модели и задается парой множеств  $U, Y$  ( $U$  — множество входов,  $Y$  — множество выходов) и отношением\* на  $U \times Y$ , формализующим связь (зависимость) между входами и выходами.

Соединение систем также является системой и задается отношением. Например, последовательное соединение систем  $S_1 \subset U_1 \times Y_1$ ,  $S_2 \subset U_2 \times Y_2$  есть отношение  $S \subset U_1 \times Y_2$  такое, что  $(u_1, u_2) \in S$ , если существуют  $y_1 \in Y_1$ ,  $u_2 \in U_2$ , удовлетворяющие условиям  $(u_1, y_1) \in S_1$ ,  $(y_1, u_2) \in R$ ,  $(u_2, y_2) \in S_2$ , где  $R \subset Y_1 \times U_2$  — отношение, определяющее связь между\*\*  $y_1$  и  $u_2$ . Таким образом можно определять сколь угодно сложные системы, исходя из простых.

Приведенное определение отражает в абстрактном виде атрибуты (свойства), присущие нашему интуитивному представлению о системе: целостность и структурированность [1].

*Целостность* (единство) означает, что система отделена от внешней среды; среда может оказывать на нее действие (акцию) через входы и воспринимать отклик (реакцию) на эти действия через выходы.

*Структурированность* означает, что система разделена внутри на несколько подсистем, связанных и взаимодействующих.

\* Напоминаем, что отношением  $R \subset X \times Y$  (отношением между  $X$  и  $Y$ ) называется подмножество множества  $X \times Y$ , т. е. некоторая совокупность пар  $R = \{(x, y)\}$ , где  $x \in X$ ,  $y \in Y$ . Например, функциональная связь (зависимость)  $y = x^2$  может быть представлена как отношение между множествами  $X = (-\infty, \infty)$ ,  $Y = [0, \infty)$ , включающее пары  $(x, y)$ , для которых  $y = x^2$ .

\*\* В простейшем случае может быть  $Y_1 = U_2$ , а  $R$  — отношение тождества  $(y_1, u_2) \in R$ , если  $y_1 = u_2$ .

ющих между собой так же, как целая система взаимодействует с внешней средой.

Третье свойство, присущее системе, — *целенаправленность* — требует задания некоторой цели, достижение которой говорит о правильной работе системы.

Приведем для сравнения другие, менее формальные определения системы.

*Система* — объективное единство закономерно связанных друг с другом предметов, явлений, а также знаний о природе и обществе (БСЭ. Т. 39. С. 158).

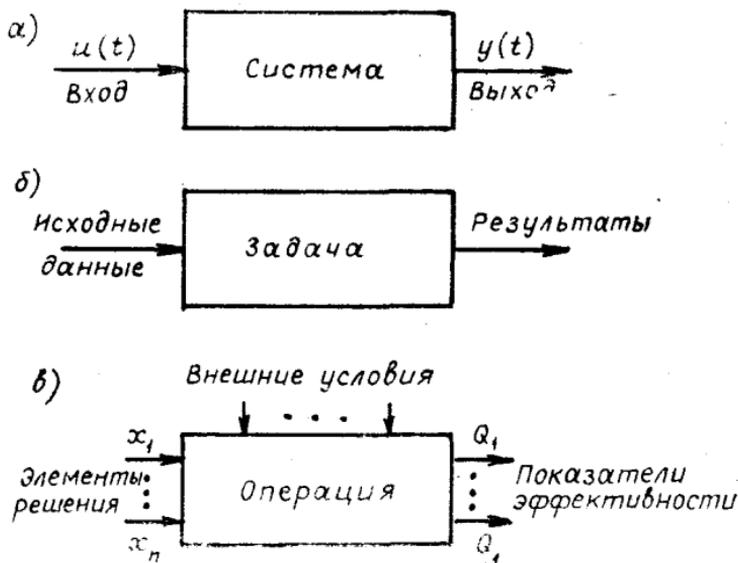


Рис. I. 1

Система — совокупность взаимосвязанных элементов (объектов, отношений), представляющих единое целое. Свойства системы могут отсутствовать у составляющих ее элементов [4].

Приведенное выше формальное определение весьма общо; под него подпадают практически все виды математических моделей систем: дифференциальные и разностные уравнения, регрессионные модели, системы массового обслуживания, конечные и стохастические автоматы, дедуктивные системы (исчисления) и т. д. Можно трактовать как систему любой преобразователь входных данных в выходные («черный ящик») (рис. I. 1, а). Например, системой можно назвать процесс ре-

шения любой задачи. При этом входами будут являться исходные данные, выходами — результаты, а целью — правильное решение (рис. I.1, б). Такой подход к системе подчеркивает ее целенаправленность и ведет свое происхождение от исследования операций [14] — научной дисциплины, занимающейся разработкой количественных методов обоснования решений. Основное понятие здесь — *операция*: действие, которое подвергается исследованию (проектирование, конструирование, управление, экономическая деятельность и т. д.). Операция соответствует некоторой системе. Входами этой системы являются элементы принимаемого решения о проводимой операции, выходами — результаты проведения операции (показатели ее эффективности (рис. I.1, в)). Для развития навыков системного подхода полезно искать примеры систем в окружающем мире. Некоторые примеры представлены в табл. I.1.\*

Подчеркнем, что функционирование системы — это процесс, разворачивающийся во времени, т. е. множества возможных входов и выходов  $U, Y$  — это множества функций времени со значениями соответственно в множествах  $U, Y$ :

$$U = \{u : T \rightarrow U\}, \quad Y = \{y : T \rightarrow Y\},$$

где  $T$  — множество моментов времени, на котором рассматривается система.

Система называется *функциональной (определенной)*, если каждой входной функции  $u(t)$  соответствует единственная выходная функция  $y(t)$ . В противном случае система называется *неопределенной*. Неопределенность обычно возникает из-за неполноты информации о внешних условиях работы системы. Важным свойством, присущим реальным системам, является *причинность*. Она означает, что если входные функции  $u_1(s)$  и  $u_2(s)$  совпадают при  $s \leq t$ , т. е.  $u_1(s) = u_2(s)$  при  $s \leq t$ , то соответствующие выходные функции удовлетворяют условию  $y_1(t) = y_2(t)$ , т. е. «настоящее не зависит от будущего при заданном прошлом».

Числовые величины, связанные с системой, делятся на переменные и параметры. *Параметры* — это величины, которые можно считать постоянными на промежутке времени рассмотрения системы. Остальные числовые величины являются *переменными*. Значения переменных и параметров опре-

\* Примеры 12—15 в табл. I.1 придуманы в 1988 г. учеником 86 класса 292 школы М. Коротеевым.

Система	Вход	Выход	Цель
Радиоприемник	Радиоволны	Звуковые волны	Неискаженный звук
Проигрыватель	Колебания иглы	"	"
Термометр	Т° воздуха (T)	Высота столбика (h)	Верное показание
Водопроводный кран	Поворот ручки (угол φ)	Струя воды (расход G)	Заданный расход
Ученик	Лекция учителя, текст в учебнике, книги, кино, телевизор	Отметки, знания, поступки	Хорошие отметки, хорошие поступки, хорошие знания
Учитель	План урока, ответы учеников	Лекции, задачи для контрольной, отметки	"
Робот	Команды	Движения	Точное исполнение команд
Популяция зайцев в лесу	Пища	Численность	Максимальная численность
Популяция лис в лесу	"	"	"
Программа ЭВМ решения уравнения $ax^2 + bx + c = 0$	Коэффициенты $a, b, c$ . Точность $E$	$x_1, x_2$	Решение с заданной точностью
Задача решения уравнения $ax^2 + bx + c = 0$	$a, b, c$	Формула $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	Правильная формула
Электромотор	Электрический ток	Вращение ротора	Вращение с заданной частотой
Костер	Дрова	Тепло, свет	Заданное количество тепла и света
Торговля	Продукты, вещи	Деньги	Получение суммы денег = стоимости товара
Бюрократ	Бумажка	Бумажка	Зарплата

деляют количественную информацию о системе. Оставшаяся информация, т. е. качественная, определяет *структуру* системы. Различие между переменными и параметрами, а также между параметрами и структурой может быть условным, однако оно полезно в методическом отношении. Так, типовым приемом построения ММ системы является *параметризация* — выбор в качестве ММ семейства функций, зависящих от конечного (обычно небольшого) количества чисел — параметров.

## 1.2. Этапы системного анализа

Системный анализ в широком смысле — это методология (совокупность методических приемов) постановки и решения задач построения и исследования систем, тесно связанная с математическим моделированием. В более узком смысле системный анализ — методология формализации сложных (трудно формализуемых, плохо структурированных) задач. Системный анализ возник как обобщение приемов, накопленных в задачах исследования операций и управления в технике, экономике, военном деле. Соответствующие модели и методы заимствовались из математической статистики, математического программирования, теории игр, теории массового обслуживания, теории автоматического управления. Фундаментом перечисленных дисциплин является теория систем [21—22].

Остановимся на различии в употреблении терминов «системный анализ» и «системный подход» [4]. Системный анализ — это целенаправленная творческая деятельность человека, на основе которой обеспечивается представление исследуемого объекта в виде системы. Системный анализ характеризуется упорядоченным составом методических приемов исследования. Что касается термина «системный подход», то традиция его применения связывает его с исследованиями, проводимыми многоаспектно, комплексно, с разных сторон изучая предмет или явление. Этот подход предполагает, что все частные задачи, решаемые на уровне подсистем, должны быть увязаны между собой и решаться с позиции целого (принцип системности). Системный анализ — более конструктивное направление, содержащее методику разделения процессов на этапы и подэтапы, систем на подсистемы, целей на подцели и т. д.

В системном анализе выработана определенная последовательность действий (этапов) при постановке и решении задач, которую будем называть алгоритмом (методикой) системного анализа (рис. I.2). Эта методика помогает более осмысленно и грамотно ставить и решать прикладные задачи. Если на каком-то этапе возникают затруднения, то

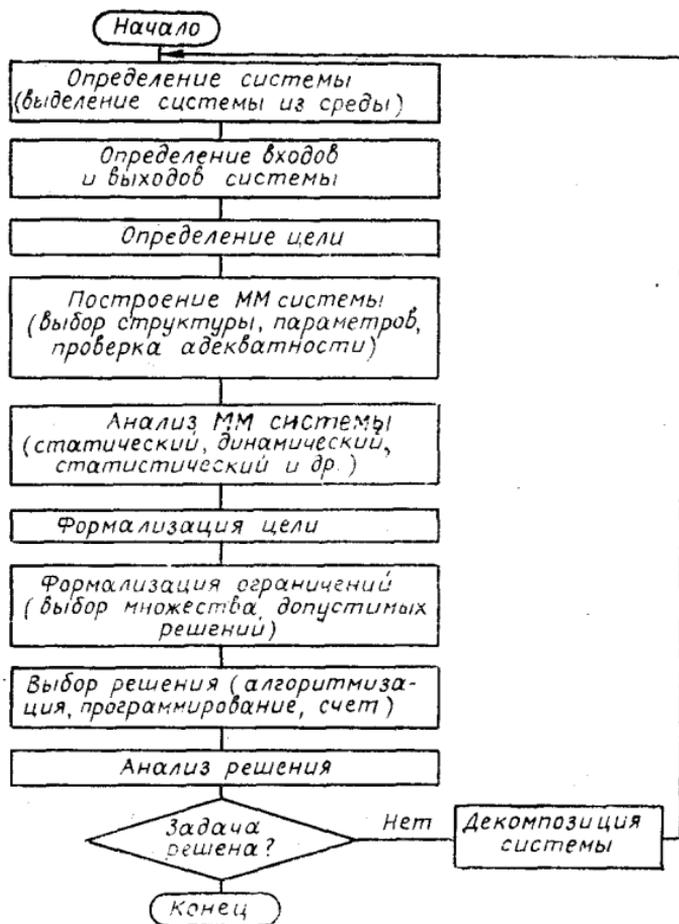


Рис. I.2

нужно вернуться на один из предыдущих этапов и изменить (модифицировать) его. Если и это не помогает, то это значит, что задача оказалась слишком сложной и ее нужно разбить на несколько более простых подзадач, т. е. провести декомпозицию (см. подразд. I.3). Каждую из полученных подзадач решают по той же методике. Для иллюстрации применения методики системного анализа приведем пример [18].

**Пример.** Рассмотрим автомобиль, находящийся перед гаражом на некотором расстоянии от него (рис. 1.3, а). Требуется поставить автомобиль в гараж и сделать это, по возможности, наилучшим образом. При решении попытаемся руководствоваться алгоритмом системного анализа (см. рис. 1.2).

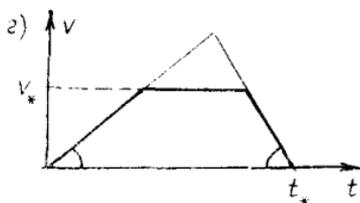
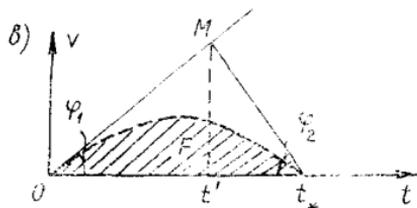
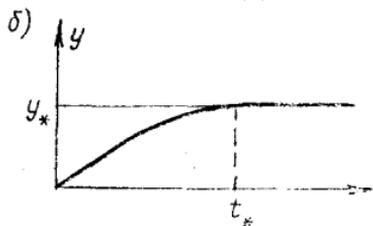
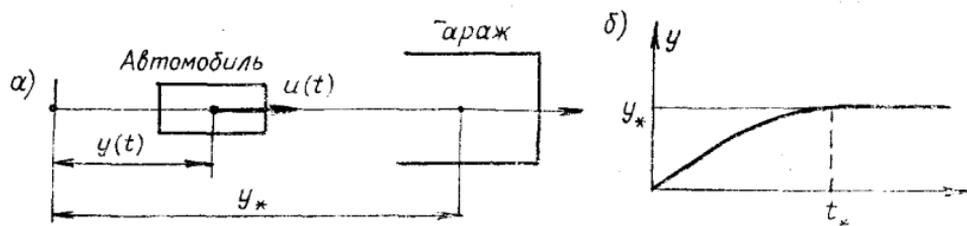


Рис. 1.3

**Этап 1.** Система: автомобиль и гараж (автомобиль, приближающийся к гаражу).

**Этап 2.** Вход: сила тяги двигателя. Выход: пройденный путь.

**Этап 3.** Цель: автомобиль должен проехать заданный путь и затормозить.

**Этап 4.** Построение ММ начинается с обозначения всех величин (переменных и постоянных), существенных для задачи. Введем следующие обозначения:

$u(t)$  — сила тяги в момент времени  $t$  (вход);

$y(t)$  — путь, пройденный к моменту  $t$  (выход);

$y_*$  — расстояние от автомобиля до гаража (параметр).

Затем выписываются все уравнения и соотношения, существующие между введенными величинами, как в школьных задачках на составление уравнений. Если возможных уравнений несколько, выбирают простейшее. В нашей задаче — это уравнение динамики (2-й закон Ньютона):

$$\frac{md^2 y(t)}{dt^2} = u(t), \quad (1.1a)$$

где  $m$  — масса автомобиля, а также начальные условия \*

$$y(0) = 0, \quad \dot{y}(0) = 0. \quad (1.1 \text{ б})$$

*Этап 5.* Модель (1.1) достаточно хорошо изучена и в детальном анализе не нуждается. Укажем лишь, что она адекватна, если можно пренебречь размерами автомобиля, ограничением на его мощность, силами трения и сопротивления и другими более второстепенными факторами.

*Этап 6.* Простейший вариант формализации цели

$$y(t_*) = y_*, \quad (1.2)$$

где  $t_*$  — момент остановки — оказывается неудовлетворительным, поскольку в (1.2) не формализовано само требование остановки  $\dot{y}(t_*) = 0$  и, значит, неясно, как система будет вести себя при  $t > t_*$ . Правильнее задать цель соотношением

$$y(t) = y_* \text{ при } t \geq t_*, \quad (1.3)$$

из которого следует, в частности, что  $\dot{y}(t) = 0$  при  $t > t_*$ .

На первый взгляд, задача поставлена и можно переходить к ее решению, т. е. к этапу 8. Но, оказывается, однозначного решения задача не имеет: здравый смысл говорит о том, что существует бесконечно много способов достичь цели (1.3). Значит, нужно дополнить цель правилом отбора способов, позволяющим отвечать на вопрос: какой способ лучше. Задан следующий разумный правилом: тот способ считается лучшим, который быстрее приводит к цели. Формально новую цель можно записать так:

$$\min\{t_* : y(t) = y_* \text{ при } t \geq t_*\}. \quad (1.4)$$

Но теперь физические соображения показывают, что решение поставленной задачи тривиально: искомый минимум в (1.4) равен нулю! Действительно, выбрав достаточно большую силу тяги, можно придать автомобилю как математическому объекту, описываемому ММ (1.1), сколь угодно большое ус-

---

\* Здесь и далее через  $\dot{y}(t)$  будет обозначаться производная по времени от функции  $y(t)$ . Будет также использоваться обозначение  $y(t) = \rho y(t)$ .

корение и сколь угодно быстро\* переместить его на любое заданное расстояние. Видимо, требуется ввести какие-то ограничения, исключающие бессмысленные решения. Можно было бы усложнить ММ системы: учесть ограниченную мощность двигателя, его инерционность, силы трения и т. д. Однако разумнее попытаться остаться в рамках ММ (1.1), (1.4), введя дополнительно лишь ограничения на силу тяги:

$$-a \leq u(t) \leq b. \quad (1.5)$$

Таким образом, чтобы придать задаче смысл, нам пришлось возвратиться на этап 7.

*Этап 8.* Для решения задачи можно было бы применить мощный и хорошо разработанный аппарат теории оптимального управления (вариационное исчисление, принцип максимума Понтрягина и др., см., например [9, 35]). Однако сначала надо попытаться решить задачу элементарными средствами. Для этого часто бывает полезно перейти к геометрической интерпретации задачи, чтобы привлечь нашу геометрическую интуицию. Естественная интерпретация (рис. 1.3, б) не дает ключа к решению, так как не позволяет в удобной форме представить ограничения на допустимые траектории движения автомобиля. Дело меняется коренным образом, если перейти к другой ММ. Введем новую переменную:  $v(t) = \dot{y}(t)$  (скорость). Тогда вместо (1.1) возникает уравнение

$$mv = u, \quad v(0) = 0, \quad (1.6)$$

цель (1.4) запишется в виде

$$\min \{t_* : \int_0^{t_*} v(s) ds = y_*, \text{ при } t > t_*\}, \quad (1.7)$$

а ограничения (1.5) превратятся в ограничения на скорость изменения новой переменной:

$$-a/m \leq \dot{v}(t) \leq b/m. \quad (1.8)$$

---

\* Математик здесь, конечно, заметит, что минимум в (1.4), строго говоря, может не достигаться и в формулировке (1.4) нужно заменить  $\min$  на  $\inf$  («инфимум» — точная нижняя грань множества). Однако ситуация от этого не изменится: формализация в данном случае не отражает сущности задачи, т. е. проведена неверно. В дальнейшем, чтобы не «пугать» инженера, мы будем пользоваться обозначениями  $\min$ ,  $\max$ , имея в виду, что при необходимости их следует заменить на более общие  $\inf$ ,  $\sup$ .

Итак, мы изменили выход системы, из-за чего пришлось заново пройти этапы 2—7.

Геометрическая интерпретация движения системы (1.6) — (1.8) в плоскости  $\{v, t\}$  изображена на рис. 1.3, в. Из него видно, что для решения задачи нужно найти кривую  $v(t)$  ( $t \geq 0$ ) с заданной площадью фигуры  $F$  под ней и наименьшей возможной координатой правого конца  $t_*$ , лежащую в треугольнике  $OMN$  с заданными углами наклона  $\varphi_1, \varphi_2$  боковых сторон (в соответствии с (1.8)  $\operatorname{tg} \varphi_1 = b/m, \operatorname{tg} \varphi_2 = a/m$ ).

Геометрическое решение очевидно: фигура  $F$  должна заполнять весь треугольник  $OMK$ . Это значит, что автомобиль должен двигаться с максимальным ускорением до некоторого момента  $t'$ , затем включить максимальное торможение и в момент  $t_*$  выключить двигатель. Формулы для определения момента переключения  $t'$  выводятся из элементарного расчета треугольника  $OMK$  по заданной площади и углам. Они имеют вид

$$t' = \sqrt{\frac{2ma \cdot *}{b(a+b)}}, \quad t_* = \sqrt{\frac{2n(a+b)y_*}{ab}}. \quad (1.9)$$

Рассмотренная геометрическая модель позволяет решать и более сложные задачи. Например, если по соображениям безопасности нужно учесть ограничение на максимальную скорость:  $|y(t)| \leq v^*$ , то решение легко усмотреть из рис. 1.3, г: график оптимальной траектории представляет собой трапецию.

Еще более сложные задачи (например, при введении ограничений на расход топлива в виде  $\int_0^{\infty} |u(t)| dt \leq A$ ) не имеют простого аналитического решения, подобного (1.9), и практически решаются лишь численно, с привлечением математического аппарата приближенной минимизации функционалов (см., например, [35. Гл. 9]). Однако и для них решение упрощенной задачи не теряет важности, поскольку оно позволяет получить начальное приближение к решению сложной задачи, установить качественные свойства решения сложной задачи, выявить факторы, наиболее сильно влияющие на решение сложной задачи, и, главное, соотнести результаты математического исследования со здравым смыслом.

Резюмируя сказанное, можно дать совет изучающему математическое моделирование: «не решай сложную задачу, не решив сначала более простую!».

### 1.3. Сложные системы и декомпозиция

Как известно, системный анализ родился как метод исследования и проектирования сложных систем. Что же такое сложная система? Понятие это неформальное, и обычно, говоря о сложных системах, перечисляют их основные особенности:

— наличие большого числа разнородных элементов (подсистем);

— сложный характер, неоднородность связей между подсистемами;

— сложность функций, выполняемых системой;

— наличие неопределенности в описании системы;

— сложность определения (организации) требуемого управляющего воздействия на систему и т. д.

Однако понятно, что каждая из этих особенностей может оказаться существенной или несущественной, все зависит от конкретной ситуации и целей исследования. Поэтому более универсальный способ выделения класса сложных систем связан со сложностью самого процесса исследования системы.

Если методика системного анализа (см. рис. 1.2) непосредственно приводит к успеху, то нет оснований называть систему «сложной». Введение этого термина оправдано, если решить задачу в исходном виде не удастся. В этом случае она разбивается на несколько вспомогательных подзадач, решаемых по отдельности. Такой прием называется *декомпозицией* и является основным методом исследования сложных систем.

При декомпозиции исходная система делится на подсистемы, а цель — на подцели. Далее для решения каждой подзадачи пользуются той же методикой, что и для всей системы. Если в ходе решения (а возможно, и до того) какие-то из подзадач окажутся слишком сложными, то снова проводится декомпозиция: возникают подзадачи следующего уровня и т. д. Результатом этого процесса является структуризация: исходная система приобретает иерархическую (многоуровневую) структуру\* [1—3, 21]. Соответствующая структура возникает и в множестве подцелей; она называется деревом

---

\* Здесь термин «структура» используется в смысле, несколько более узком, чем в подразд. 1.1, и означает состав подсистем в системе и типы связей между ними.

целей (рис. I.4), поскольку представляет собой граф\* типа дерева (без циклов).

Вообще теория графов является естественным математическим аппаратом описания сложных систем [3, 23, 24]. Действительно, каждой сложной системе ставится в соответствие граф (*структурный*), вершинами которого являются подсистемы, а дугами — имеющиеся между ними связи. Если связи направленные, т. е. наличие связи  $S_i \rightarrow S_j$  означает, что воздействие  $S_i$  на  $S_j$  не вызывает обратного воздействия или им можно пренебречь, то граф системы является ориентированным (направленным). К этому классу относятся, например, структурные схемы (граф-схемы) систем автоматиче-

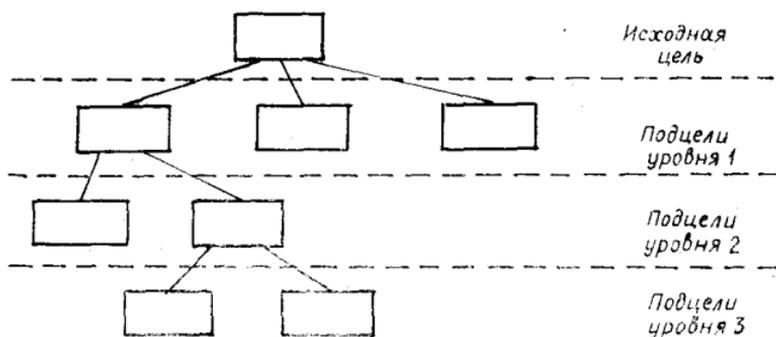


Рис. I.4

ского управления. У других систем влияние связанных подсистем обоюдное, и они описываются неориентированными графами (например, сложные электрические и электронные схемы).

Приведенное понятие декомпозиции вполне соответствует идее структурного программирования. Создание сложных программных систем — одна из важнейших областей применения системного анализа. Вопросы математического обоснования декомпозиции рассматривались в [1, 2, 24]. Отметим лишь, что раздробление системы на подсистемы обычно проводится по принципу «слабых» связей, т. е. так, чтобы связи

\* Графом называется пара  $(G, R)$ , где  $G = \{g_1 \dots g_n\}$  — конечное множество вершин, а  $R \subset G \times G$  — бинарное отношение на  $G$ . Если  $(g_i, g_j) \in R$ , тогда и только тогда, когда  $(g_j, g_i) \in R$ , то граф называется неориентированным, в противном случае — ориентированным. Пары  $(g_i, g_j) \in R$  называются дугами (ребрами), а элементы  $g_i$  множества  $G$  — вершинами графа.

между подсистемами были слабее, чем связи между элементами каждой подсистемы.

В сложных системах часто приходится проводить несколько вариантов декомпозиции и соответственно строить несколько деревьев целей. Это обычно связано с наличием нескольких критериев функционирования системы. Возникающие при этом задачи многокритериального выбора изучаются в теории принятия решений [25]. Успех декомпозиции часто определяется интуицией и опытом исследователя. Человек, по данным психологов, может мысленным взором охватить структуру декомпозированной системы, если на каждом уровне возникает не более чем  $5 \pm 2$  подзадач.

**Пример.** Система подготовки специалиста в вузе.

Пусть перед нами поставлена цель — повысить эффективность подготовки специалистов в вузе. Как ее достичь — сразу неясно, нужно проводить декомпозицию. Но как? Напрашивается разбиение по курсам, на шесть подсистем. Входы и выходы подсистем — показатели (например, показатели успеваемости, контрольных проверок и т. д.) в начале и конце соответствующего года обучения; подцели — обеспечение высоких значений показателей эффективности на выходах. Если оказывается, что принимаемые меры по курсам (в деканатах) малоэффективны, то проводится дальнейшая декомпозиция: по группам и по дисциплинам, выделяются группы и предметы, требующие применения воздействий. Следующий шаг декомпозиции — индивидуальный подход к студенту: возможные меры воздействия — проведение дополнительных занятий, использование ТСО и т. д.

Можно проводить декомпозицию и по-другому: сначала по факультетам и кафедрам, а затем по специализациям, дисциплинам и преподавателям. Подцели при этом связаны с повышением квалификации преподавателей и соответственно требуются другие меры воздействия, приводящие к улучшению других показателей функционирования системы.

В перечисленных вариантах способ декомпозиции был определен сложившейся структурой системы (процесса обучения). Но так бывает не всегда. Например, если анализировать ту же систему по критериям гуманитарной подготовки студента, то разбиение на подсистемы оказывается неоднозначным, а показатели функционирования подсистем — не вполне определенными.

## 1.4. Экспертные оценки

На начальных этапах, связанных с формализацией и предварительной алгоритмизацией решаемой задачи, исследователю, как правило, приходится иметь дело с неточной, неполной и субъективной информацией. Поэтому важно уметь пользоваться существующими методами ее сбора, согласования и обработки. Наиболее развитыми из подобных методов на сегодня оказались методы *экспертных оценок* [19, 25, 38].

Теория экспертных оценок зародилась в психологии и социологии, но в настоящее время применяется далеко за их пределами, позволяя решать широкий класс практических задач, требующих привлечения знаний и опыта экспертов, от которых поступает интересующая исследователя субъективная информация. Использование экспертных оценок, включающих в себя специально разработанные приемы сбора, согласования и математической обработки сведений, получаемых от эксперта или целенаправленно сформированной группы экспертов, является одним из наиболее простых, доступных и достаточно надежных способов получения и дальнейшего анализа априорной информации. Результаты, полученные с привлечением методов экспертных оценок, оказываются весьма полезными (например, на стадии разработки эскизного проекта и технического задания на проектирование АСУП и АСУ ТП). В ряде случаев в условиях существенной априорной неопределенности они могут представлять самостоятельный практический интерес в связи с отсутствием либо значительной сложностью получения объективных сведений о поведении объекта исследования.

Экспертные оценки следует трактовать шире, чем просто сбор и усреднение произвольно собранных точек зрения, что действительно является некорректным. Кроме того, необходимо иметь в виду, что последнее слово всегда остается за исследователем, принимающим окончательное решение об использовании результатов экспертизы, которую следует рассматривать как способ получения дополнительной информации, позволяющей уменьшить неопределенность, имеющую место при решении задачи.

В большинстве прикладных исследований методы экспертных оценок используются для выделения существенных (в том или ином смысле) факторов и их ранжирования. При этом чаще всего применяют методы опроса, основанные на

прямом (непосредственном) ранжировании, а полученные субъективные мнения обрабатывают методами ранговой корреляции. Однако область применения экспертных оценок, как и арсенала методов сбора согласования и обработки субъективной информации, намного шире. Можно указать следующие технико-экономические задачи, в которых их использование дает определенную практическую пользу:

1. Выбор целей исследования.
2. Выбор и построение критериев в задачах векторной оптимизации.
3. Принятие решений при управлении производством и выбор наилучшего варианта решения любой достаточно сложной проблемы в условиях неопределенности.
4. Задачи идентификации, начиная от выбора структуры модели и определяющих факторов и кончая приближенным построением зависимостей и их интерпретацией.
5. Построение эвристических алгоритмов управления.
6. Эргонометрические исследования.
7. Оценка качества продукции.
8. Системы обучения, основанные на построении различного вида сценариев и целенаправленном использовании результатов опроса.
9. Планирование производства, НИР и ОКР.
10. Классификация однотипных объектов по степени выраженности тех или иных характерных свойств.
11. Прогнозирование научно-технического прогресса.

Экспертные оценки не следует рассматривать как некоторый «нулевой» цикл исследования; ему предшествует (по крайней мере должен предшествовать) детальный априорный анализ, который (по крайней мере, должен предшествовать) детальный априорный анализ сведений позволяет целенаправленно и планомерно организовать экспертизу, ориентированную на извлечение новой информации. Экспертизу чаще всего следует рассматривать как последний, наиболее формальный шаг априорного анализа. При этом получают приближенную информацию по данной задаче на языке ее окончательного решения, которая в дальнейшем может использоваться в виде ограничений, отправных точек процесса оптимизации и т. д. Однако в некоторых случаях результаты экспертизы могут быть использованы и как самостоятельные решения.

Существует большое количество методов сбора и последующей обработки субъективной информации, которой располагают опытные специалисты, выбранные в качестве экс-

пертов. Классификация методов получения экспертных оценок показана на рис. 1.5 [38].

Методы коллективной работы экспертной группы (иногда их называют способами прямого получения коллективного мнения) предполагают получение обобщенного мнения в ходе совместного обсуждения и решения поставленных проблем группой специалистов, которые находятся в непосредственном контакте.

Всем методам получения коллективных экспертных оценок присущи недостатки, возникающие из-за всевозможных трудностей организационного характера. Очевидно, решение многих проблем требует участия в экспертизе ведущих специалистов в данной области. Однако собрать их вместе для участия в коллективном обсуждении практически не представляется возможным. Решить многие организационные проблемы удастся путем получения коллективного мнения на основании совместной обработки независимых индивидуальных мнений членов экспертной группы специальными методами опроса. Следует отметить, что использование методов получения индивидуального анонимного мнения членов экспертной группы способствует также устранению недостатков, обусловленных влиянием различных психологических факторов, при непосредственном контакте с экспертами. Правильное применение методов опроса позволяет повысить количество и качество исходной информации, используемой для получения обобщенного мнения. Существуют два основных вида опроса: интервью и анкетные опросы. Третий, промежуточный, который мы условно назвали «анкетированием с участием интервьюера», сочетает идеи первых двух.

Наиболее характерной особенностью интервью как специфической формы опроса является то, что исследователь и эксперт находятся в непосредственном контакте. Необходимая информация извлекается в ходе беседы, направление и организация которой определяются проблемой, представляющей собой область конкретного исследования.

Метод анкетирования заключается в том, что эксперту предлагается для заполнения анкета, содержащая набор вопросов, каждый из которых логически связан с главной задачей исследования.

Сравнительную оценку предлагаемых для анализа объектов можно осуществлять несколькими приемами. Один из них предполагает оценивание экспертами вероятности предпочтения (вероятностные оценки) одного из вариантов (объектов)

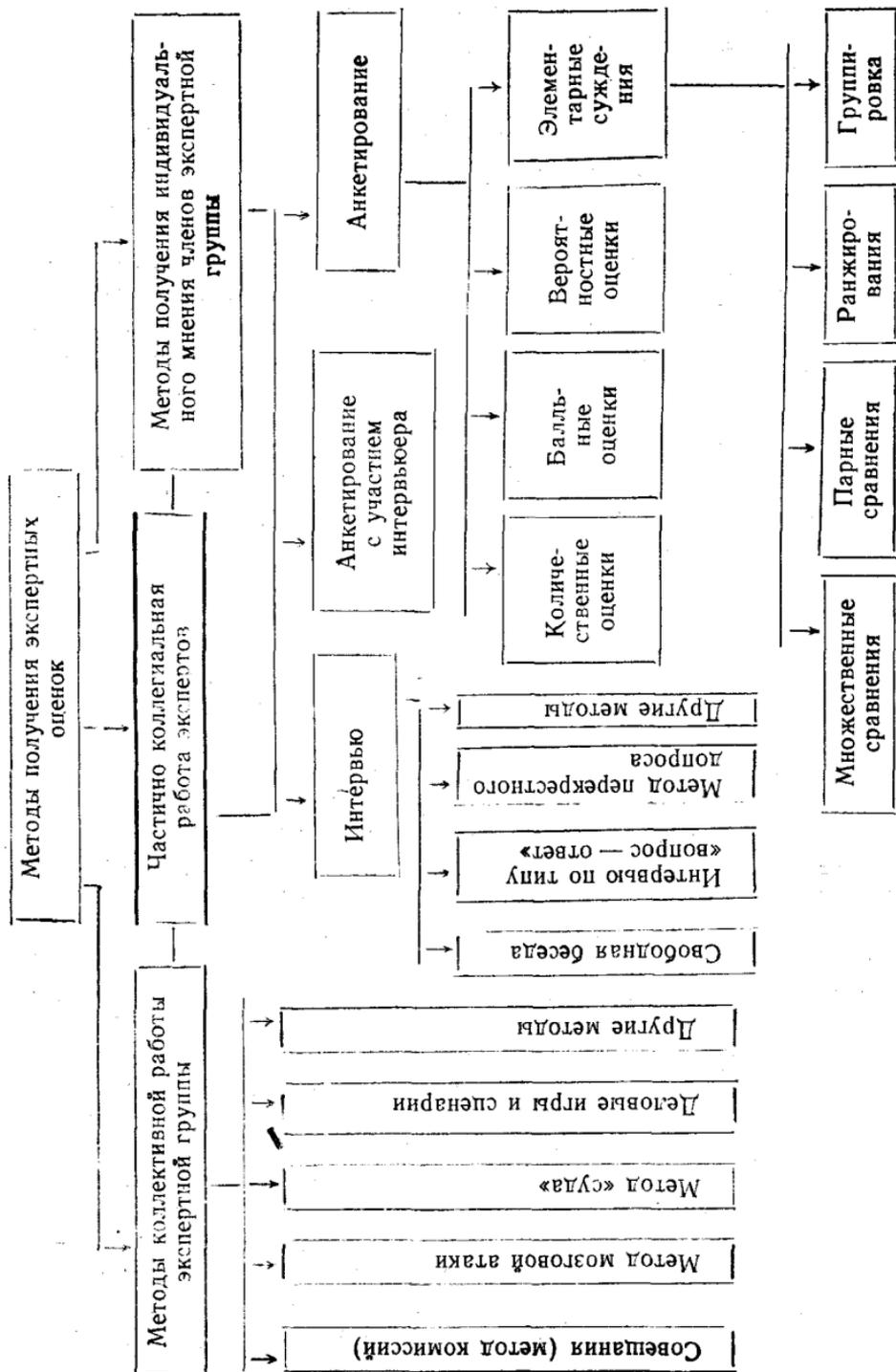


Рис. 1.5

другим. Однако использование подобного способа на практике часто приводит к определенным затруднениям, поскольку если маловероятные или, напротив, весьма вероятные события оценить довольно легко, то варианты с вероятностью предпочтения около 0,5 — сложно. Кроме того, следует иметь в виду, что понятие вероятности является весьма абстрактным и может в ряде случаев запутать эксперта, так как он обычно не представляет область своих, зачастую строго детерминированных понятий в вероятностной трактовке. Указанные трудности ограничивают использование этого способа в экспертных оценках. Улучшения результатов при оценке субъективных вероятностей можно достичь, используя процедуры, основанные на косвенном получении информации о вероятностях, либо осуществляя перед основным экспериментом ряд специально продуманных тренировочных опросов, позволяющих выработать у экспертов необходимые навыки.

В некоторых работах описываются методы, при использовании которых экспертам предлагается указать меру ценностей или различия в ценности анализируемых альтернатив путем их «взвешивания», скажем, по 10-балльной шкале. Наиболее важному, с точки зрения эксперта, фактору дается полная оценка — 10, а остальным — определенная доля полной оценки. Однако в данном случае возникает проблема обеспечения единообразного понимания всеми участниками опроса критерия оценки и достижения сопоставимости оценочных шкал отдельных экспертов, без чего невозможно получить обобщенные мнения. Эта проблема возникает потому, что эксперты неоднозначно воспринимают искусственно наведенную им балльную шкалу. В связи с этим метод «взвешивания» представляется недостаточно надежным и эффективным.

В ряде случаев заполнение анкеты требует от эксперта количественных заключений на безразмерной шкале или в единицах какого-то параметра. В первом случае имеют место те же трудности, что и при получении балльных оценок, во втором — существенно усложняется работа экспертов, так как при решении достаточно сложных проблем нередко даже высококвалифицированные эксперты не располагают необходимыми субъективными количественными оценками. Это приводит их к необходимости «переводить» свои сложившиеся в результате накопленного опыта качественные представления о существовании вопросов в количественную форму.

Хотя получение такого рода оценок бывает наиболее трудоемким, в ряде случаев, например, при прогнозировании научно-технического прогресса, они оказываются необходимыми.

Подробнее с различными методами экспертиз, основанными на получении количественных, балльных и вероятностных оценок, можно познакомиться в [19].

Представляется наиболее приемлемым использование методов анкетирования, основанных на элементарных суждениях о предпочтении объектов (будь то признаки или варианты решения), предъявляемых экспертам для оценки (см. рис. 1.5).

Важно отметить, что методы элементарных суждений являются достаточно надежными приемами получения экспертных оценок, так как они в наибольшей степени свободны от влияния побочных факторов на качество получаемой информации и требуют от специалиста только высказывания его мнений об оцениваемом объекте, не отвлекая его внимания на решение других задач, например, определения вероятностных оценок исследуемых объектов и т. п. С нашей точки зрения, методы, основанные на элементарных суждениях о предпочтении объектов, предъявляемых экспертам для анализа, наиболее перспективны для получения индивидуальных экспертных оценок. Естественно, что описываемые здесь подходы не исчерпывают всего богатого арсенала методов анкетирования, предназначенных для решения весьма широкого круга задач. Описание других процедур анкетирования дано, например, в работах [19, 38].

На рис. 1.5 показана группа методов, основанная на частично коллегиальной работе экспертов. Эти приемы экспертизы сочетают положительные качества коллективной работы и методов опроса. Смысл их заключается в том, что сбор субъективной информации осуществляется по одному из выбранных методов опроса (при этом, очевидно, более подходящими будут приемы анкетирования), но роль отдельных экспертов в этом случае выполняют небольшие коллективы специалистов, работа внутри которых может быть организована по одному из указанных выше принципов.

Из приведенной классификации выпало большое количество процедур типа Дельфи, Квест, Поспелова, Глушкова и т. д. Это специализированные методы, предназначенные для решения задач научно-технического прогнозирования и планирования производства, НИР и ОКР. Каждый из них представляет собой некоторую комплексную экспертизу, осуществ-

вляемую в несколько этапов (туров). Многоэтапность преследует цель добиться декомпозиции общей сложной проблемы на ряд подзадач, решение которых осуществляется с привлечением простых процедур экспертного опроса. Кроме того, она направлена на повышение согласованности мнений экспертов и может способствовать получению аргументированных суждений о правомерности отдельных мнений.

## 2. ВЫБОР СТРУКТУРЫ МАТЕМАТИЧЕСКОЙ МОДЕЛИ

### 2.1. Классификация моделей

Этап построения ММ системы разбивается на две части: выбор структуры и выбор параметров. Как было сказано в подразд. 1.1, структура сложной системы определяется типами моделей каждой ее подсистемы и характером связей (отношений) между ними. Все многообразие имеющихся типов ММ можно классифицировать по нескольким основным признакам (см. табл. I.2): статические — динамические; дискретные — непрерывные; детерминированные — стохастические — нечеткие; сосредоточенные — распределенные; стационарные — нестационарные; линейные — нелинейные и т. д.

Таблица I.2

Математические модели систем

Статические $y(t) = F(u(t))$	Динамические $y(t) = F(\{u(s) - \infty < s \leq t\})$
Дискретные ( $U, Y, T$ — конечные (счетные) множества)	Непрерывные ( $U, Y, T$ — континуумы (прямые, отрезки, тела в многомерных пространствах))
Детерминированные	Стохастические Нечеткие
Сосредоточенные (конечные* уравнения, разностные уравнения, обыкновенные дифференциальные уравнения)	Распределенные (уравнения с запаздыванием, в частных производных, интегральные уравнения)
Линейные	Нелинейные
Стационарные (параметры не меняются во времени)	Нестационарные (параметры изменяются во времени)

\* То есть алгебраические или трансцендентные.

Кроме того, структура модели определяется также набором размерностей: количеством переменных (входа, выхода, состояния) и параметров. Остановимся на задаче выбора типа ММ. Прежде всего, следует дать краткую характеристику основным типам ММ.

### 2.1.1. Статические и динамические модели

Математическая модель системы называется *статической*, если значение выхода  $y(t)$  зависит от значения входа  $u(t)$  только в тот же момент времени  $t$ . Символически это свойство записывается так:

$$y(t) = F(u(t)), \quad (2.1)$$

где  $F$  — символ некоторого преобразования (оператора).

Кроме явных функциональных зависимостей (2.1), статические модели могут задаваться неявно, в виде уравнения или системы:

$$\Phi(y(t), u(t)) = 0. \quad (2.2)$$

Так обычно записываются уравнения статических режимов радиоэлектронных схем, многих механических, энергетических систем и т. д. Уравнение (2.2) должно быть однозначным разрешимо относительно  $y(t)$ .

Статическими моделями пользуются, когда в рамках поставленной задачи (с точки зрения достижения выбранной цели) инерционностью и «памятью» реальной системы можно пренебречь. Это возможно при выполнении ряда условий, в число которых входят следующие:

1) система устойчива, т. е. переходные процессы после скачкообразного изменения входов затухают. Конечное время затухания с заданной точностью обозначим через  $t_{\text{пер}}$ ;

2) входы меняются медленно, т. е.  $\Delta t_{\text{вх}} > t_{\text{пер}}$ , где  $\Delta t_{\text{вх}}$  — время между изменениями входных воздействий;

3) выходы измеряются редко, т. е.  $\Delta t_{\text{вых}} > t_{\text{пер}}$ , где  $\Delta t_{\text{вых}} = t_{k+1} - t_k$  — промежутки между измерениями входных величин.

В *динамических* моделях значение  $y(t)$  может зависеть от всего прошлого (предыстории) входного процесса:

$$y(t) = F(\{u(s), s \leq t\}). \quad (2.3)$$

Динамические модели позволяют учесть наличие «памяти», инерционности системы. Математическим аппаратом опи-

сания динамических систем являются дифференциальные, разностные уравнения, конечные автоматы, случайные процессы. Некоторые классы динамических моделей рассмотрены в подразд. 2.2.

### 2.1.2. Дискретные и непрерывные модели

Система может быть дискретной или непрерывной по входам, по выходам и по времени в зависимости от того, дискретными или непрерывными являются множества  $U$ ,  $Y$ ,  $T$  соответственно. Под дискретным понимается конечное или счетное\* множество. Под непрерывным будем понимать множество объектов, для которого адекватной моделью служит отрезок, луч или прямая линия, т. е. связное числовое множество. Если система имеет несколько входов и выходов, то это значит, что соответствующие множества  $U$ ,  $T$  лежат в многомерных пространствах, т. е. непрерывность и дискретность понимаются покомпонентно.

Удобство числового множества как модели реальных совокупностей объектов состоит в том, что на нем естественным образом определяются несколько отношений, формализующих реально встречающиеся отношения между реальными объектами. Например, отношения близости, сходимости формализуют понятия похожести, сходства объектов и могут быть заданы посредством функции расстояния (метрики)  $d(x, y)$  (например,  $d(x, y) = |x - y|$ ). Числовые множества являются упорядоченными: отношение порядка следования ( $x \leq y$ ) формализует предпочтение одного объекта другому. Наконец, над элементами числовых множеств определены естественные операции\*\*, например, линейные:  $x + y$ ,  $x \cdot y$ . Если для реальных объектов на входе и выходе также имеют смысл аналогичные операции, то естественным образом возникают требования к моделям (2.1) — (2.3): быть согласованными с этими операциями, сохранять их результаты. Так

---

\* Строго говоря, счетное множество представляет собой некоторую идеализацию, которую невозможно реализовать практически из-за конечности размеров технических систем и пределов человеческого восприятия. Такие идеализированные модели (например, множество натуральных чисел  $N = \{1, 2, \dots\}$ ) имеет смысл вводить для множеств конечных, но с заранее не ограниченным (или неизвестным) числом элементов.

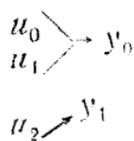
\*\* Формально понятие операции является частным случаем понятия отношения между элементами множеств. Например, операция сложения двух чисел задает 3-местное (тернарное) отношение  $R$ : тройка чисел  $(x, y, z)$  принадлежит отношению  $R$  (пишем  $(x, y, z) \in R$ ), если  $z = x + y$ .

мы приходим, например, к линейным моделям:  $y=au+b$ ,  $du/dt=ay+bu$  и т. д., являющимися простейшими моделями многих процессов.

Как правило, дискретность множества  $U$  влечет за собой дискретность  $Y$ . Кроме того, для статических систем исчезает разница между непрерывным и дискретным временем. Поэтому классификация детерминированных систем по признакам «статические — динамические», «дискретные — непрерывные» включает шесть основных групп, представленных в табл. 1.3, где для каждой группы указан математический аппарат описания систем, методы численного анализа и оценки их параметров, методы синтеза (оптимизации), а также типичные области применения.

**Пример 1.** Рассмотрим работу турникета на входе в метро. В первом, «грубом» приближении множество значений входа этой системы имеет два элемента: человек с жетоном ( $u_1$ ) и человек без жетона ( $u_2$ ), т. е.  $U=\{u_1, u_2\}$ . После небольшого размышления становится ясно, что следует включить еще отсутствие пассажира ( $u_0$ ), т. е.  $U=\{u_0, u_1, u_2\}$ . Множество значений выхода содержит элементы «открыто» ( $y_0$ ) и «закрыто» ( $y_1$ ). Таким образом,  $Y=\{y_0, y_1\}$  и система является дискретной. В простейшем случае можно пренебречь памятью системы и описывать ее статической моделью, имеющей вид таблицы или графа:

$u(t)$	$y(t)$
$u_0$	$y_0$
$u_1$	$y_0$
$u_2$	$y_1$



При необходимости хранить ММ системы в ЭВМ ее можно представить (закодировать) в виде матрицы  $F=\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  или более экономно, в виде списка  $(0, 0, 1)$ , в котором на  $i$ -м месте стоит  $j$ , если значению входа  $u_{i-1}$  соответствует значение выхода  $y_j$ .

**Пример 2.** Если нас интересует более детально устройство самого турникета (т. е. системой является турникет), то придется учесть, что входными воздействиями (сигналами) для него являются опускание пятака и прохождение человека через турникет. Таким образом, система имеет два входа, каждый из которых может принимать два значения («есть»

## Детерминированные модели систем

Типы систем	Статические		Динамические			
	дискретные по $U, Y$	непрерывные по $U, Y$	дискретные по $T$		непрерывные по $T$	
			дискретные по $U, Y$	непрерывные по $U, Y$	дискретные по $U, Y$	непрерывные по $U, Y$
Математический аппарат оптимизации	графы, таблицы соответствий, булева алгебра. Предикаты [24, 26]	функции вещественных переменных [37]	конечные автоматы [31]	разностные уравнения [35]	асинхронные автоматы, сети Петри, модели теории расписаний [32, 33]	обыкновенные дифференциальные уравнения (ОДУ)
Методы оценки параметров и анализа	методы математической логики [26]	методы интерполяции, аппроксимации [30]	теория конечных автоматов [31]	идентификация, теория устойчивости [9, 35]	методы идентификации [36]	идентификация, численное интегрирование ОДУ [6, 7, 9, 30]
Методы синтеза	карты Карно, метод Куайна, дискретное программирование [27]	методы оптимизации (линейное и нелинейное программирование [28, 29])	методы синтеза микропрограммных автоматов, динамическое программирование [24, 34]	динамическое программирование, дискретный принцип максимума [35]	динамическое программирование, теория расписаний [36]	теория управления, методы оптимизации [6, 9, 36]
Области применения	рейлейно-контактные схемы. Импульсные и логические схемы. Качественные модели исследования операций	количественные модели исследования операций	цифровые САУ, ГАП, логическое управление	импульсные и цифровые САУ	параллельные процессы в ЭВМ и ГАП	САУ, механические, тепловые, электронные и другие процессы

или «нет»). Пренебрегая возможностью одновременного опускания жетона и прохождения, вводим три значения входа:  $u_0$  — «нет воздействия»,  $u_1$  — «опускание жетона»,  $u_2$  — «прохождение». Множество  $Y$  можно задать так же, как и в примере 1. Однако теперь значение выхода  $y(t)$  не определяется только значением входа  $u(t)$ , а зависит еще и от того, был ли опущен жетон раньше, т. е. от значений  $u(s)$  при  $s < t$ . Система имеет «память». Простейший тип ММ для описания дискретных систем с памятью — это конечный автомат [31]. Для его построения вводится конечное множество внутренних состояний системы  $X$ , определяющее «память». В данном случае в  $X$  достаточно включить два элемента:  $x_0$  — «жетон не был брошен»,  $x_1$  — «жетон был брошен». Значения состояния системы в следующий момент времени и выхода в текущий момент зависят от текущих значений состояния и входа, т. е.

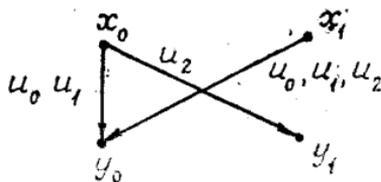
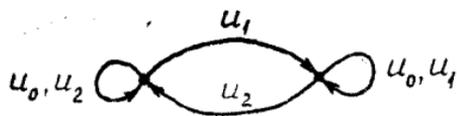
$$x(k+1) = F(x(k), u(k)), \quad y(k) = G(x(k), u(k)), \quad (2.4)$$

где  $k$  — номер момента времени такта. Отметим, что, выделив «текущий» и «следующий» моменты времени, мы незаметно ввели предположение о дискретности времени, которое при более детальном исследовании может оказаться неправомерным см. ниже п. 2.2.3). Функцию переходов  $F(x, u)$  и функцию выходов  $G(x, u)$  можно задать таблично:

	$u_0$	$u_1$	$u_2$
$F(x, u):$	$x_0$	$x_1$	$x_0$
	$x_1$	$x_1$	$x_0$

	$u_0$	$u_1$	$u_2$
$G(x, u):$	$y_0$	$y_0$	$y_1$
	$x_1$	$y_0$	$y_0$

Можно также построить графы переходов и выходов:



**Пример 3.** Рассмотрим простейшую электрическую цепь —  $RC$ -цепочку (рис. 1.6). Входом системы является напряжение источника  $u(t) = E_0(t)$ , выходом — напряжение на конденса-

торе  $y(t) = E_1(t)$ . Закон Ома дает ММ системы в виде дифференциального уравнения 1-го порядка

$$\tau \dot{y} = u - y, \quad (2.5)$$

где  $\tau = RC$  — постоянная времени цепочки. ММ (2.5) полностью непрерывна:  $U = Y = T = R^1$ . Если исследователя интересует поведение системы в статических режимах, т. е. при  $E_0(t) = \text{const}$ , то нужно положить в (2.5)  $\dot{y} = 0$  и получить статическую модель

$$y(t) = u(t). \quad (2.6)$$

Моделью (2.6) можно пользоваться как приближенной в случае, когда вход  $E_0(t)$  изменяется достаточно редко или медленно (по сравнению с  $\tau$ ).

**Пример 4.** Рассмотрим экологическую систему, состоящую из двух взаимодействующих популяций [9], существующих на некоторой территории. Предположим, что система автономна, т. е. внешними воздействиями (входами) можно пренебречь; за выходы системы примем численности популяций (видов)  $y_1(t)$ ,  $y_2(t)$ . Пусть 2-й вид является пищей для 1-го, т. е. система относится к классу «хищник — жертва» (например,  $y_1$  — численность лис в лесу, а  $y_2$  — численность зайцев; или  $y_1$  — концентрация бактерий-возбудителей заболевания в городе, а  $y_2$  — число заболевших и т. д.). В данном случае  $y_1$ ,  $y_2$  — целые числа и, на первый взгляд, в ММ системы множество  $Y$  должно быть дискретным. Однако для построения ММ удобнее считать, что  $y_1$ ,  $y_2$  могут принимать произвольные вещественные значения, т. е. перейти к непрерывной модели (при достаточно больших  $y_1$ ,  $y_2$  этот переход не внесет существенной погрешности). При этом мы сможем пользоваться такими понятиями, как скорости изменения выходных переменных  $\dot{y}_1$ ,  $\dot{y}_2$ . Простейшая модель динамики популяции получается, если предположить, что:

— при отсутствии хищников численность жертв растет экспоненциально;

— при отсутствии жертв численность хищников убывает экспоненциально;

— численность «съеденных» жертв пропорциональна величине  $y_1 y_2$ .

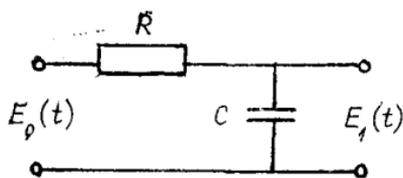


Рис. 1.6

При этих предположениях динамика системы, как нетрудно видеть, описывается так называемой моделью Лотки — Вольтерра:

$$\frac{dy_1}{dt} = ay_1 - by_1 y_2, \quad \frac{dy_2}{dt} = cy_1 y_2 - dy_2, \quad (2.7)$$

где  $a, b, c, d$  — положительные параметры. Если есть возможность изменять параметры, то они превращаются во входные переменные, например, когда изменяются коэффициенты рождаемости и смертности видов, коэффициенты размножения бактерий (при введении лекарств) и т. д.

## 2.2. Модели состояния динамических систем

### 2.2.1. Модели общего вида

Важнейшую роль при описании динамических систем играет понятие состояния. Состояние — это совокупность величин (вектор)\*  $x = \text{col}(x_1, \dots, x_n)$ , которые определяют (вместе с входным воздействием) будущее поведение системы. Например, для RC-цепочки переменная состояния есть  $E_1$ , поскольку значение  $E_1(t)$  и входного воздействия  $E_0(s)$  при  $s \geq t$  однозначно определяет (в силу (2.5)) значение  $E_1(s)$  при  $s = t$ . Для модели динамики популяций (2.7) состоянием является вектор  $x = \text{col}(y_1, y_2)$ .

В общем случае уравнения состояния — это системы дифференциальных или разностных уравнений первого порядка вместе с уравнениями для выходных величин. Начальное состояние представляет «память» системы о прошлом. Модель состояния непрерывной динамической системы записывается в виде [9, 21, 60]

$$\begin{cases} x_1 = F_1(x_1, \dots, x_n, u_1, \dots, u_m, t) \\ \dots \\ x_n = F_n(x_1, \dots, x_n, u_1, \dots, u_m, t), \end{cases} \quad (2.8a)$$

$$\begin{cases} y_1 = G_1(x_1, \dots, x_n, u_1, \dots, u_m, t), \\ \dots \\ y_l = G_l(x_1, \dots, x_n, u_1, \dots, u_m, t), \end{cases} \quad (2.8b)$$

где  $u_1, \dots, u_m$  — входные переменные,  $y_1, \dots, y_l$  — выходные переменные,  $x_1, \dots, x_n$  — переменные состояния. Вводя век-

\* Здесь и далее вектор (столбец) с компонентами  $x_1, x_2, \dots, x_n$  обозначается  $x = \text{col}(x_1, \dots, x_n)$ .

торные обозначения, можно записать (2.8) в более компактном виде:

$$\dot{x} = F(x, u, t), \quad y = G(x, u, t), \quad (2.9)$$

где  $u = \text{col}(u_1, \dots, u_m)$ ,  $y = \text{col}(y_1, \dots, y_l)$ ,  $x = \text{col}(x_1, \dots, x_n)$ .

Для моделей состояния справедлив следующий факт: любая нелинейная динамическая система может быть представлена как соединение линейных динамических и нелинейных статических звеньев. Доказательство очевидно из рис. 1.7, где в качестве линейного звена взят интегратор.

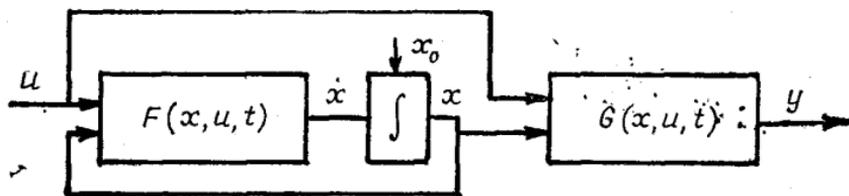


Рис. 1.7

Еще более общей формой описания динамических систем являются сингулярные дифференциальные (алгебро-дифференциальные) системы

$$\Phi(\dot{x}, x, u, t) = 0, \quad G(x, y, u, t) = 0, \quad (2.10)$$

частным случаем которых являются неявные системы

$$\Phi(y, y, u, t) = 0. \quad (2.11)$$

### 2.2.2. Линейные модели

Часто вместо (2.8) используют упрощенные ММ, основанные на том, что процессы в системе протекают, мало отклоняясь от некоторой так называемой опорной траектории  $\{\bar{x}(t), \bar{u}(t), \bar{y}(t)\}$ , удовлетворяющей уравнениям

$$\dot{\bar{x}} = F(\bar{x}, \bar{u}), \quad \bar{y} = G(\bar{x}, \bar{u}). \quad (2.12)$$

Тогда можно записать приближенную *линеаризованную* модель в отклонениях от этого режима:

$$\begin{cases} \tilde{x} = A(t)\tilde{x} + B(t)\tilde{u}, \\ \tilde{y} = C(t)\tilde{x} + D(t)\tilde{u}, \end{cases} \quad (2.13)$$

где  $\tilde{x} = x - \bar{x}$ ,  $\tilde{y} = y - \bar{y}$ ,  $\tilde{u} = u - \bar{u}$ ,

$$A(t) = \frac{\partial F}{\partial x}(\bar{x}(t), \bar{u}(t)), \quad B(t) = \frac{\partial F}{\partial u}(\bar{x}(t), \bar{u}(t)),$$

$$C(t) = \frac{\partial G}{\partial x}(\bar{x}(t), \bar{u}(t)), \quad D(t) = \frac{\partial G}{\partial u}(\bar{x}(t), \bar{u}(t)).$$

**Пример.**  $\dot{x} = -x^2u$ ;  $x, u \in R^1$ . Линеаризуем вблизи траектории, соответствующей  $\bar{u}(t) \equiv 1$ . Имеем  $\bar{x} = -x^2$ , откуда либо  $x(t) = 0$  (при  $\bar{x}(0) = 0$ ), либо  $\bar{x}(t) = 1/(t-a)$ . Рассмотрим второй случай:

$$A(t) = \frac{\partial}{\partial x}(-x^2u) \Big|_{\substack{u=1 \\ x=\bar{x}(t)}} = -2\bar{x}(t) = -\frac{2}{t-a},$$

$$B(t) = \frac{\partial}{\partial u}(-x^2u) \Big|_{\substack{u=1 \\ x=\bar{x}(t)}} = -\bar{x}^2(t) = -\frac{1}{(t-a)^2}.$$

В отклонениях  $\tilde{x} = x - 1/(t-a)$ ,  $\tilde{u} = u - 1$  линеаризованное уравнение имеет вид

$$\dot{\tilde{x}} = -\frac{2}{t-a}\tilde{x} - \frac{1}{(t-a)^2}\tilde{u}.$$

Если расчетный режим является установившимся, т. е. не зависит от времени, то коэффициенты в (2.13) также не зависят от времени:  $A(t) \equiv A$ ,  $B(t) \equiv B$  и т. д. Такие системы называются стационарными. Особенно часто на практике встречаются стационарные линейные непрерывные системы, описываемые более простыми уравнениями

$$\dot{x} = Ax + Bu, \quad y = Cx. \quad (2.14)$$

Матрицы  $A$ ,  $B$ ,  $C$  являются параметрами модели (2.14).

Если линеаризация приводит к большим погрешностям, то стараются по возможности выбрать ММ, линейную по параметрам:

$$\dot{x} = A \cdot \psi(x, u),$$

где  $A$  — матрица параметров порядка  $n \times N$ ,  $\psi(\cdot, \cdot) \in R^N$  — нелинейная вектор-функция. К этому классу относятся,

в частности, билинейные объекты, например,  $\dot{x} = a_1x + a_2xu + a_3u$ , где  $A = [a_1, a_2, a_3]$ ,  $\psi(x, u) = \text{col}(x, xu, u)$ .

Сказанное выше относится и к уравнениям дискретных по времени систем. Уравнения дискретной системы в общем случае имеют вид

$$x_{k+1} = F(x_k, u_k), \quad y_k = G(x_k, u_k). \quad (2.15)$$

Дискретным аналогом уравнений линейной стационарной системы (2.14) являются уравнения:

$$x_{k+1} = Px_k + Qu_k, \quad y_k = Rx_k. \quad (2.16)$$

Наряду с уравнениями состояния широкое применение находят также модели в переменных «вход—выход» и модели, описываемые передаточными функциями. Для непрерывного времени уравнение «вход—выход» имеет вид

$$A(p)y(t) = B(p)u(t), \quad (2.17)$$

где  $p = d/dt$  — символ дифференцирования по времени,  $A(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_0$ ,  $B(\lambda) = b_m\lambda^m + \dots + b_0$ , причем в (2.17) всегда  $m < n$ . Дробно-рациональная функция  $W(\lambda) = B(\lambda)/A(\lambda)$  называется *передаточной функцией* системы (2.17), а полином  $A(\lambda)$  — ее *характеристическим полиномом*\*. Если уравнение (2.17) получено из (2.14), то

$$W(\lambda) = C(\lambda I_n - A)^{-1}B, \quad (2.18)$$

$$A(\lambda) = \det(\lambda I_n - A). \quad (2.18a)$$

Они справедливы и в случае, когда вход и выход системы (2.14) являются векторами, при этом  $W(\lambda) = A^{-1}(\lambda)B(\lambda)$  — матрица. Пользуясь (2.18), можно показать, что замена переменных состояния в (2.14) по формуле  $x' = Tx$ , где  $T$  — неособая  $n \times n$ -матрица ( $\det T \neq 0$ ), не приводит к изменению передаточной функции (2.18a). Это значит, что обратный переход от описания «вход—выход» к уравнениям состояния (2.14) неоднозначен: при сохранении передаточной функции базис в пространстве состояний можно выбирать по-разному. На практике применяются несколько типовых способов перехода от передаточной функции к уравнениям состояния. Эти способы соответствуют так называемым каноническим представлениям системы [9]. Опишем один из них, приводящий

\*  $\lambda$  — комплексное число, аргумент полиномов  $A(\lambda)$ ,  $B(\lambda)$ .

к управляемому каноническому представлению. Вместо (2.16) вводятся два уравнения:

$$A(p)\eta = u, \quad (2.19a)$$

$$y = B(p)\eta, \quad (2.19б)$$

где  $\eta$  — вспомогательная переменная. Очевидно, что передаточные функции (2.17) и (2.19) совпадают. В качестве вектора состояния в уравнении (2.16) берется  $x = \text{col}(\eta, \eta, \dots, \eta^{(n-1)})$ , так что  $x_i = \eta^{(i-1)}$ . Из (2.19a) и соотношений  $x_i = \eta^{(i)} = x_{i+1}$ ,  $i = 1, 2, \dots, n-1$ , выводится форма матрицы  $A$  и вектора  $B$  в (2.14), а из (2.19б), записанного в виде  $y = b_0 x_1 + b_1 x_2 + \dots + b_m x_m$ , получаем строку  $C$ :

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{n-1} & -a_n & \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad (2.20)$$

$$C = [b_0 \dots b_m \ 0 \dots 0].$$

Если для системы (2.17) наблюдению доступна производная  $y^{(i)}$  от величины  $y$  при  $i \leq n - m - 1$ , то она может быть получена, если в найденных уравнениях сохранить  $A, B$  в форме (2.20) и взять  $C = [0, \dots, b_0, \dots, b_m, \dots, 0]$ .

Если в (2.17)  $m = n$  (такие передаточные функции называются *несобственными*), то систему (2.17) нельзя привести к виду (2.14), но можно привести к виду

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \quad (2.21)$$

где  $A, B$  имеют вид (2.20),  $C = [b_0 - a_0 b_n, \dots, b_{n-1} - a_{n-1} b_n]$ ,  $D = b_n$ .

### 2.2.3. Дискретизация и континуализация

**Дискретизация.** Если исходное описание линейной системы непрерывно, часто можно перейти к дискретному описанию с помощью следующей процедуры.

Пусть состояние  $x(t)$  системы (2.14) доступно измерению в дискретные моменты времени  $t_k = kh$ ,  $k = 0, 1, \dots$ , где  $h > 0$  — шаг дискретности. Пусть  $u(t)$  постоянно\* на проме-

\* Это предположение часто выполняется на практике.

жутках между моментами коррекции  $t_k$ . Тогда динамику векторов  $x_k = x(t_k)$  можно описать разностными уравнениями (2.16), в которых матрицы  $P$  и  $Q$  определяются соотношениями

$$P = e^{Ah}, \quad Q = A^{-1}(P - I_n)B. \quad (2.22)$$

Здесь  $e^{Ah}$  — экспоненциал матрицы  $A$ , определяемый формулой

$$e^{Ah} = I_n + Ah + \frac{1}{2} A^2 h^2 + \dots = \sum_{k=0}^{\infty} \frac{b^k A^k}{k!}. \quad (2.23)$$

Если предположение о кусочном постоянстве  $u(t)$  не выполняется, то переход от (2.14) к (2.16) является приближенным, но его точность растет по мере уменьшения шага  $h$ . При достаточно малых  $h$  для вычисления  $e^{Ah}$  можно удерживать лишь первые несколько членов ряда (2.23) или аппроксимировать сумму (2.23) каким-либо способом.

Например, при переходе от (2.14) к (2.16) можно пользоваться формулой  $e^{Ah} \approx I_n + Ah$ , соответствующей численному интегрированию (2.14) методом Эйлера. При такой аппроксимации передаточные функции дискретной и непрерывной систем будут связаны соотношением

$$W_1(\lambda) = W_n\left(\frac{1-\lambda}{h}\right), \quad (2.24)$$

т. е. при переходе к дискретному времени в передаточной функции  $W(p)$  системы (2.17) нужно заменить  $p$  на  $(1-z)/h$ . Если матрица  $A$  гурвицева, т. е.  $\operatorname{Re} \lambda_i(A) < 0$ , то метод Эйлера дает устойчивую аппроксимацию лишь при

$$h < \min_i [-2\operatorname{Re} \lambda_i(A)] / |\lambda_i(A)|^2, \quad (2.25)$$

где  $\lambda_i(A)$  — собственные числа матрицы  $A$  (корни полинома  $A(\lambda)$ ). Целый ряд способов перехода от (2.14) к (2.17) основан на аппроксимации матрицы  $e^{Ah}$  дробями Падэ. Частными случаями этих способов является метод Тастина (формула Падэ порядка (1, 1)):

$$e^{Ah} \approx \left[ I + \frac{Ah}{2} \right] \left[ I - \frac{Ah}{2} \right]^{-1}, \quad (2.26)$$

приводящий к соотношению между передаточными функциями

$$W_d(\lambda) = \frac{2}{1+\lambda} W_n\left(\frac{2}{h} \frac{1-\lambda^{-1}}{1+\lambda^{-1}}\right), \quad (2.27)$$

а также метод Дэвисона (формула Падэ порядка (2, 2)):

$$e^{Ah} \approx \left[ I + \frac{Ah}{2} + \frac{A^2 h^2}{12} \right] \left[ I - \frac{Ah}{2} + \frac{A^2 h^2}{12} \right]^{-1}. \quad (2.28)$$

Отметим, что формулы (2.26) и (2.28) дают устойчивые аппроксимации при любом  $h > 0$  (разумеется, если  $A$  гурвицева).

Если непрерывная система нелинейна, то для перехода к его дискретному описанию также можно использовать методы численного интегрирования. Например, метод Эйлера дает для системы (2.9) дискретное описание:

$$x_{k+1} = x_k + hF(x_k, u_k, t_k), \quad y_k = G(x_k, u_k, t_k). \quad (2.29)$$

**Континуализация** — это переход от дискретной математической модели системы к непрерывной. Если дискретная модель системы имеет вид (2.16), то перейти к непрерывной модели (2.14) можно по формулам

$$A = \frac{1}{h} \ln P, \quad B = \frac{1}{h} \ln P (P - I)^{-1} Q, \quad (2.30)$$

вытекающим из (2.22), где  $\ln P$  — логарифм матрицы — функция, обратная к экспоненциальной и также определяемая через ряд

$$\ln(I + X) = X - \frac{X^2}{2} + \dots + \frac{(-1)^{n+1}}{n} X^n,$$

сходящийся при  $\|X\| < 1$  ( $X = P - I$ ). С точностью до величин порядка  $h^2$  можно ограничиться формулами

$$A = \frac{1}{h} (P - I), \quad B = \frac{1}{h} Q,$$

соответствующими методу Эйлера. Однако удобнее всего переходить от дискретной передаточной функции к непрерывной по формулам (2.24) и (2.27). Например, по методу Эйлера (2.24) достаточно заменить в передаточной функции  $W_d(z^{-1})$  переменную  $z^{-1}$  на  $1 - hp$ .

При исследовании линейных систем получили распространение также методы упрощения описаний систем путем *редукции* (понижения порядка) [2, 39]. Взаимосвязь различных описаний динамических систем представлена на рис. 1.8.

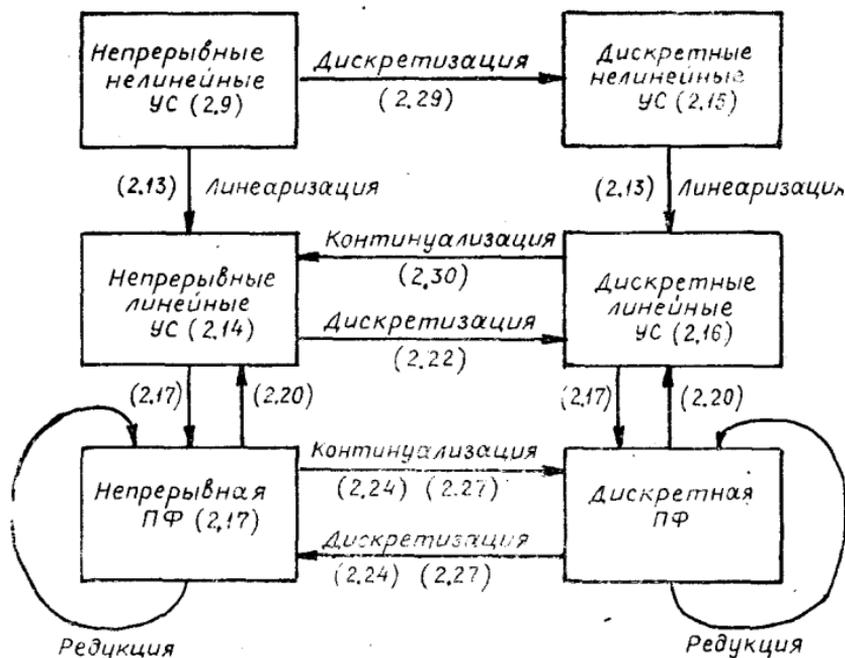


Рис. 1.8

### 2.3. Детерминированные и стохастические модели

Модели систем, о которых мы говорили до сих пор, были детерминированными (определенными), т. е. задание входного воздействия определяло выход системы однозначно. Однако на практике так бывает редко: описанию реальных систем обычно присуща неопределенность. Например, для статической модели неопределенность можно учесть, записывая вместо (2.1) соотношение

$$y(t) = F(u(t)) + \varphi(t), \quad (2.31)$$

где  $\varphi(t)$  — погрешность, приведенная к выходу системы.

Причины неопределенности разнообразны:

— погрешности и помехи измерений входов и выходов системы (естественные погрешности);

— неточность самой модели системы, что заставляет искусственно вводить в модель погрешность;

— неполнота информации о параметрах системы и т. д.

Среди различных способов уточнения и формализации неопределенности наибольшее распространение получил стохастический (вероятностный) подход, при котором неопределенные величины считаются случайными. Развитый понятийный и вычислительный аппарат теории вероятностей и математической статистики позволяет дать конкретные рекомендации по выбору структуры системы и оценке ее параметров. Классификация стохастических моделей систем и методов их исследования представлена в табл. I.4. Выводы и рекомендации основаны на эффекте усреднения: случайные отклонения результатов измерений некоторой величины от ее ожидаемого значения при суммировании взаимно уничтожаются, и среднее арифметическое большого числа измерений оказывается близким к ожидаемому значению. Математические формулировки этого эффекта даются *законом больших чисел и центральной предельной теоремой*. Закон больших чисел гласит, что если  $\xi_1, \dots, \xi_N$  — случайные величины с математическим ожиданием (средним значением)  $M_{\xi_i} = a$  и дисперсией  $M(\xi_i - a)^2 = \sigma^2$ , то

$$\frac{1}{N} (\xi_1 + \dots + \xi_N) - a \approx 0 \quad (2.32)$$

при достаточно больших  $N$ . Это говорит о принципиальной возможности сколь угодно точной оценки  $M_{\xi_i}$  по измерениям. Центральная предельная теорема, уточняющая (2.32), утверждает, что

$$\frac{1}{N} (\xi_1 + \dots + \xi_N) - a \approx \frac{\sigma}{\sqrt{N}} \xi, \quad (2.33)$$

где  $\xi$  — стандартная нормально распределенная случайная величина\* ( $M\xi = 0$ ,  $M\xi^2 = 1$ ).

Поскольку распределение величины  $\xi$  хорошо известно и заатабулировано (например, известно, что  $P\{|\xi| > 1\} = 0,32$ ,

---

\* Если величина  $\sigma^2$  неизвестна, то следует заменить в (2.33)  $\sigma$  на оценку  $s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\xi_i - \bar{\xi})^2}$ , где  $\bar{\xi} = \frac{1}{N} \sum_{i=1}^N \xi_i$ . При этом величина  $\xi$  будет распределена уже не нормально, а по закону Стьюдента, который при  $N \geq 20$  практически неотличим от нормального.

## Стохастические модели систем

Тип систем	Статические		Динамические			
	дискретные по $U, Y$	непрерывные по $U, Y$	дискретные по $T$		непрерывные по $T$	
			дискретные по $U, Y$	непрерывные по $U, Y$	дискретные по $U, Y$	непрерывные по $U, Y$
Математический аппарат описания	схема независимых испытаний [40]	регрессионные модели [40]	марковские цепи, стохастические автоматы [24]	стохастические разностные уравнения [35]	системы массового обслуживания, полумарковские процессы [12, 13]	стохастические дифференциальные уравнения [16]
Методы оценки параметров и анализа	статистические оценки вероятности, дисперсионный анализ [44, 46]	регрессионный анализ [45—47]	оценка вероятностей, статистическое моделирование [12, 48]	статистическое оценивание состояний и параметров, анализ стохастической устойчивости [35, 45]	теория массового обслуживания, имитационное моделирование [36]	теория устойчивости [35]
Методы синтеза	стохастическое программирование [28]	планирование эксперимента, стохастическое программирование [46, 47]	динамическое программирование [24]	динамическое программирование [9, 35]	перебор, методы оптимального управления [36]	оптимальное и адаптивное управление [9, 35]
Области применения	задачи выбора из конечного числа вариантов (испытания, управление)	обработка результатов измерений, испытаний	ЭВМ	импульсные и цифровые САУ	системы обслуживания (вычисл. системы, производственные ГАП и т. д.)	САУ, механ., электронные, тепловые и др. процессы

$P\{|\xi| > 2\} = 0,05$ ,  $P\{|\xi| > 3\} = 0,03$ , то соотношение (2.33) позволяет вычислять погрешность оценки. Пусть, например, требуется найти, при каком числе измерений погрешность оценки их математического ожидания с вероятностью 0,95 окажется меньше, чем 0,01, если дисперсия каждого измерения равна 0,25. Из (2.33) получаем, что должно выполняться неравенство  $\left| \frac{0,5}{\sqrt{N}} \cdot 2 \right| < 0,01$ , откуда  $N > 10000$ .

Разумеется, формулировкам (2.32), (2.33) можно придать более строгий вид, и это легко может быть сделано с помощью понятий вероятностной сходимости. Трудности возникают при попытке проверить условия этих строгих утверждений. Например, в законе больших чисел и центральной предельной теореме требуется независимость отдельных измерений (реализаций) случайной величины и конечность ее дисперсии. Если эти условия нарушаются, то могут нарушаться и выводы. Например, если все измерения совпадают:  $\xi_1 = \dots = \xi_N$  то, хотя все остальные условия выполняются, об усреднении не может быть и речи. Другой пример: закон больших чисел несправедлив, если случайные величины  $\xi_1, \dots, \xi_N$  распределены по закону Коши (с плотностью распределения  $p(x) = \frac{1}{\pi(1+x^2)}$ ), не обладающему конечными математическими ожиданиями и дисперсией. А ведь такой закон встречается в жизни! Например, по Коши распределена интегральная освещенность точек прямолинейного берега равномерно вращающимся прожектором, находящимся в море (на корабле) и включающимся в случайные моменты времени.

Но еще большие трудности вызывает проверка обоснованности самого употребления термина «случайный». Что такое случайная величина, случайное событие и т. д.? Часто говорят, что событие  $A$  случайно, если в результате эксперимента оно может наступить (с вероятностью  $p$ ) или не наступить (с вероятностью  $1-p$ ). Все, однако, не так просто. Само понятие вероятности может быть связано с результатами экспериментов лишь через частоту его наступления в некотором ряде (серии) экспериментов:  $v_N = N_A/N$ , где  $N_A$  — число экспериментов, в которых событие наступило,  $N$  — общее число экспериментов. Если числа  $v_N$  при достаточно большом  $N$  приближаются к некоторому постоянному числу  $p_A$ :

$$v_N \approx p_A, \quad (2.34)$$

то событие  $A$  можно назвать случайным, а число  $p$  — его вероятностью. При этом частоты, наблюдавшиеся в различных сериях экспериментов, должны быть близки между собой (это свойство называется *статистической устойчивостью* или *однородностью*). Сказанное относится и к понятию случайной величины, поскольку величина  $\xi$  является случайной, если случайными являются события  $\{a < \xi < b\}$  для любых чисел  $a, b$ . Частоты наступления таких событий в длинных сериях экспериментов должны группироваться около некоторых постоянных значений.

Итак, для применимости стохастического подхода должны выполняться следующие требования:

1) массовость проводимых экспериментов, т. е. достаточно большое число;

2) повторяемость условий экспериментов, оправдывающая сравнение результатов различных экспериментов;

3) статистическая устойчивость.

Стохастический подход заведомо нельзя применять к единичным экспериментам: бессмысленны выражения типа «вероятность того, что завтра будет дождь», «с вероятностью 0.8 «Зенит» выиграет кубок» и т. п. Но даже если массовость и повторяемость экспериментов имеются, статистической устойчивости может и не быть, а проверить это — непростое дело. Известные оценки допустимого отклонения частоты от вероятности основаны на центральной предельной теореме\* или неравенстве Чебышева и требуют дополнительных гипотез о независимости или слабой зависимости измерений. Опытная же проверка условия независимости еще сложнее, так как требует дополнительных экспериментов.

Более подробно методология и практические рецепты применения теории вероятностей изложены в поучительной книге В. Н. Тутубалина [40], представление о которой дают приводимые ниже цитаты:

«Чрезвычайно важно искоренить заблуждение, встречающееся иногда у недостаточно знакомых с теорией вероятностей инженеров и естествоиспытателей, что результат любого эксперимента можно рассматривать как случайную величину. В особо тяжелых случаях к этому присоединяется вера в нормальный закон распределения, а если уже сами случайные

---

\* Легко заметить, что (2.34) есть частный случай (2.32), когда берется  $\xi_i = 1$ , если событие  $A$  наступило в  $i$ -м эксперименте,  $\xi_i = 0$  в противном случае. При этом  $M\xi_i = p_A$ ,  $M(\xi_i - p_A)^2 = p_A(1 - p_A)$ .

величины не нормальны, то верят, что их логарифмы нормальны».

«По современным представлениям область применения теоретико-вероятностных методов ограничена явлениями, которым присуща статистическая устойчивость. Однако проверка статистической устойчивости трудна и всегда неполна; к тому же часто она дает отрицательный вывод. В результате в целых областях знания, например, в геологии, нормой стал такой подход, при котором статистическая устойчивость вообще не проверяется, что неизбежно приводит к серьезным ошибкам. К тому же пропаганда кибернетики\*, предпринятая нашими ведущими учеными, дала (в некоторых случаях) несколько неожиданный результат: теперь считается, что только машина (а не человек) способна получать объективные научные результаты.

В таких обстоятельствах долг каждого преподавателя — вновь и вновь пропагандировать ту старую истину, которую еще Петр I пытался (безуспешно) внушить русским купцам: что торговать надо честно, без обмана, так как в конечном счете это для самих же себя выгоднее».

Как же построить модель системы, если неопределенность в задаче есть, но стохастический подход неприменим? Ниже кратко излагается один из альтернативных подходов, основанный на теории нечетких множеств.

## 2.4. Нечеткие модели

### 2.4.1. Нечеткие множества и лингвистические переменные

В 1965 г. американский математик Л. Заде опубликовал статью под названием «Fuzzy sets», что можно перевести как «Нечеткие множества»\*\*. В статье дано новое определение понятия множества, предназначенное для описания и исследования сложных, «плохо определенных» систем. К ним, в частности относятся гуманистические системы, на поведение которых существенное влияние оказывают знания, суждения и эмоции человека. В таких системах наряду со строгими, объективными, количественными данными и результатами присутствуют неоднозначные субъективные, качественные, что требует новых подходов.

\* А сегодня можно добавить «... и информатики» (прим. автора).

\*\* Слово Fuzzy переводилось на русский язык как «размытые», «расплывчатые», «неопределенные», однако в последние годы чаще всего употребляется термин «нечеткие».

Поскольку понятие множества лежит в основе всех математических конструкций, статья Л. Заде породила новое научное направление, бурный поток публикаций, специальные конференции и т. д. Произошло «раздвоение» математики: появились нечеткие функции, нечеткие отношения, нечеткие уравнения, нечеткая логика и т. д. [38, 41, 42]. Новый математический аппарат описывает свойства нечетких систем, соответствующих трудно формализуемым, плохо структурированным задачам. В последние годы эти методы стали широко применяться в экспертных программных системах. Ниже излагаются основные понятия теории нечетких систем.

Прежде, чем говорить о нечетких системах, необходимо ввести понятие нечеткого множества.

**Определение.** *Нечетким подмножеством*  $A$  множества  $X$  назовем пару  $(X, \mu_A)$ , где  $\mu_A: X \rightarrow [0, 1]$  — функция, каждое значение которой  $\mu_A(x) \in [0, 1]$  интерпретируется как степень принадлежности точки  $x \in X$  множеству  $A$ . Функция  $\mu_A$  называется *функцией принадлежности* множества  $A$ .

Для обычного «четкого» множества  $B$  можно положить

$$\mu_B(x) = \begin{cases} 1, & x \in B, \\ 0, & x \notin B \end{cases}, \text{ т. е. классическое понятие множества}$$

является частным случаем введенного (рис. 1.9, а). Задавать функцию принадлежности можно таблично или аналитически.

**Пример 1.** Пусть  $X = \{1, 2, \dots\}$  — множество натуральных чисел, а функция  $\mu_A(x)$  задана таблицей

$x$	1	2	3	4	5	6	7	8	9	10	...
$\mu_A(x)$	0	0,1	0,6	0,8	1	1	0,9	0,7	0,2	0	...

Определенное таким образом нечеткое множество можно принять в качестве формализации понятия «несколько», ясного лишь на интуитивном уровне.

Аналогично можно ввести нечеткие множества, соответствующие понятиям «много», «мало», «около 100», «почти 20» и т. д. Легко разрешается знаменитый парадокс древних греков: «Сколько зерен составляют кучу?». Ответ состоит в том, что «куча» — нечеткое множество зерен, функцию принадлежности которого можно получить, например, путем опроса.

**Пример 2.** Пусть  $X=[0, \infty]$  — множество положительных чисел, а функция  $\mu_A(x)$  задана формулой

$$\mu_A(x) = \begin{cases} 0 & \text{при } 0 \leq x \leq 50, \\ \left(1 + \frac{25}{(x-50)^2}\right)^{-1} & \text{при } x > 50, \end{cases} \quad (2.35)$$

график которой изображен на рис. 1.9, б. Если переменную  $x$  интерпретировать как возраст, то нечеткое множество  $A$  соответствует понятию «старый». Аналогично можно формализовать понятия «молодой», «средних лет» и т. д.

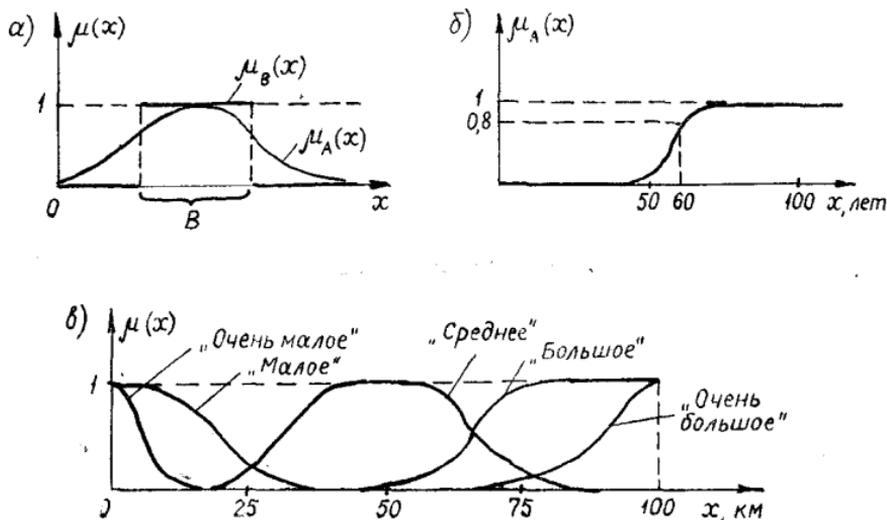


Рис. 1.9

Переменные, значениями которых являются нечеткие множества, называются *лингвистическими*. Это основной тип переменных в естественном языке людей.

**Пример 3.** Переменная «расстояние» принимает обычные числовые значения. Однако в предложениях естественного языка она может фигурировать как лингвистическая со значениями «малое», «большое», «очень малое», «среднее», «около 5 км» и т. д. Каждое значение описывается нечетким множеством, которое в рамках данной предметной области может иметь конкретную числовую интерпретацию. Например, если речь идет о поездках на такси, то в качестве универсального множества  $X$  можно взять отрезок  $[0, 100]$  км и задать функции принадлежности значений переменной «расстояние», как показано на рис. 1.9, в.

## Операция с нечеткими множествами

Операция	Лингвистический смысл	Формула для $\mu_C(x)$	График $\mu_C(x)$
Пересечение $C = A \cap B$	И	$\min\{\mu_A(x), \mu_B(x)\}$	
Объединение $C = A \cup B$	ИЛИ	$\max\{\mu_A(x), \mu_B(x)\}$	
Дополнение $C = \bar{A}$	НЕ	$1 - \mu_A(x)$	
Концентрация	ОЧЕНЬ	$[\mu_A(x)]^2$	
Размывание	НЕ ОЧЕНЬ	$\sqrt{\mu_A(x)}$	

При первом знакомстве с нечеткими множествами обычно возникает недовольство произволом и субъективизмом в задании функций принадлежности: «почему так, а не иначе?». Однако это не слабость, а сила подхода! Ведь если само понятие субъективно, то такова и его формализация, выполняемая человеком. А получаемые результаты должны носить качественный характер и достаточно слабо зависеть от конкретного задания функций принадлежности. С другой стороны, если есть необходимость в более объективных выводах,

можно получить оценки  $\mu_A(x)$  путем опроса экспертов (см. подразд. 1.4).

Для нечетких множеств вводятся операции пересечения, объединения, дополнения, концентрации, размывания (табл. 1.5). Первые три являются обобщениями обычных операций; оставшиеся специфичны для нечетких множеств. Операции позволяют конструировать сложные понятия из простых: «очень много», «не старый и не молодой» и т. п.

По аналогии с четким случаем определяется отношение включения множеств:  $A \subset B$ , если и только если  $\mu_A(x) \leq \mu_B(x)$  для всех  $x \in X$ .

### 2.4.2. Нечеткие системы

Аналогично классическому случаю, понятие нечеткой системы вводится через понятие нечеткого отношения (частными случаями которого являются понятия «нечеткое отображение», «нечеткая функция»).

**Определение.** *Нечеткое отношение*  $R$  на множествах  $X, Y$  задается функцией  $\mu_R: X \times Y \rightarrow [0, 1]$ , каждое значение которой  $\mu_R(x, y)$  интерпретируется как степень нахождения (совместимости, принадлежности) пары  $(x, y)$  в данном отношении.

Таким образом, нечеткое отношение — это нечеткое подмножество множества  $X \times Y$  всех пар  $(x, y)$ , где  $x \in X, y \in Y$ . Поэтому стандартным способом вводятся пересечение, объединение, дополнение и другие действия над отношениями. Включение отношений  $R \subset S$  ( $\mu_R(x, y) \leq \mu_S(x, y)$ ) интерпретируется как «из  $R$  следует  $S$ ».

Важную роль в теории нечетких систем играет отношение композиции  $R \circ S$ . Если дано отношение  $R$  на множествах  $X, Y$  и отношение  $S$  на множествах  $Y, Z$ , то функция принадлежности отношения  $S \circ R$  на множествах  $X, Z$  задается формулой

$$\mu_{S \circ R}(x, z) = \max_{y \in Y} \{ \min \{ \mu_R(x, y), \mu_S(y, z) \} \}. \quad (2.36)$$

Можно проверить, что (2.36) выполняется для обычных, четких отношений.

В полной аналогии с обычными системами (см. подразд. 1.1) *нечеткая система* — это нечеткое отношение между множествами  $U, Y$ , где  $U$  — множество входных функций времени:  $U(t) : T \rightarrow U$ , а  $Y$  — множество выходных функций времени  $y(\cdot) : T \rightarrow Y$ . Операция композиции отношений соот-

ветствует последовательному соединению систем. Подчеркнем, что для нечетких систем понятие однозначности, детерминированности теряет смысл: нечеткое отображение и нечеткое отношение неразличимы.

Если множества значений входов и выходов системы конечны, то, как указывалось в подразд. 2.1, ММ системы можно задать таблицами, либо набором правил (продукций), например: «ЕСЛИ  $(u = u_i)$  И  $(x = x_j)$  ТО  $(y = y_k)$ » или в более компактном виде:

$$(u = u_i, x = x_j) \rightarrow y = y_k. \quad (2.37)$$

Форма (2.37) удобна для представления в ЭВМ и придает описанию системы вид набора причинно-следственных связей\*. Аналогично обстоит дело и для нечетких систем, входные и выходные переменные которых могут принимать нечеткие значения, т. е. являются лингвистическими. Примеры нечетких правил:

$$(u = \text{«малое»}) \rightarrow (y = \text{«большое»}),$$

$$(u(t) = \text{«около 0.5»}, x(t-1) = \text{«большое»}) \rightarrow (x(t) = \text{«очень большое»}).$$

**Пример 4.** Рассмотрим систему простейшего прогноза погоды в городе, основанную на том наблюдении, что погода чаще сохраняется, чем меняется: погода завтра скорей всего будет такая же, как сегодня. Для простоты пусть множество входов системы (возможных значений переменной «погода сегодня») состоит из трех элементов: «ясно» (Я), «пасмурно» (П), «дождь» (Д), т. е.  $U = \{Я, П, Д\}$ . Таким же пусть будет и множество выходов (прогнозов на завтра):  $Y = \{Я, П, Д\}$ . Если описать ММ системы простейшего прогноза как четкую, то ее можно представить таблицей:

U	Y		
	Я	П	Д
Я	1	0	0
П	0	1	0
Д	0	0	1

\* При этом фактически причинно-следственная связь может отсутствовать (пример: ЕСЛИ «тебе за сорок» И «у тебя ничего не болит» ТО «ты умер»).

или, более экономно, набором правил:

$$(u=\text{Я}) \rightarrow (y=\text{Я}); (u=\text{П}) \rightarrow (y=\text{П}); (u=\text{Д}) \rightarrow (y=\text{Д}).$$

Однако прогноз погоды — дело ненадежное и субъективное, поэтому более адекватной является нечеткая ММ, в которой отношение между входами и выходами системы  $R_0$  задается таблицей значений функции принадлежности, имеющей, например, такой вид:

U	Y		
	Я	П	Д
Я	0.8	0.4	0.3
П	0.4	0.8	0.4
Д	0.3	0.4	0.8

(2.38)

Можно использовать и нечеткие правила (продукции), например:

$$(u=\text{Я}) \rightarrow (y=\text{Я}|0.8 \text{ или } \text{П}|0.4 \text{ или } \text{Д}|0.3).$$

Пусть теперь входная переменная  $\omega_0$  («погода сегодня») приняла некоторое значение. Оно, естественно, должно быть нечетким (ведь нет четкой границы между значениями «ясно» и «пасмурно», да и дождь может идти не по всему городу) и определяться, например, по сообщениям экспертов. Пусть в результате усреднения мнений группы экспертов  $\omega_0$  задается как

x	Я	П	Д
$\mu_{\omega_0}(x)$	0.4	0.5	0.1

(2.39)

Как узнать прогноз на завтра? Вспомним, что множество — частный случай отношения и представим (2.39) как отношение  $W_0$  с фиктивным одноэлементным множеством входов и нечетким множеством выходов, заданным (2.39). Теперь легко понять, что значение переменной «погоды завтра» ( $\omega_1$ )

определился с помощью соответствующего отношения  $W_1$  по формуле композиции отношений (2.36)\*.

Таким образом,  $W_1 = R_0 \circ W_0$  и, значит, например,

$$\begin{aligned} \mu_{w_1}(Я) &= \max\{\min\{0.1, 0.3\}, \min\{0.5, 0.4\}, \min\{0.4, 0.8\}\} = \\ &= \max\{0.1, 0.4, 0.4\} = 0.4. \end{aligned}$$

Итоговая таблица прогноза на завтра имеет вид

у	Я	П	Д
$\mu_{w_1}(y)$	0,4	0,5	0,4

Полученный результат можно снова подать на вход системы прогноза и получить новый результат «прогноз на послезавтра»:

$$W_2 = R \circ W_1 = (R \circ R) \circ W_0.$$

Приведенный пример обнаруживает глубокую аналогию между понятиями системы и логического исчисления (аксиоматической или дедуктивной системы). Действие значения входа системы соответствует в исчислении аксиомам и фактам (утверждениям), а отношение «вход—выход» (ММ системы) — набору правил (продукций, импликаций) исчисления. Вычисление значения выхода системы  $B$  по значению ее входа  $A$  соответствует шагу логического вывода по правилу

$$(A, A \rightarrow B) \Rightarrow B. \quad (2.40)$$

Правило вывода, соответствующее композиции нечетких отношений, называется *композиционным правилом вывода* и составляет основу *нечеткой логики*. В нечеткой логике значения истинности предложений лежат от нуля до единицы; закон исключенного третьего не выполняется. Единственное отличие системы от исчисления: на каждом шаге работы системы используется только текущее состояние, а исчисление обрабатывает все утверждения, выведенные ранее. Оно устра-

\* Чтобы убедиться в этом, нужно рассмотреть все возможные случаи типа «завтра ясно» ЕСЛИ {«сегодня ясно» И («сегодня ясно» ВЛЕЧЕТ «завтра ясно»)} ИЛИ {«сегодня пасмурно» И («сегодня пасмурно» ВЛЕЧЕТ «завтра ясно»)} ИЛИ {«сегодня дождь» И («сегодня дождь» ВЛЕЧЕТ «завтра ясно»)}.

При этом надо учесть, что связки И, ИЛИ формализуются по табл. 1.5, а отношение ВЛЕЧЕТ задано в (2.38).

няется, если включить в состояние системы всю ее предысторию.

### 2.4.3. Задачи группировки и упорядочения

Нечеткие отношения, как и обычные, могут обладать специальными свойствами. Для отношения  $R: X \times X \rightarrow [0, 1]$  рассмотрим свойства:

- рефлексивность  $R(x, x) = 1$  для всех  $x \in X$ ;
- симметричность  $R(x, y) = R(y, x)$  для всех  $x, y \in X$ ;
- антисимметричность  $\min\{R(x, y), R(y, x)\} = 0$  при  $x \neq y$ ;
- транзитивность  $R(x, z) \geq \min\{R(x, y), R(y, z)\}$  для всех  $x, y, z \in X$ .

Отношение называется отношением *сходства*, если оно рефлексивно и симметрично. Рефлексивность и антисимметричность характеризуют отношение *доминирования*. Если к перечисленным свойствам добавляется свойство транзитивности, то отношение называют *эквивалентностью* или *порядком* соответственно. На основе введенных определений строятся процедуры решения двух практически важных задач.

**Задача группировки (кластеризации, таксономии).** На конечном множестве объектов  $\{x_1, \dots, x_n\}$  задано отношение сходства (для любых двух объектов задана степень их близости, похожести). Требуется разбить все множество на группы объектов, близких между собой. Эта задача встречается при классификации минералов, материалов по их свойствам, систематике биологических организмов (видов), определении психологической совместимости коллективов. Процедура группировки состоит в том, что сначала исходное отношение сходства  $R$  преобразуется в отношение эквивалентности  $\bar{R}$  путем транзитивного замыкания\*, а затем производится разбиение на классы эквивалентности при различной величине порога близости  $\alpha$ . К одному классу относятся объекты  $x, y$ , для которых  $\bar{R}(x, y) \geq \alpha$ . Пример группировки приведен на рис. 1. 10, где  $N$  — число классов эквивалентности.

---

\* Транзитивное замыкание — операция, которая строит по данному отношению  $R$  наименьшее транзитивное отношение, включающее  $R$ . Она выполняется по формуле

$$\bar{R}(x, y) = \max_{(n, y_1, \dots, y_n)} \min\{R(x, y_1), R(y_1, y_2), \dots, R(y_n, y)\}.$$

**Задача упорядочения.** На конечном множестве объектов  $\{x_1, \dots, x_n\}$  задано отношение доминирования, т. е. указаны степени предпочтения для каждой пары объектов. Требуется линейно упорядочить объекты — указать наилучший, затем следующий за ним и т. д. Задача возникает, например, при выборе наиболее предпочтительного проектного варианта. Парные предпочтения вариантов задаются независимыми экспертами или критериями сравнения. Ясно, что получается

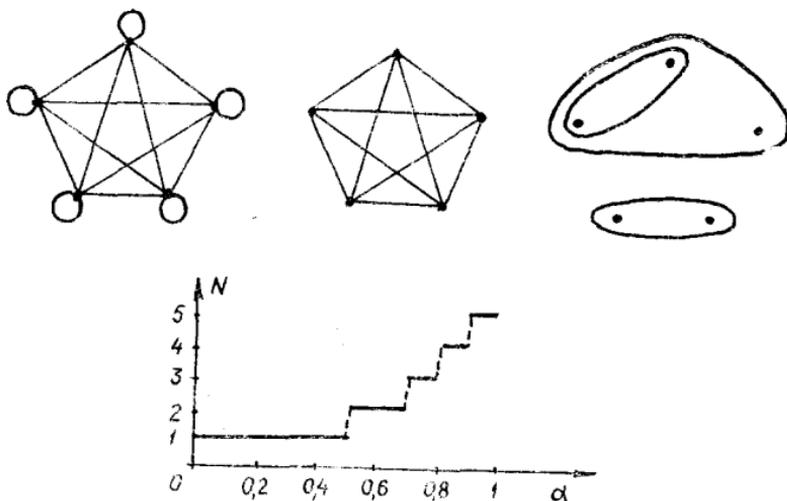


Рис. 1.10

нечеткое отношение доминирования, поскольку для каждой пары объектов  $(x_i, x_j)$  определяется  $R(x_i, x_j)$  — число экспертов\* (критериев), которые предпочитают  $x_i$ , и число  $R(x_j, x_i)$  экспертов (критериев), предпочитающих  $x_j$ . Аналогичные задачи возникают при оценке результатов работы и распределении премий в трудовом коллективе и т. п. Один из существующих способов решения задачи состоит [42] в построении так называемой *функции полезности*  $\varphi(x_i)$ , характеризующей степень предпочтения данного объекта, т. е. сводящей оценки

\* Строго говоря, числа  $R(x_i, x_j)$  не удовлетворяют определению нечеткого отношения, так как могут не лежать в промежутке  $[0, 1]$ . Можно их пронормировать, разделив, например, на максимальное, однако на результате, как видно из (2.41), это не отразится.

разных экспертов в одну. Можно использовать следующую формулу:

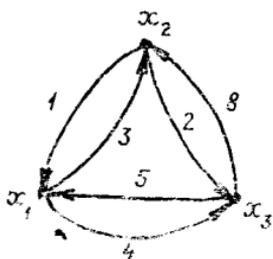
$$\varphi(x_i) = \min_{j (j \neq i)} \frac{R(x_i, x_j)}{R(x_j, x_i)}. \quad (2.41)$$

Например, для отношения  $R(x_i, x_j)$ , заданного таблицей или графом (рис. I. 11), получаем

$$\varphi(x_1) = \min \left\{ \frac{3}{1}, \frac{4}{5} \right\} = \frac{4}{5}; \quad \varphi(x_2) = \min \left\{ \frac{1}{3}, \frac{2}{8} \right\} = \frac{1}{4};$$

$$\varphi(x_3) = \min \left\{ \frac{8}{2}, \frac{5}{4} \right\} = \frac{5}{4},$$

что приводит к упорядочению: 1)  $x_3$ , 2)  $x_1$ , 3)  $x_2$ .



$x_i$	$x_j$		
	$x_1$	$x_2$	$x_3$
$x_1$	—	3	4
$x_2$	1	—	2
$x_3$	5	8	—

Рис. I. 11

Отметим, что, если число объектов достаточно велико или бесконечно, следует применять более сложные методы, разработанные для распознавания образов и искусственного интеллекта [43].

#### 2.4.4. Нечеткие числа

Рассмотрим свойства и применения нечетких подмножеств числовой оси  $R^1 = (-\infty, +\infty)$  — так называемых *нечетких чисел*. Над нечеткими числами можно производить арифметические и иные действия, правила выполнения которых вытекают из правил действий с отношениями (см. п. 2.4.2) и из того, что любую бинарную операцию можно рассматривать как тернарное (3-местное) отношение. Например, функция принадлежности нечеткой суммы  $C = A \oplus B$  нечетких чисел  $A, B$  имеет вид

$$\mu_C(z) = \sup_{x+y=z} \min \{ \mu_A(x), \mu_B(y) \}. \quad (2.42)$$

Прикладной смысл нечеткого числа — это число, заданное с погрешностью. Для того чтобы работать с такими числами, нужно задавать функции принадлежности погрешностей, а это невозможно сделать во всех  $x \in R^1$  в силу бесконечности множества  $R^1$ . Один из способов преодоления этой трудности — использование нечетких  $L-R$ -чисел ( $L-R$  — сокращение от «left—right»).

Чтобы определить нечеткие  $L-R$ -числа, на промежутке  $[0, \infty)$  задаются две невозрастающие неотрицательные функции  $L(x)$ ,  $R(x)$  со свойствами  $L(0) = R(0) = 1$ . После этого функцию принадлежности нечеткого числа  $A$  определяют в виде

$$\mu_A(x) = \begin{cases} L\left(\frac{a-x}{\alpha}\right) & \text{при } x \leq a, \\ R\left(\frac{x-a}{\beta}\right) & \text{при } x > a, \end{cases} \quad (2.43)$$

где  $a$  — вещественное число, называемое средним значением (употребляют также термины «центр», «мода») нечеткого числа;  $\alpha > 0$ ,  $\beta > 0$  — левый и правый коэффициенты нечеткости. Если  $L(x) = R(x)$ ,  $\alpha = \beta$ , то нечеткое число называют симметричным.

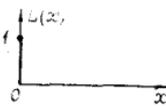
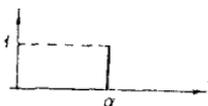
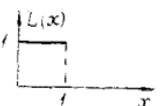
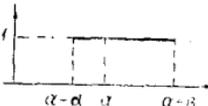
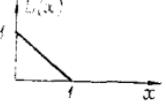
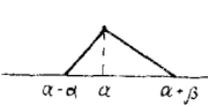
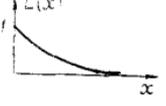
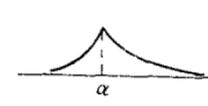
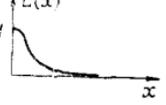
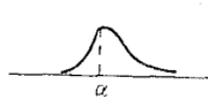
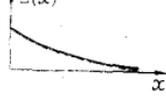
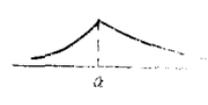
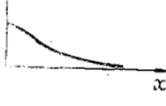
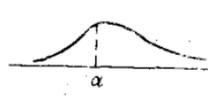
Примеры задания  $L-R$ -чисел при  $L(x) = R(x)$  приведены в табл. 1.6, из которой видно, что  $L-R$ -нечеткость можно интерпретировать как способ взвешенного учета погрешностей. Пример 1 соответствует обычному, четкому числу без погрешности. Выбрав способ из примера 2, мы выражаем уверенность в том, что искаженное погрешностью число должно лежать в промежутке от  $a - \alpha$  до  $a + \beta$  и не может оказаться где-то в другом месте. Это обычный способ учета погрешности путем указания ее границ. Примеры 3—6, 7 соответствуют различным промежуточным случаям.

Поскольку функции  $L(x)$ ,  $R(x)$  задаются заранее и не меняются, для выполнения действий с  $L-R$ -числами достаточно помнить лишь тройку  $A = \{a, \alpha, \beta\}$ . Правила арифметики  $L-R$ -чисел вытекают из общих правил арифметики нечетких чисел (см. напр. (2.42)) и напоминают правила распространения ошибок в приближенных вычислениях. Если  $A = \{a, \alpha, \beta\}$ ,  $B = \{b, \gamma, \delta\}$ , то

$$A \oplus B = \{a+b, \alpha+\gamma, \beta+\delta\},$$

$$A \ominus B = \{a-b, \alpha+\delta, \beta+\gamma\},$$

$$A \odot B = \{ab, a\gamma + b\alpha + \alpha\gamma, a\delta + b\beta + \beta\delta\} \quad (\text{при } a, b > 0).$$

№ п/п	Функция $L(x)$	График функции $L(x)$	График функции принадлежности $\mu_A(x, \alpha, \alpha, \beta)$
1	$L(x) = \begin{cases} 1, & x = 0 \\ 0, & x > 0 \end{cases}$		
2	$L(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & x > 1 \end{cases}$		
3	$L(x) = \begin{cases} 1-x, & 0 \leq x \leq 1 \\ 0, & x > 1 \end{cases}$		
4	$L(x) = e^{-x}$		
5	$L(x) = e^{-x^2}$		
6	$L(x) = \frac{1}{1+x}$		
7	$L(x) = \frac{1}{1+x^2}$		

Если  $B$  — четкое число  $\{\gamma = \delta = 0\}$ , то  $A \odot B = \{ab, \alpha|b|, \beta|b|\}$ .

Другой способ работы с нечеткими числами состоит в том, что числовая ось (отрезок) дискретизируется, т. е. разбивается на несколько участков. После этого каждый участок «размывается» — описывается некоторой функцией принад-

лежности и числовые переменные превращаются в лингвистические (см. рис. 1.10). Этот способ эффективен при построении моделей сложных систем со значительной неопределенностью исходных данных, а следовательно, и результатов. Число градаций (участков дискретизации) должно быть невелико. Психологи рекомендуют брать его в пределах  $5 \pm 2$  градаций.

### 2.4.5. Вероятность или нечеткость?

Продемонстрируем на простом примере разницу между стохастическим и нечетким подходами. Пусть сделано несколько измерений  $x_1, \dots, x_n$  некоторой неизвестной величины  $a$  с погрешностью, не превосходящей величины  $a$ . Требуется оценить значение  $a$  и определить погрешность оценки.

Предположим, что в качестве оценки выбрано среднее арифметическое  $x = \frac{1}{n} \sum_{i=1}^n x_i$ . При стохастическом подходе мы постулируем, что  $x_i$  случайны и независимы,  $Mx_i = a$  и, поскольку погрешность может быть произвольным числом из  $[-a, a]$ , считаем, что  $x_i$  равномерно распределены на  $[a-a, a+a]$ . Отсюда  $Dx_i = (2a)^2/12 = a^2/3$ . В силу независимости  $Dx = (1/n)Dx_i = a^2/3n$  и по формуле (2.33) из центральной предельной теоремы получим, что

$$|\bar{x} - a| \leq 2a/\sqrt{3n} \quad (2.44)$$

с вероятностью 0,95. Аналогичный вывод справедлив и при неизвестном  $a$ . В этом случае нужно заменить в (2.44)  $a$  на

$$\bar{a} = \sqrt{\frac{3}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}.$$

Примем теперь нечеткую модель измерений. Естественно представить измерение как нечеткое  $L-R$ -число  $X_i = \{a, a, a\}$  для  $L(x) = R(x) = 1$  при  $0 \leq x \leq 1$ ,  $L(x) = R(x) = 0$  при  $x > 1$ , т. е. как в примере 2 табл. 1.6. Тогда  $\Sigma X_i = \{na, na, na\}$ , откуда  $\bar{X} = \{a, a, a\}$ , т. е. погрешность оценки определится неравенством

$$|\bar{x} - a| \leq a. \quad (2.45)$$

Сравнивая (2.44) и (2.45), мы видим, что интервал (2.44) меньше примерно в  $\sqrt{n}$  раз. Это получено за счет эффекта усреднения. Если же нет уверенности в том, что погрешности ведут себя хаотически и уничтожаются при усреднении, то доверять (2.44) нельзя и мы возвращаемся к оценке (2.45). Однако за нечетким подходом остаются дополнительные возможности. Например, имея информацию о том, что малые значения погрешностей встречаются чаще, чем большие, мы можем взять функции  $L(x)$ ,  $R(x)$ , как в примере 3 из табл. I.6. Соответственно меняется функция принадлежности  $\bar{X}$ , и (2.45) уточняется.

Кроме того, если  $n$  мало, например  $n=10$ , то проверить правомерность усреднения практически невозможно. В результате оценка погрешности при  $n=10$  по (2.44) получается всего в 2,7 раза меньше, чем по (2.45), причем она верна лишь в 95% случаев и при труднопроверяемых предположениях.

Отметим, что выбор  $\bar{x}$  в качестве оценки не единственно возможный. В духе нечеткой теории можно взять «лингвистическую» оценку:  $\tilde{x} = x_1$  и  $x_2$ , и ..., и  $x_n$ . Поскольку  $\mu_{\tilde{x}}(x) = \min_i \mu_{x_i}(x)$ , для симметричных  $L$ — $R$ -чисел при строго убывающей  $R(x) > 0$  получим  $\tilde{x} = (\max x_i + \min x_i)/2$ .

В заключение следует заметить также, что, кроме описанных в (2.42) вариантов действий с нечеткими множествами, имеется еще несколько, среди которых наиболее употребителен способ, заимствованный из теории вероятностей:

$$\begin{aligned} \mu_{A \cap B}(x) &= \mu_A(x) \cdot \mu_B(x), \\ \mu_{A \cup B}(x) &= \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x). \end{aligned} \quad (2.46)$$

Операции, определенные в (2.46), называют иногда *произведением и суммой нечетких множеств*.

## 2.5. Принципы выбора модели

В заключение попытаемся сформулировать некоторые правила, помогающие на этапе выбора структуры ММ, — принципы выбора модели. Первейшим из них является принцип простоты: из различных вариантов структуры ММ сначала следует попробовать простейший. Например, если исследуется сложная динамическая (инерционная) система, то сначала нужно проверить, нельзя ли ограничиться статической

моделью, не учитывающей динамику. Как указывалось в п. 2.1.2, статическую модель заведомо можно применить, если система устойчива, изменения ее входов достаточно медленны, а выходы измеряются достаточно редко. Однако эти требования не обязательны и в рамках заданной цели, для которой ММ предназначена, могут быть ослаблены.

При уточнении структуры статической модели руководствуются тем же принципом. Например, если зависимость выхода от входа монотонна, то первым делом пробуют линейную ММ  $y = \theta_0 + \theta_1 u$ . Если зависимость выхода от входа носит экстремальный характер, то берут квадратичную функцию  $y = \theta_0 + \theta_1 u + \theta_2 u^2$ , а если есть подозрения, что зависимость выхода от входа имеет перегиб, то начинают с кубической функции  $y = \theta_0 + \theta_1 u + \theta_2 u^2 + \theta_3 u^3$ .

Аналогично действуют и в случае нескольких входов и выходов (см. подразд. 3.1, а также [44. С. 39—45]). Возможен различный выбор ММ в разных областях переменных. Например, если построение ММ выполняется с целью оптимизации, то вдали от экстремума можно ограничиться линейной моделью, а при приближении к экстремуму переходить на квадратичную. В любом случае предпочтительнее модели, в которые постоянные коэффициенты входят линейно (см. подразд. 3.1).

Если точность моделей с постоянными коэффициентами недостаточна, т. е. исследуемая система нестационарна, то в модель вводят зависимость коэффициентов от времени (дрейф, тренд). Дрейф может быть монотонным или периодическим, причем в большинстве случаев достаточно ограничиться простейшими моделями дрейфа: линейными  $\theta_i = a_i + b_i t$  или гармоническими  $\theta_i = a_i \cos(\omega_i t) + b_i \sin(\omega_i t)$ .

Если возникает дилемма: «выбрать ММ детерминированную или стохастическую», то предпочтение следует отдать детерминированной ММ. И только если не удастся обойтись без случайности, то вводят ее, причем сначала в наиболее простой форме, например, приведенной к выходу:  $y = F(u, \theta) + \varphi$ , где  $\varphi$  — случайное возмущение. Аналогично, переходить к нечеткой ММ следует только при неадекватности стохастических ММ (что в действительности бывает не так уж редко).

Иногда структуру системы характеризуют, кроме прочего, целым числом, определяющим порядок ее динамической модели (например, размерностью вектора состояния ММ, глубиной памяти ММ, максимальной внутренней задержкой

и т. п.). В соответствии с принципом простоты, при выборе ММ следует начинать с наименьших значений порядка, учитывая, что многие классы динамических процессов описываются моделями 1—2-го порядка. Например, для описания монотонных (так называемых «апериодических») переходных процессов часто достаточно апериодического звена — линейной стационарной дифференциальной ММ 1-го порядка, имеющей передаточную функцию (см. п. 2.2.2) от входа к вы-

ходу  $W(p) = \frac{K}{Tp + 1}$ , где  $K$  — коэффициент усиления статической ММ,  $T$  — постоянная времени звена\*.

Для описания колебательных процессов используется колебательное звено — линейное дифференциальное уравнение 2-го порядка, соответствующее передаточной функции  $W(p) = \frac{K}{T^2 p^2 + 2\xi T p + 1}$ ,

где  $\xi$  — показатель затухания колебаний ( $|\xi| < 1$ ). Перейти к ММ более высокого порядка следует лишь тогда, когда экспериментальные данные не позволят принять гипотезу об описании системы моделью данного порядка. Иначе говоря, «не решай сложную задачу, не решив сначала простую» (см. подразд. 1.2).

Отметим, что если ММ системы есть передаточная функция в непрерывном времени, то степень ее числителя не должна превосходить степени знаменателя. Иначе ММ будет включать чистое дифференцирование и ее нельзя будет привести к уравнениям состояния. Кроме того, на входные сигналы конечной мощности такая ММ будет реагировать сигналами бесконечной мощности, что говорит о ее *негрубости*. Понятие грубости ММ было введено в конце 20-х годов А. А. Андроновым и означает, что малые изменения исходных данных (входных переменных параметров) приводят к малым изменениям результатов (выходов). Грубость — очень важное свойство модели, так как реальные исходные данные для ее построения и применения всегда содержат ошибки. Если бы эти ошибки существенно сказывались на результатах, моделью просто нельзя было бы пользоваться. Резюмировать сказанное можно в виде принципа грубости: «без ошибки нет модели, а поэтому негрубые модели плохие».

Перечисленные правила следует принимать не как законы, а как рекомендации. В мире моделей царствует плюрализм

\* Величина  $T$  определяется тем свойством, что за время  $3T$  переходная составляющая процесса в апериодическом звене уменьшается до 5% от начального уровня.

и для достижения успеха нужно испытать несколько вариантов моделей. При этом самая полная модель не обязательно самая точная, а самая точная не обязательно самая хорошая. Здесь уместно перефразировать слова Э. Хэмингуэя\*: можно пренебрегать чем угодно, нужно только точно знать, как это повлияет на результат.

### 3. ВЫБОРЫ ПАРАМЕТРОВ МАТЕМАТИЧЕСКОЙ МОДЕЛИ

#### 3.1. Предварительные преобразования

##### 3.1.1. Линейно-параметризованные модели

Итак, мы выбрали структуру ММ системы, т. е. выбрали ММ с точностью до конечного набора числовых параметров. Как говорят, «модель параметризована». Что же дальше? Значения параметров лишь в редких случаях удается подобрать исходя только из теории или из априорных соображений. Как правило, оценка параметров ММ проводится по результатам наблюдений за реальным процессом или явлением в ходе нормального функционирования либо во время специальных экспериментов. Более того, без сопоставления результатов наблюдений за реальной системой и за ее ММ нет гарантий правильного выбора структуры ММ. Поэтому на этапе выбора параметров происходит уточнение и окончательный выбор структуры ММ, желательно из нескольких конкурирующих вариантов.

Задачи выбора параметров ММ (называемые также задачами идентификации) приходится решать для различных типов ММ: статических и динамических, дискретных и непрерывных, линейных и нелинейных и т. д. Однако во многих случаях имеющиеся результаты наблюдений и структуру ММ удается преобразовать к стандартной форме, позволяющей применять унифицированные методы идентификации. Такой формой является линейная по параметрам модель

$$y = \sum_{i=1}^N \theta_i x_i + \varphi, \quad (3.1)$$

где  $x_i$  — входные переменные (входы, факторы),  $y$  — выходная переменная (выход, отклик),  $\theta_i$  — параметры ММ,  $\varphi$  —

\* Э. Хэмингуэй давал следующий совет начинающим писателям: «Можно опускать что угодно, нужно только точно знать, что опускаешь».

возмущение (погрешность ММ). Если ввести векторные обозначения  $x = \text{col}(x_1, \dots, x_n) \in R^n$ ,  $\theta = \text{col}(\theta_1, \dots, \theta_n) \in R^n$ , то (3.1) можно переписать в виде

$$y = \theta^T x + \varphi. \quad (3.1a)$$

По виду (3.1) ясно, что эта ММ приспособлена для отражения количественных зависимостей, т. е. множества значений входов и выходов системы  $U, Y$  должны быть непрерывными. Этот случай (при случайных  $\varphi$  это задачи регрессионного анализа) будет подробно рассмотрен ниже. Иногда моделью (3.1) можно описать и зависимость между величинами с дискретным множеством значений (например  $x_i, y$  могут быть номерами уровней входных переменных, т. е. принимать значения 1, 2, 3, ...). Однако для исследования качественных зависимостей более приспособлены методы дисперсионного анализа [44—47], которые здесь рассматриваться не будут. Подчеркнем, что величины  $x, y, \theta, \varphi$ , вообще говоря, не совпадают с исходными входами, выходами, параметрами и возмущениями в системе, а получаются в результате преобразования (замены переменных), специально сделанного для приведения ММ системы к форме (3.1). Покажем, как можно перейти к форме (3.1) для основных типов моделей: статических; динамических, дискретных по времени; динамических, непрерывных по времени.

Для простоты изложения возмущение в (3.1) учитывать не будем.

### 3.1.2. Преобразование статических моделей

**Линейная ММ.** Чтобы привести линейную ММ с  $t$  входами

$$y = a_0 + a_1 u_1 + \dots + a_m u_m \quad (3.2)$$

к виду (3.1), полагаем  $x_1 = 1$ ,  $\theta_1 = a_0$  и для  $i = 2, \dots, m+1$ ,  $x_i = u_{i-1}$ ,  $\theta_i = a_{i-1}$ . Таким образом,  $x = \text{col}(1, u_1, \dots, u_m)$ ,  $\theta = (a_0, a_1, \dots, a_m)$ ,  $n = m+1$ .

**Нормированная линейная ММ.** Рекомендуется перед оценением параметров ММ проводить преобразование нормировки (масштабирование и центрирование) переменных, приводящее диапазоны изменения переменных к стандартному значению. Это преобразование для линейной ММ соответствует выбору в (3.1)  $x_i = (u_i - \bar{u}_i) / \sigma_i$ . Обычно выбирают числа  $\bar{u}_i, \sigma_i$ , так, чтобы нормированные переменные лежали в диа-

пазоне  $[-1, 1]$ , т. е.  $\bar{u}_i$  — «номинальное» или среднее значение переменной, а  $\sigma_i$  — максимальное отклонение от среднего. В стохастическом случае берут  $\bar{u}_i = M u_i$ ,  $\sigma_i = \sqrt{M(u_i - \bar{u}_i)^2}$ .

**Полиномиальная ММ.** Полиномиальная ММ с одним входом и выходом, имеющая вид

$$y = a_0 + a_1 u + a_2 u^2 + \dots + a_m u^m, \quad (3.3)$$

где  $m$  — предполагаемая степень полиномиальной зависимости, приводится к (3.1) выбором

$$x = \text{col}(1, u, \dots, u^m), \quad \theta = \text{col}(a_0, a_1, \dots, a_m), \quad n = m + 1.$$

**Квадратичная многофакторная ММ.** Квадратичная зависимость с несколькими входами, имеющая вид

$$y = a_0 + \sum_{i=1}^{\infty} a_i u_i + \sum_{i,j=1}^{\infty} a_{ij} u_i u_j, \quad (3.4)$$

может быть приведена к форме (3.1), если положить

$$x = \text{col}(1, u_1, \dots, u_m, u_1^2, u_1 u_2, \dots, u_1 u_m, u_2^2, u_2 u_3, \dots, u_m^2),$$

$$\theta = \text{col}(a_0, a_1, \dots, a_m, a_{11}, (a_{12} + a_{21}), \dots, (a_{1m} + a_{m1}),$$

$$a_{22}, (a_{23} + a_{32}), \dots, a_{mm}).$$

Причем, в силу симметрии, можно брать  $a_{ij} = a_{ji}$ . Общее число входов в (3.1) будет  $n = 1 + m + \frac{m(m+1)}{2} = \frac{(m+1)(m+2)}{2}$ .

**Мультипликативная (степенная) ММ.** Если ММ системы записана в мультипликативном виде:

$$\tilde{y} = C u_1^{\alpha_1} \cdot u_2^{\alpha_2} \cdot \dots \cdot u_m^{\alpha_m}, \quad (3.5)$$

то для приведения к виду (3.1) следует прологарифмировать (3.5):

$$\ln \tilde{y} = \ln C + \alpha_1 \ln u_1 + \dots + \alpha_m \ln u_m$$

и положить  $y = \ln \tilde{y}$ ,  $x_1 = 1$ ,  $\theta_1 = \ln C$  и для  $i = 2, \dots, m + 1$   $\theta_i = \alpha_{i-1}$ ,  $x_i = \ln u_{i-1}$ .

**Неявная ММ.** Если статическая характеристика системы представлена в виде неявной зависимости

$$\Phi(u, y, \theta) = 0, \quad (3.6)$$

то нет необходимости разрешать (3.5) относительно выхода. Достаточно свести (3.6) к виду

$$\Phi_0(u, y) + \sum_{i=1}^n \theta_i \Phi_i(u, y) = 0, \quad (3.7)$$

линейному относительно параметров и положить  $x = \text{col}(\Phi_1(u, y), \dots, \Phi_n(u, y))$ ,  $\tilde{y} = \Phi_0(u, y)$ .

Приведем несколько примеров, демонстрирующих приемы построения линейно-параметризованных ММ.

**Пример 1.** Одним из распространенных приемов является логарифмирование уравнений. Если ММ задана экспоненциальной зависимостью  $y = C_1 e^{C_2 u}$ , то, логарифмируя, получим соотношение  $\ln y = \ln C_1 + C_2 u$ , для приведения которого к виду (3.1) достаточно положить  $\tilde{y} = \ln y$ ,  $x_1 = 1$ ,  $x_2 = u$ ,  $\theta_1 = \ln C_1$ ,  $\theta_2 = C_2$ . После построения оценок параметров  $\hat{\theta}_1$ ,  $\hat{\theta}_2$  следует вернуться к параметрам исходной ММ:  $\hat{C}_1 = e^{\hat{\theta}_1}$ ,  $\hat{C}_2 = \hat{\theta}_2$ .

**Пример 2.** Уравнение Аррениуса, определяющее кинетику химических реакций, имеет вид

$$K \approx K_0 \exp\left(-\frac{E}{R(T+273)}\right), \quad (3.8)$$

где  $K$  — константа скорости реакции,  $T$  — температура (в °C),  $E$  — энергия активации,  $R$  — универсальная газовая постоянная. Величина  $R$  известна из теории, а  $K_0$ ,  $E$  подлежат определению из опытов. Прологарифмированное уравнение (3.8) сводится к (3.1) подстановкой  $y = \ln K$ ,  $x_1 = 1$ ,  $x_2 = \frac{-1}{R(T+273)}$ ,  $\theta_1 = \ln K_0$ ,  $\theta_2 = E$ .

**Пример 3.** Тригонометрическая модель

$$y = A \sin(\omega t + \varphi), \quad (3.9)$$

в которой  $\omega$ ,  $t$  известны,  $A$ ,  $\varphi$  неизвестны, сводится к (3.1), если записать (3.9) в виде  $y = A \cos \varphi \sin \omega t + A \sin \varphi \cos \omega t$  и обозначить  $x_1 = \sin \omega t$ ;  $x_2 = \cos \omega t$ ;  $\theta_1 = A \cos \varphi$ ;  $\theta_2 = A \sin \varphi$ . Обратный переход после построения оценок  $\hat{\theta}_1$ ,  $\hat{\theta}_2$  выполняется по формулам

$$\hat{A} = \hat{\theta}_1^2 + \hat{\theta}_2^2; \quad \hat{\varphi} = \text{arctg}(\hat{\theta}_2/\hat{\theta}_1).$$

**Пример 4.** Пусть в предыдущем примере  $\omega$  неизвестна. Тогда следует сначала оценить  $\omega$ , исключив из (3.9) остальные параметры. Это можно сделать, приведя двукратным диф-

ференцированием (3.9) к виду  $\ddot{y} + \omega^2 y = 0$  и оценив  $\omega^2$  одним из методов, описанных ниже, в п. 3.1.3. После этого можно применить подход предыдущего примера.

**Пример 5.** Гиперболическая зависимость  $y = u / (\alpha + \beta u)$  преобразуется к виду  $1/y = \alpha/u + \beta$ , который сводится к (3.1) заменой  $y = 1/y$ ,  $x_1 = 1/u$ ,  $\theta_1 = \beta$ ,  $\theta_2 = \alpha$ .

### 3.1.3. Преобразование динамических моделей

Проще всего обстоит дело с дискретными по времени линейными моделями, заданными передаточной функцией  $W(z^{-1}) = B(z^{-1})/A(z^{-1})$ , где  $A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_m z^{-m}$ ,  $B(z^{-1}) = b_0 + b_1 z^{-1} + \dots + b_m z^{-m}$ , или соответствующим разностным уравнением

$$y_k + a_1 y_{k-1} + \dots + a_m y_{k-m} = b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m}. \quad (3.10)$$

ММ (3.10) непосредственно приводится к (3.1), если положить

$$\begin{aligned} n &= m_u + m_y + 1; \quad y = y_k, \\ x &= \text{col}(-y_{k-1}, \dots, y_{k-m_y}, u_k, \dots, u_{k-m_u}), \\ \theta &= \text{col}(a_1, \dots, a_{m_y}, b_0, \dots, b_{m_u}). \end{aligned}$$

Если ММ задана уравнениями состояния

$$x_{k+1} = Ax_k + bu_k, \quad y = Cx_k, \quad (3.11)$$

то следует привести его к виду (3.10) (как это описано в п. 2.2.2). Уравнение (3.10) содержит меньше неизвестных параметров и включает только входные и выходные, т. е. измеряемые переменные. Если же измерению доступен непосредственно вектор состояния  $x_k \in R^n$ , то целесообразно представить векторное уравнение (3.11) как совокупность из  $n$  покомпонентных скалярных уравнений, коэффициенты которых (строки матриц  $A$ ,  $B$ ) оцениваются независимо. Аналогичные соображения можно применять и к нелинейным моделям.

Для оценки параметров непрерывных по времени динамических систем существуют два подхода. Первый состоит в переходе от непрерывной ММ к дискретной и оценивании параметров дискретной ММ. Способы дискретизации при-

ведены в п. 2.2.3. Затем полученные оценки  $\hat{\theta}$  параметров дискретной ММ пересчитываются в оценки исходной ММ путем континуализации (см. п. 2.2.3).

Второй подход состоит в сведении исходной динамической ММ системы к некоторой вспомогательной статической ММ. Для этого вводится функционал качества \*  $Q(\cdot)$ , характеризующий погрешность приближения ММ к исходной системе, вычисляемую по траектории системы на заданном промежутке  $[0, T]$ . Примером такого функционала является квадратичная ошибка  $Q_1 = \int_0^T \varepsilon^2(t) dt$ , где  $\varepsilon(t) = y(t) - \hat{y}(t)$ ,  $y(t)$  — реальный выход системы при входном воздействии  $u(t)$ ,  $\hat{y}(t)$  — процесс в ММ. Поскольку ММ задается вектором параметров  $\theta$ , т. е.  $\hat{y}(t) = \hat{y}(t, \theta)$  (начальные условия считаем фиксированными и одинаковыми в системе и в ММ), функционал  $Q$  оказывается функцией конечномерного вектора параметров:  $Q = Q(\theta)$ . Теперь задачу оценивания параметров можно поставить как поиск

$$\hat{\theta} = \arg \min_{\theta} Q(\theta), \quad (3.12)$$

т. е. как задачу оптимизации статической системы. Для ее решения можно использовать хорошо известные методы оптимизации [28, 29].

Перечислим некоторые, часто используемые виды функционалов качества:

— интегральная абсолютная ошибка  $Q_2 = \int_0^T |\varepsilon(t)| dt$ ;

— максимальная ошибка  $Q_3 = \sup_{0 \leq t \leq T} |\varepsilon(t)|$ ;

— терминальная ошибка на промежутке  $[0, T]$   $Q_4 = |\varepsilon(T)|$ .

Если ММ системы имеет вид уравнения состояния  $\hat{dx}/dt = F(\hat{x}, u, \theta)$ , где  $\hat{x} \in R^n$  — вектор состояния (см. подразд. 2.2), то можно формировать показатель качества из ошибки по состоянию. Например, квадратичный функционал можно задать в виде

$$Q_5 = \int_0^T \|x(t) - \hat{x}(t)\|^2 dt, \quad (3.13)$$

\* Функционалом называется отображение произвольного множества в множество чисел. В данном случае аргументом функционала является функция времени, значением — положительное число.

или в более общей форме:

$$Q_6 = \int_0^T [x(t) - \hat{x}(t)] R [x(t) - \hat{x}(t)] dt, \quad (3.14)$$

где  $R = R^T > 0$  — некоторая симметричная положительно определенная матрица весов.

Описанный подход отличается универсальностью, но обладает рядом недостатков.

1. Значение функционала качества зависит от длины промежутка  $T$  и от начальных условий  $x(0)$ . Вследствие этого величину  $T$  приходится брать достаточно большой, чтобы за время  $[0, T]$  переходный процесс успел закончиться. Минимизация должна проводиться либо при одном, фиксированном значении  $x(0)$ , либо по показателю  $\bar{Q}(\theta) = \max_{x(0) \in D} Q(\theta)$ ,

либо, наконец, по показателю  $\bar{Q}(\theta) = MQ(\theta)$  (при заданном «априорном» распределении вектора  $x(0)$ ).

2. Чтобы вычислить  $Q(\theta)$ , требуется определить траекторию системы на всем промежутке  $[0, T]$ , т. е. оценка точности и коррекция ММ производится только после окончания переходного процесса в системе.

3. Зависимость  $Q(\theta)$  обычно весьма сложна (в частности, неквадратична), вследствие чего условие оптимальности  $dQ/d\theta = 0$  представляет собой нелинейное относительно  $\theta$  уравнение. Это значит, что уравнение ММ не сводится к уравнению вида (3.1), даже если исходная ММ линейна по параметрам. Можно, конечно, линеаризовать условие оптимальности (этот подход основан на построении функции чувствительности [9]), однако при этом возникает дополнительная погрешность, с трудом поддающаяся оценке.

Последний недостаток особенно снижает привлекательность подхода, поскольку существуют другие варианты, позволяющие сводить задачу к линейной. Например, можно перейти к дискретной ММ (3.10) и задать показатель качества в виде

$$Q(\theta) = \sum_{k=0}^N (y_k - \theta^T x_k)^2. \quad (3.15)$$

Избавиться от последнего недостатка можно также, перейдя от функционалов ошибки, характеризующих разницу в выходах (или в состояниях) исходной системы и ее модели, к функционалам невязки, зависящим от погрешности уравне-

ния ММ. Например, заменив в (3.13) векторы состояния  $x(t)$ ,  $\hat{x}(t)$  их производными, приходим к показателю

$$Q_7 = \int_0^T \|\dot{x}(t) - F(x(t), u(t), \theta)\|^2 dt, \quad (3.16)$$

который в случае линейности  $F(x, u, \theta)$  по  $\theta$  является квадратичным по  $\theta$ . Надо сказать, что неквадратичность задачи не мешает применению целого ряда методов оптимизации [28]. По поводу оценивания параметров нелинейных систем см. также [48].

Продемонстрируем различные подходы на примере.

**Пример 5.** Пусть структура ММ задана дифференциальным уравнением 1-го порядка

$$\dot{y} = ay + bu, \quad (3.17)$$

параметры которого  $a$ ,  $b$  подлежат оцениванию по результатам измерений  $y_k = y(t_k)$ ,  $u_k = u(t_k)$  при  $t_k = kh$ ,  $k = 0, 1, \dots, N$ . Перечислим возможные способы формализации задачи оценивания параметров в (3.17).

а). Дискретизируя (3.17) по методу Эйлера (см. п. 2.2.3), получаем дискретную ММ в виде

$$y_{k+1} = (1 + ah)y_k + bhu_k. \quad (3.18)$$

Для сведения (3.17) к (3.1) положим  $\theta_1 = 1 - ah$ ,  $\theta_2 = bh$ ,  $y = y_{k+1}$ ,  $x_1 = y_k$ ,  $x_2 = u_k$ . После расчета оценок параметров (3.18)  $\hat{\theta}_1$ ,  $\hat{\theta}_2$  необходимо пересчитать их в оценки параметров исходной системы (3.17) по обращенным формулам

$$\hat{a} = (1 - \hat{\theta}_1)/h; \quad b = \hat{\theta}_2/h. \quad (3.19)$$

б). Заметим, что в формулах (3.19) присутствует деление на величину шага  $h$ , которая может быть малой величиной. Чтобы избежать потери точности, можно попробовать взять дискретную ММ в виде

$$(y_{k+1} - y_k)/h = ay_k + bu_k. \quad (3.20)$$

Для сведения к (3.1) полагаем

$$y = (y_{k+1} - y_k)/h; \quad \theta_1 = a; \quad \theta_2 = b; \quad x_1 = y_k, \quad x_2 = u_k.$$

Отметим также, что формы (3.18), (3.20) не изменятся, если «продифференцировать» их по времени, перейдя к приращениям:  $y_k \rightarrow y_k - y_{k-1}$ ,  $u_k \rightarrow u_k - u_{k-1}$ . Этот прием используется

для повышения точности оценок за счет увеличения «разнообразия» векторов  $x_k$ .

в). Можно дискретизовать (3.17) по методу Гастина (см. п. 2.2.3). Для этого в передаточной функции системы (3.17)  $W(p) = b/(p - a)$  делается подстановка  $p \rightarrow \frac{2}{h} \frac{1 - z^{-1}}{1 + z^{-1}}$  и результат умножается на  $2/(1+z)$ . Получим дискретную передаточную функцию

$$W_D(z) = \frac{2bh}{(2 - ah)z - (2 + ah)},$$

соответствующую разностному уравнению

$$y_{k+1} = \frac{2 + ah}{2 - ah} y_k + \frac{2bh}{2 - ah} u_k. \quad (3.21)$$

Заметим, что если вычесть из обеих частей (3.21)  $y_k$  и разделить на  $h$ , то получим модель

$$\frac{y_{k+1} - y_k}{h} = \frac{a}{1 - ah/2} y_k + \frac{b}{1 - ah/2} u_k, \quad (3.22)$$

структура которой совпадает с (3.20). Отличие от варианта б) есть лишь в формулах пересчета оценок, прямых:

$$\theta_1 = \frac{a}{1 - ah/2}; \quad \theta_2 = \frac{b}{1 - ah/2}$$

и обратных:

$$\hat{a} = \frac{\hat{\theta}_1}{1 + \hat{\theta}_1 h/2}; \quad \hat{b} = \theta_2 (1 + \hat{\theta}_1 h/2).$$

г). Проинтегрировав обе части (3.17) по отрезку  $[0, t_k]$ , получим ММ (3.1) при  $y = y_k - y_0$ :

$$x_1 = \int_0^{t_k} y(t) dt; \quad x_2 = \int_0^{t_k} u(t) dt; \quad \theta_1 = a; \quad \theta_2 = b.$$

Приближенно можно взять  $x_1 = h \sum_{i=1}^k y_i$ ,  $x_2 = h \sum_{i=1}^k u_i$ .

д) Наконец, можно поставить задачу оценивания параметров (3.17) как минимизацию функции интегральной ошибки:

$$Q_1(a, b) = \int_0^T [y(t) - \hat{y}(t, a, b)]^2 dt,$$

где  $\hat{y}(t, a, b)$  — решение уравнения  $d\hat{y}/dt = a\hat{y} + bu$  с начальным условием  $\hat{y}(0, a, b) = y(0)$ , или интегральной невязки

$$Q_2(a, b) = \int_0^T [dy/dt - ay(t) - bu(t)]^2 dt.$$

### 3.2. Регрессионный анализ и метод наименьших квадратов

После приведения ММ к стандартной форме (3.1) вычисляются оценки параметров ММ по результатам ряда наблюдений за входами и выходами системы. Пусть имеются результаты  $N$  наблюдений  $x_{i1}, \dots, x_{in}, y_i; i=1, \dots, N$ , которые объединены в матрицу из  $N$  строк и  $n$  столбцов ( $N \times n$ -матрицу)  $X = \{x_{ij}\}$  и  $N$ -вектор  $Y = \text{col}(y_1, \dots, y_N)$ .

Начиная с классической работы К. Гаусса (1809 г.), популярен способ вычисления оценок, называемый методом наименьших квадратов (МНК) и состоящий в выборе оценки  $\hat{\theta}$ , минимизирующей выражение (3.15) — среднеквадратическую ошибку. МНК-оценка удовлетворяет системе так называемых «нормальных уравнений»

$$X^T X \hat{\theta} = X^T Y \quad (3.23)$$

и имеет вид

$$\hat{\theta} = C X^T Y, \quad (3.24)$$

где  $C = (X^T X)^{-1}$  — так называемая «дисперсионная матрица». Для вычисления  $\hat{\theta}$  не требуется никаких предположений о свойствах измерений, однако чтобы сделать выводы о точности и достоверности ММ, такие предположения нужны. Часто принимается следующая гипотеза: если представить модель измерений в виде

$$Y = X\theta + \varphi, \quad (3.25)$$

где  $\varphi = \text{col}(\varphi_1, \dots, \varphi_n)$  — вектор погрешностей (невязок) ММ, то  $\varphi_i$  случайны, независимы, центрированы ( $M\varphi_i = 0$ ) и равнозначны (имеют одинаковую дисперсию  $\sigma^2 = M\varphi_i^2$ ). Эти предположения традиционны для раздела теоретической и прикладной статистики, называемого регрессионным анализом (РА). При их выполнении МНК-оценки  $\hat{\theta}$  обладают следующими свойствами:

1.  $M\hat{\theta} = \theta$  (несмещенность).

2.  $\text{cov}(\hat{\theta}) = \sigma^2 C$ , где  $\text{cov}(\hat{\theta}) = M(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T$  — ковариационная матрица оценок. В частности,

$$M(\hat{\theta}_i - \theta_i)^2 = \sigma c_{ii}. \quad (3.26)$$

3. МНК-оценки эффективны: для любой несмещенной оценки  $\theta_*$  выполняется неравенство  $M\|\theta_* - \theta\|^2 \geq M\|\hat{\theta} - \theta\|^2$ .

4. Если обозначить через  $\hat{Y} = X\hat{\theta}$   $N$ -вектор прогноза выходных величин  $Y$  по модели (3.25), а через  $R = \|Y - \hat{Y}\|^2 = \sum_{i=1}^N (y_i - \hat{y}_i)^2$  — среднеквадратичную ошибку прогноза (остаточную сумму квадратов), то  $MR = (N - r)\sigma^2$ , где  $r$  — ранг матрицы  $X^T X$ . Как правило (если входы модели (3.1) не связаны линейной зависимостью),  $r = n$  и, значит, оценку дисперсии погрешностей модели (3.25) можно брать в виде

$$\hat{\sigma}^2 = R^2 / (N - n). \quad (3.27)$$

После построения ММ (3.25) немедленно возникает вопрос о ее точности. Этот вопрос имеет две стороны:

а) нельзя ли без существенной потери точности заменить модель (3.25) более простой, т. е. упростить?

б) является ли модель (3.25) адекватной исходной системе, т. е. не следует ли ее усложнить?

Ответ в обоих случаях требует дополнительных предположений. В первом варианте будем считать, что модель (3.25) адекватна и выполнены предложения РА. Тогда можно поставить задачу о проверке гипотезы  $\theta_i = 0$  — равенства нулю некоторого коэффициента ММ. На языке математической статистики это задача проверки значимости коэффициента  $\theta_i$ . В силу свойств 2, 3 МНК-оценок и при дополнительном условии нормальности  $\varphi$  (которое не кажется слишком обременительным на фоне остальных допущений), для проверки гипотезы  $\theta_i = 0$  можно пользоваться следующей процедурой:

1) по заданному уровню надежности  $p$  и числу наблюдений  $N$  из таблиц распределения Стьюдента (см., напр., [58]) выбирается значение  $\alpha_p$ , такое, что  $P\{|t| \leq \alpha_p\} = p$ , где  $t$  — случайная величина, распределенная по закону Стьюдента. Обычно берут  $p = 0.9, 0.95, 0.99$ ;

2) определяются границы доверительного интервала для параметра  $\theta_i$  при уровне надежности  $p$ :

$$[\hat{\theta}_i - \alpha_p \hat{\sigma} \sqrt{c_{ii}}, \hat{\theta}_i + \alpha_p \hat{\sigma} \sqrt{c_{ii}}] \quad (3.28)$$

3) если интервал (3.28) содержит точку  $\theta_i=0$ , т. е. если  $|\hat{\theta}_i| < \alpha_p \hat{\sigma} \sqrt{c_{ii}}$ , то гипотеза  $\theta=0$  принимается, т. е. в ММ можно положить  $\theta_i=0$ . В противном случае коэффициент  $\theta_i$  с надежностью  $p$  считается значимым, т. е. гипотеза  $\theta_i=0$  отвергается.

Отметим, что при  $N \geq 20 \dots 30$  вместо таблиц распределения Стьюдента можно пользоваться таблицами нормального распределения и брать, в частности  $\alpha_{0,9}=1.65$ ;  $\alpha_{0,95}=1.96$ ;  $\alpha_{0,99}=2.58$ . Впрочем, при меньших  $N$ , как указывалось в подразд. 2.3, методы математической статистики вряд ли дадут надежные и нетривиальные выводы.

Для ответа на второй вопрос требуется выполнение предложений РА и возможность проведения в каждой точке  $x_{i1}, \dots, x_{in}$  нескольких независимых случайных измерений  $y_{i1}, \dots, y_{in}$ , чтобы по ним построить альтернативную оценку дисперсии ошибки

$$\hat{\sigma}^2 = \frac{1}{N(v-1)} \sum_{i=1}^N \sum_{j=1}^v (y_{ij} - \bar{y}_i)^2, \quad \bar{y}_i = \frac{1}{v} \sum_{j=1}^v y_{ij}. \quad (3.29)$$

Гипотеза об адекватности ММ (3.25) отвергается, если величина  $F = \hat{\sigma}^2 / \hat{\sigma}^2$  (дробь Фишера) оказывается достаточно велика:  $F > F_p$ . Порог  $F_p$  ищется по заданному уровню надежности  $p$  из таблиц распределения Фишера [58].

Возникает еще один практически важный вопрос: нельзя ли повысить точность и надежность результатов эксперимента путем его планирования, т. е. специального выбора точек  $x_{i1}, \dots, x_{in}$ ? Очевидно, для повышения точности оценок нужно стараться сделать дисперсионную матрицу  $C$  «поменьше», а матрицу  $X$  — «побольше». О том, как это сделать, написано в книгах по планированию регрессионных экспериментов (например [46, 47, 58]).

До сих пор мы придерживались мнения о стохастичности модели (3.25). Если же более оправдан детерминистский подход и на возмущения в (3.25) накладывается единственное ограничение

$$|\varphi_i| < \Delta_{\varphi}, \quad (3.30)$$

то точность оценивания, как, впрочем, и сами оценки, можно определить из решения соответствующей минимаксной задачи:

$$\min_{\theta} \max_{1 < i < N} |y_i - \sum_{j=1}^n \theta_j^T x_{ij}|.$$

Минимаксный подход, хотя и требует более сложных вычислений (решения задачи линейного программирования или минимизации негладкой функции [28]), дает гарантированный результат при любом числе наблюдений  $N$ .

Во всех случаях выбор подхода, числа учитываемых параметров ММ, плана эксперимента является неформальным актом, основанным на опыте и интуиции исследователя. Здесь оказывается полезным следующий «принцип надежности»: «чем проще модель, тем реже она обманет» или «чем лучше модель объясняет прошлое, тем хуже она прогнозирует будущее». В иной форме эту же мысль выразил Р. Акофф: «Степень понимания явления обратно пропорциональна числу переменных в его описании». Практически часто применяют следующее правило: число независимых наблюдений должно быть в 3...5 раз больше числа параметров модели. Наконец, не следует забывать принцип «равнопрочности», который настойчиво пропагандировал знаменитый математик, механик и кораблестроитель А. Н. Крылов: «Точность результатов не может быть выше точности исходных данных; точности промежуточных вычислений должны быть согласованы».

### 3.3. Адаптивные модели и рекуррентные методы

Если наблюдений при построении ММ оказывается много и они поступают последовательно во времени, то часто оказывается удобным обрабатывать их по мере поступления. Иногда такой способ обработки единственно возможный, например, когда исходные данные из-за большого объема не помещаются в памяти ЭВМ или когда свойства системы меняются во времени («дрейфуют») и параметры ММ требуют постоянной коррекции. Модели, которые изменяются (подстраиваются, адаптируются) в процессе наблюдений за системой, называются *адаптивными*. Соответственно методы оценивания параметров таких моделей путем коррекции по текущим наблюдениям называются *адаптивными* или *рекуррентными*. [9, 77].

Общий вид рекуррентного алгоритма следующий:

$$\theta_k = \Phi(\theta_{k-1}, y_k, u_k). \quad (3.31)$$

Многие алгоритмы можно привести к рекуррентной форме.

**Пример 1.** Пусть ММ системы имеет вид  $y_i = \theta + \varphi_i$ , где  $\theta = \mu_{y_i}$  — оцениваемый параметр,  $\varphi_i$  — помеха,  $M\varphi_i = 0$ . Обычная оценка среднего арифметического  $\theta_i = \sum_{l=1}^k y_l$  может быть преобразована к рекуррентной форме:

$$\theta_k = \theta_{k-1} + \frac{1}{k} (y_k - \theta_{k-1}) \quad (3.32)$$

(при  $\theta_0 = 0$ ). Обобщением (3.32) является метод стохастической аппроксимации [28]:

$$\theta_k = \theta_{k-1} + \gamma_k (y_k - \theta_{k-1}), \quad (3.33)$$

где  $\gamma_k > 0$  — коэффициенты алгоритма, которые для обеспечения сходимости оценок должны удовлетворять условиям Роббинса—Монро:

$$\sum_{k=1}^{\infty} \gamma_k = \infty; \quad \sum_{k=1}^{\infty} \gamma_k^2 < \infty.$$

**Пример 2.** Можно показать, что МНК-оценки  $\theta_k$  параметров ММ (3.1) удовлетворяют рекуррентным соотношениям

$$\theta_k = \theta_{k-1} + \Gamma_k x(k) [y_k - \hat{\theta}^T_{k-1} x(k)], \quad (3.34a)$$

$$\Gamma_k = \Gamma_{k-1} - \frac{\Gamma_{k-1} x(k) x(k)^T \Gamma_{k-1}}{1 + x(k)^T \Gamma_{k-1} x(k)}, \quad (3.34b)$$

где  $x(k) = \text{col}(x_{k1}, \dots, x_{kn})$  — вектор  $k$ -го измерения входов ММ,  $\Gamma_k = \Gamma_k^T > 0$  — симметричная положительно определенная  $n \times n$  матрица. При этом  $\Gamma_k = [\sum_{i=1}^k x(i) x(i)^T]^{-1}$  — дисперсионная матрица, и, значит, соотношения (3.34) соответствуют МНК лишь при определенном выборе начальных матриц  $\Gamma_k$ ;  $k=0, 1, \dots, n-1$ . Однако, если  $k \rightarrow \infty$ , то оценки, получаемые из (3.34), приближаются к МНК-оценкам при произвольных начальных условиях  $\theta_0, \Gamma_0$ . Практически часто берут  $\Gamma_0 = \gamma I$ , где  $I$  — единичная матрица,  $\gamma > 0$  — достаточно большое число.

Если (3.34b) не учитывать, а брать  $\Gamma_k = \gamma_k I$ , где  $\gamma_k > 0$  удовлетворяют условиям Роббинса—Монро, получим обобщенный (многомерный) алгоритм стохастической аппроксимации:

$$\theta_k = \theta_{k-1} + \gamma_k (y_k - \theta^T_{k-1} x(k)) x(k). \quad (3.35)$$

Алгоритм (3.35), в свою очередь, является частным случаем *градиентного алгоритма*\*

$$\theta_k = \theta_{k-1} - \gamma_k \frac{\partial Q}{\partial \theta} (\theta_{k-1}, y_k, x_k) \quad (3.36)$$

и получается из (3.36) при выборе целевой функции текущей ошибки  $Q(\theta, y, u)$  в виде

$$Q(\theta, y, x) = \frac{1}{2} (y - \theta^T x).$$

Если погрешности модели (3.1) не случайны, а обладают лишь ограниченностью, т. е. стохастический подход неприменим, то алгоритмы (3.35), (3.36) сохраняют работоспособность при определенном выборе шагов  $\gamma_k$ . Например, при выполнении (3.30) алгоритм (3.35) будет обеспечивать при достаточно большом числе наблюдений  $k$  неравенство  $|y_k - \theta_k^T x(k)| < \Delta$ , если  $\Delta > \Delta_\Phi$ , векторы  $x(k)$  ограничены  $\|x(k)\| \leq \leq \text{const}$  и коэффициенты шага выбраны в виде

$$\gamma_k = \begin{cases} \gamma / \|x(k)\|^2 & \text{при } |y_k - \theta_k^T x(k)| > \Delta, \\ 0 & \text{при } |y_k - \theta_k^T x(k)| \leq \Delta, \end{cases} \quad (3.37)$$

где  $0 < \gamma < 2(1 - \Delta_\Phi / \Delta)$ . Алгоритм получил известность под названием «полоска». Свойства подобных алгоритмов и их обобщения изучены в рамках *метода рекуррентных целевых неравенств* [9, 59].

Градиентные адаптивные алгоритмы (3.36) применяются для идентификации систем статических и дискретных динамических. Аналогом этих алгоритмов для непрерывных динамических систем являются *алгоритмы скоростного градиента* [9, 59, 66]

$$d\theta/dt = -\gamma \partial v / \partial \theta, \quad (3.38)$$

где  $v = Q(\theta, y, x)$  — скорость изменения текущей целевой функции  $Q(\dots)$  в силу уравнения ММ.

Адаптивные алгоритмы благодаря своей простоте находят применение в бортовых вычислительных устройствах, при микропроцессорной реализации систем управления, контроля, оценки в различных областях.

\* Через  $\frac{\partial Q}{\partial \theta}$  обозначается градиент функции векторного аргумента  $\theta = \text{col}(\theta^{(1)}, \dots, \theta^{(n)})$ , т. е. вектор из частных производных  $\left( \frac{\partial Q}{\partial \theta} = \text{col} \left( \frac{\partial Q}{\partial \theta^{(1)}}, \dots, \frac{\partial Q}{\partial \theta^{(n)}} \right) \right)$ .

## Часть II

# АНАЛИЗ МОДЕЛЕЙ

---

Основным классом рассматриваемых моделей являются уравнения состояния:

$$\dot{x} = F(x, u, t), \quad (\text{II. 1})$$

$$y = G(x, u, t), \quad (\text{II. 2})$$

где  $x, y, u$  — векторы размерностей  $n, m, l$ , соответственно,  $t$  — независимая переменная (обычно время). Рассматриваются также более общие алгебро-дифференциальные уравнения:

$$F(x, x, u, t) = 0, \quad (\text{II. 3})$$

$$G(y, x, u, t) = 0, \quad (\text{II. 4})$$

а также линейные стационарные модели:

$$\dot{x} = Ax + Bu, \quad (\text{II. 5})$$

$$y = Cx + Du, \quad (\text{II. 6})$$

являющиеся частным случаем (II. 1), (II. 2).

### 1. АНАЛИЗ СТАТИКИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СИСТЕМ

Задача анализа статики возникает обычно при определении статических характеристик системы или начальных условий для задачи анализа динамики и, как правило, предшествует последней. Поэтому основной целью расчета статики является определение неизвестных векторов выхода  $y_* = y(t_*)$  и состояния  $x_* = x(t_*)$  по заданным постоянным значениям вектора входа  $u_* = u(t_*)$  для фиксированного момента времени  $t_*$  при условии, что  $x(t_*) = 0$ . В зависимости от вида вектор-функции, описывающей связи в ММС, для заданных  $u_*$  и  $t_*$  может быть несколько статических режимов. В этом случае ММС называется многостабильной и решение задачи анализа статики неоднозначно.

Существует несколько подходов к решению задач анализа статики.

Первый подход (метод установления) основан на численном интегрировании исходной системы обыкновенных дифференциальных уравнений (II.1) — (II.6) при  $0 \leq t < \infty$ . Решение такой системы при неизменном векторе входов  $u_*$  с некоторых исходных значений состояния  $x(0)$  через достаточный промежуток моделируемого времени должно привести к приближенно стационарной точке, в которой  $x(t_*) \approx 0$ . Эта точка и будет точкой решения  $y(t_*)$ ,  $x(t_*)$ . Очевидно, такой подход даст результаты, только если исходная ММС является асимптотически устойчивой. Если ММС имеет несколько статических режимов, то получение других стационарных точек возможно повторением процесса решения при других векторах начальных условий  $x(0)$ , однако практически реализуемые способы выбора  $x(0)$ , гарантирующие определение всех стационарных точек, в общем случае отсутствуют [30]. Поэтому при анализе статики таких объектов следует исходить из анализа области применимости ММС и практического опыта.

Подход к анализу статики через интегрирование дифференциальных уравнений численными методами получил достаточно широкое распространение [30, 6, 7], так как позволяет с помощью одних и тех же методов решать как статические, так и динамические задачи. Однако при этом, как правило, получаются завышенные затраты машинного времени.

Более экономичны подходы, в которых в качестве ММС берется система нелинейных алгебраических уравнений (СНАУ):

$$F(v) = 0, \quad (1.1)$$

где  $v \in R^k$  — вектор неизвестных,  $F(\cdot) \in R^k$  — заданная вектор-функция.

Второй подход (итерационный) основан на применении методов решения СНАУ. В большинстве случаев эти методы являются итерационными. При этом задаются начальным приближением  $v_0$  и выполняют ряд однотипных итераций. Если вектор приближений на  $k$ -й итерации  $v_k$  стремится к вектору решения (1.1)  $v_*$ , то говорят, что метод сходится.

Частным случаем системы (1.1) является система линейных алгебраических уравнений (СЛАУ):

$$Mv = d, \quad (1.2)$$

где  $M \in R^{k \times k}$  — известная матрица,  $d \in R^k$  — заданный вектор. В этом случае решение (1.2)  $v_*$  может быть получено как прямыми, так и итерационными методами.

Любая из ММС (II.1) — (II.6) в рамках задачи анализа статики может быть приведена к ММС вида (1.1). Например, полагая в (II.1)  $x(t) = 0$  и заменяя в (II.3) и (II.4)  $u(t)$  на известный вектор  $u_*$ ,  $t$  на  $t_*$ , получаем СНАУ вида

$$\begin{aligned} f_1(0, x, u_*, t_*) &= 0, \\ g_1(y, x, u_*, t_*) &= 0, \end{aligned} \quad (1.3)$$

с неизвестными векторами  $y \in R^l$  и  $x \in R^n$ . Система (1.1) получается из (1.3), если ввести новый вектор неизвестных  $v = \text{col}(x, y)$  и вектор-функцию  $F(\cdot) = \text{col}(f_1(\cdot), g_1(\cdot))$ . Аналогично для линейной формы описания ММС (II.5) получается СЛАУ (1.2) с  $M = A$ ,  $v = x$ ,  $d = -Bu_*$  и дополнительным уравнением связи (II.6) для вектора выходов  $y$ .

В третьем подходе к решению задач анализа статики последние представляются как экстремальные. Для этого образуется скалярная целевая функция, имеющая минимум в точке решения задачи  $v_*$ . Например, для (1.1) такой функцией  $\varphi(v)$  может быть сумма квадратов невязок:

$$\varphi(v) = \|F(v)\|^2 = \sum_{i=1}^k f_i^2(v). \quad (1.4)$$

Вычисление  $v_*$  производится с применением методов оптимизации [28—30]. Функцию (1.4) следует использовать с известной осторожностью, так как может быть ухудшена численная обусловленность задачи. Для улучшения обусловленности можно использовать взвешенную целевую функцию

$$\varphi_p(v) = \sum_{i=1}^k p_i f_i(v), \quad (1.5)$$

где  $p_i \geq 0$  — весовые коэффициенты, которые полезны также для учета масштабов и важности компонентов невязки.

## 1.1. Аналитические методы

Аналитические методы для СНАУ (1.1) позволяют получить решение в виде формул. Точные методы позволяют получить решение, используя аналитические (формульные) за-

висимости. Приближенные методы основаны на построении аналитических зависимостей, аппроксимирующих решение.

### 1.1.1. Метод возмущений

Пусть требуется найти решение СНАУ (1.1) вида

$$F(x, \varepsilon) = 0, \quad (1.6)$$

где  $x \in R^n$  — неизвестный вектор,  $\varepsilon$  — числовой малый параметр. Предположим, что при  $\varepsilon = 0$  система (1.6) может быть сравнительно легко решена и найдено приближение  $x_0$ . Возьмем в качестве нового приближения к решению  $x_*$   $x = x_0 + \varepsilon \hat{x}_1$ . Подставим  $x$  в (1.6) и разрешим ее относительно  $\hat{x}_1$ . Продолжая аналогично, можно получить некоторый итеративный процесс, при котором решение (1.6) представляется в виде

$$x = \sum_{i=0}^{\infty} \varepsilon^i x_i. \quad (1.7)$$

Если при  $\varepsilon = 0$  вектор-функция  $F(x, 0)$  хорошо приближает  $F(x, \varepsilon)$ , то методы подобного типа оказываются весьма эффективными на практике.

**Пример 1.** В качестве примера рассмотрим решение нелинейного уравнения:

$$f(x) = \sin(x) - \frac{\varepsilon}{x} = 0. \quad (1.8)$$

Положим в (1.8)  $\varepsilon = 0$  и в качестве начального приближения возьмем ненулевое решение уравнения  $\sin(x) = 0$ , например,  $x_0 = \pi$ . Представим новое приближение  $x_1 = x_0 + \varepsilon \hat{x}_1 = \pi + \varepsilon \hat{x}_1$  и подставим его в (1.8). Используя тождество  $\sin(\pi + \alpha) = -\sin(\alpha)$ , получим  $-\sin(\varepsilon \hat{x}_1) + \frac{\varepsilon}{\pi + \varepsilon \hat{x}_1} = 0$ . Так как при малых  $\varepsilon \hat{x}_1$   $\sin(\varepsilon \hat{x}_1) \sim \varepsilon \hat{x}_1$ , то имеем уравнение  $-\varepsilon \hat{x}_1 + \frac{\varepsilon}{\pi + \varepsilon \hat{x}_1} = 0$ , откуда  $\hat{x}_1 = 1/\pi$  и очередное приближение  $x_1 = \pi + \varepsilon/\pi$ . Продолжая аналогично, получим второе приближение  $x_2 = \pi + \varepsilon/\pi + \varepsilon \hat{x}_2$ , подставим его в (1.8) и учитывая, что

$$\sin\left(\varepsilon\left(\frac{1}{\pi} + \varepsilon \hat{x}_2\right)\right) \approx \varepsilon\left(\frac{1}{\pi} + \varepsilon \hat{x}_2\right)$$

из уравнения  $\frac{\varepsilon}{\pi} + \varepsilon^2 \hat{x}_2 + \frac{\varepsilon}{\pi + \frac{1}{\pi} + \varepsilon^2 \hat{x}_2} = 0$ , пренебрегая чле-

нами, содержащими  $\varepsilon$  в степени выше двух, найдем  $\hat{x}_2 = -1/\pi^3$ . В результате двух итераций решение уравнения (1.8) может быть представлено в виде

$$x \approx \pi + \frac{\varepsilon}{\pi} - \frac{\varepsilon^2}{\pi^3} = O(\varepsilon^3), \quad (1.9)$$

где  $O(\varepsilon^3)$  обозначают неучтенные члены бесконечного ряда (1.7), содержащие  $\varepsilon$  в степени три и выше. Этот процесс можно продолжать неограниченно, причем погрешность приближения является малой высшего порядка относительно последнего сохраненного члена. Заметим также, что по (1.9) можно получить хорошее приближение к решению. Например, если в (1.8)  $\varepsilon = 0,1$ , то, вычисляя приближение по (1.7), получим  $\bar{x} \approx 3,173101$ . При этом невязка решения (1.8)  $f(\bar{x}) \approx 1,18 \cdot 10^{-5}$ .

Следует отметить, что приближения  $x_i$  в итерационном процессе вида (1.7) могут быть получены дифференцированием решения  $x$  по  $\varepsilon$ . Например, первое приближение  $x_1$  может быть получено по формуле

$$x_1 = \frac{dx}{d\varepsilon} = \left[ \frac{\partial F(x, \varepsilon)}{\partial x} \right]^{-1} \frac{\partial F(x, \varepsilon)}{\partial \varepsilon} \quad (1.10)$$

при  $x = x_0$  и  $\varepsilon = 0$  и предполагает существование обратной матрицы частных производных  $\left[ \frac{\partial F(x, \varepsilon)}{\partial x} \right]^{-1}$  при  $\varepsilon = 0$ . Если обратная матрица существует, то для системы (1.6) допустимо разложение (1.7) и такая задача называется регулярной. В противном случае она называется сингулярной (вырожденной) и для нее следует применять другие методы, рассматриваемые ниже.

### 1.1.2. Метод асимптотических разложений

В случаях, когда нельзя получить начальное приближение решения  $x_0$ , положив в (1.6)  $\varepsilon = 0$ , или при вырожденности матрицы  $\frac{\partial F(x, \varepsilon)}{\partial x}$  в (1.10), можно получить решение в виде ряда

$$x = \sum_{l=0}^{\infty} \varphi_l(\varepsilon), \quad (1.11)$$

где  $\varphi_i(\varepsilon)$  — некоторые функции, зависящие от малого параметра, которые строятся таким образом, чтобы выполнялось условие

$$\lim_{\varepsilon \rightarrow 0} \frac{\varphi_{i+1}(\varepsilon)}{\varphi_i(\varepsilon)} = 0. \quad (1.12)$$

Ряд (1.11) при этом называется асимптотическим разложением решения  $x$ .

**Пример 2.** В качестве примера рассмотрим решение уравнения

$$\ln \frac{x}{\varepsilon} = kx, \quad k > 0, \quad (1.13)$$

которое не удастся получить обычным методом возмущений, положив  $\varepsilon=0$ . Запишем (1.13) в виде  $kx - \ln x = -\ln \varepsilon$  и заметим, что  $\lim_{x \rightarrow \infty} \frac{\ln x}{kx} = 0$ . Следовательно, можно попробовать пренебречь  $\ln x$  и за начальную базисную функцию принять  $x_0 = \varphi_0(\varepsilon) = -\ln \varepsilon/k$ . Аналогично предыдущему пункту возьмем за следующее приближение к решению

$$x_1 = x_0 + \varphi_1(\varepsilon) = -\frac{\ln \varepsilon}{k} + \varphi_1(\varepsilon)$$

и подставим его в (1.13):

$$k \left( -\frac{\ln \varepsilon}{k} + \varphi_1(\varepsilon) \right) - \ln \left( -\frac{\ln \varepsilon}{k} + \varphi_1(\varepsilon) \right) = -\ln \varepsilon.$$

Пренебрегая в последнем уравнении величиной  $\ln \left( -\frac{1}{k} + \frac{\varphi_1(\varepsilon)}{\ln \varepsilon} \right)$ , получим  $\varphi_1(\varepsilon) = \frac{1}{k} \ln \ln \varepsilon$ . При этом решение (1.13) может быть представлено в виде

$$x = -\frac{1}{k} (-\ln \varepsilon + \ln \ln \varepsilon + \dots). \quad (1.14)$$

Подобное разложение можно продолжить дальше, причем видно, что при  $\varepsilon \rightarrow 0$  выполняется (1.12).

## 1.2. Численные методы

При решении задач анализа статики ММС (1.1) основным инструментом являются численные методы, позволяющие свести решение к выполнению конечного числа арифме-

тических действий над числами. При этом результаты получаются также в виде чисел.

При численном решении СНАУ (1.1) следует учитывать, что не существует математических результатов, позволяющих в общем виде решить вопрос о существовании и числе решений. Поиск решения можно представить как поиск в пространстве  $R^k$  точек, общих для  $k$  поверхностей, каждая из которых задается уравнением системы (1.1):  $f_i(v_1, v_2, \dots, v_k) = 0, i = 1, 2, \dots, k$ .

Другой важной особенностью численного решения системы (1.1) является то, что из-за нелинейности ММС в общем случае нельзя применять прямые методы (например, метод последовательного исключения). Поэтому разработанные методы решения являются итерационными: начиная с начального вектора  $v^0 = \text{col}(v_1^0, v_2^0, \dots, v_k^0)$  последовательно приближаются к решению с помощью итерационной процедуры вида

$$\begin{aligned} v^{r+1} &= H(v^r), \\ r &= 0, 1, \dots \end{aligned} \quad (1.15)$$

Последовательность полученных на каждом шаге векторов  $v^1, v^2, \dots$  должна сходиться к пределу  $v_*$ , который удовлетворяет соотношению

$$v_* = H(v_*). \quad (1.16)$$

Итерационный оператор  $H$  определяется таким образом, чтобы вектор  $v_*$  был решением системы (1.1).

Итерационный процесс (1.15) следует прекратить после того, как текущая точка  $v^r$  окажется в заданной  $\epsilon$ -окрестности стационарной точки  $v^*$ , т. е.

$$\|v^* - v^r\| < \epsilon. \quad (1.17)$$

Однако воспользоваться условием (1.17) нельзя, так как  $v^*$  неизвестно. Поэтому его заменяют условием

$$\|v^{r+1} - v^r\| < \epsilon, \quad (1.18)$$

которое приводит к достижению  $\epsilon$ -окрестности точки, если скорость сходимости достаточно велика, или условием для (1.1)

$$\|F(v^r)\| < \delta, \quad (1.19)$$

где  $\delta$  и  $\epsilon$  — малые положительные числа.

В практических расчетах обычно в (1.17), (1.18) вводятся специальные весовые матрицы масштабирования, учитывающие специфику решаемой задачи [63, 64].

Таким образом, при решении СНАУ возникают следующие проблемы:

- 1) определение оператора  $H(\cdot)$ ;
- 2) определение вектора начальных приближений  $v_0$ ;
- 3) анализ сходимости последовательности  $v^r$ ,  $r = 0, 1, 2, \dots$ ;
- 4) оценка скорости сходимости при численной реализации метода.

Следует также учитывать, что если с помощью какого-либо итерационного метода можно найти одно или несколько решений (для различных значений векторов начальных приближений  $v^0$ ), то из этого не следует, что метод позволяет вычислить все возможные решения.

Кроме того, опыт показывает, что результаты теоретического анализа сходимости итерационных алгоритмов (в частности, нелинейных) практически не отражают их реальную эффективность [63, 65]. Удачный алгоритм в действительности часто сходится лучше, чем может быть предсказано теорией. Примером этого является метод Ньютона, который сходится почти всегда и быстро, хотя доказательства сходимости получены только при дополнительных условиях и при хорошем начальном приближении. Однако для получения более полного представления об итерационном методе, поиска путей ускорения его сходимости и т. д. необходимо изучать их сходимость.

При исследовании итерационного алгоритма общего вида (1.15) фундаментальную роль играет следующая теорема [30, 63, 64, 65].

*Теорема о сжимающем отображении.* Пусть  $H$  — оператор, отображающий область  $D$  в  $D$ , где  $D$  есть замкнутое подмножество в  $R^h$ . Если для некоторой нормы  $\|\cdot\|$  существует  $\alpha \in (0, 1)$ , такое, что

$$\|H(x) - H(y)\| \leq \alpha \|x - y\| \quad (1.20)$$

при  $x, y \in D$ , то:

- 1) существует единственная неподвижная точка  $x_* \in D$ , такая, что  $H(x_*) = x_*$ ;
- 2) для любых  $x_0 \in D$  последовательность  $\{x_r\}$ , удовлетво-

ряющая соотношению  $x_{r+1} = H(x_r)$ ,  $r = 0, 1, \dots$ , сходится  $q$ -линейно к  $x_*$  с константой  $\alpha^*$ ;

3) для любого  $\eta \geq \|H(x_0) - x_0\|$

$$\|x_r - x_*\| \leq \frac{\eta \alpha^r}{1 - \alpha}, \quad (1.21)$$

$$r = 0, 1, \dots$$

Итерационная функция  $H(\cdot)$ , удовлетворяющая (1.20), называется сжимающей в области  $D$ . Из этого свойства следует, что, начиная из любой начальной точки  $x_0 \in D$ , длина шага  $\|x_{r+1} - x_r\|$  уменьшается на каждой итерации по крайней мере на сомножитель  $\alpha$ , так, как

$$\|x_{r+1} - x_r\| = \|H(x_r) - H(x_{r-1})\| \leq \alpha \|x_r - x_{r-1}\|. \quad (1.22)$$

Несмотря на важность этой теоремы, она может быть использована только для установления линейной сходимости. Кроме того, свойство сжатия налагает на оператор  $H(\cdot)$  очень сильные ограничения.

### 1.2.1. Метод простых итераций

Этот метод состоит в следующем: система уравнений (1.1) преобразуется к виду

$$v = G(v) \quad (1.24)$$

и итерации проводятся по формуле

$$v^{r+1} = G(v^r), \quad (1.25)$$

$$r = 0, 1, \dots$$

Если отображение  $y = G(v)$  сжимающее, то по теореме о сжимающем отображении система (1.24) имеет решение  $v_*$  и выполняется (1.22) с  $\eta = \|v^1 - v^0\|$ . При этом метод будет обладать линейной скоростью сходимости.

---

\* Если последовательность  $\{x_r\}$  сходится к  $x_*$  и существует константа  $c \in [0, 1)$  и целое  $\hat{r} \geq 0$ , такие, что для всех  $r \geq \hat{r}$   $|x_{r+1} - x_r| \leq c |x_r - x_*|$ , то говорят, что  $\{x_r\}$  является  $q$ -линейной, сходящейся к  $x_*$ . Аналогично, если  $\{x_r\}$  сходится к  $x_*$  и существуют постоянные  $p > 1$ ,  $c \geq 0$  и  $\hat{r} \geq 0$ , такие, что для всех  $r \geq \hat{r}$   $|x_{r+1} - x_r| \leq c |x_r - x_*|^p$ , то говорят, что  $\{x_r\}$  сходится к  $x_*$  с  $q$ -порядком, по меньшей мере равным  $p$ . Например, если  $p = 2$ , то говорят, что скорость сходимости является квадратичной.

Получить (1.24) из (1.1) можно, например, прибавив слева и справа к (1.1) вектор  $v$  и введя параметр  $h$ :

$$v = v + hF(v). \quad (1.26)$$

В этом случае оператор  $H$  из (1.16) имеет вид

$$H(v) = v + hF(v) \quad (1.27)$$

и итерации можно проводить по формуле

$$v^{r+1} = v^r + hF(v^r), \quad (1.28)$$

которая совпадает с формулой метода Эйлера для систем обыкновенных дифференциальных уравнений (см. подразд. 2.1). Однако, в отличие от анализа задач динамики, здесь представляет интерес только стационарная точка  $v^*$ , а не траектория, ведущая от исходной точки  $v_0$  к окончательной  $v^*$ . Поэтому можно видоизменять параметр  $h$  в (1.26) так, чтобы выполнялись условия сходимости к  $v_*$  и сокращалось число итераций.

Метод простой итерации в случае, если решается одно скалярное уравнение, имеет простую геометрическую интерпретацию [30]. Построим на плоскости  $(v, y)$  графики  $y = g(v)$  и  $y = v$  (рис. II.1). Точки пересечения этих линий соответствуют искомому решению. Если имеется точка  $(v^r, v^{r+1}) = (v^r, g(v^r))$ , то, проведя через нее прямую  $y = v^{r+1}$  до пересечения с прямой  $y = v$ , а затем прямую  $v = v^{r+1}$  до пересечения с кривой  $y = g(v)$ , получим очередное приближение  $(v^{r+1}, v^{r+2})$ . На рис. II.1 представлен метод последовательных приближений в случаях:

- а)  $0 < \partial g(v) / \partial v < 1$ ;
- б)  $-1 < \partial g(v) / \partial v < 0$ ;
- в)  $1 < \partial g(v) / \partial v$ ;
- г)  $\partial g(v) / \partial v < -1$ .

Из теоремы о локальной сходимости следует, что при  $-1 < \frac{\partial g(v)}{\partial v} < 1$  метод сходится (рис. II.1, а и б). Монотонное поведение  $v^r$  при  $\frac{\partial g(v)}{\partial v} > 0$  и колебательное при  $\frac{\partial g(v)}{\partial v} < 0$  непосредственно следуют из соотношения

$$v^{r+1} - v_* = g(v^r) - g(v_*) \sim \frac{\partial g(v)}{\partial v} (v^r - v_*). \quad (1.29)$$

Во многих случаях  $F(v)$  из (1.1) может быть представлена в виде

$$F(v) = A(v) + G(v), \quad (1.30)$$

где  $A$  — невырожденная матрица  $k \times k$ . Тогда итерации можно определить следующим образом [64, 67]:

$$Av^{r+1} + G(v^r) = 0. \quad (1.31)$$

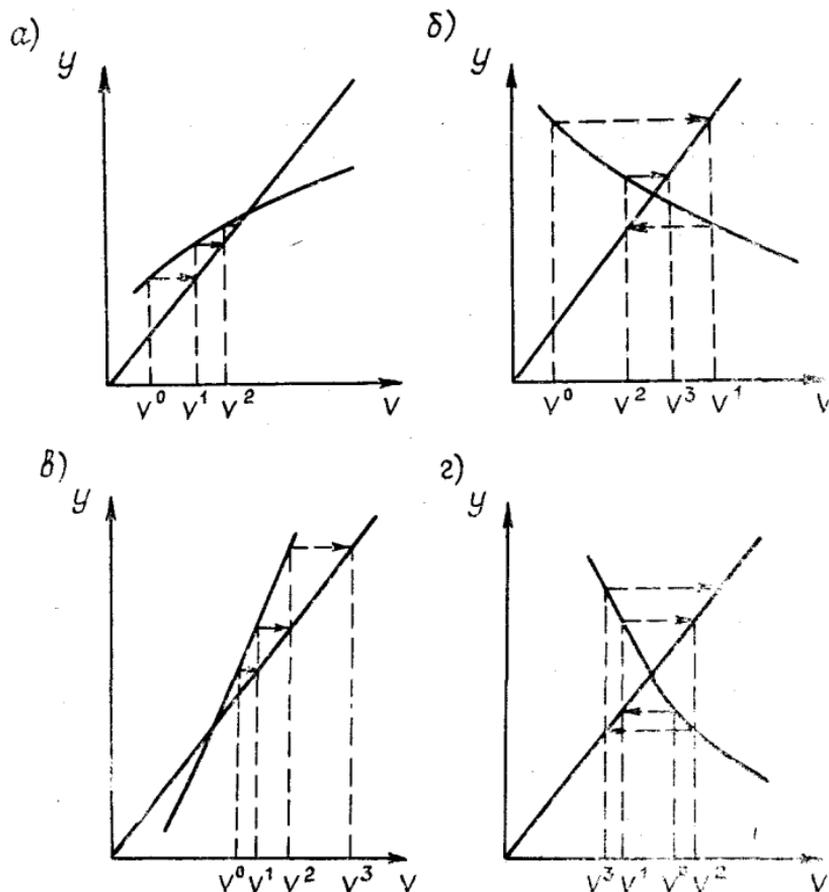


Рис. II. 1

Формула (1.31) определяет итерационный метод Пикара, который в явной форме записывается в виде

$$v^{r+1} = v^r - A^{-1}F(v^r). \quad (1.32)$$

Среди методов, имеющих  $q$ -линейную скорость сходимости, в практических расчетах применяются нелинейные аналоги методов Якоби и Гаусса — Зейделя [64, 65].

Нелинейный метод Якоби для системы (1.1) имеет вид

$$f_i(v_1^r, v_2^r, \dots, v_{i-1}^r, v_{i+1}^r, \dots, v_k^r) = 0, \quad (1.33)$$

$$i = 1, 2, \dots, k.$$

Здесь для отыскания  $x^{r+1}$  необходимо решить  $k$  независимых скалярных уравнений. Этот метод пригоден для параллельных вычислений, так как все  $k$  уравнений (1.33) могут быть решены одновременно на параллельных процессорах.

В отличие от этого нелинейный метод Гаусса — Зейделя состоит в последовательном решении  $k$  скалярных уравнений

$$f_i(v_1^{r+1}, v_2^{r+1}, \dots, v_i^{r+1}, v_{i+1}^r, \dots, v_k^r) = 0 \quad (1.34)$$

относительно переменной  $v_i^{r+1}$ ,  $i = 1, 2, \dots, k$ .

Можно построить нелинейный аналог метода релаксации, который заключается во введении в метод Зейделя параметра релаксации  $\omega$ . Вычисления при этом проводятся по следующей схеме: при известном  $v^r$  выполняется следующий шаг итерации по методу Зейделя, результат которого обозначим через  $y$ ; следующее приближение получают с учетом параметра релаксаций:

$$v^{r+1} = (1 - \omega)v^r + \omega y. \quad (1.35)$$

### 1.2.2. Метод Ньютона и его модификации

Если известно достаточно хорошее начальное приближение  $v^0$  к решению системы (1.1), то эффективным для повышения точности является метод Ньютона, основанный на замене в окрестности  $v^r$  исходной нелинейной задачи (1.1) некоторой вспомогательной линейной задачей.

Предположим, что приближение  $v^r$  уже известно. Разложим в окрестности  $v^r$  вектор-функцию  $F(v)$  в (1.1) в ряд Тейлора:

$$f_i(v_1, v_2, \dots, v_k) = f_i(v_1^r, v_2^r, \dots, v_k^r) + (v_1 - v_1^r) +$$

$$+ \frac{\partial f_i(v^r)}{\partial v_1} + \dots + (v_k - v_k^r) \frac{\partial f_i(v^r)}{\partial v_k} + 0(|v - v^r|^2) \quad (1.36)$$

и отбросим величины второго порядка малости. Тогда система (1.1) заменится системой уравнений:

$$\sum_{j=1}^k (v_j^{r+1} - v_j^r) \frac{\partial f_i(v^r)}{\partial v_j} + f_i(v^r) = 0, \quad (1.37)$$

$$i=1, 2, \dots, k,$$

линейной относительно приращений  $v_j - v_j^r$ ,  $j=1, 2, \dots, k$ . Решение  $v = \text{col}(v_1, \dots, v_k)$  системы (1.37) примем за следующее приближение и обозначим через  $v^{r+1}$ .

Таким образом, итерационный метод Ньютона для (1.1) определяется системой

$$\sum_{j=1}^k (v_j^{r+1} - v_j^r) \frac{\partial f_i(v^r)}{\partial v_j} + f_i(v^r) = 0, \quad i=1, 2, \dots, k, \quad (1.38)$$

из которой последовательно, начиная с заданного  $v^0$ , находятя векторы  $v^r$ ,  $r=1, 2, \dots$ .

Систему (1.38) можно записать в векторной форме:

$$v^{r+1} = v^r - \left[ \frac{\partial F(v^r)}{\partial v} \right]^{-1} F(v^r), \quad (1.39)$$

где  $\frac{\partial F(v^r)}{\partial v}$  — матрица Якоби вектор-функции  $F(\cdot)$ . Для реализации метода Ньютона необходимо существование  $\left[ \frac{\partial F(v^r)}{\partial v} \right]^{-1}$ , обратной к  $\frac{\partial F(v^r)}{\partial v}$ . Для метода Ньютона оператор  $H(\cdot)$  из (1.16) определяется следующим образом:

$$H(v) = v - \left[ \frac{\partial F(v^r)}{\partial v} \right]^{-1} F(v^r). \quad (1.40)$$

Изучение сходимости метода Ньютона требует привлечения сложного математического аппарата. Фундаментальные результаты в этой области получены Л. В. Канторовичем [30, 63, 64]. На практике можно руководствоваться следующим [63, 65]:

1. Стандартный метод Ньютона очень эффективен.
2. Сходимость в начале процесса чаще всего линейная.
3. Начиная с некоторого шага, номер которого трудно предсказать заранее, сходимость резко ускоряется и становится квадратичной.
4. Существуют случаи, когда метод расходится или наблюдается заикливание итераций.
5. Примененный к линейной системе уравнений  $Av - b = 0$  метод сходится за одну итерацию, однако это не дает

никакого выигрыша, поскольку на этом шаге систему  $Ax=b$  все-таки нужно решить.

Рассмотрим геометрическую интерпретацию метода Ньютона в случае решения скалярного уравнения  $f(v)=0$ . Тогда расчетная формула принимает вид

$$v^{r+1} = v^r - f(v^r)/f'(v^r). \quad (1.41)$$

Для получения  $v^{r+1}$  геометрически надо найти абсциссу точки пересечения с осью  $v$  касательной к кривой  $y=f(v)$  в точке  $(v^r, f(v^r))$  (рис. II.2). Уже в случае, когда  $f(v)$  — многочлен третьей степени, может случиться, что последова-

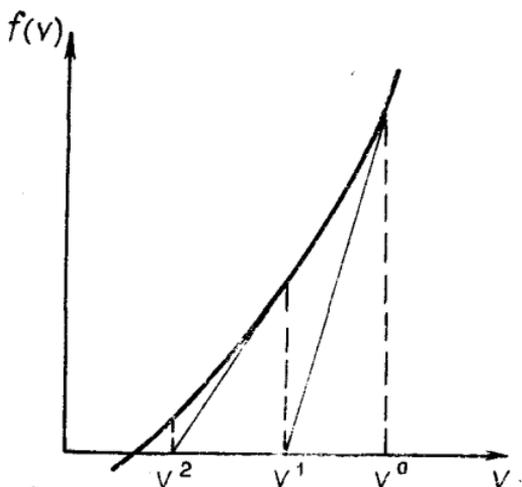


Рис. II.2

тельность  $\{v^r\}$  не сходится к корню при плохом начальном приближении. Например, в случае, изображенном на рис. II.3, все четные приближения совпадают с  $a$ , а нечетные — с  $b$ . При этом говорят, что метод «зацикливается». Для более сложных задач реальное поведение приближений  $v^r$  при плохом начальном приближении становится более запутанным и трудно поддающимся анализу [30].

Сравним скорость сходимости методов Ньютона и простой итерации. Для последнего справедлива оценка погрешности

$$\|v^{r+1} - v_*\| \leq \alpha^r \|v_0 - v_*\|, \quad \alpha < 1. \quad (1.42)$$

Чтобы погрешность стала меньше  $\epsilon$ , согласно этой оценке достаточно взять

$$r \geq \log_{\alpha^{-1}} \frac{\|v^0 - v_*\|}{\varepsilon} \sim \log_{\alpha^{-1}} \frac{1}{\varepsilon}. \quad (1.43)$$

Для метода Ньютона справедлива оценка погрешности [66]

$$\|v^r - v_*\| \leq c^{-1} (c \|v^0 - v_*\|)^{2^r}. \quad (1.44)$$

Правая часть (1.44) будет меньше  $\varepsilon$ , если

$$r \geq -\log_2 \frac{\log_2 (c \|v^0 - v_*\|)}{\log_2 (c \varepsilon)} \sim \log_2 \log_2 \frac{1}{\varepsilon}. \quad (1.45)$$

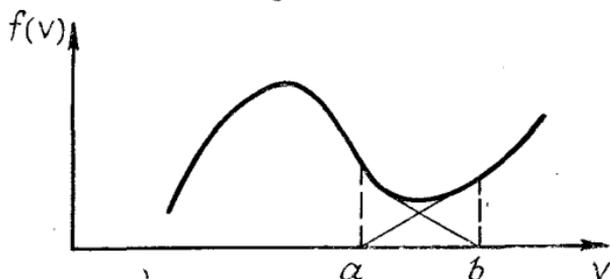


Рис. II. 3

Таким образом, асимптотически при  $\varepsilon \rightarrow 0$  метод Ньютона требует меньшего числа итераций.

Следует отметить, что, записанный в форме (1.39), он является модификацией метода простой итерации. В случае скалярного уравнения производная правой части (1.41)  $g(v) = v - f(v)/f'(v)$  по  $v$  равна:  $f(v)f''(v)/f'(v)^2$ . Таким образом,  $g'(v) = 0$ , если  $f'(v) \neq 0$  и рис. II. 1 приобретает вид рис. II. 4.

К преимуществам метода Ньютона относятся практическая эффективность и квадратичная сходимость вблизи решения. Основной недостаток состоит в повторных вычислениях на каждом шаге якобиана  $\frac{\partial F(v^r)}{\partial v}$ . Для этого нужно вычислить  $k$  функций  $f_i(\cdot)$  в (1.1) и  $k'$  функций  $\frac{\partial f_i(\cdot)}{\partial v_j}$ , т. е. всего  $k^2 + k$  раз. Так как при больших порядках  $k$  СНАУ или сложных вектор-функций получение аналитических выражений якобиана затруднено, то обычно он аппроксимируется конечными разностями:

$$\frac{\partial f_i}{\partial v_j}(v) \cong \frac{1}{h} [f_i(v_1, v_2, \dots, v_{j-1}, v_j + h, v_{j+1}, \dots, v_k) - f_i(v)]. \quad (1.46)$$

При этом требуются выражения только для самих функций  $f_i(\cdot)$ , которые нужны в любом случае. Применение конечных разностей обеспечивает некоторое «сглаживание» очередного приближения, однако при этом скорость сходимости немного уменьшается. К другим недостаткам следует отнести необходимость решать на каждом шаге систему линейных алгебраических уравнений и возможность того, что при за-

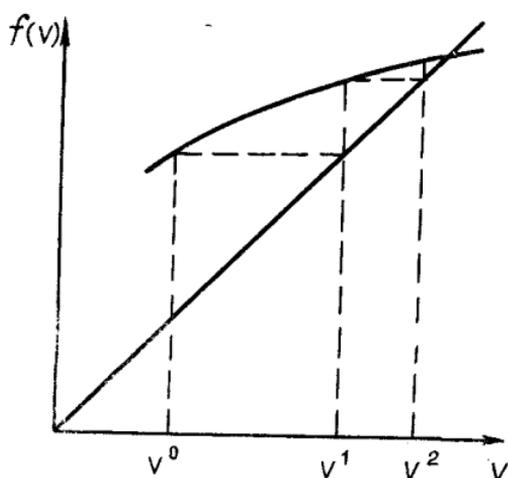


Рис. II. 4

данном начальном приближении итерации могут расходиться (так как сходимость гарантируется, если  $v^0$  достаточно близка к  $v^*$ ).

На практике обычно используются различные модификации метода Ньютона [63, 64, 68]. Например, для сокращения арифметической работы матрица Якоби вычисляется не на каждом шаге, а только при ее существенном изменении, которое контролируется, например, числом итераций, необходимым для уменьшения ошибки до заданной величины. В случаях, когда правые части системы (1.1) имеют сложный вид и их многократное вычисление обходится дорого, применяют различные способы аппроксимации якобиана, требующие для пересчета аппроксимации на каждой итерации одного-двух вычислений вектор-функции  $F(v)$  (например, как в методе секущих) [30, 63, 64]. Наконец, если заранее известны нулевые элементы якобиана (некоторые функции  $f_i(v)$  в (1.1) не зависят от переменных  $v_j$ ,  $j=1, 2, \dots, k$ ), то можно сокра-

тить количество вычислений правых частей в (1.1), вычисляя только ненулевые элементы — матрицы Якоби и применяя методы решения СЛАУ с разреженными матрицами [69]. Имеются разреженные аналоги метода секущих [63].

В настоящее время большое распространение получили гибридные методы [64]. Например, в методах Якоби и Зейделя (п. 1.2.1) имеется двойной итерационный процесс: внутри каждого внешнего шага основной итерации решается скалярное нелинейное уравнение с помощью итерационного метода Ньютона. На практике часто урезают внутренние итерации до одной-трех. Такие методы хорошо распараллеливаются и могут быть очень эффективны для задач определенной структуры.

Средством получения хорошего начального приближения  $v_0$  является его выбор из физических соображений или использование вспомогательной процедуры поиска начального приближения, например, методом продолжения по параметру, описанным ниже.

### 1.2.3. Метод продолжения по параметру

Этот метод представляет собой алгоритм, позволяющий с помощью итераций получить решение СНАУ (1.1). Его эффективность не зависит от «удачного» выбора начального приближения. Суть метода состоит в том, что наряду с исходной системой (1.1) рассматривается другая:

$$G(v) = 0, \quad (1.47)$$

решение которой известно. Затем, «деформируя» уравнение (1.47), превратим его в (1.1) с помощью конечного числа  $N$  последовательных малых приращений параметров:

$$G^r(v) = G^{r-1}(v) + [F(v) - G^{r-1}(v)] \frac{1}{N}. \quad (1.48)$$

Решение  $v^0$  исходной системы уравнений  $G^0(v)$  можно использовать как исходные значения переменных для итерационного решения системы  $G^1(v)$ . Так как она мало отличается от предыдущей, весьма вероятно, что сходимость будет обеспечена. В процессе счета решение  $v^{r-1}$  используется как исходное для получения решения  $v^r$ . В конце счета, когда  $r$  становится равным  $N$ , решаемая система уравнений становится эквивалентна исходной. Так как для превращения (1.47) в (1.48) может потребоваться значительное количество шагов, то применение метода может быть связано

с большими затратами машинного времени. Однако положительную роль играет то обстоятельство, что при малой величине шагов сходимость на отдельном шаге может быть достигнута всего несколькими итерациями. Известно [70], что этот метод удобно применять при синтезе механических систем, поскольку (1.47) можно получить, используя любой механизм рассматриваемого типа.

Если исходное уравнение (1.1) не содержит параметров, позволяющих получить систему (1.47), то его можно ввести искусственно, например, следующим образом [67, 68]:

$$\hat{F}(v, \alpha) = \alpha F(v) - (\alpha - 1)F(v_0), \quad (1.49)$$

где  $\alpha \in [0, 1)$  — параметр,  $v_0$  — некоторое начальное приближение. Ясно, что при  $\alpha = 0$  (1.49) преобразуется к виду (1.47), а при  $\alpha = 1$  — к исходной системе (1.1).

В реальных технических системах, как правило, известен номинальный режим работы и все исследования статического режима производятся при изменении вектора входов  $u$  от номинального режима  $u_n$  до порогового  $u_*$ . Поэтому в качестве системы с известным решением можно взять исходную при номинальном режиме работы, а затем менять вектор входов небольшими скачками до  $u_*$ , вычисляя после каждого скачка некоторое промежуточное решение (1.1) и принимая его за исходную точку для итераций при следующем уровне вектора входов.

### 1.3. Особенности численного решения систем линейных алгебраических уравнений

В задачах анализа статики линейных ММС, а также при решении систем нелинейных алгебраических уравнений требуется решить систему линейных алгебраических уравнений вида

$$Ax = b, \quad (1.50)$$

где  $A \in R^{n \times n}$  — квадратная невырожденная матрица,  $x \in R^n$  — вектор неизвестных,  $b \in R^n$  — заданный вектор.

Обычно для решения СЛАУ (1.50) доступны специальные библиотеки подпрограмм [63, 71]. Поэтому, как правило, нет необходимости в разработке собственной программы. Важнее знать, какую из имеющихся программ использовать для конкретного вида СЛАУ, каковы вычислительные затраты и возможные проблемы их использования. Кроме этого,

необходимо иметь базовые знания по структуре этих программ, чтобы представлять, какие из технических приемов могут быть полезны в других задачах анализа.

Методы численного решения СЛАУ делятся на две группы: прямые и итерационные. В прямых методах решение находится за конечное число арифметических действий, в итерационных — по рекуррентным соотношениям, сходящимся к решению. Ввиду того, что итерационные методы применяются, как правило, для симметричных матриц и матриц специальной структуры, ниже будут рассмотрены только прямые методы.

Прямые методы основаны на сведении матрицы СЛАУ (1.50) к матрице простой структуры (треугольной, диагональной и др.), удобной для решения (1.50), и обычно основаны на разложении  $A$  вида

$$A = M_1 \cdot M_2 \cdot \dots \cdot M_l, \quad (1.51)$$

где  $M_i \in R^{n \times n}$ ,  $i = 1, 2, \dots, l$  — невырожденные матрицы. При разложении, как правило, выясняется вопрос и о разрешимости СЛАУ. Если система разрешима, то решение  $x$  может быть получено в такой последовательности:  $M_1 b_1 = b$ ,  $M_2 b_2 = b_1, \dots, M_l b_l = b_{l-1}$ , что соответствует выражению

$$x = M_l^{-1} \cdot M_{l-1}^{-1} \cdot \dots \cdot M_2^{-1} \cdot M_1^{-1} b. \quad (1.52)$$

Таким образом, решение СЛАУ проводится в два этапа. Сначала выполняется разложение матрицы  $A$  (например, как в (1.51)), затем для этого разложения находится решение (1.50). Поэтому большинство надежных вычислительных алгоритмов решения СЛАУ состоят из двух отдельных программ, реализующих разложение и нахождение решения для разложенной матрицы. Такая организация вычислительной схемы позволяет существенно сэкономить затраты машинного времени в задачах, когда требуется многократно решать СЛАУ (1.50) с одной и той же матрицей  $A$  и различными векторами правых частей. При этом разложение матрицы  $A$  проводится один раз, а затем многократно вычисляется решение для различных векторов  $b$ . Экономия достигается за счет того, что арифметические затраты первого этапа, как правило, на порядок больше, чем второго.

В табл. II.1 приведена арифметическая трудоемкость (главные члены) наиболее применяемых алгоритмов разложения матрицы  $A$ , а в табл. II.2 — арифметические затраты

Таблица II.1

Метод	Асимптотическое число операций	Количество вычислений квадратных корней
Гаусса	$\frac{1}{3} n^3$	—
Холецкого (только для симметрических матриц)	$\frac{1}{6} n^3$	$n$
Хаусхолдера	$\frac{2}{3} n^3$	$n$
Гивенса	$\frac{4}{3} n^3$	$n^2$
Гивенса-Джентельмена	$\frac{2}{3} n^3$	—

Таблица II.2

Тип матрицы	Асимптотическое число операций
Ортогональная	$2n^2$
Диагональная	$n$
Блочно-диагональная с размерами блоков $1 \times 1$ или $2 \times 2$	$\frac{7}{2} n$
Верхняя или нижняя треугольная	$n^2$

для получения решения СЛАУ в зависимости от типа матрицы  $M_i$  в разложении (1.51). Под арифметическими операциями понимается операция «умножить» или «разделить» или «сложить»,  $n$  — размерность СЛАУ (1.50).

Как видно из табл. II.1, наиболее эффективным является метод Гаусса, требующий примерно вдвое меньше арифметических операций, чем в методах Хаусхолдера и Гивенса-Джентельмена, и в четыре раза меньше, чем в обычном методе Гивенса. При решении СЛАУ с симметричными матрицами лучше применять метод Холецкого. Однако для метода Гаусса характерна неустойчивость, связанная с ростом промежуточных элементов матриц  $M_i$ . Этого недостатка лишены методы Хаусхолдера и Гивенса, которые являются бо-

лее устойчивыми. На практике при использовании метода Гаусса сильный рост элементов наблюдается крайне редко и обычно легко контролируется [72]. Вычислительная работа в методах Хаусхолдера и Гивенса-Джентельмена примерно одинакова и нет причин для предпочтения второго метода первому.

Подробнее методы решения СЛАУ рассмотрены в [63, 69, 72—76].

## 2. АНАЛИЗ ДИНАМИКИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СИСТЕМЫ

Основной целью анализа динамики ММС (точнее, расчета ее во временной области) является определение выхода  $y(t)$  и состояния  $x(t)$  ММС по заданному вектору входных воздействий  $u(t)$ . При этом предполагается известным временной интервал исследования ММС  $t \in [t_0, T]$  и вектор начальных значений состояния  $x(t_0)$ , который может быть получен экспериментально из физических соображений или в результате статического анализа. Таким образом, анализ динамики ММС сводится к нахождению решения системы алгебро-дифференциальных (II.3), (II.4) или дифференциальных (II.1), (II.2) уравнений. Методы их решения могут быть разделены на две группы: аналитические и численные.

### 2. 1. Аналитические методы

#### 2.1.1. Метод возмущений

**Пример 1.** Рассмотрим динамику популяции двух видов, взаимодействующих между собой по типу хищник-жертва [67]. Построим математическую модель системы, предполагая, что жертва может найти достаточно пищи для пропитания, но при каждой встрече с хищником последний убивает жертву.

Обозначим соответственно через  $x(t)$  и  $y(t)$  количество жертв и хищников в момент времени  $t$ . Предположим, что норма рождаемости жертв  $x_b$  и норма естественной смертности  $x_d$  являются константами и  $x_b > x_d$ . Следовательно, в отсутствие хищников популяция жертв будет расти со скоростью  $(x_b - x_d)x(t)$ . Будем считать, что число случаев, когда хищник убивает жертву, зависит от вероятности их

встречи и пропорционально произведению  $x(t) \cdot y(t)$ . Таким образом, популяция жертв может быть описана дифференциальным уравнением

$$\dot{x}(t) = \alpha x(t) + \beta x(t)y(t), \quad (2.1)$$

где  $\alpha = x_b - x_d > 0$ , а  $\beta < 0$ .

Чтобы вывести уравнение, описывающее популяцию хищников, предположим, что при отсутствии жертв число хищников по естественным причинам убывает пропорционально  $y(t)$ . В то же время в результате встреч с жертвами число хищников увеличивается, что ведет к уравнению

$$\dot{y}(t) = \gamma y(t) + \delta x(t)y(t), \quad (2.2)$$

где  $\gamma < 0$  и  $\delta > 0$ .

Таким образом, мы получили ММС (2.1), (2.2) типа хищник-жертва. Эти уравнения впервые получены в 1925 г. и известны как уравнения Лотки-Вольтерра. Для полной постановки задачи анализа динамики ММС необходимо задать рассматриваемый промежуток  $t \in [t_0, T]$  и количество жертв  $x(t_0) = x_0$  и хищников  $y(t_0) = y_0$  в начальный момент времени  $t_0$ . Поскольку аналитическое представление нетривиальных решений задачи (2.1), (2.2) затруднительно, попытаемся использовать приближенный метод, основанный на теории возмущений.

При этом подходе исходная непрерывная задача заменяется близкой к ней, но более простой непрерывной задачей, решение которой может быть найдено аналитически.

Первый шаг состоит в определении стационарных точек  $(x_s, y_s)$ , называемых также точками равновесия. В данном случае стационарные состояния определяются формулами

$$x = x_s = -\gamma/\delta, \quad y = y_s = -\alpha/\beta, \quad (2.3)$$

так как

$$x'|_{(x_s, y_s)} = x_s (\alpha + \beta y_s) = 0, \quad y'|_{(x_s, y_s)} = y_s (\gamma + \delta x_s) = 0. \quad (2.4)$$

Разлагая правые части в ряд Тейлора в окрестности точки  $(x_s, y_s)$ , получим

$$\begin{aligned} x(\alpha + \beta y) &= \beta x_s (y - y_s) + \dots; \\ y(\gamma + \delta x) &= \delta y_s (x - x_s) + \dots \end{aligned} \quad (2.5)$$

Следовательно, в окрестности этой точки можем аппроксимировать исходные уравнения линейными:

$$\dot{x} = \beta x_s (y - y_s), \quad \dot{y} = \delta y_s (x - x_s). \quad (2.6)$$

Можно показать, что их решения удовлетворяют соотношению

$$\frac{(x - x_s)^2}{\beta x_s} + \frac{(y - y_s)^2}{\delta y_s} = C, \quad (2.7)$$

где  $C$  — некоторая постоянная, определяемая начальными условиями.

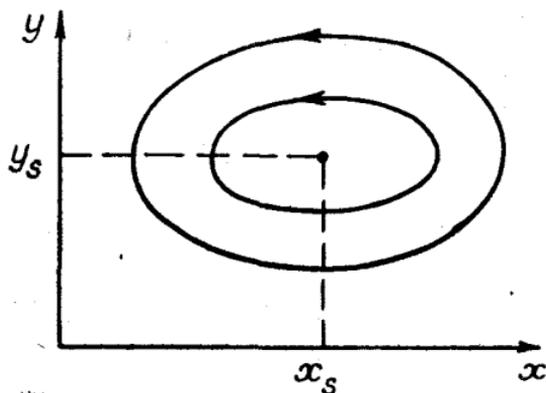


Рис. II. 5

Данное соотношение является уравнением эллипса с центром в точке  $(x_s, y_s)$ , причем различные начальные условия  $x_0, y_0$  задают различные эллипсы. На рис. II. 5 показано семейство подобных эллипсов с центром в точке  $(x_s, y_s)$ , причем стрелками указываются направления, соответствующие возрастанию времени.

Такой анализ на основе теории возмущений может дать полезную информацию о поведении ММС (2.1), (2.2) в окрестности стационарной точки. Поскольку линейризованные уравнения (2.6) аппроксимируют исходные (2.1), (2.2), можно ожидать, что (2.1), (2.2) будут близки к эллипсам. Это подтверждают результаты вычислительных экспериментов, полученные с помощью численных методов.

### 2.1.2. Метод асимптотических разложений

**Пример 2.** Рассмотрим задачу нахождения решения ММС, описываемой дифференциальным уравнением [78]

$$\ddot{x}(t) + x(t) = a \frac{x(t)}{t} \quad (2.8)$$

для больших  $t$  ( $t \gg 1$ ).

Считая известной правую часть (2.8), воспользуемся известной из курса дифференциальных уравнений формулой Коши:

$$x(t) = A \cos(t - a) + a \int_{t_0}^t \sin(t - \tau) \frac{x(\tau)}{\tau^2} d\tau. \quad (2.9)$$

Известно, что интеграл в (2.9) сходится при  $t \rightarrow \infty$  и, следовательно, можно принять в (2.9)  $t = \infty$ .

Аналогично подразд. 1.2 будем искать приближенное решение (2.9) в виде (1.11). В качестве начального приближения примем

$$x(t) = A \cos(t - a) + \varepsilon_1(t). \quad (2.10)$$

Подставляя (2.10) в (2.9), найдем

$$\begin{aligned} \varepsilon_1(t) &= aA \int_{\infty}^t \sin(t - \tau) \cos(t - a) \frac{dt}{t^2} + O\left(\frac{1}{t}\right) = \\ &= -\frac{aA}{2t} \sin(t - a) + O\left(\frac{1}{t}\right). \end{aligned} \quad (2.11)$$

Получаем второе приближение:

$$x(t) = A \cos(t - a) - \frac{aA}{2t} \sin(t - a) + O\left(\frac{1}{t}\right) + \varepsilon_2(t). \quad (2.12)$$

Подставляя (2.12) в (2.9), находим

$$\begin{aligned} \varepsilon_2(t) &= \frac{aA}{2} \int_{\infty}^t \sin(t - 2\tau + a) \frac{dt}{t^2} - \frac{a^2 A}{2} \int_{\infty}^t \sin(t - \tau) \sin(t - a) \times \\ &\times \frac{dt}{t^2} + O\left(\frac{1}{t^2}\right) = \frac{aA}{4t^2} \cos(t - a) - \frac{a^2 A}{8t^2} \cos(t - a) + O\left(\frac{1}{t^2}\right). \end{aligned} \quad (2.13)$$

Первый интеграл в (2.13) преобразуется интегрированием по частям, второй — по формуле тригонометрии, остающиеся интегралы входят в величину  $O\left(\frac{1}{t^2}\right)$ . Таким образом, имеем третье приближение:

$$x(t) = A \cos(t - a) - \frac{aA}{2t} \sin(t - a) + \frac{aA}{4t^2} \left(1 - \frac{a}{2}\right) \cos(t - a) + O\left(\frac{1}{t^3}\right). \quad (2.14)$$

Приближение можно уточнять неограниченно. Заметим, что в (1.12)

$$\varphi_{i+1}(t) = \frac{\cos\left(x - a + n \frac{\pi}{2}\right)}{t^n}, \quad \varphi_i(t) = \frac{1}{t^n}. \quad (2.15)$$

## 2.2. Численные методы

Хотя решение некоторых задач анализа динамики может быть найдено аналитически, в большинстве практических случаев такой путь оказывается невозможным. Поэтому основным средством исследования являются численные методы.

В зависимости от числа независимых переменных и, следовательно, типа входящих в них производных дифференциальные уравнения, описывающие ММС, делятся на две существенно различные категории: обыкновенные, содержащие одну независимую переменную, и уравнения в частных производных, содержащие несколько независимых переменных. Здесь рассматриваются методы решения обыкновенных дифференциальных уравнений.

Чтобы решить обыкновенное дифференциальное уравнение, необходимо знать значения зависимой переменной и (или) ее производных при некоторых значениях независимой переменной. Если эти дополнительные условия задаются при одном значении независимой переменной, то такая задача называется задачей с начальными условиями или задачей Коши. Если же условия задаются при двух и более значениях независимой переменной, то задача называется краевой. В задаче Коши дополнительные условия называются начальными, а в краевой — граничными.

Несмотря на внешнее сходство этих задач, для их решения применяются существенно различные методы и вычислительные алгоритмы. Ниже будет рассматриваться только

задача Коши, которую можно сформулировать следующим образом. Пусть дано дифференциальное уравнение\*:

$$\dot{x} = f(x, t) \quad (2.16)$$

с начальным условием  $x(t_0) = x_0$  на интервале  $t_0 \leq t \leq T$ . Требуется найти функцию  $x(t)$ , удовлетворяющую как уравнению (2.16), так и начальному условию. Обычно численное решение этой задачи получают, вычисляя сначала значение производной при фиксированном значении  $t_k$ , а затем задавая малое приращение  $h$  по независимой переменной  $t$  и переходя к новой точке  $t_{k+1} = t_k + h$ .

Положение новой точки определяется по наклону кривой, вычисленной с помощью дифференциального уравнения. Таким образом, график численного решения выглядит как последовательность коротких кривых, которыми аппроксимируется истинная кривая, а решение представляет собой таблицу с шагом  $h$  (который в общем случае может быть непостоянным) для  $t_0 \leq t \leq T$ .

При этом численный метод определяет порядок действий при переходе от текущей точки  $t_k$  к следующей  $t_{k+1}$ .

подавляющее большинство используемых в настоящее время методов численного интегрирования могут быть заданы формулой [67]\*\*:

$$x_{k+1} = \sum_{i=1}^m \alpha_i x_{k+1-i} +$$

$$+ h \varphi(f; t_{k+1}, t_k, \dots, t_{k+1-m}, x_{k+1}, x_k, \dots, x_{k+1-m}). \quad (2.17)$$

Если для нахождения следующей точки  $x_{k+1}$  требуется информация лишь об одном предыдущем шаге ( $m=1$ ), то метод называется одношаговым. Примером являются методы семейства Рунге — Кутты. В случае, когда для отыскания следующей точки требуется информация более чем об одной из предыдущих точек, метод называется многошаговым. К многошаговым относятся методы Милна, Хемминга, Адамса и др.

Если функция  $\varphi(\cdot)$  в (2.17) не зависит от текущего значения независимой переменной  $t_{k+1}$  (соответственно не содержит  $x_{k+1}$ ), то метод называется явным, в противном слу-

\* Практически все рассматриваемые здесь методы распространяются на системы обыкновенных дифференциальных уравнений. Поэтому для простоты везде (если не будет оговорено специально) будет рассматриваться задача Коши с одним дифференциальным уравнением.

\*\* Ниже для компактности  $x(t_k)$  будет обозначаться через  $x_k$ .

чае — неявным и для нахождения решения в точке  $t_{k+1}$  требуется решать систему нелинейных уравнений относительно  $x_{k+1}$ .

Особую группу представляют линейные многошаговые методы, которые получаются из (2.17), если в качестве функции  $\varphi(\cdot)$  взять

$$\varphi(\cdot) = \sum_{i=0}^m \beta_i f(t_{k+1-i}, x_{k+1-i}). \quad (2.18)$$

Они образуют большой класс методов, называемых методами прогноза и коррекции.

Прежде чем перейти к рассмотрению конкретных методов численного интегрирования, остановимся на источниках погрешностей [79].

1. Может оказаться, что начальное условие  $x(t_0)$  в (2.17) известно неточно и определяется в результате эксперимента, например, с помощью измерений или в результате решения какой-либо другой задачи. В этом случае вместо точного начального условия  $x(t_0)$  приходится использовать его приближенное значение  $x_0$ , а вместо задачи Коши (2.16) решать задачу

$$\tilde{x}' = f(t, \tilde{x}), \quad \tilde{x}(t_0) = \tilde{x}_0 \quad (2.19)$$

с измененными начальными условиями

$$x_0 - \tilde{x}_0 = R_0 \neq 0. \quad (2.20)$$

Решение задачи (2.19) зависит от  $\tilde{x}_0$  и не совпадает с искомым решением  $x(t)$  (2.16). Разность

$$\xi_k = x(t_k) - \tilde{x}(t_k) \quad (2.21)$$

называется неустранимой погрешностью решения  $\tilde{x}(t)$ .

2. Погрешность округления обусловлена ограничениями на представление чисел в ЭВМ и, как следствие, погрешностью вычислений правой части (2.16) и по формуле (2.17). Фактически найденные значения  $\tilde{x}_{k+1}$  удовлетворяют не (2.17), а условию

$$\sum_{i=1}^m \alpha_i \tilde{x}_{k+1-i} +$$

$$h\varphi(f; t_{k+1}, \dots, t_{k+1-m}, \tilde{x}_{k+1}, \dots, \tilde{x}_{k+1-m}) - \\ - \tilde{x}_{k+1} = \delta_{k+1}. \quad (2.22)$$

Невязка  $\delta_k$  называется погрешностью округления на  $k$ -м шаге.

3. Погрешность метода (или погрешность усечения) связана с тем, что при аппроксимации функции в правой части (2.16) вместо бесконечных рядов часто используются лишь несколько первых членов. Эта погрешность целиком обусловлена применяемым методом (2.17) и не зависит от характеристик ЭВМ. Она может быть определена как разность между истинным значением решения  $x(tk)$  задачи (2.19) и его приближенным значением  $x_k$ , полученным по формуле (2.17):

$$\varepsilon_k = \tilde{x}(t_k) - x_k. \quad (2.23)$$

Разность между точным решением  $x(t_k)$  задачи (2.16) и приближенным фактически найденным решением  $\tilde{x}_k$

$$R_k = x(t_k) - \tilde{x}_k \quad (2.24)$$

называется полной погрешностью приближенного решения, величина

$$\eta_k = x_k - \tilde{x}_k \quad (2.25)$$

— вычислительной погрешностью.

Из соотношений (2.21), (2.23), (2.24) и (2.25) следует, что

$$R_k = \xi_k + \varepsilon_k + \eta_k, \quad (2.26)$$

т. е. полная погрешность приближенного решения равна сумме неустранимой погрешности, погрешности метода и вычислительной погрешности.

Указанные источники погрешностей являются причиной наблюдаемых ошибок двух типов:

1) локальной ошибки — суммы погрешностей, вносимых в вычислительный процесс на каждом шаге;

2) глобальной ошибки — разности между вычисленным и точным решением на каждом этапе реализации численного алгоритма, определяющей суммарную погрешность, накопившуюся с момента начала вычислений.

## 2.2.1 Одношаговые методы решения задачи Коши

Пусть требуется найти решение задачи (2.16) на отрезке  $[t_0, T]$ . Возьмем разбиение отрезка точками  $t_0 < t_1 < t_2 < \dots < t_N = T$ . Этот набор точек называется сеткой, а точки  $t_k, k=0, 1, \dots, N$  — узлами сетки. Рассмотрим методы вида (2.17), которые последовательно дают приближения  $x_{k+1}$  к значениям точного решения  $x(t_{k+1})$  в каждом узле сетки  $t_{k+1}$  на основе известного приближения  $x_k$  к решению в предыдущем узле  $t_k$ . В общем виде их можно записать так:

$$x_{k+1} = \varphi(f, t_{k+1}, t_k, x_{k+1}, x_k). \quad (2.27)$$

Простейшим одношаговым методом является метод Эйлера. Он основан на разложении решения  $x(t)$  в окрестности точки  $t_k$  в ряд Тейлора. Если предположить, что правая часть  $f(t, x)$  дифференциального уравнения (2.16) имеет непрерывные частные производные до порядка  $s$ , то искомое решение  $x(t)$  имеет непрерывные производные до  $(s+1)$ -го порядка включительно. При этом точное решение в узле  $t_{k+1}$  имеет вид

$$x_{k+1} = x_k + h x_k' + \frac{h^2}{2!} x_k'' + \dots + \frac{h^s}{s!} x_k^{(s)} + \frac{h^{s+1}}{(s+1)!} x^{(s+1)}(\xi), \quad (2.28)$$

где  $x_k^{(i)} = x^{(i)}(t_k)$ ,  $h = t_{k+1} - t_k$ ,  $t_k < \xi < t_{k+1}$ . В предположении, что  $h$  мало, можно пренебречь в (2.28) членами, содержащими  $h$  во второй или более высоких степенях. Тогда явный метод Эйлера может быть определен следующим образом:

$$x_{k+1} = x_k + h x_k' = x_k + h f(t_k, x_k), \quad (2.29)$$

$$x_0 = x^0, \quad k=0, 1, \dots, N.$$

Таким образом, можно получить приближенное значение зависимой переменной при малом смещении  $h$  от текущей точки. Ошибка метода имеет порядок  $O(h^2)$ , так как члены, содержащие  $h$  во второй и более высоких степенях, отбрасываются. Геометрический смысл метода Эйлера заключается в аппроксимации решения на отрезке  $[t_k, t_{k+1}]$  касательной, проведенной к графику в точке  $t_k$  (рис. II. 6).

Следует отметить, что разложение  $x(t)$  в ряд Тейлора аналогично (2.28) можно выполнить и в окрестности точки  $x(t_{k+1})$ :

$$x(t_{k+1} - h) = x_{k+1} - hx'_{k+1} + \frac{h^2}{2} x''_{k+1} + \dots +$$

$$+ (-1)^s \frac{h^s}{s!} x^{(s)}_{k+1} + (-1)^{s+1} \frac{h^{s+1}}{(s+1)!} x^{(s+1)}(\xi), \quad (2.30)$$

где  $x_{k+1}^{(i)} = x^{(i)}(t_{k+1})$ ,  $h = t_{k+1} - t_k$ ,  $x(t_{k+1} - h) < \xi < x_{k+1}$ . Методы, полученные по этой формуле, образуют семейство неявных одношаговых методов. Простейшим из них является неявный метод Эйлера, получаемый из (2.30) при пренебрежении членами, содержащими  $h$  во второй и более высоких степенях:

$$x_{k+1} = x_k + hx'_{k+1} = x_k + hf(t_{k+1}, x_{k+1}),$$

$$x_0 = x^0, \quad k = 0, 1, \dots, N. \quad (2.31)$$

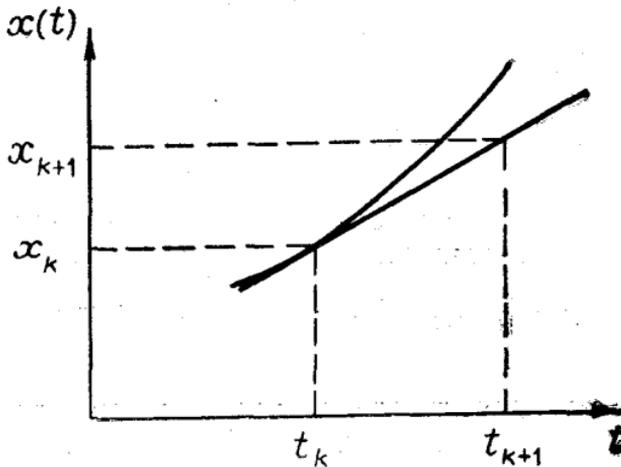


Рис. 1.6

Метод имеет ошибку порядка  $O(h^2)$ . В общем случае для получения решения  $x_{k+1}$  в узле  $t_{k+1}$  следует решать систему нелинейных алгебраических уравнений (2.31) относительно  $x_{k+1}$  итерационным методом (см. подразд. 1.2).

Если удерживать в разложении (2.28) или (2.30) с первых членов, то можно получить одношаговый метод  $s$ -го порядка с ошибкой, имеющей порядок  $O(h^{s+1})$ . Однако с увеличением порядка  $s$  вычисление производных в (2.28) и (2.30) становится более громоздким, что требует большего количества вычислений правых частей  $f(t, x(t))$  в (2.16) и является существенным недостатком метода.

Немецкий математик XIX в. Рунге предложил следующую идею, основанную на вычислении приближенного решения  $x_{k+1}$  в узле  $t_k + h$  в виде линейной комбинации с постоянными коэффициентами [80]:

$$x_{k+1} = x_k + h \sum_{i=1}^s b_i k_i, \quad (2.32)$$

где

$$k_i = f(t_k + c_i h, x_k + h \sum_{j=1}^s a_{ij} k_j), \quad (2.33)$$

$$1 \leq i \leq s.$$

Числа  $a_{ij}$ ,  $b_i$ ,  $c_i$  выбираются так, чтобы разложение выражения (2.32) по степеням  $h$  совпадало с разложением (2.28) или (2.30) до максимально возможной степени при произвольной правой части  $f(t, x)$  и произвольном шаге  $h$ .

Это эквивалентно следующему. Если ввести вспомогательную функцию

$$g(h) = x(t_k + h) - x(t_k) - h \sum_{i=1}^s b_i k_i(h), \quad (2.34)$$

то ее разложение по степеням  $h$  должно начинаться с максимально возможной степени:

$$g(h) = \frac{h^{s+1}}{(s+1)!} \varphi^{(s+1)}(0) + O(h^{s+1}). \quad (2.35)$$

Если можно определить эти постоянные так, чтобы разложение  $g(h)$  имело вид (2.28) или (2.30), то говорят, что формулы (2.32), (2.33) с выбранными коэффициентами имеют порядок точности  $s$ .

Величина  $\varphi(h)$  называется погрешностью метода на шаге или локальной погрешностью метода, а первое слагаемое в (2.35) — главным членом локальной погрешности метода.

Формулы (2.32), (2.33) образуют семейство методов Рунге — Кутты порядка  $s$  и для их реализации требуется  $s$  вычислений функций  $f(t, x(t))$ , представляющих производные решения. Если  $a_{ij} = 0$  при  $j \geq i$  для всех  $i$ , то  $k_i$  может быть вычислено явным образом по значениям  $k_1, \dots, k_{i-1}$ . Поэтому такие формулы называются явными. Если  $a_{ij} = 0$  при  $j > i$  и  $a_{ii} \neq 0$ , то каждое  $k_i$  неявно определяется уравнением

$$k_i = f(t_k + c_i h, x_k + h \sum_{j=1}^{i-1} a_{ij} k_j + h a_{ii} k_i). \quad (2.36)$$

В этом случае необходимо каким-либо способом вычислить приближенное значение  $k_i$ , например, с помощью итераций по методу Ньютона. Такие методы Рунге — Кутта называются диагонально- неявными [80]. Тем самым выполнение одного шага диагонально- неявным методом для систем дифференциальных уравнений порядка  $n$  включает решение  $s$  систем нелинейных алгебраических уравнений размерности  $n$ . Если метод не является явным или диагонально неявным, его называют неявным. В этом случае все  $k_i, i=1, 2, \dots, s$  находятся одновременно в результате итерационного решения нелинейной системы алгебраических уравнений порядка  $n \times s$ .

В общем случае доказано [79, 80], что если  $s=1, 2, 3, 4$ , то всегда можно выбрать коэффициенты  $a_{ij}, b_i, c_i$  в (2.32), (2.33), чтобы получить метод Рунге — Кутта порядка точности  $s$ . При  $s=5$  невозможно построить метод типа Рунге — Кутта пятого порядка точности, необходимо брать в комбинации (2.32) более пяти членов.

Примером явного метода Рунге — Кутта второго порядка ( $s=2$ ) является метод Хойна [80], который в случае, если правая часть (2.16) не зависит от  $t$ , переходит в квадратурную формулу трапеций. При  $s=2, a_{11}=0, a_{12}=0, a_{21}=1, a_{22}=0, b_1=b_2=0,5, c_1=0, c_2=1$  метод определяется формулами

$$x_{k+1} = x_k + \frac{1}{2} h (k_1 + k_2)$$

$$k_1 = f(t_k, x_k), \quad k_2 = f(t_k + h, x_k + k_1)$$

и имеет погрешность порядка  $O(h^3)$ .

Метод Хойна имеет простую геометрическую интерпретацию (рис. II.7). Через точку  $A$  проводится прямая с угловым коэффициентом, равным среднему арифметическому угловых коэффициентов касательных  $AB$  и  $BC$ , которые строятся в методе Эйлера и в качестве решения берется ордината точки  $D$  пересечения этой прямой с прямой  $t=t_k$ .

В практических расчетах обычно используются методы Рунге — Кутта порядка  $s=4$ , для которых ошибка на шаге имеет порядок  $O(h^5)$ .

Следует отметить, что метод Эйлера по сути сам является методом Рунге — Кутта первого порядка. Однако по сравнению с ним методы Рунге — Кутта при  $s > 1$  имеют важное преимущество, так как обеспечивают более высокую точность, что позволяет увеличивать шаг интегрирования. Чтобы повысить эффективность вычислительного процесса,

величину шага следует выбрать из соображений максимальной допустимой ошибки на шаге. Такой выбор часто осуществляется автоматически и включается как составная часть в алгоритм, построенный по методу Рунге — Кутты. Среди методов типа Рунге — Кутта с автоматическим выбором шага и контролем погрешности обычно применяются методы Рунге — Кутта — Фельберга [77, 79], Рунге — Кутта — Мерсона [79] и другие. Особое семейство методов типа Рунге — Кутта, основанное на введении в уравнения (2.32), (2.33) матрицы Якоби системы (2.16), образуют методы Рунге —

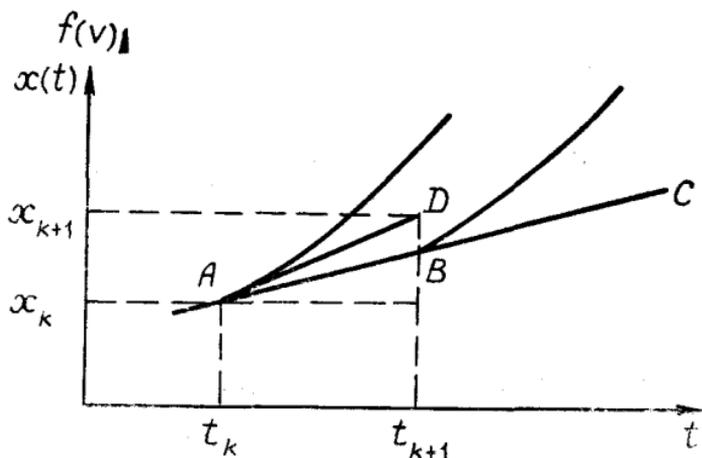


Рис. II. 7

Кутта — Розенброка [79], позволяющие более эффективно решать задачи определенного класса.

Таким образом, всем одношаговым методам присущи определенные общие черты [70]:

1. Чтобы получить информацию в новом узле, надо иметь данные лишь в одном предыдущем. Это свойство называется «самостартованием» метода.

2. В основе всех одношаговых методов лежит разложение функции правой части (2.16) в ряд Тейлора, в котором сохраняются члены, содержащие  $h$  в степени до порядка  $s$  включительно. Погрешность на шаге имеет порядок  $s+1$ .

3. Все одношаговые методы не требуют действительного вычисления производных — вычисляется лишь сама функция, значения которой могут потребоваться в нескольких промежуточных точках. Это влечет за собой дополнительные затраты времени.

4. Относительная простота оценки погрешности и свойство самостартования позволяют легко изменять шаг интегрирования.

### 2.2.2. Многошаговые методы решения задачи Коши

В данном пункте рассматриваются численные методы решения задачи Коши (2.16), которые задаются формулой (2.17). При этом значение решения  $x_k$  в точке  $t_k$  определяется через значение решения в  $m$  точках, предшествующих  $t_k$ . Такой метод называется  $m$ -шаговым. Из этого класса выделяются линейные многошаговые методы вида

$$\sum_{i=0}^m \alpha_i x_{k+i} = h \sum_{i=0}^m \beta_i f(x_{k+i}, t_{k+i}), \quad (2.37)$$

$$\alpha_i \neq 0$$

применяемые на сетке с постоянным шагом  $t_k = t_0 + hk$ ,  $k = 0, 1, \dots, N$ . Разность между наибольшим и наименьшим значениями индекса неизвестной функции  $x_k$  из (2.37) равна  $m$ . Поэтому уравнение (2.37) является разностным уравнением  $m$ -го порядка, общее решение которого зависит от  $m$  параметров. Чтобы выделить единственное решение этого уравнения, необходимо задать  $m$  дополнительных условий на функцию  $x_k$ , которыми являются значения функции  $x_k$  при  $k = 0, 1, \dots, m - 1$  и предполагаются известными. Таким образом, численные методы данного класса состоят в решении разностной задачи Коши для разностного уравнения (2.37). Если искомое решение  $x_{k+m}$  входит в правую часть уравнения (2.37) ( $\beta_m \neq 0$ ), то метод называется неявным. В противном случае ( $\beta_m = 0$ ) уравнение (2.37) может быть явно разрешено относительно  $x_{k+m}$ . В этом случае (2.37) определяет явный метод. Методы, задаваемые (2.37), часто называются конечно-разностными схемами [30, 79].

Коэффициенты  $\alpha_i$ ,  $\beta_i$  выбираются таким образом, чтобы разностное уравнение (2.37) аппроксимировало дифференциальное уравнение (2.16) с достаточно гладкой правой частью с некоторым порядком  $s$ . Это означает, что невязка  $\rho$ , которая получается после подстановки точного решения  $x(t)$  дифференциального уравнения (2.16) в разностное уравнение (2.37), является величиной  $O(h^{s+1})$ . Число  $s$  называется порядком аппроксимации или степенью разностного уравнения (2.37) (невязка  $\rho$  — погрешностью аппроксимации диф-

ференциального уравнения (2.16) разностным уравнением (2.37). Общим приемом для расчета коэффициентов в (2.37) является метод неопределенных коэффициентов [30]. В зависимости от выбора интерполяционного многочлена, который в узлах  $t_k$ ,  $k=m-s, m-s+1, \dots, m$  принимает значения  $f(t_k, x(t_k))$ , получаются различные многошаговые методы.

В практических расчетах обычно применяются сразу две формулы вида (2.36): явная, по которой осуществляется прогноз  $\hat{x}_{k+m}$  в точке  $t_{k+m}$ , и неявная, по которой проводится уточнение прогнозированного значения. Эти две формулы называются соответственно формулами прогноза и коррекции. Схемы алгоритмов для таких методов примерно одинаковы, а сами они различаются лишь формулами. Прежде чем применять метод прогноза и коррекции, вычисляются начальные значения для (2.37) с помощью какого-либо одношагового метода. Далее по прогнозированному значению  $\hat{x}_{k+m}$  находят по (2.16) производную  $\hat{x}'_{k+m} = f(t_{k+m}, \hat{x}_{k+m})$ , которая затем подставляется в формулу коррекции для уточнения значения  $x_{k+m}$ . В свою очередь  $x_{k+m}$  используется для получения более точного значения производной  $\dot{x}_{k+m}$ . Если это значение производной недостаточно близко к предыдущему, то оно вводится в формулу коррекции и итерационный процесс продолжается.

В качестве примеров рассмотрим часто используемые методы прогноза и коррекции. В методе Милна на этапе прогноза используется формула [70]

$$x_{k+1} = x_{k-3} + \frac{4}{3} h (2f(t_k, x_k) - f(t_{k-1}, x_{k-1}) + 2f(t_{k-2}, x_{k-2})), \quad (2.38)$$

а на этапе коррекции — формула Симпсона:

$$x_{k+1} = x_{k-1} + \frac{1}{3} h (f(t_{k+1}, x_{k+1}) + 4f(t_k, x_k) + f(t_{k-1}, x_{k-1})). \quad (2.39)$$

Этот метод относится к методам четвертого порядка точности. Остаточный член формулы (2.38) имеет порядок  $O\left(\frac{28}{90} h^5 x^{(5)}\right)$ , а формулы (2.39)  $O\left(\frac{1}{90} h^5 x^{(5)}\right)$ . Следовательно, формула коррекции (2.39) обеспечивает в 28 раз меньшую ошибку усечения, чем (2.38), и является более

точной. В практических расчетах метод Милна используется реже, чем другие, так как ему присуща неустойчивость (см. п. 2.3). Это означает, что сумарная погрешность может расти экспоненциально, особенно при интегрировании на больших интервалах  $[t_0, T]$ . Такое утверждение справедливо для всех формул коррекции, основанных на правиле Симпсона.

Метод Адамса — Башфорта также имеет четвертый порядок точности [79]. Используемая в нем формула прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид

$$x_{k+1} = x_k + \frac{1}{24} h (55f(t_k, x_k) - 59f(t_{k-1}, x_{k-1}) + 37f(t_{k-2}, x_{k-2}) - 9f(t_{k-3}, x_{k-3})). \quad (2.40)$$

На этапе коррекции используется формула

$$x_{k+1} = x_k + \frac{1}{24} h (9f(t_{k+1}, x_{k+1}) - 19f(t_k, x_k) - 5f(t_{k-1}, x_{k-1}) + f(t_{k-2}, x_{k-2})). \quad (2.41)$$

Расчеты по этому методу аналогичны методу Милна, однако формула (2.41) является устойчивой.

В методе Хэмминга [70, 79] используются формулы прогноза:

$$x_{k+1} = x_{k-3} + \frac{4}{3} h (2f(t_k, x_k) - f(t_{k-1}, x_{k-1}) + 2f(t_{k-2}, x_{k-2})). \quad (2.42)$$

и коррекции:

$$x_{k+1} = \frac{1}{8} (9x_k - x_{k-3} + 3h(f(t_{k+1}, x_{k+1}) + 2f(t_k, x_k) - f(t_{k-1}, x_{k-1}))). \quad (2.43)$$

Метод имеет четвертый порядок точности. Особенность его в том что он позволяет оценивать погрешности, вносимые на стадиях прогноза и коррекции, и устранять их. Благодаря простоте и устойчивости этот метод является наиболее распространенным среди применяемых в практических расчетах.

Отдельный класс методов вида (2.37) составляют методы, основанные на формулах дифференцирования назад (ФДН) [62, 79, 81]. Общий вид этих формул:

$$\dot{x}_k = \frac{1}{h_k} \sum_{i=0}^{m-1} \alpha_i x_{k-i}, \quad (2.44)$$

где  $h_k = t_k - t_{k-1}$  — величина  $k$ -го шага интегрирования,  $\alpha_i$  — коэффициенты, величина которых зависит от порядка  $m$  формулы (2.44) и значений  $m$  предшествующих шагов интегрирования. ФДН могут быть построены, если производные  $f(t_k, x_k)$  аппроксимировать в точке  $x = x_k$  с помощью односторонних формул численного дифференцирования [79]. К числу методов, основанных на ФДН, относятся метод Гира [79] и его модификации [79, 81], применяемые обычно для решения жестких задач (см. ниже п. 2.2.7). Следует отметить, что методы ФДН первого порядка ( $m=1$ ) совпадают с неявным методом Эйлера (2.31). Часто применяется ФДН второго порядка, называемая формулой Шихмана, которая при  $h_k = \text{const}$  имеет вид [81]

$$\dot{x}_k = \frac{1}{h} \left( \frac{3}{2} x_k - 3x_{k-1} + \frac{1}{2} x_{k-2} \right). \quad (2.45)$$

По сравнению с одношаговыми конечно-разностные методы имеют ряд особенностей [70]:

1. Для реализации многошаговых методов необходимо иметь информацию о нескольких предыдущих точках, т. е. они не относятся к самостартующим. Для начала вычислений приходится прибегать к какому-либо одношаговому методу.

2. При применении многошаговых методов определенная трудность связана с изменением шага, так как формула (2.37) справедлива при постоянном шаге интегрирования. При изменениях шага может потребоваться пересчет новых стартовых значений для (2.37).

3. Одношаговые и многошаговые методы обеспечивают примерно одинаковую точность результатов. Однако вторые в отличие от первых позволяют легко оценить погрешность на шаге. По этой причине, пользуясь одношаговыми методами, величину шага обычно выбирают несколько меньшей, чем это необходимо. Поэтому многошаговые методы оказываются более эффективными.

4. Применяя метод Рунге — Кутты четвертого порядка точности, на каждом шаге приходится вычислять четыре значения функции, в то время как для многошагового метода того

же порядка точности часто достаточно двух значений функции. Поэтому многошаговые методы, возможно, могут потребовать вдвое меньше машинного времени, чем методы Рунге-Кутты сравнимой точности.

### 2.2.3. Методы решения систем дифференциальных уравнений, неразрешенных относительно производной

В случае, когда ММС задана в форме (II.3), (II.4), система дифференциальных уравнений, подлежащая интегрированию, имеет вид

$$f(\dot{x}, x, t) = 0, \quad (2.46)$$

$$x(0) = x^0, \quad t_0 \leq t \leq T, \quad x \in R^n.$$

Обычно система (2.46) после дискретизации независимой переменной  $t$  в каждой точке  $t_k$  представляется в виде системы из  $n$  алгебраических уравнений:

$$f(\dot{x}_k, x_k, t_k) = 0, \quad (2.47)$$

которая доопределяется уравнениями алгебраизации

$$\dot{x}_k = g(x_k), \quad (2.48)$$

задаваемыми выбранным численным методом. Система (2.47) — (2.48) решается в каждой точке относительно  $x_k$  с помощью итерационной процедуры. В качестве уравнений алгебраизации (2.48) часто применяются ФДН (2.44). Например, при использовании ФДН первого порядка, при которой  $\dot{x}_k = \frac{1}{h}(x_k - x_{k-1})$ , система (2.47) — (2.48) преобразуется к системе нелинейных алгебраических уравнений

$$f\left(\frac{1}{h}(x_k - x_{k-1}), x_k, t_k\right) = 0, \quad (2.49)$$

которая решается относительно  $x_k$  в каждом  $k$ -м узле сетки.

Можно попытаться привести систему (2.46) к канонической форме Коши (2.16) и затем решать обычным способом. Однако при этом, пользуясь итерационными методами, необходимо на каждом шаге решать систему нелинейных алгебраических уравнений вида (2.47) относительно производной  $\dot{x}_k$ , что может оказаться слишком трудоемким.

Наконец, если продифференцировать  $f(\cdot)$  в (2.46):

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \dot{x} + \frac{\partial f}{\partial \dot{x}} \ddot{x} = 0, \quad (2.50)$$

можно получить дифференциальное уравнение второго порядка:

$$\ddot{x} = \varphi(t, x, \dot{x}), \quad (2.51)$$

которое приводится к форме Коши (2.16) введением дополнительных переменных. Обозначая в (2.51)  $z_1 = x$ ,  $z_2 = \dot{x}$ , получаем систему

$$\begin{aligned} \dot{z}_1 &= z_2, \\ \dot{z}_2 &= \varphi(t, z_1, z_2) \end{aligned} \quad (2.52)$$

размерности  $2n$ . Однако для решения (2.51) имеются методы [30, 65], позволяющие непосредственно получить решение. Например, можно воспользоваться формулами [65]:

$$\begin{aligned} x_{k+1} &= x_k + h\dot{x}_k + \frac{h^2}{2} [\alpha\varphi(t_{k+1}, x_{k+1}, \dot{x}_{k+1}) + \\ &+ (1 - \alpha)\varphi(t_k, x_k, \dot{x}_k)], \end{aligned} \quad (2.53)$$

$$\dot{x}_{k+1} = \dot{x}_k + h[\theta\varphi(t_{k+1}, x_{k+1}, \dot{x}_{k+1}) + (1 - \theta)\varphi(t_k, x_k, \dot{x}_k)],$$

которые при  $\alpha = \frac{1}{6}$  и  $\theta = \frac{1}{2}$  обеспечивают погрешность пятого порядка.

#### 2.2.4. Методы решения линейных систем дифференциальных уравнений

В случае, когда ММС имеет вид (II.5), (II.6), можно построить систему разностных уравнений, решение которой удовлетворяет решению (II.5), (II.6) с любым шагом дискретности.

Для простоты рассмотрим получение численного решения ММС (II.5), (II.6) при постоянном векторе входных воздействий  $u(t) = u_*$ . Полагая в (II.5)  $b = Bu_*$ , имеем неоднородную систему уравнений

$$\dot{x}(t) = Ax(t) + b, \quad (2.54)$$

$$x(0) = x_0, \quad x(t) \in R^n, \quad t \in [0, T],$$

решение которой записывается в виде

$$x(t) = \exp(At)x_0 + \int_0^t \exp(A\tau) d\tau b. \quad (2.55)$$

Известно [82], что матричная экспонента  $\exp(At)$  (экспоненциал) может быть определена как сумма равномерно сходящегося ряда:

$$\exp(At) = \sum_{i=0}^{\infty} \frac{A^i t^i}{i!}, \quad A^0 = I, \quad (2.56)$$

где  $I$  — единичная матрица. Степенное представление интеграла от экспоненциала имеет вид

$$\int_0^t \exp(A\tau) d\tau = \sum_{i=0}^{\infty} \frac{A^i t^{i+1}}{(i+1)!}. \quad (2.57)$$

Применять формулы (2.56) — (2.57) для достаточно больших  $t$  нецелесообразно, так как для достижения необходимой точности решения потребуется учесть слишком большое число слагаемых, растущее с увеличением  $t$ .

В настоящее время существует достаточно много способов вычисления экспоненциала и интеграла от него [82, 83]. Рассмотрим способ, основанный на дискретизации (2.54), предложенный в [83].

Пусть  $H$  — шаг, с которым требуется получить решение. Запишем (2.55) для  $t+H$ :

$$x(t+H) = \exp(AH + At)x_0 + \int_0^{t+H} \exp(A\tau) d\tau b. \quad (2.58)$$

Умножая (2.55) на  $\exp(AH)$  и вычитая из (2.66), получим

$$x(t+H) = \exp(AH)x(t) + \int_0^H \exp(A\tau) d\tau b. \quad (2.59)$$

Введем дискретные значения  $t_k = kH$ ,  $k=0, 1, 2, \dots$ . Обозначив  $x(t_{k+1}) = x_{k+1}$ ,  $x(t_k) = x_k$ , приходим к разностному уравнению вида

$$x_{k+1} = \exp(AH)x_k + \int_0^H \exp(A\tau) d\tau b, \\ x_k |_{k=0} = x_0. \quad (2.60)$$

Использование (2.60) требует вычисления матрицы  $\exp(AH)$  и вектора  $\int_0^H \exp(A\tau) d\tau b$ , не зависящих от  $t_k$ , которое может быть проведено с использованием рекуррентных формул. Пусть

$$h = H/2^N, \quad (2.61)$$

где  $N$  — целое число. Вычислив  $\exp(Ah)$  по (2.57), можем найти  $e^{AN}$ , применяя рекуррентные матричные соотношения

$$\varphi_{i+1} = \varphi_i^2, \quad (2.62)$$

где

$$\varphi_i = \exp(2^i Ah) \quad (2.63)$$

$$i = 0, 1, 2, \dots, N.$$

Очевидно, что  $\varphi_N = \exp(2^N Ah) = \exp(AH)$ .

Таким образом, можно выбрать настолько малую величину начального шага  $h$ , что обеспечивается сколь угодно высокая точность представления  $\exp(Ah)$  разложением (2.57) с малым числом членов:

$$\varphi_0 = \exp(Ah) = E + Ah + \dots + \frac{A^v h^v}{v!}, \quad (2.64)$$

а далее применять рекуррентное соотношение (2.62) для определения матрицы  $\exp(AH)$ .

Аналогично может быть получено выражение для второго слагаемого в (2.59):

$$\Phi_{i+1} = \Phi_i(I + \varphi_i), \quad (2.65)$$

где  $\Phi_i = \int_0^{2^i h} \exp(A\tau) d\tau$ ,  $i = 0, 1, \dots, N$ , а  $\varphi_i$  определяется (2.63).

На основе равенства

$$\exp(2^i Ah) = I + A\Phi_i = I + A \int_0^{2^i h} \exp(A\tau) d\tau \quad (2.66)$$

рекуррентное соотношение (2.65) преобразуется к виду

$$\Phi_{i+1} = \Phi_i(2I + A\Phi_i). \quad (2.67)$$

В этом случае матрица  $\Phi_i$  вычисляется независимо от  $\varphi_i$ . С целью минимизации числа арифметических операций при

решении (2.54) можно построить рекуррентные соотношения для вектора

$$g_i = \int_0^{2^i h} \exp(A\tau) d\tau b = \Phi_i b, \quad (2.68)$$

$$i=0, 1, \dots, N$$

вида

$$g_{i+1} = (I + \Phi_i) g_i. \quad (2.69)$$

Последнее соотношение используется совместно с (2.62). При достаточно малом  $h$  аналогично (2.64) можно применять разложение (2.59) с малым числом членов.

Оценим эффективность применения рекуррентных соотношений. Допустим, что для интегрирования (2.59) применяется явный метод Рунге-Кутты четвертого порядка:

$$x(kh + h) = \left( I + Ah + \frac{A^2 h^2}{2} + \frac{A^3 h^3}{6} + \frac{A^4 h^4}{24} \right) x(kh) +$$

$$+ h \left( I + \frac{Ah}{2} + \frac{A^2 h^2}{6} + \frac{A^3 h^3}{24} \right) b = \quad (2.70)$$

$$\bar{\varphi}_0 x(kh) + \bar{\Phi}_0 b = \bar{\varphi}_0 x(kh) + \bar{g}_0 b.$$

Пусть  $L = T/h$  — целое число. Тогда использование (2.70) потребует  $L$  умножений матрицы на вектор. С другой стороны, разностное уравнение (2.60) с учетом рекуррентных соотношений (2.62), (2.69) имеет вид

$$x(kH + H) = \varphi_N x(kH) + g_N, \quad k=0, 1, 2, \dots,$$

$$H = 2^N h. \quad (2.71)$$

При обеспечении той же точности, что и в (2.70) ( $v=4$  в (2.64)), последняя формула требует

$$T/H = 2^{-NT}/h \quad (2.72)$$

операций умножения матрицы на вектор, а также  $N(n+1)$  таких операций для формирования  $\varphi_N$  и  $g_N$  согласно (2.62), (2.69). Число  $n$  есть размерность системы (2.54), а черточки над векторами и матрицами означают их приближенное вычисление.

В тех случаях, когда величина  $T/h$  оказывается большой, формула (2.71) дает значительное преимущество по сравнению с (2.70).

Следует также отметить, что шаг дискретности  $H=2^N h$  выбирается исходя из возможности наблюдения процесса, описанного системой (2.54), а  $h=H/2^N$  должен быть достаточно малым. Например, можно выбирать

$$h \ll \frac{1}{\|A\|}, \quad (2.73)$$

где  $\|A\|$  — любая из матричных норм, и находить по известным  $H$  и  $h$  значение  $N$ . Число слагаемых в разложениях (2.58), (2.59), как правило, не превышает  $v=4$ .

Источником погрешности в определении решения являются не только приближения в (2.56) и (2.57), но и ограниченность разрядной сетки ЭВМ. Поэтому, как показывает опыт использования этого алгоритма, рекуррентные соотношения (2.62), (2.65) и (2.71) значительно менее чувствительны к влиянию вычислительной погрешности, чем (2.67). Однако при этом рекомендуется производить вычисления с двойной точностью.

### 2.2.5. Устойчивость численного интегрирования

Для сравнительной оценки явных и неявных методов интегрирования необходимо установить факторы, ограничивающие увеличение шага  $h$ , и на этой основе сделать прогноз требуемого количества шагов для решения типовой задачи, в качестве которой обычно выбирается линейная однородная система дифференциальных уравнений с постоянной матрицей коэффициентов:

$$\dot{x} = Ax, \quad x(0) = x^0, \quad (2.74)$$

$$x \in R^n, \quad A_{n \times n}.$$

Обычно с ростом  $h$  увеличивается погрешность  $\varepsilon$  решения задачи. Характер роста погрешности может быть различным. В ряде случаев при достижении  $h$  некоторого критического значения  $h_*$  наблюдается резкий рост погрешности. При  $h \geq h_*$  значения решения  $x(t)$  начинают меняться со все возрастающей амплитудой, и при вычислениях на ЭВМ обычно быстро наступает переполнение разрядной сетки. Получаемые при этом значения никак не отражают правильного решения. Такое явление катастрофического роста погрешности называется неустойчивостью (потерей устойчивости) вычислительного процесса. Если бы удалось ликвидировать при-

чины возникновения неустойчивости, то появилась бы возможность существенно увеличить шаг при обеспечении приемлемой точности. Поэтому в неустойчивых методах ограничением величины шага  $h$  является именно  $h_*$  и шаг не удастся увеличить из-за потери устойчивости.

Для примера определим  $h_*$  для явного метода Эйлера (2.29) на примере модельной задачи (2.74), который в данном случае имеет вид

$$x_{k+1} = (I + Ah)x_k, \quad (2.75)$$

где  $I$  —  $n \times n$  единичная матрица. Решение (2.75) через  $m$  шагов может быть записано как

$$x_{k+m} = (I + Ah)^m x_k, \quad (2.76)$$

поэтому для устойчивости решения (2.75) ( $x_k \rightarrow 0$  при  $k \rightarrow \infty$ ) необходимо выполнение условия

$$\lim_{k \rightarrow \infty} (I + hA)^k = 0. \quad (2.77)$$

Воспользуемся известным в теории матриц преобразованием подобия и представим матрицу  $\Phi = I + hA$  в виде

$$\Phi = T \operatorname{diag} (\lambda_{\Phi_j}) T^{-1}, \quad (2.78)$$

где  $T$  — неособая матрица преобразования,  $\operatorname{diag} (\lambda_{\Phi_j})$  — диагональная матрица с собственными значениями  $\lambda_{\Phi_j}$ ,  $j = 1, 2, \dots, n$  матрицы  $\Phi$  на диагонали. Так как

$$\Phi^k = (I + hA)^k = T \operatorname{diag} (\lambda_{\Phi_j}^k) T^{-1}, \quad (2.79)$$

то для выполнения (2.77) необходимо

$$|\lambda_{\Phi_j}| < 1, \quad j = 1, 2, \dots, n, \quad (2.80)$$

т. е. чтобы все собственные числа матрицы  $I + hA$  по модулю были меньше единицы. В теории матриц доказывается соотношение, что собственные значения  $\lambda_B$  и  $\lambda_C$  коммутирующих матриц  $B$  и  $C$  связаны между собой таким же соотношением, как и сами матрицы  $B$  и  $C$ . Поэтому на основании (2.80)

$$|\lambda_{\Phi_j}| = |\lambda_{(I+hA)_j}| = |1 + h\lambda_{\Phi_j}| < 1 \quad (2.81)$$

и, следовательно, для обеспечения устойчивости явного метода Эйлера с устойчивой матрицей  $A^*$  в (2.74) необходимо выполнение условия:

$$0 < h < \frac{2 |\operatorname{Re} \lambda_{A_j}|}{|\lambda_{A_j}|}. \quad (2.82)$$

В частном случае, когда все собственные значения матрицы  $A$  — вещественные числа, можно использовать понятие постоянной времени:

$$\tau_{A_j} = \frac{1}{\lambda_{A_j}}. \quad (2.83)$$

При этом условие устойчивости (2.82) приобретает вид

$$0 < h < 2\tau_{A_j}, \quad j = 1, \dots, n. \quad (2.84)$$

Очевидно, что наиболее жесткое ограничение на  $h$  накладывается минимальной постоянной времени  $\tau_{\min}$  (максимальным по модулю собственным значением  $\lambda_{\max}$ ):

$$0 < h < 2\tau_{\min}. \quad (2.85)$$

Таким образом, для явного метода Эйлера в случае системы (2.74) с устойчивой матрицей  $A$  можно принять критическое значение шага  $h_* = 2\tau_{\min}$ . В случае нелинейных систем вида (2.1) аналогичное условие накладывается на мгновенные значения собственных чисел матрицы Якоби:

$$\left[ \frac{\partial f(t_k, x(t_k))}{\partial x(t_k)} \right], \quad k = 0, 1, \dots$$

Исследование условий устойчивости для других явных методов дает сходную картину:  $h_* = c\tau_{\min}$ , причем константа  $c$  хотя и растет с увеличением порядка метода  $s$ , но несущественно. Например, для метода Рунге-Кутты четвертого порядка ( $s=4$ )  $c=2,78$ .

Среди неявных методов интегрирования имеется ряд устойчивых методов, примером которых является неявный метод Эйлера. Применяя (2.31) к уравнению (2.74), получим

$$x_{k+1} = (I + hA)^{-1} x_k. \quad (2.86)$$

\* Матрица  $A$  называется устойчивой (гурвицевой), если вещественные части ее собственных значений отрицательны, т. е.  $\operatorname{Re}(\lambda_{A_j}) < 0$ ,  $j = 1, 2, \dots, n$ .

Рассуждая аналогично явному методу Эйлера, можно получить условия устойчивости:

$$\left| \frac{1}{1 - h\lambda_{A_j}} \right| < 1, \quad (2.87)$$

которые очевидно выполняются для устойчивой матрицы  $A$  при любых шагах  $h > 0$ .

Методы, в которых при решении модельных уравнений вида (2.74) с любым постоянным шагом  $h$  полная погрешность стремится к нулю при  $k$  и  $t_k \rightarrow \infty$  ( $\text{Re } \lambda_j \leq 0$ ), принято называть абсолютно устойчивыми или  $A$ -устойчивыми. Применение  $A$ -устойчивых методов избавляет численные методы интегрирования от ограничения на величину максимального шага интегрирования. Требование  $A$ -устойчивости налагает сильные ограничения на метод интегрирования. В частности, порядок линейных многошаговых формул (2.37) не может превышать двух и они обязательно должны быть неявными. Кроме того, эти формулы недостаточно точны (наименьшей погрешностью обладает неявная формула трапеций с погрешностью  $O\left(\frac{1}{12} h^3 x^{(3)}\right)$  и шаг интегрирования часто приходится сокращать, чтобы гарантировать приемлемую точность результата. Однако существуют  $A$ -устойчивые одношаговые методы с порядком  $s > 2$  [80].

В общем случае при исследовании условий устойчивости численного метода исследуют характер расположения корней характеристического многочлена

$$p(z) = \sum_{i=0}^s a_i z^i, \quad (2.88)$$

в котором коэффициенты  $a_i$  определяют свойства конкретного метода. Метод называется устойчивым, если все корни (2.88) расположены в единичном круге на комплексной плоскости с центром в начале координат, а на границе круга нет кратных корней. Если характеристический многочлен имеет один корень  $z_1 = 1$ , а все остальные лежат внутри единичного круга, то метод называется сильно устойчивым. Если кроме корня  $z_1 = 1$  имеются несколько простых корней на границе, то метод называется слабо устойчивым. Применение слабо устойчивых формул для интегрирования на больших промежутках связано с определенными трудностями [70, 79].

## 2.2.6. Выбор шага интегрирования

Одним из важных вопросов является выбор подходящей величины шага интегрирования  $h$ . Если шаг слишком мал, то расчет потребует неоправданно большого машинного времени, если же, наоборот, шаг выбран слишком большим, то значительной будет погрешность усечения.

Обычно, выбирая величину шага, стремятся, чтобы локальная погрешность на шаге была меньше некоторой заданной величины. В общем случае, если метод имеет порядок  $s$ , то ошибка усечения  $\epsilon_k$  имеет порядок  $O(h^{s+1})$ . Поэтому за

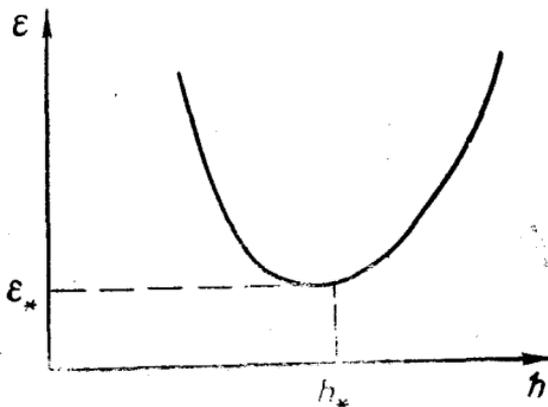


Рис. II.8

счет уменьшения шага интегрирования можно сделать ошибку усечения сколь угодно малой. Однако чем меньше  $h$ , тем больше потребуется шагов и тем больше скажутся на полученном решении ошибки округления  $\delta_k$ . На практике, когда при выполнении арифметических операций используется разрядная сетка фиксированной длины, всегда существует граница шага  $h$ , меньше которой вклад ошибок округления начинает доминировать в суммарной ошибке. Эта ситуация отражена на рис. II.8, где  $h_*$  определяет минимальный шаг, который можно использовать в практических расчетах. Эту величину шага очень трудно установить заранее, но в задачах, где не требуется высокая точность, необходимый шаг обычно будет значительно больше, чем  $h_*$ , и основной вклад в полную ошибку  $R_k$  будет вносить ошибка усечения.

В практических расчетах оценить совершаемую локальную ошибку при заданном шаге интегрирования можно различными способами. Наиболее простым является способ, ос-

нованный на правиле Рунге [30, 70, 79], которое заключается в том, что по одной и той же формуле численного метода вычисляются два приближения к решению в одной точке, но с разными шагами, которые затем используются для получения апостериорной оценки погрешности. Так, например, если для вычисления значения искомой функции  $x(t)$  в точке  $t_k$  используется шаг  $h$ , то разность между истинным значением  $x(t_k)$  и вычисленным  $x_k$  равна:

$$x(t_k) - x_k = ch^{s+1}, \quad (2.89)$$

где  $c$  — некоторая константа,  $s$  — порядок метода. Если уменьшить шаг вдвое и вычислить  $\bar{x}_k$ , то получим

$$x(t_k) - \bar{x}_k = c \left( \frac{h}{2} \right)^{s+1}. \quad (2.90)$$

Вычитая из (2.90) (2.89), можно найти оценку локальной погрешности:

$$ch^{s+1} + \frac{2^{s+1}(x_k - \bar{x}_k)}{2^{s+1} - 1}. \quad (2.91)$$

Недостатком такого подхода является то, что значение  $x_k$  в узле  $t_k$  приходится вычислять дважды, поэтому объем вычислений увеличивается более чем вдвое.

Другой способ оценки локальной погрешности также основан на использовании двух приближенных значений решения в одной точке. Однако, в отличие от правила Рунге, приближения вычисляются по двум формулам разного порядка точности  $p$  и  $s$  с одним и тем же шагом. Если  $p > s$ , то оценка локальной погрешности имеет вид [79]

$$\rho_k^s = x_k^p - x_k^s + O(h^{p+1}), \quad (2.92)$$

где  $x_k^p$ ,  $x_k^s$  — соответственно два решения в узле  $t_k$ , полученные по формулам порядка  $p$  и  $s$ . Полученная оценка погрешности может потребовать меньшего количества вычислений правой части (2.1), чем по правилу Рунге. При специальном выборе формул порядка  $p$  и  $s$  можно значительно снизить дополнительное количество вычислений правой части (2.1). В этом случае  $\rho_k^s$  называют контрольным членом. На практике широко используются методы Рунге-Кутты, использующие контрольные члены Фельберга, Мерсона, Ингланда и другие [79].

Рассмотренные способы оценки локальной погрешности позволяют менять величину шага в процессе счета и выби-

рять шаг автоматически. При этом обычно исходят из того, чтобы на каждый шаг приходилась примерно одинаковая погрешность. Наиболее простой способ изменения шага интегрирования основан на методе удвоения или деления пополам. Пусть  $\rho_{k+1}$  — оценка локальной погрешности метода на шаге  $h$ , допущенная в узле  $t_k + h$ . Если оценка превосходит некоторую заданную границу  $\varepsilon$ , то считается, что значение  $x_{k+1}$  не удовлетворяет заданной точности, и шаг  $h$  дробится пополам, после чего вычисляют решение с шагом  $h$  для точки  $t_k + h/2$ . Шаг  $h$  дробится пополам до тех пор, пока  $|\rho_k|$  не станет меньше  $\varepsilon$ . Полученное решение принимается за  $x_{k+1}$ . Дальнейшее интегрирование осуществляется с шагом  $\hat{h}$ , который выбирается из следующих соображений: если  $|\rho_{k+1}| \leq \varepsilon/L$ , где  $L$  — некоторая константа, то считается, что достигнута точность, превышающая заданную, и шаг удваивается  $\hat{h} = 2h_r$ , где  $\hat{h}_r = h/2^r$ ,  $r$  — число дроблений шага; если  $\varepsilon/L \leq |\rho_{k+1}| \leq \varepsilon$ , то считается, что решение удовлетворяет заданной точности, и шаг интегрирования не изменяется  $\hat{h} = h_r$ . Таким образом, на тех участках изменения независимой переменной, где достигается высокая точность, шаг интегрирования возрастает, а там, где точность не достигается, он сокращается. Тем самым обеспечивается выбор величины шага в зависимости от характера поведения решения  $x(t)$ . Константа  $L$  обычно получается равной  $2^v$ , где  $v$  — порядок используемой оценки локальной погрешности.

Во многих методах величина шага интегрирования выбирается максимальной, исходя из заданной точности  $\varepsilon$ . Обычно такой способ основан на том факте, что оценка локальной погрешности метода равна с точностью до более высокого порядка малости главному члену погрешности метода:

$$\rho_{k+1} \cong ch^{s+1}, \quad (2.93)$$

где  $c$  — некоторая константа. Если оценка  $\rho_{k+1}$  погрешности превосходит заданную границу  $\varepsilon$ , то считается, что на данном шаге  $h$  требуемая точность не достигнута, и вычисленное значение  $x_{k+1}$  вместе с узлом  $t_{k+1}$  исключаются из рассмотрения. В этом случае новый размер шага вычисляется из соотношения

$$\hat{h} = ah, \quad (2.94)$$

где  $a$  находят из условия равенства:

$$|ch^{s+1}| = \varepsilon. \quad (2.95)$$

Из (2.93), (2.94) получаем, что

$$\alpha = \sqrt[s+1]{\varepsilon/|\rho_{k+1}|}. \quad (2.96)$$

Таким образом, шаг интегрирования изменяется по формуле

$$\hat{h} = h^{s+1} \sqrt[s+1]{\varepsilon/|\rho_{k+1}|}. \quad (2.97)$$

В рассмотренном случае  $\alpha < 1$  и новый шаг  $\hat{h}$  уменьшается. При выполнении условия  $|\rho_{k+1}| \leq \varepsilon$ ,  $\alpha > 1$  и шаг интегрирования увеличивается. Преимущество данного алгоритма выбора шага заключается в большой гибкости по сравнению с алгоритмом дробления — удвоение шага. Это обстоятельство приводит к более сглаженному изменению шага интегрирования и, как следствие, к сокращению общего количества шагов и снижению вычислительных затрат. Следует также отметить, что для сокращения шагов, в которых заданная точность не достигается, параметр  $\alpha$  в (2.94) берется несколько меньшим, чем вычисленный по формуле (2.96).

### 2.2.7. Особенности интегрирования жестких систем

Задачи, называемые жесткими, весьма разнообразны, поэтому в литературе [30, 50, 64, 67, 79—81, 83] можно встретить различные определения жесткости, различающиеся по степени строгости. Сущность явления жесткости состоит в том, что решение, которое нужно вычислить, меняется медленно, однако существуют быстро затухающие возмущения. Наличие таких возмущений затрудняет получение медленно меняющихся решений численным способом. Для подтверждения этого рассмотрим ряд простых примеров.

Исследуем дифференциальное уравнение первого порядка [80]:

$$\begin{aligned} \dot{x}(t) &= \lambda x(t) + F(t) - \lambda F(t), \\ t &\geq 0, \quad x(0) = x_0, \quad \lambda \ll 0, \end{aligned} \quad (2.98)$$

где  $F(t)$  — медленно меняющаяся функция от  $t$ . Решение (2.98) в момент времени  $t$  имеет вид

$$x(t) = F(t) + e^{\lambda t} [x(0) - F(0)]. \quad (2.99)$$

Так как  $\lambda \ll 0$ , то ясно, что уже после очень небольшого отрезка времени второе слагаемое в (2.93) в  $x(t)$  больше не присутствует. Это означает, что независимо от начального значения  $x_0$  в численном решении на большей части отрезка интегрирования преобладает медленно меняющаяся функция  $F(t)$ .

Для выяснения особенностей численного решения задачи (2.98) применим к ней явный (2.29) и неявный (2.31) методы Эйлера и сравним аппроксимации (2.101), (2.102) с точным решением  $x(t_k)$ ,  $t_{k+1} = t_k + h$ , заданным (2.100). Имеем:

$$x(t_{k+1}) = e^{h\lambda}[x(t_k) - F(t_k)] + F(t_{k+1}), \quad (2.100)$$

$$x_{k+1} = (1 + h\lambda)[x_k - F(t_k)] + F(t_k) + h\dot{F}(t_k), \quad (2.101)$$

$$x_{k+1} = (1 + h\lambda)^{-1}[x_k - F(t_k)] + (1 - h\lambda)^{-1} \times \\ \times [F(t_k) + h\dot{F}(t_{k+1}) - h\lambda F(t_{k+1})]. \quad (2.102)$$

Как уже указывалось, разность  $x(t_k) - F(t_k)$  может быть интерпретирована как возмущение гладкого решения  $F(t)$  в момент времени  $t = t_k$ . Это возмущение быстро убывает из-за экспоненты с отрицательным показателем  $e^{h\lambda}$ ,  $h\lambda \ll 0$ . Поэтому численный метод должен быть в состоянии подавить эти разности для значений  $h\lambda \ll 0$ , и шаг  $h$  должен определяться только в зависимости от изменений функции  $F(t)$ .

Из (2.102) видно, что неявный метод Эйлера вполне подходит для решения (2.98). Разности  $x_k - F(t_k)$  быстро убывают и для любого  $h > 0$   $x_{k+1} \rightarrow F(t_{k+1})$  при  $h\lambda \rightarrow -\infty$ . Явный метод Эйлера способен подавить разность  $x_k - F(t_k)$  только при  $-2 < h\lambda < 0$  (см. п. 2.1.4). Это условие устойчивости накладывает сильное ограничение на шаг  $h$  при  $\lambda h \ll 0$ , даже в случае, когда разность  $x_k - F(t_k)$  пренебрежимо мала. С другой стороны, если  $x \simeq F(t_k)$  и  $F(t)$  — очень гладкая функция, то аппроксимация  $F(t_{k+1})$  выражением  $F(t_k) + h\dot{F}(t_k)$  будет приемлемой для много больших значений  $h$ , чем для тех из них, которые допускаются условием устойчивости  $h < -2/\lambda$ . Эта ситуация является типичной при применении явного метода к решению жесткой задачи. Величина шага ограничивается скорее численной устойчивостью, чем точностью.

Аналогичная ситуация возникает при интегрировании

дифференциального уравнения первого порядка с малым параметром  $\varepsilon$  при производной [83]:

$$\varepsilon \frac{dx}{dt} = f(t, x), \quad (2.103)$$

$$\varepsilon > 0, x \in R^1$$

для случая, когда вырожденное уравнение ( $\varepsilon=0$ ), соответствующее (2.103),

$$f(t, x) = 0 \quad (2.104)$$

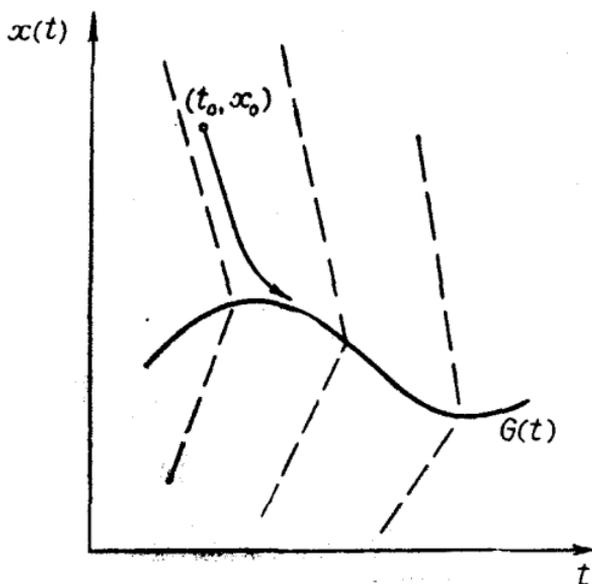


Рис. II.9

имеет единственное решение

$$\bar{x}(t) = G(t) \quad (2.105)$$

и в окрестности этого решения величина  $\frac{\partial f}{\partial x} < 0$ . Последнее условие означает устойчивость решения (2.105).

Характер поведения решения (2.103) показан на рис. II.9. Штриховыми линиями обозначено поле направления касательных к интегральным кривым. Для достаточно малого  $\varepsilon$  касательные к интегральным кривым почти параллельны оси  $x$ . И чем меньше величина  $\varepsilon$ , тем быстрее осуществляется сближение интегральной кривой и решения (2.105) вырожденного уравнения (2.104). При этом в рассматривае-

мой области выделяются два участка с существенно различным поведением решения, причем продолжительность первого участка значительно меньше второго. Первый участок с быстрым изменением искомой функции отражает стремление интегральной кривой к графику функции  $x(t) = G(t)$  и называется пограничным слоем. На втором участке производные решения значительно меньше, а интегральная кривая практически совпадает с графиком  $G(t)$ . Различный характер поведения решения на обоих участках тем отчетливее, чем меньше величина параметра  $\varepsilon$ . Таким образом, вне пограничного слоя для описания решения дифференциального уравнения (2.103) может быть использовано решение алгебраического уравнения (2.104).

Кривые, аналогичные рис. II.9, могут отвечать и более сложным системам уравнений. При этом если явно выделен малый параметр при производных, то компоненты вектора решений могут быть разделены на «быстрые» и «медленные». Уравнения с явным вхождением малого параметра называются сингулярно-возмущенными. Подобная ситуация возникает не только в этих уравнениях, но и в любых других системах с сильно разнесенными постоянными времени. Например, в линейных ММС вида (II.5) система считается жесткой, если для собственных чисел  $\lambda_i, i=1, \dots, n$  матрицы  $A_{n \times n}$  справедливо неравенство:

$$\mu(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|} \gg 1, \quad (2.106)$$

$$\operatorname{Re}(\lambda_i) < 0, \quad i=1, 2, \dots, n.$$

где  $\operatorname{Re}(\lambda_i)$  — действительная часть  $\lambda_i$ . В нелинейных ММС система может считаться жесткой, если для собственных чисел мгновенных значений матрицы Якоби  $\left[ \frac{\partial f(x, t)}{\partial x} \right]$  выполняется (2.106). Величина  $\mu(A)$  в (2.106) называется спектральным числом обусловленности матрицы  $A$ .

Рассмотренная выше ситуация иллюстрирует явление жесткости, смысл которого состоит в необходимости привлечения для полного описания процессов на любом отрезке наблюдения двух видов функций: с большими производными и с малыми, причем функции с большими производными быстро убывают, так что большую часть времени протекания процесса доступны наблюдению лишь функции с малы-

ми производными. Однако в любой момент времени наблюдения снова возможно возникновение быстрозатухающего процесса.

Дифференциальные уравнения, моделирующие процессы с описанным выше явлением жесткости, называются жесткими [83]. Необходимость выделения уравнений данного типа в отдельный класс была вызвана трудностями их численного интегрирования классическими методами, например, явными методами типа Адамса или Рунге-Кутта. Выяснилось, что малый шаг интегрирования, используемый для воспроизведения быстропротекающих процессов в пограничном слое, не может быть увеличен и вне пограничного слоя, хотя производные становятся существенно меньше. Даже незначительное увеличение шага интегрирования приводит к резкому возрастанию погрешности и потере устойчивости численного метода. Требование уменьшения шага интегрирования приводит к недопустимо большим затратам машинного времени.

Для численного интегрирования жестких систем дифференциальных уравнений целесообразно воспользоваться следующими рекомендациями [30, 64, 68, 79—81, 83]. При интегрировании линейных ММС вида (II.5), (II.6) в случае кусочно-постоянных входных воздействий  $u(t)$  наиболее эффективным способом является дискретизация ММС, например, как в подразд. 2.2. В этом случае основные вычислительные трудности заключаются в приближенном вычислении или использовании какой-либо аппроксимации матричной экспоненты  $\exp(AN)$ , где  $N$  — шаг дискретизации. Ряд алгоритмов вычисления экспоненциала рассмотрен в [82, 83]. Однако для систем больших порядков могут возникнуть значительные вычислительные погрешности при вычислении степеней матрицы  $A$ . Поэтому можно использовать аппроксимацию Падэ матричной экспоненты (см. ч. I, п. 2.2.3).

При численном интегрировании систем общего вида наиболее эффективными являются неявные линейные многошаговые методы интегрирования, основанные на формулах дифференцирования назад (подразд. 2.1) [68, 79, 81]. В этом случае для получения численного решения используются следующие алгоритмические приемы:

— замена производной  $x(t)$  формулой обратной разности, которая является функцией от состояния  $x(t)$  (например, как в (2.44));

— автоматический выбор шага и порядка метода для обеспечения заданной точности;

— решение результирующей нелинейной системы (2.49) с помощью квадратично-сходящегося метода Ньютона;

— решение системы линейных алгебраических уравнений на каждой итерации метода Ньютона с помощью метода Гаусса.

Методы этого класса особенно эффективны для решения больших жестких систем с разреженными матрицами. Однако при сильно осциллирующих процессах или разрывных правых частях ММС для обеспечения устойчивости этих методов требуется применение специальных мер, не исключая переход к явным методам интегрирования.

В настоящее время много внимания уделяется использованию неявных и полужявных методов Рунге-Кутты для жестких систем [79, 80]. Среди них перспективными с вычислительной точки зрения являются диагонально неявные и однократно неявные методы. В то же время большинство из них принадлежит к классу методов Рунге-Кутты-Розенброка. Эти методы способны давать устойчивые решения нелинейных задач в случае, если функция в правой части (II.1) может быть расщеплена на жесткую линейную и нежесткую нелинейную части.

Дальнейшей тенденцией повышения эффективности методов численного интегрирования больших жестких систем является использование методов декомпозиции, среди которых выделяются две категории: декомпозиция на основе разбиения системы на отдельные подсистемы простой структуры и временное разложение, допускающее независимый анализ отдельных подсистем.

Следует отметить, что в настоящее время проблема численного решения жестких систем является актуальной. При этом расширение классов решаемых задач, как правило, приводит к тому, что явление жесткости в задаче анализа динамики ММС — скорее правило, чем исключение.

### *2.2.8. Выбор метода интегрирования*

Для успешного применения метода численного интегрирования к задаче анализа динамики ММС он должен обеспечивать малые затраты машинного времени и быть достаточно универсальным [62].

Наибольшие трудности возникают при определении приемлемых затрат машинного времени  $T_m$ , которые зависят от размерности  $n$  решаемых задач. Если не принимать специальных мер по учету особенностей структуры и разреженности матриц Якоби ММС, то применение неявных методов приводит к кубической зависимости  $T_m$  от  $n$ . Обычно неявный метод требует выполнения  $N$  шагов, на каждом из которых производится в среднем  $M$  итераций Ньютона для решения СНАУ, на каждой итерации методом Гаусса решается СЛАУ за  $n^3/3 + 2n^2$  арифметических операций. Поэтому при

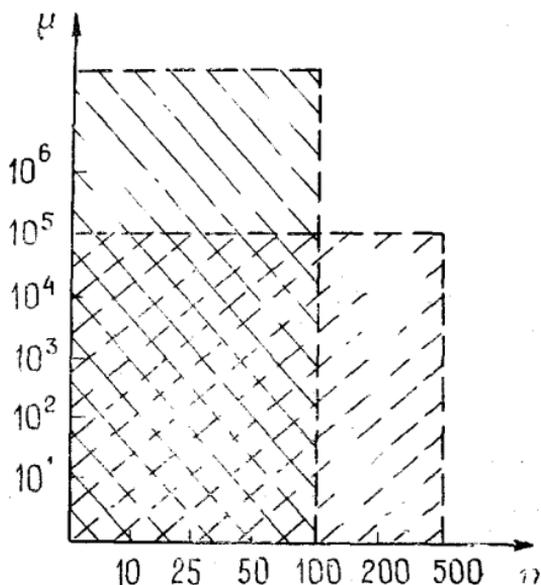


Рис. II. 10

неучете разреженности матрицы Якоби затраты машинного времени можно рассчитать по формуле [62]

$$T_m \cong NH/\tau \left( 2 \left( \frac{n^3}{3} + 2n^2 \right) + N_{\text{я}} + N_{\text{н}} \right), \quad (2.107)$$

где  $\tau$  — быстродействие ЭВМ,  $N_{\text{я}}$  и  $N_{\text{н}}$  — количество арифметических операций, затрачиваемых на вычисление матрицы Якоби и вектора невязки соответственно. В общем случае  $N_{\text{я}}$  зависит квадратично от  $n$ , а  $N_{\text{н}}$  — линейная функция от  $n$ .

Явные методы обеспечивают меньшие вычислительные затраты в пределах одного шага, где требуется вычисление лишь правых частей ММС. Поэтому количество операций на одном шаге здесь равно  $N_{\text{н}}$  и линейно зависит от  $n$ . Но из-за

ограничений на величину шага интегрирования, вытекающих из условий устойчивости, существует зависимость числа шагов  $h$  от числа обусловленности  $\mu$  матрицы Якоби исследуемой ММС. Поэтому явные методы рекомендуется применять только при анализе динамики ММС с умеренным числом обусловленности  $\mu$ , лежащим в диапазоне до  $10^5$ .

На рис. II. 10 представлены в координатах  $\mu - n$  [61] ориентировочные области, в которых удобнее применять рассмотренные методы: сплошными линиями выделена область применения неявных методов, пунктирными — явных.

Следует отметить, что методы явного интегрирования обеспечивают линейную зависимость трудоемкости одного шага  $N_n$  от размерности задачи  $n$  только для ММС вида (II. 1), (II. 2) (заданной в форме Коши). Следовательно, для получения решения ММС (II. 3), (II. 4) требуется предварительное приведение системы к форме Коши, что является недостатком явных методов интегрирования по сравнению с неявными.

Сравнение методов разного порядка точности  $s$  показывает [61], что в явных методах при интегрировании с шагом  $h$ , близким к  $h_{кр}$ , при котором наступает неустойчивость, методы более высокого порядка не обеспечивают более высокой точности, но приводят к росту арифметических операций на шаге. Аналогичное явление возможно и в неявных методах, однако применение алгоритмов с автоматическим выбором шага  $h$  и порядка метода  $s$  неявных формул вида (2.44) позволяет значительно повысить их эффективность.

Таким образом, сравнение явных и неявных методов численного интегрирования свидетельствует о большей универсальности последних и лучшей эффективности при анализе сложных ММС. Явные методы могут давать лучшие результаты в хорошо обусловленных задачах умеренной размерности.

В практических ситуациях при наличии хорошего пакета анализа динамики или библиотеки программ численного анализа при выборе метода численного интегрирования можно воспользоваться рис. II. 11 [79].

### 3. КУЛЬТУРА ВЫЧИСЛЕНИЙ НА ЭВМ

До сих пор мы говорили о постановках задач и об алгоритмах их решения. Однако успех решения задачи с применением ЭВМ определяется, очевидно, успехом каждого этапа

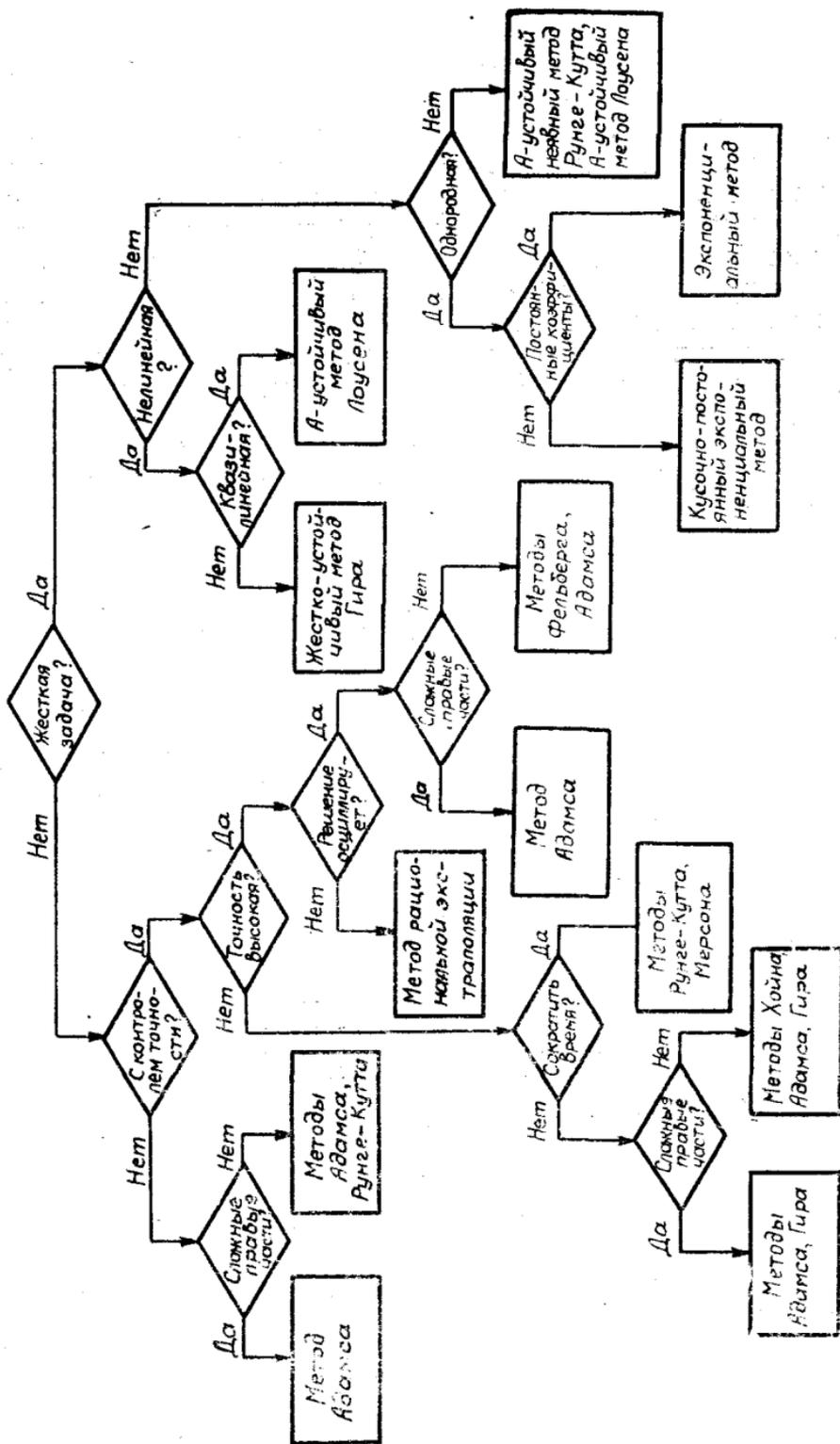


Рис. II. II

цепочки «модель — алгоритм — программа». Это значит, что самая точная математическая модель, самый великолепный алгоритм решения задачи могут быть безнадежно испорчены при плохой программной реализации. Одной из возможных причин этого является несовпадение машинной арифметики с обычной из-за конечности разрядной сетки ЭВМ. Возникающие ошибки могут привести к большим неприятностям, если их не контролировать и не соблюдать некоторые элементарные правила организации вычислений. Правила эти неформальны и напоминают правила хорошего тона. Уровень их выполнения определяет уровень вычислительной культуры пользователя ЭВМ. Поясним на примерах основные из этих правил.

**Пример 1.** Выполняя вычисления как на обычном микрокалькуляторе (например на БЗ-34), так и на ЕС ЭВМ\*, легко убедиться, что  $10^8 + 1 - 10^8 = 0$ ,  $10^{20} + 10^6 - 10^{20} = 0$ , но в то же время  $10^8 - 10^8 + 1 = 1$ ,  $10^{20} - 10^{20} + 10^6 = 10^6$ . Таким образом, в машинной арифметике нарушаются законы коммутативности и ассоциативности действий. Применимость основных выводов элементарной математики ставится под сомнение.

**Пример 2.** Хорошо известно, что  $\lim_{n \rightarrow \infty} e_n = e$ , где  $e_n = (1 + 1/n)^n$ ,  $e \approx 2.718281828 \dots$  — основание натуральных логарифмов. Попробуем приближенно вычислить этот предел на ЭВМ. Пользуясь, например, микрокалькулятором БЗ-34, получаем

$n$	$e_n$
$10^5$	2.7182682
$10^7$	2.7182817
$1.1 \cdot 10^7$	3.0041656
$1.2 \cdot 10^7$	3.3201163
$2 \cdot 10^7$	7.3890551
$3 \cdot 10^7$	1

Таким образом, в ЭВМ нарушается представление о пределе последовательности как о числе, к которому приближаются все ее удаленные члены. Применимость основного понятия высшей математики — предела — также ставится под сомнение.

\* Действия выполняются слева направо.

Для определения границ применимости привычных понятий удобно ввести определение.

**Определение.** *Машинным эpsilonом* называется наименьшее представимое в ЭВМ число  $\epsilon$ , удовлетворяющее условию  $1+\epsilon > 1$ , т. е.  $\epsilon_m = \min\{\epsilon : 1+\epsilon > 1\}$ .

**Правило 1.** *Величина  $\epsilon_m$  характеризует наименьшую относительную погрешность вычислений и зависит от конкретной ЭВМ.*

Величина  $\epsilon_m$  может быть вычислена с помощью простой программы. Такая программа содержится в современных коммерческих пакетах программ и производит настройку внутренних параметров пакета под данную ЭВМ. Например, для БЗ-34  $\epsilon_m = 0,45 \cdot 10^{-7}$ , а для ЕС ЭВМ  $\epsilon_m$  колеблется вблизи величины  $5 \cdot 10^{-7}$  в зависимости от экземпляра ЭВМ (при обычной точности вычислений). Очевидно, если  $\epsilon_m > 10^{-k}$ , то на данной ЭВМ нельзя гарантировать, что в результатах будет содержаться не менее  $k$  верных значащих цифр. Напомним, что цифра числа называется верной, если абсолютная погрешность числа не превосходит половины единицы того разряда, в котором эта цифра находится. Два следующих примера показывают, что эквивалентные математические формулы могут быть неэквивалентными по точности вычислений.

**Пример 3.** Пусть  $\epsilon_m = 10^{-2}$  и требуется решить уравнение  $x^2 + 9,9x - 1 = 0$ . Очевидно, все результаты можно округлять до двух значащих цифр. Если применять для решения уравнения  $x^2 + px + q = 0$  формулу  $x_{1,2} = (-p \pm \sqrt{D})/2$ , где  $D^2 = p^2 - 4q$ , то получим  $\sqrt{D} \approx 10$ ,  $x_1 \approx 0,05$ ;  $x^2 \approx -9,95 \approx -10$ . Однако верный ответ для  $x_1$  будет  $x_1 \approx 0,1$ . Ошибка получена при вычитании близких чисел. Если же вычислять  $x_1$  по эквивалентной формуле  $x_{1,2} = -2q/(p \pm \sqrt{D})$ , то получаем  $x_1 \approx 2/(9,9 + 10) \approx 0,1$ . Правда, по этой формуле уже  $x_2$  будет вычислено с двукратной погрешностью.

**Пример 4.** Известно, что при статистических расчетах оценку дисперсии случайной величины по измерениям можно

вычислять по формуле  $\hat{\sigma}_2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2$ , где  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ , или

по эквивалентной формуле  $\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$ . Попробуем

оценить дисперсию выборки  $x_1 = 1234,1$ ,  $x_2 = 1234,2$ ,  $x_3 = 1234,3$ . Вычисляя на микрокалькуляторе БЗ-34, имеем

$\bar{x}=1234,2$ ,  $\bar{x^2}=1,5232496$ ;  $\frac{1}{3} \sum x_i^2=1,5232496$  и по первой формуле  $\hat{\sigma}^2=0$ . По второй формуле получаем результат:  $\hat{\sigma}^2=\frac{1}{3}(0,1^2+0^2+0,1^2)=0,67 \cdot 10^{-2}$ , который легко получить и устно.

**Правило 2.** При выборе формулы и порядка вычислений избегать вычитания близких чисел и деления на малые величины.

**Пример 5.** Найдем решения двух очень похожих систем из двух линейных уравнений [52].

Система 1:

$$\begin{cases} x+5y=17, \\ 1,5x+7,501y=25,5. \end{cases}$$

Решение:

$$\begin{cases} x=17. \\ y=0. \end{cases}$$

Система 2:

$$\begin{cases} x+5y=17, \\ 1,5x+7,499y=25,5. \end{cases}$$

Решение:

$$\begin{cases} x=32, \\ y=-3. \end{cases}$$

Столь большие отличия в ответах возникли из-за того, что матрицы коэффициентов систем  $A_i$ ,  $i=1, 2$ , плохо обусловлены: величины  $\Delta_i=\det A_i$  малы. Действительно,  $\Delta_1=0,001$ ,  $\Delta_2=-0,001$  ( $\Delta=a_{11}a_{22}-a_{12}a_{21}$ ).

**Правило 3.** Избегать плохо обусловленных матриц. Если это не удастся, использовать для работы с ними специальные методы [49, 50, 57].

**Пример 6.** Числа, представимые, например, в ЕС ЭВМ, лежат в диапазоне  $10^{-78} \leq |x| \leq 10^{76}$ . При выходе результата действия за левую границу диапазона выдается сообщение underflow (исчезновение порядка), при выходе за правую границу — overflow (переполнение). При переполнении обычно говорят, что плохи исходные данные, а при исчезновении порядка полагают результат равным нулю. Но не следует торопиться. Пусть, например, на ЕС ЭВМ вычисляется величина  $x=ab/(cd)$  при  $a=10^{-30}$ ,  $b=10^{-60}$ ,  $c=10^{-40}$ ,  $d=10^{-50}$ . Если выполнять действия в следующем порядке:  $x=a \cdot b/c/d$ , то компьютер сообщит об исчезновении порядка; если вычислить  $x=1/c/d \cdot a \cdot b$ , то получим переполнение. Если же вычислить, например,  $x=a/c \cdot b/d$ , то получим правильный ответ  $x=1$ . Этот же ответ можно получить, если отмасштабировать переменные, например, умножив на  $10^{40}$ .

**Правило 4.** При переполнении или исчезновении порядка следует попытаться изменить последовательность дей-

ствий, ввести масштабные множители и т. д. При исчезновении порядка не всегда следует обнулять результат.

**Пример 7.** Пусть снова  $\epsilon_m = 10^{-2}$  и требуется вычислить сумму [52]

$$S = 100 + \underbrace{0,1 + \dots + 0,1}_{2000 \text{ слагаемых}}$$

Если вести суммирование слева направо, то получим  $S = 100$ , поскольку с учетом округления до двух значащих цифр  $100 + 0,1 = 100$ . Если вычислять справа налево, то после сложения тысячи слагаемых получим 100, дальнейшее прибавление по 0,1 ничего не изменит и результат окажется:  $S = 200$ , что, конечно, уже ближе к истине. Но правильный результат  $S = 300$  можно получить, только если сложить 1000 чисел по 0,1, затем еще 1000 чисел по 0,1, а после этого сложить промежуточные суммы.

**Правило 5.** При сложении следует располагать слагаемые в порядке возрастания абсолютных величин, стараясь, чтобы при каждом сложении порядки величин различались мало. При необходимости цикл суммирования разбивается на несколько более коротких.

Аналогичное правило действует при перемножении большого числа сомножителей.

**Пример 8.** При расчетах методами последовательных приближений часто ведут вычисления до тех пор, пока поправка (разность между текущими и последующими приближениями) не станет меньше заданного порога. При этом, как правило, не обеспечивается заданная погрешность результата.

Пусть, например, требуется вычислить величину  $S = \sum_{k=1}^{\infty} \frac{1}{k^2}$

с точностью до  $10^{-3}$ . Если вести вычисления до тех пор, пока общий член ряда  $1/k^2$  не станет меньше  $10^{-3}$ , т. е. до  $k = 32$ , то получим  $S = 1,610$ , в то время как на самом деле  $S = \pi^2/6 = 1,650 \dots$ . Если же попытаться вычислить таким образом приближенную сумму ряда

$\sum_{k=1}^{\infty} \frac{1}{k}$ , то погрешность останется бесконечной, как бы мала ни становилась величина  $1/k$ , поскольку ряд расходится.

**Правило 6.** Нужно помнить, что остановка итерационного процесса  $x_1, x_2 \dots$  по косвенному критерию (например, по критерию  $|x_n - x_{n-1}| < \epsilon$  или  $|F(x_n)| < \epsilon$  в задаче реше-

ния уравнения  $F(x) = 0$ , по критерию  $\| \frac{\partial f}{\partial x}(x_n) \| < \varepsilon$  в задаче оптимизации  $f(x)$  и т. д.) *не гарантирует достижения заданной погрешности*  $|x_n - \lim_{n \rightarrow \infty} x_n| < \varepsilon$ .

**Пример 9** [50]. Пусть требуется вычислить интегралы

$$E_n = \int_0^1 x_n e^{x-1} dx, \quad n=1, 2, \dots, 10 \quad \text{при } \varepsilon_m = 10^{-7}.$$

Для величин  $E_n$  можно получить рекуррентное соотношение

$$E_n = 1 - nE_{n-1}. \quad (3.1)$$

Попытавшись вычислить интегралы рекуррентно, пользуясь (4.1), получим  $E_1 = 0,367879; \dots E_7 = 0,110160, E_8 = 0,118720, E_9 = -0,0684800$ , т. е. уже для  $E_9$  получен бессмысленный результат. Причина в том, что начальная ошибка округления быстро накапливается: при вычислении  $E_2$  она умножается на 2, затем на 3, 4, ..., 9. Попробуем переписать формулу (3.1) в другом виде:

$$E_{n-1} = (1 - E_n)/n \quad (3.2)$$

и будем вычислять в обратном порядке, начав с произвольного начального приближения для  $E_n, n < 10$ . Например, начав с  $E_{20} = 0$ , получим:

$E_{19} = 0,0500000,$	$E_{14} = 0,0627322,$
$E_{18} = 0,0500000,$	$E_{13} = 0,0669477,$
$E_{17} = 0,0527778,$	$E_{12} = 0,0717733,$
$E_{16} = 0,0557190,$	$E_{11} = 0,0773523,$
$E_{15} = 0,0590176,$	$E_{10} = 0,0838771,$
	$E_9 = 0,0916123.$

После первого шага начальная ошибка уменьшится в 20 раз, после второго — еще в 19 раз, и уже для  $E_9$  мы получим все шесть значащих цифр верных. Все дело в том, что формула (3.2), в отличие от (3.1), определяет устойчивый вычислительный процесс: погрешность результата каждого шага меньше погрешности исходных данных. Подробнее об устойчивости численных методов см., например, [30, 49, 50].

**Правило 7.** *Пользуйтесь только устойчивыми численными алгоритмами!*

MATLAB — это высокоэффективное программное обеспечение для научных и инженерных расчетов, работающее на всех основных типах ЭВМ, включая IBM-PC.

Работая в среде MATLAB, пользователь, даже не имея предварительной программистской подготовки, получает возможность быстро и легко решать вычислительные задачи в областях линейной алгебры, теории управления, теории систем, обработки сигналов и ряда других дисциплин. Кроме этого, в настоящее время MATLAB фактически стал международным стандартом учебного программного обеспечения в перечисленных отраслях знания.

Одно из основных достоинств системы MATLAB — это сочетание высококачественных численных процедур (из пакетов LINPACK и EISPACK) и простой командной среды, позволяющей вводить выражения в их естественной математической записи, без использования каких-либо изощренных приемов программирования.

Возможности системы MATLAB значительно расширяются при использовании дополнительных пакетов программ, так называемых *тулбоксов* (TOOLBOX — ящик с инструментами).

Тулбоксы представляют собой проблемно-ориентированный набор функций, предоставляющий дополнительные возможности для решения задач, характерных для конкретной научной отрасли. Основными тулбоксами являются:

- «Системы управления» (Control System);
  - «Робастное управление» (Robust Control);
  - «Оптимизация» (Optimization);
  - «Идентификация систем» (System Identification)
- (см. [77]);
- «Сплайны» (Spline);
  - «Обработка сигналов» (Signal Processing);
  - «Нейронные сети» (Neural Network).

Примечание. В дальнейшем изложении функции собственно среды MATLAB, в отличие от функций тулбоксов, будут иногда именоваться базовыми функциями.

Надо отметить, что деятельность, направленная на разработку новых тулбоксов, не прекращается как фирмой-разработчиком — MathWorks, Inc., так и другими группами пользователей, что ведет к постепенному пополнению набора существующих тулбоксов и расширению возможностей и области применения системы MATLAB.

Версия среды MATLAB, предназначенная для работы на IBM-PC и других совместимых с MS-DOS современных персональных компьютерах, называется PC-MATLAB, и на указанную версию ориентированы все приводимые ниже примеры (которые подходят также для AT-MATLAB).

## 1. БАЗОВЫЕ ВОЗМОЖНОСТИ СИСТЕМЫ MATLAB

### 1.1. Начало работы

Если пакет программ MATLAB должным образом установлен на жестком диске Вашего компьютера [85], то для того, чтобы его активизировать, надо в одноименном каталоге, в поддиректории BIN (или в любой, если указан путь в файле AUTOEXEC.BAT) выполнить команду

MATLAB,

которая запускает командный файл MATLAB.BAT, а он, в свою очередь — выполняемый файл PCMATLAB.EXE. После запуска на экране появится заголовок и приглашение типа «>>» или «п», указывающее, что система ожидает ввода команд от пользователя.

Ввод каждой команды должен заканчиваться нажатием клавиши Enter (возврат каретки). С помощью клавиш перемещений курсора можно передвигаться по последовательности введенных команд: клавиша «курсор вверх» (↑) высвечивает на экране предыдущую команду; клавиша «курсор вниз» (↓) показывает последующую команду, если перед этим была нажата клавиша «курсор вверх» и текущая команда не последняя.

MATLAB имеет функцию оперативной подсказки — HELP. Для ее запуска надо ввести команду

help,

которая выведет на экран необходимую информацию о работе в среде MATLAB (на английском языке).

## 1.2. Действия с матрицами

### 1.2.1. Ввод и простейшие операции

Все действия MATLAB выполняет, по существу, только с одним типом объектов — прямоугольными матрицами (само название MATLAB есть сокращение английского словосочетания *matrix laboratory* — лаборатория матриц).

Элементами матриц могут быть целые, вещественные или комплексные числа; скалярным значениям соответствуют матрицы размера  $1 \times 1$ , а векторам — матрицы с одним столбцом или строкой.

Важная особенность: при определении переменных не требуется указания размерностей, что дает большую свободу обращения с данными.

С примеров выполнения операций с матрицами мы и начнем иллюстрацию возможностей языка MATLAB.

Допустим, нам требуется выполнить умножение двух

матриц, например,  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$ ,  $B = \begin{bmatrix} 7 & 8 & 1 \\ 9 & 0 & 2 \end{bmatrix}$ .

Рассмотрим последовательность необходимых для этого действий. После активизации среды и появления приглашения к работе надо сначала задать матрицы A и B. Для этого необходимо ввести две команды, например:

```
A=[1 2 ; 3 4 ; 5 0];  
B=[7 8 1 ; 9 0 2 ];
```

Теперь можно выполнить умножение введенных матриц:

```
C=A * B
```

После ввода последней команды на экране появится результат:

```
C=  
25 8 5  
57 24 11  
35 40 5
```

Заметим, что в конце первых двух команд стоит знак «;» (точка с запятой). Так следует заканчивать при последовательном вводе те команды присваивания, результаты действия которых не требуется немедленно по их выполнении отображать на экране. После последнего присваивания сим-

вол «;» не стоит, что позволяет сразу получить на экране результат умножения. Точкой с запятой также отделяются строки вводимых матриц; элементы внутри строки разделяются пробелами.

Теперь все три матрицы А, В и С сохраняются в памяти и доступны для дальнейшего использования. Например, последнюю матрицу можно транспонировать:

$$D=C'$$

В результате на экране появится ответ:

$$D = \begin{matrix} 25 & 57 & 35 \\ 8 & 24 & 40 \\ 5 & 11 & 5 \end{matrix}$$

Можно получить собственные числа с помощью функции eig:

$$v = \text{eig}(C),$$

которая даст в результате вектор v, содержащий искомые собственные числа:

$$v = \begin{matrix} 57.9031 \\ 0.0000 \\ -3.9031 \end{matrix}$$

Чтобы вычислить определитель матрицы, достаточно ввести строку

$$d = \text{det}(C)$$

и получить ответ:

$$d = 0$$

MATLAB позволяет так же просто выполнять практически все возможные действия и операции с матрицами и векторами: сложение, вычитание, деление, возведение в степень, взятие логарифма, обращение, вычисление ранга и т. д. Полностью списки операций с матрицами и матричных функций приведены в прил. 1. Смысл всех операций очевиден и они имеют естественную форму записи, поэтому здесь мы остановимся только на одной из них, требующей комментария, — операции деления матриц.

### 1.2.2. Деление матриц

Вследствие некоммутативности в общем случае матричного умножения в среде MATLAB имеются две разновидности операции деления матриц: правое деление и левое, обозначаемые символами «/» и «\» соответственно. Если, например,  $D$  — невырожденная квадратная матрица, тогда

$X=B/D$  есть решение матричного уравнения  $X*D=B$ .  
 $X=D\backslash B$  соответствует системе уравнений  $D*X=B$ .

Если решения не существует, т. е. система уравнений переопределена, то в качестве ответа берется наилучшее возможное решение в смысле наименьших квадратов. Если решение неединственно, т. е. система недоопределена, тогда берется решение в смысле метода наименьших квадратов, имеющее минимальную длину.

Необходимым условием успешного выполнения всех операций над матрицами является соответствие размерностей матричных операндов известным правилам из линейной алгебры. В противном случае будет выдаваться сообщение об ошибке.

### 1.2.3. Модификация матриц

В языке MATLAB существует возможность добавления и удаления строк и столбцов матрицы. Например, введем вектор

```
b=[7 8];
```

и добавим его как строку к матрице  $A$  из предыдущего примера:

```
A=[A; b]
```

В результате получим

```
A=  
1 2  
3 4  
5 0  
7 8
```

Если же после этого введем команду

```
A=A(1:3, :),
```

то вернемся обратно к исходной матрице A:

A=

```
1 2
3 4
5 0
```

Можно выбрать один элемент

A(3, 1)

и получить на экране ответ

```
ans =
     5
```

Переменная ans генерируется автоматически, если вводится выражение без знака присваивания.

MATLAB допускает сложные выражения и использование скобок. При этом действуют правила приоритетов, аналогичные известному языку программирования фортран.

### 1.3. Условные операторы и циклы

Язык MATLAB имеет свои операторы управляющей логики: операторы циклов FOR и WHILE и оператор условного перехода IF. Синтаксис и выполняемые действия этих операторов аналогичны соответствующим операторам во многих распространенных языках программирования, таких, например, как Бейсик и Паскаль. Область действия каждого оператора управляющей логики должна ограничиваться словом end. MATLAB допускает вложенность циклов и возможность досрочного выхода из них с помощью команды break.

Для иллюстрации приведем следующие программные фрагменты. Создание вектора нечетных чисел с использованием цикла FOR:

```
for i=1:n
    x(i)=2*i-1;
end
```

то же самое действие с использованием цикла WHILE:

```
i=1;
while i <=n
    x(i)=2*i-1;
    i=i+1;
end
```

и совсем уж искусственный вариант той же операции:

```
i=1;
while 1
  if i>n
    break
  else
    x(i)=2*i-1;
    i=i+1;
  end
end
```

## 1.4. Функции и М-файлы

Выше уже упоминалось о некоторых функциях языка (eig, det), выполняющих действия с матрицами. Всего MATLAB содержит около 200 базовых функций, полный перечень которых по соответствующим разделам приведен в прил. 1. Некоторые из функций «встроены» в среду MATLAB, и их тексты не доступны для пользователей. Остальные же функции являются доступными и составляют библиотеку так называемых функциональных М-файлов, находящихся в поддиректории \MATLAB.

М-файл представляет собой обычный текстовый файл, содержащий набор операторов (программу) языка MATLAB. Если М-файл содержит текст функции, то его имя должно совпадать с именем этой функции с добавлением расширения «.m». Например, функция, вычисляющая след матрицы, trace содержится в М-файле с именем trace.m.

Одним из важных достоинств среды MATLAB является ее открытость. Пользователь может добавлять к имеющемуся набору функций среды свои собственные функции для решения своих конкретных задач посредством создания новых функциональных М-файлов.

### 1.4.1. Функциональные М-файлы

Первая строка М-файлов функций должна содержать обязательное слово «function» (функция), которое указывает системе, что это функциональный М-файл. Затем уже должно идти тело функции, т. е. программа реализуемого ею алгоритма, написанная на языке MATLAB.

Рассмотрим для примера одну из базовых функций языка MATLAB с именем `mean`, вычисляющую среднее значение. Вот как выглядит соответствующий файл `mean.m`:

```
function y=mean(x)
% MEAN Average or mean value. For vectors,
% MEAN(x) returns the mean value. For
% matrices, MEAN(x) is a row vector
% containing the mean value of each column.
[m, n]=size(x);
if m==1
    m=n; % Handle isolated row vector.
end
y=sum(x)/m;
```

Разберем построчно этот текст.

Первая строка после ключевого слова `function` содержит объявление функции: ее имя, список входных аргументов в круглых скобках (в нашем случае один аргумент `x`) и список выходных аргументов перед знаком присваивания (в данной функции один аргумент `y`). Если функция возвращает несколько аргументов, то их список заключается в квадратные скобки и аргументы отделяются запятыми, например:

```
function [mean, stdev]=stat(x)
```

Следующие четыре строки начинаются с символа `%`, который вводит строчный комментарий. Все последующее за символом `%` содержание строки игнорируется, т. е. не влияет на выполнение программы.

В шестой строке вводятся две переменные `m` и `n`, значения которых возвращает функция вычисления длины вектора или размера матрицы `size`. Так как функция `size` возвращает два значения, переменные в левой части присваивания стоят в квадратных скобках.

Далее следует условный оператор `IF`, обрабатывающий вариант: если `x` есть  $1 \times n$ -матрица. Обратите внимание на знак операции сравнения «`==`» в логическом условии оператора `IF`. Довольно часто начинающие программисты ставят вместо него знак присваивания «`=`», что является ошибкой.

Завершает тело функции вычисление значения возвращаемого аргумента `y`.

Переменные `m`, `n` и `y` являются локальными для функции `mean`. Это означает, что их область действия ограничива-

ется телом функции и они не существуют после ее завершения.

### 1.4.2. Скрипт-файлы

M-файл может также состоять просто из фрагмента программы, т. е. из некоторой последовательности операторов языка MATLAB, не выделенной как отдельная функция. Такие файлы называются скрипт-файлами (script files). В отличие от функциональных файлов они не должны содержать слова «function». Все переменные, объявленные в скрипт-файле, становятся глобальными для программы, в которую он включен.

Выделение части программы в M-файл бывает оправдано, если эта часть состоит из большого числа операторов, используется в программе многократно или ее присутствие в явном виде в тексте программы затрудняет чтение и работу с последней.

### 1.5. Графика

Проиллюстрируем графические возможности языка MATLAB на нескольких простых примерах.

Если требуется построить график какой-либо функции, например  $y = \cos(x)$ , то для этого нужно написать следующую последовательность операторов:

```
x=0:.05:6*pi; % Создаем вектор x с элементами от 0
                % до 6*pi с шагом 0.05.
y=cos(x);      % y-вектор, элементы которого есть
plot(x,y)      % значения функции cos.
```

В результате на экране увидим график желаемой зависимости (рис. III.1).

Как можно видеть из примера, графический вывод осуществляется с помощью функции plot. Команда plot(x, y) строит в декартовых координатах график зависимости y от x, причем y и x — векторы одинакового размера. (Аргументами функции plot могут быть также матрицы — см. подсказку help или User's Guide) Масштабирование графика производится автоматически.

На одном графике может быть построено несколько кривых. В этом случае каждая зависимость должна задаваться парой входных аргументов функции plot:

```

t=0:.05:6*pi;
y1=cos(t);
y2=sin(t);
plot(t, y1, t, y2)
grid

```

Результатом будет график двух функций (рис. III.2).

MATLAB позволяет задавать цвет и тип кривых на графике, отличные от установленных по умолчанию. Например, команда

```
plot(X, Y, '*r')
```

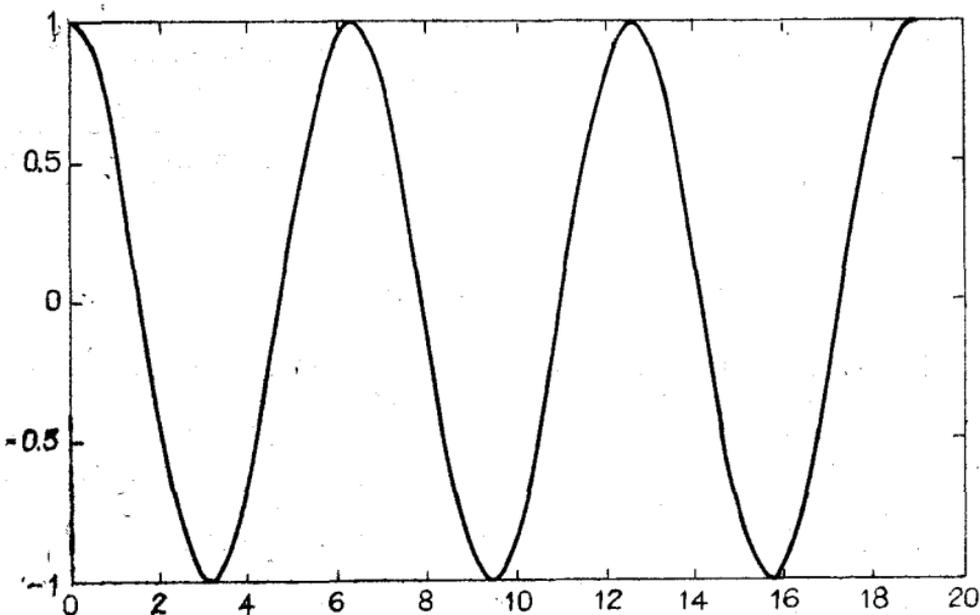


Рис. III.1

нарисует кривую зависимости  $Y$  от  $X$  символами '\*' красного цвета. Перечень возможных линий, символов и цветов графического вывода приведен в прил. I.

Помимо декартовых координат имеются функции, которые позволяют строить графики в полярных координатах и в логарифмическом масштабе, трехмерные поверхности и линии уровня.

Для вывода графиков в логарифмическом и полулогарифмическом масштабах имеются следующие функции:

$\text{loglog}(X, Y)$  — график с логарифмическим масштабом по обоим осям;

`semilogx(X, Y)` — график с логарифмической осью  $x$  и линейной  $y$ ;

`semilogy(X, Y)` — график с логарифмической осью  $y$  и линейной  $x$ .

Информацию об остальных графических функциях можно получить из контекстной подсказки `help` или демонстрационной программы `demo`.

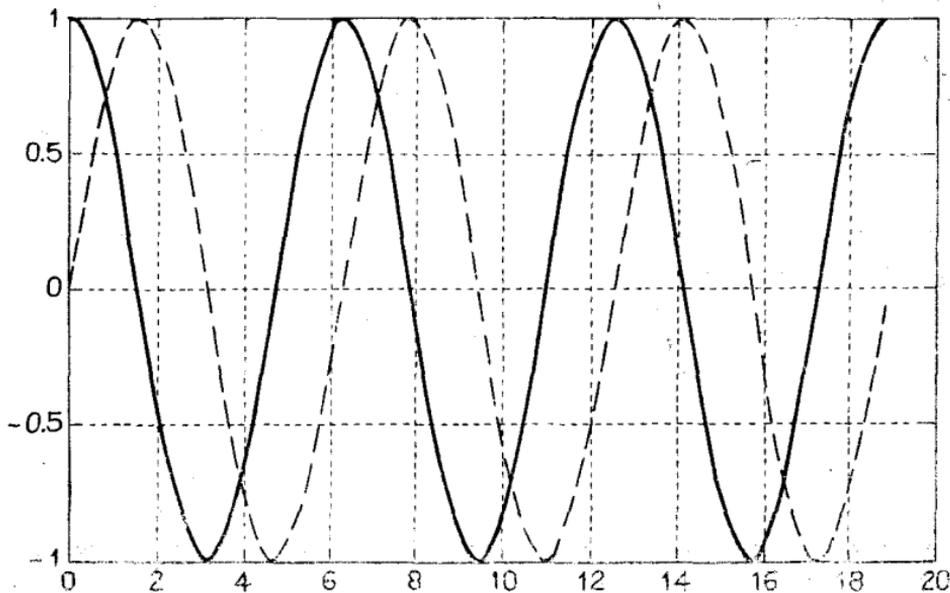


Рис. III. 2

MATLAB предоставляет возможность разбиения графического экрана на составные части, или окна, для одновременного вывода нескольких графиков. Оператор

`subplot(mnp)`

разбивает экран сеткой  $m \times p$  и использует  $p$ -е окно для следующего графического вывода. Возможны также обозначения осей и вывод заголовка графика (см. программы `help`, `demo`, или [85]). Список всех графических функций имеется в прил. 1.

## 2. РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ АЛГЕБРЫ

Как уже отмечалось в части II, ряд задач линейной алгебры и задачи анализа статики линейных систем требуют решения системы алгебраических уравнений

$$Ax=b, \quad (2.1)$$

где  $A$  — квадратная невырожденная матрица,  $b$  — заданный вектор,  $x$  — вектор неизвестных.

Обычно в основе численных методов решения матричного уравнения (2.1) лежат разложения матрицы  $A$  на множители, которые представляют собой матрицы простой структуры (см. ч. II, подразд. 1.3). В языке MATLAB имеется ряд функций, выполняющих такие разложения. Рассмотрим одну из них — функцию  $\text{lu}(A)$ , реализующую так называемое треугольное разложение, или, как его еще иначе называют, LU-факторизация, от слов Low-Upper — нижний-верхний. В результате этого разложения исходная матрица  $A$  выражается в виде произведения двух матриц, одна из которых является результатом умножения нижней треугольной матрицы на матрицу перестановки, а другая есть верхняя треугольная матрица. Рассмотрим пример.

Допустим, у нас задана матрица

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix}$$

и нам требуется разложить ее на «треугольные» множители. Для этого необходимо ввести следующий оператор:

$$[L, U] = \text{lu}(A)$$

В результате получим искомые матричные множители:

$$L = \begin{pmatrix} 0.1429 & 1.0000 & 0 \\ 0.5714 & 0.5000 & 1.0000 \\ 1.0000 & 0 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} 7.0000 & 8.0000 & 0 \\ 0 & 0.8571 & 3.0000 \\ 0 & 0 & 4.5000 \end{pmatrix}$$

Матрица  $L$  представляет собой нижнюю треугольную матрицу с переставленными столбцами, а  $U$  — верхнюю. При этом выполняется равенство

$$A = L * U,$$

что мы можем легко проверить.

Можно также использовать три переменные в левой части оператора присваивания с функцией lu:

$$[L, U, P]=lu(A)$$

В этом случае результатом у нас будут та же матрица U, что в предыдущем случае, нижняя треугольная матрица L и матрица перестановки P:

$$L = \begin{matrix} 1.0000 & & 0 & & 0 \\ 0.1429 & & 1.0000 & & 0 \\ 0.5714 & & 0.5000 & & 1.0000 \end{matrix}$$

$$P = \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$$

Полученные матрицы удовлетворяют равенству

$$L*U=P*A.$$

Функция разложения lu является базовой для таких функций языка MATLAB в области линейной алгебры, как:

— функция обращения матрицы inv. Операция нахождения обратной матрицы

$$X=inv(A)$$

на самом деле вычисляется как произведение треугольных множителей

$$X=inv(U)*inv(L);$$

— функция вычисления определителя det

$$d=det(A)$$

использует также множители треугольного разложения:

$$d=det(L)*det(U);$$

— наконец, решение уравнения  $Ax=b$ , получаемое при помощи операции матричного деления:

$$x=A \setminus b$$

вычисляется посредством решения двух систем с треугольными матрицами:

$$Y=L \setminus b, \quad x=U \setminus y$$

Остальные действия с матрицами в языке MATLAB описаны в прил. 1.

### 3. РЕШЕНИЕ ЗАДАЧ ЧИСЛЕННОГО АНАЛИЗА

#### 3.1. Решение систем нелинейных алгебраических уравнений

Рассмотрим следующую систему уравнений:

$$\cos(x) + y^2 + 2 \ln(z) = 10,$$

$$7x + 2y - z^{-4} = -3,$$

$$x + y + z = 7.$$

Чтобы решить эту систему в среде MATLAB, мы должны выполнить указанную ниже последовательность действий.

1. Привести исходную систему к такому виду, чтобы в правой части каждого уравнения стоял ноль:

$$\cos(x) + y^2 + 2 \ln(z) - 10 = 0,$$

$$7x + 2y - z^{-4} + 3 = 0,$$

$$x + y + z - 7 = 0.$$

2. Создать М-файл (см. подразд. 4.1), функция которого должна принимать в качестве входного параметра вектор переменных системы, в данном случае  $x$ ,  $y$ ,  $z$ , а возвращать вектор невязок уравнений. Для нашей системы М-файл функции должен выглядеть примерно так:

```
function q=sfun(p)
x=p(1); y=p(2); z=p(3); q=zeros(3, 1);
q(1)=cos(x)+y^2+sqrt(2)*log(z)-10;
q(2)=7*x+2*y-z^(-4)+3;
q(3)=x+y+z-7;
```

3. Для получения решения системы выполнить команду

$$x=\text{fsolve}(\text{'sfun'}, [1 \ 1 \ 1]')$$

В результате получим вектор решений

```
x=
  3.0163
  3.5755
  0.4081
```

Функция решения системы нелинейных алгебраических уравнений `fsolve` имеет входными параметрами имя М-файла ('`sfun`') и вектор начальных оценок (`[1 1 1]`), начиная от которых будет производиться решение. Алгоритм вычисления, реализуемый функцией `fsolve`, представляет собой модифицированный метод Ньютона (см. ч. II, подразд. 1.1.2). Более подробную информацию о функции `fsolve` можно получить из контекстной подсказки `help` или из [85].

### 3.2. Решение обыкновенных дифференциальных уравнений

Покажем последовательность решения задачи интегрирования обыкновенных дифференциальных уравнений на следующем примере.

Допустим, требуется проинтегрировать дифференциальное уравнение 2-го порядка (известное как уравнение Ван дер Поля):

$$\ddot{x} + (x^2 - 1)\dot{x} + x = 0.$$

Решение включает в себя три этапа.

1. Приведение исходного уравнения к системе уравнений первого порядка:

$$\dot{x}_1 = x_1(1 - x_2^2) - x_2,$$

$$\dot{x}_2 = x_1.$$

2. Создание М-файла, содержащего функцию, входными параметрами которой должны быть скаляр  $t$  (независимая переменная — обычно время) и вектор столбец  $x$  (вектор состояния), а возвращаемым параметром — вектор производных состояния  $\dot{x}$ :

```
function xdot=vdpol(t, x)
    xdot(1)=x(1).*(1-x(2).^2)-x(2);
    xdot(2)=x(1);
```

Примечание. Точки перед знаками операций «`*`» и «`^`» указывают, что это поэлементные операции, в отличие от соответствующих матричных операций. Например:

```
x=[1 2 3]; y=[4 5 6];
gm=x*y' % Матричное умножение
gm=
```

```
32
```

```
ge=x.*y % Поэлементное умножение
```

re=

4 10 18

3. Интегрирование уравнений с использованием функции `ode23` или `ode45`:

```
t0=0; tf=25; % Задание интервала  $0 \leq t \leq 25$  и  
x0=[-2 1.5]; % вектора начальных значений  $x_0$ .  
[t, x]=ode23('vdpol', t0, tf, x0);  
plot(t, x)
```

В результате будем иметь графики искомых функций  $x_1(t)$  и  $x_2(t)$  (рис. II.3).

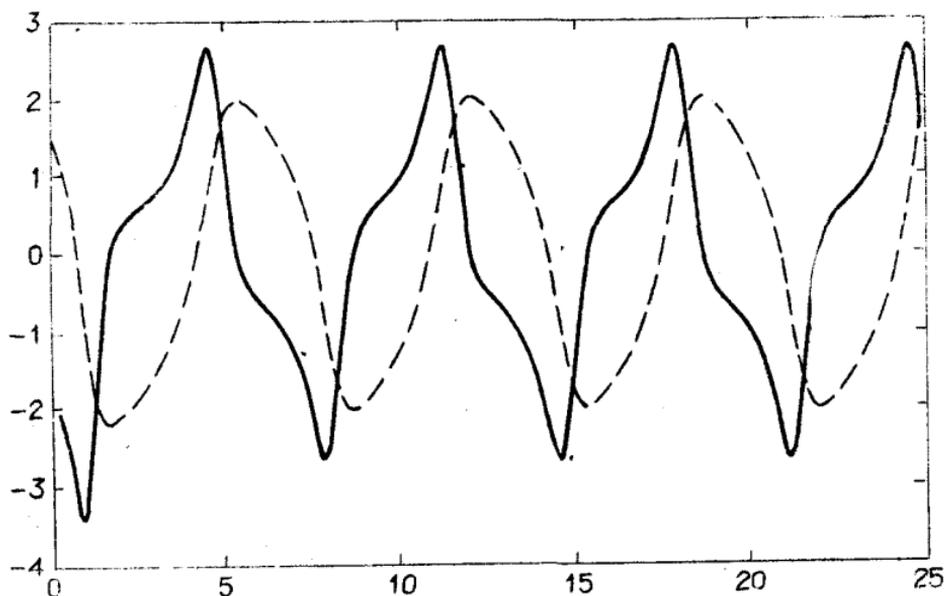


Рис. II.3

Функции `ode23` и `ode45` используют метод Рунге-Кутты 2-го — 3-го и 4-го — 5-го порядка соответственно, с автоматическим выбором шага. Точность вычисления по умолчанию устанавливается равной  $10^{-3}$  и при необходимости может быть изменена добавлением необязательного входного параметра:

```
[t, x]=ode23('vdpol' t0, tf, x0, 1e-4);
```

Для получения более подробной информации о функциях интегрирования дифференциальных уравнений `ode23` и `ode45` используйте оперативную подсказку `help` или см. [85].

## 4. РЕШЕНИЕ ЗАДАЧ ТЕОРИИ СИСТЕМ

### 4.1. Расчет частотных и временных характеристик систем с обратной связью

Для решения задач, связанных с разработкой и анализом систем управления, в среде MATLAB используется дополнительный набор специальных функций, или тулбокс, который соответственно и называется — «Системы управления» (Control System Toolbox).

Тулбокс «Системы управления» (СУ) представляет собой библиотеку алгоритмов, содержащихся в функциональных М-файлах и реализующих наиболее общие методы расчета, анализа и построения (моделирования) систем управления.

Чтобы иметь возможность использовать функции тулбокса, необходимо в командном файле matlab.bat, находящемся в директории \MATLAB\BIN, указать имя пути к каталогу тулбокса СУ — \MATLAB\CONTROL в команде SET MATLABPATH [85] (если, конечно, тулбокс установлен). Строка команды должна выглядеть после этого примерно так:

```
SET MATLABPATH=\MATLAB\MATLAB ;  
\MATLAB\DEMO; MATLAB\CONTROL
```

Приведем пример анализа системы управления в среде MATLAB с использованием функций тулбокса СУ. Рассмотрим простую систему, реализующую управление угловой скоростью вращения ротора двигателя при условии действия постоянного по модулю возмущения. Объект управления описывается дифференциальным уравнением 1-го порядка (обозначение зависимости по времени  $t$  опускаем):

$$\dot{\omega}_x = -\frac{1}{T} \omega_x + \frac{k_0}{T} u_x + f, \quad (4.1)$$

где  $\omega_x$  — угловая скорость вращения ротора;  $T$  — постоянная времени двигателя;  $k_0$  — коэффициент передачи двигателя по скорости;  $u_x$  — сигнал управления;  $f$  — возмущение. В качестве алгоритма управления выберем П-регулятор:

$$u_x = k_{\text{п}} x \quad (4.2)$$

$$x = g - \omega_x,$$

где  $g$  — задающее воздействие,  $k_{\text{п}}$  — коэффициент регулятора.

Структурная схема системы выглядит следующим образом (рис. III 4):

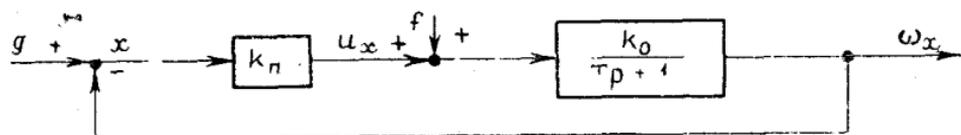


Рис. III. 4

Поставим задачу получить:

- 1) амплитудно-частотную, фазо-частотную и амплитудно-фазовую характеристики;
- 2) переходную характеристику;
- 3) реакцию системы (переходный процесс) на произвольное задающее воздействие.

Каждому пункту поставленной задачи соответствует одна функция тулбокса СУ, поэтому для решения всей задачи нам потребуются всего три функции.

1. Функция расчета амплитудно-фазовой характеристики:

$$[\text{mag, phase}] = \text{bode}(a, b, c, d, \text{iu}, w)$$

Функция принимает в качестве входных параметров матрицы системы  $a, b, c, d$ , номер входа, для которого производится расчет  $\text{iu}$ , и вектор диапазона частот  $w$ , формируемый для построения логарифмической частотной характеристики с помощью функции `logspace` (см. ниже текст программы). Возвращает функция `bode` две матрицы: `mag` (magnitude — амплитуда) и `phase` (фаза). У этих матриц количество столбцов совпадает с размером вектора выхода системы, а число строк — с числом расчетных точек в интервале частот, т. е. с размером вектора  $w$ .

2. Функция расчета переходной характеристики (реакции на единичный скачок задающего воздействия) для  $\text{iu}$ -го входа:

$$Y = \text{step}(a, b, c, d, u, \text{iu}, t).$$

Функция `step` возвращает матрицу  $Y$ , число столбцов, у которой равно размеру вектора выхода системы, а число строк — количеству элементов в векторе временного интервала  $t$ .

3. Функция расчета реакции системы на произвольное задающее воздействие:

$$Y = \text{lsim}(a, b, c, d, u, t, x0)$$

Функция `lsim` возвращает матрицу  $Y$ , размерность которой определяется так же, как у функции `step`, которая содержит таблицу значений реакции системы на предысторию входного воздействия, хранящуюся в матрице  $u$ . Количество столбцов матрицы  $u$  соответствует числу входов системы  $u_i$ , а каждая строка — это состояние вектора входа в момент временного интервала  $t$ .  $x_0$  есть вектор начальных значений состояния системы.

Для всех перечисленных функций входными параметрами являются матрицы  $a$ ,  $b$ ,  $c$  и  $d$  разомкнутой системы вида

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du. \end{aligned} \quad (4.3)$$

Поэтому, чтобы пользоваться приведенными функциями для анализа замкнутых систем, подобных нашей, надо найти аналогичные матрицы, но соответствующие системе, замкнутой обратной связью и регулятором

$$u = -Kx + Ng, \quad (4.4)$$

где  $x$  — вектор состояния,  $g$  — задающее воздействие. Подставляя (4.4) в (4.3), получим

$$\begin{aligned} \dot{x} &= Ax - BKx + BNg = (A - BK)x + BNg \\ y &= Cx - DKx + DNg = (C - DK)x + DNg \end{aligned}$$

Создадим теперь `M-файл` с функцией, формирующей матрицы замкнутой системы:

```
function [aa, bb, cc, dd]=feedback(a, b, c, d, k, n)
aa=a - b*k;
bb=d*n;
cc=c - d*k;
dd=d*n;
```

Для заданной системы (4.1) — (4.2) матрицы, соответствующие разомкнутой системе вида (4.2) и регулятору (4.4), выглядят следующим образом:

$$\begin{aligned} A &= \left[ -\frac{1}{T} \right]; & B &= \left[ \frac{k_0}{T} \quad 1 \right]; \\ C &= [1]; & D &= [0 \quad 0]; \\ K &= \begin{bmatrix} k_n \\ 0 \end{bmatrix}; & N &= \begin{bmatrix} k_n & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

при этом

$$x = [\omega_x]; \quad u = [u_x \ f]^T;$$

$$y = [\omega_x]; \quad g = [g_x \ f]^T.$$

Теперь, для расчета результатов, осталось ввести следующую программу

% Ввод численных значений параметров системы:

$$T=0.5; K_0=1.0; K_p=2.5;$$

% Формирование матриц ОУ и регулятора:

$$A=[-1/T]; B=[K_0/T \ 1];$$

$$C=[1]; D=[0 \ 0];$$

$$K=[K_p; 0]; N=[K_p \ 0; 0 \ 1];$$

% Получение матриц замкнутой системы a, b, c и d:

$$[a, b, c, d]=\text{feedback}(A, B, C, D, K, N);$$

% Формирования вектора диапазона частот w логарифмического масштаба от 0.1 до 10000 для построения ЛАХ и ЛФЧХ

$$w=\text{logspace}(-1, 4);$$

% Задание вектора временного интервала  $0 \leq t \leq 10$  с шагом 0.1

$$t=0:.1:10;$$

% Формирование задающего воздействия и возмущения

$$g=\sin(t); f=\text{sign}(g);$$

% Формирование вектора входного воздействия

$$u=[g; f];$$

% Ввод начального значения вектора скорости  $\omega_x$

$$x_0=0.2;$$

% На этом формирование исходных данных заканчивается и далее % идет уже непосредственное решение задачи:

% 1. Расчет АЧХ

$$[\text{mag}, \text{phase}]=\text{bode}(a, d, c, b, l, w);$$

% 2. Расчет переходной хар-ки

$ys = \text{step}(a, b, c, d, 1, t);$

% 3. Расчет реакции на синусоидальное входное воздействие

$y = \text{lsim}(a, b, c, d, u, t, x0);$

На этом расчет задачи закончен. Мы имеем массивы, содержащие все значения, необходимые нам для вывода результатов. Получим на одном экране графики амплитудно-фазовых и переходной характеристик:

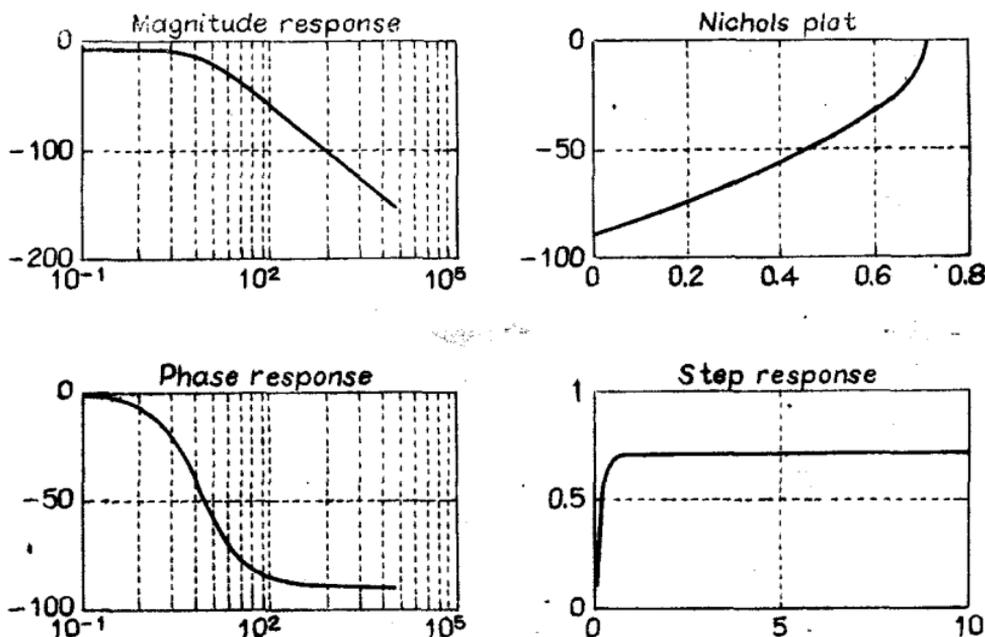


Рис. III. 5

```
subplot(221), semilogx(w, 20*log(mag))  
title('Magnitude response'), grid  
subplot(223), semilogx(w, phase)  
title('Phase response'), grid  
subplot(222), plot(mag, phase)  
title('Nichols plot'), grid  
subplot(224), plot(t, ys)  
title('Step response'), grid
```

В результате на экране увидим четыре графика (рис. III. 5). Для возвращения обратно в текстовый командный режим

надо нажать любую клавишу. Выведем теперь графики переходных процессов при задающем синусоидальном воздействии:

```

clg      % Стираем графический экран
plot(t, y, t, g), title('Sine response'), grid
    
```

На одном графике получим кривые задающего воздействия и выходного сигнала (рис. III. 6).

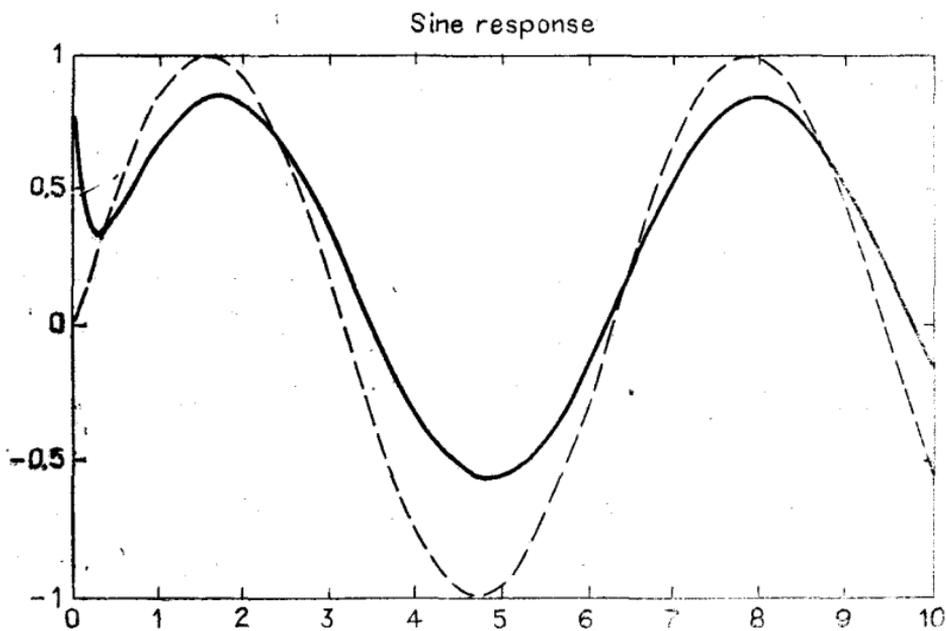


Рис. III. 6

## 4.2. Преобразование математических моделей

Большой раздел функций тулбокса «Системы управления» посвящен задачам преобразования различных описаний динамических систем. В части I пособия уже были рассмотрены взаимные связи между математическими моделями: дискретизация, континуализация и переходы от уравнений состояния (УС) к передаточной функции (ПФ) системы и обратно (см. ч. I, подразд. 2.2). Все указанные преобразования можно легко проделать в среде MATLAB с помощью соответствующих функций тулбокса СУ. Например, рассмотрим систему управления, заданную управлениями состояния:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -2x_1 + x_2 + u, \quad y = x_1 + 5x_2. \end{cases} \quad (4.5)$$

Допустим, надо найти ПФ данной системы. Для этого из уравнений (4.5) найдем матрицы системы A, B, C и D:

$$A=[0 \ 1; -2 \ 1]; \ B=[0; \ 1];$$

$$C=[1 \ 5]; \ D=[0];$$

Теперь для определения вида ПФ достаточно выполнить команду

$$[\text{num}, \text{den}]=\text{ss2tf}(A, B, C, D, 1).$$

Функция `ss2tf` осуществляет переход от УС к ПФ системы. Название `ss2tf` является своеобразным сокращением фразы: `state-space to` («to» совпадает с произношением слова «two» — 2) `transfer function`, т. е. пространство состояний к передаточной функции. Входными параметрами для функции являются матрицы системы A, B, C, D и номер входа, относительно которого будет рассчитываться ПФ. Возвращает функция вектор `den`, содержащий коэффициенты знаменателя ПФ в порядке убывания степени оператора дифференцирования `p`, и матрицу `num`, в которой возвращаются коэффициенты числителя и которая имеет столько строк, сколько выходов у системы (4.5).

В результате выполнения введенной команды получим искомую ПФ:

$$\text{num} =$$

0	1	5
---	---	---

$$\text{den} =$$

1	-1	2
---	----	---

Аналогично можно выполнить обратный переход:

$$[A, B, C, D]=\text{tf2ss}(\text{num}, \text{den})$$

Всего в тулбоксе `CV` имеется 9 функций преобразований систем, список которых вместе со всеми функциями тулбокса (всего более 50) приведен в прил. 2.

### 4.3. Вычисление матричной экспоненты

В задачах анализа динамики систем важную роль играет вычисление экспоненциальной матрицы  $e^A$ . В среде `MATLAB` экспонента матрицы A вычисляется с помощью функции `expm(A)`. Например, пусть

$$A=[0 \ 1; -2 \ 1];$$

Тогда оператор

$$B = \expm(A)$$

даст результат

$$B = \begin{array}{cc} -0.1995 & 1.2082 \\ -2.4164 & 1.0087 \end{array}$$

Функция `expm` является «встроенной» (т. е. ее текст не доступен пользователю), но она использует «доступный» алгоритм аппроксимации матрицы  $e^A$  рядом Падэ, который находится в файле `expm1.m`. Кроме разложения по Падэ, MATLAB содержит еще два алгоритма вычисления матричной экспоненты: это разложение в ряд Тейлора и алгоритм приведения матрицы к диагональному виду с помощью функции вычисления собственных значений. Тексты этих алгоритмов находятся, соответственно, в файлах `expm2.m` и `expm3.m`.

#### 4.4. Оценка параметров модели

Задача выбора параметров математической модели (или задача идентификации) подробно рассматривалась в части I пособия. Напомним здесь только основные моменты.

Оценка параметров математической модели (ММ) проводится по результатам наблюдений за реальным процессом, при этом иногда удается ММ преобразовать к стандартной форме:

$$y = \sum_{i=1}^n \theta_i x_i + \varphi_i,$$

что значительно облегчает решение задачи, позволяя применять унифицированные методы идентификации.

Фактически оценка параметров сводится к аппроксимации полученной экспериментально зависимости выхода  $y$  от вектора входа  $x$  полиномом степени  $n$  с коэффициентами  $\theta_i$ ,  $i=1 \dots n$  при возмущениях (погрешностях)  $\varphi_i$ . MATLAB позволяет выполнять такую аппроксимацию с помощью функций `polyfit` (Polynomial fitting — аппроксимация полиномом):

$$c = \text{polyfit}(x, y, n).$$

Функция возвращает вектор  $c = [c_1 \dots c_n]$  коэффициентов многочлена аппроксимации степени  $n$   $p(x)$  зависимости  $y = f(x)$ :

$$p(x) = c_1 x^n + c_2 x^{n-1} + \dots + c_d$$

Многочлен подбирается методом наименьших квадратов посредством решения линейного уравнения

$$Ac = y,$$

где столбцы матрицы  $A$  есть следующие один за другим степени вектора  $x$ , а количество строк равно числу расчетных точек функции  $y = f(x)$ . Решение уравнения  $Ac = y$  осуществляется с помощью операции «деление матриц» языка MATLAB (см. подразд. 2.2):

$$c = A \setminus y$$

Polynomial fitting

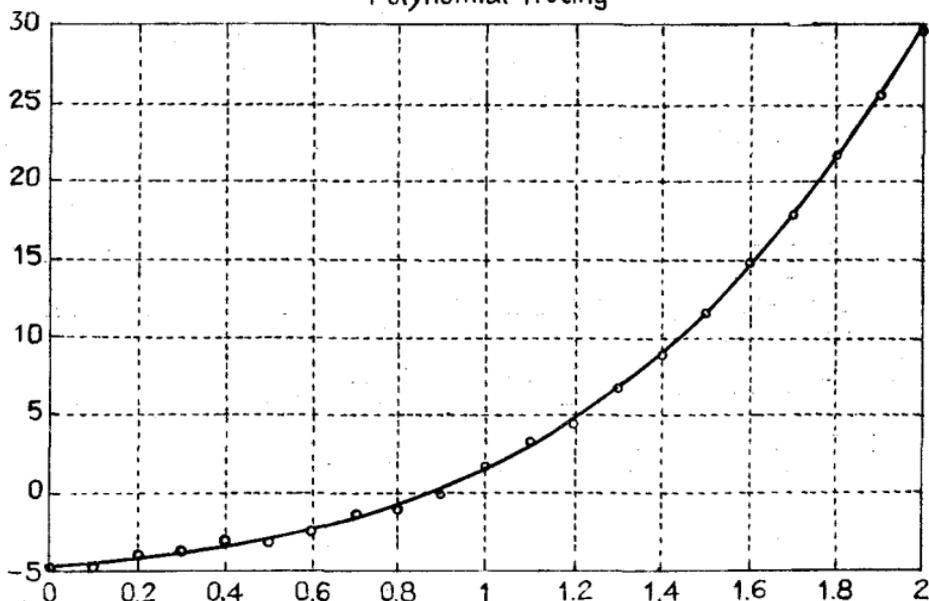


Рис. III. 7

Приведем пример аппроксимации. Сначала создадим зависимость  $y = \theta^T x + \varphi$  при помощи функции вычисления полинома `polyval`:

```
tet=[3 2 1 -5]; % вектор коэффициентов  $\theta$ 
x=0:0.1:2; % область определения
y=polyval(tet, x) + rand(x); % rand — сл. помеха
```

Теперь аппроксимируем полученную зависимость и выведем график:

```
c=polyfit(x, y, 3);  
fit=polyval(c, x);  
plot(x, fit, x, y, 'o')  
title('Polynomial fitting')  
grid
```

В результате на экране увидим графики действительных значений и результатов аппроксимации (рис. III.7).

## 5. РЕШЕНИЕ ЗАДАЧ АНАЛИЗА ДИНАМИКИ СИСТЕМ ПРИ ПОМОЩИ ПАКЕТА АДАМ

АДАМ (Анализ Дифференциально-Алгебраических Моделей) — удобный интерактивный пакет программ для численного анализа динамических систем различной структуры. АДАМ является дополнением к пакету MATLAB и может рассматриваться как тулбокс к нему. Модель анализируемой системы может быть составлена из взаимодействующих подсистем, описываемых линейными или нелинейными, явными или неявными, дифференциальными, разностными или алгебраическими уравнениями. Пользователи могут описывать подсистемы независимо, создавая собственные библиотеки моделей. Язык описания моделей АДАМ позволяет использовать любые матричные операции и функции пакета MATLAB, а также модели из библиотек, описанные на языках АДАМ, MATLAB, Си, фортран-77 и др. Порядок системы и число уровней иерархии подсистем произвольны и ограничены лишь объемом памяти ЭВМ.

После описания по желанию пользователя может быть проведено имитационное моделирование, линеаризация или дискретизация системы. Для моделирования доступен ряд явных и неявных методов численного интегрирования (Эйлера, Рунге-Кутта, формул дифференцирования назад — ФДН и т. д.). Результаты можно обрабатывать стандартными средствами пакета MATLAB (статистика, графика и др.).

АДАМ предоставляет пользователю язык описания различных типов динамических систем и манипулирования с ними, отличающийся от языка MATLAB наличием одного нового понятия — модели. Чтобы описать модель с именем, например ModelName, необходимо создать файл описания ModelName.A — так называемый А-файл. При этом можно использовать матричную нотацию и любые функции MATLAB. Затем А-файл транслируется функцией adam в форму, воспри-

нимаемую системой MATLAB, инициализируется и может быть проинтегрирован, линеаризован или дискретизован вызовом соответствующих команд. Чтобы описать последовательность подобных действий, следует, так же как и при пользовании системой MATLAB, создать и запустить М-файл. Проиллюстрируем применение пакета АДАМ на примерах.

**Пример 1.** Этот пример показывает, как описывать и интегрировать простые непрерывные нелинейные системы. Пусть требуется промоделировать систему

$$dx/dt=y(t), \quad dy/dt=-Ax(t) + \varepsilon(1 - x^2(t))y(t) + Bu(t)$$

(уравнение Ван дер Поля) со значениями параметров  $A=2$ ,  $B=20$ ,  $\varepsilon=5$ , и начальными условиями  $x(0)=0.25$ ,  $y(0)=0$ , на промежутке времени  $[0, 16\text{с}]$ . Для простоты будем считать, что  $u(t)=0$ , т. е. входное возмущение отсутствует.

Создадим файл VDPOL.A, содержащий текст:

```
model sX=Vdpol(U)          % Van der Pol equation
initial
    A=2 B=20 Eps=5        % Initial values of parameters
    sX=0.25 sY=0          % Initial values of states
continuous
    dX=sY
    dY=-A*sX+Eps*(1 - sX ^2)*sY+B*U
end
```

Чтобы создать файл, вызовите из среды MATLAB редактор DOS командой

```
ledit vdpol.a
```

В созданном описании системы после ключевого слова model указываются выходы, имя и входы модели. В секции initial задаются значения параметров и начальные условия. Секция continuous содержит уравнения модели, связей и возмущений. Заметим, что все имена переменных и моделей должны начинаться с заглавной буквы. Префиксы s- и d- используются для указания переменных состояния и их производных, соответственно.

Теперь модель описана и можно создать М-файл EX1.M для ее моделирования:

```
im=adam('Vdpol');
Vdpol_U=0;
```

```
[t, y]=sim(im, 16);  
plot(t, y), title('VDPOL time history')
```

Первая строка файла EX1.M содержит вызов `adam` — функции трансляции описания модели. Вся необходимая для дальнейшего информация о модели запоминается в переменной `im` — информационной матрице. Во второй строке задается значение внешнего возмущения (для доступа к любым переменным моделей при работе функции `adam` создаются соответствующие глобальные переменные, уточняемые именем модели). В третьей строке содержится вызов функции моделирования `sim`, выполняющей численное интегрирование указанной модели на указанном промежутке с указанным правым концом (остальные параметры функции приняли значения по умолчанию, в частности, левая граница интервала интегрирования — ноль). В результате работы функции `sim` будут созданы однострочные матрицы `t, y` результатов моделирования. Последняя строка вызывает стандартные функции системы MATLAB для построения графиков.

После запуска созданного файла командой

```
ex1
```

вы увидите результаты моделирования на экране.

**Пример 2.** Этот пример показывает, как описывать дискретные модели и соединять их связями. Поставим задачу стабилизации системы `Vdpol` из примера 1 в начале координат (моделирование показало, что это состояние равновесия системы неустойчиво). Для стабилизации будем использовать регулятор, который измеряет выход объекта и реализует в дискретном времени стандартный непрерывный пропорционально-интегрально-дифференциальный (ПИД) закон управления

$$dE_i/dt = E/T_i, \quad E_d = T_d dE/dt, \quad u = K(E + E_i + E_d),$$

где  $K$  — общий коэффициент усиления;  $T_i$  — постоянная времени интегрирования;  $T_d$  — постоянная времени дифференцирования.

Дискретная форма ПИД-закона может быть описана в файле `DPID.A` следующим образом:

```
model U=DPID(E)    % Discrete PID-controller  
initial  
    K=0.12          % Gain  
    Ti=20           % Integrator time constant
```

```

Td=2.5           % Differentiator time constant
T0=0.1          % Sampling interval
oEi=0           % Initial value of discrete integrator

oEd=0           % Initial value of holder
discrete (T0)
nEd=Td*E
nEi=oEi+E*T0/Ti
U=K*(E+nEi+(nEd-oEd)/T0)
end

```

Здесь префикс n- (new) указывает на текущее значение переменной, тогда как префикс o- (old) соответствует ее значению на предыдущем временном шаге. Величина интервала дискретизации  $T_0$  указывается в заголовке секции discrete.

Описав субмодели Vdpol и DPID, вы можете соединить их в файле VDPID.A не просто, а очень просто:

```

model [X, U]=VDPID(Xref)
continuous
  X=mVdpol(U)
  U=mDPID(Xref-X)
end

```

Создав теперь M-файл для моделирования EX2.M

```

im=adam('VDPID');
VDPID_Xref=0;
[t, y]=sim(im, 5);
plot(t, y(1,:), t, y(2,:), 'x'), ...
title('VDPID time history')

```

и запустив его, вы увидите на экране переходные процессы в замкнутой системе.

**БАЗОВЫЕ ФУНКЦИИ СРЕДЫ  
MATLAB**

**Функции общего назначения**

help	оперативная подсказка
demo	запуск демонстрационных программ
who	список переменных в памяти
what	список М-файлов на диске
size	размерность переменной
length	длина вектора
clear	очистка рабочей области
computer	тип компьютера
^ C	прерывание работы программы
quit	выход из системы MATLAB
exit	то же, что и quit

**Матричные операции**

+	сложение
-	вычитание
*	умножение
/	правое деление
\	левое деление
^	возведение в степень
'	транспонирование с сопряжением

**Поэлементные операции**

+	сложение
-	вычитание
.*	умножение
./	правое деление
.\	левое деление
.^	возведение в степень
.'	транспонирование

**Логические операции и отношения порядка**

<	меньше
<=	меньше или равно
>	больше
>=	больше или равно
==	равно
~=	не равно

&	логическое И
	логическое ИЛИ
~	логическое НЕ

## Специальные символы

=	оператор присваивания
[	используется для создания векторов и матриц
]	см. [
(	приоритет арифметических операций
)	см. (
.	десятичная точка
...	продолжение оператора на следующую строку
,	разделяет индексы и аргументы функции
:	заканчивает строки без отображения результата
%	вводит комментарии
:	сечение и генерирование векторов
!	выполняет команду операционной системы

## Специальные значения

ans	ответ, если выражение без присваивания
eps	точность значений с плавающей точкой
pi	число $\pi$
i, j	$\sqrt{-1}$
inf	$\infty$
NaN	не-число (Not-a-Number)
clock	часы
date	дата
flops	число операций с плавающей точкой
nargin	количество входных аргументов функции
nargout	количество выходных аргументов функции

## Функции обработки строк и текста

abs	преобразует строку в ASCII значения
eval	вычисляет значение текстового макроса
num2str	преобразует число в строку
int2str	преобразует целое значение в строку
setstr	устанавливает флаг индикации строки
sprintf	преобразует значение в строку
fprintf	то же, что и sprintf, но с выводом строки на экран
isstr	обнаруживает строковые переменные
strcmp	сравнивает строковые переменные
hex2num	преобразует строку с шестнадцатиричным значением в число с плавающей точкой

## Графический вывод

plot	линейный X — Y график
loglog	логарифмический X — Y график
semilogx	полулогарифмический X — Y график
semilogy	полулогарифмический X — Y график
polar	график в полярных координатах
mesh	трехмерная контурная поверхность
contour	линии уровня
meshdom	область определения для графиков поверхности контура
bar	столбцовая диаграмма
stairs	лестничная диаграмма
errorbar	изображение интервальных допусков

## Графические комментарии

title	заголовок графика
xlabel	обозначение оси абсцисс
ylabel	обозначение оси ординат
grid	наложение сетки на график
text	вывод текста в произвольной позиции графического экрана
gtext	вывод текста в позиции, выбранной с помощью мыши
ginput	ввод в графическом режиме

## Управление графическим окном

axis	масштабирование осей вручную
hold	фиксация графика на экране
shg	показывает графический экран
clf	стирает графический экран
subplot	разбивает графический экран на окна

## Вывод графического окна на печать

print	печать текущего графика
prtsc	печать графического экрана
meta	открывает графический метафайл для последующей обработки и распечатки графиков

## Функции командного окна

clc	стирает командный экран
home	устанавливает курсор в начальную позицию
format	установка формата вывода на дисплей
disp	вывод на дисплей текста или матрицы
fprintf	форматированный вывод значений на дисплей
echo	устанавливает эхо-контроль

## Управляющая логика

if	оператор условного выполнения
elseif	используется вместе с if
else	используется вместе с if
end	ограничивает if, for, while
for	выполняет фрагмент программы определенное число раз
while	выполняет фрагмент программы, пока не выполнено заданное условие
break	досрочный выход из циклов for и while
return	возврат из функции
pause	приостановка выполнения до нажатия клавиши

## Программирование и М-файлы

input	ввод значений с клавиатуры
keyboard	в М-файле передает управление клавиатуре
error	выводит на экран сообщение об ошибке
function	определяет функцию
eval	выполняет текст, содержащийся в строке
feval	выполняет функцию, заданную в строке
echo	устанавливает эхоконтроль
exist	проверяет, существует ли переменная
casesen	чувствительность к регистрам клавиатуры
global	определяет глобальные переменные
startup	стартовый М-файл
getenv	строка переменной операционной системы
menu	выбор пунктов из меню
etime	хронометр

## Работа с диском

<b>chdir</b>	замена текущего каталога
<b>delete</b>	удаление файла
<b>diary</b>	запись в файл сеанса работы
<b>dir</b>	каталог файлов на диске
<b>load</b>	загрузка данных из файла
<b>save</b>	сохранение данных в файле
<b>type</b>	выдача списка функций и M-файлов
<b>what</b>	список M-файлов в текущем каталоге
<b>fprintf</b>	форматированный вывод в файл
<b>pack</b>	уплотнение памяти посредством save

## Отношение порядка и логические функции

<b>any</b>	проверка массивов на логические условия
<b>all</b>	проверка массивов на логические условия
<b>find</b>	нахождение индексов ненулевых элементов массива
<b>exist</b>	проверка существования переменной
<b>isnan</b>	обнаружение не-чисел (NaNs)
<b>finite</b>	обнаружение бесконечности
<b>isempty</b>	обнаружение пустых матриц
<b>isstr</b>	обнаружение строковых переменных
<b>strcmp</b>	сравнение переменных-строк

## Тригонометрические функции

<b>sin</b>	синус
<b>cos</b>	косинус
<b>tan</b>	тангенс
<b>asin</b>	арксинус
<b>acos</b>	арккосинус
<b>atan</b>	арктангенс
<b>atan2</b>	значения арктангенса для четырех квадрантов
<b>sinh</b>	синус гиперболический
<b>cosh</b>	косинус гиперболический
<b>tanh</b>	тангенс гиперболический
<b>asinh</b>	арксинус гиперболический
<b>acosh</b>	арккосинус гиперболический
<b>atanh</b>	арктангенс гиперболический

## Элементарные математические функции

abs	модуль числа или комплексная амплитуда
angle	фазовый угол
sqrt	квадратный корень
real	вещественная часть
imag	мнимая часть
conj	комплексно-сопряженная величина
round	округление до ближайшего целого
fix	округление отбрасыванием дробной части
floor	округление вниз
ceil	округление вверх
sign	функция знака (сигнум)
rem	остаток или модуль
exp	экспоненциальная функция
log	натуральный логарифм
log10	десятичный логарифм

## Специальные функции

bessel	функция Бесселя
gamma	полная и неполная гамма-функции
rat	рациональная аппроксимация
erf	функция ошибки
inverf	обратная функция ошибки
ellipk	полный эллиптический интеграл первого вида
ellipj	эллиптические функции Якоби

## Полиномы

poly	характеристический полином
roots	корни полинома — метод сопряженной матрицы
rootsl	корни полинома — метод Лагерра
polyval	вычисление полинома
poluvalm	нахождение матричного полинома
conv	умножение
deconv	деление
residue	разложение на простейшие дроби
polyfit	аппроксимация полиномом

## Действия с матрицами

rot90	поворот
fliplr	симметрия матрицы относительно вертикальной оси
flipud	симметрия матрицы относительно горизонтальной оси
diag	выделение или создание диагональной матрицы
tril	нижняя треугольная подматрица
triu	верхняя треугольная подматрица
reshape	переформирование матрицы
'	транспонирование матрицы

## Специальные матрицы

compan	сопряженная матрица
diag	диагональная матрица
eye	единичная матрица
gallery	«галерея» тестовых матриц
hadamard	матрица Адамара
hankel	матрица Ганкеля
hilb	матрица Гильберта
invhilb	обратная матрица Гильберта
linspace	вектор членов арифметической прогрессии
logspace	вектор членов геометрической прогрессии
meshdom	область определения для трехмерного графика
ones	матрица единиц
rand	матрица случайных элементов
toeplitz	матрица Тёплица
vander	матрица Вандермонда
zeros	нулевая матрица

## Число обусловленности матрицы

cond	число обусловленности в норме $L^2$
norm	норма $L^1$ , норма $L^2$ , F-норма, $\infty$ -норма
rank	ранг матрицы
rcond	оценка обусловленности

## Разложение и факторизация

balance	балансировка матрицы
backsub	обратная подстановка
cdf2rdf	преобразование в комплексно-диагональную форму в вещественно-диагональную
chol	факторизация Холецкого
eig	собственные числа и собственные векторы
hess	форма Гессенберга
inv	обращение
lu	множители из Гауссова исключения
npls	метод неотрицательных наименьших квадратов
null	нуль-пространство
orth	ортогонализация
pinv	псевдообращение
qr	ортогонально-треугольное разложение
qz	QZ алгоритм
rsf2csf	преобразование вещественной формы Шура в комплексную
schur	разложение Шура
svd	разложение по сингулярным числам

## Элементарные матричные функции

expm	матричная экспонента
logm	матричный логарифм
sqrtn	матричный квадратный корень
funm	произвольная матричная функция
poly	характеристический полином
det	определитель
trace	след
kron	тензорное произведение Кронекера

## Интерполяция

spline	кубический сплайн
table1	просмотр одномерной таблицы
table2	просмотр двумерной таблицы

## Решение дифференциальных уравнений

ode23	метод Рунге-Кутты 2-го — 3-го порядка
ode45	метод Рунге-Кутты 4-го — 5-го порядка

## Численное интегрирование

quad	численное интегрирование функции
quad8	численное интегрирование функции

## Нелинейные уравнения и оптимизация

fmin	минимум функции одного переменного
fmins	минимум функции многих переменных с ограничениями
fsolve	решение системы нелинейных уравнений
fzero	нахождение корня функции одной переменной

## Анализ данных по столбцам

max	максимальное значение
min	минимальное значение
mean	среднее значение
median	медиана
std	среднее отклонение (МО)
sort	сортировка
sum	сумма элементов
prod	произведение элементов
cumsum	кумулятивная сумма элементов
cumprod	кумулятивное произведение элементов
diff	приближенное произведение
hist	гистограмма
corrcoef	коэффициенты корреляции
cov	матрица ковариации
cp1xpair	переупорядочение в комплексных парах

## Обработка сигналов

abs	комплексная амплитуда
angle	фаза
conv	свёртка
corrcoef	коэффициенты корреляции
cov	ковариационная матрица
deconv	обратная свертка
fft	быстрое преобразование Фурье (БПФ)
fft2	двухмерное БПФ
ifft	обратное БПФ
ifft2	обратное двухмерное БПФ
fftshift	перекоммутация секторов матриц

## ПРИЛОЖЕНИЕ 2

### ФУНКЦИИ ТУЛБОКСА «СИСТЕМЫ УПРАВЛЕНИЯ» (CONTROL SYSTEM TOOLBOX)

#### Построение моделей

append	добавление уравнений динамики системы
connect	моделирование блок-схем
parallel	параллельное соединение систем
series	последовательное соединение систем
ord2	формирование матриц A, B, C, D для системы 2-го порядка

#### Преобразования моделей

ss2tf	пространство состояний → передаточная функция
ss2zp	пространство состояний → нули и полюса
tf2ss	передаточная функция → пространство состояний
tf2zp	передаточная функция → нули и полюса
zp2tf	нули и полюса → передаточная функция
zp2ss	нули и полюса → пространство состояний
residue	разложение на простейшие дроби
c2d	непрерывная модель → дискретная модель
d2c	дискретная модель → непрерывная модель

### Реализации моделей

ctrbf	управляемая блочная форма
obsvt	наблюдаемая блочная форма
minreal	минимальная реализация системы
balreal	сбалансированная реализация
modred	понижение порядка модели
dbalreal	дискретная сбалансированная реализация
dmodred	понижение порядка дискретной модели

### Свойства моделей

damp	собственные частоты и коэффициенты демпфирования
gram	грамианы направляемости и наблюдаемости
dgram	грамианы управляемости и наблюдаемости
ctrb	матрица управляемости
obsv	матрица наблюдаемости
tzero	передаточные нули

### Временные характеристики

impulse	импульсная характеристика
step	переходная характеристика
lsim	реакция непрерывной системы на произвольное задающее воздействие
dimpulse	дискретная импульсная характеристика
dstep	дискретная переходная характеристика
dlsim	реакция дискретной системы на произвольное задающее воздействие

### Частотные характеристики

bode	амплитудно-фазовые частотные характеристики (АФЧХ)
nyquist	годограф Найквиста
dbode	дискретные АФЧХ
freqz	z-преобразование частотной характеристики
freqs	преобразование Лапласа частотной характеристики

## Расчет регуляторов

lqr	расчет	линейно-квадратичного	оптимального	регуля-
	то			
lqe	расчет	линейно-квадратичного	оптимального	наблю-
	дателя			
dlqr	расчет	линейно-квадратичного	дискретного	регулятора
dlqe	расчет	линейно-квадратичного	дискретного	наблюда-
	теля			
margin	расчет	запаса по амплитуде и по фазе		
place	размещение	полюсов		
rlocus	годограф	полюсов замкнутой системы		
lyap	решение	уравнения Ляпунова		
dlyap	решение	дискретного уравнения Ляпунова		

## ЗАКЛЮЧЕНИЕ

Некоторые из изложенных понятий и подходов первоначально были разработаны в кибернетике и теории автоматического управления, а в других областях являются пока непривычными. Тем не менее, они носят универсальный характер.

Когда-то считалось, что явление изучено, если построена его математическая модель. Понадобился гений Р. Декарта и определенный уровень развития науки, чтобы сформулировать новый принцип: для каждого явления возможны несколько моделей: вопрос в том, какая из них правильная. Сегодня мы начинаем понимать, что не существует единственной «правильной» модели! Даже в рамках одного и того же исследования может одновременно использоваться несколько ММ сообразно с целями исследования и вычислительными средствами.

Хочется подчеркнуть, что современное понимание места и роли математического моделирования в научном и инженерном исследовании, представленное в ряде работ и, в частности, в данном пособии, сложилось под действием бурного развития вычислительной техники. Программы, реализующие отдельные модели и методы, в настоящее время подвергаются обобщению и интегрируются в пакеты прикладных программ (ППП), поддерживающие все этапы математического моделирования: построение ММ, исследование (анализ) и использование (синтез). Примерами таких ППП являются программная система MATLAB с тулбоксами, описанная в части III, а также пакет АВАНС (АВтоматизация АНализа и Синтеза), разработанный в ЛМИ [53, 54]. АВАНС предназначен для автоматизации проектирования сложных систем в различных областях (систем автоматического управления, механических, электромеханических систем и т. п.). АВАНС поддерживает основные этапы математического моделирования статических и динамических, дискретных и непрерывных, линейных и нелинейных, детерминированных и стохастических систем. В нижеприведенной таблице перечислены основные операции, выполняемые на каждом этапе, с указа-

нием поддерживающих их операторов входного языка АВАНС.

Дальнейшее развитие математического моделирования связано с созданием новых информационных и исследовательских технологий, ориентированных на персональные ЭВМ. Раньше решением задачи на ЭВМ занималось несколько человек: специалист-эксперт ставил задачу, математик-прикладник выбирал алгоритм решения, а программист писал и отлаживал программу. Теперь же специалист получает доступ к компьютеру на всех стадиях работы. Наибольших успехов добиваются те, кто владеет навыками применения компьютера на всех стадиях. Меняется и характер используемых программных средств: закрытые ППП уступают место инструментальным системам, и средам, при работе с которыми пользователь создает собственные библиотеки моделей и методов обработки данных (МАТЛАВ и АВАНС с успехом используются в качестве таких сред). Таким образом, исследователь «перекачивает» в программы часть своих знаний и, опираясь на них, получает новые знания, которые снова фиксирует в ЭВМ. Происходит автоформализация профессиональных знаний [51], в корне меняющая стиль и темпы исследования.

Подводя итог, перечислим некоторые упоминавшиеся выше полезные советы, которые можно назвать «заповедями» математического моделирования.

1. Не решай сложную задачу, не решив простую (принцип простоты).

2. Без ошибки нет модели, а потому негрубые модели — плохие (принцип А. А. Андропова).

3. Можно пренебрегать чем угодно, нужно только точно знать, как это повлияет на результат (принцип Э. Хэмингуэя).

4. Чем проще модель, тем реже она обманет (принцип надежности).

5. Точность результатов не может быть выше точности исходных данных; точности промежуточных вычислений должны быть согласованы (принцип А. Н. Крылова).

6. Цель расчетов — не числа, а понимание. Прежде чем решать задачу, подумай, что делать с ее решением (закон Р. Хеминга).

Наконец, хочется еще раз повторить, что окружающая действительность является началом и концом математического моделирования. Именно в этом смысле следует понимать эпиграф, помещенный в начале книги.

№ п/п	Этапы математического моделирования	Средства поддержки в ППП АВАНС
1	<p><b>Построение ММ системы</b></p> <p>1.1. Определение входных/выходных переменных и параметров</p> <p>1.2. Определение целей и показателей качества</p> <p>1.3. Выбор структуры ММ системы (см. разд. ч. I)</p> <p>1.4. Выбор (оценка) параметров ММ по таблицам временных и частотных характеристик</p> <p>1.5. Описание ММ и ввод в ЭВМ</p>	<p>IDENT, IDENW</p> <p>SYST, WNN, SYSTAD, библиотеки типовых линейных динамических звеньев и типовых нелинейностей</p>
2	<p><b>Исследование ММ системы</b></p> <p>2.1. Преобразования ММ (п. 2.2 ч. I)</p> <ul style="list-style-type: none"> <li>— линеаризация</li> <li>— расчет передаточных функций</li> <li>— расчет диагональной канонической формы</li> <li>— редуцирование</li> <li>— дискретизация</li> <li>— континуализация</li> </ul> <p>2.2. Анализ ММ (ч. II).</p> <ul style="list-style-type: none"> <li>— анализ статического режима</li> <li>— анализ динамики (имитационное моделирование)</li> <li>— расчет собственных чисел <math>\lambda_i(A)</math></li> <li>— расчет нулей и полюсов ПФ (<math>\text{Re } \lambda_i, \text{Im } \lambda_i</math>)</li> <li>— расчет частотных характеристик <math>W(j\omega), A(\omega), \Phi(\omega), L(\omega)</math></li> <li>— расчет показателей качества</li> </ul> <p>2.3. Комплексное исследование</p>	<p>LIN TRANE WDIAG</p> <p>WREDUC, SIM+GDENT WDISC</p> <p>STAT, IMPLS SIM, INTEGR</p> <p>QR</p> <p>WNULPOL</p> <p>LFC</p> <p>Библиотека типовых показателей</p> <p>ANALYS</p>
3	<p><b>Использование ММ на примере проектирования САУ</b></p> <p>3.1. Синтез базовых структур регуляторов</p> <p>3.2. Описание и ввод в ЭВМ базовых структур регуляторов</p>	<p>Библиотека типовых регуляторов</p> <p>То же</p>

№ п/п	Этапы математического моделирования	Средства поддержки в ППП АВАНС
3	3.3. Расчет параметров регуляторов по упрощенной ММ объекта управления — анализ замкнутой САУ  — перебор — оптимизация  3.4. Предварительный отбор регуляторов 3.5. Уточнение параметров регуляторов по полной ММ объекта — перебор  — оптимизация  3.6. Окончательный отбор регуляторов 3.7. Анализ замкнутой САУ	SIM + библиотека типовых показателей (см. также п. 2.2 ч. I)  MIN + библиотека типовых показателей   SIM + библиотека типовых показателей MIN + библиотека типовых показателей  См. п. 2.2 ч. I

## СПИСОК ЛИТЕРАТУРЫ

1. Перегудов Ф. И., Тарасенко Ф. П. Введение в системный анализ. М.: Высшая школа, 1989. 367 с.
2. Моисеев Н. Н. Математические задачи системного анализа. М.: Наука, 1981. 488 с.
3. Теория систем и методы системного анализа в управлении и связи/Под ред. В. Г. Лазарева и др. М.: Радио и связь, 1983. 248 с.
4. Дудорин В. И., Алексеев Ю. Н. Системный анализ экономики на ЭВМ. М.: Финансы и статистика, 1986. 190 с.
5. Компьютеры, модели, вычислительный эксперимент/Под ред. А. А. Самарского. М.: Наука, 1988. 169 с.
6. Норенков И. П. Введение в автоматизированное проектирование технических устройств и систем. М.: Высшая школа, 1986. 311 с.
7. Корячко В. П., Курейчик В. М., Норенков И. П. Теоретические основы САПР. М.: Энергоатомиздат, 1987. 400 с.
8. Растринин Л. А. Современные принципы управления сложными объектами. М.: Сов. радио, 1980. 232 с.
9. Справочник по теории автоматического управления/Под ред. А. А. Красовского. М.: Наука, 1987. 711 с.
10. Блехман И. И., Мышкис А. Д., Пановко Я. Г. Механика и прикладная математика. Логика и особенности приложений математики М.: Наука, 1990. 356 с.

11. *Веников В. А.* Теория подобия и моделирование. М.: Высшая школа, 1984. 439 с.
12. *Бусленко Н. П.* Моделирование сложных систем. М.: Наука, 1978. 399 с.
13. *Советов Б. Я., Яковлев С. А.* Моделирование систем. М.: Высшая школа, 1985. 272 с.
14. *Вентцель Е. С.* Исследование операций. Задачи, принципы, методология. М.: Наука, 1988. 208 с.
15. *Краснощечков П. С., Петров А. А.* Принципы построения моделей. М.: Изд-во МГУ, 1983. 264 с.
16. *Левин Б. Р., Шварц В.* Вероятностные модели и методы в системах связи и управления. М.: Радио и связь, 1985. 312 с.
17. *Снапелев Ю. М., Старосельский В. А.* Моделирование и управление в сложных системах. М.: Сов. радио, 1974. 264 с.
18. Математическое моделирование/Под ред. Дж. Эндрюса, Р. Маклоуна. М.: Мир, 1979. 277 с.
19. *Бешелев С. Д., Гурвич Ф. Г.* Математико-статистические методы экспертных оценок. М.: Статистика, 1974. 160 с.
20. *Фрадков А. Л.* Основы математического моделирования/Ленингр. мех. ин-т. Л., 1989. 88 с.
21. *Месарович Д., Такахара Я.* Общая теория систем: математические основы. М.: Мир, 1978. 312 с.
22. *Директор С., Рорер Р.* Введение в теорию систем. М.: Мир, 1974. 644 с.
23. *Касты Дж.* Большие системы: связность, сложность и катастрофы. М.: Мир, 1982. 216 с.
24. *Робертс Ф.* Дискретные модели и их приложения в технике, биологии, экологии. М.: Наука, 1986. 494 с.
25. Теория выбора и принятия решений/*И. М. Макаров* и др. М.: Наука, 1982. 328 с.
26. *Поспелов Д. А.* Логико-лингвистические модели в системах управления. М.: Энергонддат, 1981. 232 с.
27. *Кофман А., Анри-Лабордер А.* Методы и модели исследования операций: Целочисленное программирование. М.: Мир, 1977. 432 с.
28. *Поляк Б. Т.* Введение в оптимизацию. М.: Наука, 1983. 384 с.
29. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. М.: Мир, 1985. 509 с.
30. *Бахвалов Н. С., Жидков Н. П., Кобельков Г. М.* Численные методы. М.: Наука, 1987. 599 с.
31. *Глушков В. М.* Синтез цифровых автоматов. М.: Физматгиз, 1962. 476 с.
32. *Питерсон Дж.* Теория сетей Петри и моделирование систем. М.: Мир, 1984. 356 с.
33. *Танаев В. С., Шкурба В. В.* Введение в теорию расписаний. М.: Наука, 1975. 256 с.
34. *Хоггер К.* Введение в логическое программирование. М.: Мир, 1988. 348 с.
35. *Первозванский А. А.* Курс теории автоматического управления. М.: Наука, 1986. 615 с.
36. Управление и системное моделирование в гибких производственных системах/Под ред. *С. В. Емельянова*. М.: Наука, 1988. 312 с.
37. *Шилов Г. Е.* Математический анализ. Функции нескольких вещественных переменных. М.: Наука, 1972. 622 с.
38. *Каинов В. Х., Селютин А. А., Дубровский С. А.* Методы обра-

ботки данных в системах с нечеткой информацией. Фрунзе: Илим, 1988. 188 с.

39. Цурков В. Н. Агрегирование данных при решении динамических задач большой размерности. М.: Наука, 1987. 484 с.

40. Тутубалин В. Н. Теория вероятностей. М.: Изд-во МГУ, 1972. 231 с.

41. Нечеткие множества в моделях управления и искусственного интеллекта/Под ред. Д. А. Поспелова. М.: Наука, 1986. 396 с.

42. Обработка нечеткой информации в системах принятия решений/А. Н. Борисов и др. М.: Радио и связь, 1989. 304 с.

43. Хант Э. Искусственный интеллект М.: Мир, 1978. 500 с.

44. Кокс Д., Снелл Э. Прикладная статистика. Принципы и примеры. М.: Мир, 1984. 200 с.

45. Райбман Н. С., Чадеев В. М. Построение моделей процессов производства. М.: Энергия, 1975. 375 с.

46. Хартман К., Лецкий Э., Шефер В. Планирование эксперимента в исследовании технологических процессов. М.: Энергия, 1977. 52 с.

47. Плескунин В. И., Воронина Е. Д. Теоретические основы организации и анализа выборочных данных в эксперименте. Л.: Изд-во ЛГУ, 1979. 232 с.

48. Горский В. Г., Адлер Ю. П., Талалай А. М. Планирование промышленных экспериментов. М.: Металлургия, 1978. 112 с.

49. Хемминг Р. В. Численные методы. М.: Наука, 1972. 400 с.

50. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. М.: Мир, 1980. 279 с.

51. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации. М.: Наука, 1984. 240 с.

52. Практикум по программированию/Под ред. Н. С. Бахвалова, А. В. Михалева. М.: Изд-во МГУ, 1986. 208 с.

53. Автоматизация проектирования алгоритмических структур систем управления/Б. Р. Андриевский, Д. П. Деревицкий, В. Н. Уткин, А. Л. Фрадков; Ленингр. мех. ин-т. Л., 1985. 100 с.

54. Основы анализа и синтеза сложных динамических систем с использованием ППП АВАНС/Б. Р. Андриевский, Д. П. Деревицкий, В. В. Касаткин и др.; Ленингр. мех. ин-т. Л., 1988. 72 с.

55. Смирнов Ю. М., Чекмарев О. А. Автоматизация моделирования и обработки экспериментальных данных/Ленингр. мех. ин-т; Л.; 1984. Л., 98 с.

56. Кулаков Ф. М. Основы машинного моделирования сложных технических систем/Ленингр. мех. ин-т, Л., 1988. 68 с.

57. Воеводин В. В. Вычислительные основы линейной алгебры. М.: Наука, 1977. 303 с.

58. Митропольский А. К. Техника статистических вычислений. М.: Наука, 1971. 576 с.

59. Деревицкий Д. П., Фрадков А. Л. Прикладная теория дискретных адаптивных систем управления. М.: Наука, 1981. 216 с.

60. McClamroch N. H. State models of dynamic systems. — A case study approach. Springer-Verlag, 1980, 246 p.

61. Fradkov A. L., Egorenkov D. L., Spiridonov A. A. ADAM. User's Guide. St. Petersburg, 1992.

62. Egorenkov D. L., Fradkov A. L., Spiridonov A. A. Algorithms and software for simulation of implicit dynamic systems with constraints// 16<sup>th</sup> IFIP conference on System Modelling and Optimization. Compiegne, France, July 5—9, 1993.

63. Дэнис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений. М.: Мир, 1988. 440 с.
64. Самарский А. А., Гулин А. В. Численные методы. М.: Наука, 1988. 440 с.
65. Математика и САПР: В 2-х кн./Жермен-Лакур П., Жорж П. Л., Пистр Ф., Безье П. М.: Мир, 1989. 264 с.
66. Фрадков А. Л. Адаптивное управление в сложных системах. М.: Наука, 1990. 294 с.
67. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений. М.: Наука, 1986. 288 с.
68. Методы вычислений на ЭВМ: Справочное пособие/Иванов В. В. Киев. Наук. думка, 1986. 584 с.
69. Писсанецки С. Технология разреженных матриц. М.: Мир, 1988. 410 с.
70. Шуп Т. Решение инженерных задач на ЭВМ. М.: Мир, 1982. 235 с.
71. Райс Дж. Матричные вычисления и математическое обеспечение. М.: Мир, 1984. 264 с.
72. Икрамов Х. Д. Численные методы линейной алгебры//Математика и кибернетика. 1987. № 4.
73. Икрамов Х. Д. Численное решение матричных уравнений. М.: Наука, 1984. 190 с.
74. Икрамов Х. Д. Численные методы для симметричных линейных систем. М.: Наука, 1988. 160 с.
75. Эстребю О., Златев З. Прямые методы для разреженных матриц. М.: Мир, 1987. 120 с.
76. Лоусон Ч., Хенсон Р. Численное решение задач метода наименьших квадратов. М.: Наука, 1986. 232 с.
77. Льюнг Л. Идентификация систем. Теория для пользователей. М.: Наука, 1992. 431 с.
78. Лаврентьев М. А., Шабат Б. В. Методы теории функций комплексного переменного. М.: Наука, 1987. 688 с.
79. Арушанян О. Б., Залеткин С. Ф. Численное решение обыкновенных дифференциальных уравнений на фортране. М.: Изд-во МГУ, 1990. 336 с.
80. Деккер К., Вервер Я. Устойчивость методов Рунге-Кутты для жестких нелинейных дифференциальных уравнений. М.: Мир, 1988. 334 с.
81. Петренко А. И., Власов А. И., Тимеченко А. П. Табличные методы моделирования электронных схем на ЭЦВМ. Киев: Вища школа, 1977. 192 с.
82. Андреев Ю. Н. Управление конечномерными линейными системами. М.: Наука, 1976. 424 с.
83. Ракитский Ю. В., Устинов С. М., Чернооруцкий И. Г. Численные методы решения жестких систем. М.: Наука, 1979. 208 с.
84. Влах И., Синглов К. Машинные методы анализа и проектирования электронных схем. М.: Радио и связь, 1988. 560 с.
85. MATLAB. Demo Disk User's Guide. The MathWorks, Inc., 1989.

# ОГЛАВЛЕНИЕ

Предисловие . . . . .	3
-----------------------	---

## Часть I. ПОСТРОЕНИЕ МОДЕЛЕЙ

1. Методология математического моделирования. Системный анализ	6
1.1. Понятие системы. Примеры систем	6
1.2. Этапы системного анализа	10
1.3. Сложные системы и декомпозиция	16
1.4. Экспертные оценки	19
2. Выбор структуры математической модели	25
2.1. Классификация моделей	25
2.1.1. Статические и динамические модели	26
2.1.2. Дискретные и непрерывные модели	27
2.2. Модели состояния динамических систем	32
2.2.1. Модели общего вида	32
2.2.2. Линейные модели	33
2.2.3. Дискретизация и континуализация	36
2.3. Детерминированные и стохастические модели	39
2.4. Нечеткие модели	44
2.4.1. Нечеткие множества и лингвистические переменные	44
2.4.2. Нечеткие системы	48
2.4.3. Задачи группировки и упорядочения	52
2.4.4. Нечеткие числа	54
2.4.5. Вероятность или нечеткость?	57
2.5. Принципы выбора модели	58
3. Выборы параметров математической модели	61
3.1. Предварительные преобразования	61
3.1.1. Линейно параметризованные модели	61
3.1.2. Преобразование статических моделей	62
3.1.3. Преобразование динамических моделей	65
3.2. Регрессионный анализ и метод наименьших квадратов	70
3.3. Адаптивные модели и рекуррентные методы	73

## Часть II. АНАЛИЗ МОДЕЛЕЙ

1. Анализ статики математических моделей систем	76
1.1. Аналитические методы	78
1.1.1. Метод возмущений	79
1.1.2. Метод асимптотических разложений	80
1.2. Численные методы	81
1.2.1. Метод простых итераций	85
1.2.2. Метод Ньютона и его модификации	87
1.2.3. Метод продолжения по параметру	92
1.3. Особенности численного решения систем линейных алгебраических уравнений	93

2. Анализ динамики математических моделей систем . . . . .	96
2.1. Аналитические методы . . . . .	96
2.1.1. Метод возмущений . . . . .	96
2.1.2. Метод асимптотических разложений . . . . .	98
2.2. Численные методы . . . . .	100
2.2.1. Одношаговые методы решения задачи Коши . . . . .	104
2.2.2. Многошаговые методы решения задачи Коши . . . . .	109
2.2.3. Методы решения систем дифференциальных уравнений, неразрешенных относительно производной . . . . .	113
2.2.4. Методы решения линейных систем дифференциальных уравнений . . . . .	114
2.2.5. Устойчивость численного интегрирования . . . . .	118
2.2.6. Выбор шага интегрирования . . . . .	122
2.2.7. Особенности интегрирования жестких систем . . . . .	125
2.2.8. Выбор метода интегрирования . . . . .	130
3. Культура вычислений на ЭВМ . . . . .	132

### Часть III. РЕШЕНИЕ ЗАДАЧ В СРЕДЕ ПРОГРАММИРОВАНИЯ MATLAB

1. Базовые возможности системы MATLAB . . . . .	140
1.1. Начало работы . . . . .	140
1.2. Действия с матрицами . . . . .	141
1.2.1. Ввод и простейшие операции . . . . .	141
1.2.2. Деление матриц . . . . .	143
1.2.3. Модификация матриц . . . . .	143
1.3. Условные операторы и циклы . . . . .	144
1.4. Функции и М-файлы . . . . .	145
1.4.1. Функциональные М-файлы . . . . .	145
1.4.2. Скрипт-файлы . . . . .	147
1.5. Графика . . . . .	147
2. Решение задач линейной алгебры . . . . .	149
3. Решение задач численного анализа . . . . .	152
3.1. Решение систем нелинейных алгебраических уравнений . . . . .	152
3.2. Решение обыкновенных дифференциальных уравнений . . . . .	153
4. Решение задач теории систем . . . . .	155
4.1. Расчет частотных и временных характеристик систем с обратной связью . . . . .	155
4.2. Преобразование математических моделей . . . . .	160
4.3. Вычисление матричной экспоненты . . . . .	161
4.4. Оценка параметров модели . . . . .	162
5. Решение задач анализа динамики систем при помощи пакета АДАМ . . . . .	164
Приложение 1. Базовые функции среды MATLAB . . . . .	168
Приложение 2. Функции тулбокса и «системы управления» (CONTROL SYSTEM TOOLBOX) . . . . .	177
Заключение . . . . .	180
Список литературы . . . . .	183