

519.6(75)
М.19
В. Т. МАЛИКОВ
Р. Н. КВЕТНЫЙ

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ и применение ЭВМ

```
PROGRAM MAL
  REAL X(1000)
  PI=3.1415926
  I1=13
  I2=17
  N=0
  R1=RAN(I1, I2)
  R2=RAN(I1, I2)
  X(N+1)=SQRT(-2.*ALOG(R2))
  &      COS(2.*PI*R1)
  X(N+2)=SQRT(-2.*ALOG(R1))
  &      COS(2.*PI*R2)
  N=N+2
  IF(N.LT.1000)GOTO 10
  CALL GST(X,1000,7)
  STOP
  END
  MX = 2.1615995E-02
  DX = 0.9916891
  SKD= 0.9958359
  ЛЕВАЯ ГРАНИЦА ГИСТОГРАММ
  -2.965892
  ПРАВАЯ ГРАНИЦА ГИСТОГРАММ
  3.009124
```

2293-130

В. Т. МАЛИКОВ
Р. Н. КВЕТНЫЙ

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ и применение ЭВМ

Допущено Министерством
высшего и среднего
специального образования УССР
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальностям
«Автоматика и управление
в технических системах»
и «Прикладная математика»

АБОНЕМ. 2

К и е в
Головное издательство
издательского объединения
«Выща школа»
1989

ББК 22.19я73
М19
УДК 681.3(07)

Рецензенты: д-р техн. наук, проф. *В. Д. Циделко*;
канд. физ.-мат. наук, доц. *Б. Н. Белоусов* (Киевский
политехнический институт); д-р техн. наук, проф.
Б. И. Швецкий (Львовский политехнический институт)

Редакция информатики и автоматике
Редактор *Л. Н. Чмиль*

Маликов В. Т., Кветный Р. Н.

М19 Вычислительные методы и применение ЭВМ:
Учеб. пособие.— К.: Выща шк. Головное изд-во,
1989.— 213 с.
ISBN 5—11—001183—4.

Рассмотрены традиционные методы вычислительной математики (методы решения нелинейных уравнений, численные методы линейной алгебры, методы решения дифференциальных уравнений, методы обработки данных), а также вопросы, связанные с разработкой математических моделей, в частности информационных, включающих принципы статистического моделирования. Приведены примеры решения типовых задач и к ним — программы для ЭВМ. Описаны принципы построения и основные сведения, связанные с разработкой и использованием пакетов прикладных программ для решения инженерных и научных задач.

Для студентов вузов, обучающихся по специальностям «Автоматика и управление в технических системах» и «Прикладная математика»

М $\frac{2004010000-146}{M211(04)-89}$ 190-89

ISBN 5—11—001183—4

ББК 22.19я73

© Издательское объединение
«Выща школа», 1989

НТБ ДПИ
г. ВИННИЦА

ОГЛАВЛЕНИЕ

Предисловие	5
Глава 1. Математическое моделирование	8
1.1. Алгоритмы и программирование	9
1.2. Особенности построения математических моделей	13
1.3. Способы описания математических моделей	15
1.4. Погрешности вычислений, алгоритмов, математических моделей	41
1.5. Математическое моделирование и вычислительные методы в задачах автоматизации и информационно-измерительной техники	43
<i>Контрольные вопросы и задания</i>	45
Глава 2. Численные методы линейной алгебры	46
2.1. Основные сведения из теории матриц	46
2.2. Решение систем линейных уравнений	49
2.3. Решение задач нахождения собственных значений матриц	58
<i>Контрольные вопросы и задания</i>	70
Глава 3. Методы решения нелинейных уравнений и систем	72
3.1. Методы нахождения действительных корней нелинейного уравнения	74
3.2. Методы нахождения комплексных корней при решении алгебраических уравнений	80
3.3. Решение систем нелинейных уравнений	81
3.4. Выбор метода решения нелинейных уравнений и систем	87
<i>Контрольные вопросы и задания</i>	88
Глава 4. Решение обыкновенных дифференциальных уравнений	89
4.1. Постановка задачи	89
4.2. Методы решения задачи Коши	90
4.3. Методы решения краевых задач	102
<i>Контрольные вопросы и задания</i>	110

Глава 5. Методы решения дифференциальных уравнений и частных производных	111
5.1. Метод конечных элементов	112
5.2. Метод конечных разностей	115
5.3. Решение различных видов дифференциальных уравнений в частных производных	117
5.4. Общие рекомендации по решению дифференциальных уравнений в частных производных	124
<i>Контрольные вопросы и задания</i>	125
Глава 6. Методы обработки данных	
6.1. Методы интерполяции	126
6.2. Аппроксимация данных	136
6.3. Численное дифференцирование	138
6.4. Численное интегрирование	139
6.5. Статистическая обработка данных	149
<i>Контрольные вопросы и задания</i>	151
Глава 7. Пакеты прикладных программ для решения инженерных задач	151
7.1. Универсальные пакеты прикладных программ	155
7.2. Специализированные пакеты прикладных программ	157
7.3. Системы для аналитических вычислений на ЭВМ	158
7.4. Пакеты прикладных программ для моделирования систем	163
<i>Контрольные вопросы и задания</i>	165
Глава 8. Примеры решения инженерных задач на ЭВМ	165
Приложения	192
Список рекомендуемой литературы	212

ПРЕДИСЛОВИЕ

Процесс перестройки народно-хозяйственного механизма нашей страны требует подготовки высококвалифицированных инженерных и научных кадров, умеющих на практике использовать достижения научно-технической революции.

Главным направлением научно-технического прогресса в настоящее время является развитие методов и средств информатики и вычислительной техники. Использование методов математического моделирования и решения инженерных задач на ЭВМ позволяет значительно повысить эффективность процессов проектирования и управления.

Внедрение персональных компьютеров, расширение возможностей мини-ЭВМ, создание сверхмощных ЭВМ пятого поколения, развитие методов математического моделирования привели к расширению как практической, так и теоретической базы вычислительной математики. В последнее время, в частности, широкое распространение получили интервальный анализ, прикладная теория информации, теория оптимизации. Вычислительная математика, являясь одной из бурно развивающихся отраслей науки, находится в постоянном и взаимно обогащающем контакте с практикой. Существенный вклад в ее развитие внесли русские и советские ученые — А. Н. Крылов, Н. И. Лобачевский, П. Л. Чебышев и др. Так, основополагающие работы в теории разностных схем выполнены А. А. Самарским, им же разработана методология вычислительного эксперимента, ставшего в настоящее время одним из основных методов проведения научных исследований. Вычислительные методы в линейной алгебре были развиты в работах В. В. Воеводина и Д. К. Фаддеева, а приложения этих методов к решению кибернетических и оптимизационных задач — в работах Л. В. Канторовича и А. Н. Колмогорова. Большой вклад

в разработку численных методов и их практическое применение к задачам математической физики внесли А. А. Дородницын, М. В. Келдыш, В. И. Крылов, Г. И. Марчук, А. Н. Тихонов, А. А. Самарский, Н. Н. Яненко. Академики Е. П. Велихов, А. А. Дородницын, Г. И. Марчук, Б. Н. Наумов, Г. С. Поспелов, А. А. Самарский стоят во главе развития нового научного направления — информатики, объединяющего широкий круг вопросов, связанных с разработкой, созданием и использованием методов и средств математического моделирования, обработки информации и вычислений на ЭВМ. Методической основой для внедрения ЭВМ в процессе проектирования стали работы Н. П. Бусленко, В. М. Глушкова, И. В. Кузьмина, В. И. Скурихина, Б. Я. Советова, С. А. Яковлева.

Высокие требования к эффективности вычислительных алгоритмов определяются также и тем, что какими бы огромными возможностями ни обладали современные ЭВМ, их уже не хватает для проведения крупномасштабных прикладных исследований и управления сложными динамическими процессами. Спрос математиков-вычислителей пока превышает предложения, которые исходят от конструкторов и разработчиков ЭВМ. Стремительное развитие науки и техники приводит к быстрому росту сложности задач, стоящих перед прикладной математикой.

С развитием вычислительной математики неразрывно связано развитие программирования, которое идет по пути упрощения способов общения человека с ЭВМ, приближения его форм к естественным, привычным для человека. На современном этапе наряду с развитием и созданием новых языков высокого уровня интенсивно развиваются проблемно-ориентированные языки программирования, создаются пакеты прикладных программ, которые, основываясь на применении эффективных вычислительных алгоритмов, делают доступными для использования широкими кругами специалистов возможности современных ЭВМ. Осуществляется постепенный переход от эвристического программирования к программированию, подчиняющемуся четким законам и алгоритмам синтеза. Возникают и интенсивно развиваются структурное программирование и специализированные языки для разработки структурированных программ.

В настоящем учебном пособии рассмотрены наиболее распространенные в инженерной практике методы прикладной математики.

Первая глава посвящена анализу основных принципов теории алгоритмов и построения математических моделей. Большое внимание уделено современным методам математического моделирования, которые используются в анализе и синтезе систем и устройств автоматического измерения, контроля и управления. Следующие главы и разделы излагаются в последовательности, которая соответствует логике развития вычислительной математики, а также возрастающей сложности математического аппарата и используемых понятий. Материал проиллюстрирован примерами с программами и результатами решения на ЭВМ. В заключительных главах рассмотрены пакеты прикладных программ, используемые в процессе проектирования, моделирования и решения на ЭВМ инженерных задач, и некоторые типовые задачи измерительной техники, автоматики и управления, решаемые на ЭВМ методами вычислительной математики. Эти программы могут быть использованы студентами при выполнении курсовых и дипломных работ, а также аспирантами, инженерами и научными сотрудниками в своей научной и практической деятельности.

Излагаемый материал не сопровождается чрезмерно строгими доказательствами и математическими преобразованиями, а ориентирован на конечные результаты в виде алгоритмов решения инженерных задач на ЭВМ и основан на знаниях, полученных студентами в объеме курсов высшей математики и программирования.

Глава 1. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Методы моделирования широко используются в различных сферах человеческой деятельности, особенно в сферах проектирования и управления, где основными являются процессы принятия эффективных решений на основе получаемой информации.

Целью моделирования являются получение, обработка, представление и использование информации об объектах, которые взаимодействуют между собой и внешней средой; а модель здесь выступает как средство познания свойств и закономерностей поведения объекта.

Модель всегда строится с определенной целью, которая оказывает влияние на то, какие свойства объективного явления оказываются существенными, а какие — нет. Модель представляет собой как бы проекцию объективной реальности под определенным углом зрения. Иногда в зависимости от целей можно получить ряд проекций объективной реальности, вступающих в противоречие. Это характерно, как правило, для сложных систем, у которых каждая проекция выделяет существенное для определенной цели из множества несущественного.

Теория моделирования является разделом науки, изучающим способы исследования свойств объектов (оригиналов) на основе замещения их другими объектами (моделями) [21].

Остановимся на одном из наиболее универсальных видов моделирования — математическом, ставящем в соответствие моделируемому физическому процессу систему математических соотношений, решение которой позволяет получить ответ на вопрос о поведении объекта без создания физической модели, часто оказывающейся дорогостоящей и малоэффективной.

Многие явления и процессы разной природы описываются аналогичными соотношениями, например элек-

троакустическая аналогия, электро-, магнито- и гидродинамика. Поэтому для анализа (решения, расчета) математических моделей необходимо владеть развитым математическим аппаратом, охватывающим все виды типовых задач прикладной математики. Применительно к использованию ЭВМ основным этапом расчета математических моделей является их алгоритмизация, т. е. разработка структуры алгоритма, представленного в виде блок-схемы, граф-схемы или программно-реализованного с использованием принципов структурного программирования.

1.1. Алгоритмы и программирование

Термин «алгоритм» («алгорифм», «алгоризм») произошел от имени узбекского математика Ал-Хорезми, который в IX в. разработал правила арифметических действий над числами в десятичной системе.

Алгоритм — это правило, сформулированное на некотором языке и определяющее процесс переработки допустимых исходных данных в искомые результаты. Алгоритм характеризуется: детерминированностью (определенностью) — однозначностью результата процесса при заданных исходных данных; дискретностью — расчлененностью алгоритмического процесса на отдельные элементарные акты, возможность выполнения которых человеком или машиной не вызывает сомнений; массовостью — исходные данные для алгоритма можно выбрать из некоторого множества данных (потенциально бесконечного); понятностью для исполнителя.

В структуре алгоритмов можно выделить ряд блоков, представляющих собой совокупность элементарных операций, выполняющих определенную функцию. Типовыми блоками в структурной схеме алгоритма являются: процесс, решение, модификация, предопределенный процесс, ввод-вывод, соединитель, пуск-останов. Их условные обозначения представлены на рис. 1.1.

Алгоритм может иметь линейную, разветвляющуюся и циклическую структуры. Линейная структура характеризуется отсутствием условных блоков. На рис. 1.2 приведен пример алгоритма для решения задачи переместить мест содержимого двух ячеек памяти ЭВМ R и P . Причем оказывается необходимым использование третьей ячейки C . В качестве примера разветвляющейся

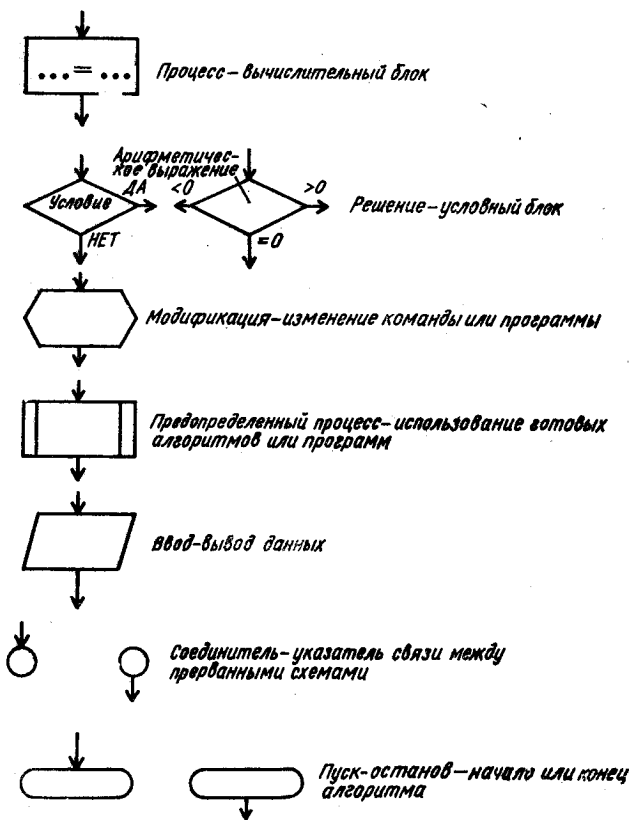


Рис. 1.1.



Рис. 1.2.

структуры алгоритма, имеющей хотя бы один условный блок, может служить алгоритм выбора наибольшего значения переменных N и M (рис. 1.3). Наиболее распространенной при решении инженерных задач является циклическая структура (рис. 1.4), где 1 — подготовка к первому выполнению тела цикла; 2 — тело цикла, которым называется многократно повторяющаяся часть вычислительного процесса; 3 — подготовка к очередному

выполнению тела цикла; 4 — управление циклом, осуществляющее проверку на окончание цикла.

Различают следующие виды циклов: с заданным либо вычисляемым числом повторений; итерационные, в ко-

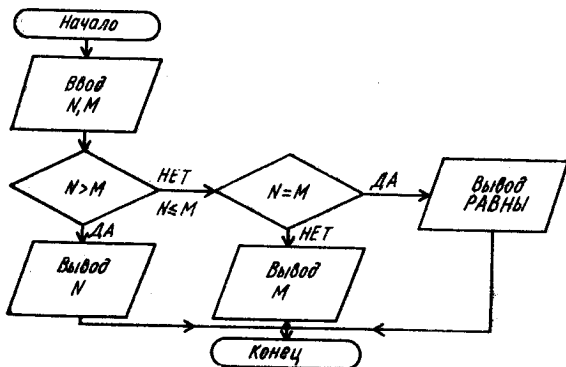


Рис. 1.3.

торых число повторений заранее неизвестно; сложные — с разветвлением в теле цикла и вложенными циклами (кратные).

Существуют и другие способы записи алгоритмов — в виде операторной или граф-схемы.

Для изображения операторной схемы алгоритма используются арифметические и логические операторы. Арифметические операторы производят действия, связанные с вычислениями.

Обозначим операторы заглавными буквами латинского алфавита с индексами, указывающими номер оператора. После выполнения операций, предусмотренных арифметическим оператором, процесс вычислений может быть продолжен по единственному пути, независимо от результатов, выдаваемых оператором. Передача управления от арифметического оператора обозначается номером того оператора, которому передается управление, записываемым справа сверху от символа данного оператора. Например, запись A_P^S означает, что от оператора A_P управление передается оператору с номером S .

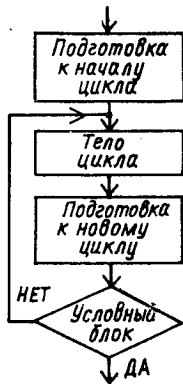
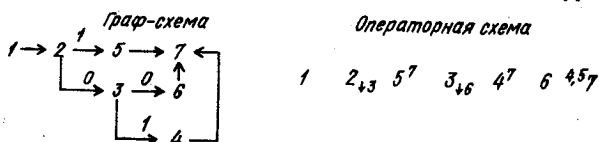


Рис. 1.4.

Логические операторы предназначены для проверки выполнения заданных условий. Обозначим их буквой P с указанием номера оператора. После реализации логического оператора управление передается одному из двух операторов в зависимости от выполнения проверяемого условия. Передача управления от логического оператора обозначается стрелками с номерами тех операторов, которым передается управление. Например, $P_{K,ij}$ означает, что от логического оператора P_K управление передается оператору с номером i , если условие, проверяемое оператором P_K , выполнено, или же оператору с номером j , если оно не выполнено. Для операторов, как арифметических, так и логических, обозначение передачи от одного оператора к другому, непосредственно за ним следующему, опускается.

Передача управления данному оператору от других обозначается номером того оператора, от которого передается управление, записываемым слева сверху от символа данного оператора. Например, запись $^{l,n}A_m$ означает, что оператору A_m управление передается от операторов с номерами l и n . В этом случае алгоритм, структура которого представлена на рис. 1.3, может быть записан следующим образом (блоки обозначаются цифрами):



В настоящее время все большую популярность приобретают методы структурного программирования, где в качестве единственно используемых трех видов структур алгоритмов применяются СОЧЛЕНЕНИЕ, ВЫБОР и ПОВТОРЕНИЕ. Основным методом создания программ при этом является алгоритм пошаговой детализации, в котором без составления блок-схемы программист постепенно передвигается по тексту программы, последовательно организуя и детализируя слои, соответствующие разным уровням абстракции, используя при этом специальный универсальный язык структурного программирования, например PDL. Составленная таким образом программа может быть легко и однозначно переведена на любой, удобный пользователю, язык программирования.

Нахождение алгоритмов решения различных классов задач — одна из целей математики. Цель же прикладной математики применительно к использованию ЭВМ — нахождение алгоритмов решения практических (инженерных) задач на ЭВМ.

В настоящем учебном пособии иллюстрирующие программы реализуются на языке программирования ФОРТРАН, специально предназначенном для решения инженерных задач и, несмотря на появление других, возможно, более универсальных языков (например, АЛГОЛ-68, ПЛ/1, ПАСКАЛЬ, ПРОЛОГ), продолжающим оставаться наиболее распространенным в инженерной практике.

1.2. Особенности построения математических моделей

Математические модели как проекции реальных объектов характеризуются рядом особенностей.

Математическое моделирование можно рассматривать как средство изучения реальной системы путем ее замены более удобной для экспериментального исследования системой (моделью), сохраняющей существенные черты оригинала. При моделировании производится аппроксимация функции описания более простой и удобной для практического анализа функцией — моделью.

Модель называется изоморфной (одинаковой по форме), если между нею и реальной системой существует полное поэлементное соответствие, и гомоморфной, если существует соответствие лишь между наиболее значительными составными частями объекта и модели.

Математическое моделирование включает следующие этапы: исследование объекта и составление его математического описания; построение алгоритма, моделирующего поведение объекта; проверку адекватности модели и объекта; использование модели.

Исследование объекта моделирования и составление его математического описания заключаются в установлении связей между параметрами процесса, выявлении его граничных и начальных условий и формализации процесса в виде системы математических соотношений.

Математическое описание составляется на основе законов физики, химии и пр., характеризующих дина-

мику и статику процессов в исследуемом объекте, и выражается на языке любых разделов математики. Наибольшее распространение при построении детерминированных моделей получили алгебраические уравнения, обыкновенные дифференциальные уравнения и дифференциальные уравнения в частных производных, матричная алгебра, а при стохастическом моделировании, когда учитывается случайный характер процессов, также и методы теории вероятностей и математической статистики. Если априорные сведения об объекте недостаточны, вид математических моделей уточняется с помощью методов многомерной статистики: регрессионного, корреляционного, многофакторного и других анализов, а также проведения пассивного или планирования активного экспериментов. По принципам построения модели разделяют на аналитические и имитационные. Аналитические модели позволяют получить явные функциональные зависимости для искомых величин или определить численные решения для конкретных начальных условий и количественные характеристики модели. Однако по мере усложнения объекта моделирования построение аналитической модели превращается в трудно-разрешимую проблему. В то же время широкое распространение получили имитационные модели, которые рассматриваются как проводимые на ЭВМ эксперименты с математическими моделями, имитирующими поведение реальных объектов. Особенности функционирования объектов моделирования и вид используемого математического описания определяют непрерывный или дискретный характер модели, цели проектирования — детерминированный или стохастический (вероятностно-статистический) подход к построению модели. Например, для моделирования функций преобразования измерительных устройств достаточно воспользоваться детерминированным способом описания, тогда как для анализа погрешностей, оценки информационных характеристик необходимо применить вероятностно-статистические методы.

Метод математического моделирования позволяет исключить необходимость изготовления громоздких физических моделей, связанную с материальными затратами; сокращать время определения характеристик (в особенности при расчете математических моделей на ЭВМ и применении эффективных вычислительных методов

и алгоритмов); изучать поведение объекта моделирования при различных значениях параметров; анализировать применимость различных элементов; получать характеристики и показатели, которые сложно снимать экспериментально (корреляционные, частотные, параметрической чувствительности).

Рассмотрим основные способы построения математических моделей, наиболее распространенные в автоматике, управлении, информационно-измерительной технике.

1.3. Способы описания математических моделей

В общем случае математическая модель устройства, системы, процесса представляется в виде системы функционалов

$$\Phi(X, Y, Z, t) = 0, \quad (1.1)$$

где X, Y — векторы входных и выходных координат; Z — вектор внешних воздействий; t — координата времени.

Способ представления Φ зависит от целей моделирования, назначения объекта, объема информации и характера исходных данных. В дальнейшем будем пользоваться для краткого определения вида модели следующими обозначениями:

первая буква: D — детерминированная, S — стохастическая;

вторая буква: N — непрерывная, D — дискретная;

третья буква: A — аналитическая, I — имитационная.

Детерминированные модели. Поведение большинства технических систем можно охарактеризовать при помощи так называемых фазовых переменных — физических величин типа потока и потенциала. При этом целесообразно выделить в объектах моделирования достаточно крупные элементы, рассматриваемые как неделимые единицы. Законы функционирования элементов системы задаются компонентными уравнениями, связывающими разнородные фазовые переменные.

Общность описания процессов, происходящих в различных технических системах, позволяет выделить несколько типов элементов: R — элемент рассеяния энергии; C и L — элементы накопления энергии. Сочетанием этих простейших элементов и источников фазовых переменных получают эквивалентную схему технической системы любой сложности и ее математическую модель.

Конкретный смысл фазовых переменных и простейших элементов физических систем приведен в табл. 1.1.

Математической моделью типа ДНА технического объекта в общем случае является система обыкновенных дифференциальных уравнений (1.1). Подобную систему можно решить только численными методами, заменяя непрерывную независимую переменную ее дискретным аналогом. Эта операция определяет изменение типа модели на ДДА.

Совокупность значений фазовых переменных и их производных на k -м шаге интегрирования получается как решение системы n алгебраических уравнений (в общем случае нелинейных) с n неизвестными x_1, x_2, \dots, x_n :

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0; \\ f_2(x_1, x_2, \dots, x_n) &= 0; \\ \dots \dots \dots \dots \dots \dots \dots \\ f_n(x_1, x_2, \dots, x_n) &= 0. \end{aligned}$$

Решение такой системы уравнений возможно итерационными методами, наиболее распространенным из которых является метод Ньютона. В основе этого метода лежит нахождение матрицы производных, называемой матрицей Якоби,

$$W(X) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Зависимости между неизвестными величинами фазовых переменных и их производными, входящими в систему уравнений (1.1), определяются с использованием разностных методов аппроксимации.

Система уравнений (1.1) представляет собой объединение компонентных, топологических и разностных уравнений. Топологическими уравнениями задается связь между однородными фазовыми переменными, относящимися к разным элементам системы. Такие уравнения в большинстве физических систем базируются на урав-

Таблица 1.1. Аналоги фазовых переменных и элементов

Подсистема	Фазовые переменные			Элементы		
	типа потока	типа потенциала	типа Р	типа С	типа L	
Электрическая	Ток	Напряжение	Сопротивление	Емкость	Индуктивность	
Механическая поступательная	Сила	Скорость	Трение	Масса	Упругость	
Механическая вращательная	Момент	Угловая скорость	»	Момент инерции	Вращательная гибкость	
Гидравлическая (пневматическая)	Расход	Давление	»	Гидравлическая емкость	Гидравлическая индуктивность	
Тепловая	Тепловой поток	Температура	Теплопроводность	Теплоемкость	—	

НТБ ВПИ
г. ВИННИЦА

нениях равновесия и непрерывности (например, система уравнений первого и второго законов Кирхгофа).

В качестве примера рассмотрим математическую модель биполярного транзистора.

Эквивалентная схема биполярного транзистора представлена на рис. 1.5. Здесь приняты следующие обозначения: $I_{\text{Э}}$, $C_{\text{Э}}$, $R_{y\text{Э}}$, $I_{\text{К}}$, $C_{\text{К}}$, $R_{y\text{К}}$ — соответственно элементы p - n переходов эмиттер — база и коллектор — база; $I_r = \beta I_{\text{Э}} - \beta_n I_{\text{К}}$ — источник тока, отражающий пролет несомненных носителей через базу и определяющий усилительные свойства транзистора (β и β_n — нормальный и инверсный коэффициенты усиления тока); $r_{\text{Э}}$, $r_{\text{К}}$ и $r_{\text{Б}}$ — объем-

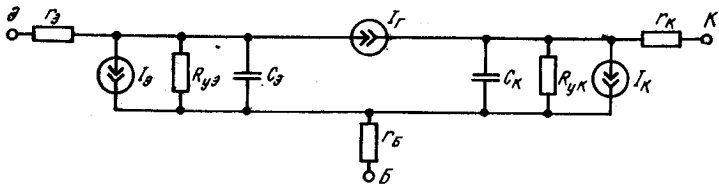


Рис. 1.5.

ные сопротивления областей соответственно эмиттера, коллектора и базы.

Запишем компонентные уравнения для каждого элемента. Получим следующую систему уравнений:

$$\left. \begin{aligned}
 I_{r\text{Э}} - U_{r\text{Э}}/r_{\text{Э}} &= 0; \\
 I_{R_{y\text{Э}}} - U_{\text{Э}}/R_{y\text{Э}} &= 0; \\
 I_{\text{Э}} - I_{T\text{Э}} \exp(U_{\text{Э}}/m\varphi_{T\text{Э}} - 1) &= 0; \\
 I_{C_{\text{Э}}} - [C_{B\text{Э}} + \tau/(m\varphi_{T\text{Э}})(I_{\text{Э}} + I_{T\text{Э}})] \dot{U}_{C_{\text{Э}}} &= 0; \\
 I_r - \beta I_{\text{Э}} + \beta_n I_{\text{К}} &= 0; \\
 I_{C_{\text{К}}} - [C_{B\text{К}} + \tau_n/m\varphi_{T\text{К}}(I_{\text{К}} + I_{T\text{К}})] \dot{U}_{C_{\text{К}}} &= 0; \\
 I_{\text{К}} - I_{T\text{К}} \exp(U_{\text{К}}/m\varphi_{T\text{К}} - 1) &= 0; \\
 I_{R_{y\text{К}}} - U_{\text{К}}/R_{y\text{К}} &= 0; \\
 I_{r\text{К}} - U_{r\text{К}}/r_{\text{К}} &= 0; \\
 I_{r\text{Б}} - U_{r\text{Б}}/r_{\text{Б}} &= 0,
 \end{aligned} \right\} (1.2)$$

где $I_{T\text{Э}}$ — тепловой ток перехода база — эмиттер; m — эмпирический коэффициент; $\varphi_{T\text{Э}}$ — температурный потенциал эмиттера; $C_{B\text{Э}}$ — барьерная емкость перехода база — эмиттер; $C_{B\text{К}}$ — барьерная емкость перехода база — коллектор; $\varphi_{T\text{К}}$ — температурный потенциал коллектора; $I_{T\text{К}}$ — теп-

ловой ток перехода база — коллектор; τ_n, τ — параметры, характеризующие время пролета носителей тока через области транзистора.

Неизвестными переменными здесь являются $I_{r\varepsilon}, I_{Ry\varepsilon}, I_\varepsilon, I_{c\varepsilon}, I_r, I_{CK}, I_K, I_{RyK}, I_{rK}, I_{rB}, U_{r\varepsilon}, U_\varepsilon, U_{c\varepsilon}, U_{CK}, U_K, U_{rK}, U_{rB}, U_r$. Из этого перечня следует, что в модели учтены некоторые топологические уравнения: исключены величины $U_{c\varepsilon}$ и U_{CK} , U_R и U_{RK} , совпадающие соответственно с величинами U_ε и U_K .

Запишем топологические уравнения системы:

$$\left. \begin{aligned} I_{r\varepsilon} - I_{Ry\varepsilon} - I_\varepsilon + I_{c\varepsilon} - I_r &= 0; \\ I_r + I_{CK} - I_K - I_{RK} - I_{rK} &= 0; \\ I_{R\varepsilon} + I_\varepsilon - I_{c\varepsilon} - I_{CK} + I_K + I_{RyK} - I_{rB} &= 0; \\ U_\varepsilon - U_K - U_r &= 0; \\ U_{r\varepsilon} + U_\varepsilon + U_{rB} - U_{B\varepsilon} &= 0; \\ -U_K + U_{rK} - U_{rB} + U_{BK} &= 0. \end{aligned} \right\} (1.3)$$

В двух последних уравнениях $U_{B\varepsilon}$ и U_{BK} — величины напряжений соответственно база — эмиттер и база — коллектор. В систему включаются разностные аппроксимации для производных $U_{c\varepsilon}, U_{CK}$ с шагом h .

Таким образом, математическая модель биполярного транзистора представляет собой систему уравнений (1.2), (1.3). Матрица Якоби для этой системы представлена в табл. 1.2 (нулевые элементы не обозначены).

В этой матрице приняты следующие обозначения коэффициентов:

$$\begin{aligned} a_1 &= -\frac{I_{T\varepsilon}}{m\varphi_{T\varepsilon}} \exp [U_\varepsilon / (m\varphi_{T\varepsilon})]; \\ a_2 &= -\frac{\tau}{m\varphi_{T\varepsilon}} U_{c\varepsilon}; \\ a_3 &= -[C_{B\varepsilon} + \frac{\tau}{m\varphi_{T\varepsilon}} (I_\varepsilon + I_{T\varepsilon})]; \\ a_4 &= -\frac{\tau_n}{m\varphi_{TK}} U_{CK}; \\ a_5 &= -[C_{BK} + \frac{\tau_n}{m\varphi_{TK}} (I_K + I_{TK})]; \\ a_6 &= -\frac{I_{TK}}{m\varphi_{TK}} \exp [U_K / (m\varphi_{TK})]. \end{aligned}$$

Таблица 1.2. Матрица Якоби для математической модели биполярного транзистора]

№ переменной	Переменная																	
	I_{r2}	I_{Ry2}	I_2	I_{c2}	I_r	I_{cK}	I_K	I_{RyK}	I_{rK}	I_{rB}	U_2	U_3	\bar{U}_{c2}	\bar{U}_{cK}	U_K	U_{rK}	U_{rB}	U_r
1	1										$-1/g_2$							
2		1									$-1/g_2$							
3			1								a_1							
4			a_2	1								a_3						
5			$-b$		1		$b\eta$											
6						1	a_4							a_5				
7							1											
8								1									a_6	
9									1								$-1/R_{yK}$	
10										1							$-1/R_K$	
11	1	-1	-1	1	-1													
12					1	1	-1	-1	-1									
13		1	1	-1		-1	1	1		-1								
14											1				-1			-1
15										1	1						1	
16															-1	1	-1	
17											$-1/h$	1						
18														1	$-1/h$			

Стохастические модели. При описании объектов автоматики и информационно-измерительной техники можно выделить следующие виды стохастического (вероятностного) моделирования.

Статистическое моделирование, которое также называют методом Монте-Карло, имеет приложение кроме задач математического моделирования к решению ряда частных задач численных методов, например, взятия интегралов и решения дифференциальных уравнений. Статистические модели случайных процессов могут быть реализованы как на обычных ЭВМ (аналоговых и цифровых), так и на специализированных статистических ЭВМ, снабженных моделирующими блоками для выработки и преобразования случайных чисел.

Аналитическое вероятностное моделирование на основе построения моделей, оперирующих не с конкретными случайными числовыми последовательностями, а непосредственно с их вероятностными (законы распределения вероятностей) и спектральными (спектральные плотности или корреляционные функции) характеристиками [14]. В общем случае построение аналитических вероят-

ностных моделей представляет собой сложную вычислительную задачу, что не позволяет в полной мере использовать такие их преимущества, как возможность точного аналитического задания характеристик случайных процессов, отсутствие необходимости генерации и обработки больших выборок случайных чисел, приспособленность к оперативной оптимизации. В литературе описаны результаты исследований, направленных на создание специализированных проблемно-ориентированных систем и пакетов прикладных программ, объединяющих численные алгоритмы решения наиболее характерных вычислительных процедур аналитического вероятностного моделирования и способы описания структур систем [9]. Создание такого программного обеспечения наряду с разработкой специализированных моделирующих комплексов позволит значительно повысить эффективность построения и использования аналитических моделей и на этой основе решения задач стохастического моделирования.

В настоящее время основным методом стохастического моделирования является метод статистического моделирования на ЭВМ, оперирующий с моделями типа СДИ.

Методика статистического моделирования включает ряд последовательных этапов: моделирование на ЭВМ псевдослучайных числовых последовательностей с заданной корреляцией и законом распределения вероятностей, имитирующих входные сигналы и возмущающие воздействия в системе; моделирование преобразования полученных числовых последовательностей в системе; статистическая обработка результатов моделирования. Рассмотрим перечисленные этапы.

Моделирование на ЭВМ псевдослучайных числовых последовательностей с заданными характеристиками. При построении имитационной модели системы возникает задача получения на ЭВМ псевдослучайных числовых последовательностей с заданной корреляцией и законом распределения вероятностей. Известен метод получения числовых последовательностей с заданными статистическими характеристиками с помощью сортировки исходных последовательностей. Метод основан на том, что коэффициент корреляции случайных чисел больше зависит от порядка их следования, чем от величины. Поэтому две псевдослучайные последовательности, соответствующие двум разным распределениям,

в случае, если они упорядочены одинаковым образом, будут иметь примерно равные коэффициенты корреляции.

В соответствии с методом сортировки генерируется псевдослучайная последовательность $X(n)$ с заданной корреляционной функцией, но произвольным распределением. В соответствие ей ставится последовательность целых чисел $I(n) = n$. Затем обе последовательности попарно сортируются. При этом величины $X(n)$ располагаются по возрастанию, а массив $I(n)$ запоминает их предыдущее положение (места в неупорядоченном массиве $X(n)$). Таким образом, целочисленный массив $I(n)$ отображает корреляцию между элементами массива $X(n)$. После упорядочения массив $X(n)$ интереса не представляет, так как вся информация о корреляционной функции теперь содержится в массиве $I(n)$. Затем генерируется псевдослучайная последовательность $Y(n)$ с заданным распределением и нулевой корреляцией и записывается на место массива $X(n)$. После этого она сортируется в увеличивающемся порядке. Далее массивы $I(n)$ и $Y(n)$ попарно сортируются, причем массив $I(n)$ располагается в возрастающем порядке. Структурная схема алгоритма приведена на рис. 1.6.

В результате выполнения данного алгоритма получим псевдослучайную числовую последовательность, содержащую величины, распределенные по заданному закону и имеющие заданную корреляционную функцию.

Алгоритм сортировки целесообразно использовать в случаях, когда для статистического моделирования системы имеется небольшой объем статистических данных, не требующих оперирования числовыми массивами большой размерности. При большой размерности массивов существенно увеличивается время моделирования.

Известен алгоритм фильтрации, требующий меньших затрат машинного времени для получения случайного процесса с заданной корреляцией и законом распределения вероятностей. В качестве исходного выбирается нормальный стационарный случайный процесс $X(t)$. Всегда существует такое нелинейное безынерционное преобразование $Y = W_N(X)$, которое превращает нормальную функцию плотности $f_x(x)$ процесса $X(t)$ в заданную функцию плотности $f_y(y)$. Если исходный процесс $X(t)$ имеет корреляционную функцию $R_{XX}(\tau)$, то преобразованный процесс $Y(t)$ будет иметь корреляционную функцию

$R_{YY}(\tau)$, отличающуюся от функции $R_{XX}(\tau)$ и связанную с ней некоторой зависимостью $R_{YY} = \varphi(R_{XX})$. Вид этой зависимости определяется преобразованием $Y = W_N(X)$. Для того чтобы корреляционная функция преобразованного процесса была требуемой, необходимо выбрать корреляционную функцию исходного процесса:

$$R_{XX}(\tau) = \varphi^*[R_{YY}(\tau)],$$

где φ^* — функция, обратная φ .

При использовании этого способа подготовительная работа состоит из нескольких этапов:



Рис. 1.6.

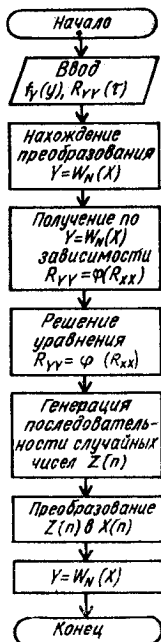


Рис. 1.7.

нахождения функции преобразования $Y = W_N(X)$ по заданной функции плотности $f_Y(y)$;

получения по найденной функции $Y = W_N(X)$ зависимости $R_{YY} = \varphi(R_{XX})$;

решения уравнения $R_{YY} = \varphi(R_{XX})$ относительно R_{XX} , т. е. определения корреляционной функции $R_{XX}(\tau)$ исходного нормального процесса $X(t)$.

После окончания подготовительной работы моделирование случайного процесса с заданными характеристиками сводится к формированию дискретных реализаций $X(n)$ нормального случайного процесса $X(t)$ и к преобразованию этих реализаций по формуле

$$Y(n) = W_N \{X(n)\}.$$

Структурная схема описанного алгоритма приведена на рис. 1.7.

Описанный алгоритм требует меньших затрат машинного времени, чем алгоритм сортировки, не требует накопления и хранения в памяти числовых массивов большой размерности. Недостатками метода являются его сложность для программирования и большой объем подготовительной работы. Затруднение принципиального характера состоит в том, что в общем случае не представляется возможным доказать существование решения уравнения $R_{YY}(\tau) = \varphi[R_{XX}(\tau)]$ относительно $R_{XX}(\tau)$.

При использовании обоих описанных алгоритмов возникает задача генерирования на ЭВМ псевдослучайных числовых последовательностей с заданными законами распределения и нулевой корреляцией и псевдослучайных числовых последовательностей с заданной корреляционной функцией и произвольным распределением.

Задачу генерирования случайных чисел с заданным законом распределения решают в несколько этапов. Вначале получают последовательность равномерно распределенных на интервале $[0, 1]$ псевдослучайных чисел, а из нее — последовательность псевдослучайных чисел с заданным законом распределения.

Рассмотрим алгоритмические (программные) методы получения равномерно распределенных случайных чисел (на практике иногда применяют также физическое моделирование с использованием специальной приставки к ЭВМ). Сущность программных методов заключается в том, что равномерно распределенные псевдослучайные числа получают с помощью некоторой рекуррентной формулы, где каждое следующее $(i + 1)$ -е значение образуется из предыдущего i (или группы предыдущих) путем применения некоторого алгоритма, содержащего логические и арифметические операции.

Известно большое количество способов имитации равномерного распределения (методы вычетов, сумми-

рования, усечения, перемешивания) [10]. Общими для всех этих методов являются требования, предъявляемые к генерируемой последовательности равномерно распределенных случайных чисел: количество операций для получения каждого псевдослучайного числа должно быть минимальным; случайные числа генерируются как можно менее коррелированными, а их распределение — близким к равномерному, причем вид распределения и степень коррелированности чисел не должны меняться в процессе работы программы.

В стандартном математическом и программном обеспечении различных типов ЭВМ имеются специальные процедуры и подпрограммы для генерации равномерно распределенных последовательностей псевдослучайных чисел.

Используя равномерно распределенную в интервале $[0, 1]$ случайную величину X , можно получить последовательность случайных чисел с произвольным заданным законом распределения вероятностей. Различают три основных способа формирования таких последовательностей:

1) прямое преобразование числа X_i , являющегося реализацией случайной величины X , равномерно распределенной на интервале $[0, 1]$ с помощью некоторой функции W_N , в число y_i , которое может рассматриваться как реализация случайной величины Y , имеющей заданный закон распределения;

2) отсеивание чисел из первоначальной последовательности равномерно распределенных на интервале $[0, 1]$ псевдослучайных чисел таким образом, чтобы оставшиеся числа были распределены по заданному закону;

3) моделирование условий соответствующих предельных теорем теории вероятностей.

Рассмотрим их особенности.

Пусть $X(n)$ — последовательность равномерно распределенных на интервале $[0, 1]$ чисел с плотностью вероятности $f_X(x)$. Необходимо получить последовательность псевдослучайных чисел с заданной плотностью вероятностей $f_Y(y)$ с помощью некоторого преобразования $Y = W_N(X)$, где W_N — однозначная монотонная функция X .

Законы распределения случайных величин при нелинейном преобразовании связаны соотношением

$$f_Y(y) = \frac{d}{dy} \dot{W}_N(y) f_X[\dot{W}_N(y)], \quad (1.4)$$

где \dot{W}_N — функция, обратная W_N .

Решая уравнение (1.4) относительно неизвестного W_N , получим нелинейное преобразование, которое необходимо применять к каждому случайному числу x_i для получения последовательности $\{y_i\}$, распределенной по заданному закону. Однако в большинстве случаев уравнение (1.4) относительно W_N точно неразрешимо. Поэтому на практике часто применяют приближенные приемы преобразования случайных чисел (например, метод кусочной аппроксимации функций плотности).

Второй способ получения последовательностей псевдослучайных чисел с заданным законом распределения вероятностей, основанный на отсеивании чисел из первоначальной последовательности $X(n)$, реализуется по следующему алгоритму:

из исходной последовательности псевдослучайных чисел $X(n)$ выбираются пары чисел x_{2l-1}, x_{l-1} ;

далее осуществляется проверка следующего неравенства для этих чисел:

$$x_{2l} \leq \frac{1}{f_{Y_{\max}}} f_Y[a + (b - a)x_{2l-1}],$$

где $a = y_{\min}$; $b = y_{\max}$;

в случае выполнения последнего неравенства

$$y_l = a + (b - a)x_{2l-1};$$

если неравенство не выполняется, то описанная процедура повторяется для следующей пары, и так до тех пор, пока неравенство не станет справедливым. Пары, не удовлетворяющие неравенству, отбрасываются и не дают следующего значения y_l . Реализация этого способа показана на рис. 1.8.

К третьему способу моделирования псевдослучайных чисел относится группа приемов преобразования исходной последовательности случайных чисел, основанная на приближенном воспроизведении условий, при которых справедливы соответствующие предельные теоремы теории вероятностей. Например, центральная предельная теорема позволяет получить нормальное распределение последовательности случайных чисел $Y(n)$, рассматривая их как суммы k последовательных случайных чисел с произвольным законом распределения вероятностей.

Способ, основанный на использовании предельных теорем теории вероятностей, более быстродействующий, но менее точный, он позволяет сформировать лишь предельные законы распределения. Основным его недостатком является необходимость поиска экстремума моделируемого закона, что значительно ухудшает быстродействие алгоритма и ограничивает область его применения одномерными законами. Способ, основанный на решении уравнения (1.4), является более точным, но менее быстродействующим.

Широкое распространение получают способы ускоренной генерации случайных чисел. Так, значительный эффект увеличения быстродействия при имитации случайных чисел, распределенных по нормальному закону, по сравнению со способом, основанным на использова-

нии центральной предельной теоремы теории вероятностей, дает способ, включающий предложенный Мюллером алгоритм, с помощью которого пара независимых равномерно распределенных на отрезке $[0, 1]$ чисел x_1, x_2 преобразовывается в пару независимых нормально распределенных величин

$$\xi_1 = \sqrt{-2 \ln x_2} \cos 2\pi x_1; \xi_2 = \sqrt{-2 \ln x_1} \cos 2\pi x_2.$$

Отметим, что этот способ является теоретически точным и требует наименьшего количества равномерно распределенных чисел $X(n)$. Его программная реализация представлена на рис. 1.9.

Известны специальные способы получения случайных чисел, подчиняющихся ряду законов распределения вероятностей. Например, распределение Релея определяется одним параметром σ , равным среднему квадратическому отклонению исходного двумерного нормального распределения. Отсюда вытекает следующий способ имитации распределения Релея:

$$\eta_i = \sigma \sqrt{\xi_1^2 + \xi_2^2},$$

где η_i — случайное число, распределенное по закону Релея, а ξ_1 и ξ_2 — случайные числа, имеющие нормальное распределение с нулевым математическим ожиданием и равным единице средним квадратическим отклонением.

Существует также соотношение, связывающее случайные числа, распределенные по закону Релея, со случайными числами x_i , равномерно распределенными на отрезке $[0, 1]$, которое определяет другой способ генерации:

$$\eta_i = \sigma \sqrt{-2 \ln x_i}.$$

Для имитации закона распределения Максвелла можно воспользоваться тем, что случайная величина, подчиняющаяся распределению Максвелла, может рассматриваться как модуль трехмерного случайного вектора, проекции которого на оси координат подчиняются нормальному распределению с равными между собой средним квадратическим отклонением и математическими ожиданиями, равными нулю. Поэтому можно воспользоваться следующей формулой для имитации закона распределения Максвелла:

$$\theta_i = \sigma \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2},$$

```

PROGRAM MAL
      REAL X(1000)
      PI=3.1415926
      I1=13
      I2=17
      N=0
10    R1=RAN(I1, I2)
      R2=RAN(I1, I2)
      X(N+1)=SQRT(-2.*ALOG(R2))*
&      COS(2.*PI*R1)
      X(N+2)=SQRT(-2.*ALOG(R1))*
&      COS(2.*PI*R2)
      N=N+2
      IF(N.LT.1000)GOTO 10
      CALL GST(X, 1000, 7)
      STOP
      END

```

```

MX = 2.1615995E-02
DX = 0.9916891
SKO= 0.9958359
      ЛЕВАЯ ГРАНИЦА ГИСТОГРАММЫ:
      -2.965892
      ПРАВАЯ ГРАНИЦА ГИСТОГРАММЫ:
      3.009124

```

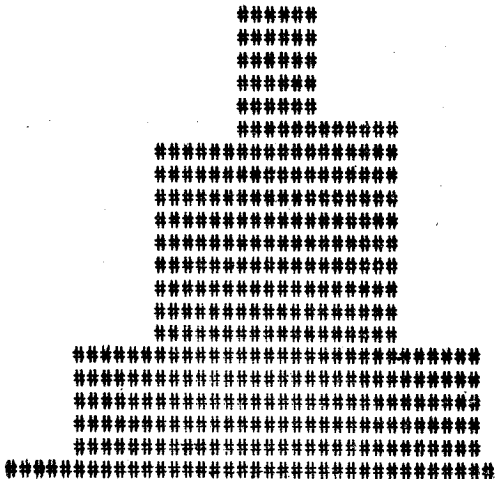


Рис. 1.9.

где θ_i — случайное число, распределенное по закону Максвелла; ξ_1, ξ_2, ξ_3 — случайные числа, имеющие нормальное распределение с математическими ожиданиями, равными нулю, и средними квадратическими отклонениями, равными единице.

При анализе объектов контроля, погрешностей измерений и оценке качества изделий методами статистического моделирования часто возникает необходимость вводить в ЭВМ экспериментально полученные законы распределения случайных величин в виде гистограмм распределений. В связи с этим остановимся на моделировании случайных величин, подчиняющихся закону распределения, выражаемому в виде экспериментально снятых

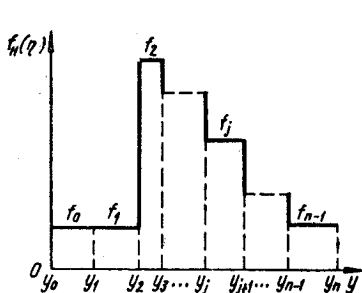


Рис. 1.10.

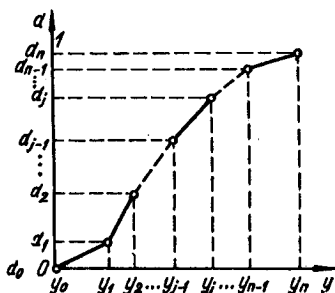


Рис. 1.11.

гистограмм. Ступенчатая плотность вероятностей (рис. 1.10), соответствующая кусочно-линейному интегральному распределению (рис. 1.11), является еще и удобной аппроксимацией для плотностей сложной формы.

Обозначим

$$P_j = f_j (y_j - y_{j-1}), \quad j = \overline{1, n},$$

вероятности

$$P(y_{j-1} < n \leq y_j).$$

Тогда

$$d_j = \sum_{i=1}^j P_i.$$

Воспользовавшись соотношением, связывающим случайные числа η с плотностью распределения $f_n(\eta)$ и числа x_i , имеющие равномерное распределение на отрезке $[0, 1]$, можно записать

$$\eta_i = \frac{y_j - y_{j-1}}{P_j} (x_i - d_{j-1}) + y_{j-1}, \quad (1.5)$$

если

$$d_{j-1} < x_i < d_j.$$

Выражение (1.5) получим решением относительно η_i уравнения (1.4) с учетом специфики рассматриваемого закона распределения. Процедура нахождения соответствующего интервала $[d_{j-1}, d_j]$ сводится к следующему. Интервал $[0, 1]$, расположенный на оси ординат (рис. 1.11), разбивается на интервалы, длины которых составляют P_1, P_2, \dots, P_n , а координаты точек деления $d_1 = P_1, d_2 = P_1 + P_2, \dots, d_n = P_1 + P_2 + \dots + P_n$. Следовательно, каждый раз, моделируя равномерно распределенное случайное число x_i , можно установить принадлежность его к интервалам $[d_{j-1}, d_j]$, образующим совокупность непересекающихся интервалов, принадлежащих интервалу $[0, 1]$. Далее в соответствии с формулой (1.5) определяется значение величины η_i .

Рассмотрим способы получения псевдослучайных числовых последовательностей с заданной корреляционной функцией. Известен алгоритм, где n некоррелированных случайных чисел x_1, \dots, x_n подвергают такому линейному преобразованию, после которого полученные величины y_1, \dots, y_n имеют заданную корреляционную матрицу $K(R_{nn})$.

Таким образом, величины y_1, \dots, y_n находятся из матричного уравнения

$$K(Y) = W_N [K(X)],$$

где W_N — линейное преобразование вектора-столбца $K(X)$ в $K(Y)$.

В развернутом виде получаем:

$$\begin{aligned} y_1 &= a_{11}(x_1 - m_X) + m_{Y_1}; \\ y_2 &= a_{12}(x_1 - m_X) + a_{22}(x_2 - m_X) + m_{Y_2}; \\ &\dots \\ y_n &= a_{1n}(x_1 - m_X) + a_{nn}(x_n - m_X) + m_{Y_n}, \end{aligned}$$

где коэффициенты преобразования a_{lk} определяются из уравнения вида

$$R_{lk} = a_{1l}a_{1k} + a_{2l}a_{2k} + \dots + a_{ll}a_{lk},$$

а $K[m_Y]$ — вектор-столбец математических ожиданий y_i .

При больших n изложенный способ генерирования коррелированных псевдослучайных последовательностей становится неудобным для реализации на ЭВМ, так как запоминание элементов матрицы $K(a)$ требует большого объема оперативной памяти ($N = n(n+1)/2$ ячеек) и большого объема вычислений (затрат машинного времени). В связи с этим в ряде случаев оказывается более удобным моделирование коррелированных случайных процессов по методу канонических разложений.

Пусть непрерывный случайный процесс $Y(t)$ задан каноническим разложением

$$Y(t) = \sum_{k=1}^{\infty} V_k J_k(t), \quad (1.6)$$

где V_k — некоррелированные случайные коэффициенты; $J_k(t)$ — система некоторых детерминированных координатных функций.

Цифровое моделирование случайного процесса, заданного каноническим разложением, осуществляется следующим образом. В качестве V_k используются значения некоррелированных случайных величин x_k , а бесконечный ряд (1.6) при вычислениях приближенно заменяется усеченным конечным рядом. Используя каноническое разложение, получим соотношение для моделирования

$$Y_n = \sum_{k=1}^N V_k J_k(n),$$

в котором дисперсии D_{V_k} некоррелированных случайных величин x_k и дискретные координатные функции $J_k(n)$ находятся из следующих рекуррентных соотношений:

$$D_{V_k} = R_{YY}(1, 1);$$

$$J_1(n) = \frac{1}{D_{V_1}} R_{YY}(n, 1);$$

$$D_{V_k} = R_{YY}(k, k) - \sum_{i=1}^{k-1} D_{V_i} J_i(k), \quad k = \overline{2, N};$$

$$J_k(n) = \frac{1}{D_{V_k}} \left[R_{YY}(n, k) - \sum_{i=1}^{k-1} D_{V_i} J_i(n) J_i(k) \right].$$

Отсюда получаем

$$Y_n = \sum_{k=1}^N \sigma_k x_k J_k(n).$$

Последовательность Y_n будет иметь нормальное распределение и заданную корреляционную функцию $R_{XX}(\tau)$.

Методы сортировки и фильтрации в сочетании с рассмотренными способами генерирования псевдослучайных числовых последовательностей с заданными корреляционными функциями позволяют получить последо-

вательности случайных чисел, имитирующие входные сигналы и возмущающие воздействия в объекте моделирования.

Моделирование процессов преобразования случайных сигналов. Рассмотрим способы построения алгоритмов для моделирования процессов преобразования случайных сигналов различными преобразователями и системами. При моделировании преобразования случайных сигналов линейными динамическими системами удобно пользоваться их импульсной характеристикой $g(t)$.

Для получения цифровой модели преобразования входного случайного сигнала необходимо моделировать операцию свертки функций $x(t)$ и $g(t)$.

Осуществляя замену интеграла Дюамеля суммой дискретных значений по методу прямоугольников, получаем

$$y(n) = \sum_{k=0}^{n-m} g(k) x(n-k), \quad (1.7)$$

где m — дискретный аналог длительности переходного процесса. Программная реализация этого метода представлена на рис. 1.12.

Существуют и более точные методы интегрирования — трапеций, Симпсона и др. В этих случаях формула (1.7) принимает вид

$$y(n) = \sum_{k=0}^n c(k) g(k) x(n-k),$$

где коэффициенты $c(k)$ определяются в зависимости от используемого метода численного интегрирования.

Существенным преимуществом метода моделирования линейных динамических систем на основе дискретного аналога интеграла свертки является возможность обобщения его на случай моделирования линейных динамических систем с переменными параметрами (нестационарных систем).

Нестационарная система описывается импульсной характеристикой $g(t, \tau)$, зависящей от двух переменных. В этом случае реакцию нестационарной системы на случайный сигнал запишем в виде

$$y(t) = \int_0^t x(\tau) g(\tau, t-\tau) d\tau,$$

С
С
С
С
С

ПРОГРАММА МОДЕЛИРОВАНИЯ
ЛИНЕЙНОГО
ДИНАМИЧЕСКОГО ЗВЕНА.

```

PROGRAM LDZ
  REAL X(100)
  REAL Y(100)
  DATA X/100*1.0/
  DATA Y(1)/0./
  H(T)=2.5*EXP(-T/2.)
  DT=0.5
  DO 20 N=2,100
    Y(N)=0
    DO 10 K=1,N
      Y(N)=Y(N)+X(K)*
      H((N-K)*DT)*DT
    &
  CONTINUE
  CONTINUE
  ПОСТРОЕНИЕ ГРАФИКА.
  CALL GRAF(X,Y,40)
  STOP
  END

```

10
20
С

```

SUBROUTINE GRAF(Y,Z,NZ)
  BYTE S(51)
  REAL Y(NZ)
  REAL Z(NZ)
  INTEGER IY(60)
  INTEGER IZ(60)
  DATA S/50*' ',' ',' ' /
  YMIN=AMIN1(Y(1),Z(1))
  YMAX=AMAX1(Y(1),Z(1))
  DO 20 I=1,NZ
    IF(Y(I).GT.YMAX)YMAX=Y(I)
    IF(Z(I).GT.YMAX)YMAX=Z(I)
    IF(Y(I).LT.YMIN)YMIN=Y(I)
    IF(Z(I).LT.YMIN)YMIN=Z(I)
  20 CONTINUE
  DO 30 I=1,NZ
    IY(I)=(Y(I)-YMIN)*
    50./(YMAX-YMIN)+1
    &
    IZ(I)=(Z(I)-YMIN)*
    50./(YMAX-YMIN)+1
    &
  30 CONTINUE
  IOS=-YMIN*50./(YMAX-YMIN)+1
  PRINT1010
  DO 50 I=1,NZ
    S(IOS)='I'
    S(IY(I))='X'
    S(IZ(I))='Y'
    PRINT1000,S
    DO 40 J=1,51
      S(J)=' '
    40 CONTINUE

```

20

30

40

```

50      CONTINUE
        S(IOS)='U'
        PRINT1000,S
        TYPE*,YMIN,YMAX
        RETURN
1000    FORMAT(X,51A1)
1010    FORMAT('1')
        END

```

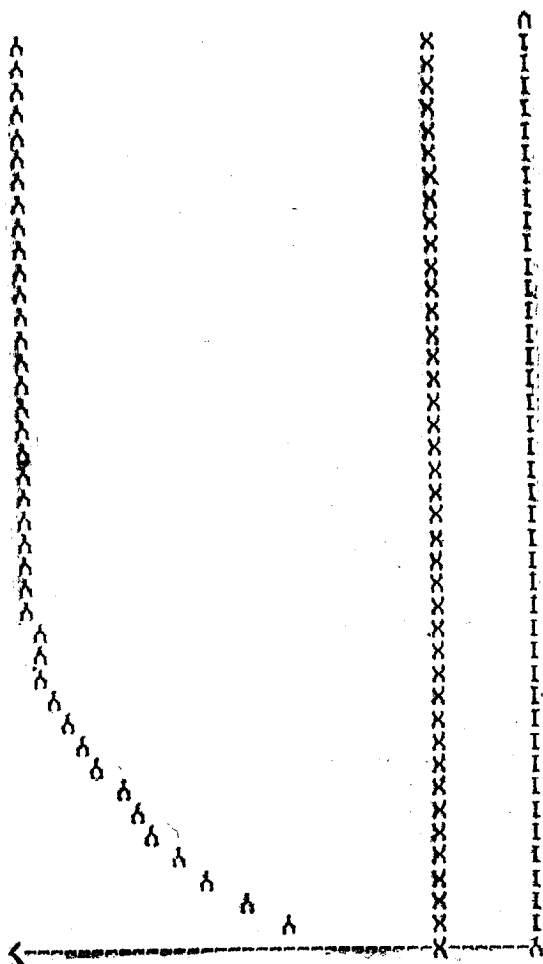


Рис. 1.12.

или в дискретной форме

$$y(n) = \sum_{k=0}^{n-1} x(k) g(k, n-k) c(n-k).$$

Описанный метод применяется в тех случаях, когда моделирование входного случайного сигнала осуществляется с помощью алгоритмов нелинейного преобразования, фильтрации и моделирования условий предельных

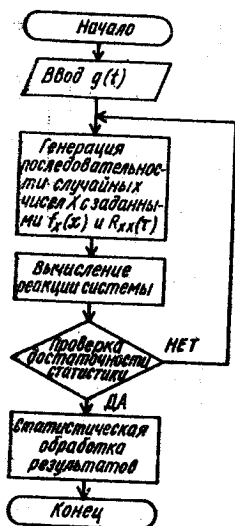


Рис. 1.13.

теорем теории вероятностей. Структурная схема алгоритма моделирования представлена на рис. 1.13. Отметим, что в случае моделирования входного случайного сигнала на основе алгоритма сортировки (см. рис. 1.6) построить имитационную модель системы невозможно и исследовать ее целесообразно матричными методами. Цифровые модели замкнутых нелинейных систем, являющихся комбинированными комбинациями линейных динамических и нелинейных статических преобразователей, можно построить на основе алгоритмов, моделирующих преобразование сигналов в отдельных преобразователях (рис. 1.14).

Моделирование описанных систем часто сопряжено со значительными трудностями, однако в ряде случаев структурную схему системы можно представить в упрощенном виде (рис. 1.15).

В этом случае

$$\begin{aligned} \varepsilon(n) &= x(n) - y(n); \\ \varepsilon_1(n) &= W_N[x(n) - y(n)], \end{aligned} \quad (1.8)$$

где W_N — оператор нелинейного преобразователя системы.

Применяя далее описанный выше алгоритм для цифрового моделирования линейных динамических систем, получаем:

$$y(n) = \sum_{k=0}^{n-1} g(k) \varepsilon_1(n-k).$$

Таким образом, особенностью цифровых моделей нелинейных замкнутых систем является необходимость решения на каждом шаге нелинейных алгебраических уравнений вида (1.8) при условии, что линейные динамические звенья системы моделируются на основе дискретной свертки. Решение этой задачи можно упростить, если в цепь обратной связи системы ввести элемент задержки

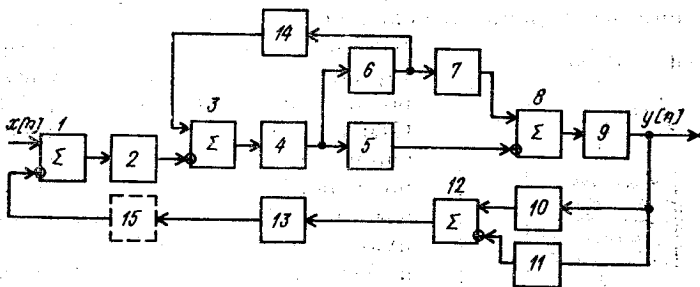


Рис. 1.14.

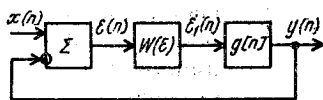


Рис. 1.15.

на один период. Тогда нелинейное уравнение (1.8) преобразуется в рекуррентное:

$$y(n) = \sum_{k=0}^{n-1} g(k) W_N \{x(n-k-1) - y(n-k-1)\}.$$

Введение в цепь обратной связи элемента задержки вносит дополнительную погрешность в цифровую модель. Однако при шаге дискретизации $\tau_{\Delta} \rightarrow 0$ эквивалентная дискретная система с элементом задержки так же, как и без него, совпадает с исходной непрерывной системой. Поэтому выбором шага дискретизации можно добиться сколь угодно малого влияния погрешности задержки.

Для получения имитационной модели системы, представленной на рис. 1.14, составим направленный граф системы с учетом элементов задержки в обратных связях, вершинами которого будут ее элементы, а ребрами — связи между соответствующими элементами (рис. 1.16).

На каждом цикле пересчета будем получать на выходе девятого преобразователя число $y(n)$ — дискретное значение

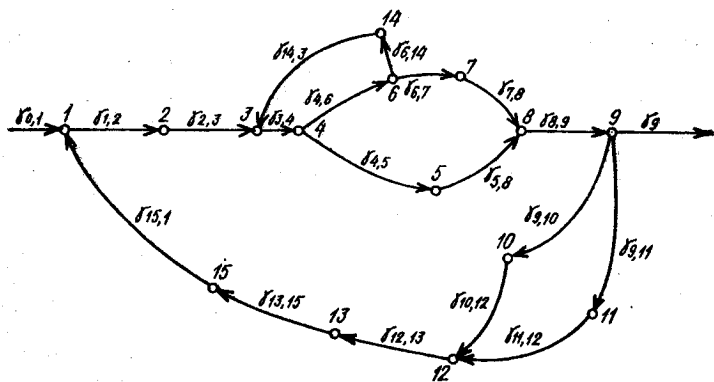


Рис. 1.16.

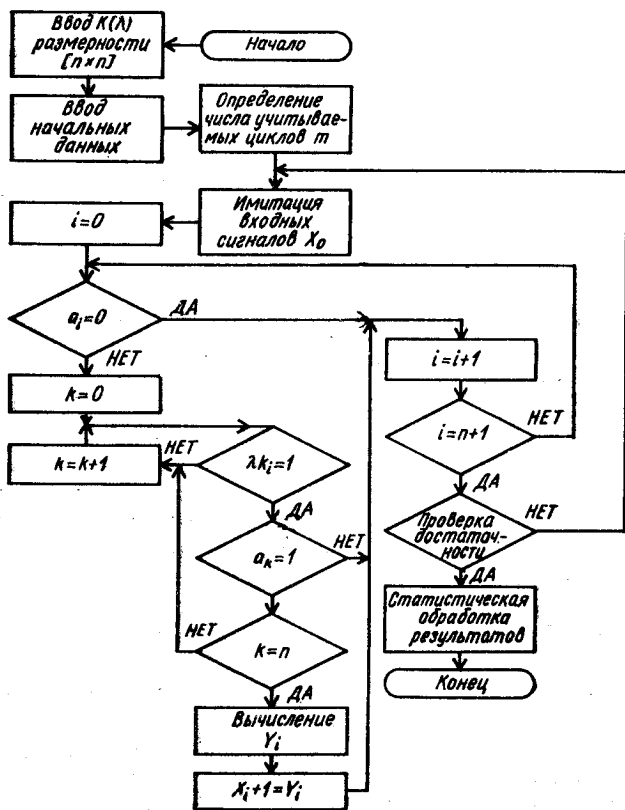


Рис. 1.17.

выходного сигнала системы в соответствующий момент времени. При этом число x (n) пошагово преобразуется на базе моделирования алгоритмов для отдельных звеньев в соответствии с графом системы. В одном цикле каждый элемент системы (каждая вершина графа) учитывается только один раз.

Алгоритм моделирования нелинейных динамических систем представлен на рис. 1.17. В соответствии с алгоритмом для моделирования необходимо ввести в ЭВМ матрицу смежности K [Δ], соответствующую графу системы, а также характеристики операторов системы (статические характеристики, импульсные и переходные функции).

Одной из наиболее сложных задач, решаемых с помощью алгоритма, является выбор последовательности моделирования процессов в преобразователях системы. Эта задача решается последовательным перебором преобразователей и анализа достаточности информации для их моделирования. Достаточность информации определяется с помощью индикаторной переменной a_i , которой присваивается значение 0, если значение выходного сигнала i -го преобразователя в данном цикле моделирования еще не определялось, и 1, если определялось.

Число предшествовавших циклов m , учитываемых при моделировании, определяется как отношение длительности переходного процесса на выходе наиболее инерционного звена к интервалу поступления случайных чисел на вход системы.

Заключительным этапом статистического моделирования является *статистическая обработка результатов*.

В прил. 1 приведены аналитические выражения для описания наиболее часто встречающихся видов законов распределения вероятностей.

Количество реализаций и точность вычислений. Количество реализаций при решении задач методом статистического моделирования определяется требуемой точностью получаемых результатов.

Пусть целью моделирования будет вычисление вероятности P появления некоторого случайного события A (например, при исследовании точности контроля размеров изделий вероятность принадлежности размеров изделия к определенной сортировочной группе; вероятность того, что изделие будет застраховано при вторичном контроле, если при первичном контроле признано годным). В качестве оценки для искомой вероятности P применяется частота L/N наступления события A при N реализациях, где L — количество испытаний, при которых происходит событие A . В силу центральной предельной теоремы теории вероятностей частота L/N при достаточно больших N имеет нормальное распределение, определяемое математическим ожиданием $M(L/N) = P$ и дисперсией $D(L/N) = P(1 - P)/N$.

Следовательно,

$$P(L/N < x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi D(L/N)}} \exp\left\{-\frac{[(L/N) - M(L/N)]^2}{2D(L/N)}\right\} d(L/N).$$

При достаточно больших N получаем

$$P\left[\frac{(L/N) - M(L/N)}{\sqrt{D(L/N)}} < \frac{\varepsilon}{\sqrt{D(L/N)}}\right] \approx \Phi\left[\frac{N\varepsilon}{\sqrt{D(L/N)}}\right], \quad (1.9)$$

где $\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2} dz$; ε — точность неравенства.

Задаваясь определенной вероятностью P , найдем по таблицам нормального распределения значение $D(L/N)$, удовлетворяющее уравнению $\Phi(t) = 0$, где $t = \varepsilon/D(L/N)$.

Доверительную оценку L/N получаем в виде

$$P[|(L/N) - P| < \varepsilon] = t_p \sqrt{P(1-P)/N}. \quad (1.10)$$

По формуле (1.10) с вероятностью, большей чем 0,997, величина L/N удовлетворяет условию

$$|(L/N) - P| < 3\sqrt{P(1-P)/N}.$$

Таким образом, погрешность метода статистического моделирования при вычислении вероятности события A никогда не превышает величины $\varepsilon = 3\sqrt{P(1-P)/N}$ и убывает с увеличением числа испытаний обратно пропорционально корню квадратному из N . Отсюда можно определить количество реализаций N , необходимых для получения оценки L/N с точностью ε и достоверностью P

$$N = t_p^2 P(1-P)/\varepsilon^2$$

или для $P = 0,997$

$$N = 9P(1-P)/\varepsilon^2.$$

Аналогично можно оценить количество реализаций, необходимых для оценки по результатам моделирования среднего значения случайной величины. Предположим, осуществляется формирование N реализаций случайной

величины X , имеющей среднее значение M и дисперсию σ^2 . Определим

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i.$$

В силу центральной предельной теоремы теории вероятностей

$$P\left(\frac{\bar{X}-M}{\sigma/\sqrt{N}} \varepsilon < \sqrt{N}/\sigma\right) \approx \Phi(\varepsilon\sqrt{N}/\sigma).$$

Тогда

$$P[|X - M| < \varepsilon] = t_p \sigma / \sqrt{N},$$

а точность

$$\varepsilon = t_p \sigma / \sqrt{N}. \quad (1.11)$$

Решая (1.11) относительно N , получаем

$$N = t_p^2 \sigma^2 / \varepsilon^2. \quad (1.12)$$

При $P = 0,997$ формулы (1.11) и (1.12) приобретают соответственно вид

$$\varepsilon = 3\sigma / \sqrt{N}, \quad N = 9\sigma^2 / \varepsilon^2.$$

Погрешность метода статистического моделирования как при вычислении вероятности события A , так и при оценке среднего значения случайной величины составляет $\varepsilon = 1/\sqrt{N}$. Уменьшение ошибки ε приближенного решения задачи методом вероятностного моделирования связано со значительным увеличением числа испытаний N , а значит, и с увеличением времени вычислений. Например, увеличение точности на порядок приводит к удлинению времени решения задачи в сто раз.

1.4. Погрешности вычислений, алгоритмов, математических моделей

Замена оригинала моделью всегда связана с некоторым приближением, упрощением, отбрасыванием несущественных (по принятым критериям) свойств, параметров, факторов. Это определяет наличие в математическом описании неустранимых погрешностей, определяемых методологией моделирования и называемых

погрешностями метода моделирования. Например, в рассмотренном ранее статистическом моделировании эти погрешности связаны с дискретизацией непрерывных случайных величин, ограниченным объемом выборки, псевдослучайным характером генерируемых числовых последовательностей. Неустраняемая погрешность всегда присутствует в исходных данных, полученных экспериментальным путем. При этом особого внимания требуют задачи и алгоритмы, чувствительные к изменению исходных данных, поскольку может возникнуть значительный рост погрешностей [22]. Проведение расчетов математических моделей на ЭВМ требует реализации алгоритма в виде последовательности арифметических и логических операций, что определяет наличие погрешностей численного (приближенного) метода решения задачи на ЭВМ. Их можно разделить на несколько видов:

погрешность округления вследствие ограниченного количества разрядов представления чисел в ЭВМ;

погрешность ограничения (усечения), связанная с численным методом решения, когда для приближенного описания функции вместо бесконечных рядов используется несколько их первых членов (например, разностное описание производных);

погрешность распространения, являющаяся результатом накопления погрешностей, появившихся на предыдущих этапах расчета.

Указанные погрешности приводят к возникновению двух типов ошибок: локальной — суммы погрешностей, вносимых в вычислительный процесс на каждом шаге вычислений; глобальной — суммы погрешностей, накопившихся с момента начала вычислений.

По способу представления различают: абсолютную погрешность Δ , определяемую как модуль разности между истинным A и вычисленным a значениями величины,

$$\Delta = |A - a|,$$

и относительную погрешность

$$\sigma_a = \frac{\Delta}{|a|}.$$

Точность вычислений определяется количеством цифр результата, заслуживающих доверия. Цифра приближенного числа называется верной, если абсолютная погрешность не превосходит половины единицы разряда,

в котором находится эта цифра. Очевидно, что все цифры, предшествующие верной, — верные.

Перечислим основные правила трансформации погрешностей в процессе вычислений:

1) абсолютная погрешность суммы конечного числа приближенных чисел не превосходит суммы абсолютных погрешностей этих чисел;

2) относительная погрешность суммы конечного числа приближенных чисел не превосходит наибольшей из относительных погрешностей слагаемых;

3) относительная погрешность произведения при достаточно малых погрешностях ($\delta_a \leq 0,1$) не превосходит суммы относительных погрешностей сомножителей.

Применительно к алгоритмам вычислительных методов важную роль играют понятия сходимости и устойчивости, связанные с оценкой погрешности. *Сходимость* — это свойство алгоритма путем изменения его параметров производить вычисления со сколь угодно малой погрешностью для заданного класса входных данных.

Устойчивость алгоритма — способность производить вычисления и получать конечный результат с заданной точностью при изменении параметров алгоритма и входных данных в некоторой области, называемой областью устойчивости.

В ряде случаев (например, при моделировании измерительных систем и устройств в условиях воздействия случайных влияющих факторов) пользуются вероятностным подходом к оценке погрешностей [4].

Погрешность — одна из основных характеристик качества вычислительного процесса, и ее оценкой должно сопровождаться решение любых инженерных и научных задач численными методами на ЭВМ.

1.5. Математическое моделирование и вычислительные методы в задачах автоматики и информационно-измерительной техники

Математическое моделирование является частью процесса создания технических средств автоматики и информационно-измерительной техники. Вычислительные методы, программно-реализуемые на ЭВМ, стали основным практическим инструментом разработчиков автоматических измерительных информационных устройств и систем. Накоплен большой опыт использования вычисли-

тельной техники, применения численных методов, создания специального программного обеспечения при решении разнообразных задач в этой области. Рассмотрим основные их типы.

1. Идентификация динамических характеристик линейных звеньев при использовании различных описаний сигналов на их входах и выходах.

2. Использование метода наименьших квадратов для идентификации передаточной характеристики по массивам данных, описывающих переходную и частотную характеристики или сигналы на входах и выходах звена.

3. Исследование устойчивости линейных динамических систем на основе использования различных критериев. Построение области устойчивости на плоскости параметров системы.

4. Анализ качества линейных систем автоматического управления. Определение оптимальных управляющих воздействий — решение алгебраического уравнения Риккати (непрерывный и дискретный случаи), к которому сводится задача об оптимальном линейном регуляторе. Решение уравнения Риккати связано с выполнением ряда преобразований и решения частных задач (составления исходной матрицы, преобразования подобия), позволяющих привести матрицы к виду Гессенберга и виду Шура; нахождение собственных значений матриц [8].

5. Исследование нелинейных автоматических систем на основе приближенных методов решения нелинейных дифференциальных уравнений. Применение методов гармонической линеаризации и кусочно-линейной аппроксимации.

6. Статистическое имитационное моделирование измерительных устройств и систем на ЭВМ, включающее: генерацию псевдослучайных числовых последовательностей, имитирующих измеряемые величины и влияющие факторы; моделирование преобразования информативных параметров сигналов в исследуемых устройствах; обработку выходных псевдослучайных числовых последовательностей (построение гистограмм, корреляционных функций, оценка по критериям) [4, 9].

7. Аналитическое вероятностное моделирование измерительных информационных устройств и систем на основе взаимосвязанных вероятностных (законы распределе-

ния вероятностей) и энергетических (спектральные плотности мощности) моделей [9, 14].

8. Решение задачи определения распределения значений измеряемой величины (скоростей потока, звуковой волны, температур) в замкнутой области [13, 18].

9. Исследование устройств автоматики методом планирования эксперимента.

10. Анализ спектров различных сигналов с использованием преобразования Фурье в задачах распознавания образов и цифровой обработки сигналов.

11. Анализ и оценка погрешностей измерительных устройств и систем на основе методов частных производных, вероятностного моделирования, интервального анализа и др.

12. Решение задач автоматизации проектирования систем автоматического управления. В этом направлении накоплен большой опыт в создании разнообразных пакетов прикладных программ [21].

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Что такое модель, математическая модель?

2. В чем заключаются преимущества математических моделей?

3. Дайте определение алгоритма.

4. Назовите типовые блоки в структурной схеме алгоритма, нарисуйте их условные обозначения.

5. Какие существуют структуры алгоритмов? Приведите примеры.

6. Какие виды циклов вам известны?

7. Опишите правила записи алгоритмов в виде графических и операторных схем.

8. На каких принципах основано структурное программирование?

9. Какие виды и примеры математических моделей вы знаете?

10. Опишите основные этапы математического моделирования.

11. Дайте сравнительную оценку аналитических и имитационных моделей.

12. Как классифицируются математические модели?

13. На каких принципах базируется построение детерминированных моделей?

14. На чем основано использование метода статистического моделирования на ЭВМ? Из каких этапов состоит процесс построения статистических моделей?

15. Дайте анализ методов генерации случайных чисел с заданными свойствами.

16. Как статистически моделируется на ЭВМ работа различных звеньев и преобразователей систем и устройств?

17. Опишите алгоритмы статистической обработки данных (построение гистограммы, корреляционной матрицы, вычисление моментов).

18. Как связана точность моделирования и объем выборки?

19. Обоснуйте правила трансформации погрешностей, приведенные в п. 1.4.

20. Составьте программы генерации на ЭВМ случайных чисел, распределенных по нормальному закону распределения, законам распределения Релея и Максвелла.

21. Приведите примеры использования математического моделирования и вычислительных методов в задачах специальности.

Глава 2. ЧИСЛЕННЫЕ МЕТОДЫ ЛИНЕЙНОЙ АЛГЕБРЫ

Рассмотрим два раздела линейной алгебры: методы решения систем линейных уравнений и нахождение собственных значений матриц. Практические приложения этих методов широко применяются при решении инженерных задач. Так, к системам линейных уравнений сводятся многие задачи идентификации, оптимального управления, они являются основой решения других задач вычислительной математики, например, систем дифференциальных уравнений. В теории автоматического управления нахождение собственных значений позволяет определить критические параметры управляющих воздействий, превышение которых ведет к потере устойчивости системы. При динамическом анализе резонансных колебательных систем собственные значения соответствуют собственным частотам колебаний, а собственные векторы характеризуют моды этих колебаний.

Математический аппарат линейной алгебры базируется на теории матриц, поэтому целесообразно в первом разделе этой главы привести основные определения матричного анализа [5].

2.1. Основные сведения из теории матриц

Матрицу A обычно определяют как прямоугольную таблицу чисел

$$\begin{array}{ccccccc} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ a_{m1} & a_{m2} & & & & a_{mn} \end{array}$$

где элементы матрицы a_{ij} ($1 \leq i \leq m$, $1 \leq j \leq n$) — числа из множества вещественных или комплексных чисел.

Размер матрицы в терминах числа строк m и числа столбцов n обозначается через $[m, n]$. Если $n = m$, то считают, что матрица квадратная порядка n , а в общем случае ее называют прямоугольной.

Квадратная матрица A называется *симметричной*, если

$$a_{ij} = a_{ji}.$$

Элементы единичной матрицы E определяются как

$$\sigma_{ij} \begin{cases} \delta_{ij} = 0, & \text{если } i \neq j; \\ \delta_{ij} = 1, & \text{если } i = j, \end{cases} \quad (2.1)$$

причем δ_{ij} , определяемый из (2.1), называют символом Кронекера.

Одним из видов матриц является трехдиагональная, все элементы которой, кроме находящихся на главной и двух примыкающих к ней диагоналях, равны нулю. К такому виду приводятся произвольные матрицы в результате применения некоторых методов преобразований подобия. Аналогично определяются пятидиагональные, семидиагональные и т. д. матрицы, называемые ленточными.

К матрицам применимы операции транспонирования и обращения. Транспонированием матрицы называется замена ее строк столбцами. Транспонированная матрица обозначается A^T . Обратная матрица A^{-1} определяется как матрица, умножение которой как слева, так и справа на матрицу A дает единичную матрицу:

$$A^{-1} \cdot A = A \cdot A^{-1} = E.$$

Операция нахождения обратной матрицы называется обращением. Определитель (детерминант) матрицы находится из соотношения

$$\det A = |A| = \sum_{(\alpha_1, \alpha_2, \dots, \alpha_n)} (-1)^x a_{1\alpha_1} a_{2\alpha_2} \dots a_{n\alpha_n}.$$

Операция суммирования распространена на все возможные перестановки $(\alpha_1, \alpha_2, \dots, \alpha_n)$ элементов $1, 2, \dots, n$ и содержит $n!$ слагаемых, причем $x = 0$, если перестановка четная, и $x = 1$, если она нечетная. Квадратная матрица A называется неособенной (несингулярной), если $\det A \neq 0$. В случае, когда $\det A = 0$, матрица A — сингулярная.

Матрица A называется ортогональной, если

$$A^T \cdot A = E,$$

т. е.

$$A^{-1} = A^T.$$

Одним из наиболее важных и употребительных в линейной алгебре является определение подобных матриц: матрицы A и B подобны, если существует такая неособенная матрица P , что справедливо соотношение

$$B = P^{-1} \cdot AP.$$

Над матрицами выполняют следующие действия [5, 7, 8]:

суммирование и вычитание — действия производятся над элементами с одинаковыми индексами матриц одного типа;

умножение матрицы на число — умножаются все элементы матрицы, причем:

$$\det \alpha A = \alpha^n \det A,$$

где α — вещественное число; A — квадратная матрица порядка n ;

умножение матриц — выполняется при условии, что число столбцов матрицы A равно числу строк матрицы B :

$$C = AB,$$

где A — матрица размера $[q, n]$; B — матрица размера $[n, m]$; C — матрица размера $[q, m]$.

Элементы матрицы C определяются как

$$C_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj},$$

где $i = 1, 2, \dots, q$; $j = 1, 2, \dots, m$.

Из правила умножения следуют свойства матрицы:

$$AB \neq BA,$$

если $AB = BA$, то A и B — коммутативные (перестановочные) матрицы;

$$AE = EA,$$

т. е. в случае, когда A — квадратная матрица, E играет роль единицы при умножении;

$$\det (AB) = \det A \det B,$$

если A и B — квадратные;

$$A(B + C) = AB + AC \text{ и } A(BC) = (AB)C,$$

если размеры матриц позволяют производить эти действия.

2.2. Решение систем линейных уравнений

В общем случае задача формулируется следующим образом: найти значения x_1, x_2, \dots, x_n , удовлетворяющие системе (2.10) n линейных уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= c_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= c_2; \\ \dots & \dots \dots \dots \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= c_n, \end{aligned}$$

или в матричной форме

$$AX = C. \tag{2.2}$$

Необходимым и достаточным условием существования решения является неравенство нулю определителя A (нелинейная независимость уравнений):

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \neq 0.$$

Методы решения систем линейных уравнений можно разделить на прямые и итерационные. К прямым, позволяющим получить точное решение, относятся методы определителей Крамера, Гаусса, прогонки [3, 7]. Итерационные методы, основанные на получении и уточнении последовательных приближений к точному решению, эффективны в случае, когда много нулевых коэффициентов или высок порядок системы (метод Гаусса эффективен до порядка 10^4 , итерационные — до 10^6). Отметим, что метод Крамера менее эффективен для использования при решении задач на ЭВМ, поскольку требует большого числа арифметических операций.

Метод Гаусса. Метод Гаусса (метод исключения) основан на приведении матрицы коэффициентов системы

к треугольному виду

$$\begin{vmatrix} \times & \times & \times & \dots & \times & \times \\ 0 & \times & \times & \dots & \times & \times \\ 0 & 0 & \times & \dots & \times & \times \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & \times & \times \\ 0 & 0 & 0 & \dots & 0 & \times \end{vmatrix}$$

и состоит из двух этапов: прямого хода и обратной подстановки. Этап прямого хода заканчивается, когда одно из уравнений системы становится уравнением с одним неизвестным. Далее, осуществляя обратную подстановку, находят все неизвестные. Этот метод легко алгоритмируется на ЭВМ [13].

Сначала делением коэффициентов на a_{11} нормируется первое уравнение, затем полученное уравнение умножают на коэффициенты a_{i1} и последовательно вычитают из всех уравнений. Таким образом x_1 исключается из всех уравнений, кроме первого. На следующем этапе аналогичная процедура применяется к остальным $(n - 1)$ -му уравнениям и повторяется, пока система не будет приведена к треугольному виду.

На k -м шаге нормируются коэффициенты k -го уравнения:

$$b_{kj} = \frac{a_{kj}}{a_{kk}},$$

а новые коэффициенты в следующих уравнениях определяются как

$$b_{ij} = a_{ij} - a_{ik}b_{kj}, \quad i > k.$$

Коэффициенты a_{ij} изменяются на каждом шаге.

Рассмотрим применение метода Гаусса на примере решения следующей системы уравнений:

$$\begin{aligned} x_1 + x_2 + x_3 - x_4 &= 2; \\ x_1 - x_2 - x_3 + x_4 &= 0; \\ 2x_1 + x_2 - x_3 + 2x_4 &= 9; \\ 3x_1 + x_2 + 2x_3 - x_4 &= 7. \end{aligned}$$

Для удобства обозначим уравнения буквами и выпишем только коэффициенты при неизвестных и свободные

члены уравнений:

$$\begin{array}{l}
 A_1 \quad 1 \quad 1 \quad 1-12 \quad B_1 = A_1/1 \quad 1 \quad 1 \quad 1-1 \quad 2 \\
 A_2 \quad 1-1-1 \quad 10 \text{ исключая } B_2 = A_2 - B_1 \quad 0-2-2 \quad 2-2 \quad x_2 \\
 A_3 \quad 2 \quad 1-1 \quad 29 \text{ члены с } x_1 \Rightarrow B_3 = A_3 - 2B_1 \quad 0-1-3 \quad 4 \quad 5 \Rightarrow \\
 A_4 \quad 3 \quad 1 \quad 2-17, \quad B_4 = A_4 - 3B_1 \quad 0-2-1 \quad 2 \quad 1 \\
 \\
 B_1 \quad \quad \quad 11 \quad 1-12 \quad B_1 \quad \quad \quad 111 \quad -1 \quad 2 \\
 C_2 = B_2/(-2) \quad 0 \quad 1 \quad 1-11 \quad x_3 \quad C_2 \quad \quad \quad 011 \quad -1 \quad 1 \quad x_4 \\
 C_3 = B_3 + C_2 \quad 00-2 \quad 36 \Rightarrow D_3 = \frac{C_3}{(-2)} \quad 001-3/2-3 \Rightarrow \\
 C_4 = B_4 + 2C_2 \quad 00 \quad 1 \quad 03 \quad D_4 = C_4 - D_3 \quad 000 \quad 3/2 \quad 6 \\
 \\
 B_1 \quad \quad \quad 111 \quad -1 \quad 2 \quad x_1 + x_2 + x_3 - x_4 = 2 \\
 C_3 \quad \quad \quad 011 \quad -1 \quad 1 \quad x_2 + x_3 - x_4 = 1 \\
 D_3 \quad \quad \quad 010 \quad -3/2 \quad 1 \Rightarrow x_3 - \frac{3}{2}x_4 = -3 \\
 E_4 = 2 D_4/3 \quad 000 \quad 1 \quad 4 \quad x_4 = 4
 \end{array}$$

Далее обратной подстановкой находим $x_4 = 4$, $x_3 = 3$, $x_2 = 2$, $x_1 = 1$. Желательно сделать нулевыми все элементы матрицы коэффициентов, стоящие вне главной диагонали, т. е. привести ее к диагональному виду.

На рис. 2.1 представлены программа и пример решения системы уравнений методом Гаусса.

Число арифметических операций для реализации метода

$$N \approx \frac{2}{3} n^3.$$

Метод исключения Гаусса—Жордана. Этот метод позволяет привести матрицу коэффициентов к диагональному виду. Единственным его формальным отличием от предыдущего метода является то, что при $i > k$ подставляются $i \neq k$ (назовем k -й ряд ведущим). В методе Гаусса преобразования затрагивали только уравнения, стоящие ниже ведущего ряда. В методе Гаусса — Жордана преобразовываются уравнения, стоящие и под ведущим рядом, и над ним.

Решим методом Гаусса — Жордана предыдущую задачу.

$$\begin{array}{l}
 A_1 \quad 1 \quad 1 \quad 1-12 \quad B_1 = A_1/1 \quad 1 \quad 1 \quad 1-1 \quad 2 \\
 A_2 \quad 1-1-1 \quad 10x_1 \quad B_2 = A_2 - B_1 \quad 0-2-2 \quad 2-2 \quad x_2 \\
 A_3 \quad 2 \quad 1-1 \quad 29 \Rightarrow B_3 = A_3 - 2B_1 \quad 0-1-3 \quad 4 \quad 5 \Rightarrow \\
 A_4 \quad 3 \quad 1 \quad 2-17 \quad B_4 = A_4 - 3B_1 \quad 0-2-1 \quad 2 \quad 1
 \end{array}$$

```

PROGRAM PGAU
REAL A(5,5),B(5),C(5)
READ(1,*) ((A(I,J),J=1,5),I=1,5)
READ(1,*)B
DO 20 I=1,5
    PRINT1000,((A(I,J),J,
                & J=1,5),B(I)
20    & CONTINUE
    CALL GAUSS(A,B,C,5)
    STOP
1000    & FORMAT(5X,5(F5.2,X,'C',I1,
                & ') + '), ' = ',F5.2)
    END
C
C
SUBROUTINE GAUSS(A,B,C,N)
REAL A(N,N)
REAL B(N)
REAL C(N)
NM1=N-1
DO 300 K=1,NM1
    KP1=K+1
    L=K
    DO 400 I=KP1,N
        IF(ABS(A(I,K))-
        & ABS(A(L,K)))400,400,401
401    & L=I
400    CONTINUE
405    IF(L-K)500,500,405
    DO 410 J=K,N
        TEMP=A(K,J)
        A(K,J)=A(L,J)
410    & A(L,J)=TEMP
        TEMP=B(K)
        B(K)=B(L)
        B(L)=TEMP
500    DO 300 I=KP1,N
        & FACTOR=A(I,K)/
        & A(K,K)
        & A(I,K)=0.
        DO 301 J=KP1,N
        & A(I,J)=A(I,J)-
        & FACTOR*A(K,J)
301    & B(I)=B(I)-FACTOR*B(K)
        C(N)=B(N)/A(N,N)
        I=NM1
710    IP1=I+1
        SUM=0.
        DO 700 J=IP1,N
700    & SUM=SUM+A(I,J)*C(J)
        C(I)=(B(I)-SUM)/A(I,I)
        I=I-1
        IF(I)800,800,710
800    & DO 900 I=1,N

```

```

900 PRINT 901, I, C(I)
901 FORMAT(' C(', I2, ') = ', F5.2)
RETURN
END

```

$$1 C_1 + 2 C_2 + 3 C_3 + 1 C_4 - 1 C_5 = 13$$

$$1 C_1 + 0 C_2 - 3 C_3 + 0 C_4 + 1 C_5 = -3$$

$$0 C_1 + 1 C_2 + 1 C_3 - 2 C_4 + 1 C_5 = 2$$

$$2 C_1 + 0 C_2 + 1 C_3 + 1 C_4 - 2 C_5 = -1$$

$$5 C_1 + 0 C_2 + 0 C_3 + 0 C_4 - 1 C_5 = 0$$

$$C(1) = 1.00$$

$$C(2) = 2.00$$

$$C(3) = 3.00$$

$$C(4) = 4.00$$

$$C(5) = 5.00$$

Рис. 2.1.

На этом этапе происходит изменение алгоритма решения по сравнению с методом Гаусса.

$$\begin{aligned}
C_1 &= B_1 - C_2 & 10 & 0 & 01 & D_1 &= C_1 - 10D_3 & 100 & 0 & 1 \\
C_2 &= B_2 / (-2) & 01 & 1 & -11x_3 & D_2 &= C_2 - D_3 & 010 & 1/2 & 9x_4 \\
C_3 &= B_3 + C_2 & 00 & -2 & 36 & \Rightarrow D_3 &= C_3 / (-2) & 001 & -3/2 & -3 \Rightarrow \\
C_4 &= B_4 + 2C_2 & 00 & 1 & 03 & D_4 &= C_4 - D_3 & 000 & 3/2 & 6
\end{aligned}$$

$$E_1 = D_1 + (0)E_4 \quad 10001$$

$$E_2 = D_2 - \left(\frac{1}{2}\right)E_4 \quad 01002$$

$$E_3 = D_3 + \left(\frac{3}{2}\right)E_4 \quad 00103$$

$$E_4 = D_4 / \left(\frac{3}{2}\right) \quad 00014$$

Этот метод облегчает получение решения, но сопровождается увеличением объема вычислений. Применение обоих методов затруднительно, если какой-либо из элементов ведущего ряда равен нулю. Трудности этой можно

избежать, если изменить порядок, в котором расположены уравнения системы. Максимальная точность достигается в случае, когда ведущий элемент имеет наибольшее значение. Строку с нулевым или малым ведущим элементом надо заменить на ту из стоящих под ней строк, в которой в том же столбце стоит элемент, имеющий наибольшее значение.

Модифицированный метод Гаусса. Во многих случаях возникает необходимость решения систем линейных уравнений с изменяющейся матрицей коэффициентов и постоянным столбцом свободных членов. Наиболее часто для решения таких задач используется модифицированный метод Гаусса. В этом методе матрицу коэффициентов A из матричного уравнения (2.2) представляют в виде произведения *левой и правой треугольных матриц:

$$LR = A.$$

Поскольку диагональные элементы одной из матриц равны единице, их можно не запоминать и обе матрицы хранить в памяти ЭВМ на месте матрицы коэффициентов.

В варианте метода, называемом методом Краута, используется следующая последовательность нахождения элементов матриц L и R , $k = 1, 2, \dots, n$:

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} r_{pk}, \quad i = k, k+1, \dots, n;$$

$$l_{kk} = 1/l_{kk} \quad (2.3)$$

и

$$r_{kj} = l_{kk} (a_{kj} - \sum_{p=1}^{k-1} l_{kp} r_{pj}), \quad j = k+1, \dots, n;$$

$$r_{kk} = 1.$$

Система $AX = C$ сводится к системе $LRX = C$, решение которой заменяется решением двух систем с треугольными матрицами

$$\left. \begin{aligned} LY &= C; \\ RX &= Y. \end{aligned} \right\} \quad (2.4)$$

Элементы Y и X находят из следующих соотношений:

$$y_i = l_{ii} (C_i - \sum_{p=1}^{i-1} l_{ip} y_p), \quad i = 1, \dots, n;$$

$$x_i = y_i - \sum_{p=i+1}^n r_{ip} x_p; \quad i = n, n-1, \dots, 1. \quad (2.5)$$

Таким образом с изменением S отпадает необходимость полного повторения процесса исключения, а алгоритм, составленный по формулам (2.3), (2.4), (2.5), подвергается минимальной модификации. Число арифметических операций, необходимых для решения этим методом системы линейных алгебраических уравнений, $N = 2n^2$.

Метод прогонки. Метод прогонки применяется для решения систем уравнений с ленточными матрицами коэффициентов. Рассмотрим его применение для решения трехдиагональной системы, к которой часто сводится линейная краевая задача [12].

Запишем систему в виде

$$\left. \begin{aligned} b_0 y_0 + c_0 y_1 &= \varphi_0; \\ a_1 y_0 + b_1 y_1 + c_1 y_2 &= \varphi_1; \\ a_2 y_1 + b_2 y_2 + c_2 y_3 &= \varphi_2; \\ \dots &\dots \\ a_i y_{i-1} + b_i y_i + c_i y_{i+1} &= \varphi_i; \\ a_{n-1} y_{n-2} + b_{n-1} y_{n-1} + c_{n-1} y_n &= \varphi_{n-1}; \\ a_n y_{n-1} + b_n y_n &= \varphi_n. \end{aligned} \right\} \quad (2.6)$$

Для решения полученной системы выполним аналог прямого хода метода Гаусса. Тогда получим систему вида

$$\left. \begin{aligned} y_0 - u_0 y_1 &= v_0; \\ y_1 - u_1 y_2 &= v_1; \\ \dots &\dots \\ y_{i-1} - u_{i-1} y_i &= v_{i-1}; \\ y_i - u_i y_{i+1} &= v_i; \\ y_{n-1} - u_{n-1} y_n &= v_{n-1}; \\ y_n &= v_n, \end{aligned} \right\} \quad (2.7)$$

где $u_0, v_0, u_1, v_1, \dots, u_n, v_n$ — некоторые коэффициенты, называемые прогоночными. Отметим, что

$$u_0 = -\frac{c_0}{b_0}; \quad v_0 = \frac{\varphi_0}{b_0}. \quad (2.8)$$

Прогоночные коэффициенты, определенные из последней системы, дают возможность найти y_n, y_{n-1}, \dots, y_0 . Таким образом, вычисление y_n, y_{n-1}, \dots, y_0 свелось к вычислению прогоночных коэффициентов. Для их определения рассмотрим соотношения (2.6) и (2.7). Исключая из этих соотношений y_{i-1} , получим

$$y_i + \frac{c_i}{a_i u_{i-1} + b_i} y_{i+1} = \frac{\varphi_i - a_i v_{i-1}}{a_i u_{i-1} + b_i}. \quad (2.9)$$

Сравнивая (2.9) с равенством (2.7), приходим к рекуррентным формулам:

$$u_i = - \frac{c_i}{a_i u_{i-1} + b_i}, \quad i = 1, 2, \dots, n; \quad (2.10)$$
$$v_i = \frac{\varphi_i - a_i v_{i-1}}{a_i u_{i-1} + b_i}, \quad i = 1, 2, \dots, n.$$

Вместе со значениями v_0 и u_0 (2.8) найденные формулы определяют прогоночные коэффициенты. После этого из (2.7) определяются искомые значения y_n, y_{n-1}, \dots, y_0 :

$$y_n = v_n, \quad y_i = u_i y_{i+1} + v_i, \quad i = n-1, \dots, 1, 0. \quad (2.11)$$

Нахождение прогоночных коэффициентов по (2.10) называется прямым ходом метода прогонки, а нахождение y_n, y_{n-1}, \dots, y_0 по (2.11) — обратным ходом.

Метод прогонки легко алгоритмизируется, используя его программы часто входят в стандартное математическое обеспечение ЭВМ. Применение метода прогонки позволяет значительно сократить число арифметических операций по сравнению с методом Гаусса $N \approx 3n$.

Итерационные методы решения систем линейных уравнений. Итерационные методы особенно эффективны при большом порядке системы. Они применяются к системам, предварительно приведенным к виду

$$\left. \begin{aligned} x_1 &= b_{1,n}x_n + b_{1,n-1}x_{n-1} + \dots + b_{1,1}x_1 + b_{1,0}; \\ x_2 &= b_{2,n}x_n + b_{2,n-1}x_{n-1} + \dots + b_{2,1}x_1 + b_{2,0}; \\ &\dots \\ x_n &= b_{n,n-1}x_{n-1} + \dots + b_{n,n}x_n + b_{n,1}x_1 + b_{n,0}. \end{aligned} \right\} \quad (2.12)$$

Существует несколько разновидностей итерационных методов [18, 23]. Это методы Якоби (простой итерации), Гаусса — Зейделя и последовательной верхней релаксации, в основе которых лежит систематическое уточнение значений переменных, заданных в начале счета.

В методе Якоби исходные значения переменных используются для вычисления новых значений x_1, x_2, \dots, x_n с помощью приведенных выше уравнений. Процесс прекращается, когда все новые значения оказываются достаточно близкими к исходным. В противном случае новые значения используются вместо исходных. Эта процедура повторяется до тех пор, пока не будет достигнута сходимость или станет ясно, что процесс расходится. В этом методе замена значений всех переменных производится одновременно (одновременное смещение).

В методе Гаусса — Зейделя уточненное значение x_1 сразу же используется для вычисления x_2 . Затем по новым значениям x_1 и x_2 определяются x_3 и т. д. Это позволяет существенно увеличить быстроту сходимости.

В методе последовательной верхней релаксации новые значения каждой переменной представляются в виде:

$$x_i^{(n+1)} = x_i^{(n)} + \omega (\bar{x}_i^{(n+1)} - x_i^{(n)}),$$

где $\bar{x}_i^{(n+1)}$ — уточненное значение $x_i^{(n)}$ по методу Гаусса — Зейделя; ω — параметр релаксации ($1 \leq \omega \leq 2$).

При $\omega = 1$ этот метод тождествен методу Гаусса — Зейделя. Скорость сходимости зависит от ω .

Анализ сходимости итерационных методов решения систем линейных уравнений связан с понятием нормы матрицы.

Нормой матрицы B называется такое вещественное число $\|B\|$, которое удовлетворяет следующим свойствам:

- 1) $\|B\| > 0$, если $B \neq 0$ (причем $\|0\| = 0$);
- 2) $\|\lambda B\| = |\lambda| \cdot \|B\|$;
- 3) $\|A + B\| \leq \|A\| + \|B\|$;
- 4) $\|AB\| \leq \|A\| \cdot \|B\|$,

где λ — любое вещественное число; A, B — матрицы с произвольными вещественными элементами, для которых определена операция AB .

Норма называется мультипликативной, если обладает всеми перечисленными свойствами, и аддитивной или обобщенной, если удовлетворяет только свойствам 1, 2, 3.

Наибольшее распространение получили следующие нормы матриц:

$$\|B\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}| \text{ — 1 норма;}$$

$$\|B\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| \text{ — } \infty \text{ норма;}$$

$$\|B\|_E = \sqrt{\sum_{i,j=1}^n b_{ij}^2} \text{ — E норма (евклидова).}$$

Существует правило: если хотя бы одна из перечисленных норм матрицы коэффициентов уравнения (2.12)

меньше единицы, то итерационный алгоритм для решения этого уравнения будет сходящимся при произвольном выборе начального приближения.

2.3. Решение задач нахождения собственных значений матриц

Большое количество инженерных задач сводится к рассмотрению систем уравнений, имеющих единственное решение лишь в том случае, если известно значение некоторого входящего в них параметра. Этот особый параметр называют характеристическим, или собственным, параметром (значением) системы.

Проблема нахождения собственных значений, ее практическое приложение и методы решения широко освещены в литературе [7, 8, 23].

Формулируем проблему: требуется найти n скалярных значений λ и собственных векторов X , удовлетворяющих матричному уравнению

$$AX = \lambda X.$$

Наряду с обычной проблемой собственных значений существует так называемая обобщенная проблема собственных значений (задача для двух матриц A и F):

$$X = \lambda FX.$$

Практический интерес представляют следующие свойства собственных значений, на которых основаны способы их нахождения.

1. Все n собственных значений симметричной матрицы размерности $[n, n]$, состоящей из действительных чисел, действительные.

2. Если собственные значения матрицы различны, то ее собственные векторы ортогональны. Совокупность n линейно независимых собственных векторов образует базис рассматриваемого пространства. Следовательно, для совокупности линейно независимых собственных векторов $X^i, i = 1, 2, \dots, n$, любой произвольный вектор в том же пространстве можно выразить через собственные векторы

$$Y = \sum_{i=1}^n a_i x^i.$$

3. Если две матрицы подобны, то их собственные значения совпадают. Из подобия A и B следует, что $B = P^{-1}AP$, так как $AX = \lambda X$, то $P^{-1}AX = \lambda P^{-1}X$. Если принять $X = PY$, то $P^{-1}APY = \lambda Y$, а $BY = \lambda Y$, т. е. A и B не только имеют одинаковые λ , но и их собственные векторы связаны соотношением $X = PY$.

4. Умножив собственный вектор матрицы на скаляр, получают собственный вектор той же матрицы. Обычно собственные векторы нормализуют, разделив их на наибольший элемент или на сумму квадратов элементов.

Выбор наиболее эффективного метода определения собственных значений или собственных векторов для данной задачи зависит от ряда факторов (тип уравнений, число и характер собственных значений, вид матрицы). Алгоритмы нахождения собственных значений можно разделить на три группы:

прямые, основанные на раскрытии характеристических (вековых) определителей $\det(A - \lambda E) = 0$ и решении характеристических уравнений;

итерационные, основанные на многократном применении итерационного алгоритма, приближающего получаемый в каждом цикле собственный вектор к точному решению;

преобразований подобия, использующих свойства подобных матриц, имеющих одинаковые собственные значения и ортогональные собственные векторы.

Прямые методы. Наиболее очевидным методом решения задачи нахождения собственных значений является их определение из системы уравнений

$$(A - \lambda E)X = 0,$$

которая имеет ненулевое решение лишь в случае, когда

$$\det(A - \lambda E) = 0.$$

Решение в этом случае состоит из двух этапов: 1) развертывания векового определителя (непосредственно или методами: Данилевского, Крылова, Леверье, неопределенных коэффициентов, интерполирования и т. д.), 2) решения полученного характеристического уравнения; корни которого и будут собственными значениями матрицы.

В табл. 2.1 приведены результаты сравнения эффективности различных методов развертывания вековых

Таблица 2.1. Количество действий, используемых различными методами разветвления векового определителя, в зависимости от его порядка

Метод	Порядок									
	3		4		5		7		9	
	У-Д*	С-В**	У-Д	С-В	У-Д	С-В	У-Д	С-В	У-Д	С-В
Непосредственного разветвления	12	10	60	46	320	238	13692	10078	986400	725758
Данилевского	14	12	42	36	92	80	282	252	632	576
Крылова	67	38	179	118	389	280	1287	1022	3209	2688
Леверье	41	27	153	114	414	330	1791	1533	5228	4644
Неопределенных коэффициентов	67	41	171	116	364	265	1189	945	2966	2481
Интерполирования	46	38	125	102	279	230	972	826	2525	2202

* У-Д — умножение — деление

** С-В — сложение — вычитание

определителей, где критерием является количество вычислительных операций.

Наиболее предпочтителен метод Данилевского, сущность которого заключается в приведении векового определителя к так называемому виду Фробениуса

$$\begin{vmatrix} a_1 - \lambda & a_2 & a_3 & \dots & a_n \\ 1 & -\lambda & 0 & \dots & 0 \\ 0 & 1 & -\lambda & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -\lambda \end{vmatrix} \quad (2.13)$$

Вековой определитель матрицы (2.13) легко разлагается в алгебраическое уравнение по элементам первой строки.

Итерационные методы. Итерационные методы основаны на многократном применении итерационного алгоритма, приближающего получаемый в каждом цикле собственный вектор к точному решению [23].

Пример итерационного алгоритма для определения наибольшего собственного значения приведен на рис. 2.2, а программа для его реализации и пример использования — на рис. 2.3.

Процедура начинается с пробного нормированного вектора $X^{(0)}$, который умножается слева на исходную матрицу A , и результат приравнивается произведению постоянной (собственные значения λ) и нормированному вектору $X^{(1)}$

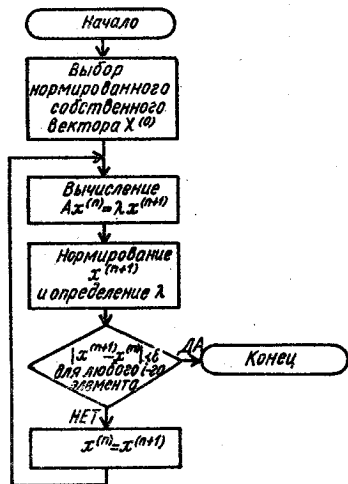


Рис. 2.2.

$$AX^{(0)} = \lambda X^{(1)}$$

Если вектор $X^{(1)}$ совпадает с $X^{(0)}$, то счет прекращается. В противном случае вектор $X^{(1)}$ используется в качестве исходного, и вся процедура повторяется до тех пор, пока

$$|x_i^{(n)} - x_i^{(n-1)}| \leq \epsilon,$$

```

С      ПРОГРАММА ОПРЕДЕЛЕНИЯ НАИБОЛЬШЕГО
С      СОБСТВЕННОГО ЗНАЧЕНИЯ МАТРИЦЫ
С      МЕТОДОМ ИТЕРАЦИЙ. СЧЕТ ПРЕКРАЩАЕТСЯ
С      КОГДА ИЗМЕНЕНИЕ СОБСТВЕННОГО ЗНА-
С      ЧЕНИЯ СТАНОВИТСЯ МЕНЕЕ 0.01 ПРОЦ.
С      ИЛИ ЧИСЛО ИТЕРАЦИЙ ПРЕВЫШАЕТ 50.
С      ДЛЯ УМНОЖЕНИЯ МАТРИЦ ИСПОЛЬЗУЕТСЯ
С      ПОДПРОГРАММА GMPRD ИЗ ППП SSP.
С      ДЛЯ НОРМАЛИЗАЦИИ ВЕКТОРА- ПОДПРО-
С      ГРАМА NORML ( ПРОГРАММА ПРИВЕДЕНА
С      В КНИГЕ ШУП Т. "РЕШЕНИЕ ИНЖЕНЕРНЫХ
С      ЗАДАЧ НА ЭВМ", М., МИР, 1982)
С      PROGRAM MSOB
                REAL S(3,3),X(3),R(3)
                DATA S/10.E06,5.E06,6.E06,5.E06,
& 20.E06,4.E06,6.E06,
& 4.E06,30.E06/,X/1.,0.,0./
                XOLD=0.
                I=0.
                PRINT 100
                PRINT 101
                PRINT 102
                PRINT 100
                PRINT 104,I,X
                DO 1 I=1,50
                        CALL GMPRD(S,X,R,3,3,1)
                        DO 2 J=1,3
                                X(J)=R(J)
                                CALL NORML(XLAM,X)
                                PRINT 103,I,XLAM,X
                                IF (ABS((XOLD-XLAM)/XLAM)
& .LE.0.0001) GO TO 3
                1      XOLD=XLAM
                3      PRINT 100
                        STOP
                100     FORMAT(X,54(' '))
                101     FORMAT(2X,' НОМЕР ',3X,
& ' СОБСТВЕННОЕ',5X,
& ' СОБСТВЕННЫЙ ВЕКТОР')
                102     FORMAT(3X,' ИТЕРАЦИИ',4X,' ЗНАЧЕНИЕ'/
& 15X,' (N/M**2)',5X,' X(1)',6X,' X(2)',
& 6X,' X(3)')
                103     FORMAT(X,15,7X,E12.5,3F10.5)
                104     FORMAT(X,15,19X,3F10.5)
                        END
                SUBROUTINE NORML(XL,X)
                DIMENSION X(3)
                XBIG=X(1)
                IF (X(2).GT.XBIG) XBIG=X(2)
                IF (X(3).GT.XBIG) XBIG=X(3)
                X(1)=X(1)/XBIG
                X(2)=X(2)/XBIG
                X(3)=X(3)/XBIG
                XL=XBIG
                RETURN
                END

```

НОМЕР ИТЕРАЦИИ (N/M**2)	СОБСТВЕННОЕ ЗНАЧЕНИЕ X(1)	СОБСТВЕННЫЙ ВЕКТОР X(2) X(3)	
0	1.0000	0.0000	0.0
10	1.00E+08	1.0000	0.5000 0.6
20	2.60E+08	0.6192	0.6692 1.0
30	3.63E+08	0.4269	0.5627 1.0
40	3.48E+08	0.3758	0.4995 1.0
50	3.42E+08	0.3578	0.4633 1.0
60	3.40E+08	0.3498	0.4428 1.0
70	3.38E+08	0.3459	0.4312 1.0
80	3.38E+08	0.3436	0.4246 1.0
90	3.37E+08	0.3424	0.4209 1.0
100	3.37E+08	0.3417	0.4188 1.0
110	3.37E+08	0.3413	0.4176 1.0
120	3.37E+08	0.3411	0.4169 1.0
130	3.37E+08	0.3409	0.4165 1.0
140	3.37E+08	0.3409	0.4163 1.0

Рис. 2.3.

где $X^{(n)} \in \{x_i^{(n)}\}$ и $X^{(n-1)} \in \{x_i^{(n-1)}\}$ — векторы, полученные в $(n-1)$ -м и n -м итерационных циклах, а ϵ — заданная точность вычислений.

Если процесс сходится, то полученный постоянный множитель соответствует истинному наибольшему собственному значению, а нормированный вектор — соответствующему собственному вектору.

Аналогично можно найти наименьшее собственное значение. Для этого исходная система уравнений предвзвешенно умножается на обратную матрицу A^{-1} :

$$A^{-1} \cdot AX = \lambda A^{-1}X.$$

В результате получаем

$$\frac{1}{\lambda} X = A^{-1}X.$$

Решая эту задачу на собственные значения по алгоритму (см. рис. 2.2), получаем максимальное собственное значение $\lambda_{\max}^* = \frac{1}{\lambda_{\min}}$, по которому находим наименьшее собственное значение λ_{\min} .

Определив максимальное или минимальное собственное значение, можно найти следующее за ним по величине, заменив исходную матрицу матрицей, содержащей лишь оставшиеся собственные значения. Такой метод

представляется последовательностью полиномов Штурма

$$f_m(\lambda) = (a_m - \lambda) f_{m-1}(\lambda) - b_m^2 f_{m-2}(\lambda),$$

при условии $f_0(\lambda) = 1$ и $f_1(\lambda) = a_1 - \lambda$. Принимая $m = 2, \dots, n$ и последовательно определяя корни, найдем собственные значения так же, как и корни полинома n -го порядка.

Если в случае симметричных матриц эффективным способом нахождения собственных значений является приведение матрицы к трехдиагональному виду, то для несимметричных матриц целесообразно пользоваться методами преобразований подобия для приведения исходной матрицы к виду Гессенберга:

$$\begin{vmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & \times & \times \end{vmatrix}.$$

Для приведения матрицы к трехдиагональному виду или виду Гессенберга существует ряд методов, наиболее известными из которых являются методы Якоби, Гивенса, Хаусхолдера.

Метод Якоби позволяет привести матрицу к диагональному виду, исключая последовательно все элементы, стоящие вне главной диагонали. Приведение к чисто диагональному виду требует бесконечно большого числа шагов, так как образование нового нулевого элемента часто ведет к его исчезновению в другом месте. Метод Якоби позволяет подойти близко к диагональной форме, например, образовать трехдиагональную матрицу. В случае симметрии матрицы A действительных чисел преобразование выполняется с помощью ортогональных матриц, полученных в результате вращений, выполненных в действительной плоскости.

Из A получаем матрицу A_1 , если $A = P_1 A P_1^T$. При этом P_1 выбирается так, чтобы в A_1 появился нулевой элемент вне главной диагонали. Эту процедуру продолжают так, стремясь, чтобы на каждом шаге в нуль обращался наибольший внедиагональный элемент.

Преобразующая матрица на каждом шаге конструируется следующим образом. Если элемент a_{kl} имеет

ний, например, симметричная матрица размерности 5×5 приобретает следующие виды:

$$\begin{array}{c}
 \left| \begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array} \right| \Rightarrow \left| \begin{array}{ccccc} * & * & 0 & 0 & 0 \\ * & \text{штрихованная} & * & * & * \\ 0 & \text{штрихованная} & * & * & * \\ 0 & \text{штрихованная} & * & * & * \\ 0 & \text{штрихованная} & * & * & * \end{array} \right| \Rightarrow \\
 \left| \begin{array}{ccccc} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{array} \right| \Rightarrow \left| \begin{array}{ccccc} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{array} \right|
 \end{array}$$

На каждом основном k -м шаге преобразуется только матрица порядка $[n - k + 1]$, занимающая правый нижний угол (заштрихованный) исходной матрицы. На каждой следующей стадии выполняется меньшее число преобразований, чем на предыдущей. Всего для приведения матрицы к трехдиагональному виду требуется выполнить $[n^2 - 3n + 2]/2$ преобразований.

Метод Хаусхолдера позволяет привести симметричную матрицу к трехдиагональному виду, выполнив почти вдвое меньше вычислений, чем в других методах. Однако этот метод связан с выполнением более сложных преобразований. Вместо плоских преобразований (вращений) применяют эрмитовы ортогональные преобразования матриц (отображения). Поэтому методы Якоби и Гивенса называют методами вращений, а Хаусхолдера — отображений. При использовании метода Хаусхолдера выполняются следующие преобразования (на $(n - 2)$ -х основных шагах):

$$A_k = P_k A_{k-1} P_k^T, \quad k = 1, 2, \dots, n - 2,$$

где $A_0 = A$.

Каждая преобразующая матрица имеет вид

$$P_k = E - \frac{u_k u_k^T}{2K_k^2},$$

где $u_{i,k} = 0$ ($i = 1, 2, \dots, k$); $u_{i,k} = a_{k,i}$ ($i = k + 2, \dots, n$);

$$u_{k+1,k} = a_{k,k+1} \mp S_k; \quad S_k = \left[\sum_{i=k+1}^n a_{k,i}^2 \right];$$

$$2u_k^2 = S_k^2 \mp a_{k,k+1} S_k.$$

В этих уравнениях берется знак, соответствующий элементу $a_{k,k+1}$, что позволяет выбрать значение $U_{k+1,k}$ максимальным.

Методами Гивенса и Хаусхолдера можно пользоваться и в случае несимметричных матриц, приводя их к матрице Гессенберга.

Наиболее общими из методов преобразований подобия являются методы LR- и QR-[5, 23], основанные на представлении матрицы в виде произведений

$$A = LR \text{ и } A = QR,$$

где L — левая треугольная матрица с единичными диагональными элементами; R — правая треугольная матрица с единичными диагональными элементами; Q — ортогональная матрица.

Далее, применяя преобразования подобия

$$A^* = L^{-1}AL = L^{-1}(LR)L = RL$$

и

$$A^* = Q^tAL = Q^t(QR)Q = RQ,$$

получаем последовательности матриц, стремящихся к квазидиагональному виду.

Выбор метода нахождения собственных значений. Основными критериями, позволяющими оценить эффективность методов нахождения собственных значений применительно к их алгоритмизации и использованию для решения задач на ЭВМ, являются точность, быстродействие (количество вычислительных операций) и необходимый размер памяти ЭВМ.

Важной характеристикой методов нахождения собственных значений является их общность (универсальность), т. е. применимость к матрицам разных видов, независимо от симметрии и вида элементов (действительные или комплексные). Проанализируем с точки зрения этих критериев методы нахождения собственных значений.

Прямые методы приводят к необходимости решения нелинейных алгебраических уравнений. Однако в задачах на собственные значения часто встречаются кратные корни, при которых итерационные методы решения нелинейных уравнений не гарантируют получения решения. Быстродействие этих методов резко уменьшается по мере повышения порядка матрицы вследствие услож-

Таблица 2.2. Выбор алгоритма решения задачи на собственные значения

Название алгоритма	Применяется для	Результат	Рекомендуется для отыскания собственных значений			Примечание
			наибольшего или наименьшего	σ	λ	
Определитель (итерация)	матриц общего вида то же	Собственные значения Собственные значения и собственные векторы	+	+	+	Требуется нахождения корневой полинома общего вида Обеспечивает наилучшую точность для наибольшего и наименьшего собственных значений
Метод Якоби (преобразование)	симметричных матриц то же	Диагональный вид матрицы	+	+	+	Теоретически требует бесконечного числа шагов
Метод Гивенса (преобразование)	несимметричных матриц	Трёхдиагональный вид матрицы	+	+	+	Требуется знания корневой полинома
Метод Хаусхолдера (преобразование)	симметричных матриц	Матрица вида Гессенберга	+	+	+	Требуется применения дополнительного метода
Метод Хаусхолдера (преобразование)	несимметричных матриц	Трёхдиагональная форма матрицы	+	+	+	Требуется знания корневой полинома
Метод LR (преобразование)	матриц общего вида	Форма Гессенберга	+	+	+	Требуется применения дополнительного метода
Метод QR (преобразование)	матриц общего вида то же	Квазидиагональная форма матрицы То же	+	+	+	Бывает неустойчив

нения этапа раскрытия векового определителя. Это затрудняет использование прямых методов нахождения собственных значений для больших матриц (порядка больше 10) и требует разработки других методов, алгоритмов и программ для ЭВМ.

Существенным недостатком итерационных методов нахождения собственных значений является накопление погрешностей при каждом шаге итерационного процесса (при определении максимального или минимального собственного значения, образовании новой матрицы), поэтому практически ими можно пользоваться для определения не более трех или четырех собственных значений.

В других случаях целесообразно пользоваться методами преобразований подобия.

Наиболее эффективные методы преобразований подобия гарантируют получение результата при достаточно высокой точности решения. Часто наибольшая эффективность решения задачи обеспечивается при одновременном использовании нескольких методов преобразований подобия. Например, методом Хаусхолдера матрица приводится к виду Гессенберга, а затем методом QR определяются собственные значения. При этом используется как быстрота, обеспечиваемая методом Хаусхолдера, так и универсальность алгоритма QR .

Использование табл. 2.2 позволяет облегчить выбор метода решения задачи на собственные значения [23]. Обычно пакеты математического обеспечения ЭВМ содержат подпрограммы, в которых используется большинство из этих алгоритмов [20]. Одним из эффективных способов использования имеющихся ресурсов ЭВМ является одновременное применение двух подпрограмм.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Что такое матрица?
2. Назовите основные типы и свойства матриц.
3. Какие действия производятся над матрицами?
4. Какие матрицы называются подобными?
5. Какие матрицы называются ортогональными?
6. Как осуществляется обращение и транспонирование матриц?
7. Дайте сравнительную оценку методам решений линейных систем уравнений.

8. Решите систему уравнений

$$\begin{aligned}x_1 + x_2 + x_3 &= 6; \\5x_1 - 2x_2 + x_3 &= 4; \\x_1 - 6x_3 &= -17\end{aligned}$$

модифицированным методом Гаусса.

9. Напишите программы решения на ЭВМ системы из пяти линейных уравнений методом Гаусса — Жордана.

10. В чем отличие метода Гаусса — Жордана, модифицированного метода Гаусса от метода Гаусса?

11. Когда эффективно использование метода прогонки? Сравните его с другими методами. Оцените время, необходимое для решения на ЭВМ с быстродействием 1 млн. операций / с системы из 100 линейных уравнений методами Гаусса, Гаусса — Жордана, прогонки.

12. Подсчитайте, сколько арифметических операций необходимо для решения методом Гаусса — Жордана системы из n линейных уравнений. Как строится алгоритм итерационных методов решения систем линейных уравнений?

13. Охарактеризуйте и сравните основные разновидности итерационных методов решения систем линейных уравнений.

14. Сформулируйте проблему нахождения собственных значений.

15. Охарактеризуйте прямые методы нахождения собственных значений.

16. В чем сущность метода Данилевского разложения векового определителя?

17. Разработайте структурную схему алгоритма определения наибольшего (или наименьшего) собственного значения итерационным методом и составьте программу для его реализации на ЭВМ.

18. Как осуществляется нахождение собственных значений итерационным методом исчерпывания?

19. Охарактеризуйте методы преобразований подобия нахождения собственных значений.

20. Как найти собственные значения симметричной трехдиагональной матрицы?

21. Как найти собственные значения с помощью QR- и RL-алгоритмов?

22. Какие стандартные программы из математического обеспечения ЭВМ предназначены для нахождения собственных значений?

23. Дайте сравнительную оценку методам нахождения собственных значений.

24. При решении каких задач автоматизации, управления, информационно-измерительной техники, радиотехники возникает необходимость нахождения собственных значений матриц?

Глава 3. МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И СИСТЕМ

Решение нелинейных уравнений и систем является не только важной самостоятельной задачей, но и частью других задач вычислительной математики, например, решения нелинейных дифференциальных уравнений или нахождения собственных значений матриц. С ними связано построение разнообразных моделей устройств и систем автоматики и информационно-измерительной техники.



Рис. 3.1.

Задачи, сводящиеся к решению алгебраических и трансцендентных уравнений, можно классифицировать по числу уравнений и в зависимости от предполагаемого характера и числа решений (рис. 3.1).

Трансцендентными называются нелинейные уравнения, содержащие тригонометрические или другие специальные функции, например, логарифмическую или экспоненциальную. Уравнения, не содержащие специальных функций, а только степени аргумента с соответствующими коэффициентами, являются нелинейными алгебраическими.

Применение прямых методов решения таких уравнений возможно лишь для алгебраических уравнений, причем практически целесообразно при порядке не более третьего. Поэтому на первый план выходят итерационные методы, особенно при наличии эффективных алгоритмов, реализующих их на ЭВМ.

Применительно к алгебраическим уравнениям можно указать ряд свойств, использование которых позволит ускорить процесс решения задачи.

Общий вид алгебраического уравнения:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0. \quad (3.1)$$

Свойства алгебраических уравнений:

1. Основная теорема алгебры. Алгебраическое уравнение порядка n имеет n корней, которые могут быть как действительными, так и комплексными.

Каждый корень считается столько раз, какова его кратность. Кратность корня x_0 равна k , если

$$f'(x_0) = f''(x_0) = \dots = f^{(k-1)}(x_0) = 0.$$

2. Если все коэффициенты a_i уравнения (3.1) действительные, то все комплексные корни образуют комплексно сопряженные пары.

3. Теорема Декарта. Число положительных действительных корней равно или меньше числа перемен знаков в последовательности коэффициентов a_i (то же утверждение справедливо относительно числа отрицательных действительных корней при замене в (3.1) x на $(-x)$).

4. Теорема Лагранжа. Верхняя граница положительных действительных корней определяется как

$$R = 1 + \sqrt[k]{\frac{B}{a_0}}, \quad a_0 > 0,$$

где a_k — первый отрицательный коэффициент; B — наибольшая абсолютная величина отрицательного коэффициента.

5. Теорема Гюа. Если уравнение (3.1) имеет действительные корни и действительные коэффициенты, то

$$a_k^2 > a_{k-1} a_{k+1}.$$

В строго математическом смысле, найдя один какой-нибудь корень x_1 , можно разделить уравнение на $(x - x_1)$, тем самым понизив его степень. Однако эта процедура может привести к накоплению ошибок в коэффициентах получаемых уравнений, а следовательно, и в корнях.

Прежде чем приступить к нахождению действительных корней, целесообразно вспомнить две теоремы, одна из которых позволяет отделить корни, т. е. установить возможно тесные промежутки $[\alpha, \beta]$, в которых содер-

жится один и только один корень уравнения, а вторая — оценить степень приближения.

Теорема 1. Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[\alpha, \beta]$, т. е. $f(\alpha)f(\beta) < 0$, то внутри этого отрезка содержится по меньшей мере один корень уравнения $f(x) = 0$, т. е. найдется хотя бы одно число $\xi \in (\alpha, \beta)$ такое, что $f(\xi) = 0$.

Теорема 2. Пусть ξ — точный, а \bar{x} — приближенный корни уравнения $f(x) = 0$, находящиеся на одном и том же отрезке $[\alpha, \beta]$, причем $|f'(x)| \geq m$, $\alpha < x < \beta$.

Тогда

$$|\bar{x} - \xi| \leq \frac{|f(\bar{x})|}{m}.$$

3.1. Методы нахождения действительных корней нелинейного уравнения

Существует ряд методов численного решения нелинейных уравнений, целесообразность применения каждого из которых определяется видом уравнения, его порядком, требуемой точностью и т. д. Эти методы подробно рассмотрены в [3, 7, 12, 23].

Метод половинного деления. В этом методе сначала вычисляются значения функции в точках, расположенных через равные интервалы на оси x . Когда $f(x_n)$ и $f(x_{n+1})$ имеют противоположные знаки, находят $x_{\text{ср}} = \frac{x_{n+1} + x_n}{2}$ и $f(x_{\text{ср}})$.

Если знак $f(x_{\text{ср}})$ совпадает со знаком $f(x_n)$, то в дальнейшем вместо $f(x_n)$ используется $f(x_{\text{ср}})$. Если же $f(x_{\text{ср}})$ имеет знак, противоположный знаку $f(x_n)$, т. е. совпадает со знаком $f(x_{n+1})$, то на $f(x_{\text{ср}})$ заменяется это значение. Если $f(x_{\text{ср}})$ достаточно близко к 0, то счет прекращается. Отметим, что в этом и в последующих методах в качестве условия прекращения итерационного процесса часто более целесообразно пользоваться условием $|x_{n+1} - x_n| \leq \varepsilon$, где ε — заданная погрешность нахождения корня.

Структура алгоритма представлена на рис. 3.2, а графически его применение проиллюстрировано на рис. 3.3.

Этот метод имеет малую скорость сходимости. По сравнению с впервые найденным интервалом, в котором заключен корень, его ширина после N итераций убывает в 2^N раз:

$$x_{n+1} - x_n = \frac{1}{2^N}(x_2 - x_1).$$

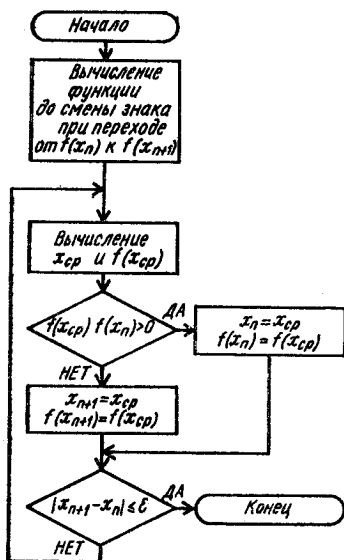


Рис. 3.2.

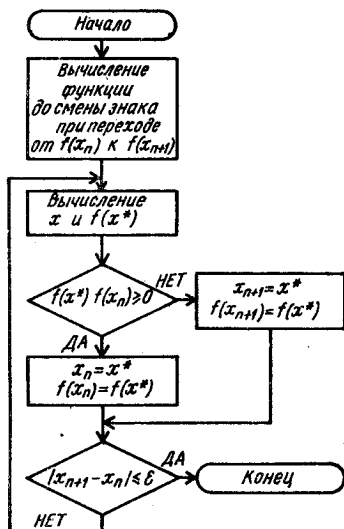


Рис. 3.4.

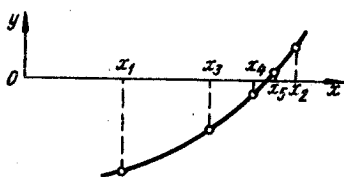


Рис. 3.3.

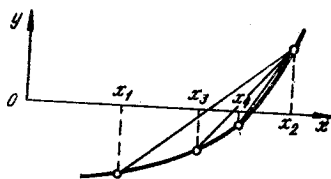


Рис. 3.5.

Погрешность найденного решения заключена в пределах

$$0 < \xi - x_n < \frac{1}{2^N}(x_2 - x_1).$$

Метод ложного положения (хорд). В основе этого метода лежит линейная интерполяция по двум значениям функции, имеющим противоположные знаки. При отыска-

нии корня метод обеспечивает более быструю сходимость, чем предыдущий. Структура алгоритма представлена на рис. 3.4. Определяются значения функции в точках, расположенных на оси через равные интервалы. Это делается до тех пор, пока $f(x_n)$ и $f(x_{n+1})$ не будут иметь разных знаков.

Прямая, проведенная через эти две точки, пересекает ось x при значении

$$x^* = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}.$$

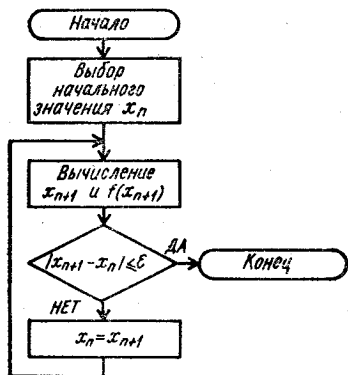


Рис. 3.6.

Далее определяют $f(x)$ и сравнивают его с $f(x_n)$ и $f(x_{n+1})$. В дальнейшем пользуются $f(x^*)$ вместо того значения, с которым оно совпадает по знаку. Если $f(x^*)$

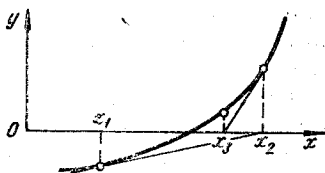


Рис. 3.7.

сильно отличается от 0, то вся процедура повторяется сначала. Этот процесс графически представлен на рис. 3.5. При $|x_{n+1} - x_n| \ll \epsilon$ можно считать, что $|\xi - x_n| \ll \epsilon$. Это справедливо при узком интервале и когда производная изменяется плавно (меньше чем в два раза).

Погрешность решения оценивается по формуле

$$|\xi - x_n| \leq \frac{M_1 - m_1}{m_1} |x_n - x_{n-1}|,$$

где M_1 и m_1 — соответственно наибольшее и наименьшее значение модуля производной на отрезке $[x_1, x_2]$.

Метод Ньютона (касательных). Метод Ньютона очень широко используется при построении итерационных алгоритмов. Его популярность объясняется тем, что, в отличие от двух предыдущих методов, для определения интервала, в котором заключен корень, не требуется находить значения функции с противоположными знаками. Вместо интерполяции (приближения) по двум значениям функции в методе Ньютона осуществляется

экстраполяция (предсказание) с помощью касательной к кривой в данной точке.

Структура алгоритма представлена на рис. 3.6, а графическая интерпретация — на рис. 3.7.

В основе этого алгоритма лежит разложение функции в ряд Тейлора [11]:

$$f(x_n + h) = f(x_n) + hf'(x_n) + \frac{h^2}{2} f''(x_n) + \dots$$

Члены, содержащие h во второй и более степенях, отбрасываются. Используется соотношение $x_n + h = x_{n+1}$. Предлагается, что переход от x_n к x_{n+1} приближает значение функции к нулю тем, что $f(x_n + h)$. Тогда

$$x_{n+1} = x_n - f(x_n)/f'(x_n). \quad (3.2)$$

Это значение соответствует точке, в которой касательная к кривой пересекает ось x . Затем процедура повторяется, причем вместо x_n используется x_{n+1} . Счет прекращается при достижении достаточно малого значения $f(x_{n+1})$.

Скорость сходимости в большой мере зависит от удачного выбора исходной точки. Если в процессе итераций тангенс угла наклона касательной $f'(x)$ обращается в нуль, то применение метода осложняется. Можно также показать, что в случае очень большого $f''(x)$ или кратных корней метод Ньютона не эффективен.

Начальное приближение x_0 выбирается из условия

$$(f'(x_0))^2 > f''(x_0) f(x_0) > 0.$$

Программа и пример реализации этого метода представлена на рис. 3.8.

Погрешность метода определяется порядком отбрасываемых членов при разложении в ряд Тейлора и оценивается как

$$|\xi - x_n| \leq \frac{M_2}{2m_1} (x_n - x_{n-1})^2,$$

где M_2 — наибольшее значение модуля второй производной на отрезке $[x_1, x_2]$.

Метод секущих. Один из недостатков метода Ньютона — необходимость нахождения производной $f'(x)$. Если нахождение $f'(x)$ затруднено, то можно воспользоваться некоторым приближением, составляющим основу метода секущих. Заменяв $f'(x_n)$ в методе Ньютона в уравнении

```

C
C
C      ПРОГРАММА РЕШЕНИЯ
C      НЕЛИНЕЙНОГО УРАВНЕНИЯ
C       $F(X)=X^{*}3-1-EXP(X)=0$ 
C      МЕТОДОМ НЬЮТОНА.
C      СЧЕТ ПРЕКРАЩАЕТСЯ ПРИ F МЕНЕЕ
C      0.00001 ИЛИ ЕСЛИ ЧИСЛО
C      ИТЕРАЦИЙ ПРЕВЫШАЕТ 30.
C

```

```

PROGRAM NJU
X=1.
PRINT100
100  FORMAT(X,20('-''))
PRINT101
101  &  FORMAT('  НОМЕР  ',
            &      2X,'ЗНАЧЕНИЕ')
PRINT102
102  &  FORMAT(' ИТЕРАЦИИ  ',
            &      2X,'      X      ')
PRINT100
PRINT104,X
104  FORMAT(' СТАРТ',5X,F10.4)
DO.1 I=1,30

C
C      ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИИ
C       $F=X^{*}3-1-EXP(X)$ 

C
C      ВЫЧИСЛЕНИЕ ПРОИЗВОДНОЙ
C      ФУНКЦИИ В ТОЧКЕ X
C       $DF=3.*X^{*}2-EXP(X)$ 
C       $X=X-F/DF$ 
PRINT103,I,X
1  IF(ABS(F).LE.0.00001)GOTO 2
103 FORMAT(X,I5,5X,F10.4)
2  PRINT100
STOP
END

```

НОМЕР ИТЕРАЦИИ	ЗНАЧЕНИЕ X
СТАРТ	1.0000
1	10.6489
2	9.6697
3	8.7097
4	7.7838
5	6.9139
6	6.1296
7	5.4660
8	4.9608
9	4.6469
10	4.5223
11	4.5040
12	4.5037
13	4.5037

Рис. 3.8.

(3.2) разностью последовательных значений функции, отнесенных к разности значений аргумента

$$F'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

получим

$$x_{n+1} = x_n - \frac{f(x_n)}{F'(x_n)}.$$

Структура алгоритма имеет тот же вид, что и для метода Ньютона (при другой итерационной формуле).

Метод секущих представляет собой комбинацию методов интерполяции и экстраполяции. В интерполяционной части он эквивалентен методу ложного положения, а в экстраполяционной — методу Ньютона.

Как и в методе Ньютона, счет заканчивается при достижении необходимой точности последовательных значений x или когда $f(x)$ близко к 0. Здесь возникают те же трудности, что и для отыскания кратных корней.

Метод простой итерации. Для применения этого метода уравнение $f(x) = 0$ представляется в следующем виде:

$$x = g(x).$$

Соответствующая итерационная формула имеет вид

$$x_{n+1} = g(x_n).$$

Структура алгоритма метода простой итерации (последовательных приближений) представлена на рис. 3.9.

Этот метод прост, но не всегда обеспечивает сходимость. Поэтому для любой программы, использующей этот алгоритм, необходим контроль сходимости и прекращение счета, если сходимость не обеспечивается.

Погрешность метода:

$$|\xi - x_n| \leq \frac{q^n}{1-q} |x_1 - x_0|,$$

где q — максимальное значение первой производной функции на отрезке $[x_1, x_2]$; если $q < 1$, то итерационный процесс сходится независимо от выбора начального значения $x_0 \in [a, b]$.

3.2. Методы нахождения комплексных корней при решении алгебраических уравнений

Для вычисления комплексных корней применимы те же методы, что и для вычисления действительных. При этом приходится оперировать арифметикой комплексных чисел, что не всегда удобно.

Существует ряд специальных методов нахождения комплексных корней. Все они почти всегда связаны

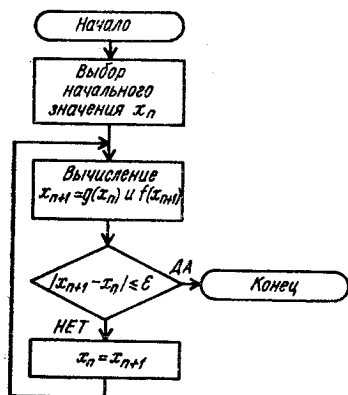


Рис. 3.9.

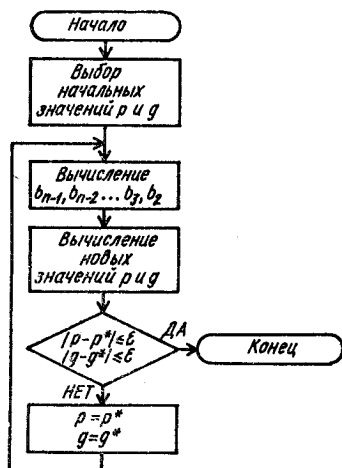


Рис. 3.10.

с выделением из исходного алгебраического уравнения квадратичного множителя $x^2 + px + q$. Широко применяемым методом этого типа является метод Лина. В его основе представление алгебраического уравнения

$$x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 = 0$$

в виде

$$(x^2 + px + q)(x^{n-2} + b_{n-1}x^{n-3} + \dots + b_3x + b_2) + b_1x + b_0 = 0,$$

где $b_1x + b_0$ — линейный остаточный член, который стремятся сделать равным нулю, если это возможно, то исходное уравнение делится на квадратичный множитель без остатка. Предположим, что $b_1 = b_0 = 0$, тогда имеем

$$b_{n-1} = a_{n-1} - p;$$

$$b_{n-2} = a_{n-2} - pb_{n-1} - q;$$

$$b_{n-j} = a_{n-j} - pb_{n+1-j} - qb_{n+2-j};$$

$$p = \frac{a_1 - qb_3}{b_2};$$

$$q = \frac{a_0}{b_0}.$$

Структура алгоритма, реализующего метод Лина, представлена на рис. 3.10.

3.3. Решение систем нелинейных уравнений

В отличие от систем линейных уравнений для нелинейных уравнений не существует прямых методов решения, и поэтому всегда применяются итерационные методы. В общем случае система представляется в виде

$$f_1(x_1, x_2, \dots, x_n) = 0;$$

$$f_2(x_1, x_2, \dots, x_n) = 0;$$

.....

$$f_n(x_1, x_2, \dots, x_n) = 0.$$

Именно к такому виду сводятся многие задачи в автоматике, электронике, информационно-измерительной технике, например, в задаче о разработке модели электронных устройств (п. 1.3).

Метод простой итерации. Метод простой итерации для решения систем нелинейных уравнений является развитием аналогичного метода для решения одного уравнения. Он основан на допущении, что систему можно привести к виду

$$x_1^* = q_1(x_1, x_2, \dots, x_n);$$

$$x_2^* = q_2(x_1^*, x_2, \dots, x_n);$$

.....

$$x_n^* = q_n(x_1^*, x_2^*, \dots, x_n).$$

Структура алгоритма представлена на рис. 3.11.

Эта разновидность метода простой итерации построена аналогично методу Зейделя для систем линейных уравнений, когда для нахождения каждого последующего значения неизвестных используются новые найденные значения предыдущих неизвестных.

Метод имеет свои недостатки. Например, если исходные значения неизвестных слишком сильно отличаются от истинного решения, то процесс не сойдется. Область, в которой заданные исходные значения сходятся к решению, называется областью сходимости. С увеличением числа неизвестных область сходимости уменьшается, и в случае больших систем сходимость обеспечивается лишь в том случае, когда исходные значения очень близки к решению.

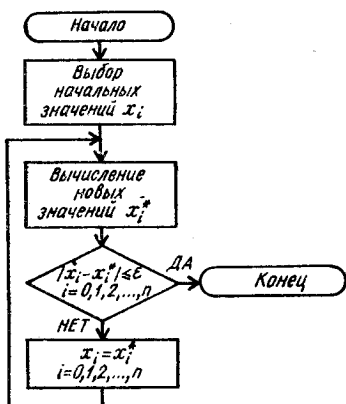


Рис. 3.11.

Метод Ньютона. Это наиболее распространенный метод для решения систем нелинейных уравнений. Он обеспечивает гораздо более быструю сходимость, чем метод простой итерации. В основе метода лежит представление всех n уравнений в виде рядов Тейлора:

$$\begin{aligned}
 f_1(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) &= f_1(x_1, \dots, x_n) + \\
 &+ \Delta x_1 \frac{\partial f_1}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_1}{\partial x_n} + R_n; \\
 f_n(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) &= f_n(x_1, \dots, x_n) + \\
 &+ \Delta x_1 \frac{\partial f_n}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_n}{\partial x_n} + R_n,
 \end{aligned}$$

где R_n — члены второго и более высоких порядков.

Если приращения переменных Δx_i таковы, что f_i принимает значение, близкое к корню, то будем считать, что левые части этих уравнений обращаются в нули.

Таким образом, задача сводится к отысканию такой совокупности приращений Δx_i , при которой достигается указанная цель. Отбросив R_n , задача сводится к решению системы линейных уравнений:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \dots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \dots \\ -f_n \end{bmatrix}.$$

В этой системе матрицу частных производных, которая называется матрицей Якоби, и вектор-столбец правой части можно оценить для любого приближенного решения. Найденные Δx_i используются как поправка к исходному приближенному решению

$$\begin{aligned} x_1 &= x_1 + \Delta x_1; \\ &\dots \\ x_n &= x_n + \Delta x_n. \end{aligned}$$

Окончательная итерационная формула имеет вид

$$x^{(n+1)} = x^{(n)} + W^{-1}(x^{(n)}) F(x^{(n)}),$$

где $x^{(n)}$, $x^{(n+1)}$ — векторы-столбцы приближенных решений системы соответственно на (n) и $(n+1)$ шаге итерационного процесса; $F(x^{(n)})$ — вектор-столбец значений функций f_1, f_2, \dots, f_n для приближенных значений $x^{(n)}$; $W^{-1}(x)$ — матрица, обратная матрице Якоби

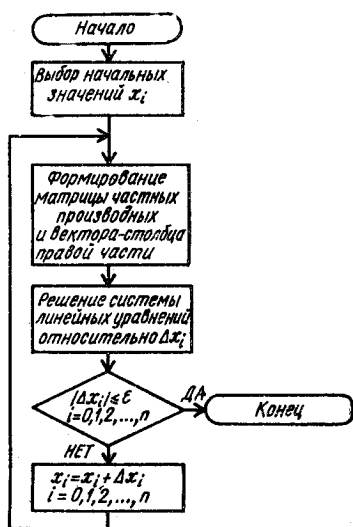


Рис. 3.12.

Если все $\Delta x_i \rightarrow 0$, то счет прекращается, в противном случае новые значения используются как приближенные значения корней, и процесс повторяется до тех пор, пока не будет найдено решение или станет ясно, что процесс расходится. Структура алгоритма представлена на рис. 3.12, а программа для его реализации и пример использования — на рис. 3.13.

```

C      ПРОГРАММА РЕШЕНИЯ МЕТОДОМ
C      НЬЮТОНА СИСТЕМЫ УРАВНЕНИЙ
      REAL*4 F, W, W0, KL, E, X, EPS
      REAL*4 DET, Q, U, U1, A, L, C
      INTEGER I, J, I1, J1, K
      DIMENSION F(4), W(4,4),
*          WT(4,4), W0(4,4),
*          KL(3,3),
*          E(4), X(4), Q(3),
*          A(4,4), C(4)
      DO 15 I=1,4
          X(I)=1
15      CONTINUE
          K=0
          EPS=1, E-05
          CALL FUN(X, F)
20
C      ОПРЕДЕЛЕНИЕ ЭЛЕМЕНТОВ
C      МАТРИЦЫ ЯКОБИ
          W(1,1)=1
          W(1,2)=2
          W(1,3)=1
          W(1,4)=4
          W(2,1)=2*(X(1)+X(2))
          W(2,2)=2*X(1)
          W(2,3)=0
          W(2,4)=3*(X(4)**2)
          W(3,1)=2*X(1)
          W(3,2)=0
          W(3,3)=2*X(3)
          W(3,4)=1
          W(4,1)=0
          W(4,2)=3
          W(4,3)=X(4)
          W(4,4)=X(3)
C
C      ОПРЕДЕЛЕНИЕ ЭЛЕМЕНТОВ МАТРИЦЫ
C      АЛГЕБРАИЧЕСКИХ ДОПОЛНЕНИЙ
C      ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ
          DET=0
          DO 1 I=1,4,1
              DO 2 J=1,4,1
                  DO 5 I1=1,4,1
                      DO 6 J1=1,4,1
                          IF((I1, EQ, I), OR,
& (J1, EQ, J)) GOTO 7
                          IF((I1, LT, I).AND,
& (J1, LT, J)) KL(I1, J1)=W(I1, J1)
                          IF((I1, LT, I).AND,
& (J1, GT, J)) KL(I1, J1-1)=W(I1, J1)
                          IF((I1, GT, I).AND,
& (J1, LT, J)) KL(I1-1, J1)=W(I1, J1)
                          IF((I1, GT, I).AND,
& (J1, GT, J)) KL(I1-1, J1-1)=W(I1, J1)
                          L=0
          CONTINUE

```

7
6
5

```

Q(1)=KL(1,1)*(KL(2,2)*KL(3,3)-
& KL(2,3)*KL(3,2))
Q(2)=-KL(1,2)*(KL(2,1)*KL(3,3)-
& KL(3,1)*KL(2,3))
Q(3)=KL(1,3)*(KL(2,1)*KL(3,2)-
& KL(3,1)*KL(2,2))
L=(-1)**(I+J)
A(I,J)=L*(Q(1)+Q(2)+Q(3))
2 CONTINUE
1 CONTINUE
DET=W(1,1)*A(1,1)+W(1,2)*A(1,2)+
& W(1,3)*A(1,3)+W(1,4)*A(1,4)
C
C ОПРЕДЕЛЕНИЕ:
C WT(I,J) - ЭЛЕМЕНТЫ ТРАНСП. МАТРИЦЫ
C WO(I,J) - ЭЛЕМЕНТЫ ОБРАТНОЙ МАТРИЦЫ
C
DO 10 I=1,4,1
DO 11 J=1,4,1
WT(I,J)=A(J,I)
WO(I,J)=WT(I,J)/DET
11 CONTINUE
10 CONTINUE
C
C ОСУЩЕСТВЛЕНИЕ ИТЕРАЦИОННОГО ПРОЦЕССА
C
DO 13 I=1,4,1
E(I)=WO(I,1)*F(1)+WO(I,2)*
& F(2)+WO(I,3)*F(3)+WO(I,4)*F(4)
X(I)=X(I)-E(I)
13 CONTINUE
C
C ПРОВЕРКА ПОГРЕШНОСТИ
C И ВЫВОД РЕЗУЛЬТАТА
C
U=ABS(E(1))
DO 14 I=2,4
U1=ABS(E(I))
IF(U.LT.U1)U=U1
14 CONTINUE
IF(U.GT.EPS)GOTO 20
K=K+1
WRITE(5,25)K,U
25 FORMAT(X,'КОЛ-ВО ИТЕРАЦИЙ : K=',
& I2,' U=',F15.10)
DO 26 I=1,4
WRITE(5,30)I,X(I)
30 FORMAT(X,'X(',I1,')=',F10.6)
26 CONTINUE

```

```

C      ПРОВЕРКА
C
54      WRITE(5,54)
        FORMAT(//X,
&      ' ПРОВЕРКА РЕЗУЛЬТАТА' /)
        CALL FUN(X,F)
        DO 40 I=1,4
          WRITE(5,35) I, F(I)
35      FORMAT(X, ' F(', I1, ') = ', F12.7)
40      CONTINUE
        STOP
        END

C
C      ПОДПРОГРАММА ВЫЧИСЛЕНИЯ
C      ЗНАЧЕНИЙ ФУНКЦИЙ.
C
SUBROUTINE FUN(X,F)
REAL F(4),X(4)
F(1)=X(1)+2.*X(2)+X(3)+
&      X(4)*4.-20.7
F(2)=X(1)**2+2*X(1)*X(2)+
&      X(4)**3-15.88
F(3)=X(1)**2+X(3)**2+
&      X(4)-21.218
F(4)=3*X(2)+X(3)*X(4)-7.9
RETURN
END

```

КОЛИЧЕСТВО ИТЕРАЦИЙ : K= 1	ПРОВЕРКА РЕЗУЛЬТАТА
U= 0.0000026143	
X(1)= -0.941679	F(1)= 0.0000000
X(2)= 6.836753	F(2)= 0.0000019
X(3)= -4.159253	F(3)= 0.0000019
X(4)= 3.031857	F(4)= 0.0000005

Рис. 3.13.

При проверке сходимости следует пользоваться условием

$$\frac{\Delta x_i}{x_i} \leq \varepsilon,$$

где ε — допустимая погрешность нахождения решения.

Для метода Ньютона тоже существует проблема сходимости. Величина области сходимости обратно пропорциональна степени сложности и числу уравнений. Существуют способы оценки области сходимости и выбора исходного приближения [12].

Если в этом методе точные значения производных найти не удастся, то можно пользоваться их приближенными значениями, найденными методом секущих.

Метод возмущения параметров. Этот метод реализует алгоритм, позволяющий с помощью итераций получить решение системы нелинейных уравнений. Его эффективность не зависит от выбора начального приближения. Суть этого метода состоит в следующем. Сначала наряду с системой уравнений

$$f_j(x_i) = 0, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n,$$

рассмотрим

$$g_j(x_i) = 0,$$

решение которой известно. Затем, изменяя уравнение $g_j = 0$, превратим его в уравнение $f_j = 0$ с помощью конечного числа последовательных малых приращений параметров:

$$g_i^{(k)}(x_i) = g_i^{(k-1)}(x_i) + [f_i(x_i) - g_i^{(k-1)}(x_i)] \frac{k}{N},$$

где $k = 1, 2, \dots, N$.

Решение $x_i^{(0)}$ исходной системы уравнений $g_j^{(0)}(x_i)$ можно использовать как исходные значения переменных для итерационного решения системы $g_j^{(1)}(x_i)$. Эта система мало отличается от предыдущей, и можно сделать вывод, что сходимость будет обеспечена. В процессе счета решение $x_i^{(k-1)}$ используется как исходное для получения решения x_i^k . Когда $k = N$, система превращается в исходную. Поскольку для этого превращения часто требуется много шагов, метод может потребовать больших затрат времени ЭВМ. Но его преимущество в том, что при малом числе шагов сходимость на каждом шаге достигается всего за несколько итераций.

3.4. Выбор метода решения нелинейных уравнений и систем

При выборе метода и разработке алгоритма решения нелинейных уравнений и систем в первую очередь следует принимать во внимание порядок уравнения, характер изменения функции $f(x)$ (гладкость, разрывность), требования к точности и скорости получения решения, имеющиеся ресурсы ЭВМ.

Наиболее быстродействующий метод, позволяющий значительно сократить объем вычислений как для уравнений, так и для систем, — метод Ньютона. При слож-

ности нахождения производных он может быть заменен методом секущих. Однако эти методы имеют ряд ограничений и их применение эффективно лишь для гладких функций.

При неизвестном характере изменения $f(x)$ наиболее надежен метод половинного деления.

В зависимости от порядка уравнения может быть рекомендовано использование следующих методов [23]: при порядке от 3 до 5 — Ньютона; от 6 до 85 — секущих; свыше 85 — специальных методов.

Вместо составления собственного алгоритма иногда удобно пользоваться имеющимся в распоряжении стандартным математическим обеспечением ЭВМ или пакетами прикладных программ [20].

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Приведите классификацию нелинейных уравнений и систем.
2. Сформулируйте и докажите основную теорему алгебры.
3. Сформулируйте и докажите теорему Декарта.
4. Сформулируйте теоремы Гюа и Лагранжа.
5. Докажите теорему 1.
6. Докажите теорему 2.
7. Разработайте и нарисуйте структуру алгоритма, реализующего решение нелинейного уравнения методом половинного деления. Предусмотрите в алгоритме оценку погрешности решения, счет количества шагов. Составьте программу для ЭВМ на языке ФОРТРАН для решения уравнения $x^3 - x + 1 = 0$.
8. Представьте структуру алгоритма и составьте программу для ЭВМ решения уравнения из задания 7 методом хорд. Сравните эффективность метода половинного деления и метода хорд.
9. В чем отличия методов Ньютона и секущих и в каких случаях их использование не эффективно? Приведите примеры.
10. Составьте программу для решения на ЭВМ уравнения $x^2 - x + \operatorname{tg} x = 0$ методом простой итерации. Предусмотрите в программе контроль сходимости.
11. Составьте программу для решения нелинейного уравнения с комплексными корнями методом Лина.
12. Почему для решения систем нелинейных уравнений возможно применение только итерационных методов?
13. Дайте сравнительную оценку методов решения систем нелинейных уравнений.
14. Составьте программу для решения на ЭВМ методом Ньютона системы нелинейных уравнений.
15. Какие рекомендации в зависимости от вида и порядка уравнений могут быть использованы при выборе метода решения нелинейных уравнений на ЭВМ?
16. Приведите известные Вам процедуры и программы для решения нелинейных уравнений и систем из стандартного математического обеспечения ЭВМ.

Глава 4. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Во многих практических задачах приходится сталкиваться с необходимостью решения уравнений, содержащих одну или несколько производных. Такие уравнения описывают большинство законов физики, к их решению сводится рассмотрение различных электронных и автоматических систем. В зависимости от числа независимых переменных и, следовательно, типа входящих в них производных дифференциальные уравнения делятся на обыкновенные дифференциальные, содержащие одну независимую переменную и производные по ней, и уравнения в частных производных, имеющие несколько независимых переменных и производные (частные) по ним.

Существует много приемов для нахождения решений дифференциальных уравнений через элементарные или специальные функции (например, функции Бесселя) [11]. Такие методы называют аналитическими, или классическими, однако в большинстве задач они либо вообще неприменимы, либо приводят к сложным и громоздким расчетам. При задании коэффициентов или функций в дифференциальных уравнениях в виде таблиц экспериментальных данных использование классических методов принципиально невозможно. Это обуславливает важность рассматриваемых в этой главе численных методов решения дифференциальных уравнений, являющихся основой при составлении алгоритмов и программ для ЭВМ.

4.1. Постановка задачи

Обыкновенное дифференциальное уравнение имеет бесконечное множество решений. Для отыскания какого-либо конкретного решения требуются дополнительные условия. Эти условия могут быть различными. В случае, когда дополнительные условия задаются при одном значении независимой переменной, имеет место задача Коши (задача с начальными условиями). Если же условия задаются при двух или более значениях независимой переменной, то задача называется краевой. В задаче Коши дополнительные условия называются начальными, а в краевой — граничными. При решении этих задач используются разные методы и алгоритмы.

Сформулируем задачу Коши.

Пусть дано дифференциальное уравнение

$$\frac{\partial y}{\partial x} = f(x, y) \quad (4.1)$$

и начальное условие $y(0) = y$. Требуется найти функцию на отрезке от $x = a$ до $x = b$, удовлетворяющую как указанному уравнению, так и начальному условию (при этом всегда предполагается, что существует единственное решение на всем отрезке).

Краевую задачу рассмотрим на примере обыкновенного дифференциального уравнения второго порядка

$$\frac{\partial^2 y}{\partial x^2} = f(x, y, y')$$

при граничных условиях $y(a) = A$, $y(b) = B$. Методы решения уравнений более высоких порядков аналогичны.

4.2. Методы решения задачи Коши

В основе численных методов решения дифференциальных уравнений лежит разложение функции y в ряд Тейлора в окрестностях исходной точки x_0 :

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2} h^2 y''(x_0) + \dots \\ \dots + \frac{1}{i!} h^i y^{(i)}(x_0),$$

где h — расстояние (шаг) между исходной точкой x_0 и точкой $x_1 = x_0 + h$, в которой отыскивается решение.

Причем в разных методах учитывается разное количество членов разложения (в многошаговых методах в сочетании с интерполяционными формулами), что определяет точность вычислений. Считают, что порядок ошибки равен p , если существует такое число C , и

$$|\Delta| \leq Ch^{p+1},$$

где Δ — локальная ошибка; h — шаг дискретизации.

Число C не зависит от номера шага и его величины, а определяется производными и длиной интервала. При аппроксимации решения рядами Тейлора оно связано со степенью отбрасываемых членов ряда.

Методы решения задачи Коши можно разделить на две группы: одношаговые, в которых для нахождения следу-

ющей точки на кривой $y = f(x)$ требуется информация лишь об одном предыдущем шаге (методы Эйлера и Рунге — Кутты); многошаговые (прогноза и коррекции), в которых для нахождения следующей точки на кривой $y = f(x)$ требуется информация более чем об одной из предыдущих точек. Для получения достаточно точного численного значения часто используется итерационная процедура (например, в методах Милна, Адамса — Башфорта и Хемминга).

Одношаговые методы. Наиболее простым одношаговым методом, требующим минимальных затрат вычислительных ресурсов, но обладающим сравнительно низкой точностью, является метод Эйлера.

В этом методе для оценки следующей точки на кривой $y = f(x)$ используется только один линейный член в формуле Тейлора (рис. 4.1):

$$\begin{aligned} y(x_0 + h) &= \\ &= y(x_0) + hy'(x_0), \end{aligned}$$

где $y'(x_0)$ определяется из исходного уравнения (4.1).

Этот процесс можно распространить на последующие шаги:

$$y_{n+1} = y_n + hf(x_n, y_n).$$

Метод Эйлера является методом первого порядка:

$$|\Delta| \ll Ch^2,$$

где

$$\begin{aligned} C &= (M_1 + M_0 M_2)/2; \\ M_0 &\geq |f(x, y)|; \\ M_1 &\geq \left| \frac{\partial x(x, y)}{\partial x} \right|; \\ M_2 &\geq \left| \frac{\partial f(x, y)}{\partial y} \right| \end{aligned}$$

для всех $x \in [a, b]$ и $y = y(x)$.

Метод Эйлера, кроме значительной погрешности усечения, часто бывает неустойчив (малые локальные ошибки приводят к значительному увеличению глобальной). Этот метод можно усовершенствовать различными

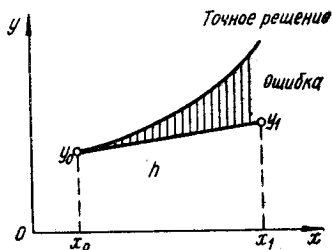


Рис. 4.1.

способами. Наиболее известны два из них: исправленный метод Эйлера и модифицированный метод Эйлера.

Итерационные формулы для этих методов следующие:

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_n + h, y_n + hy_n^*)];$$

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} y_n^*\right), \quad (4.2)$$

где

$$y_n^* = f(x_n, y_n).$$

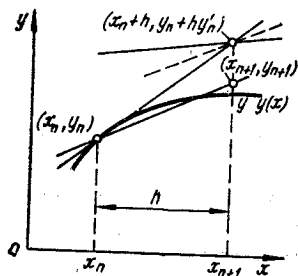


Рис. 4.2.

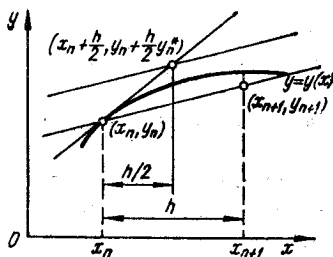


Рис. 4.3.

Геометрическая интерпретация представлена на рис. 4.2 и 4.3. Погрешность этих методов имеет второй порядок, что достигается улучшением аппроксимации производной. Идея заключается в попытке сохранить или оценить член второго порядка в формуле Тейлора. Однако повышение точности связано с дополнительными затратами машинного времени на вычисление y_n^* . Еще более высокая точность может быть достигнута при вычислении высших производных и сохранении большего числа членов ряда Тейлора. Такими методами являются методы Рунге — Кутты.

Принцип, на котором основан модифицированный метод Эйлера, можно пояснить, пользуясь рядом Тейлора и сохранив в нем член с h^3 , аппроксимировав при этом вторую производную $y''(x_n)$ конечной разностью

$$y''(x_n) = \frac{\Delta y'}{\Delta x} = \frac{y'(x_n + h) - y'(x_n)}{h},$$

что совпадает с (4.2) при условии замены $y'(x_n + h)$ на $f(x_{n+1}, y_{n+1})$.

Аналогично вычислению второй производной в конечно-разностном виде можно вычислить и другие про-

изводные, например n -ю, по значениям предыдущей $(n - 1)$ -й.

Метод Рунге — Кутта дает набор формул для расчета координат внутренних точек, требуемых для реализации этой идеи. Поскольку существует ряд способов нахождения этих точек, метод Рунге — Кутта объединяет семейство методов для решения дифференциальных уравнений первого порядка. Чаще других используется классический метод — метод четвертого порядка точности

$$y_{n+1} = y_n + \frac{K_0 + 2K_1 + 2K_2 + K_3}{6},$$

где

$$K_0 = hf(x_n, y_n); K_1 = hf\left(x_n + \frac{h}{2}; y_n + \frac{K_0}{2}\right);$$

$$K_2 = hf\left(x_n + \frac{h}{2}; y_n + \frac{K_1}{2}\right); K_3 = hf(x_n + h; y_n + K_2).$$

Метод Эйлера и его модификации по существу являются методами Рунге — Кутта первого и второго порядков. Метод Рунге — Кутта имеет значительно более высокую точность, позволяет увеличить шаг интегрирования. Его максимальную величину определяет допустимая погрешность. Такой выбор часто осуществляется автоматически и включается как составная часть в алгоритм, построенный по методу Рунге — Кутта. На рис. 4.4 представлен пример решения на ЭВМ обыкновенного дифференциального уравнения одношаговыми методами и сравнения их погрешностей.

Любую из формул Рунге — Кутта можно использовать для решения дифференциальных уравнений более высоких порядков и, следовательно, для решения систем дифференциальных уравнений, так как уравнение более высокого порядка (n) можно свести к n дифференциальным уравнениям первого порядка.

В качестве примера рассмотрим решение обыкновенного дифференциального уравнения второго порядка

$$\frac{d^2y}{dx^2} = g\left(x, y, \frac{dy}{dx}\right).$$

Примем

$$z = \frac{dy}{dx} \Rightarrow \begin{cases} \frac{dz}{dx} = g(x, y, z); \\ \frac{dy}{dx} = f(x, y, z), \end{cases}$$

где $f(x, y, z) = z$.

```

PROGRAM DU
REAL X(11)
REAL YT(11)
REAL YE(11)
REAL YEM(11)
REAL YRK(11)
REAL K0, K1, K2, K3
REAL IE, IEM, IRK
F(X, Y)=2.*X**2+2.*Y
H=0.1
X(1)=0.
YT(1)=1.
YE(1)=1.
YEM(1)=1.
YRK(1)=1.
DO 10 I=1, 11
  X(I)=(I-1)*H
  YT(I)=1.5*EXP(2.*X(I))-
    X(I)**2-X(I)-0.5
10 & CONTINUE
PRINT1010
PRINT1020
PRINT1030
PRINT1040
PRINT1010
DO 20 I=2, 11
  YE(I)=YE(I-1)+H*F(X(I-1),
&     YE(I-1))
  YEM(I)=YEM(I-1)+0.5*H*(F(X(I),
&     YE(I))+F(X(I-1), YEM(I-1)))
  K0=H*F(X(I-1), YRK(I-1))
  K1=H*F((X(I-1)+0.5*H),
&     (YRK(I-1)+0.5*K0))
  K2=H*F((X(I-1)+0.5*H),
&     (YRK(I-1)+0.5*K1))
  K3=H*F((X(I-1)+H),
&     (YRK(I-1)+K2))
  YRK(I)=YRK(I-1)+
&     (K0+2.*K1+2.*K2+K3)/6.
  IE=ABS(YE(I)-YT(I))*100./
&     YT(I)
  IEM=ABS(YEM(I)-YT(I))*100./
&     YT(I)
  IRK=ABS(YRK(I)-YT(I))*100./
&     YT(I)
  PRINT1050, X(I), YE(I),
&     YEM(I), YRK(I), YT(I)
  PRINT1060, IE, IEM, IRK
  PRINT1010
20 CONTINUE
PRINT*, ' ПРИМЕЧАНИЕ:',
& ' В СКОБКАХ УКАЗАНА ПОГРЕШНОСТЬ.'
STOP
1000 FORMAT(X, F3.1, 3(5X, F6.4, 2X,
&     F6.3, ' %'), 5X, F6.4)
1010 FORMAT(X, 66(' -'))

```

```

1020  FORMAT(' |          | МЕТОД          |')
      &      ' |          | МЕТОД          |',
      &      ' |          | МЕТОД          |',
      &      ' | ТОЧНОЕ |')
1030  FORMAT(' | XN | ЭЙЛЕРА |',
      &      ' | ЭЙЛЕРА | РУНГЕ-КУТТА |',
      &      ' | РЕШЕНИЕ |')
1040  FORMAT(' |          |          |',
      &      ' | МОДИФИЦИРОВАННЫЙ |',
      &      ' |          |')
1050  FORMAT(' | 'F3.1,' | 'F7.4,'
      &      ' | 'F7.4,' | 'F7.4,'
      &      ' | 'F7.4,' | 'F7.4,'
1060  FORMAT(' | '3X,' | 'F5.2,'
      &      ' | 'F5.2,' | 'F5.2,'
      &      ' | 'F6.4,' | '7X,' |')
      END

```

	МЕТОД ЭЙЛЕРА	МЕТОД ЭЙЛЕРА МОДИФ.	МЕТОД РУНГЕ-КУТТА	ТОЧНОЕ РЕШЕНИЕ
0.1	1.2000 (1.81 %)	1.2210 (0.09 %)	1.2221 (0.0002 %)	1.2221
0.2	1.4420 (3.72 %)	1.4923 (0.36 %)	1.4977 (0.0004 %)	1.4977
0.3	1.7384 (5.68 %)	1.8284 (0.80 %)	1.8432 (0.0007 %)	1.8432
0.4	2.1041 (7.65 %)	2.2466 (1.39 %)	2.2783 (0.0009 %)	2.2783
0.5	2.5569 (9.57 %)	2.7680 (2.10 %)	2.8274 (0.0012 %)	2.8274
0.6	3.1183 (11.42 %)	3.4176 (2.91 %)	3.5201 (0.0014 %)	3.5202
0.7	3.8139 (13.18 %)	4.225 (3.80 %)	4.3927 (0.0017 %)	4.3928
0.8	4.6747 (14.84 %)	5.2288 (4.75 %)	5.4894 (0.0019 %)	5.4895
0.9	5.7377 (16.42 %)	6.4704 (5.74 %)	6.8643 (0.0021 %)	6.8645
1.0	7.0472 (17.90 %)	8.0032 (6.76 %)	8.5834 (0.0024 %)	8.5836

ПРИМЕЧАНИЕ: В СКОБКАХ УКАЗАНА ПОГРЕШНОСТЬ.

Рис.4.4.

Задача Коши в этом случае содержит два начальных условия $y(x_0) = y_0$ и $z(x_0) = z_0$.

Формулы Рунге — Кутта для рассматриваемого случая имеют вид

$$y_{n+1} = y_n + K \text{ и } z_{n+1} = z_n + L,$$

где

$$K = \frac{K_1 + 2K_2 + 2K_3 + K_4}{6}; \quad L = \frac{L_1 + 2L_2 + 2L_3 + L_4}{6}.$$

Здесь

$$K_1 = hf(x_n, y_n, z_n), \quad L_1 = hg(x_n, y_n, z_n);$$

$$K_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}, z_n + \frac{L_1}{2}\right),$$

$$L_2 = hg\left(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}, z_n + \frac{L_1}{2}\right);$$

$$K_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}, z_n + \frac{L_2}{2}\right),$$

$$L_3 = hg\left(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}, z_n + \frac{L_2}{2}\right);$$

$$K_4 = hf(x_n + h, y_n + K_3, z_n + L_3),$$

$$L_4 = hg(x_n + h, y_n + K_3, z_n + L_3).$$

Ранее было отмечено, что ошибка усечения при использовании метода Рунге — Кутта n -го порядка составляет

$$\Delta = Ch^{n+1}.$$

Вычисление верхних пределов для коэффициента C представляет собой сложную задачу, связанную с необходимостью оценки ряда дополнительных параметров. В [13] приведены соотношения для оценки этой величины при $n = 2, 3$ и 4 . Существует несколько способов для оперативного вычисления C . Например, экстраполяционный Ричардсона, когда последовательно находят значения y_n с шагом h и с шагом $\frac{h}{2}$, а затем, приравнявая полученные величины, определяют C из уравнения

$$y_n^{(h)} + Ch^{n+1} = y_n^{(\frac{h}{2})} + C\left(\frac{h}{2}\right)^{n+1}.$$

Получим оценочное соотношение

$$C = \frac{2^{n+1}}{2^{n+1} - 1} \cdot \frac{[y_n^{(\frac{h}{2})} - y_n^{(h)}]}{h^{n+1}}.$$

В [13] приведено следующее правило для контроля погрешности на шаге при использовании метода Рунге — Кутта четвертого порядка: если $\xi = \left| \frac{K_2 - K_3}{K_1 - K_2} \right|$ превышает несколько сотых, то шаг необходимо уменьшить.

Рассмотрим общие черты одношаговых методов.

1. Для получения информации в новой точке нужны данные лишь об одной предыдущей точке.

2. В основе всех одношаговых методов лежит разложение функции в ряд Тейлора, в котором сохраняются члены, содержащие h в степени до k включительно. Целое число k называется порядком метода. Погрешность на шаге имеет порядок $k + 1$.

3. Одношаговые методы не требуют вычисления производных — вычисляется лишь функция, но может потребоваться ее значение в нескольких промежуточных точках.

4. Возможность изменения величины шага вычислений.

Многошаговые методы. Для вычисления значения новой точки в этих методах используется информация о нескольких ранее полученных значениях. Для этого используются две формулы: прогноза и коррекции. Алгоритм вычислений для всех методов прогноза и коррекции одинаков и представлен на рис. 4.5. Разные методы различаются лишь формулами.

Величину шага вычислений в этих методах изменить нельзя, поэтому, прежде чем применять метод прогноза и коррекции, приходится вычислять исходные данные с помощью какого-либо одношагового метода, например метода Рунге — Кутта.

Вычисления выполняем следующим образом. Сначала по формуле прогноза и исходным значениям переменных определяем значение $y_{n+1}^{(0)}$. Индекс (0) означает, что прогнозируемое значение является одним из последовательности значений y_{n+1} по мере их уточнения. По значению $y_{n+1}^{(0)}$ с помощью исходного дифференциального уравнения (4.1) находим производную

$$y_{n+1}^{(0)'} = f(x_{n+1}, y_{n+1}^{(0)}),$$

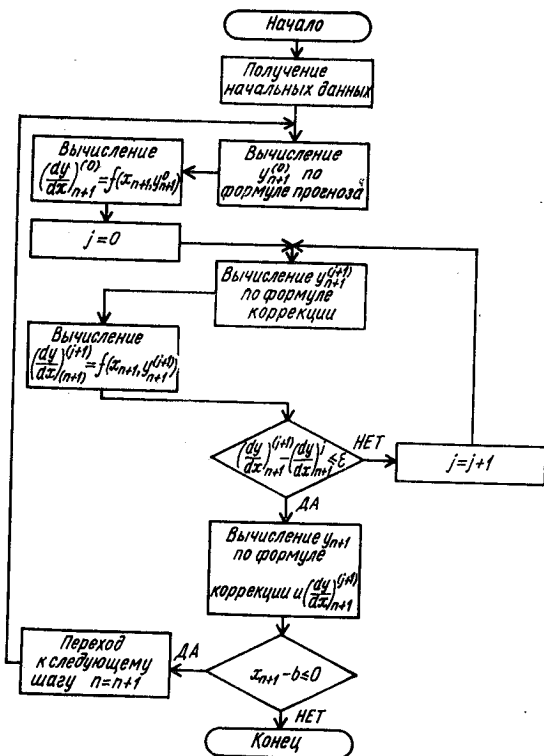


Рис. 4.5.

которую затем подставляем в формулу коррекции для вычисления уточненного значения $y_{n+1}^{(j+1)}$. Далее по $y_{n+1}^{(j+1)}$ определяем производную

$$y_{n+1}^{(j+1)} = f(x_{n+1}, y_{n+1}^{(j+1)}).$$

Если это значение недостаточно близко к предыдущему, то оно вводится в формулу коррекции, и итерационный процесс продолжается. В случае близости значений производных определяется y_{n+1} , которое и является окончательным. После этого процесс повторяется — делается следующий шаг, на котором вычисляется y_{n+2} .

Обычно при выводе формул прогноза и коррекции решение уравнения рассматривают как процесс приближенного интегрирования, а сами формулы получают с помощью конечно-разностных методов. Если дифферен-

циальное уравнение $y' = f(x, y)$ проинтегрировать в интервале значений от x_n до x_{n+k} , то получим следующий результат:

$$y(x_{n+k}) - y(x_n) = \int_{x_n}^{x_{n+k}} f(x, y) dx.$$

Этот интеграл нельзя вычислить непосредственно, так как $y(x)$ неизвестная функция. Приближенное значение интеграла можно найти с помощью одного из конечно-разностных методов. Выбор метода и будет определять метод решения дифференциальных уравнений. На этапе прогноза можно использовать любую формулу численного интегрирования, если в нее не входит предварительное значение $y'(x_{n+1})$.

В табл. 4.1 сведены наиболее часто используемые формулы прогноза и коррекции.

Для большинства методов прогноза и коррекции можно оценить погрешность, применив следующее соотношение [13]:

$$\Delta = \frac{1}{5} [y_n^{(0)} - y_n^{(i)}].$$

При выборе величины шага можно воспользоваться условием

$$h < \frac{2}{M_2}, \quad (4.3)$$

выполнение которого необходимо для сходимости итерационного процесса отыскания решения.

Однако во многих случаях сложность оценки величины, стоящей в знаменателе выражения (4.3), приводит к тому, что наиболее удобным для выбора шага является способ, основанный на оценке Δ в процессе вычислений и уменьшении шага, если погрешность слишком велика. При этом необходимо учитывать, что оптимальное число итераций равно двум.

Можно выделить следующие основные особенности многошаговых методов:

- 1) для решения задачи необходима информация о значениях функции в нескольких точках.
- 2) возможность получения оценки погрешности усечения без вычисления дополнительных величин.
- 3) методы прогноза и коррекции не позволяют легко изменять шаг вычислений. Для этого необходимо начинать заново весь итерационный процесс.

Таблица 4.1. Формулы прогноза и коррекции

Метод	Формула прогноза	Формула коррекции
Милна	$y_{n+1} = y_{n-3} + \frac{4}{3} h (2y'_n - y'_{n-1} + 2y'_{n-2}) + \frac{28}{90} h^5 y^{(5)}$	$y_{n+1} = y_{n-1} + \frac{1}{3} h (y'_{n+1} + 4y'_n + y'_{n-1}) - \frac{1}{90} h^5 y^{(5)}$
Адамса — Башфорта	$y_{n+1} = y_n + \frac{1}{24} h (55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}) + \frac{251}{720} h^5 y^{(5)}$	$y_{n+1} = y_n + \frac{1}{24} h (9y'_{n+1} - 19y'_n - 5y'_{n-1} + y'_{n-2}) + \frac{19}{720} h^5 y^{(5)}$
Хэмминга	$y_{n+1}^{(0)} = y_{n-3} + \frac{4}{3} h (2y'_n - y'_{n-1} + 2y'_{n-2})$ <p>Уточнение прогноза</p> $\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} + \frac{112}{121} (y_n - y_n^{(0)})$ $(\bar{y}_{n+1}^{(0)})' = f(x_{n+1}, \bar{y}_{n+1}^{(0)})$	$y_{n+1}^{(j+1)} = \frac{1}{8} (9y_n - y_{n-2}) + \frac{3}{8} h ((\bar{y}_{n+1}^{(j)})' + 2y'_n - y'_{n-1})$
	$y_{n+1} = y_{n-3} + \frac{4}{3} h (2y'_n - y'_{n-1} + 2y'_{n-2}) + \frac{28}{90} h^5 y^{(5)}$	$y_{n+1} = \frac{1}{8} [9y_n - y_{n-2} + 3h (y'_{n+1} + 2y'_n - y_{n-1})] - \frac{1}{40} h^5 y^{(5)}$

«Жесткие» задачи. Существуют обыкновенные дифференциальные уравнения, для которых трудно получить удовлетворительное решение задач с использованием описанных методов. Определение таких задач связано с понятием постоянной времени дифференциального уравнения, которое вводится применительно к аналитическому решению. Для уравнений первого порядка — это промежуток времени, когда изменяющаяся часть ре-

шения убывает в ϵ раз. Уравнение порядка n имеет n постоянных времени; если любые две из них сильно (на практике в сто и более раз) отличаются по величине или какая-либо из них достаточно мала по сравнению с интервалом времени, на котором отыскивается решение, то задача называется «жесткой» и ее практически невозможно решить обычными методами. Коэффициенты в таких уравнениях отличаются друг от друга на несколько порядков. Например, жесткими являются системы дифференциальных уравнений, описывающие управляемое движение робота-манипулятора, так как переходные процессы в системе управления приводом затухают быстрее, чем переходные процессы в механической части робота. При решении «жестких» задач обычными методами шаг должен быть достаточно мал, чтобы можно было учитывать приращение наиболее быстро изменяющихся составляющих решения даже после того, как их вклад станет практически незаметным. Но уменьшение шага приводит к увеличению затрат времени ЭВМ, накоплению ошибок. Причем даже на гладком участке решения увеличение шага приводит к нарастанию погрешностей округления и дискретизации [22].

«Жесткие» задачи часто встречаются в теории автоматического управления, например при анализе переходных процессов в системе, содержащей звенья высоких порядков, коэффициенты которых значительно отличаются друг от друга.

Наиболее простым путем решения «жестких» задач является так называемый неявный метод Эйлера, в котором решение находится из следующего уравнения, содержащего y_{n+1} в неявном виде:

$$y_{n+1} = y_n + hf(x_n, y_{n+1}).$$

Выбор метода решения задачи Коши. При сравнении эффективности одношаговых и многошаговых методов можно выделить следующие особенности.

1. Многошаговые методы требуют большого объема памяти ЭВМ, так как оперируют большим количеством исходных данных.

2. При использовании многошаговых методов существует возможность оценки погрешности на шаге. Поэтому величина шага выбирается оптимальной, а в одношаговых — с некоторым запасом, что снижает быстродействие.

3. При сравнимой точности многошаговые методы требуют меньшего объема вычислений. Например, применяя метод Рунге — Кутты четвертого порядка точности, приходится вычислить четыре значения функции на каждом шаге, а для обеспечения сходимости метода прогноза и коррекции того же порядка точности — два.

4. Одношаговые методы, в отличие от многошаговых, позволяют начать решение задачи (самостартование) и легко изменять шаг в процессе решения.

Перед началом решения задачи необходимо произвести проверку на «жесткость» и в случае положительного результата применить специальные методы. Если задача Коши очень сложна, то обычно предпочтение отдается методу прогноза и коррекции, который обладает к тому же более высоким быстродействием. Начало решения задачи при этом производится с помощью одношаговых методов. Если для вычисления очередного значения y_i требуется более двух итераций или если ошибка усечения слишком велика, то необходимо уменьшить величину шага h . С другой стороны, при очень малой погрешности усечения можно увеличить шаг, тем самым повысив быстродействие, но при этом весь процесс решения производится сначала. Иногда на практике требуется минимизировать время подготовки задачи к решению. Тогда целесообразно использовать методы Рунге — Кутты.

Для эффективного решения задачи большое значение имеет опыт, интуиция и квалификация исследователя как при постановке задачи, так и в процессе выбора метода, разработки алгоритма и программы решения задачи на ЭВМ. При этом часто определяющее значение придается имеющимся в наличии программным средствам: стандартным процедурам и пакетам прикладных программ (гл. 7).

4.3. Методы решения краевых задач

Методы решения краевых задач рассматриваются на примере обыкновенного дифференциального уравнения второго порядка

$$\frac{d^2y}{dx^2} = f(x, y, y')$$

при граничных условиях $y(a) = A$, $y(b) = B$.

Методы решения краевых задач можно разделить на три группы: разностные, проекционные и методы, основанные на замене решения краевой задачи решением нескольких задач Коши (методы «стрельбы»).

Метод «стрельбы». Если обыкновенное дифференциальное уравнение второго порядка — линейное, то оно имеет вид

$$y'' = f_1(x)y' + f_2(x)y + f_3(x). \quad (4.4)$$

Краевая задача ставится при следующих граничных условиях:

$$y(a) = A; y(b) = B.$$

Поскольку в общем случае обыкновенное дифференциальное уравнение второго порядка имеет множество решений, удовлетворяющих первому граничному условию $y(a) = A$ и различным значениям в точке $x = a$ первой производной $y'(a)$, то задача сводится к отысканию такого решения, которое удовлетворяло бы и второму граничному условию $y(b) = B$.

Краевую задачу можно свести к задаче Коши введением дополнительного начального условия $y'(a) = \alpha_1$ и дополнительно к $y(a) = A$.

Найдя решение $y'(x)$, можно поставить другое начальное условие

$$y'(a) = \alpha_2$$

и получить второе решение $y_2(x)$. Если $y_1(b) = \beta_1$, а $y_2(b) = \beta_2$, причем $\beta_1 \neq \beta_2$, то решение

$$y(x) = \frac{1}{\beta_1 - \beta_2} [(B - \beta_2) y_1(x) + (\beta_1 - B) y_2(x)]$$

удовлетворяет обоим начальным условиям. Это общее решение, полученное как линейная комбинация $y_1(x)$ и $y_2(x)$, и является решением краевой задачи для линейного обыкновенного дифференциального уравнения (4.4).

При решении нелинейного уравнения краевая задача сводится к решению нескольких задач Коши с последовательным вводом в начальные условия значения α :

$$y(a) = A \text{ и } y'(a) = \alpha,$$

и решением, удовлетворяющим условию $y(b) = B$.

При этом алгоритм достижения цели строится на основе одного из методов оптимизации [23]. Однако этот

метод решения задачи связан с большими вычислениями, поэтому в случае нелинейных дифференциальных уравнений предпочтение отдается разностным и проекционным методам.

Разностные методы. Аппарат разностных методов является мощным средством численного решения обыкновенных дифференциальных уравнений и дифференциальных уравнений в частных производных [18, 19]. В его основе лежит представление независимого аргумента на отрезке $[a, b]$ в виде дискретного множества точек $x_i, i = 0, \dots, n, x_0 = a, x_n = b$, которое называется сеткой.

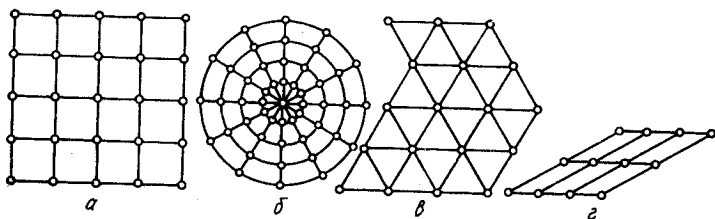


Рис. 4.6.

Наибольшее распространение получила равномерная сетка с шагом $x_i - x_{i-1} = h$. При этом вместо непрерывной функции $f(x)$ рассматривается сеточная функция $y_i = f(x_i)$. Аналогично проводится дискретизация функций многих переменных, например двух $x_{ij}, i = 0, \dots, n; j = 0, \dots, m$;

$$y_{ij} = f(x_{ij}).$$

Кроме наиболее распространенной прямоугольной сетки применяют полярную, треугольную и скошенную (рис. 4.6). Многомерные сетки используют в задаче с частными производными по нескольким независимым переменным.

Решение задачи разностными методами состоит из двух этапов: 1) получения дискретной (разностной) аппроксимации дифференциальных уравнений и исследования получаемых при этом разностных уравнений; 2) решения разностных уравнений.

При получении разностной схемы важную роль играет общее требование, чтобы разностная схема как можно лучше приближала основные свойства исходного дифференциального уравнения. Такие разностные схемы можно

получать при помощи вариационных принципов и интегральных соотношений. Оценка точности разностной схемы сводится к изучению погрешности аппроксимации и устойчивости.

Сеточную функцию можно рассматривать как функцию целочисленного аргумента

$$y(i) = y_i, \quad i = 0, \pm 1, \pm 2, \dots$$

Для y_i можно ввести операции, являющиеся дискретным аналогом операций дифференцирования и интегрирования.

Аналогом первой производной являются разности первого порядка:

$$\Delta y_i = y_{i+1} - y_i \text{ — правая;}$$

$$\nabla y_i = y_i - y_{i-1} \text{ — левая;}$$

$$\delta y_i = \frac{1}{2} (\Delta y_i + \nabla y_i) = \frac{1}{2} (y_{i+1} - y_{i-1}) \text{ — центральная.}$$

Легко заметить, что $\Delta y_i = \nabla y_{i+1}$.

Далее запишем разности второго порядка:

$$\Delta^2 y_i = \Delta (\Delta y_i) = \Delta (y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i;$$

$$\Delta \nabla y_i = \Delta (y_i - y_{i-1}) = (y_{i+1} - y_i) - (y_i - y_{i-1}) = \\ = y_{i+1} - 2y_i + y_{i-1},$$

т. е.

$$\Delta^2 y_i = \Delta \nabla y_{i+1}.$$

Аналогично определим разность m -го порядка

$$\Delta^m y_i = \Delta (\Delta^{m-1} y_i).$$

Очевидно, что

$$\sum_{j=k}^i \Delta y_j = y_{i+1} - y_k; \quad \sum_{j=k}^i \nabla y_j = y_i - y_{k-1}.$$

На множестве узлов сетки, называемом шаблоном, заменим непрерывный дифференциальный оператор Ly разностным $L_h y$. Например, разностные операторы для первой производной на трех узлах сетки ($x-h$, x , $x+h$):

$$L_h^+ y = \frac{y(x+h) - y(x)}{h} = y_x^+ \text{ — правая;}$$

$$L_h^- y = \frac{y(x) - y(x-h)}{h} = y_x^- \text{ — левая;}$$

$$L_h^0 y = \frac{y(x+h) - y(x-h)}{2h} = y_x^0 \text{ — центральная.}$$

Аналогично для второй производной:

$$L_h y = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = \frac{y_x^+(x) - y_x^-(x)}{h} = \\ = \frac{y_x^-(x+h) - y_x^-(x)}{h} = y_{\bar{x}\bar{x}}(x).$$

При решении краевой задачи записываются разностные для всех n узлов области изменения x ($[a, b]$). Учитывая два крайних условия $y(0) = y(a)$ и $y_n = y(b)$, получаем систему из $n - 1$ алгебраических уравнений с $n - 1$ неизвестными y_i . Если исходное обыкновенное дифференциальное уравнение линейное, то задача сводится к решению системы линейных алгебраических уравнений, а если нелинейная, то нелинейных или трансцендентных алгебраических систем (гл. 2, 3). Привести краевую задачу, решаемую методом конечных разностей, к виду, удобному для разработки стандартной программы для ЭВМ трудно, так как формулировка каждой задачи зависит от вида рассматриваемого дифференциального уравнения.

Разностные методы являются универсальными для решения численных задач, хотя и рассматриваются в большинстве случаев в связи с решением краевых задач и дифференциальных уравнений в частных производных. Например, метод Эйлера решения задачи Коши может быть интерпретирован как использование одномерного разностного оператора.

Проекционные методы. Сущность проекционных методов вычислительной математики состоит в представлении решения задачи множеством проекций (отсчетов) в определенной системе координатных функций [15]. Рассмотрим применение проекционных методов на примере решения краевой задачи с нулевыми граничными условиями.

В традиционном методе, предложенном Б. П. Галеркиным и развитым в работах М. В. Келдыша, приближенное решение $y(x)$ отыскивается в виде

$$y_n = c_1 \Phi_1(x) + c_2 \Phi_2(x) + \dots + c_n \Phi_n(x), \quad (4.5)$$

где $\Phi_1(x), \Phi_2(x), \dots, \Phi_n(x)$ — система базисных функций, удовлетворяющих исходным граничным условиям; c_1, c_2, \dots, c_n — неизвестные постоянные коэффициенты, ко-

торые определяются из следующей системы уравнений (моментных уравнений Галеркина):

$$\int_a^b [y_n'' - f_1(x) y_n' - f_2(x) y_n] \varphi_j(x) dx = \\ = \int_a^b f_3(x) \varphi_j(x) dx, \quad j = 1, \dots, n. \quad (4.6)$$

Подставив (4.5) в (4.6), убедимся, что уравнение (4.6) сводится к системе линейных алгебраических уравнений

$$\sum_{i=1}^n l_{ij} = f_j, \quad j = 1, \dots, n, \quad (4.7)$$

где

$$l_{ij} = \int_a^b [\varphi_i''(x) - f_1(x) \varphi_i'(x) - f_2(x) \varphi_i(x)] \varphi_j(x) dx; \\ f_j = \int_a^b f_3(x) \varphi_j(x) dx.$$

Коэффициенты l_{ij} удобно представить в виде

$$l_{ij} = a_{ij} + b_{ij} + d_{ij}, \\ \text{где} \quad a_{ij} = \int_a^b \varphi_i''(x) \varphi_j(x) dx; \\ b_{ij} = - \int_a^b f_1(x) \varphi_i'(x) \varphi_j(x) dx; \\ d_{ij} = - \int_a^b f_2(x) \varphi_i(x) \varphi_j(x) dx.$$

Решение системы линейных уравнений (4.7) относительно неизвестных c_1, c_2, \dots, c_n позволяет определить y_n .

На практике наиболее трудным этапом является выбор системы базисных функций. Значительное упрощение метода Галеркина достигается в случае кусочно-линейных базисных функций, что приводит к методу конечных элементов [6]. При этом для однозначности

решения исходное дифференциальное уравнение удобно рассматривать в виде

$$-\frac{d}{dx} \left(q_1(x) \frac{dy}{dx} \right) + q_2(x) y = q_3(x),$$

где $q_1(x)$, $q_2(x)$, $q_3(x)$ — непрерывные функции на отрезке (a, b) .

Разбиваем отрезок $[a, b]$ на n частей:

$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$ с шагом h .



Рис. 4.7.

Функцию $\varphi_j(x)$ (рис. 4.7) можно аналитически описать как

$$\varphi_j(x) = \begin{cases} 0 & \text{при } x_0 < x < x_{j-1}, \quad x_{j+1} < x < x_n; \\ \frac{1}{h} (x - x_{j-1}) & \text{при } x_{j-1} < x < x_j; \\ \frac{1}{h} (x_{j+1} - x) & \text{при } x_j < x < x_{j+1}. \end{cases} \quad (4.8)$$

В соответствии с (4.5) решение будет являться линейной комбинацией функций $\varphi_j(x)$, т. е. также кусочно-линейной функцией. На участке $[x_{j-1}, x_j]$ имеет место равенство

$$y'(x) = \frac{(y_j - y_{j-1})}{h},$$

где $y_j = y_n(x_j)$. Кроме того, очевидно, $y_0 = 0$, $y_n = 0$, $y_i = c_i$, $i = 1, \dots, n-1$. Составляем моментное уравнение Галеркина

$$\begin{aligned} \int_a^b \left[-\frac{d}{dx} (q_1(x) y_n') + q_2(x) y \right] \varphi_j(x) dx &= \\ &= \int_a^b q_3(x) \varphi_j(x) dx, \quad j = 1, \dots \end{aligned}$$

и, производя интегрирование по частям в первом слагаемом левой части, получаем

$$\begin{aligned} & \int_a^b q_1(x) y_n' \varphi_j(x) dx + \int_a^b q(x) y_n \varphi_j(x) dx = \\ & = \int_a^b q_3(x) \varphi_j(x) dx, \quad j = 1, \dots, n-1. \end{aligned} \quad (4.9)$$

Учитывая (4.8), система (4.9) принимает вид

$$\begin{aligned} & \int_{x_{j-1}}^{x_{j+1}} q_1(x) y_n' \varphi_j(x) dx + \int_{x_{j-1}}^{x_{j+1}} q_2(x) y_n \varphi_j(x) dx = \\ & = \int_{x_{j-1}}^{x_{j+1}} q_3(x) \varphi_j(x) dx, \quad j = 1, \dots, n-1. \end{aligned} \quad (4.10)$$

После вычисления интегралов система (4.10) приводится к трехдиагональной системе линейных алгебраических уравнений

$$(a_{j+1} + l_{j+1}) y_{j+1} + (b_j + f_j) y_j + (d_{j-1} + q_{j-1}) y_{j-1} = F_j,$$

где

$$\left. \begin{aligned} a_{j+1} &= -\frac{1}{h^2} \int_{x_j}^{x_{j+1}} q_1(x) dx; \\ b_j &= \frac{1}{h^2} \int_{x_{j-1}}^{x_{j+1}} q_1(x) dx; \\ d_{j-1} &= -\frac{1}{h^2} \int_{x_{j-1}}^{x_j} q_1(x) dx; \\ l_{j+1} &= \frac{1}{h^2} \int_{x_j}^{x_{j+1}} q_2(x) (x - x_j) (x - x_{j-1}) dx; \\ f_j &= \frac{1}{h^2} \left[\int_{x_{j-1}}^{x_j} q_2(x) (x - x_{j+1})^2 dx + \right. \end{aligned} \right\} \quad (4.11)$$

$$\begin{aligned}
 & + \int_{x_j}^{x_{j+1}} q_2(x) (x_{j+1} - x)^2 dx \Big]; \\
 q_{j-1} & = \int_{x_{j-1}}^{x_j} q_2(x) (x_j - x)(x - x_{j-1}) dx; \\
 F_j & = \int_{x_{j-1}}^{x_{j+1}} q_3(x) \varphi_j(x) dx.
 \end{aligned}$$

Приближенное решение задачи получаем, как и решение системы (4.11), методами, рассмотренными в гл. 2, из которых для трехдиагональных систем наиболее часто используют метод прогонки.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Сформулируйте задачу Коши и краевую задачу. В чем отличие этих задач?

2. Приведите примеры обыкновенных дифференциальных уравнений, решаемых только численными методами.

3. Дайте сравнительную характеристику и определите области применения методов решения обыкновенных дифференциальных уравнений.

4. Возможна ли краевая задача для уравнений первого порядка?

5. Что такое погрешность усечения (ошибка ограничения)? Чем определяется порядок ошибки метода?

6. Покажите, что общая формула для метода Рунге — Кутты, примененная к уравнению $y' = -y$ с начальным условием $y(0) = 1$, приводит к соотношению

$$y_n = \left(1 - h + \frac{h^2}{2}\right)^n \approx e^{-x_n}.$$

Составьте программу для решения на ЭВМ этого уравнения.

7. В чем сходство между методами прогноза и коррекции и исправленным методом Эйлера?

8. Как оцениваются погрешности методов Эйлера, Рунге — Кутты, прогноза и коррекции?

9. Дайте геометрическую интерпретацию методов Эйлера.

10. Что такое свойство «самостартования»? Какие методы решения обыкновенных дифференциальных уравнений им обладают?

11. Как выбирается шаг при использовании многошаговых методов решения?

12. Какие уравнения называют «жесткими»? В чем заключаются особенности их решения?

13. Для решения каких краевых задач может быть использован метод «стрельбы»? Как получается решение краевой зада-

чи из решения нескольких задач Коши в этом методе? Составьте программу для решения на ЭВМ краевой задачи для линейного дифференциального уравнения второго порядка методом «стрельбы».

14. Из каких этапов состоит решение задачи разностными методами?

15. Запишите разностные операторы для первой и второй производных.

16. Как определяются разностные операторы для производных более высоких порядков?

17. Из каких соображений выбирается метод решения обыкновенных дифференциальных уравнений?

18. В чем сущность проекционных методов решения краевых задач?

19. Как составляются моментные уравнения Галеркина?

20. Нарисуйте структурную схему алгоритма решения краевой задачи методом конечных элементов на ЭВМ.

Глава 5. МЕТОДЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

С дифференциальными уравнениями в частных производных связано решение инженерных задач во многих областях науки и техники, например, в аэро- и гидродинамике, ядерной физике, радиосвязи. Математические модели на основе уравнений используются при оценке качества и синтеза систем автоматического управления, анализе объектов контроля и оценке чувствительности в измерительной технике и радиоэлектронике. В них содержатся частные производные, и искомая функция одновременно зависит от нескольких переменных. Рассмотрим дифференциальное уравнение второго порядка с двумя независимыми переменными

$$A(x, y) \frac{\partial^2 f}{\partial x^2} + B(x, y) \frac{\partial^2 f}{\partial x \partial y} + C(x, y) \frac{\partial^2 f}{\partial y^2} + E(x, y) \frac{\partial f}{\partial y} + D(x, y) \frac{\partial f}{\partial x} + F(x, y) = G. \quad (5.1)$$

Аналогично обыкновенным дифференциальным уравнениям единственное решение уравнения (5.1) можно получить в случае, когда заданы дополнительные условия, но, поскольку здесь две независимые переменные x и y , условие должно задаваться для какой-то кривой в плоскости x, y . Это условие может быть наложено на функцию f или (\ln) на ее производную и зависит от типа

уравнения, определяющего вид и характер ее изменения.

Существует три вида дифференциальных уравнений второго порядка [13]: эллиптические — при $B^2 - 4AC < 0$; параболические — при $B^2 - 4AC = 0$; гиперболические при $B^2 - 4AC > 0$.

Уравнения могут переходить из одного вида в другой в зависимости от значений коэффициентов.

Эллиптические уравнения описывают стационарные процессы, причем задача ставится в замкнутой области и в каждой точке границы этой области задаются граничные условия. Другие два вида уравнений описывают эволюционные процессы. В таких задачах наиболее распространен случай, когда на одной части границы ставятся граничные условия, а на другой — начальные.

Примеры дифференциальных уравнений в частных производных, описывающих различные типы задач, приведены в табл. 5.1. В ней использованы принятые обозначения наиболее распространенных операторов:

$$\text{Лапласа} - \Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2};$$

$$\text{бигармонического} - \Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2 \frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4}.$$

Широко используются два метода решения дифференциальных уравнений в частных производных: метод конечных разностей и метод конечных элементов [6, 18, 19, 23]. В современной прикладной математике оба метода рассматриваются как интерпретация применения общей теории разностных схем к решению дифференциальных уравнений в частных производных [18].

5.1. Метод конечных элементов

В основе метода конечных элементов лежит вариационное исчисление. Дифференциальные уравнения и соответствующие граничные условия используются для постановки вариационной задачи. В методе конечных элементов физическая задача заменяется кусочно-гладкой моделью [6].

Применительно к одномерным задачам метод конечных элементов рассмотрен в п. 4.3. Аналогичный подход может быть применен и в случае многомерных областей определения искомой функции. При решении дифференциальных уравнений второго порядка в частных про-

Таблица 5.1. Типы дифференциальных уравнений в частных производных

Уравнение	Математическая форма	Примеры задач
Лапласа	$\Delta f = 0$	Установившееся течение жидкости. Стационарные тепловые поля
Пуассона	$\Delta f = -k$	Теплопередача с внутренними источниками тепла
диффузии	$\Delta f = \frac{1}{h^2} \frac{\partial f}{\partial t}$	Нестационарная теплопроводность
волновое	$\Delta f = \frac{1}{c^2} \frac{\partial^2 f}{\partial t^2}$	Распространение волн (звуковых, электромагнитных)
бигармоническое	$\Delta^2 f = F(x, y)$	Деформация пластин

изводных наиболее часто употребляются элементы в форме треугольников и четырехугольников, имеющие как прямо-, так и криволинейные границы, что позволяет с достаточной степенью точности аппроксимировать границу области любой формы.

В общем случае алгоритм метода состоит из следующих четырех этапов.

1. Разбиение заданной области на конечные элементы. От качества разбиения во многом зависит точность получаемых результатов. Разбиение на элементы, близкие по форме к равносторонним треугольникам, во многих случаях оптимально. Разработан ряд алгоритмов решения этой задачи, которые используются в виде отдельных подпрограмм в программных комплексах для решения дифференциальных уравнений в частных производных. Часто для этого используют способ, основанный на деформации регулярной сетки с тем же числом узлов, что и в заданной области.

2. Определение интерполирующего полинома для каждого элемента. Значение функции в каждой точке i -го элемента определяется как

$$f_i = A_i R + A_0, \quad (5.2)$$

где A_i — вектор-строка коэффициентов полинома; A_0 — свободный член; R — вектор координат в рассматриваемой точке.

Далее на этом этапе определяются A_i и A_0 . Используем известные узловые значения функции, позволяющие с учетом уравнения (5.2) выразить через них искомые коэффициенты. В результате получим

$$f_i = \varphi_i \Phi, \quad (5.3)$$

где Φ — вектор узловых значений функции; φ_i — элемент вектор-строки функций формы конечного элемента, причем функция формы равна 1 в узле с соответствующим номером и 0 во всех других узлах.

3. Объединение конечных элементов в ансамбль. На этом этапе уравнения (5.3), относящиеся к отдельным элементам, объединяются в систему алгебраических уравнений, которая в матричной форме имеет вид

$$f = \varphi \Phi. \quad (5.4)$$

4. Определение вектора Φ узловых значений функции, который вначале неизвестен. Это наиболее сложная процедура в методе конечных элементов. Разработано несколько алгоритмов вычисления Φ . Наибольшее применение получили методы, основанные на составлении моментных уравнений Галеркина (аналогично (4.6)) и на минимизации некоторого функционала, определение которого связано с физическим смыслом решаемой задачи.

В результате получается система алгебраических уравнений

$$K\Phi = B,$$

где K — матрица коэффициентов, которая по традиции носит название матрицы жесткости; B — вектор нагрузки.

Найденные значения вектора Φ подставляют в (5.4) и определяют значение функции f в любой точке заданной области.

На практике реализация каждого из описанных этапов (особенно четвертого) представляет собой трудоемкую задачу, решение которой требует опыта, теоретической подготовки и знания физической сущности задачи.

Метод конечных элементов широко применяется для решения специальных задач в теоретической механике, гидродинамике, теории поля, в последнее время используется как основа для разработки систем автоматизации проектирования.

5.2. Метод конечных разностей

Ранее были рассмотрены основные определения и правила построения разностных систем (п. 4.5). Для дифференциальных уравнений второго порядка в частных производных наиболее часто используется двумерная прямоугольная сетка. Центрально-разностные шаблоны применительно к двумерной квадратной сетке с шагом h (рис. 5.1) могут быть получены аналогично одномерному случаю, описанному в п. 4.5. Для удобства заменим обозначение $f(x_i + h, y_j)$ на $f_{i+1,j}$. Используя эти обозначения, получим выражения для частных производных.

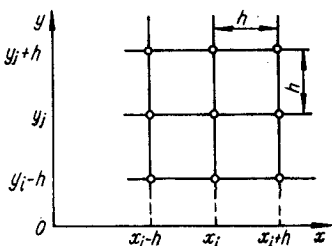


Рис. 5.1.

$$\frac{\partial f}{\partial x} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h}$$

$$\approx \frac{1}{2h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}_{i,j};$$

$$\frac{\partial f}{\partial y} \approx \frac{f_{i,j+1} - f_{i,j-1}}{2h}$$

$$\approx \frac{1}{2h} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}_{i,j};$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2}$$

$$\approx \frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}_{i,j};$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2}$$

$$\approx \frac{1}{h^2} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}_{i,j};$$

$$\frac{\partial^2 f}{\partial x \partial y} \approx \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4h^2} \approx \frac{1}{4h^2}$$

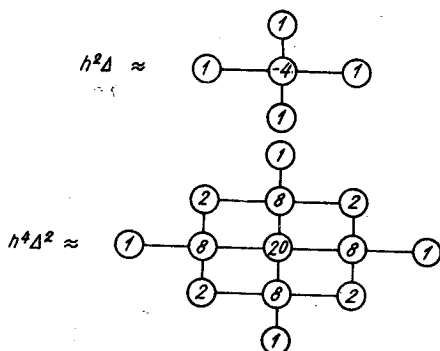
$$\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}_{i,j};$$

$$\frac{\partial^4 f}{\partial x^4} \approx \frac{1}{h^4} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}_{i,j};$$

$$\frac{\partial^4 f}{\partial y^4} \approx \frac{1}{h^4} \begin{bmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{bmatrix}_{i,j}$$

Из этих элементов строятся более сложные вычислительные шаблоны для дифференциальных уравнений.

Сложение производных осуществляется суперпозицией соответствующих вычислительных шаблонов. Этим методом конструируются шаблоны для Δf и $\Delta^2 f$.



Все приведенные вычислительные шаблоны имеют погрешность порядка h^2 . Можно построить более точные вычислительные шаблоны, если включить в рассмотрение

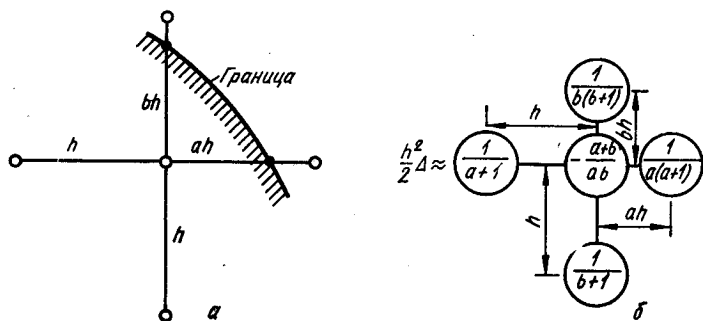


Рис. 5.2.

дополнительные узлы. Иногда, чтобы свести к минимуму распространение ошибок, пользуются левыми или правыми разностями.

Часто трудности, связанные с использованием прямоугольной сетки, возникают из-за границы неправильной конфигурации, не проходящей через узлы сетки. Рассмотрим пример решения такой задачи для вычислительного шаблона уравнения Лапласа в области, ограниченной произвольной кривой (рис. 5.2,а).

Вторые частные производные для узлов, лежащих на границе области, можно записать в виде

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{\frac{f_a - f_{i,j}}{ah} - \frac{f_{i,j} - f_{i-1,j}}{h}}{0,5(ah+h)};$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{\frac{f_b - f_{i,j}}{bh} - \frac{f_{i,j-1}}{h}}{0,5(bh+h)}.$$

Сложив эти две производные, получим (рис. 5.2, б)

$$\Delta f \approx \frac{2}{h^2} \left(\frac{f_{i-1,j}}{1+a} + \frac{f_a}{a(1+a)} + \frac{f_b}{b(1+b)} + \frac{f_{i,j-1}}{1+b} - \frac{a+b}{ab} f_{i,j} \right).$$

Применив вычислительный шаблон к каждому из n узлов сетки, получим систему из n уравнений, которая может быть линейной, если исходное дифференциальное уравнение имеет соответствующую структуру. В этом случае решение задачи сводится к системе уравнений вида

$$\left[\begin{array}{c} \text{матрица} \\ \text{коэффициентов} \end{array} \right] \left[\begin{array}{c} \text{неизвестные} \\ \text{значения в узлах} \\ \text{(вектор-столбец)} \end{array} \right] = \left[\begin{array}{c} \text{вектор-столбец} \\ \text{свободных} \\ \text{членов.} \end{array} \right]$$

5.3. Решение различных видов дифференциальных уравнений в частных производных

Практические методы и алгоритмы решения различных видов дифференциальных уравнений в частных производных имеют определенные особенности и требуют отдельного рассмотрения на примере наиболее распространенных задач.

Решение эллиптических уравнений. К эллиптическим уравнениям приводится множество различных физических задач: расчет напряжений, возникающих при упругом кручении длинного цилиндрического стержня; распределение электрических напряжений на проводящей плоскости; задача о стационарных потоках тепла в двумерном теле и др. Возникает необходимость решения таких задач и в теории автоматического управления, информационно-измерительной технике.

Большинство эллиптических уравнений описывается уравнением Пуассона или его частным случаем — уравнением Лапласа.

Рассмотрим классическую задачу Дирихле для уравнения Лапласа в прямоугольной области. Найти непрерывную функцию $f(x, y)$, удовлетворяющую внутри прямоугольной области $\Omega = \{(x; y) | 0 \leq x \leq a, 0 < y < b\}$ уравнению Лапласа

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

и принимающую на границе области заданные значения:

$$\begin{aligned} x = 0, f(0, y) &= f_1(y); \\ x = a, f(a, y) &= f_2(y); \\ y = 0, f(x, 0) &= f_3(x); \\ y = b, f(x, b) &= f_4(x). \end{aligned}$$

Введем в области решения двумерную сетку с шагом h по оси x и l по оси y . Тогда, используя принятые в предыдущих разделах обозначения и аппроксимируя уравнение Лапласа разностным уравнением, получим следующую систему линейных уравнений (принимая для простоты $l = h$):

$$\left. \begin{aligned} f_{i,j} &= \frac{1}{4} (f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}); \\ f_{i,0} &= f_3(x_i), f_{i,m} = f_4(x_i), f_{0,j} = f_1(y_j), f_{n,j} = f_2(y_j), \\ i &= 1, 2, \dots, n-1; j = 1, \dots, m-1 \end{aligned} \right\} \quad (5.5)$$

Эта система уравнений имеет большое количество нулевых элементов и удовлетворяет условиям сходимости при использовании итерационных методов. Для решения таких систем наиболее широко используют метод Гаусса—Зейделя, который применительно к эллиптическим разностным уравнениям называется методом Либмана, или методом последовательных смещений. Порядок итераций можно проследить, записав систему (5.5) следующим образом:

$$\left. \begin{aligned} f_{1,1}^{(1)} &= \frac{1}{4} [f_3(h) + f_1(h) + f_{2,1}^{(0)} + f_{1,2}^{(0)}]; \\ f_{2,1}^{(1)} &= \frac{1}{4} [f_3(2h) + f_{1,1}^{(1)} + f_{3,1}^{(0)} + f_{2,2}^{(0)}]; \end{aligned} \right\}$$

$$\left. \begin{aligned} f_{3,1}^{(1)} &= \frac{1}{4} [f_3(3h) + f_{2,1}^{(1)} + f_{4,1}^{(0)} + f_{3,1}^{(0)}]; \\ \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ f_{n-1,1}^{(1)} &= \frac{1}{4} \{f_3[(n-1)h] + f_2(h) + f_{n-2,1}^{(1)} + f_{n-1,2}^{(0)}\}; \\ f_{1,2}^{(1)} &= \frac{1}{4} [f_1(2h) + f_{1,1}^{(1)} + f_{2,2}^{(0)} + f_{1,3}^{(0)}]; \\ f_{2,2}^{(1)} &= \frac{1}{4} [f_{2,1}^{(1)} + f_{1,2}^{(1)} + f_{3,2}^{(0)} + f_{2,3}^{(0)}], \\ \dots & \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \end{aligned} \right\} (5.6)$$

где верхними индексами обозначен порядковый номер итерации.

Обычно полагают $f_{i,j}^{(0)} = 0$ для всех i, j . Система уравнений (5.6) легко решается на ЭВМ. На рис. 5.3 приведен пример решения уравнения Лапласа на ЭВМ. Отметим, что любые эллиптические уравнения, не содержащие $\frac{d^2f}{dxdy}$, приводятся к системам разностных уравнений, которые можно решать как методом Либмана, так и другими итерационными методами, поскольку для них выполняются достаточные условия сходимости. Для эллиптических уравнений, содержащих $\frac{d^2f}{dxdy}$, в общем виде вопрос о сходимости итерационных методов не имеет теоретического решения, поэтому необходимо рассматривать получаемую систему уравнений в каждом конкретном случае.

Решение гиперболических уравнений. Наиболее часто встречающимся в инженерной практике видом гиперболического уравнения в частных производных является волновое уравнение, описывающее различные виды колебаний: колебания струны или мембраны, распространение звуковых волн в различных средах и т. д.

В общем виде задача формулируется следующим образом: найти функцию $f(x, t)$, удовлетворяющую внутри области $\Omega = \{(x, t) \mid 0 \leq x \leq a, 0 \leq t \leq T\}$ уравнению

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \quad (c = \text{const} > 0), \tag{5.7}$$

начальным

$$f(x, 0) = f_0(x); \quad \left. \frac{\partial f}{\partial t} \right|_{x,0} = g(x)$$

СПРОГРАММА НАХОЖДЕНИЯ
 ССТАЦИОНАРНОГО РАСПРЕДЕЛЕНИЯ
 СТЕМПЕРАТУРЫ В КВАДРАТНОЙ
 СПЛАСТИНЕ.

С
 СНАХОЖДЕНИЕ ПАРАМЕТРА
 СРЕЛАКСАЦИИ:
 REAL*4 W
 DIMENSION T(3,3), T1(3,3), Z(3,3)
 W=(1+SQRT((-1)**2+4*0.6))/2
 WRITE(5,1)W
 1FORMAT(10X, 'ПАРАМЕТР',
 & 'РЕЛАКСАЦИИ W=', F5.2)
 СПРИБЛИЖЕННЫЕ ЗНАЧЕНИЯ:

С
 T1(1,1)=95.703
 T1(2,1)=203.125
 T1(3,1)=333.985
 T1(1,2)=66.407
 T1(2,2)=156.25
 T1(3,2)=292.969
 T1(1,3)=37.11
 T1(2,3)=109.375
 T1(3,3)=251.954

С
 ССИСТЕМА УРАВНЕНИЙ:
 С

6Z(1,1)=0.25*(T1(2,1)+
 & 125+T1(1,2))
 Z(2,1)=0.25*(Z(1,1)+T1(3,1)+
 & 250+T1(2,2))
 Z(3,1)=0.25*(Z(2,1)+875+
 & T1(3,2))
 Z(1,2)=0.25*(Z(1,1)+T1(1,3)+
 & T1(2,2))
 Z(2,2)=0.25*(Z(1,2)+T1(3,2)+
 & Z(2,1)+T1(2,3))
 Z(3,2)=0.25*(Z(2,2)+500+
 & Z(3,1)+T1(3,3))
 Z(1,3)=0.25*(T1(2,3)+Z(1,2)+
 & 7.813)
 Z(2,3)=0.25*(Z(1,3)+T1(3,3)+
 & Z(2,2)+62.5)
 Z(3,3)=0.25*(Z(2,3)+Z(3,2)+
 & 710.938)

С
 СУТОЧНЕНИЕ МЕТОДОМ РЕЛАКСАЦИИ:
 С

DO 2 J=1,3
 DO 3 I=1,3
 T(I,J)=T1(I,J)+W*
 &(I,J)-T1(I,J))
 3 CONTINUE
 2CONTINUE

N=0

```

DO 4 J=1,3
  DO 5 I=1,3
    IF (ABS(T(I,J)-T1(I,J)),
      & LT.0.001)N=N+1
    T1(I,J)=T(I,J)
  5 CONTINUE
  4CONTINUE
IF(N.LT.9)GO TO 6
WRITE(5,9)T(1,1),T(2,1),
  &T(3,1),T(1,2),T(2,2),T(3,2),
  *T(1,3),T(2,3),T(3,3)
  9 FORMAT(10X,'T(1,1)=' ,F8.3/
  * 10X,'T(2,1)=' ,F8.3/
  * 10X,'T(3,1)=' ,F8.3/
  * 10X,'T(1,2)=' ,F8.3/
  * 10X,'T(2,2)=' ,F8.3/
  * 10X,'T(3,2)=' ,F8.3/
  * 10X,'T(1,3)=' ,F8.3/
  * 10X,'T(2,3)=' ,F8.3/
  * 10X,'T(3,3)=' ,F8.3)
STOP
END

```

```

ПАРАМЕТР РЕЛАКСАЦИИW=1.42
T(1,1)= 114.119
T(2,1)= 234.096
T(3,1)= 363.281
T(1,2)= 97.377
T(2,2)= 208.984
T(3,2)= 344.029
T(1,3)= 66.406
T(2,3)= 160.435
T(3,3)= 303.851

```

Рис. 5.3.

и краевым

$$f(0, t) = \mu_1(t); \quad f(a, t) = \mu_2(t)$$

условиям.

Поскольку замена переменных $t \rightarrow ct$ приводит уравнение (5.7) к виду

$$\frac{\partial^2 f}{\partial t^2} = \frac{\partial^2 f}{\partial x^2},$$

то в дальнейшем принимаем $c=1$.

Переходя к разностному уравнению на сетке с шагом h по x и τ по t с центральными разностями, получим

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\tau^2}.$$

Если ввести $r = \frac{\tau}{h}$, то выражение для $f_{i, j+1}$ примет вид

$$f_{i, j+1} = r^2 (f_{i+1, j} + f_{i-1, j}) + 2(1 - r^2) f_{i, j} - f_{i, j-1}. \quad (5.8)$$

Схема решения по уравнению (5.8) называется трехслойной, так как связывает значение $f_{i, j}$ на трех временных слоях $j - 1, j, j + 1$. Эта схема явная и позволяет в явном виде выразить $f_{i, j}$ через значения f из предыдущих слоев (существуют неявные схемы, основанные на использовании других вычислительных шаблонов, однако они требуют большего объема вычислений при решении системы уравнений). Для решения уравнений на первом слое обычно используют интерполяционные методы, например

$$f_{i, 1} = f_{i, 0} + \tau q(x_i).$$

Соотношения сторон сетки определяются величиной r , которая является мерой устойчивости получаемого решения. При $r > 1$ решение неустойчиво, при $r < 1$ решение хотя и устойчиво, но точность его при уменьшении r убывает; при $r = 1$ разностное решение устойчиво и совпадает с точным. Выбор $r = 1$ удобен еще и тем, что при этом упрощается соотношение (5.8)

$$f_{i, j+1} = f_{i+1, j} + f_{i-1, j} - f_{i, j-1}.$$

Решение параболических уравнений. Примером задачи, приводящей к параболическому уравнению в частных производных, является задача о теплопередаче по длинному стержню, описываемая уравнением теплопередачи (или диффузии).

Задача состоит в нахождении функции $f(x, t)$, удовлетворяющей в области $\Omega = \{(x, t) \mid 0 < x < a, 0 < t < T\}$ уравнению

$$\frac{\partial f}{\partial t} = k \frac{d^2 f}{dx^2}, \quad k = \text{const} > 0; \quad (5.9)$$

начальному условию

$$f(x, 0) = f_0(x)$$

и краевым условиям первого рода

$$f(0, t) = \mu_1(t); \quad f(a, t) = \mu_2(t).$$

Замена переменных $t \rightarrow kt$ приводит уравнение (5.9) к виду

$$\frac{\partial f}{\partial t} = \frac{\partial f^2}{\partial x^2},$$

поэтому в дальнейшем будем считать $k = 1$.

Возможны два варианта получения разностного уравнения на сетке с шагом h по x и τ по t (рис. 5.4, 5.5).

Вариант с аппроксимацией на четырехточечном шаблоне (рис. 5.4) приводит к неявной двухслойной разностной схеме

$$r f_{i+1, j} - (1 + 2r) f_{i, j} + 2 f_{i-1, j} = -f_{i, j-1},$$

где $r = \frac{\tau}{h^2}$.

Эта схема, дополненная уравнениями, получаемыми из краевых условий

$$f_{0, j} = \mu_1(t_j); \quad f_{n, j} = \mu_2(t_j),$$

приводит к системе уравнений, имеющей устойчивое решение при любых значениях r .

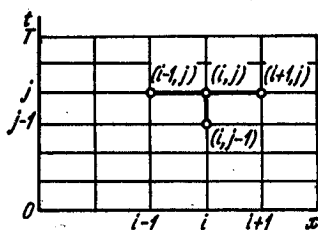


Рис. 5.4.

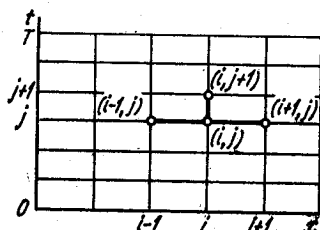


Рис. 5.5.

Вариант с аппроксимацией на четырехточечном шаблоне рис. 5.5 приводит к явной двухслойной схеме

$$f_{i, j+1} = r f_{i+1, j} + (1 - 2r) f_{i, j} + 2 f_{i-1, j}.$$

Эта схема устойчива только при $r \leq 0,5$, что приводит к необходимости проводить вычисления с очень малым шагом по t ($\tau \leq \frac{h^2}{2}$), ограничивающим быстродействие и требующим больших затрат времени ЭВМ. Поэтому для параболических уравнений более широкое распространение получила неявная схема.

5.4. Общие рекомендации по решению дифференциальных уравнений в частных производных

При сравнении методов решения дифференциальных уравнений в частных производных необходимо помнить, что в методе конечных элементов аппроксимируется решение задачи, тогда как в методе конечных разностей — производные искомых функций. Метод конечных элементов, в отличие от разностного метода, требует трудоемкой постановки задачи, высокой квалификации и опыта исследователя, однако удобен при решении задачи со сложной формой границы, неоднородным распределением параметров.

На начальном этапе решения дифференциальных уравнений в частных производных выбирается метод решения задачи. Обычно проще использовать метод конечных разностей, требующий более простой подготовки задачи к решению, однако в ряде случаев для задач с хорошо разработанной теорией (например, задач механики) целесообразно обращаться к методу конечных элементов.

При определении шага решения задачи основным фактором является точность: если высокая, то необходимы либо очень мелкая сетка, либо разбиение на очень мелкие элементы. При этом необходимо учитывать, что погрешность конечно-разностных методов имеет первый порядок, т. е. соизмерима с h^2 .

В случае симметрии в области решения можно число узлов уменьшить в два или в четыре раза при симметрии по обеим осям координат. Это позволяет экономить время и объем памяти ЭВМ.

Большое значение для эффективного решения задачи имеет выбор начальных значений переменных, от этого при использовании итерационных методов в значительной степени зависит скорость сходимости. Часто целесообразно решать задачу в несколько этапов: на первом с помощью грубой сетки (или разбиения на крупные элементы) получают хорошее исходное приближение, а затем ищут точное решение на мелкой сетке.

Для решения дифференциальных уравнений в частных производных разработан ряд программных комплексов, используемых в задачах автоматизации проектирования. Наиболее известными из них являются

ГАММА, ТЕКОН — для решения задач аэро- и гидродинамики, расчета теплообменных процессов методом конечных разностей и **ПРОЧНОСТЬ, ASKA, NASTRAN** — для расчета на прочность деталей и узлов строительных конструкций, авиационных и космических аппаратов и др. на основе метода конечных элементов.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры инженерных задач, описываемых дифференциальными уравнениями в частных производных.

2. Какие существуют виды дифференциальных уравнений второго порядка в частных производных в зависимости от входящих в них коэффициентов?

3. Перечислите методы решения дифференциальных уравнений в частных производных? Дайте их сравнительную характеристику.

4. Как конструируются вычислительные шаблоны для частных производных? Составьте вычислительные шаблоны для оператора Лапласа и бигармонического оператора.

5. Как применяется метод конечных разностей в случае, когда граница области, в которой отыскивается решение, не проходит через узлы сетки? Приведите пример вычислительного шаблона.

6. Какими разностными методами решается классическая задача Дирихле для уравнения Лапласа в прямоугольной области?

7. Составьте трехслойную схему решения гиперболических уравнений. В каких случаях решение устойчиво?

8. Сравните эффективность использования явной и неявной схем решения параболических уравнений в частных производных.

9. Как выбирается шаг при решении дифференциальных уравнений в частных производных?

10. Приведите общий алгоритм решения дифференциальных уравнений в частных производных методом конечных элементов.

11. Составьте вычислительный шаблон оператора Лапласа для трех координат.

Глава 6. МЕТОДЫ ОБРАБОТКИ ДАННЫХ

Рассмотрим вопросы обработки данных, полученных в результате научных или производственных экспериментов, исследования различных процессов, выявления закономерностей и изучения поведения объектов и т. д. Такие методы широко используют в численном анализе систем автоматического управления, при эксперимен-

тальном исследовании объектов контроля и измерения в информационно-измерительной технике, построении аналитических моделей кибернетических систем.

6.1. Методы интерполяции

Задача интерполяции заключается в построении функции $F(x)$, принимающей в некоторых точках x_i ($[a, b]$) ($i = 0, 1, \dots, n$), называемых узлами интерполяции, значения

$$F(x_0) = y_0, F(x_1) = y_1, \dots, F(x_i) = y_i, \dots, F(x_n) = y_n, \quad (6.1)$$

совпадающие с заранее заданными значениями в этих же точках неизвестной функции $y = f(x)$. Геометрически это означает, что нужно найти кривую $y = F(x)$ некоторого определенного типа, проходящую через систему точек $M(x_i, y_i)$ ($i = 0, 1, \dots, n$).

В общем случае эта задача имеет бесконечное множество решений или совсем не имеет решения, однако она становится однозначной, если вместо произвольной функции $F(x)$ искать полином $P_n(x)$ степени не выше n , удовлетворяющий условиям (6.1), т. е.

$$P_n(x_0) = y_0, P_n(x_1) = y_1, \dots, P_n(x_i) = y_i, \dots, P_n(x_n) = y_n.$$

Полученную интерполяционную формулу

$$Y = F(x)$$

обычно используют для приближенного вычисления значений данной функции $f(x)$ для $x \neq x_i$ ($i = 0, 1, \dots, n$). Такая операция называется интерполяцией, или интерполированием. Причем различают интерполяцию в узком смысле, когда $x \in [x_0, x_n]$, и экстраполяцию, когда $x \notin [x_0, x_n]$. Наиболее известными методами интерполяции являются: конечно-разностные (по формулам Ньютона, Бесселя, Гаусса и др.) Лагранжа, итерационные, сплайн-интерполяции [7, 12, 23].

Конечно-разностные методы. Существует много конечно-разностных методов интерполяции. Наибольшее распространение получил метод Ньютона — Грегори для интерполирования «вперед». Интерполяционный многочлен в этом случае имеет вид

$$P_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots \\ \dots + c_n(x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

Коэффициенты c_i находят из уравнения

$$P_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n,$$

составляющих систему

$$\left. \begin{aligned} c_0 &= y_0; \\ c_0 + c_1(x_1 - x_0) &= y_1; \\ c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) &= y_2; \\ &\dots \\ c_0 + c_1(x_n - x_0) + \dots + c_n(x_n - x_0)(x_n - \\ &- x_1) \dots (x_n - x_{n-1}) = y_n. \end{aligned} \right\} (6.2)$$

Это линейная система уравнений с треугольной матрицей.

Если принять шаг $x_{i+1} - x_i = h$, то в области изменения x ($[x_0, x_n]$) получим одномерную равномерную сетку и можем воспользоваться разностным представлением системы (6.2), которое приводится к следующим разностным выражениям для коэффициентов:

$$\begin{aligned} c_0 &= y_0; \\ c_1 &= \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}; \\ c_2 &= \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2}; \\ &\dots \\ c_j &= \frac{\Delta^j y_0}{(j!) h^j}. \end{aligned}$$

Тогда

$$\begin{aligned} p_n(x) &= y_0 + \frac{\Delta y_0}{1! h} (x - x_0) + \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_1) + \dots \\ &\dots + \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned} \quad (6.3)$$

Правила получения и преобразования разностных выражений рассмотрены в п. 4.3. Для определения разностей высших порядков используется выражение

$$\Delta^j y_i = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i, \quad i = 0, 1, 2, \dots, n - j.$$

При $n = 1$ из (6.3) получим формулу линейной интерполяции

$$P_1(x) = y_0 + \frac{\Delta y_0}{h} (x - x_0),$$

а при $n = 2$ — формулу параболической или квадратической интерполяции

$$P_2(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2h^2}(x - x_0)(x - x_1).$$

Если дано неограниченное количество значений функции y , то число n может быть любым. Практически его выбирают так, чтобы разность $\Delta^n y_i$ была постоянной с заданной точностью. За начальное значение x_0 можно принимать любое табличное значение аргумента x . Когда количество значений функции конечно, то число n ограничено и не может быть больше числа значений функции y , уменьшенного на единицу.

Формула (6.3) называется первой интерполяционной формулой Ньютона. Это выражение неудобно для интерполяции вблизи последних значений y . В этом случае обычно применяется вторая интерполяционная формула Ньютона, которая получается при использовании левых разностей от последнего значения (x_n, y_n) (интерполирование «назад»). Тогда интерполяционный полином имеет вид

$$P_n(x) = c_0 + c_1(x - x_n) + c_2(x - x_n)(x - x_{n-1}) + \\ + c_3(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \dots + \\ + c_n(x - x_n)(x - x_{n-1}) \dots (x - x_1),$$

где

$$c_0 = y_n; \\ c_1 = \frac{\Delta y_{n-1}}{h}; \\ c_2 = \frac{\Delta^2 y_{n-2}}{2! h^2}; \\ \dots \dots \dots \\ c_j = \frac{\Delta^j y_{n-j}}{j! h^j}.$$

Окончательное выражение для второй интерполяционной формулы Ньютона

$$P_n(x) = y_n + \frac{\Delta y_{n-1}}{1! h}(x - x_n) + \frac{\Delta^2 y_{n-2}}{2! h^2}(x - x_n)(x - x_{n-1}) + \\ + \dots + \frac{\Delta^n y_0}{n! h^n}(x - x_n)(x - x_{n-1}) \dots (x - x_1).$$

Интерполяционные формулы Ньютона могут быть использованы для экстраполяции функции. Если $x < x_0$, то удобно применять первую интерполяционную формулу Ньютона, причем

$$\frac{x - x_0}{h} < 0.$$

Если же $x > x_n$, то используется вторая интерполяционная формула Ньютона, где

$$\frac{x - x_n}{h} > 0.$$

Таким образом, первая интерполяционная формула Ньютона обычно используется для интерполирования «вперед» и экстраполирования «назад», а вторая — для интерполирования «назад» и экстраполирования «вперед».

В формулах Ньютона используются левые и правые разности. Использование центральных разностей для получения интерполяционных формул приводит к формулам Гаусса, Стирлинга и Бесселя.

Эти формулы удобнее рассматривать на $(2n + 1)$ -м равноотстоящих узлах интерполирования

$$x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_{n-1}, x_n.$$

При этом

$$\Delta x_i = x_{i+1} - x_i = h = \text{const}, i = -n, -(n-1), \dots, n-1,$$

а для функции $y = f(x)$ известны ее значения в этих узлах $y_i = f(x_i)$.

Пусть требуется построить полином $P(x)$ степени не выше $2n$ такой, что

$$P(x_i) = y_i. \quad (6.4)$$

Полином $P(x)$ отыскивается в виде

$$\begin{aligned} P(x) = & c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \\ & + c_3(x - x_{-1})(x - x_0)(x - x_1) + \dots + c_{2n-1}(x - \\ & - x_{-(n-1)}) \dots (x - x_{-1})(x - x_0)(x - x_1) \dots \\ & \dots (x - x_{n-1})(x - x_n). \end{aligned} \quad (6.5)$$

Аналогично интерполяционным формулам Ньютона находим, используя (6.4) и (6.5), что

$$\begin{aligned} c_0 &= y_0; \\ c_1 &= \frac{\Delta y_0}{1! h}; \end{aligned}$$

$$c_2 = \frac{\Delta^2 y_{-1}}{2! h^2};$$

$$\dots$$

$$c_{2n-1} = \frac{\Delta^{2n-1} y_{-(n-1)}}{(2n-1)! h^{2n-1}};$$

$$c_{2n} = \frac{\Delta^{2n} y_{-n}}{(2n)! h^{2n}}.$$

Подставляя полученные значения коэффициентов в (6.5), приходим к первой интерполяционной формуле Гаусса, содержащей разности

$$\Delta y_0, \Delta^2 y_{-1}, \Delta^3 y_{-1}, \Delta^4 y_{-2}, \Delta^5 y_{-2}, \Delta^6 y_{-2}, \dots$$

Аналогично можно получить вторую интерполяционную формулу Гаусса, содержащую центральные разности

$$\Delta y_{-1}, \Delta^2 y_{-1}, \Delta^3 y_{-2}, \Delta^4 y_{-2}, \Delta^5 y_{-3}, \Delta^6 y_{-3}, \dots$$

Взяв среднее арифметическое первой и второй интерполяционной формул Гаусса, получим формулу Стирлинга. Из этих же формул выводится интерполяционная формула Бесселя. Использование интерполяционных формул с центральными разностями целесообразно в середине интервала, тогда как по его краям обычно применяют формулы Ньютона (табл. 6.1).

Погрешности интерполяции для формул Ньютона можно оценить соответственно для первой и второй формул, как

$$\Delta_n(x) \approx \frac{q(q-1) \dots (q-n)}{(n+1)!} \Delta^{n+1} y_0;$$

$$\Delta_n(x) \approx \frac{q(q+1) \dots (q+n)}{(n+1)!} \Delta^{n+1} y_n,$$

где

$$q = \frac{x - x_n}{h}.$$

Для формулы Стирлинга

$$\Delta_n(x) \approx \frac{\Delta^{2n+1} y_{-(n-1)} + \Delta^{2n+1} y_{-n}}{2(2n+1)!} q(q^2-1)(q^2-2) \dots$$

$$\dots (q^2-n^2).$$

Таблица 6.1. Применение разностных формул интерполяции.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	Примечания
						2-я формула Ньютона
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$		$\Delta^4 y_{-4}$	
		Δy_{-2}		$\Delta^3 y_{-3}$		
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$	
		Δy_{-1}		$\Delta^3 y_{-2}$		
x_0	y_0		$\Delta^2 y_{-1}$		$\Delta^4 y_{-2}$	Формула Стирлинга
		Δy_0		$\Delta^3 y_{-1}$		Формула Бесселя
x_1	y_1		$\Delta^2 y_0$		$\Delta^4 y_{-1}$	
		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$		1-я формула Ньютона
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	

Для случая неравноотстоящих значений аргумента можно получить интерполяционные формулы, используя понятие разделенных разностей.

Например, отношение

$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

называется разделенной разностью первого порядка, а отношение

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i}$$

— разделенной разностью второго порядка, и, следовательно, разделенные разности порядка n получаются из рекуррентного соотношения

$$[x_i, x_{i+2}, \dots, x_{i+n}] = \frac{[x_{i+1}, \dots, x_{i+n}] - [x_i, \dots, x_{i+n-1}]}{x_{i+n} - x_i}.$$

Можно получить интерполяционную формулу Гьютона для неравноотстоящих значений аргумента

$$P(x) = y_0 + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + [x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

Интерполяция по Лагранжу. Интерполяция по Лагранжу применяется в общем случае для произвольно расположенных узлов.

Интерполяционный многочлен для метода Лагранжа представляется в виде

$$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \dots + y_n b_n(x),$$

где все $b_j(x)$ — многочлены степени n , коэффициенты которых можно найти с помощью $(n + 1)$ -го уравнения

$$P_n(x_i) = y_i.$$

В результате получаем систему уравнений

$$y_0 b_0(x_0) + y_1 b_1(x_0) + \dots + y_n b_n(x_0) = y_0;$$

.....

$$y_0 b_0(x_n) + y_1 b_1(x_n) + \dots + y_n b_n(x_n) = y_n.$$

Если значения $b_j(x_i)$ выбрать так, что

$$b_j(x_i) = \begin{cases} 1 & \text{при } i = j; \\ 0 & \text{при } i \neq j, \end{cases}$$

то записанная система уравнений будет удовлетворена.

Это условие означает, что любой многочлен $b_j(x_i)$ равен нулю при каждом x_i , кроме x_j . Следовательно, в общем случае многочлен $b_j(x)$ имеет вид

$$b_j(x) = c_j (x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots \dots (x - x_n).$$

Поскольку $b_j(x_j) = 1$, то коэффициенты c_j определяются из выражения

$$c_j = 1 / (x_j - x_0) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n).$$

Наконец, для искомого многочлена получаем

$$P_n(x) = \sum_{j=0}^n y_j \frac{(x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}.$$

Вводя обозначения

$$L_j(x) = (x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n),$$

можно записать

$$P_n(y) = \sum_{j=0}^n y_j \frac{L_j(x)}{L_j(x_j)}.$$

Отметим два свойства полиномов Лагранжа:

$$1) \sum_{j=0}^n \frac{L_j(x)}{L_j(x_j)} = 1;$$

2) поскольку $P_n(x)$ линейно зависит от y_j , то справедлив принцип суперпозиции: интерполяционный многочлен суммы нескольких функций равен сумме интерполяционных многочленов слагаемых.

Погрешность при интерполяции по Лагранжу может быть оценена как

$$|\Delta_n^{(x)}| \leq \frac{M_{n+1}}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n),$$

где

$$M_{n+1} = \max_{x_0 < x < x_n} |f^{(n+1)}(x)|.$$

Итерационные методы интерполяции. Итерационные методы интерполяции основаны на многократном применении простой интерполяционной процедуры.

Наиболее известным из них является метод Эйткена, заключающийся в многократном применении линейной интерполяции.

Линейная интерполяция между точками (x_0, y_0) и (x_i, y_i) осуществляется по формуле

$$y_{i1}(x) = \frac{1}{x_i - x_0} [y_0(x_i - x) - y_i(x_0 - x)],$$

с помощью которой, задав значение x_i , можно составить таблицу функции $y_{i1}(x)$, где $i = 1, 2, \dots, n$. Пользуясь этими функциями, с помощью повторной линейной интерполяции получим новое семейство соотношений

$$y_{i2}(x) = \frac{1}{x_i - x_1} [y_{i1}(x)(x_i - x) - y_1(x)(x_1 - x)].$$

Простой подстановкой можно показать, что выражения для $y_{i2}(x)$ представляют собой многочлены второй степени, описывающие кривые, проходящие через точки x_0, y_0, x_1, y_1 и (x_i, y_i) . Получив многочлены $y_{i2}(x)$,

можно записать выражение для многочлена третьей степени

$$y_{i3}(x) = \frac{1}{x_i - x_2} [y_{i2}(x)(x_i - x) - y_{i2}(x)(x_2 - x)],$$

описывающего кривые, проходящие через точки (x_0, y_0) , (x_1, y_1) , (x_2, y_2) и (x_i, y_i) . Продолжая этот процесс, будем получать значения $y_{i1}(x)$, которые будут стремиться к значению $f(x)$. Несмотря на то что этот метод позволяет вводить многочлены любой степени, обычно

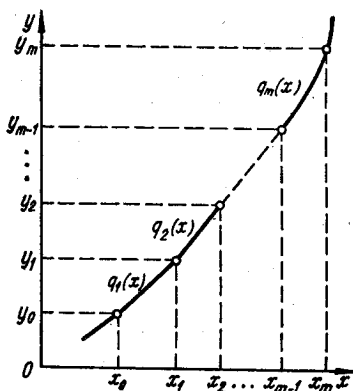


Рис. 6.1.

этого не делают во избежание увеличения числа ошибок.

Сплайн-интерполяция.

Сплайны стали широко использоваться в вычислительной математике сравнительно недавно. В машиностроительном черчении они применяются уже давно, так как сплайны — это лекала или гибкие линейки, деформация которых позволяет провести кривую через заданные точки (x_i, y_i) .

Используя теорию изгиба бруса при малых деформациях, можно показать, что сплайн — это группа кубических многочленов, в местах сопряжения которых первая и вторая производная непрерывны. Такие функции называются кубическими сплайнами. Для их построения необходимо задать коэффициенты, которые единственным образом определяют многочлен в промежутке между данными точками.

Например, для некоторых функций (рис. 6.1) необходимо задать все кубические функции $q_1(x)$, $q_2(x)$, ..., $q_m(x)$. В наиболее общем случае эти многочлены имеют вид

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3, \quad i = 1, 2, \dots, m,$$

где k_{ji} — постоянные, определяемые описанными ранее условиями.

Первые $2m$ условий требуют, чтобы сплайны соприкасались в заданных точках:

$$\begin{aligned} q_i(x_i) &= y_i, \quad i = 1, \dots, m; \\ q_{i+1}(x_i) &= y_i, \quad i = 0, \dots, m-1. \end{aligned}$$

Следующие $(2m - 2)$ условий требуют, чтобы в местах соприкосновения сплайнов были равны первые и вторые производные:

$$\begin{aligned} q'_{i+1}(x_i) &= q'_i(x_i), \quad i = 1, \dots, m-1; \\ q''_{i+1}(x_i) &= q''_i(x_i), \quad i = 1, \dots, m-1. \end{aligned}$$

Система алгебраических уравнений имеет решение, если число уравнений соответствует числу неизвестных. Для этого необходимы еще два уравнения. Обычно используются следующие дополнительные условия:

$$q'_1(x_0) = 0; \quad q''_m(x_m) = 0.$$

Полученный таким способом сплайн называется естественным кубическим сплайном. Найдя коэффициенты сплайна, используют эту кусочно-гладкую полиномиальную функцию для представления данных при интерполяции.

Если специально выбрать вид кубических многочленов, можно значительно упростить задачу (уменьшить число уравнений). В случае, когда отдельные кубические уравнения имеют вид

$$q_i(x) = ty_i + \bar{t}y_{i-1} + \Delta x_i [(k_{i-1} - d) \bar{t}t^2 - (k_i - d) t^2 \bar{t}],$$

$$i = 1, \dots, m,$$

где

$$\Delta x_i = x_i - x_{i-1}; \quad t = \frac{x - x_{i-1}}{\Delta x_i}, \quad \bar{t} = 1 - t,$$

а

$$\Delta y_i = y_i - y_{i-1}; \quad \frac{\Delta y_i}{\Delta x_i} = d_i,$$

каждое из уравнений $q_i(x)$ содержит только два постоянных неизвестных коэффициента. После того как первое уравнение $q_i(x)$ записано, с каждым следующим уравнением добавляется только один новый неизвестный коэффициент. Тогда при $x = x_{i-1}$ $t = 0$, $\bar{t} = 1$, а при $x = x_i$ $\bar{t} = 1$, $t = 0$. Следовательно, все условия, кроме условий для вторых производных, удовлетворяются. Вторые

производные для внутренних точек выражаются соотношениями

$$k_{i-1}\Delta x_{i+1} + 2k_i(\Delta x_i + \Delta x_{i+1}) + k_{i+1}\Delta x_i = 3(d_i\Delta x_{i+1} + d_{i+1}\Delta x_i),$$

а для двух внешних —

$$2k_0 + k_1 = 3d_1 \text{ и } k_{m-1} + 2k_m = 3d_m.$$

Таким образом, решаемая система уравнений является линейной, а ее матрица — трехдиагональной. Методы решения таких систем рассмотрены в гл. 3.

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ \Delta x_2 & 2(\Delta x_1 + \Delta x_2) & \Delta x_1 & 0 & 0 \\ 0 & \Delta x_3 & 2(\Delta x_2 + \Delta x_3) & \Delta x_2 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \Delta x_m & 2(\Delta x_{m-1} + \Delta x_m) & \Delta x_{m-1} \\ 0 & \dots & \dots & 1 & 2 \end{bmatrix} \times \begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_m \end{bmatrix} = 3 \begin{bmatrix} d_1 \\ d_1\Delta x_2 + d_2\Delta x_1 \\ d_2\Delta x_3 + d_3\Delta x_2 \\ \dots \\ d_{m-1}\Delta x_m + d_m\Delta x_{m-1} \\ d_m \end{bmatrix}.$$

Во многих случаях метод сплайнов является наиболее удобным, так как позволяет получить аналитическую кусочную полиномиальную функцию. Существуют сплайны и более высоких порядков. Применение этого метода возможно и в других областях вычислительной математики, например, в численном интегрировании и решении дифференциальных уравнений.

6.2. Аппроксимация данных

Существуют два основных подхода к аппроксимации табличных данных. При одном из них требуют, чтобы аппроксимирующая кривая (возможно, кусочно-гладкая) проходила через все точки, заданные таблицей. Это удастся сделать с помощью описанных ранее методов интерполяции. При другом подходе данные аппроксимируют простой функцией, применимой во всем диапазоне табличных данных, но не обязательно проходящей через

все точки. Такой подход называется подгонкой кривой, которую стремятся провести так, чтобы ее отклонения от табличных данных были минимальны. Обычно пользуются методом наименьших квадратов — сводят к минимуму сумму квадратов разностей между значениями функции, определяемыми выбранной кривой и таблицей [1, 12].

Пусть в таблице задана $(n + 1)$ точка $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ и требуется найти аппроксимирующую кривую $g(x)$ в диапазоне $x_0 \leq x \leq x_n$. В этом случае погрешность в каждой табличной точке

$$\varepsilon_i = g(x_i).$$

Тогда сумма квадратов погрешностей определяется выражением

$$E = \sum_{i=0}^n [g(x_i) - y_i]^2.$$

Обычно функцию $g(x)$ выбирают в виде линейной комбинации подходящих функций

$$g(x) = c_1 g_1(x) + c_2 g_2(x) + \dots + c_k g_k(x).$$

Условие минимума E определяется следующим образом:

$$\frac{\partial E}{\partial c_1} = \frac{\partial E}{\partial c_2} = \dots = \frac{\partial E}{\partial c_k} = 0.$$

Поскольку

$$E = \sum_{i=0}^n [c_1 g_1(x_i) + c_2 g_2(x_i) + \dots + c_k g_k(x_i) - y_i]^2,$$

это условие эквивалентно системе уравнений

$$\frac{\partial E}{\partial c_1} = 2 \sum [c_1 g_1(x_i) + \dots + c_k g_k(x_i) - y_i] g_1(x_i) = 0;$$

$$\frac{\partial E}{\partial c_2} = 2 \sum [c_1 g_1(x_i) + \dots + c_k g_k(x_i) - y_i] g_2(x_i) = 0;$$

.....

$$\frac{\partial E}{\partial c_k} = 2 \sum [c_1 g_1(x_i) + \dots + c_k g_k(x_i) - y_i] g_k(x_i) = 0.$$

Эти k уравнений можно представить в виде

$$\begin{bmatrix} \sum g_1^2(x_i) & \sum g_1(x_i) g_2(x_i) & \cdots & \sum g_1(x_i) g_k(x_i) \\ \sum g_1(x_i) g_2(x_i) & \sum g_2^2(x_i) & \cdots & \sum g_2(x_i) g_k(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum g_1(x_i) g_k(x_i) & \sum g_2(x_i) g_k(x_i) & \cdots & \sum g_k^2(x_i) \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} \sum g_1(x_i) y_i \\ \sum g_2(x_i) y_i \\ \vdots \\ \sum g_k(x_i) y_i \end{bmatrix}. \quad (6.6)$$

Так как элементы матрицы в левой части и вектора-столбца в правой определяются табличными данными, то полученная система k линейных уравнений с k неизвестными может быть решена. Функции $g(x)$ должны выбираться с учетом характера табличных данных, их периодичности, свойств симметрии, наличия асимптотики. Иногда таблицу разбивают на несколько частей и подбирают отдельную аппроксимирующую кривую для каждой части. Такой подход оправдан в случаях, когда данные соответствуют разным физическим состояниям системы.

Если при построении аппроксимирующей функции в качестве $g_j(x)$ используются ортогональные полиномы [1], для которых

$$\sum g_j(x_i) g_k(x_i) = 0, \quad j \neq k,$$

то система (6.6) упрощается, и матрица становится диагональной. Коэффициенты определяются как

$$c_j = \frac{\sum_{i=0}^n g_j(x_i) y_i}{\sum_{i=0}^n g_j^2(x_i)}.$$

Поэтому во многих стандартных программах подгонки кривых используются ортогональные полиномы.

6.3. Численное дифференцирование

При решении практических задач часто возникает необходимость нахождения производных функций, заданных таблично. В этом случае непосредственное дифференцирование может дать бессмысленные результаты, так как не отделен шум, который искажает производную

(рис. 6.2), где U_n — истинный сигнал, U_g — измеренный сигнал. Поэтому практическим решением этой задачи является интерполяция, а затем дифференцирование интерполяционного многочлена. При этом для получения производных высших порядков необходимо ставить условие их совпадения для сигнала и его интерполяционного многочлена. Погрешность производной интерполирующей функции $\Delta^*(x)$ равна производной от погрешности этой функции $\Delta'(x)$:

$$\Delta x = f(x) - P(x);$$

$$\Delta^*(x) = f'(x) - P'(x),$$

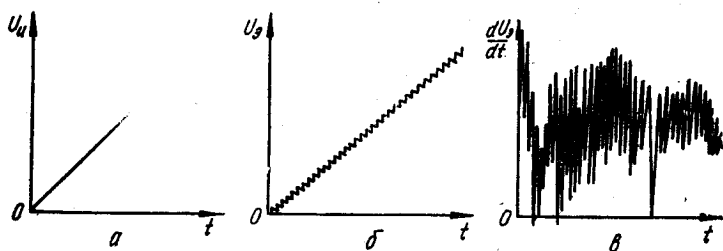


Рис. 6.2.

и вследствие линейности операций дифференцирования и вычитания имеем

$$\Delta^*(x) = \Delta'(x).$$

Существуют таблицы, позволяющие находить с использованием различных разностных интерполяционных формул (Лагранжа, Ньютона, Гаусса, Бесселя, Стирлинга) значения производных.

6.4. Численное интегрирование

В задачах, связанных с анализом, идентификацией, оценкой качества, моделированием различных устройств автоматики, управления, информационно-измерительной техники, радиоэлектроники, возникает необходимость вычисления определенных интегралов.

Если функция $f(x)$ непрерывна на отрезке $[a, b]$ и известна ее первообразная $F(x)$, то определенный

интеграл от этой функции в пределах от a до b может быть вычислен по формуле Ньютона — Лейбница

$$I = \int_a^b f(x) dx = F(b) - F(a),$$

где

$$F'(x) = f(x). \quad (6.7)$$

Однако в ряде случаев точное вычисление интеграла по формуле (6.7) затруднительно из-за большой сложности аналитического определения первообразной. В распространенной задаче, когда подынтегральная функция задается таблично (массивом значений), понятие первообразной теряет смысл, и интеграл может быть вычислен только численно.

Задача численного интегрирования функции заключается в вычислении значения определенного интеграла на основании ряда значений подынтегральной функции. Графически интеграл определяется площадью, ограничиваемой графиком функции

$$y = f(x).$$

Наиболее известными из используемых на практике методов нахождения определенных интегралов являются:

методы Ньютона — Котеса, Гаусса, Чебышева, основанные на использовании так называемых квадратурных формул, получающихся заменой $f(x)$ интерполяционными многочленами;

методы Монте-Карло, основанные на использовании статистических моделей.

Формулы Ньютона — Котеса. Для вывода формул Ньютона — Котеса интеграл (6.7) представляется в виде

$$\int_a^b f(x) dx = \sum_{i=0}^n A_i f(x_i) + \Delta, \quad (6.8)$$

где x_i — узлы интерполяции; A_i — коэффициенты, зависящие от вида формулы; Δ — погрешность квадратурной формулы.

Заменяя в (6.8) подынтегральную функцию соответствующим интерполяционным полиномом Лангража для n равноотстоящих узлов с шагом $h = \frac{b-a}{n}$, можно полу-

чить следующую формулу для расчета коэффициентов A_i при произвольном количестве узлов

$$A_i = \frac{b-a}{n} \frac{(-1)^{n-1}}{i!(n-i)!} \int_0^n \frac{q(q-1)\dots(q-n)}{(q-i)} dq, \quad (6.9)$$

$$i = 0, 1, 2, \dots, n,$$

где $q = \frac{x-a}{h}$ — приведенная переменная.

Обычно коэффициенты $H_i = \frac{A_i}{b-a}$ называют коэффициентами Котеса.

При этом формула (6.9) принимает вид

$$\int_a^b f(x) dx = (b-a) \sum_{i=0}^n H_i f(x_i) \quad (6.10)$$

и обладает следующими очевидными свойствами:

$$\sum_{i=0}^n H_i = 1 \text{ и } H_i = H_{n-i}.$$

При $n = 1$ и $n = 2$ из (6.9) и (6.10) получаем формулы трапеций и Симпсона:

$$I = \frac{h}{2} [f(x_0) + f(x_1)]$$

и

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)].$$

Погрешности этих формул определяются соответственно из выражений

$$\Delta = -\frac{h^3}{12} M_2 \text{ и } \Delta = -\frac{h^5}{90} M_4,$$

где M_2 и M_4 — максимальные значения второй и четвертой производных $f(x)$ при $x \in (a, b)$.

В табл. 6.2 приведены значения коэффициентов Котеса и оценок погрешности для значений n от 1 до 8. Поскольку коэффициенты Котеса при большом числе ординат сложны, практически для приближенного вычисления определенных интегралов разбивают промежуток интегрирования на большое число мелких промежутков и к каждому из них применяют квадратурную формулу Ньютона — Котеса с малым числом ординат. Получа-

Таблица 6.2. Коэффициенты Котеса

$\hat{H}_i = \frac{1}{N} \sum_{j=0}^n \hat{H}_i^j$	\hat{H}_0	\hat{H}_1	\hat{H}_2	\hat{H}_3	\hat{H}_4	\hat{H}_5	\hat{H}_6	\hat{H}_7	\hat{H}_8	Общий знаменатель N
1	1	1								2
2	1	4	1							6
3	1	3	3	1						8
4	7	32	12	32	7					90
5	19	75	50	50	75	19				288
6	41	216	27	272	27	216	41			840
7	751	3577	1223	2989	2989	1323	3577	751		17280
8	989	5888	-928	10496	-4540	10496	-928	5888	989	28350

ются формулы более простой структуры, точность которых может быть произвольно высокой. Например, полученные таким образом формулы трапеций и Симпсона (парабол) имеют вид

$$I = \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)]$$

и

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)].$$

При вычислении определенных интегралов следует учитывать погрешность нахождения значений $f(x_i)$. Если $f(x_i)$, например, будут заданы с одинаковой абсолютной погрешностью ε , то суммарная погрешность Δ_ε составит

$$\Delta_\varepsilon = \frac{b-a}{n} \sum_{i=1}^n \varepsilon = (b-a)\varepsilon.$$

Если использование формул оценки погрешности метода сопряжено с трудностями, обусловленными необходимостью нахождения высших производных, то можно применять практический метод экстраполяции Ричардсона (см. гл. 4).

Точность квадратурных формул с фиксированным расположением равноотстоящих узлов ограничена возможностями используемых методов интерполяции.

Формула Чебышева. Формула (6.8) может быть приведена к виду

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^n A_i f(t_i) \quad (6.11)$$

заменой переменных

$$x = \frac{a+b}{2} + \frac{b-a}{2} t.$$

При выводе формулы Чебышева используются следующие условия: коэффициенты A_i равны между собой; квадратурная формула (6.11) точна для всех полиномов до степени n включительно.

Учитывая, что

$$A_1 = A_2 = \dots = A_n = A$$

и при $f(t) = 1$

$$\sum_{i=1}^n A_i = nA = 2,$$

получим

$$A = \frac{2}{n}.$$

Тогда формула (6.11) будет иметь вид

$$\int_{-1}^1 f(t) dt = \frac{2}{n} \sum_{i=1}^n f(t_i). \quad (6.12)$$

Для нахождения t_i используем второе условие, согласно которому формула (6.12) должна быть точной для функций вида

$$f(t) = t^k, \quad k = 1, \dots, n.$$

После подстановки этих функций в (6.12) получаем систему уравнений

$$\left. \begin{aligned} t_1 + t_2 + \dots + t_n &= 0; \\ t_1^2 + t_2^2 + \dots + t_n^2 &= \frac{n}{3}; \\ \dots & \\ t_1^n + t_2^n + \dots + t_n^n &= \frac{n[1 - (-1)^{n+1}]}{2(n+1)}. \end{aligned} \right\} \quad (6.13)$$

Система уравнений (6.13) имеет решение при $n < 8$ и $n = 9$. Значения абсцисс t_i в формуле Чебышева приведены в табл. 6.3. В этой ограниченной точности и состоит принципиальный недостаток формулы Чебышева.

Таблица 6.3. Значения абсцисс t_i в формуле Чебышева

n	t	t_i	n	t	t_i
2	1; 2	$\mp 0,577350$	6	1; 6	$\mp 0,866247$
3	1; 3	$\mp 0,707107$		2; 5	$\mp 0,422519$
	2	0		3; 4	$\mp 0,266635$
4	1; 4	$\mp 0,794654$	7	1; 7	$\mp 0,883862$
	2; 3	$\mp 0,187592$		2; 6	$\mp 0,529657$
5	1; 5	$\mp 0,832498$		3; 5	$\mp 0,323912$
	2; 4	$\mp 0,3745413$		4	0
	3	0			

Формула Гаусса. Формула Гаусса называется формулой наивысшей алгебраической точности. Для формулы вида (6.11) наивысшая точность может быть достигнута для полиномов степени $(2n - 1)$, которые определяются $2n$ постоянными t_i и A_i ($i = 1, 2, \dots, n$).

Для обеспечения этого условия необходимо и достаточно, чтобы оно выполнялось для функций вида

$$f(t) = t^k, \quad k = 0, 1, \dots, 2n - 1.$$

Действительно, считая, что $f(t)$ может быть аппроксимирована полиномами степени $(2n - 1)$

$$f(t) = \sum_{k=0}^{2n-1} C_k t^k,$$

получим

$$\begin{aligned} \int_{-1}^1 f(t) dt &= \sum_{k=0}^{2n-1} C_k \int_{-1}^1 t^k dt = \sum_{k=0}^{2n-1} C_k \sum_{i=1}^n A_i t_i^k = \\ &= \sum_{i=1}^n A_i \sum_{k=0}^{2n-1} C_k t_i^k A_i f(t_i). \end{aligned}$$

Задача состоит в определении коэффициентов A_i и абсцисс точек t_i .

Для нахождения этих постоянных рассмотрим выполнение формулы (6.11) для функций вида

$$f(t) = t^k, \quad k = 0, 1, \dots, 2n - 1.$$

Учитывая, что

$$\int_{-1}^1 t^k dt = \begin{cases} 2/(k+1), \\ 0 \end{cases}$$

получим систему уравнений

$$\left. \begin{aligned} \sum_{i=1}^n A_i &= 2; \\ \sum_{i=1}^n A_i t_i &= 0; \\ \sum_{i=1}^n A_i t_i^2 &= 1; \\ \dots &\dots \dots \dots \dots \dots \\ \sum_{i=1}^n A_i t_i^{2n-2} &= \frac{2}{2n-1}; \\ \sum_{i=1}^n A_i t_i^{2n-1} &= 0. \end{aligned} \right\} \quad (6.14)$$

Эта система нелинейна, и ее обычное решение связано со значительными вычислительными трудностями. Однако если использовать систему для полиномов вида

$$f(t) = t^k P_n(t), \quad k = 0, 1, \dots, n-1, \quad (6.15)$$

где $P_n(t)$ — полином Лежандра, то ее можно свести к линейной относительно коэффициентов A_i с заданными точками t_i .

Полиномами Лежандра называются полиномы вида

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n],$$

обладающие следующими основными свойствами:

- 1) $P_n(1) = 1$, $P_n(-1) = (-1)^n$ для любого целого n ;
- 2) ортогональностью

$$\int_{-1}^1 P_n(x) Q_k(x) dx = 0, \quad (6.16)$$

где $Q_k(x)$ — любой полином степени $k < n$;

- 3) наличием n действительных корней на интервале $(-1; 1)$.

Первые пять полиномов Лежандра имеют вид:

$$\begin{aligned} P_0(x) &= 1; \\ P_1(x) &= x; \\ P_2(x) &= \frac{1}{2}(3x^2 - 1); \\ P_3(x) &= \frac{1}{3}(5x^3 - 3x); \\ P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3). \end{aligned}$$

Поскольку степени полиномов в соотношении (6.15) не превышают $2n - 1$, должна выполняться система (6.14) и формула (6.11)

$$\int_{-1}^1 t^k P_n(t) dt = \sum_{i=1}^n A_i t_i^k P_n(t_i). \quad (6.17)$$

Вследствие свойства ортогональности (6.16) левая часть выражения (6.17) равна нулю, тогда

$$\sum_{i=1}^n A_i t_i^k P_n(t_i) = 0,$$

что всегда обеспечивается при любых значениях A_i в точках t_i , соответствующих корням соответствующих полиномов Лагранжа.

Подставив эти значения t_i в систему (6.14) и учитывая первые n уравнений, можно определить коэффициенты A_i .

Формула (6.11), где t_i — нули полинома Лежандра $P_n(t)$, а A_i , $i = 1, 2, \dots, n$, определяются из системы (6.14), называется формулой Гаусса.

В табл. 6.4 приведены значения t_i и A_i в формуле Гаусса для различных n от 1 до 8.

Оценка погрешности формулы Гаусса с n узлами определяется из соотношения

$$\Delta = \frac{(b-a)^{2n+1} (n!)^4 M_{2n}}{[(2n)!]^3 (2n+1)},$$

где M_{2n} — максимальное значение $2n$ -й производной на участке $[a, b]$.

Стандартными программами, использующими формулы Гаусса с различным числом узлов как формулы, обеспечивающие наивысшую точность, оснащены многие

Таблица 6.4. Элементы формулы Гаусса

n	i	t_i	A_i
1	1	0	2
2	1; 2	$\mp 0,57735027$	1
3	1; 3	$\mp 0,77459667$	$\frac{5}{9} = 0,55555556$
	2	0	$\frac{8}{9} = 0,88888889$
4	1; 4	$\mp 0,86113631$	0,34785484
	2; 3	$\mp 0,33998104$	0,65214516
6	1; 6	$\mp 0,93246951$	0,17132450
	2; 5	$\mp 0,66120939$	0,36076158
	3; 4	$\mp 0,23861919$	0,46791394
7	1; 7	$\mp 0,94910791$	0,12948496
	2; 6	$\mp 0,74153119$	0,27970540
	3; 5	$\mp 0,40584515$	0,38183006
	4	0	0,41795918
8	1; 8	$\mp 0,96028986$	0,10122854
	2; 7	$\mp 0,79666648$	0,22238104
	3; 6	$\mp 0,52553142$	0,31370664
	4; 5	$\mp 0,18343464$	0,36268378

пакеты программ для научных исследований и инженерных расчетов, имеющиеся на различных вычислительных центрах (гл. 7).

Метод Монте-Карло. Этот метод является наиболее известным приложением общего метода статистического моделирования к решению инженерных задач.

Если последовательность случайных чисел $\{x_i\}$ (X с законом распределения вероятностей $f_X(x)$) подвергнуть функциональной обработке

$$Y = \varphi(x),$$

то математическое ожидание полученной последовательности случайных чисел $\{y_i\}$ (Y)

$$m_Y = \int_{-\infty}^{\infty} \varphi(x) f_X(x) dx \quad (6.18)$$

может быть при достаточно большом объеме выборки ($n > 1000$) с высокой точностью ($\Delta_{\text{отн}} < 0,1$) оценено по формуле

$$m_Y = \frac{1}{n} \sum_{i=1}^n y_i. \quad (6.19)$$

Если в выражения (6.18) и (6.19) ввести индикатор области $[a, b]$ функцию

$$1[a, b, x] = \begin{cases} 1, & a \leq x \leq b; \\ 0, & x < a, x > b \end{cases}$$

и выбрать

$$\varphi(x) = \frac{f(x)}{f_X(x)},$$

где $f(x)$ — подынтегральная функция, а a, b — пределы интегрирования из формулы (6.11), то окончательное выражение примет вид

$$I = m_Y = \int_a^b f(x) dx = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{f_X(x_i)} 1[a, b, x].$$

Алгоритм вычисления определенного интеграла по методу Монте-Карло на ЭВМ показан на рис. 6.3.

Погрешность метода Монте-Карло определяется погрешностью генерации псевдослучайной числовой последовательности на ЭВМ и объемом выборки и может быть оценена из соотношения

$$\Delta = \frac{1}{2\sqrt{n(1-P)}},$$

где P — гарантированная вероятность попадания погрешности в интервал $[-\Delta; +\Delta]$.

Число испытаний n не зависит от размерности интеграла I , поэтому метод Монте-Карло выгодно использовать для вычисления многократных интегралов, где применять другие методы численного интегрирования трудно (например, вычисление десятикратного интеграла в единичном объеме с шагом $h = 0,1$ требует вычисления суммы примерно 10^{10} слагаемых).

Численное интегрирование не ограничивает область применения метода статистических испытаний в инженерных расчетах. Известны его достаточно эффективные приложения для решения систем алгебраических и дифференциальных уравнений и др.

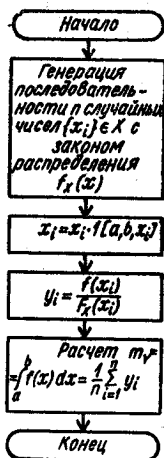


Рис. 6.3.

6.5. Статистическая обработка данных

При обработке результатов экспериментов в измерительной технике (например, для оценки погрешности измерений), автоматике (решение задач идентификации, оптимального управления), статистической радиотехнике возникает необходимость оценки характеристик случайной величины [4].

В качестве оценки \bar{x} неизвестного математического ожидания m_X случайной величины X используется среднее арифметическое результатов n независимых испытаний

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n},$$

а для оценки дисперсии σ_X^2 соотношение

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

При допущении нормального закона распределения величины x можно показать, что величина

$$T = \frac{\bar{x} - m_X}{\bar{\delta}/\sqrt{n}}$$

имеет t -распределение Стьюдента с $k = n - 1$ степенями свободы. Отсюда можно определить доверительный интервал для истинного значения x : по известным значениям доверительной вероятности P из таблицы (прил. 2) находим ϵ , откуда

$$\Delta = \epsilon \cdot \frac{\bar{\delta}}{\sqrt{n}}.$$

Таким образом, случайная величина x распределена по нормальному закону с математическим ожиданием m_X и дисперсией σ_X^2 . Истинное значение x находится в интервале $(m_X - \Delta, m_X + \Delta)$ с доверительной вероятностью P .

Для оценки вида закона распределения наибольшее применение имеют критерии Колмогорова и Пирсона, которые позволяют на основе сравнения эмпирической

функции распределения $f_x^*(x)$, полученной в виде гистограммы в результате обработки экспериментальных данных, с гипотетической $f_x(x)$, соответствующей выдвинутой гипотезе, сделать вывод об их совпадении или несовпадении при уровне значимости α , определяемом как вероятность того, что будет отвергнута верная гипотеза.

В критерии Колмогорова мерой служит величина

$$\lambda = |f_X(x) - f_X^*(x)|_{\max} \sqrt{n},$$

которая сравнивается с критическим значением, задаваемым по табл. 6.5.

Таблица 6.5. Критические значения λ_0 в зависимости от уровня значимости

α	0,50	0,40	0,30	0,20	0,10	0,05	0,02	0,01	0,001
λ_0	0,828	0,895	0,974	1,073	1,224	1,358	1,520	1,627	1,950

При $\lambda < \lambda_{кр}$ гипотеза о совпадении $f_X(x)$ и $f_X^*(x)$ принимается.

В критерии Пирсона вычисляется величина

$$\chi^2 = \sum_{i=0}^k \frac{[f_X(x_i) - f_X^*(x_i)]^2}{f_X(x_i)},$$

где k — число разрядов гистограммы (дискретных значений $f_X(x_i)$).

По таблице (прил. 3) определяют критическое значение χ^2 , учитывая α и число степеней свободы

$$r = k - l - 1,$$

где l — число параметров, входящих в закон распределения (для нормального $l = 2$, пуассоновского $l = 1$ и т. д.).

При $\chi^2 < \chi^2_{кр}$ гипотеза принимается.

Если сравнивают аналитически полученные законы распределения вероятностей, то мерой их близости служит значение средней квадратической погрешности.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Сформулируйте задачу интерполяции.
2. В каких случаях интерполяция невозможна?
3. С какой точностью можно вычислить $\sqrt{2}$ с помощью формулы Лагранжа для функции \sqrt{x} , выбрав узлы интерполяции $x_0=81$, $x_1=1$, $x_2=4$?
4. Выведите интерполяционную формулу Ньютона для равноотстоящих узлов.
5. Составьте алгоритм вычисления полинома Лагранжа.
6. Что такое сплайн-интерполяция? Как определяются коэффициенты сплайнов?
7. Опишите области применения разностных методов интерполяции.
8. Что такое экстраполяция?
9. Какими путями решается задача аппроксимации?
10. Выведите систему уравнений для определения коэффициентов аппроксимирующего полинома в методе наименьших квадратов.
11. Какие полиномы называются ортогональными? Приведите примеры.
12. Как осуществляется численное дифференцирование?
13. Какие методы численного интегрирования используются при решении задач на ЭВМ?
14. Как получаются формулы Ньютона—Котеса? Получите формулы трапеций и Симпсона для численного интегрирования. Выведите составные формулы Ньютона—Котеса.
15. Какой порядок погрешностей формул Ньютона — Котеса? Как оценивается погрешность интегрирования при использовании этих формул?
16. Выведите формулу Чебышева для численного интегрирования функции, заданной в трех точках.
17. Как определяются коэффициенты в формуле Гаусса для численного интегрирования? Почему говорят, что эта формула имеет наивысшую алгебраическую точность?
18. Разработайте алгоритм и приведите пример программы для ЭВМ численного интегрирования по методу Монте-Карло для двукратного интеграла.
19. Разработайте алгоритм и приведите пример программы для ЭВМ оценки статистических характеристик результатов измерений величины x при наличии случайных погрешностей измерения.

Глава 7. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ

Опыт исследователей при алгоритмизации и решении на ЭВМ различных инженерных задач накапливается в виде разнообразных пакетов прикладных программ (ППП), представляющих собой комплекс завершенных программных модулей и системных средств, объеди-

ненных общей структурой, правилами использования и доступа и предусматривающих возможность решения определенных классов инженерных и научных задач.

Среди ППП могут быть выделены пакеты программ для инженерных и научных расчетов, объединяющие программы для решения типовых задач вычислительной математики и математической физики, и пакеты программ для моделирования систем, базирующиеся на различных математических методах анализа и синтеза систем и представляющие наибольший интерес для специалистов в области общей теории систем управления, информационно-измерительной техники, автоматики и телемеханики.

Отдельную группу составляют ППП для аналитических вычислений, получающие в последнее время все более широкое распространение. Также следует отметить программные средства общего назначения (системы управления базами данных, информационно-поисковые и информационно-справочные системы), ППП по технологии и автоматизации программирования, организации вычислительного процесса, обработке данных и планированию в экономических задачах и ППП оптимизации.

В ППП, как правило, объединяются наиболее эффективные программные средства, имеющие высокие характеристики и прошедшие всестороннюю проверку.

По структуре и принципам использования ППП делят на несколько типов.

1. «Черные ящики», представляющие собой замкнутые многовариантные универсальные системы. Механизм действий таких пакетов скрыт для пользователя, они удобны в эксплуатации, но в большинстве случаев недоступны для модификации. Как правило, «черные ящики» используют специализированный проблемно-ориентированный язык программирования.

2. Препроцессоры, объединяющие части программ на языке высокого уровня и стандартные процедуры или подпрограммы. К таким ППП относится большинство пакетов общего назначения.

3. Конструкторы, в которых подпрограммы объединяются с помощью языков Ассемблера. Допускается использование в таком режиме пакетов общего назначения.

4. Адаптивные диалоговые системы, являющиеся комбинациями ППП других типов, оснащенные диалого-

выми средствами, обеспечивающими возможность оперативного управления выполнением заданий.

В общем виде ППП состоит из трех основных компонентов: функционального и системного наполнения и языка управления заданиями.

Функциональное наполнение отражает особенности предметной области ППП и представляет собой совокупность программных модулей, реализующих решение частных задач. Состав функционального наполнения отражает объем прикладных знаний, заложенных в пакете. Язык управления заданиями является средством общения пользователя с пакетом. Он позволяет описать необходимую последовательность выполнения различных операций для решения поставленной задачи. Ряд ППП, в частности универсальные пакеты, предназначенные для решения широкого класса инженерных задач, не предусматривают создания специального проблемно-ориентированного языка, а используют для управления заданиями какой-либо из известных языков программирования высокого уровня (чаще всего ФОРТРАН), причем обращение к прикладной программе осуществляется по правилам этого языка, например с помощью оператора CALL.

Такие ППП не имеют системного наполнения. Многие ППП представляют собой системы моделирования и решения инженерных задач на ЭВМ и имеют специализированные проблемно-ориентированные языки программирования, которые используют операторы, и обозначения, соответствующие данной предметной области. Для таких систем характерно наличие системного наполнения, представляющего собой совокупность программ, обеспечивающих автоматизацию выполнения задания и взаимодействие пользователя с пакетом в рамках принятой в пакете технологии программирования.

В настоящее время в связи с созданием проектов и началом разработки ЭВМ пятого поколения проводятся исследования по созданию интеллектуальных пакетов прикладных программ, которые дают возможность пользователю решать задачи по их словесным описаниям и исходным данным без программирования процесса решения задачи (например, ППП ПРИЗ и МАВР). Использование таких ППП предусматривает автоматическое построение математической модели задачи по словесному описанию и построение программы для ее решения.


```

C
C
SUBROUTINE GRAF(Y,Z,NZ)
BYTE S(51)
REAL Y(NZ)
REAL Z(NZ)
INTEGER IY(60)
INTEGER IZ(60)
DATA S/50*' ','>'/
YMIN=AMIN1(Y(1),Z(1))
YMAX=AMAX1(Y(1),Z(1))
DO 20 I=1,NZ
  IF(Y(I).GT.YMAX)YMAX=Y(I)
  IF(Z(I).GT.YMAX)YMAX=Z(I)
  IF(Y(I).LT.YMIN)YMIN=Y(I)
  IF(Z(I).LT.YMIN)YMIN=Z(I)
20CONTINUE
DO 30 I=1,NZ
  IY(I)=(Y(I)-YMIN)*50./
    & (YMAX-YMIN)+1
  IZ(I)=(Z(I)-YMIN)*50./
    & (YMAX-YMIN)+1
30CONTINUE
IOS=-YMIN*50./<YMAX-YMIN)+1
PRINT1010
DO 50 I=1,NZ
S(IOS)='I'
S(IY(I))='Y'
S(IZ(I))='Z'
PRINT1000,S
DO 40 J=1,51
S(J)=' '.
40CONTINUE
50CONTINUE
S(IOS)='U'
PRINT1000,S
RETURN
1000FORMAT(X,51A1)
1010FORMAT('1')
END

```

Рис. 7.1.

7.1. Универсальные пакеты прикладных программ

Универсальные ППП широко используются в вычислительных центрах [20]. Существуют пакеты, ориентированные на различные типы вычислительных машин и операционных систем.

В качестве примера рассмотрим ППП SSP для ЭВМ типа СМ-4 и ее модификаций (СМ-1407, СМ-1420) в операционной системе ОС РВ. Состав этого пакета в основном совпадает с ППП, поставляемым в качестве

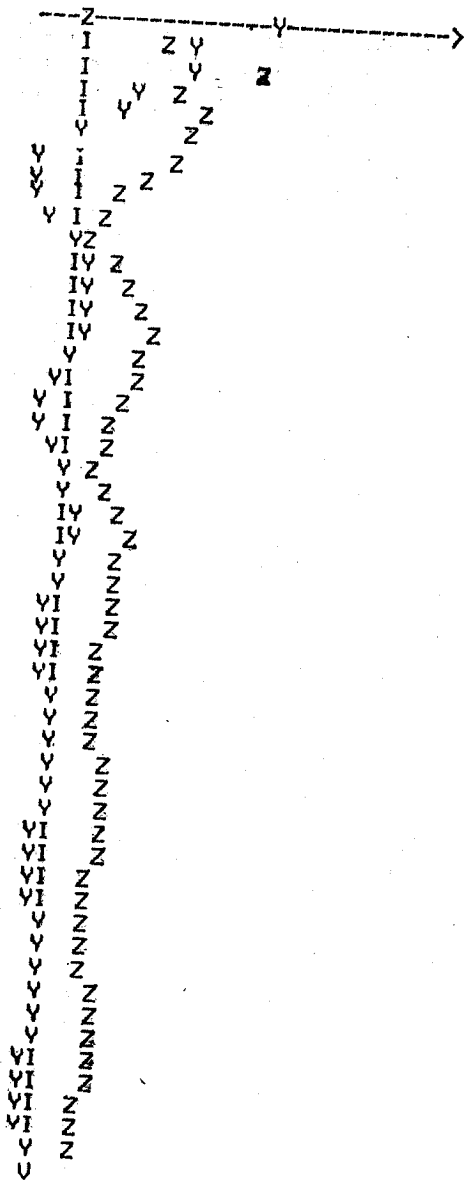


Рис. 7.1.

стандартного математического обеспечения для ЕС ЭВМ и библиотекой программ для научно-технических расчетов (БНТР) для СМ ЭВМ в операционной системе РАФОС.

Пакет SSP состоит из набора программ, написанных на языке ФОРТРАН (без подпрограмм ввода-вывода), которые могут быть скомбинированы с программами пользователей. Прикладные программы из пакета SSP могут быть использованы при решении многих распространенных инженерных вычислительных и статистических задач. В прил. 4 приведены перечень и назначение прикладных программ из пакета SSP. На рис. 7.1 приведен пример интегрирования функции, заданной таблицей значений в неравноотстоящих точках с помощью программы QTFG из ППП SSP.

Во многих вычислительных центрах известные ППП дополняются новыми программами, что позволяет значительно расширить их функциональные возможности.

7.2. Специализированные пакеты прикладных программ

ППП, предназначенные для решения определенных классов задач, относятся к специализированным. Как правило, такие ППП кроме функционального наполнения имеют собственный язык управления заданиями и системное наполнение. В качестве примера рассмотрим получивший широкую известность ППП САФРА, разработанный в институте прикладной математики им. М. В. Келдыша АН СССР. Его функциональное наполнение ориентировано на широкий класс задач газовой динамики, магнитной и радиационной газовой динамики с учетом различных физических процессов. Пакет САФРА используется в ОС ДИСПАК на ЭВМ БЭСМ-6.

Система САФРА включает архив пакета, предназначенный для хранения накапливаемого программного фонда и формулируемой на языке заданий информации о структуре собираемых программ.

Хранимые в архиве единицы информации называются модулями (программные и специальные конструктивные или отладочные единицы) и являются основными элементами обработки в ППП САФРА. Разработка программ ведется по специальной технологии, использующей принцип, аналогичный пошаговой детализации. Работа

с пакетом организуется посредством выполнения заданий, написанных пользователем на языке заданий. Задание состоит из совокупности этапов, выполняемых последовательно и включающих такие действия, как запись и исключение модулей из архива; редактирование текстов, хранящихся в архиве модулей; сборку и запуск на счет вариантов программы; вспомогательные операции (распечатка текстов, каталога).

ППП САФРА представляет собой сложную систему, эффективное использование которой пользователями возможно лишь при наличии достаточно полной и удобной системы информационного обеспечения. В качестве такой системы в ИПМ им. М. В. Келдыша АН СССР разработана и используется диалоговая система ИНФРА, которая связана с системным наполнением ППП САФРА, однако функционирует автономно на собственных ресурсах.

7.3. Системы для аналитических вычислений на ЭВМ

Создание алгоритмических языков высокого уровня (ФОРТРАН, АЛГОЛ, ПЛ/1, ЛИСП) обусловило успехи в реализации на ЭВМ различных аналитических операций.

В настоящее время создано более тридцати ППП — систем для аналитических вычислений (САВ), многие из которых широко применяются для решения различных инженерных, физических и математических задач [2].

С помощью САВ выполняются такие аналитические операции, как дифференцирование, упрощение выражений (приведение подобных членов), подстановка вместо символа или выражения другого выражения и т. п. Результат вычисления при этом представляет собой некоторое аналитическое выражение, например функцию с явной зависимостью от ее аргументов.

Аналитические вычисления удалось реализовать на цифровых вычислительных машинах благодаря развитию ряда специальных методов программирования. При выполнении численных расчетов распределение информации в памяти машины обычно фиксируется, и ее объем известен еще до начала работы программы. При проведении алгебраических выкладок трудно, а иногда и невозможно заранее точно учесть объем памяти, который

потребуется на промежуточных этапах. Один из способов решения этой проблемы — динамическое распределение памяти: на промежуточных этапах вычислений стирается уже ненужная информация и освободившееся место заполняется новой.

Другой особенностью аналитических преобразований является более сложная структура элементарных операций: сложения, умножения и т. д., которые должны быть модифицированы программным путем. Кроме того, усложняются команды по управлению как исходной информацией, так и результатами вычислений. В этом основное различие аналитических и численных выкладок.

В некоторых случаях для проведения аналитических выкладок можно обойтись без непосредственного использования символов, которые, так же как и числа, легко записать в машинной памяти в двоичном коде. Например, полином $(54/7)x^3yz^2$ по переменным x, y, z, w может быть совершенно однозначно описан последовательностью чисел: 54, 7, 3, 1, 2, 0. На основе такого представления легко составить компактную и быструю систему, но даже такую простую символьную операцию, как дифференцирование, запрограммировать очень сложно.

САВ на ЭВМ может быть создана только в случае, когда процедура нахождения решения достаточно ясна. Их можно рассматривать как мощный и практически единственный инструмент решения задач, требующих непомерно больших затрат ручного труда или очень чувствительных к потере точности при численном решении.

К задачам первого типа относится, например, задача обращения матрицы достаточно высокого порядка, элементы которой являются символами или алгебраическими выражениями. Примером задачи второго типа является анализ устойчивости плазмы в установке термоядерного синтеза типа «Токамак», который сводится к условию существования нуля некоторой функции в заданной области. Положение этого нуля очень чувствительно к потере точности в промежуточных вычислениях. Применение САВ в данной задаче позволило получить важные практические результаты.

Все существующие в настоящее время САВ условно делят на три группы.

1. Специализированные системы — ориентированные на определенную область применения и в основном на

длинные выкладки. Для таких систем характерен узкий класс встроенных математических операций, использующих специальные алгоритмы и требующих высокого быстродействия и относительно небольшого объема памяти ЭВМ, например, САМАЛ, СЛАМ, АВТОАНАЛИТИК и др.

2. Системы общего назначения — со встроенными математическими операциями, типичными для многих задач физики и математики, например, дифференцированием, алгеброй векторов и матриц и др. (FORMAC, ALTRAN, АНАЛИТИК-74).

3. Универсальные системы, например, REDUCE-2, MACSYMA и SCRATCHPAD. Две последние являются наиболее мощными из существующих САВ и выполняют большинство аналитических операций, которые реализуются на современных ЭВМ. Система REDUCE-2 обладает значительно меньшим числом встроенных операций, чем MACSYMA, и SCRATCHPAD, однако имеет широкие возможности для определения новых объектов и новых математических операций средствами внешнего языка системы. Многие специализированные системы и системы общего назначения написаны на языке Ассемблера, что во многом и объясняет их компактность и быстродействие. Универсальные системы написаны на языке ЛИСП, обуславливающим широкие возможности для аналитических выкладок, большого машинного времени на выполнение отдельных операций.

Характеристики некоторых САВ представлены в табл. 7.1.

В отличие от численных программ, САВ требуют значительно большего объема памяти ЭВМ — десятков, а иногда и сотен тысяч машинных слов. Такие требования обусловлены характером используемых нечисленных алгоритмов и необходимостью сохранять в памяти все промежуточные результаты. Поэтому наиболее мощные САВ реализованы на больших ЭВМ. Примером САВ, реализованной на малой ЭВМ, может служить система АНАЛИТИК-74 для машин серии МИР-3, разработанной в СССР, и система VAXIMA для мини-ЭВМ PDP-11 — в США.

Из разработанных в нашей стране отметим такие САВ, как СИРИУС для М-20 и М-222, АВТО-АНАЛИТИК и АМС для БЭСМ-6, АНАЛИТИК-74. Помимо систем создан также ряд процессоров и ППП, например, про-

Таблица 7.1. Характеристики некоторых программных систем для аналитических вычислений

СAB	SCHOONSCHIP	SLAM	REDUCE-2	АВТО-АНАЛИТИК	MACSYMA
Год разработки версии ЭВМ	1977 CDC-6500	1972 CDC-6500	1973 CDC-6500 EC-1040	1973 БЭСМ-6	1977 DEC PDP-10
Объем системы	25000 слов	20000 слов	65000 слов (CDC) 300 кбайт (EC)	30000 слов	221000 слов
Язык реализации	Ассемблер	Ассемблер	ЛИСП	Машинный код	ЛИСП
Тип внешнего языка	Специальный язык программирования	ЛИСП	АЛГОЛ	Специальный язык программирования	АЛГОЛ
Диалоговый режим	Есть	Нет	Есть	Нет	Есть
Область основных применений	Квантовая теория поля	Общая теория относительности	Универсальная	Математическая физика	Универсальная
Страна-разработчик	США	Великобритания	США	СССР	США

Таблица 7.2. Встроенные математические объекты и операции над ними в различных САВ

САВ	SCHOONSCHIP	CLAM	REDUCE-2	АВТО-АНАЛИТИК	MACSYMA
Элементарные функции	Нет	Большинство	Некоторые	Большинство	Все
Дробно-рациональные выражения	Нет	Есть	Есть	Нет	Есть
Поиск наибольшего общего делителя	Нет	Нет	Есть	Нет	Есть
Дифференцирование	Нет	Есть	Есть	Есть	Есть
Интегрирование	Нет	Нет	Нет	В простых случаях	Есть
Комплексные величины	Есть	Нет	Есть	Нет	Есть
Рациональные числа	Есть	Есть	Есть	Нет	Есть
Арифметика чисел с плавающей запятой	Очень быстрая	Нет	Медленная	Очень быстрая	Быстрая
Работа с отрезками степенных рядов	Нет	Нет	Неплохая	Нет	Отличная
Работа с отрезками рядов Фурье	Нет	Нет	Нет	Нет	Хорошая
Алгебра векторов и тензоров	Неплохая	Специального вида	Хорошая	Нет	Отличная
Матричная алгебра	Нет	Нет	Хорошая	Неплохая	Отличная

цессоры КИНО и ПАССИВ для анализа групповых свойств дифференциальных уравнений; ППП, реализующие некоторые выкладки квантовой электродинамики, и т. д. В табл. 7.2 представлены перечень основных задач, решаемых различными САВ, а также степень их реализации.

В качестве примера использования САВ рассмотрим генерацию полиномов Лежандра с помощью системы REDUCE-2.

Для вычислений используем формулу

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n].$$

```

ARRAY P(3);
FOR I:=1%3 DO
  & BEGIN P(I)%=(X**2-1)**I;
FOR J:=1%I DO
  & P(I)%=DF(P(I),X)/(2*J);
WRITE P(I), = ,P(I) END;
P(I)=X

P(2)=(3*X2-1)/2
P(3)=(X*(5*X2-3))/2
END

```

Рис. 7.2.

Программа генерации полиномов $P_n(x)$ с $n = 1-3$ по этой формуле, использующая встроенный в систему REDUCE-2 оператор дифференцирования $DF(Q, x) = dQ/dx$, и результаты вычислений показаны на рис. 7.2.

7.4. Пакеты прикладных программ для моделирования систем

Метод математического моделирования на ЭВМ в настоящее время все более широко используются при исследованиях и проектировании систем, применительно к решению задач на разных этапах создания автоматизированных систем управления, систем автоматизации научных исследований и экспериментов, систем автоматизации проектирования, различных информационно-измерительных систем и т. д. Этими проблемами занято большое количество различных специалистов, часто не имеющих навыков работы с ЭВМ. Для таких пользователей должны быть разработаны средства подготовки и общения с ЭВМ.

В связи с этим возникает вопрос о создании автоматизированной системы моделирования, основой которой является специализированный ППП для моделирования систем. Одной из важнейших составных частей системы моделирования является база данных моделирования,

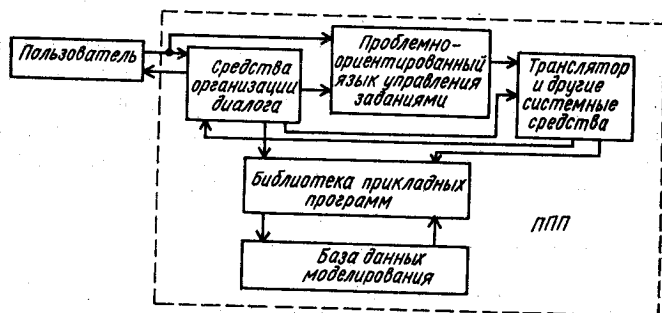


Рис. 7.3.

Таблица 7.3. Пакеты прикладных программ для моделирования систем

Наименование	Вид модели			Место разработки
	имитационная или аналитическая	дискретная или непрерывная	стохастическая или детерминированная	
SLAM	И	Н—Д	Д—С	Pritsker Associates (США) Университет шт. Висконсин (США)
DYSAC	А—И	Н—Д	Д	
GASP	И	Д	С	U. S. Steel Corporation (США) IBM (США)
GPSS	И	Н	Д	
SIPAC	А—И	Н—Д	С	Бухарестский университет (СРР)
CDSP	И	Н—Д	Д	Фирма «Мицубиси» (Япония) Институт кибернетики АН УССР им. В. М. Глушкова (СССР)
НЕДИС	А—И	Н—Д	С	
МАСС	А	Н—Д	Д	Московский энергетический институт (СССР)
МИКГ	А—И	Н—Д	С	Московский энергетический институт (СССР)
АРАМИС	А	Н—Д	С	Винницкий политехнический институт (СССР)

объединяющая сведения о свойствах и режимах функционирования объекта.

На рис. 7.3 показана структура взаимодействия пользователя с составными частями системы моделирования. В табл. 7.3 приведены сведения о некоторых разработанных ППП для моделирования систем.

Внедрение ППП для моделирования систем и устройств информационно-измерительной техники, автоматики, телемеханики, радиоэлектроники позволяет значительно повысить эффективность процесса проектирования и служит основой для разработки систем автоматизации проектирования (САПР).

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение ППП.
2. Как можно классифицировать ППП?
3. Что понимается под функциональным и системным наполнением ППП?
4. Как используются прикладные программы в универсальных пакетах и какие задачи они позволяют решать?
5. Как осуществляется обращение к программам из универсальных ППП?
6. Охарактеризуйте специализированные ППП. Приведите примеры.
7. Дайте общую характеристику САВ.
8. Какие пути построения и виды САВ Вам известны? Приведите примеры и характеристики разработанных САВ.
9. Какие стандартные программы для численного интегрирования, решения обыкновенных дифференциальных уравнений имеются в стандартном математическом обеспечении ЭВМ? Как ими пользоваться? Приведите пример программы.
10. Какие известные системы и ППП получили распространение для моделирования систем управления, технических средств кибернетики и информационно-измерительной техники?
11. Как происходит взаимодействие пользователя с ППП для моделирования систем?

Глава 8. ПРИМЕРЫ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ НА ЭВМ

Рассмотрим примеры решения типовых инженерных задач вычислительными методами на ЭВМ. Примеры относятся к разным областям инженерной практики, однако в них подобраны задачи, с которыми наиболее часто сталкиваются специалисты в области автоматики, телемеханики, информационно-измерительной техники и радиоэлектроники.

Это исследование устойчивости динамических систем на основе матричного критерия Зубова или путем нахождения собственных значений матрицы коэффициентов; спектральный анализ сигналов с использованием алгоритма быстрого преобразования Фурье; анализ вероятностных и спектральных характеристик случайных сигналов и их преобразования в различных устройствах; исследование поведения объектов, описываемых дифференциальными уравнениями. Примеры проиллюстрированы листингами программ с результатами решения на языке ФОРТРАН.

Пример 1. Для построения вероятностной модели измерительной информационной системы получить значения корреляционной функции на выходе цепочки преобразова-

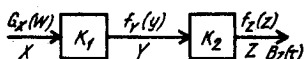


Рис. 8.1.

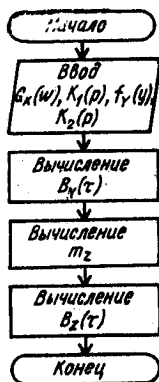


Рис. 8.2.

лей (рис. 8.1) при условии стационарности случайного процесса на входе:

Исходные данные:

Спектральная плотность $G_X(\omega)$	$K_1(p)$	Закон распределения $f_Y(y)$	$K_2(p)$
100, $\omega \in [0; 10]$	Линейный динамический преобразователь $\frac{1}{1+2j\omega}$	Нормальный $m_Y = 0$	Нелинейный статический преобразователь $z = e^{-V y }$

Общий алгоритм решения задачи представлен на рис. 8.2. В соответствии с этим алгоритмом проводим аналитические преобразования и расчеты [14].

Спектральная плотность сигнала Y определяется из следующего выражения:

$$G_Y(\omega) = |K_1(j\omega)|^2 G_X(\omega) = \frac{100}{1+4\omega^2},$$

$$0 \leq \omega \leq 10.$$

Корреляционная функция $B_Y(\tau)$ находится как результат обратного преобразования Хинчина-Винера от $G_Y(\omega)$:

$$B_Y(\tau) = \int_0^{10} G_Y(\omega) e^{j\omega\tau} d\omega = 200 \int_0^{10} \frac{\cos \omega\tau}{1+4\omega^2} d\omega. \quad (8.1)$$

Математическое ожидание

$$\begin{aligned} m_Z &= \int_{-\infty}^{\infty} z f_Z(z) dz = \int_{-\infty}^{\infty} e^{-V|y|} f_Y(y) dy = \\ &= \int_{-3\sigma_Y}^{3\sigma_Y} \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left(-V|y| - \frac{y^2}{2\sigma_Y^2}\right) dy, \end{aligned}$$

где

$$\begin{aligned} \sigma_Y^2 = D_Y = B_Y(0) &= \int_0^{10} G_Y(\omega) d\omega = \int_0^{10} \frac{100}{1+4\omega^2} d\omega = \\ &= 100 \operatorname{arctg} 10 = 177. \end{aligned}$$

Тогда

$$m_Z = \frac{1}{33,6} \int_{-40}^{40} \exp\left(-V|y| - \frac{y^2}{354}\right) dy. \quad (8.2)$$

Корреляционная функция $B_Z(\tau)$ находится как

$$\begin{aligned} B_Z(\tau) &= \int_{-3\sigma_Y}^{3\sigma_Y} \int_{-3\sigma_Y}^{3\sigma_Y} (e^{-V|y_1|} - m_Z)(e^{-V|y_2|} - m_Z) \times \\ &\quad \times f_Y(y_1, y_2, \tau) dy_1 dy_2, \end{aligned} \quad (8.3)$$

где $f_Y(y_1, y_2, \tau)$ — двумерный закон распределения y , определяющийся из соотношения

$$\begin{aligned} f_Y(y_1, y_2, \tau) &= \frac{1}{\sqrt{2\pi}\sigma_Y \sqrt{1 - B_Y^2(\tau)}} \times \\ &\quad \times \exp\left\{-\frac{y_1^2 + y_2^2 - 2B_Y(\tau)y_1y_2}{2\sigma_Y^2[1 - B_Y^2(\tau)]}\right\}. \end{aligned}$$

Таким образом, задача сводится к нахождению значений $B_Z(\tau_i)$ из (8.3), учитывая предварительно полученный массив значений $B_Y(\tau_i)$ из (8.1) и вычисленное из (8.2) m_Z .

Интервал дискретизации τ определяется в соответствии с теоремой Котельникова и равен

$$\Delta t = \frac{\pi}{\omega_{\max}} = 0,314.$$

Задача решается с применением стандартной библиотечной программы интегрирования нетабличной функции по квадратурным формулам Гаусса (рис. 8.3).

Пример 2. Для анализа устойчивости линейной системы автоматического управления найти собственные значения матрицы коэффициентов:

$$\begin{bmatrix} 9,1 & 3,0 & 2,6 & 4,0 \\ 4,2 & 5,3 & 4,7 & 1,6 \\ 3,2 & 1,7 & 9,4 & 1,0 \\ 6,1 & 4,9 & 3,5 & 6,2 \end{bmatrix}.$$

В соответствии с рекомендациями (гл. 2) выбираем QR -метод, как наиболее общий, приводящий к блочной треугольной матрице

$$\begin{bmatrix} x_1 & & & & * \\ & x_2 & & & \\ & & x_3 & & \\ & & & x_4 & \\ & & & & \ddots \\ 0 & & & & & \end{bmatrix},$$

где блоки x_i представляют собой матрицы размерности $[2, 2]$, расположенные на главной и двух прилегающих к ней диагоналях. Собственные значения блоков такой матрицы являются одновременно собственными значениями всей матрицы.

Для получения преобразованной матрицы необходимо исходную представить в виде произведения ортогональной матрицы на правую треугольную, а потом перемножить последние, предварительно поменяв их местами.

Такое преобразование производится до тех пор, пока преобразуемая и преобразованная матрицы не будут отличаться на достаточно малую величину (обычно для этого рассматривают наибольшую разность между двумя соответствующими элементами матриц).

Матрицы Q и R вычисляются по следующему алгоритму.

1. Первый столбец ортогональной матрицы Q берется таким же, как и у матрицы A размера $[n, n]$.

```

FUNKTION F(Y1,Y2,T,DY)
PI=3.1415926
A=2.*DY*(1-(BY(T)/DY)**2)
B=Y1**2+Y2**2-2.*Y1*Y2*BY(T)/DY
F=EXP(-B/A)/SQRT(PI*A)
RETURN
END

```

C
C
C

```

FUNKTION BZ(T,MZ,DY)
REAL MZ
N=10
H=80./N
S=0.
DO 20 I=1,N
Y1=-40.+H*I
DO 10 J=1,N
Y2=-40.+H*J
SS=(EXP(-SQRT(ABS(Y1)))-MZ)*
& (EXP(-SQRT(ABS(Y1)))-MZ)
S=S+SS*F(Y1,Y2,T,DY)
10 CONTINUE
20 CONTINUE
BZ=S
RETURN
END

```

MZ= 0.1104290 DY= 151.8871

T=0.001	BZ=1.0000								
T=0.002	BZ=0.4977								
T=0.003	BZ=0.3291								
T=0.004	BZ=0.2477								
T=0.005	BZ=0.1982								
T=0.006	BZ=0.1650								
T=0.007	BZ=0.1415								
T=0.008	BZ=0.1238								
T=0.009	BZ=0.1101								
T=0.010	BZ=0.0991								
T=0.011	BZ=0.0901								
T=0.012	BZ=0.0825								
T=0.013	BZ=0.0762								
T=0.014	BZ=0.0708								
T=0.015	BZ=0.0660								
T=0.016	BZ=0.0619								
T=0.017	BZ=0.0583								
T=0.018	BZ=0.0551								
T=0.019	BZ=0.0522								
T=0.020	BZ=0.0496								

```

PROGRAM MAL
REAL MZ
BYTE GG(30)
DATA GG/30*
N=100
H=80./N
MZ=0.
DO 10 I=1,N
Y=I*H-40.
MZ=MZ+EXP(-SQRT(ABS(Y))-Y**2
& /354.)*H/33.6
10 CONTINUE
DY=BY(0.)
PRINT*, 'MZ=', MZ, 'DY=', DY
T=1E-3
DZ=BZ(T, MZ, DY)
DO 20 I=1,20
T=.001*I
AA=BZ(T, MZ, DY)/DZ
IGG=30.*AA
GG(IGG)='*'
15 PRINT15, T, AA, GG
& FORMAT(' T=', F6.3,
& ' BZ=', F7.4,
& 5X, 30(A1))
GG(IGG)=' '
20 CONTINUE
END

C
C
C
C

FUNCTION BY(T)
COMMON/R/TD
EXTERNAL BUF
TD=T
XL=0.
XU=10.
CALL Q610(XL, XU, BUF, Y)
BY=Y
RETURN
END

C
C
C

FUNCTION BUF(W)
COMMON/R/TD
BUF=200.*COS(W*TD)/(1+4.*W**2)
RETURN
END

```

Рис. 8.3.

2. Элемент правой треугольной матрицы R , находящийся в i -й строке и j -м столбце, вычисляется по формуле

$$R_{ij} = \frac{\sum_{k=1}^n A_{kj} Q_{ki}}{\sum_{k=1}^n Q_{kj}^2}.$$

3. j -й столбец матрицы Q (кроме первого) вычисляется n -кратным (для каждой строки) применением формулы

$$Q_{ij} = \sum_{k=1}^{j-1} R_{kj} Q_{ik}.$$

Для нахождения собственных значений исходной матрицы необходимо найти собственные значения всех блоков квазидиагональной матрицы. Например, для блока

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

собственные значения находят по формуле

$$\lambda_{1,2} = [x_{11} + x_{22} \pm \sqrt{(x_{11} + x_{22})^2 - 4(x_{11}x_{22} - x_{12}x_{21})}] / 2.$$

Программа и результаты представлены на рис. 8.4.

Пример 3. Решить методом сплайнов краевую задачу нахождения функции y , удовлетворяющей дифференциальному уравнению

$$y' - 100y = 0$$

на участке $0 \leq x \leq 1$ при граничных условиях $y(0) = y(1) = 1$.

Такая задача возникает, например, в процессе проектирования измерителя температуры в плоском стержне. При этом функция y описывает изменение температуры стержня при заданной температуре на его концах и отсутствии внешних возмущающих воздействий.

Данная задача является частным случаем следующей краевой задачи:

$$y'' - g(x)y = r(x); \quad y(a) = y_a; \quad y(b) = y_b. \quad (8.4)$$

Для приближенного решения задачи (8.4) применяются параболические или кубические сплайны. Предпо-

```

PROGRAM SOB
COMPLEX RE,RC,RD,RF
REAL A(4,4),B(4,4),C(4,4)
REAL SP(4),P(4),PK(4)
N=4
OPEN(UNIT=1,NAME='SOB.DAT',
      &TYPE='OLD',READONLY)
READ(1,*)A
PRINT21,A
DO 2 I=1,N
DO 2 J=1,N
2B(I,J)=A(I,J)
DO 10 IL=1,N
PL=IL
ILL=IL-1
SP(IL)=0.
DO 3 I=1,N
3SP(IL)=SP(IL)+B(I,I)
P(IL)=-SP(IL)
IF(IL.EQ.1)GO TO 5
DO 4 I=1,ILL
4P(IL)=P(IL)-P(I)*SP(IL-I)
P(IL)=P(IL)/PL
5CONTINUE
CALL STE(A,B,N)
10CONTINUE
DO 12 I=1,8
CALL STE(B,B,N)
BMAX=B(1,1)
DO 18 IU=2,N
IF(B(IU,IU).GT.BMAX)BMAX=B(IU,IU)
18CONTINUE
DO 19 IU=1,N
DO 19 JU=1,N
19B(IU,JU)=B(IU,JU)/BMAX
12CONTINUE
SP(1)=0.
SP(2)=0.
DO 14 I=1,N
14SP(1)=SP(1)+B(I,I)
CALL STE(A,B,N)
DO 15 I=1,N
15SP(2)=SP(2)+B(I,I)
X1=SP(2)/SP(1)
PK(1)=X1+P(1)
DO 16 I=2,ILL
16PK(I)=X1*PK(I-1)+P(1)
21FORMAT(/X,4E20.4)
Z=PK(1)
ZZ=Z*Z*Z/3.
PE=-Z*Z/3.+PK(2)
QM=2./9.*ZZ-PK(1)*PK(2)/3.+PK(3)
Q=PE*PE*PE/27.+QM*QM/4.
AK=ZZ*(Z)**(1./3.)
BK=-QM/2.-SQRT(Q)

```

```

Z=ABS(BK)
ZZ=BK/Z
BK=ZZ*(Z)**(1./3.)
X2=AK+BK
D23=-X2/2.-PK(1)/3.
X2=X2-PK(1)/3.
DIM=(AK-BK)/2.*SQRT(3.)
PRINT17,X1,X2,D23,DIM

17FORMAT(' СОБСТВЕННЫЕ
      &ЧИСЛА ',
      &' X1=',F8.3,' X2=',F8.3/
      &' REAL X3,X4=',F8.3,
      &' IMAGE X3,X4=',F8.3)
RA=X1**4+P(1)*X1**3+
      &P(2)*X1**2+P(3)*X1+P(4)
RB=X2**4+P(1)*X2**3+
      &P(2)*X2**2+P(3)*X2+P(4)
RC=CMPLX(D23,DIM)
RE=RC**4+P(1)*RD**3+
      &P(2)*RC**2+P(3)*RC+P(4)
RD=CMPLX(D23,-DIM)
RF=RD**4+P(1)*RD**3+
      &P(2)*RD**2+P(3)*RD+P(4)
PRINT21,RA,RB,RE,RF
STOP
END
C
C
C
SUBROUTINE STE(A,B,N)
REAL A(4,4),B(4,4),C(4,4)
DO 1 I=1,N
DO 1 J=1,N
C(I,J)=0.
DO 1 K=1,N
1C(I,J)=C(I,J)+B(I,K)*A(K,J)
DO 2 I=1,N
DO 2 J=1,N
2B(I,J)=C(I,J)
RETURN
END

```

```

9.1  4.2  3.2  6.1
3.0  5.3  1.7  4.9
2.6  4.7  9.4  3.5
4.0  1.6  1.0  6.2

```

СОБСТВЕННЫЕ ЧИСЛА

```

X1=17.476 X2=6.787
REAL X3,X4=2.868
      IMAGE X3,X4=0.689

```

```

-0.927E-02  0.488E-03  0.488E-03
-0.122E-03  0.488E-03  0.122E-03

```

Рис. 8.4.

ЛОЖИМ, ЧТО

$$g(x) > 0 \text{ и } g(x)r(x) \in C^k[a, b], \quad 0 \leq k \leq 2.$$

Приближенное решение будем искать с помощью параболических сплайнов

$$S_2(x) = \sum_{k=-1}^{n+1} d_k B_k(x),$$

где

$$B_0(x+a) = \frac{h^{-2}}{2} \sum_{k=0}^3 (-1)^k C_3^k(x + \frac{3h}{2} - kh)_+^2;$$

$$z_+^m = [\max(0, z)]^m;$$

$$B_k(x+a) = B_0(x+a - kh).$$

Потребуем, чтобы сплайн $S_2(x)$ удовлетворял условиям

$$S_2''(x_i) - g(x_i) S_2(x_i) = r(x_i), \quad i = 0, \dots, n; \quad (8.5)$$

$$S_2(a) = y_a, \quad S_2(b) = y_b.$$

Так как

$$S_2(x_i) = \frac{d_{i-1} + 6d_i + d_{i+1}}{8}, \quad i = 0, \dots, n; \quad (8.6)$$

$$S_2''(x_0) = g(x_0) S_2(x_0) + r(x_0) = y''(x_0);$$

$$S_2''(x_n) = g(x_n) S_2(x_n) + r(x_n) = y''(x_n);$$

$$S_2''(x_i) = \frac{d_{i-1} - 2d_i + d_{i+1}}{h^2}, \quad i = 1, \dots, n-1,$$

то, учитывая равенство (8.6), соотношения (8.5) можно записать следующим образом:

$$\left. \begin{aligned} d_{-1} + 6d_0 + d_1 &= 8y_a; \\ d_{-1} - 2d_0 + d_1 &= y''(x_0) h^2; \\ &\dots \dots \dots \\ l(d_i) &= \left(\frac{1}{h^2} - \frac{1}{8} g_i\right) d_{i-1} - \left(\frac{2}{h^2} + \frac{3}{4} g_i\right) d_i + \\ &\quad + \left(\frac{1}{h^2} - \frac{1}{8} g_i\right) d_{i+1} = r_i, \\ &\dots \dots \dots \\ d_{n-1} - 2d_n + d_{n+1} &= y''(x_n) h^2; \\ d_{n-1} + 6d_n + d_{n+1} &= 8y_b, \end{aligned} \right\} \quad (8.7)$$

где

$$g_i = g(x_i) \text{ и } r_i = r(x_i), \quad i = 1, \dots, n-1.$$

Вычитая из первого уравнения второе и из последнего предпоследнее, преобразуем систему (8.7) к виду

$$\left. \begin{aligned} d_0 &= y_a - \frac{1}{8} h^2 y''(x_0) = y_a - \frac{1}{8} h^2 [g(a) y_a + r(a)]; \\ l(d_i) &= r_i, \quad i = 1, \dots, n-1; \\ d_n &= y_b - \frac{1}{8} h^2 y''(x_n) = y_b - \frac{1}{8} h^2 [g(b) y_b + r(b)]. \end{aligned} \right\} (8.8)$$

Из системы (8.8) находим $d_i, i = 0, \dots, n$, а d_{-1} и d_{n+1} вычисляем из первого и последнего уравнений системы (8.7).

При этом вместо системы (8.7) можно решать более простую систему уравнений

$$\left. \begin{aligned} d_0 &= y_a; \\ l(d_i) &= \frac{1}{h^2} d_{i-1} - \left(\frac{2}{h^2} + g_i \right) d_i + \frac{1}{h^2} d_{i+1} = r_i; \\ d_n &= y_b, \quad i = 1, \dots, n-1. \end{aligned} \right\} (8.9)$$

Решаем трехдиагональную систему (8.9) методом прогонки, который в данном случае состоит из следующей последовательности вычислений.

Сначала находим:

$$E_m = \frac{A}{B - A E_{m-1}}, \quad m = 1, \dots, n-1;$$

$$F_m = E_m F_{m-1}, \quad \text{где } A = \frac{1}{h^2}, \quad B = \frac{2}{h^2} + 100.$$

Для исходной задачи

$$E_0 = 0, \quad F_0 = d_0,$$

d_i получаем по формуле

$$d_i = E_i d_{i+1} + F_i, \quad i = n-1, n-2, \dots, 1.$$

Затем находим решение уравнения по формуле (8.6).

Результаты, представленные на рис. 8.5, можно сравнить с точным решением и с решением с помощью кубических сплайнов.

Пример 4. Составить программу быстрого преобразования Фурье для проведения гармонического анализа на отрезке $[0, 2\pi]$ функции

$$y = e^{5 \sin x} + e^{1.5 \cos x}.$$

PROGRAM DJ	1.0000
REAL E(21),F(21),Q(21)	0.4299
M=11	0.1646
H=.1	0.0640
DO 10 ITE=1,2	0.0274
E(1)=0.	0.0183
F(1)=1.	0.0274
Q(1)=1.	0.0640
Q(M)=1.	0.1646
IM=M-1	0.4299
AM=1./H/H	1.0000
BM=AM+AM+100.	РЕШЕНИЕ ДИФФ. УРАВНЕНИЯ
DO 1 I=2,M	1.0000
E(I)=AM/(BM-AM*E(I-1))	0.6287
F(I)=E(I)*F(I-1)	0.3834
DO 2 I=1,IM	0.2338
MI=M-1	0.1428
2Q(MI)=E(MI)*Q(MI+1)+F(MI)	0.0874
PRINT 3	0.0539
3FORMAT(' РЕШЕНИЕ ДИФФ.	0.0339
&УРАВНЕНИЯ')	0.0224
PRINT 5,Q(M)	0.0164
5FORMAT(1X,F20.4)	0.0146
M=21	0.0164
10H=.05	0.0224
STOP	0.0339
END	0.0539
	0.0874
	0.1428
	0.2338
	0.3834
	0.6287
	1.0000

Рис. 8.5.

Периодическая функция может быть разложена в ряд Фурье

$$f(x) = \sum_{q=-\infty}^{\infty} a_q \exp(2\pi i q x),$$

причем

$$\sum_{q=-\infty}^{\infty} |a_q| < \infty.$$

Если функция $f(x)$ рассматривается лишь на сетке из точек $x_l = l/N$, где l, N — целые, N — фиксирована и $f(x_l) = f_l$, то суммирование будет производиться в конечных пределах.

Пусть $q_2 - q_1 = kN$, где k — целое; тогда $q_2 x_l - q_1 x_l = kN x_l = k_l$ — целое и, следовательно,

$$\exp(2\pi i q_1 x) = \exp(2\pi i q_2 x)$$

в узлах сетки. После приведения подобных членов получим

$$f_l = \sum_{q=0}^{N-1} A_q \exp(2\pi i q x_l), \quad (8.10)$$

где

$$A_q = \sum_{s=-\infty}^{\infty} a_{q+sN}.$$

Функции $g_q = \exp(2\pi i q x_l)$ при $0 \leq q \leq N$ образуют ортонормированную систему относительно следующим образом определенного скалярного произведения для функций на сетке:

$$(f, g) = \frac{1}{N} \sum_{l=0}^{N-1} f_l g_l,$$

где

$$g_l = \exp(-2\pi i l x_l).$$

Умножая (8.10) скалярно на g_j , получим равенство

$$A_j = (f, g_j) = \frac{1}{N} \sum_{l=0}^{N-1} f_l \exp(-2\pi i j x_l). \quad (8.11)$$

Соотношения (8.10) и (8.11) соответствуют дискретному ряду Фурье и дискретным коэффициентам Фурье.

Осуществление прямого и обратного преобразований Фурье

$$(f_0, \dots, f_{N-1}) \Leftrightarrow (A_0, \dots, A_{N-1})$$

является составной частью решения многих задач. Непосредственное осуществление этих преобразований по формулам (8.10) и (8.11) требует N^2 арифметических операций.

Идея построения алгоритмов быстрого преобразования Фурье основана на том, что при N составном среди

слагаемых правой части (8.11) можно выделить группы, которые входят в выражения для различных коэффициентов.

Пусть $N = p_1 p_2$, $p_1, p_2 \neq 1$.

Подставим q, j , лежащие в пределах $0 \leq q, j \leq N$, в виде

$$q = q_1 + p_1 q_2; \quad j = j_2 + p_2 j_1,$$

где

$$0 \leq q_1, j_1 < p_1; \quad 0 \leq q_2, j_2 < p_2.$$

В результате получаем следующие соотношения:

$$\begin{aligned} A_q &= A(q_1, q_2) = \frac{1}{N} \sum_{j=0}^{N-1} f_j \exp(-2\pi i \frac{qj}{N}) = \\ &= \frac{1}{N} \sum_{j_1=0}^{p_1-1} \sum_{j_2=0}^{p_2-1} \left\{ f_{j_2+p_2 j_1} \exp \left[-2\pi i \frac{(q_1 + p_1 q_2)(j_2 + p_2 j_1)}{p_1 p_2} \right] \right\}. \end{aligned}$$

Из равенства

$$\frac{(q_1 + p_1 q_2)(j_2 + p_2 j_1)}{p_1 p_2} = q_2 j_1 + \frac{q j_2}{N} + \frac{j_1 q_1}{p_1}$$

и предыдущего соотношения получаем

$$A(q_1, q_2) = \frac{1}{p_2} \sum_{j_2=0}^{p_2-1} A^{(1)}(q_1, j_2) \exp \left[-2\pi i \left(q_2 j_1 + \frac{q j_2}{N} \right) \right],$$

где

$$A^{(1)}(q_1, j_2) = \frac{1}{p_1} \sum_{j_1=0}^{p_1-1} \left[f_{j_2+p_2 j_1} \exp \left(-2\pi i \frac{j_1 q_1}{p_1} \right) \right].$$

Непосредственное вычисление всех $A^{(1)}(q_1, j_2)$ требует $p_1^2 p_2$ арифметических операций, а последующее вычисление $A(q_1, q_2)$ — еще $p_1 p_2^2$ операций. Следовательно, при $p_1, p_2 = N$ общее число операций будет равно $N^{3/2}$. Точно так же при $N = p_1, \dots, p_r$ строится алгоритм вычисления совокупности значений A_q , для которого общее число операций не превосходит $cN(p_1 + \dots + p_r)$, здесь c — постоянная.

В программе используются наиболее употребительные расчетные формулы для случая, когда $p_1 = \dots = p_r = 2$.

При их выводе используются следующие представления чисел q и j :

$$q = \sum_{k=1}^r q_k 2^{k-1}, \quad j = \sum_{m=1}^r j_{r+1-m} 2^{m-1},$$

где

$$q_k, j_m = 0, 1.$$

В результате получаем соотношения

$$\begin{aligned} & A^{(1)}(q_1, \dots, q_m; j_{m+1}, \dots, j_r) = \\ & = \frac{1}{2} \sum_{l_m=0}^1 \exp\left(-2\pi i j_m 2^{-m} \sum_{k=1}^m q_k 2^{k-1}\right) \times \\ & \times A^{(m-1)}(q_1, \dots, q_{m-1}; j_m, \dots, j_r), \quad m = 1, \dots, r, \end{aligned} \quad (8.12)$$

где

$$\begin{aligned} A^{(0)}(j_1, \dots, j_r) &= j_{r+1} 2^r + \dots + j_1 \cdot 2^0; \\ A^{(r)}(q_1, \dots, q_r) &= A(q_1, \dots, q_r). \end{aligned} \quad (8.13)$$

Переход от каждой совокупности $A^{(m-1)}$ требует N арифметических и логических операций, а общее число операций на r шагах

$$N_r = N \log_2 N.$$

Для вычисления значений номеров элементов нулевого слоя в соответствии с (8.13) необходимо найти числа, соответствующие инвертированным номерам элементов (табл. 8.1).

Таблица 8.1. Двоично-инверсная перестановка

Десятичное число	Двоичный код	Двоично-инверсный код	Десятичное число после перестановки
0	0 0 0	0 0 0	0
1	0 0 1	1 0 0	4
2	0 1 0	0 1 0	2
3	0 1 1	1 1 0	6
4	1 0 0	0 0 1	1
5	1 0 1	1 0 1	5
6	1 1 0	0 1 1	3
7	1 1 1	1 1 1	7

Сначала по $j = j_1 + j_2 2 + \dots + j_r 2^{r-1}$ находятся числа $k = j_1 2^{r-1} + \dots + j_2 2^{r-2} + \dots + j_r$ и записываются значения $f_k BA_j$. (В программе K хранятся в массиве M). В цикле с меткой 5 производятся рекуррентные вычисления по формуле (8.12). (В программе номеру слоя m соответствует переменная l .) Переход между 1, 2 и 3 слоями показан на рис. 8.6.

Стрелки \dashrightarrow соответствуют $j_m = 0$ и единичному множителю (в формуле (8.12) $\exp = 0$). Число

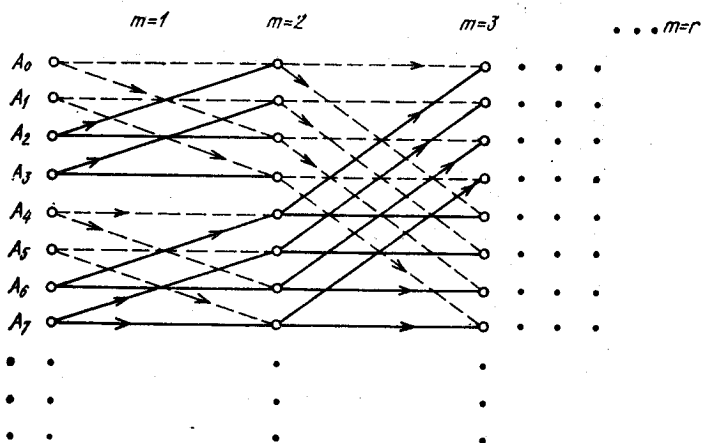


Рис. 8.6.

различных значений экспонент на каждом слое равно 2^m . Вычисления повторяются в цикле с меткой 4 и переменными разрядами (q_1, \dots, q_m).

Циклу по этим числам соответствует в программе цикл с меткой 10 и переменной K .

Программа и результаты вычислений представлены на рис. 8.7.

Пример 5. Проанализировать, используя матричный критерий Зубова, устойчивость системы автоматического управления, структурная схема которой представлена на рис. 8.8. Определить границы устойчивости в зависимости от значения K .

Замкнутая автоматическая система управления состоит из четырех инерционных звеньев с одинаковыми постоянными времени, равными 1.

```

PROGRAM COMPL
  INTEGER JJ(20),M(20)
  COMPLEX A(512),EX(256)
  COMPLEX AX,X,DX,R

```

```

СФУНКЦИЯ
F(Z)=EXP(5.*SIN(2.*PI*Z))
      *+EXP(1.5*COS(2.*PI*Z))
PI=3.14159265
IR=3
N=2**IR
AN=N
DZ=1./AN

```

```

С НОМЕРА НАЧАЛЬНОГО РАСПРЕДЕЛЕНИЯ

```

```

M(1)=0
IRR=IR-1
DO 13 LL=1,IR
  13JJ(LL)=0
DO 19 L=2,N
  K=0
  14 K=K+1
  IF(JJ(K).EQ.0)GOTO 15
  JJ(K)=0
  GOTO14
  15JJ(K)=1
  IP=2
  M(L)=JJ(IR)
DO 19 I=1,IRR
  IF(JJ(IR-I).EQ.1)M(L)=M(L)+IP
  19IP=IP+IP
С НАЧАЛЬНОЕ РАСПРЕДЕЛЕНИЕ
С ЗНАЧЕНИЙ
Z1=0.
DO 7 I=1,N
  L=M(I)+1
  A(L)=F(Z1)
  7Z1=Z1+DZ
СПРЕОБРАЗОВАНИЕ
EX(1)=(1.,0.)
ISH=1
AN=-PI
DO 5 I=1,IR
  IS=ISH+ISH
  DX= SMPL(0.,AN)
  X=(0.,0.)
DO 4 K=2,IS
  X=X+DX
  4EX(K)=CEXP(X)
DO 3 J=1,N,IS
  JU=J+ISH-1
  L=0
DO 10 K=J,JU
  L=L+1
  R=A(K)

```

```

A(K)=A(K)+A(K+ISH)*EX(L)
A(K+ISH)=A(K+ISH)*EX(L+ISH)+R
100CONTINUE
3CONTINUE
ISH=IS
AN=AN/2.
100FORMAT(1X,8E15.6)
5CONTINUE
DO 8 LI=1,N
8A(LI)=A(LI)*DZ
PRINT 121
121FORMAT(' ККОМПЛ. АМПЛИТУДЫ ',
&' ГАРМОНИК'/1X,
&60(' _'))
PRINT 120,(I,A(I),I=1,N)
120FORMAT(' I=',I3,' RE A='
&E11.4,' IM A=',E11.4)
STOP
END

```

КОМПЛ. АМПЛИТУДЫ ГАРМОНИК

```

I=1 RE A= 0.29E+02 IM A= 0.00E+00
I=2 RE A= 0.98E+00 IM A=-0.24E+02
I=3 RE A=-0.17E+02 IM A=-0.47E-06
I=4 RE A= 0.82E-01 IM A= 0.12E+02
I=5 RE A= 0.10E+02 IM A= 0.00E+00
I=6 RE A= 0.82E-01 IM A=-0.12E+02
I=7 RE A=-0.17E+02 IM A= 0.47E-06

```

Рис. 8.7.

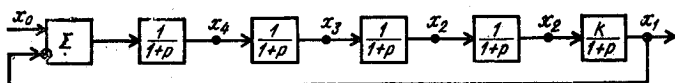


Рис. 8.8.

Для исследования устойчивости тривиального решения запишем систему уравнений относительно координат x_1, \dots, x_4 :

$$\left. \begin{aligned}
 \frac{\partial x_1}{\partial t} &= kx_2 - x_1; \\
 \frac{\partial x_2}{\partial t} &= x_3 - x_2; \\
 \frac{\partial x_3}{\partial t} &= x_4 - x_3; \\
 \frac{\partial x_4}{\partial t} &= x_1 - x_4,
 \end{aligned} \right\} \quad (8.14)$$

где x_1, x_2, x_3, x_4 — координаты состояния, характеризующие работу системы. В матричной записи система дифференциальных уравнений вида Коши (8.14) будет иметь вид

$$x_1 = Ax, \quad (8.15)$$

где элементы матрицы A a_{ij} , $i, j = 1, 2, \dots, n$ — постоянные числа, зависящие от параметров системы.

В нашем случае

$$A = \begin{vmatrix} -1 & k & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 1 \end{vmatrix}.$$

Дальнейшие рассуждения состоят в следующем. Пусть на плоскости комплексного переменного λ задана некоторая область G . Требуется найти все необходимые и достаточные условия того, чтобы все собственные значения λ_i исходной матрицы коэффициентов A для системы уравнений (8.15) располагались внутри области G . Обозначим A_G — совокупность всех матриц порядка n , все собственные значения которых располагаются внутри области G ; B_ρ — совокупность всех матриц порядка n , все собственные значения которых лежат внутри единичного круга с центром в начале координат на плоскости комплексного переменного ρ .

Предположим, что существует оператор \mathbf{H} , устанавливающий взаимнооднозначное соответствие двух множеств A_G и B_ρ . Тогда если $A \in A_G$, то оператор \mathbf{H} от матрицы A есть такая матрица порядка n , что $\mathbf{H}(A) \in B_\rho$, и, наоборот, если B — матрица из семейства B_ρ , то $\mathbf{H}^{-1}(B) \in A_G$. Чтобы матрица B принадлежала области B_ρ , необходимо и достаточно, чтобы выполнялось условие

$$\lim_{k \rightarrow \infty} B^k = 0,$$

для чего необходимо и достаточно, чтобы $|\rho_i| < 1$ для всех собственных значений матрицы B .

Область A_G представляет собой левую полуплоскость

$$\operatorname{Re} \lambda < 0.$$

Матрица B строится с использованием дробно-линейного преобразования

$$\lambda = \frac{\rho + 1}{\rho - 1},$$

обладающего тем свойством, что мнимая ось, рассматриваемая как окружность бесконечного радиуса, переводится в окружность единичного радиуса с центром в начале координат.

Подставив λ через ρ в характеристическое уравнение

$$|A - \lambda E| = 0,$$

получаем

$$\left| A - \frac{\rho + 1}{\rho - 1} E \right| = 0,$$

или

$$|B - \rho E| = 0,$$

где

$$B = E + 2(A - E)^{-1}.$$

На основании этого (чтобы исключить переход от матрицы к характеристическому многочлену) В. И. Зубовым был предложен матричный критерий устойчивости. Наиболее существенным моментом при использовании этого критерия является окончание процесса возведения в степень. Последовательность вычислений следующая:

- 1) вводят размер матрицы N , предельное значение $L_{\text{пред}}$ (например, 20), матрицу коэффициентов A , переменную $L = 1$;
- 2) вычисляют матрицу $A - E$;
- 3) обращают матрицу $(A - E)^{-1}$;
- 4) вычисляют матрицу $B = E + 2(A - E)^{-1}$, $\epsilon_{ps} = 1/n$;
- 5) проверяют справедливость неравенства $|B_{ij}| \leq \epsilon_{ps}$ для всех $i, j = 1, \dots, n$.

Если неравенство выполняется, то производят останов программы и сигнализируют об устойчивости системы (если не выполняется, то переходят к п. 6).

6) вычисляют матрицу B по формуле $B = B \times B$;

7) увеличивают переменную $L: L = L + 1$;

8) проверяют условие $L < L_{\text{пред}}$. Если справедливо, то переходят в п. 5, если несправедливо, т. е. $k > k_{\text{пред}}$, то принимают решение о неустойчивости системы. Следует учесть, что при больших L в матрице B^{2L} из-за накопления ошибок многие коэффициенты вычислены с большой погрешностью, даже при большой точности ЭВМ.

Для завершения рассмотренного алгоритма введем оператор сравнения каждого элемента $|B_{ij}|$ с очень

большим числом, например 10^{10} , и разместим его перед п. 5.

Если $|B_{ij}| > 10^{10}$, то прекращается выполнение алгоритма. При последовательном возведении в квадрат эти элементы будут увеличиваться.

В преобразованную матрицу коэффициентов B входит операция построения обратной матрицы $(A - E)^{-1}$, которая часто связана с вычислительными сложностями. Один из приемов, устраняющих операцию обращения матрицы, состоит в представлении матричным степенным рядом

$$B \approx E - 2(E + A + A^2 + \dots + A^n).$$

Для сходимости ряда достаточно, чтобы любая из норм матрицы A была меньше единицы.

Другой метод состоит в аппроксимации области разложения всех собственных значений матрицы A кругом подходящего радиуса с центром в точке $(-R, 0)$.

Критерий формулируется следующим образом: для того чтобы все собственные значения находились внутри заданного круга радиуса R с центром в точке $(-R, 0)$, необходимо и достаточно, чтобы выполнялось условие

$$\left[E + \frac{A}{R} \right]^k \rightarrow 0 \text{ при } k \rightarrow \infty.$$

Большие значения радиусов удлинняют время решения задачи.

Можно найти обратную матрицу, применяя существующие хорошо разработанные программы. Однако применительно к задачам анализа и синтеза систем автоматического управления следует учитывать возможные осложнения. Многие из эффективных способов обращения неудобны для решения задачи определения границ области устойчивости в пространстве допустимых значений параметров системы управления. Метод Леверье, который при порядке матрицы $n = 6 - 9$ дает вполне удовлетворительные результаты, состоит в следующей последовательности вычислений:

$$\begin{aligned} A_1 &= A; S_p A_1 = q_1; B_1 = A_1 - q_1 E; \\ A_2 &= AB_1; \frac{S_p A_2}{2} = q_2; B_2 = A_2 - q_2 E; \\ &\dots \dots \dots \\ A_n &= AB_{n-1}; \frac{S_p A_n}{n} = q_n; B_n = A_n - q_n E, \end{aligned}$$

где $S_p A$ — след матрицы A и если матрица A — неособенная, то справедливо следующее утверждение:

$$A^{-1} = B_{n-1}/q_n.$$

В программе, представленной на рис. 8.9, матрица $(A - E)^{-1}$ рассчитывается по следующему ее аналитическому выражению:

$$(A - E)^{-1} = \frac{1}{16+k} \begin{vmatrix} -8 & -4k & -2k & -k \\ 1 & -8 & -4 & -2 \\ 2 & k & -8 & -4 \\ 4 & 2k & k & -8 \end{vmatrix}.$$

При $K = 2$ система не устойчива.

Пример 6. Идентифицировать передаточную функцию динамического измерительного преобразователя по методу Симою.

Исходные данные. Массив данных (точки переходной характеристики $h(t)$):

0,000; 1,106; 1,967; 2,638; 3,161; 3,567; 3,884; 4,131; 4,323; 4,473; 4,590; 4,680; 4,751; 4,806; 4,849; 4,882; 4,908; 4,929; 4,944; 4,957; 4,966 (количество точек $m = 21$);

шаг квантования $\Delta t = 0,5$;

коэффициент передачи $K(0) = 5$;

порядок аппроксимации $n = 4$.

Если на вход линейного объекта подано воздействие — единичный скачок, то на его выходе при нулевых начальных условиях переходная характеристика $h(t)$ определяется выражением

$$h(t) = L^{-1}(W(p)/p),$$

где L^{-1} обозначает обратное преобразование Лапласа; $W(p)$ — передаточная функция; $1/P$ — изображение единичного скачка.

Наиболее удобной аппроксимацией передаточной функции при $\lambda(0) = 0$ является

$$W(p) = \frac{k}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + 1}.$$

Учитывая, что погрешности вычисления коэффициентов a_i полинома возрастают с увеличением i , в алгоритме решения задачи порядок полинома ограничен $n \leq 4$. Кроме того, излагаемый алгоритм предназначен для идентификации объектов без интегрирующих звеньев,

```

PROGRAM MATR
REAL A(20,20),B(20,20)
N=4
EPS=1./N
BIG=10.**10
DO 1 I=1,N
  1B(I,I)=-8.
  B(2,1)=1.
  B(3,1)=2.
  B(4,1)=4.
  B(2,3)=-4.
  B(2,4)=-2
  B(3,4)=-4
  KMAX=20
  LPRED=20
  PRINT 105
  105FORMAT('      СТЕПЕНЬ
            &МАТРИЦЫ РАВНА 2**L')
  DO 2 K=1,KMAX
  PRINT 106,K
  106FORMAT('/'/'      КОЭФИЦИЕНТ
            &K=',I2/1X,60(''-'))
  B(1,2)=-4.*K
  B(1,3)=-2.*K
  B(1,4)=-K
  B(3,2)=K
  B(4,2)=2.*K
  B(4,3)=K
  DO 3 I=1,N
  DO 3 J=1,N
  3B(I,J)=B(I,J)/(16.+K)*2
  DO 6 I=1,N
  6B(I,I)=B(I,I)+1.
  DO 4 L=1,LPRED
  PRINT 102,L
  102FORMAT('      МАТРИЦА В ПРИ L=',I2)
  DO 7 I8=1,N
  7PRINT 101,(B(I8,J8),J8=1,N)
  IPR11=0
  IPR12=0
  DO 5 I=1,N
  DO 5 J=1,N
  AB=ABS(B(I,J))
  IF(AB.LT.BIG)IPR11=1
  IF(AB.GT.EPS)IPR12=1
  5CONTINUE
  IF(IPR12.EQ.0)GO TO 1000
  IF(IPR11.EQ.0)GO TO 999
  BIG=10.**10
  CALL QUADR(B,N)
  4CONTINUE

```

```

PRINT 104,LPRED
104FORMAT(' СТЕПЕНЬ СТЕПЕНИ
&МАТРИЦЫ В БОЛЬШЕ 'I3)
GO TO 1001.
1000PRINT 100,L,K
PRINT 107
107FORMAT(' СИСТЕМА УСТОЙЧИВА')
2CONTINUE
GO TO 1001
999PRINT 100,L,K
PRINT 103,BIG
103FORMAT(' ЭЛЕМЕНТЫ МАТРИЦЫ Б
&БОЛЬШЕ ', E11.2)
100FORMAT(X,2I3)
101FORMAT(X,10G9.2)
1001CONTINUE
STOP
END
C
C
C
C
SUBROUTINE QUADR(B,N)
REAL A(20,20),B(20,20)
DO 1 I=1,N
DO 1 J=1,N
A(I,J)=0.
DO 1 K=1,N
1A(I,J)=A(I,J)+B(I,K)*B(K,J)
DO 2 I=1,N
DO 2 J=1,N
2B(I,J)=A(I,J)
RETURN
END

```

СТЕПЕНЬ МАТРИЦЫ РАВНА 2**N

КОЭФИЦИЕНТ K= 1

МАТРИЦА В ПРИ L= 1

0.59E-01-0.47	-0.24	-0.12	
0.12	0.59E-01-0.47	-0.24	
0.24	0.12	0.59E-01-0.47	
0.47	0.24	0.12	0.59E-01

МАТРИЦА В ПРИ L= 2

-0.16	-0.11	0.18	0.21
-0.21	-0.16	-0.11	0.18
0.18	-0.21	-0.16	-0.11
0.11	-0.18	-0.21	-0.16

2 1
СИСТЕМА УСТОЙЧИВА

КОЭФФИЦИЕНТ $K = 2$

МАТРИЦА В ПРИ $L = 1$

0.98	-0.89	-0.44	-0.22
-0.23E-01	0.98	-0.12E-01	0.20E-01
-0.20E-01	0.22	0.98	-0.12E-01
0.12E-01	0.44	0.22	0.98

МАТРИЦА В ПРИ $L = 2$

0.99	-1.9	-0.91	-0.45
-0.45E-01	0.99	-0.95E-02	0.45E-01
-0.45E-01	0.45	0.97	-0.15E-01
0.95E-02	0.91	0.43	0.97

МАТРИЦА В ПРИ $L = 3$

1.1	-4.7	-2.0	-0.95
-0.88E-01	1.1	0.41E-01	0.11
-0.11	0.95	0.97	0.10E-01
-0.41E-01	2.0	0.81	0.97

МАТРИЦА В ПРИ $L = 4$

1.9	-14.	-5.0	-2.5
-0.20	1.9	0.34	0.31
-0.31	2.5	1.2	0.22
-0.34	5.0	1.7	1.2

МАТРИЦА В ПРИ $L = 5$

8.8	-78.	-25.	-13.
-0.98	8.8	2.6	1.5
-1.5	13.	4.2	2.1
-2.6	25.	7.5	4.2

МАТРИЦА В ПРИ $L = 6$

0.22E+03	-0.20E+04	-0.62E+03	-0.34E+03
-25.	0.22E+03	69.	38.
-38.	0.34E+03	0.10E+03	57.
-69.	0.62E+03	0.19E+03	0.10E+03

МАТРИЦА В ПРИ $L = 7$

0.15E+06	-0.13E+07	-0.41E+06	-0.22E+06
-0.17E+05	0.15E+06	0.46E+05	0.25E+05
-0.25E+05	0.22E+06	0.69E+05	0.38E+05
-0.46E+05	0.41E+06	0.13E+06	0.69E+05

МАТРИЦА В ПРИ $L = 8$

0.64E+11	-0.58E+12	-0.18E+12	-0.97E+11
-0.72E+10	0.64E+11	0.20E+11	0.11E+11
-0.11E+11	0.97E+11	0.30E+11	0.16E+11
-0.20E+11	0.18E+12	0.55E+11	0.30E+11

МАТРИЦА В ПРИ $L = 9$

0.12E+23	-0.11E+24	-0.34E+23	-0.18E+23
-0.14E+22	0.12E+23	0.37E+22	0.21E+22
-0.21E+22	0.18E+23	0.57E+22	0.31E+22
-0.37E+22	0.34E+23	0.10E+23	0.57E+22

9 2
ЭЛЕМЕНТЫ МАТРИЦЫ В БОЛЬШЕ 0.10E+11

Рис. 8.9.

т. е. объектов, обладающих свойством самовыравнивания.

Коэффициент передачи объекта K определяет установившееся значение переходной характеристики, т. е. $\lim_{t \rightarrow \infty} h(t) = K$.

Коэффициенты a_i передаточных функций находятся в соответствии с методом Симоу по формуле

$$a_i = S_i/k,$$

где S_i — площадь i -го порядка под кривой $k - h(t)$. Площадь S_1 определяется следующим образом:

$$S_1 = \int_0^{\infty} (k - h(t)) dt. \quad (8.16)$$

Далее, изменяя масштаб времени $\tau = t/S_1$, определяем

$$S_i = S_1^i \int_0^{\infty} x(\tau) f(i, \tau) d\tau, \quad (8.17)$$

где $x(\tau) = K - h(\tau)$; $f(i, \tau)$ функция от аргумента τ и номера вычисляемой площади i .

Приведем развернутые формулы для S_2, S_3, S_4 :

$$S_2 = S_1^2 \int_0^{\infty} (K - h(\tau))(1 - \tau) d\tau;$$

$$S_3 = S_1^3 \int_0^{\infty} (K - h(\tau))(1 - 2\tau + \tau^2/2) d\tau;$$

$$S_4 = S_1^4 \int_0^{\infty} (K - h(\tau))(S_3/S_1^3 - S_2/S_1 \tau + \tau^2/2 - \tau^3/6) d\tau.$$

Если $h(0) \neq 0$, то используется аппроксимация передаточной функции вида

$$W(p) = \frac{K(1 + b_n p^n)}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + 1},$$

где коэффициенты b_n определяются как

$$b_n = h(0) \frac{a_n}{K}.$$

Программа и результаты решения задачи представлены на рис. 8.10, интегралы (8.16) и (8.17) определены по методу Симпсона.

```

PROGRAM SIMOJU
REAL K
REAL H(21)
REAL Y(21)
REAL Z(21)
DATA H/0.000, 1.106, 1.967,
&      2.638, 3.161, 3.567, 3.884,
&      4.131, 4.323, 4.473,
&      4.590, 4.680, 4.751, 4.806,
&      4.849, 4.882, 4.908,
&      4.929, 4.944, 4.957, 4.966/
H(T)=5.*(1-EXP(-T/2.))
DT=0.5
K=5.
CPOACHET S1
DO 10 I=1, 21
Y(I)=K-H(I)
10CONTINUE
CALL QSF(DT, Y, Z, 21)
S1=Z(21)
CPOACHET S2
DO 20 I=1, 21
IT=REAL(I-1)/S1+0.5
IF(IT.LT.1)IT=1
Y(I)=S1**2*(K-H(IT))*
&      (1-DT*IT/S1)
20CONTINUE
CALL QSF(DT/S1, Y, Z, 21)
S2=Z(21)
DO 30 I=1, 21
IT=REAL(I-1)/S1+0.5
IF(IT.LT.1)IT=1
Y(I)=S1**3*(K-H(IT))*
&(1-2.*IT*DT+(IT*
&DT/S1)**2/2.)
30CONTINUE
CALL QSF(DT/S1, Y, Z, 21)
S3=Z(21)
A1=S1/K
A2=S2/K
A3=S3/K
TYPE*, ' A1=', A1
TYPE*, ' A2=', A2
TYPE*, ' A3=', A3
STOP
END

```

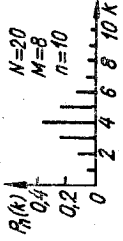
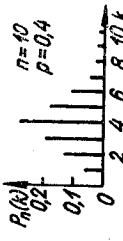
```


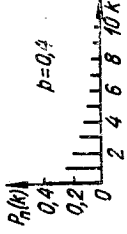
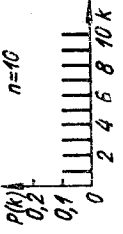
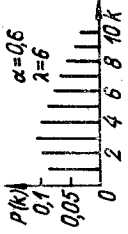
A1=1.9868
A2=1.590E-03
A3=1.265E-04

```

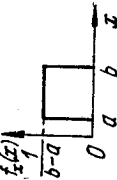
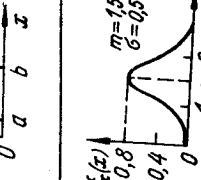
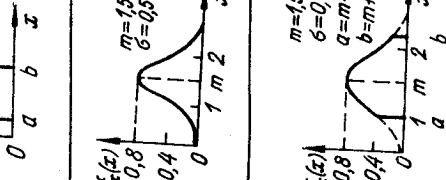
Рис. 8.10.

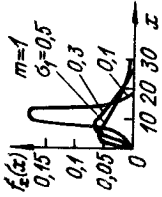
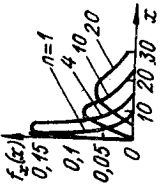
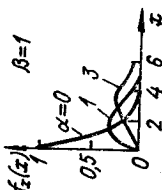
Законы распределения дискретной случайной величины

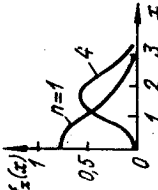
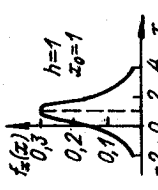
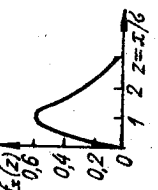
Закон распределения	Область изменения значений случайной величины	Аналитическое выражение закона распределения	Определяющие параметры	График закона распределения
Гипергеометрический	$k = 0, 1, 2, \dots, \min(M, n)$	$P_n(k) = \frac{C_M^k C_{N-M}^{n-k}}{C_N^n}$	N, M, n	
Биномиальный (Бернулли)	$k = 0, 1, 2, \dots, n$	$P_n(k) = C_n^k p^k (1-p)^{n-k}$	n, p	
Полиномиальный	$k_1 = 0, 1, 2, \dots, n_1$ $k_2 = 0, 1, 2, \dots, n_2$ \dots $k_m = 0, 1, 2, \dots, n_m$ $\sum_{i=1}^m k_i = n$	$P_n(k_1, k_2, \dots, k_m) = \frac{n!}{k_1! k_2! \dots k_m!} p_1^{k_1} \dots p_m^{k_m}$ $p_1 + p_2 + \dots + p_m = 1$	n и любые $m-1$ величин из p_1, p_2, \dots, p_m	


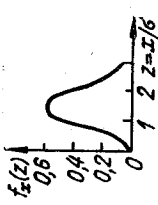
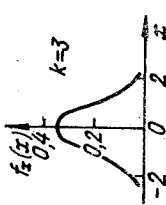
Пуассона	$k = 0, 1, 2, \dots$	$P_n(k) = \frac{\lambda^k}{k!} e^{-\lambda}$	λ	
Геометрический (Фарри)	$k = 0, 1, 2, \dots$	$P_n(k) = p(1-p)^k$	p	
Равномерный	$k = 1, 2, \dots, n$	$P(k) = \frac{1}{n}$	n	
Полюса	$k = 0, 1, 2, \dots$ при $\alpha = 0$; $k = 1, 2, \dots$ при $\alpha > 0$	$P(k) = P_0 \left(\frac{\lambda}{1 + \alpha \lambda} \right)^k \times \frac{1}{1 + \alpha} \dots \frac{1}{1 + (k-1)\alpha} \times \frac{1}{k!},$ $\alpha \geq 0, \lambda > 0$ $P_0 = P(0) = (1 + \alpha \lambda)^{-1/\alpha}$	α, λ	

Законы распределения непрерывной случайной величины

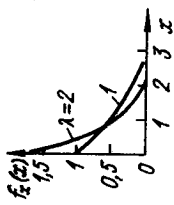
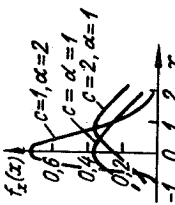
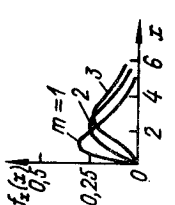
Закон распределения	Область значений случайной величины	Аналитическое выражение плотности вероятности $f(x)$	Определяющие параметры	График плотности вероятности $f(x)$
Равномерный (прямоугольный)	$a < x < b$	$\frac{1}{b-a}$	a, b	
Гауссовский (нормальный)	$-\infty < x < \infty$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right]$	m, σ	
Усеченный гауссовский	$a < x < b$	$\frac{c}{6\sqrt{2\pi}} \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right],$ $c = \frac{1}{\Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)}$	m, σ a, b	

<p>Логарифмически-гауссовский</p>	$0 < x < \infty$	$\frac{\log e}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\log x - m)^2}{2\sigma^2}\right]$ $m = M(\log X)$ $\sigma^2 = D(\log x)$	<p>m, σ</p>	
<p>χ^2-распределение</p>	$0 < x < \infty$	$\frac{1}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}$	<p>n</p>	
<p>Гамма-распределение</p>	$0 < x < \infty$	$\frac{1}{\beta^{\alpha+1} \Gamma(\alpha+1)} x^{\alpha} e^{-\frac{x}{\beta}}$	<p>α, β $\alpha > -1,$ $\beta > 0$</p>	

Закон распределения	Область значений случайной величины	Аналитическое выражение плотности вероятности $f(x)$	Определяющие параметры	График плотности вероятности $f(x)$
Х-распределение	$0 < x < \infty$	$\frac{1}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)} x^{n-1} e^{-\frac{x^2}{2}}$	n	
Коши	$-\infty < x < \infty$	$\frac{1}{\pi} \frac{h}{h^2 + (x - x_0)^2}$	h, x_0	
Релея	$0 < x < \infty$	$\frac{x}{\sigma^2} e^{-x^2/2\sigma^2}$	σ	

Обобщенный закон Релея (Райса)	$0 < x < \infty$	$\frac{x}{\sigma^2} \exp\left(-\frac{x^2 + a^2}{2\sigma^2}\right) I_0\left(\frac{ax}{\sigma^2}\right)$	a, σ	
Максвелла	$0 < x < \infty$	$\sqrt{\frac{2}{\pi}} \frac{1}{\sigma^3} x^2 e^{-x^2/2\sigma^2}$	σ	
Стьюдента (t-распределение)	$-\infty < x < \infty$	$\frac{\Gamma\left(\frac{k+1}{2}\right)}{\sqrt{k\pi}\Gamma\left(\frac{k}{2}\right)} \times \left(1 + \frac{x^2}{k}\right)^{-\frac{k+1}{2}}$	k	

Закон распределения	Область значений случайной величины	Аналитическое выражение плотности вероятности $f(x)$	Определяющие параметры	График плотности вероятности $f(x)$
Вейбулла	$0 < x < \infty$	$cx^\alpha - 1 - e^{-cx^\alpha}$	$c > 0$ $\alpha > 0$	
Фишера — Снедекора (F-распределение)	$0 < x < \infty$	$\frac{\Gamma\left(\frac{n_1+n_2}{2}\right)}{\Gamma\left(\frac{n_1}{2}\right)\Gamma\left(\frac{n_2}{2}\right)} \left(\frac{n_1}{n_2}\right)^{\frac{n_1}{2}} x^{\frac{n_1}{2}-1} \times$ $\times \left(1 + \frac{n_1}{n_2}x\right)^{-\frac{n_1+n_2}{2}}$	n_1, n_2	
Лапласа (двухсторонний экспоненциальный)	$-\infty < x < \infty$	$\frac{\lambda}{2} e^{-\lambda x-\mu }$	λ, μ	

<p>Экспоненциальный односторонний (показательный)</p>	$0 < x < \infty$	$\lambda e^{-\lambda x}$	λ	
<p>Двойное показательное распределение</p>	$-\infty < x < \infty$	$cde^{-dx} - ce^{-\alpha x}$	$c > 0$ $\alpha > 0$	
<p>Показательно-степенной</p>	$0 < x < \infty$	$\frac{x^m}{m!} e^{-x}$	m	

Продолжение табл.

Закон распределения	Область значений случайной величины	Аналитическое выражение плотности вероятности $f(x)$	Определяющие параметры	График плотности вероятности $f(x)$
Симпсона (треугольный)	$a < x < b$	$\begin{cases} 0, & -\infty < x < a, \\ \frac{4(x-a)}{(b-a)^2}, & a < x < \frac{a+b}{2}, \\ \frac{4(b-x)}{(b-a)^2}, & \frac{a+b}{2} < x < b, \\ 0, & b < x < \infty \end{cases}$	a, b	
Арксинуса	$-a < x < a$	$\frac{1}{\pi \sqrt{a^2 - x^2}}$	a	
Тихонова	$-\pi < x < \pi$	$\frac{1}{2\pi I_0(D)} e^{D \cos x}$	D	

Приложение 2

Значения для доверительного интервала $-e < t < e$, где величина t имеет распределение Стьюдента в зависимости от доверительной вероятности P и числа степеней свободы k

k	$P=0,90$	$P=0,95$	$P=0,99$	k	$P=0,90$	$P=0,95$	$P=0,99$
1	6,31	12,71	63,7	16	1,746	2,12	2,92
2	2,92	4,30	9,92	17	1,740	2,11	2,90
3	2,35	3,18	5,84	18	1,734	2,10	2,86
4	2,13	2,77	4,60	19	1,729	2,09	2,86
5	2,02	2,57	4,03	20	1,725	2,08	2,84
6	1,943	2,45	3,71	22	1,717	2,07	2,82
7	1,895	2,36	3,50	24	1,711	2,06	2,80
8	1,860	2,31	3,36	26	1,706	2,06	2,78
9	1,833	2,26	3,25	28	1,701	2,05	2,76
10	1,812	2,23	3,17	30	1,697	2,04	2,75
11	1,796	2,20	3,11	40	1,684	2,02	2,70
12	1,782	2,18	3,06	60	1,671	2,00	2,66
13	1,771	2,16	3,01	120	1,658	1,98	2,62
14	1,761	2,14	2,98		1,645	1,96	2,58
15	1,753	2,13	2,95				

Приложение 3

Критические точки распределения x — случайная величина, распределенная по закону χ^2 с k степенями свободы (таблица содержит значения α , полученные из условия $P(x) \leq \alpha$)

Число степеней свободы k	$\alpha=0,01$	$\alpha=0; 0,025$	$\alpha=0,05$	$\alpha=0,95$	$\alpha=0,975$	$\alpha=0,99$
1	6,6	6,0	3,8	0,0039	0,00098	0,0016
2	9,2	7,4	6,0	0,103	0,051	0,020
3	11,3	9,4	7,8	0,352	0,216	0,115
4	13,3	11,1	9,5	0,711	0,484	0,297
5	15,1	12,8	11,1	1,15	0,831	0,554
6	16,8	14,4	12,6	1,64	1,24	0,872
7	18,5	16,0	14,1	2,17	1,69	1,24
8	20,1	17,5	15,5	2,73	2,18	1,65
9	21,7	19,0	16,9	3,33	2,70	2,09
10	23,2	20,5	18,3	3,94	3,25	2,56
11	24,7	21,9	19,7	4,57	3,82	3,05
12	26,2	23,3	21,0	5,23	4,40	3,57
13	27,7	24,7	22,4	5,89	5,01	4,11
14	29,1	26,1	23,7	6,57	5,63	4,66
15	30,6	27,5	25,0	7,26	6,26	5,23
16	32,0	28,8	26,3	7,96	6,91	5,81

Число степеней свободы k	$\alpha=0,01$	$\alpha=0; 0,025$	$\alpha=0,05$	$\alpha=0,95$	$\alpha=0,975$	$\alpha=0,99$
17	33,4	30,2	27,6	8,67	7,56	6,41
18	34,8	31,5	28,9	9,39	8,23	7,01
19	36,2	32,9	30,1	10,1	8,91	7,63
20	37,6	34,2	31,4	10,9	9,59	8,26
21	38,9	35,5	32,7	11,6	10,3	8,90
22	40,3	36,8	33,9	12,3	11,0	9,54
23	41,6	38,1	35,2	13,1	11,7	10,2
24	43,0	39,4	36,4	13,8	12,4	10,9
25	44,3	40,6	37,7	14,6	13,1	11,5
26	45,6	41,9	38,9	15,4	13,8	12,2
27	47,0	43,2	40,1	16,2	14,6	12,9
28	48,3	44,5	41,3	16,9	15,3	13,6
29	49,6	45,7	42,6	17,7	16,0	14,3
30	50,9	47,0	43,8	18,5	16,8	15,0

Приложение 4

Библиотека стандартных подпрограмм SSP

Операции с полиномами

- PADD — сложение двух полиномов;
 PSUB — вычитание одного полинома из другого;
 PMPY — умножение двух полиномов;
 PDIV — деление одного полинома на другой;
 PCLA — замена одного полинома другим;
 PADDM — сложение одного полинома с другим, умноженным на число;
 PVAL — вычисление значения полинома;
 PVSUB — замена переменной полинома другим полиномом;
 PILD — вычисление значения полинома и его первой производной;
 PDER — производная полинома;
 PINT — интеграл от полинома;
 PQSD — деление полинома на квадратный трехчлен;
 PCLD — вычисление коэффициентов $P(Z - U)$ по заданным коэффициентам $P(Z)$ и U ;
 PGCD — нахождение наибольшего общего делителя двух полиномов;
 PNORM — нормировка вектора коэффициентов полинома;
 PECN, DPECN — экономизация полинома для симметричного интервала;
 PECS, DPECS — экономизация полинома для несимметричного интервала;

Корни полиномов

- POLRT — вычисление действительных и комплексных корней действительного полинома;
 PRQD, DPRQD — вычисление корней полинома с действительными коэффициентами посредством QD-алгоритма со смещением;
 PRBM, DPRBM — вычисление корней полинома с действительными коэффициентами методом Берстоу;
 PQFB, DPQFB — определение квадратичного множителя полинома с действительными коэффициентами;

Полиномы специального вида

- CNP, DCNP — вычисление значений полинома Чебышева;
 CNPS, DCNPS — вычисление значения разложения в ряд по полиномам Чебышева;
 TCNP, DTCNP — преобразование разложения по полиномам Чебышева с независимой переменной X в полином с независимой переменной Z , где $Z = A * X + B$;
 CSP, DCSP — вычисление значений смещенных полиномов;
 CSPS, DCSPS — вычисление разложения в ряд по смещенным полиномам Чебышева;
 TCSP, DTCSP — преобразование разложения по смещенным полиномам Чебышева с независимой переменной X в полином с независимой переменной Z , где $X = A * Z + B$;
 HEP, DHEP — вычисление значения полинома Эрмита;
 HEPS, DHEPS — вычисление значения разложения в ряд по полиномам Эрмита;
 THEP, DTHEP — преобразование разложения по полиномам Эрмита в полином;
 LAP, DLAP — вычисление полиномов Лагерра;
 LAPS, DLAPS — вычисление значения разложения в ряд по полиномам Лагерра;
 TLAP, DTLAP — преобразование разложения по полиномам Лагерра в полином;
 LEP, DLEP — вычисление значений полинома Лежандра;
 LEPS, DLEPS — вычисление значения разложения в ряд по полиномам Лежандра;
 TLEP, DTLEP — преобразование разложения по полиномам Лежандра в полином.

Корни линейных уравнений

- RTWI, DRTWI — уточнение корня итерационным методом Вегстейна;
 RTMI, DRTMI — вычисление корня внутри интервала методом Мюллера;
 RTNI, DRTNI — уточнение корня методом Ньютона;

Экстремумы функции

- FMFP, DFMFP — вычисление локального минимума функции нескольких переменных методом Флетчера.
 FMCG, DFMCG — вычисление безусловного минимума функции нескольких переменных методом сопряженных градиентов.

Численное дифференцирование функций

- DGT3, DDGT3 — дифференцирование функции, заданной таблицей значений в неравноотстоящих точках;
 DET3, DDET3,
 DET5, DDET5 — дифференцирование функции, заданной таблицей значений в равноотстоящих точках;
 DCAR, DDCAR — дифференцирование функции в центре интервала методом экстраполяции Ричардсона и Ромберга;
 DBAR, DDBAR — дифференцирование функции на границе интервала методом экстраполяции Ричардсона и Ромберга;
 RKGS, DRKGS — решение системы обыкновенных дифференциальных уравнений первого порядка с начальными условиями методом Рунге — Кутта;
 HPCG, DHPCG — решение системы обыкновенных дифференциальных уравнений первого порядка с заданными начальными условиями;
 HPCL, DHPCCL — решение системы линейных обыкновенных дифференциальных уравнений первого порядка с заданными начальными условиями;
 LBVP, DLBVP — решение линейных дифференциальных уравнений первого порядка с линейным граничным условием;
 RK1 — интегрирование дифференциального уравнения первого порядка до заданных конечных условий;
 RK2 — интегрирование дифференциального уравнения первого порядка методом Рунге — Кутта с выдачей результатов в табличном виде.

Численное интегрирование функций

- QTFG, DQTFG — интегрирование функции, заданной таблицей значений в неравноотстоящих точках, по методу трапеций;
 QTFE, DQTFE — интегрирование функции, заданной таблицей значений в равноотстоящих точках, по методу трапеций;
 QSF, DQTSF — интегрирование функции, заданной таблицей значений в равноотстоящих точках, по методу Симпсона;

- QHFG, DQHFG — интегрирование функции, заданной таблицей значений функции и ее первой производной в неравноотстоящих точках, по правилу Эрмита первого порядка;
- QHFE, DQHFE — то же по правилу Эрмита первого порядка, но в равноотстоящих точках;
- QHSG, DQHSG — интегрирование функции, заданной таблицей значений функции и ее первой и второй производных, по правилу Эрмита второго порядка в неравноотстоящих точках;
- QHSE, DQHSE — то же по правилу Эрмита второго порядка, но в равноотстоящих точках;
- QATR, DQATR — интегрирование функции по правилу трапеций с экстраполяцией по методу Ромберга;
- QL2, QL3... QL10 — интегрирование функции по квадратурным формулам.
- DQL4, DQL8,
DQL12, DQL16,
DQL24, DQL32 — Гаусса—Лагерра;
- QG2, QG3—QG10 — интегрирование функции по квадратурным формулам Гаусса;
- DQG4, DQG8,
DQG12, DQG16,
DQG24, DQG32 — интегрирование функции по квадратурным формулам Гаусса — Эрмита;
- QH2, QH3...
QH10, DQH8,
DQH16, DQG24,
DQG32, DQH48,
DQH64 — интегрирование функции по присоединенным квадратурным формулам Гаусса — Лагерра;
- QA2, QA3...
QA10, DQA4,
DQA8, DQA12,
DQA16, DQA24,
DQA32

Интерполирование функций

- ALI, DALI — интерполирование функции с помощью интерполяционного процесса Эйткена — Лагранжа;
- ANI, DANI — интерполирование функции с помощью интерполяционного процесса Эйткена — Эрмита;
- ACFI, DACFI — интерполирование функции с помощью непрерывной дроби;
- ATSG, DATSG — выборка таблицы из таблицы общего вида;
- ATSM, DATSM — выборка таблицы из таблицы с монотонными аргументами;
- ATSE, DATSE — выборка таблицы из таблицы с равноотстоящими аргументами.

Сглаживание функций

- SG13 — сглаживание функции, заданной таблицей значений в неравноотстоящих точках, методом наименьших квадратов;

- SE13, DSE13 — сглаживание функции, заданной таблицей значений в равноотстоящих точках, методом наименьших квадратов;
 SE15, DSE15, SE35, DSE35 — сглаживание функции, заданной таблицей значений в равноотстоящих точках, методом наименьших квадратов.

Аппроксимация функций

- APCH, DAPCH — приближение функции методом наименьших квадратов. Составление системы нормальных уравнений, возникающей при приближении дискретной функции многочленом, по методу наименьших квадратов;
 ARAT, DARAT — аппроксимация дискретной функции рациональной функцией, наилучшей в смысле метода наименьших квадратов;
 FRAT, DFRAT — используется для ручной обработки данных и основных функций с рациональной аппроксимацией;
 APLL, DAPLL — составление системы нормальных уравнений, возникающей при аппроксимации дискретной функции линейной комбинацией функций, наилучшей в смысле метода наименьших квадратов;
 APMM, DAPMM — линейная чебышевская аппроксимация на дискретном множестве точек.

Анализ Фурье

- FORIF — вычисление коэффициентов Фурье периодической функции;
 FORIT — вычисление коэффициентов Фурье периодической функции, заданной таблицей значений;
 HARM, DHARM — комплексный трехмерный анализ Фурье;
 RHARM, DRHARM — одномерный действительный анализ Фурье.

Подстановки

- PPRCN — произведение подстановок;
 PERM — операции α подстановками и транспозициями.

Суммы и пределы последовательностей

- TEAS, DTEAS — вычисление предела данной последовательности;
 P TEVL, DTEVL — вычисление суммы функциональной последовательности.

Специальные функции

- GAMMA — вычисление гамма-функции;
 DLGAM — вычисление с удвоенной точностью логарифма гамма-функции;
 BESY — вычисление функции Бесселя;

- INVE — вычисление функции Бесселя N -го порядка;
- IO — вычисление модифицированной функции Бесселя 0-го порядка;
- BESJ — вычисление функции Бесселя;
- BESK — вычисление функции Бесселя;
- EXPI — вычисление показательного интеграла;
- SICI — вычисление интегрального синуса и косинуса;
- CS — вычисление интегралов Френеля;
- CEL1, DCEL1 — вычисление полного эллиптического интеграла;
- CEL2, DCEL2 — вычисление обобщенного полного эллиптического интеграла второго рода;
- ELI1, DELI1 — вычисление эллиптического интеграла первого рода;
- ELI2, DELI2 — вычисление обобщенного эллиптического интеграла второго рода;
- JELF, DJELF — вычисление эллиптических функций Якоби.

Статистика

Предварительная обработка данных

- TALLY — суммы, средние, стандартные отклонения, минимумы, максимумы;
- BOUND — отбор наблюдений, находящихся в заданных границах;
- SUBST — выборка последовательности из матрицы наблюдений;
- ABSNT — определение пропущенных данных;
- TAB1 — табулирование данных (одна переменная);
- TAB2 — табулирование данных (две переменных);
- SUBMX — построение сокращенной матрицы.

Корреляция и регрессия
(сглаживание, факторизация)

- CORRE — средние, стандартные (средние квадратические) отклонения и корреляция;
- MISR — средние, стандартные отклонения, третьи и четвертые моменты корреляции, коэффициенты простой регрессии и их стандартные ошибки, выявление возможных пропусков данных;
- BISER — вычисление двухсерийного коэффициента корреляции между двумя непрерывными переменными, одна из которых искусственно дихотомизированна;
- PHI — вычисление коэффициента корреляции между двумя дихотомизированными переменными;

Продолжение прил. 4

- POINT — вычисление точечного двухсерийного коэффициента корреляции между двумя переменными, когда одна из переменных разделена на две части, а другая — непрерывна;
- TETRA — вычисление тетракорического коэффициента корреляции между двумя переменными, когда данные обеих переменных подразделены на два класса;
- ORDER — перегруппировка взаимной корреляции;
- MULTR — множественная линейная регрессия;
- GDATA — образование матрицы данных для полиномиальных регрессий;
- STPRG — пошаговая множественная линейная регрессия;
- PROBT — вероятностный анализ;
- MOMEN — отыскание первых четырех моментов для группированных данных на равных интервалах классификации;
- CANOR — каноническая корреляция.

Планируемый анализ

- AVDAT — распределение данных в памяти;
- DATA — считывание данных с устройства ввода;
- AVCAL, MEANQ — операции со средними квадратами.

Дискриминантный анализ

- DMATX — вычисление значений переменных в каждой группе и дисперсии матрицы для всех групп;
- DISCR — дискриминантный анализ.

Факторный анализ

- IRACE — накапливаемая процентовка собственных значений;
- LOAD — введение фактора;
- VARMX — варимаксное вращение.

Временные ряды (сглаживание)

- AUTO — автоковариация;
- CROSS — взаимные ковариации;
- SMO — применение коэффициентов фильтра (весов);
- EXSMO — тройная степень фильтрации;
- NROOT — собственные значения и собственные векторы специальной матрицы.

Непараметрические статистики

- KOLMO — проверка Колмогорова — Смирнова одной выборки;
- KOLM2 — проверка Колмогорова — Смирнова двух выборок;

- SMIRN — значение распределений пределов Колмогорова — Смирнова;
 CHISQ — проверка для таблиц смежности;
 KRANK — ранговая корреляция по Кендэлу;
 MPAIR — тест рангов значимости по Уилкоксоу;
 QTEST — Q тест Кокрена;
 TTEST — проверка среднего в совокупности;
 RANK — ранжирование наблюдений;
 SIGNT — знаковый тест;
 SRANK — ранговая корреляция по Спирмэну;
 TIE — подсчет связей в ранжируемых наблюдениях;
 TWOAV — дисперсионный статистический анализ по Фридману;
 UTEST — U тест по Манну—Уитнею;
 WTEST — коэффициенты согласия по Кендэлу;
 SPATE — вычисление коэффициентов дожития.

Генераторы случайных переменных и функции распределения

- RANDU — равномерное распределение случайных чисел;
 GAUSS — нормальное распределение;
 NDTR — функция нормального распределения;
 CDTR — функция распределения хи-квадрат;
 NDTRI — обратная функция нормального распределения.

Матричная и линейная алгебра

Х р а н е н и е м а т р и ц ы

- MCPY — запись матрицы;
 RCPY — запись строк матрицы в виде вектора;
 CCPY — запись столбцов матрицы в виде вектора;
 DCPY — запись диагонали матрицы в виде вектора;
 XCPY — выбор и запись подматрицы из заданной матрицы;
 MSTR — изменение способа хранения;
 LOC — вычисление индекса в уплотненно хранимой матрице;
 CONVT — преобразование числа из одинарной точности в двойную и наоборот;
 ARRAY — преобразование массива, хранящегося векторным способом, в матричный способ хранения.

О п е р а ц и и

- GMADD, MADD — сложение двух матриц;
 MSUB, GMSUB — вычитание двух матриц;
 MPRD, GMPRD — умножение двух матриц;
 TPRD, GTPRD — транспонирование произведения двух матриц;
 MATA — транспонирование произведения матрицы самой на себя;

- SADD — сложение матрицы со скаляром;
 SSUB — вычитание скаляра из матрицы;
 SMPY — умножение матрицы на скаляр;
 SDIV — деление матрицы на скаляр;
 GMTRA — транспонирование матрицы при общем способе хранения;
 MTRA — транспонирование матрицы;
 SCLA — очистка матрицы и сложение со скаляром (засылка скаляра на место всех элементов матрицы);
 DCLA — замена диагональных элементов скаляром;
 RADD — сложение строки одной матрицы со строкой другой матрицы;
 CADD — сложение столбца одной матрицы со столбцом другой матрицы;
 SRMA — умножение строки матрицы на скаляр и сложение с другой строкой;
 SCMA — умножение столбца матрицы на скаляр и сложение с другим столбцом той же матрицы;
 RINT — перестановка двух строк;
 CINT — перестановка двух столбцов;
 RSUM — сложение строк матрицы;
 CSUM — сложение столбцов матрицы;
 RTAB — табулирование строк матрицы;
 CTAB — табулирование столбцов матрицы;
 RSRT — сортировка строк матрицы;
 CSRT — сортировка столбцов матрицы;
 RCUT — разделение по строкам;
 CCUT — разделение по столбцам;
 RTIE — объединение двух матриц по строкам;
 CTIE — объединение двух матриц по столбцам;
 MPRC, DMPRC — перестановка строк или столбцов;
 MFUN — преобразование матрицы при помощи функции;
 RFCP — обратная функция в MFVN.

Обращение матрицы.

Системы линейных уравнений.

- MINV — обращение матрицы;
 SINV, DSINV — обращение симметрической положительно определенной матрицы;
 SIMQ — решение системы совместных алгебраических линейных уравнений;
 GELB, DGELB — решение системы совместных линейных уравнений общего вида с диагональной матрицей, исключение Гаусса;
 GELG, DGELG — решение системы совместных линейных уравнений общего вида методом Гаусса;
 RSLMC — решение системы совместных линейных уравнений итеративным методом;
 FACTR — треугольная факторизация несингулярной матрицы;

- MFGR, DMFGR — факторизация матрицы и определение ранга;
GELS, DGELS — решение системы линейных уравнений общего вида с симметрической матрицей коэффициентов;
MTDS, DMTDS — деление матрицы на треугольную матрицу;
MLSS, DMLSS — решение системы совместных линейных уравнений с симметрической положительно определенной матрицей;
MCHB, DMCHB — треугольная факторизация симметрической положительно определенной ленточной матрицы;
MFSS, DMFSS — треугольная факторизация и определение ранга симметрической положительно полуопределенной матрицы;
MFSD, DMFSD — треугольная факторизация симметрической положительно определенной матрицы;
LLSQ, DLLSQ — решение линейных задач методом наименьших квадратов.

Анализ собственных значений

- EIGEN — собственные значения и собственные векторы действительной матрицы;
NROOT — собственные значения и собственные векторы специальной несимметрической матрицы;
ATEIG — собственные значения действительной почти треугольной матрицы;
HSBG — приведение действительной матрицы в почти треугольную форму.

Решение систем уравнений

- MATIN — ввод матрицы;
MXOUT — вывод матрицы.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. *Абрамовиц М., Стиган И.* Справочник по специальным функциям.— М.: Наука, 1979.— 830 с.
2. *Арайс Е. А., Яковлев Н. А.* Автоматизация аналитических вычислений в научных исследованиях.— Новосибирск : Наука, 1985.— 222 с.
3. *Бахвалов Н. С., Жидков Н. П., Кобельков Г. М.* Численные методы.— М.: Наука, 1987.— 598 с.
4. *Вероятностные методы в вычислительной технике /* Под ред. А. Н. Лебедева, Е. А. Чернявского.— М.: Высшая шк., 1986.— 312 с.
5. *Воеводин В. В.* Матрицы и вычисления.— М.: Наука, 1984.— 318 с.
6. *Галлагер Р.* Метод конечных элементов. Основы.— М.: Мир, 1984.— 428 с.
7. *Демидович Б. П., Марон И. А.* Основы вычислительной математики.— М.: Наука, 1970.— 664 с.
8. *Икрамов К. Д.* Численное решение матричных задач. Ортогональные методы.— М.: Наука, 1984.— 190 с.
9. *Кветный Р. Н., Маликов В. Т.* Информационная теория измерений — от модели к изделию.— М.: Знание, 1988.— 32 с.
10. *Кнут Д.* Искусство программирования на ЭВМ.— М.: Мир, 1978. Т. 1, 2.— 735 с., 726 с.
11. *Корн Г., Корн Т.* Справочник по математике для инженеров и научных работников.— М.: Наука, 1974.— 830 с.
12. *Краскевич В. Е., Зеленский К. Х., Гречко В. И.* Численные методы в инженерных исследованиях.— К.: Вища шк., 1986.— 186 с.
13. *Мак-Кракен Д., Дорн У.* Численные методы и программирование на ФОРТРАНЕ.— М.: Мир, 1977.— 584 с.
14. *Маликов В. Т., Дубовой В. М., Кветный Р. Н. и др.* Анализ измерительных информационных систем.— Ташкент : ФАН, 1984.— 176 с.
15. *Марчук Г. И., Агошков В. И.* Введение в проекционно-сеточные методы.— М.: Наука, 1982.— 476 с.
16. *Основы моделирования сложных систем /* Под общ. ред. И. В. Кузьмина.— К., Вища шк., 1981.— 359 с.
17. *Ракитский Ю. В., Устинов С. М., Черноруцкий И. Г.* Численные методы решения жестких систем.— М.: Наука, 1979.— 208 с.
18. *Самарский А. А.* Введение в численные методы.— М.: Наука, 1987.— 286 с.
19. *Самарский А. А.* Теория разностных схем.— М.: Наука, 1977.— 656 с.

20. Сборник научных программ на ФОРТРАНЕ.— М.: Статистика, 1974.— 316 с.

21. Советов Б. Я., Яковлев С. А. Моделирование систем.— М.: Высшая шк., 1985.— 271 с.

22. Форсайт Дж., Малькольм М., Моулдер К. Машинные методы математических вычислений.— М.: Мир, 1980.— 279 с.

23. Шуп Т. Решение инженерных задач на ЭВМ.— М.: Мир, 1982.— 235 с.

Учебное пособие

Маликов Виктор Тихонович
Кветный Роман Наумович

**ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ
И ПРИМЕНЕНИЕ ЭВМ**

Художественный редактор *С. П. Духленко*
Обложка художника *Н. Г. Колбасовой*
Технический редактор *Н. Н. Горбунова*
Корректор *И. Е. Бей*

ИБ № 12844

Сдано в набор 23.09.88. Подписано в печать
06.07.89. БФ 05090. Формат 84×108¹/₃₂. Бум. тип.
№ 2. Гарнитура литературная. Высокая печать.
Усл. печ. л. 11,34. Усл. кр.-отг. 11,55. Уч.-изд.

л. 9,87. Тираж 12 000 экз. Изд. № 8134. Зак. 9-82.
Цена 50 к.

Главное издательство издательского объединения
«Выща школа», 252054, Киев-54, ул. Гоголев-
ская, 7.

Книжная фабрика имени М. В. Фрунзе, 310057,
Харьков-57, ул. Донец-Захаржевского, 6/8.

В Головном издательстве издательского объединения «Выща школа» в 1990 году выйдут в свет новые учебные пособия:

Згуровский М. З. Интегрированные системы оптимального управления и проектирования.

28 л., ил. Язык русский. 1 р. 30 к.

Рассмотрены машинно-ориентированные методы оптимизации процессов управления и проектирования промышленных систем распределенного характера, основанных на фундаментальных явлениях нестационарной диффузии, тепломассообмена и колебательных процессах, широко распространенных в энергетике, металлургии, механике твердых тел, нефтяной, химической промышленности, в области охраны окружающей среды. Эти методы характеризуются повышенной степенью унификации, типизации вычислительных процедур и пониженными требованиями к машинным ресурсам. Поэтому их можно использовать в задачах автоматизированного проектирования математического обеспечения сложных технических систем и процессов.

Для студентов вузов, обучающихся по специальностям «Прикладная математика», «Автоматика и управление в технических системах».

Аннотировалось в ТП 1990 г., поз. 62.

Основы системного анализа и проектирования АСУ. Руководитель авт. кол. А. А. Павлов.

20 л., ил. Язык русский. 1 р.

Изложение вопроса анализа и проектирования производственных систем осуществлено с единых методологических позиций, базирующихся на системном анализе. Рассмотрена современная технология производственных процессов, ориентированная на использование гибких производственных систем и интегрированных автоматизированных систем их управления, а также технология проектирования основных компонент интегрированных систем управления производством. Описан математико-операционный аппарат реализации функций управления.

Для студентов вузов, обучающихся по специальностям «Автоматизированные системы управления», «Автоматика и управление в технических системах», «Прикладная математика», «Экономическая кибернетика».

Аннотировалось в ТП 1990 г., поз. 116.

Уважаемые товарищи!

Эти книги можно заказать в магазинах облкниготоргов, облпотребсоюзов, а также в специализированном магазине «Книга — почтой» (252117, г. Киев-117, ул. Попудренко, 26).